Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

The Kernel Kalman Rule — Efficient Nonparametric Inference with Recursive Least Squares

Gregor H. W. Gebhardt
Technische Universität Darmstadt
Hochschulstr. 10
64289 Darmstadt, Germany
gebhardt@ias.tu-darmstadt.de

Andras Kupcsik
School of Computing
National University of Singapore
13 Computing Drive, Singapore 117417
kupcsik@comp.nus.edu.sg

Gerhard Neumann
School of Computer Science
University of Lincoln
Lincoln, LN6 7TS, UK
gneumann@lincoln.ac.uk

Abstract

Nonparametric inference techniques provide promising tools for probabilistic reasoning in high-dimensional nonlinear systems. Most of these techniques embed distributions into reproducing kernel Hilbert spaces (RKHS) and rely on the kernel Bayes' rule (KBR) to manipulate the embeddings. However, the computational demands of the KBR scale poorly with the number of samples and the KBR often suffers from numerical instabilities. In this paper, we present the *kernel Kalman rule* (KKR) as an alternative to the KBR. The derivation of the KKR is based on recursive least squares, inspired by the derivation of the Kalman innovation update. We apply the KKR to filtering tasks where we use RKHS embeddings to represent the belief state, resulting in the *kernel Kalman filter* (KKF). We show on a nonlinear state estimation task with high dimensional observations that our approach provides a significantly improved estimation accuracy while the computational demands are significantly decreased.

1 Introduction

State estimation and prediction for continuous partially observable stochastic processes is a significant challenge in many high-dimensional applications, such as robotics. However, analytical solutions are only applicable for a limited set of models with special structure. When assuming linear models with Gaussian noise, for instance, the Kalman filter (Kalman 1960) is known to provide an optimal solution. For more complex models, approximate solutions have to be used instead (McElhoe 1966; Smith, Schmidt, and McGee 1962; Julier and Uhlmann 1997; Wan and Van Der Merwe 2000). These approximations are again difficult to scale to high-dimensional problems, they often assume a unimodal Gaussian observation prior and also require that the model of the system is known.

The recently introduced methods for nonparametric inference (Song, Fukumizu, and Gretton 2013; Fukumizu, Song, and Gretton 2013) alleviate the problems of traditional state estimation methods for nonlinear systems. The idea of these methods is to represent probability distributions as points in reproducing kernel Hilbert spaces. Based on the kernelized versions of the sum rule, the chain rule, and the Bayes' rule, inference can be performed entirely in the RKHS. Additionally, Song, Fukumizu, and Gretton (2013) use the kernel sum rule and the kernel Bayes' rule to construct the kernel Bayes' filter (KBF). The KBF learns the transition and observation models from observed samples and can be applied to nonlinear systems with high-dimensional observations. However, the computational complexity of the KBR update scales poorly with the number of samples such that hyper-parameter optimization becomes prohibitively expensive. Moreover, the KBR requires mathematical tricks that may cause numerical instabilities and also render the objective which is optimized by the KBR unclear.

In this paper, we present the *kernel Kalman rule* (KKR) as alternative to the kernel Bayes' rule. Our derivations closely follow the derivations of the innovation update used in the Kalman filter and are based on a recursive least squares minimization objective in a reproducing kernel Hilbert space. We show that our update of the mean embedding is unbiased and has minimum variance. While the update equations are formulated in a potentially infinite dimensional RKHS, we derive, through the application of the kernel trick and by virtue of the representer theorem (Schölkopf, Herbrich, and Smola 2001), an algorithm using only products and inversions of finite kernel matrices. We employ the kernel Kalman rule together with the kernel sum rule for filtering, which results in the kernel Kalman filter (KKF). In contrast to filtering techniques that rely on the KBR, the KKF allows to precompute expensive matrix inversions which significantly reduces the computational complexity. This allows us also to apply hyper-parameter optimization for the KKF.

To scale gracefully with larger data sets, we rederive the KKR and the KKF with the subspace conditional operator (Gebhardt, Kupcsik, and Neumann 2015). Here, only a subset of the samples is used to span a feature space, while all training samples are used to estimate the models.

We compare our approach to different versions of the KBR and demonstrate its improved estimation accuracy and computational efficiency. Furthermore, we evaluate the KKR on a simulated 4-link pendulum task and a human motion capture data set (Wojtusch and von Stryk 2015).

1.1 Related Work

To the best of our knowledge, the kernel Bayes' rule exists in three different versions. It was first introduced in its

original version by Fukumizu, Song, and Gretton (2013). Here, the KBR is derived, similar to the conditional operator, using prior modified covariance operators. These prior-modified covariance operators are approximated by weighting the feature mappings with the weights of the embedding of the prior distribution. Since these weights are potentially negative, the covariance operator might become indefinite which renders its inversion impossible. To overcome this drawback, the authors have to apply a form of the Tikhonov regularization that significantly decreases accuracy and increases the computational costs. A second version of the KBR was introduced by Song, Fukumizu, and Gretton (2013) in which they use a different approach to approximate the prior-modified covariance operator. In the experiments conducted for this paper, this second version often leads to more stable algorithms than the first version. Boots, Gretton, and Gordon (2013) introduced a third version of the KBR where they apply only the simple form of the Tikhonov regularization. However, this rule requires the inversion of a matrix that is often indefinite, and therefore, high regularization constants are required, which again degrades the performance. In our experiments, we refer to these different versions with KBR(b) for the first, KBR(a) for the second (both adapted from the literature), and KBR(c) for the third version.

For filtering tasks with known linear system equations and Gaussian noise, the Kalman filter (KF) yields the solution that minimizes the squared error of the estimate to the true state. Two widely known and applied approaches to extend the Kalman filter to non-linear systems are the *extended Kalman filter* (EKF) (McElhoe 1966; Smith, Schmidt, and McGee 1962) and the *unscented Kalman filter* (UKF) (Wan and Van Der Merwe 2000; Julier and Uhlmann 1997). Both, the EKF and the UKF, assume that the non-linear system dynamics are known and use them to update the prediction mean. Yet, updating the prediction covariance is not straightforward. In the EKF the system dynamics are linearized at the current estimate of the state, and in the UKF the covariance is updated by applying the system dynamics to a set of sample-points (sigma points). While these approximations make the computations tractable, they can significantly reduce the quality of the state estimation, in particular for high-dimensional systems.

Hsu, Kakade, and Zhang (2012) recently proposed an algorithm for learning Hidden Markov Models (HMMs) by exploiting the spectral properties of observable measures to derive an observable representation of the HMM (Jaeger 2000). An RKHS embedded version thereof was presented in (Song et al. 2010). While this method is applicable for continuous state spaces, it still assumes a finite number of discrete hidden states.

Other closely related algorithms to our approach are the kernelized version of the Kalman filter by Ralaivola and d'Alche Buc (2005) and the kernel Kalman filter based on the conditional embedding operator (KKF-CEO) by Zhu, Chen, and Principe (2014). The former approach formulates the Kalman filter in a sub-space of the infinite feature space that is defined by the kernels. Hence, this approach does not fully leverage the kernel idea of using an infinite feature space. In contrast, the KKF-CEO approach embeds the belief state also in an RKHS. However, they require that the observation is a noisy version of the full state of the system, and thus, they cannot handle partial observations. Moreover, they also deviate from the standard derivation of the Kalman filter, which, as our experiments show, decreases the estimation accuracy. The full observability assumption is needed in order to implement a simplified version of the innovation update of the Kalman filter in the RKHS. The KKF does not suffer from this restriction. It also provides update equations that are much closer to the original Kalman filter and outperforms the KKF-CEO algorithm as shown in our experiments.

2 Preliminaries

Our work is based on embeddings of probability densities into reproducing kernel Hilbert spaces (RKHS). For a detailed survey see the work of Song, Fukumizu, and Gretton (2013). An RKHS $\mathcal{H}_k$ is a Hilbert space associated with an inner product $\langle \cdot, \cdot \rangle$ that is implicitly defined by a kernel function $k : \Omega \times \Omega \to \mathbb{R}$ as $\langle \boldsymbol{\varphi}(\boldsymbol{x}), \boldsymbol{\varphi}(\boldsymbol{x}') \rangle := k(\boldsymbol{x}, \boldsymbol{x}')$. Here, $\boldsymbol{\varphi}(\boldsymbol{x})$ is a feature mapping into a possibly infinite dimensional space, intrinsic to the kernel function. The kernel function k satisfies the reproducing property (Aronszajn 1950), i.e., $f(\boldsymbol{x}) = \langle f, \boldsymbol{\varphi}(\boldsymbol{x}) \rangle$ for any $f \in \mathcal{H}_k$.

A marginal density $P(X)$ over the random variable X can be embedded as the expected feature mapping (or mean map) $\boldsymbol{\mu}_X := \mathbb{E}_X [\boldsymbol{\varphi}(X)]$ (Smola et al. 2007). Using a finite set of samples from $P(X)$, the mean map can be estimated as

$$\hat{\boldsymbol{\mu}}_X = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\varphi}(\boldsymbol{x}_i) = \frac{1}{m} \boldsymbol{\Upsilon}_x^\intercal \mathbf{1}_m, \tag{1}$$

where $\boldsymbol{\Upsilon}_x = [\boldsymbol{\varphi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_m)]$ is a matrix consisting of the feature mappings of the samples and $\mathbf{1}_m \in \mathbb{R}^m$ is an m dimensional all-ones vector. Alternatively, a distribution can be embedded in a tensor product RKHS $\mathcal{H}_k \times \mathcal{H}_k$ as the expected tensor product of the feature mappings, i.e., $\mathcal{C}_{XX} := \mathbb{E}_{XX} [\boldsymbol{\varphi}(X) \otimes \boldsymbol{\varphi}(X)] - \boldsymbol{\mu}_X \otimes \boldsymbol{\mu}_X$ (Smola et al. 2007). This embedding is also called the centered covariance operator. The finite sample estimator is given by

$$\hat{\mathcal{C}}_{XX} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\varphi}(\boldsymbol{x}_i) \otimes \boldsymbol{\varphi}(\boldsymbol{x}_i) - \hat{\boldsymbol{\mu}}_X \otimes \hat{\boldsymbol{\mu}}_X. \tag{2}$$

Similarly, we can define the (uncentered) cross-covariance operator for a joint distribution $p(X, Y)$ of two variables X and Y as $\hat{\mathcal{C}}_{XY} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\varphi}(x_i) \otimes \boldsymbol{\phi}(y_i)$. Here, $\boldsymbol{\phi}(\cdot)$ is the intrinsic feature mapping of another kernel function defined on the Cartesian product of the domain of $\boldsymbol{y}$.

The embedding of a conditional distribution $P(Y|X)$ is defined as a conditional embedding operator $\mathcal{C}_{Y|X}$ that satisfies $\boldsymbol{\mu}_{Y|\boldsymbol{x}} := \mathbb{E}_{Y|\boldsymbol{x}} [\boldsymbol{\phi}(Y)] = \mathcal{C}_{Y|X} \boldsymbol{\varphi}(\boldsymbol{x})$ (Song, Fukumizu, and Gretton 2013). Given a finite set of samples, the conditional embedding operator can be estimated as

$$\hat{\mathcal{C}}_{Y|X} = \boldsymbol{\Phi}_y (\mathbf{K}_{xx} + \lambda \boldsymbol{I}_m)^{-1} \boldsymbol{\Upsilon}_x^\intercal, \tag{3}$$

with the feature mappings $\boldsymbol{\Phi}_y := [\boldsymbol{\phi}(\boldsymbol{y}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{y}_m)]$, the Gram matrix $\boldsymbol{K}_{xx} = \boldsymbol{\Upsilon}_x^\intercal \boldsymbol{\Upsilon}_x \in \mathbb{R}^{m \times m}$, the regularization parameter λ, and the identity matrix $\boldsymbol{I}_m \in \mathbb{R}^{m \times m}$.

The kernel Bayes' rule (KBR) infers the mean embedding $\boldsymbol{\mu}^{\pi}_{X|y}$ of the posterior distribution $Q(X|Y) = P(Y|X)\pi(X)/\sum_X P(Y|X)\pi(X)$. It uses the kernel sum and the kernel chain rule to obtain the prior-modified covariance operators $\mathcal{C}^{\pi}_{YY}$ and $\mathcal{C}^{\pi}_{YX}$. For a more detailed discussion of these operators, we refer to Song, Fukumizu, and Gretton (2013). There exist, to the authors' knowledge, three different versions of the KBR. In the first version of the KBR, Fukumizu, Song, and Gretton (2013) used the tensor product conditional operator in the kernel chain rule, i.e., $\hat{\boldsymbol{\mu}}^{\pi}_{X|y} = \boldsymbol{\Upsilon}_x \boldsymbol{D}\boldsymbol{G}_{yy}\left((\boldsymbol{D}\boldsymbol{G}_{yy})^2 + \kappa \boldsymbol{I}_m\right)^{-1}\boldsymbol{D}\boldsymbol{g}_y$, with the diagonal $\boldsymbol{D} := \mathrm{diag}((\boldsymbol{K}_{xx} + \lambda \boldsymbol{I}_m)^{-1}\boldsymbol{K}_{xx}\boldsymbol{\alpha})$, the Gram matrix $\boldsymbol{G}_{yy} = \boldsymbol{\Phi}_y^{\intercal}\boldsymbol{\Phi}_y$, the kernel vector $\boldsymbol{g}_y$ and κ as regularization parameter. Song, Fukumizu, and Gretton (2013) additionally derived the KBR, using an alternative formulation of the kernel chain rule, as $\hat{\boldsymbol{\mu}}^{\pi}_{X|y} = \boldsymbol{\Upsilon}_x \boldsymbol{\Lambda}^{\intercal}\left((\boldsymbol{D}\boldsymbol{G}_{yy})^2 + \kappa \boldsymbol{I}_m\right)^{-1}\boldsymbol{G}_{yy}\boldsymbol{D}\boldsymbol{g}_y$, with $\boldsymbol{\Lambda} := (\boldsymbol{K}_{xx} + \lambda \boldsymbol{I}_m)^{-1}\boldsymbol{K}_{xx}\mathrm{diag}(\boldsymbol{\alpha})$. As the matrix $\boldsymbol{D}\boldsymbol{G}_{yy}$ is typically not invertible, both of these versions of the KBR use a form of the Tikhonov regularization in which the matrix in the inverse is squared. Boots, Gretton, and Gordon (2013) use a third form of the KBR which is derived analogously to the first version but does not use the squared form of the Tikhonov regularization, i.e., $\hat{\boldsymbol{\mu}}^{\pi}_{X|y} = \boldsymbol{\Upsilon}_x\left(\boldsymbol{D}\boldsymbol{G}_{yy} + \kappa \boldsymbol{I}_m\right)^{-1}\boldsymbol{D}\boldsymbol{g}_y$. Since the product $\boldsymbol{D}\boldsymbol{G}_{yy}$ is often not positive definite, a strong regularization parameter is needed to make the matrix invertible.

All three versions of the kernel Bayes' rule presented above have drawbacks. First, due to the approximation of the prior modified covariance operators, these operators are not guaranteed to be positive definite and, thus, their inversion requires either a harsh form of the Tikhonov regularization or a strong regularization factor and are often still numerically instable. Furthermore, the inverse is dependent on the embedding of the prior distribution and, hence, needs to be recomputed for every Bayesian update of the mean map. This recomputation significantly increases the computational costs, for example, if we want to apply hyperparameter optimization techniques.

3 The Kernel Kalman Rule

In this section, we will present a new approach to Bayesian reasoning in Hilbert spaces inspired by recursive least squares estimation (Gauss 1823; Sorenson 1970; Simon 2006). That is, we want to infer the embedding of a distribution $\boldsymbol{\mu}^{+}_{X,t} = \mathbb{E}_{X_t|\boldsymbol{y}_{1:t}}[\boldsymbol{\varphi}(X)]$ a-posteriori to a new measurement y_t, given the embedding of the distribution $\boldsymbol{\mu}^{-}_{X,t} = \mathbb{E}_{X_t|\boldsymbol{y}_{1:t-1}}[\boldsymbol{\varphi}(X)]$ a-priori to the measurement. We assume a conditional embedding operator $\mathcal{C}_{Y|X}$ of the distribution $P(Y|X)$ as given. Usually, this conditional operator can be estimated from a training set of state-measurement-pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{m}$.

3.1 Recursive Least Squares for Estimating the Posterior Embedding

The derivations for the *kernel Kalman rule* (KKR) are inspired by the ansatz from recursive least squares. Hence, the objective of the KKR is to find the mean embedding $\boldsymbol{\mu}_X$ that minimizes the squared error

$$\sum_t \left(\boldsymbol{\phi}(\boldsymbol{y}_t) - \mathcal{C}_{Y|X}\boldsymbol{\mu}_X\right)^{\intercal}\mathcal{R}^{-1}\left(\boldsymbol{\phi}(\boldsymbol{y}_t) - \mathcal{C}_{Y|X}\boldsymbol{\mu}_X\right), \quad (4)$$

with the metric $\mathcal{R}$, in an iterative fashion. In each iteration, we want to update the prior mean map $\boldsymbol{\mu}^{-}_{X,t}$ based on the measurement $\boldsymbol{y}_t$ to obtain the posterior mean map $\boldsymbol{\mu}^{+}_{X,t}$.

From the recursive least squares solution, we know that the update rule for obtaining the posterior mean map $\boldsymbol{\mu}^{+}_{X,t}$ is

$$\boldsymbol{\mu}^{+}_{X,t} = \boldsymbol{\mu}^{-}_{X,t} + \mathcal{Q}_t(\boldsymbol{\phi}(\boldsymbol{y}_t) - \mathcal{C}_{Y|X}\boldsymbol{\mu}^{-}_{X,t}), \quad (5)$$

where $\mathcal{Q}_t$ is the Hilbert space Kalman gain operator that is applied to the correction term $\boldsymbol{\delta}_t = \boldsymbol{\phi}(\boldsymbol{y}_t) - \mathcal{C}_{Y|X}\boldsymbol{\mu}^{-}_{X,t}$.

We denote the error of the a-posteriori estimator to the embedding of the true state as $\boldsymbol{\varepsilon}^{+}_t = \boldsymbol{\varphi}(\boldsymbol{x}_t) - \boldsymbol{\mu}^{+}_{X,t}$, and analogously, the error of the a-priori estimator as $\boldsymbol{\varepsilon}^{-}_t$. It is easy to show that the kernel Kalman update, given an unbiased a-priori estimator ($\mathbb{E}[\boldsymbol{\varepsilon}^{-}_t] = 0$), yields an unbiased a-posteriori estimator ($\mathbb{E}[\boldsymbol{\varepsilon}^{+}_t] = 0$), independent of the choice of $\mathcal{Q}_t$. For an elaborate discussion of the unbiasedness, we refer to the supplement. Yet, we want to obtain the kernel Kalman gain operator $\mathcal{Q}_t$ that minimizes the squared loss which is equivalent to minimizing the variance of the estimator. The objective of minimizing the variance can be written as minimizing the trace of the a-posteriori covariance operator $\mathcal{C}^{+}_{XX,t}$ of the state $\boldsymbol{x}_t$ at time t , i.e.,

$$\begin{aligned}\min_{\mathcal{Q}_t} \mathbb{E}\left[\left(\boldsymbol{\varepsilon}^{+}_t\right)^{\intercal}\boldsymbol{\varepsilon}^{+}_t\right] &= \min_{\mathcal{Q}_t} \mathrm{Tr}\, \mathbb{E}\left[\boldsymbol{\varepsilon}^{+}_t\left(\boldsymbol{\varepsilon}^{+}_t\right)^{\intercal}\right] \\ &= \min_{\mathcal{Q}_t} \mathrm{Tr}\, \mathcal{C}^{+}_{XX,t}.\end{aligned} \quad (6)$$

Since the measurement residual $\boldsymbol{\epsilon}_t$ is assumed to be independent from the estimation error, and by substituting the posterior error with $\boldsymbol{\varepsilon}^{+}_t = (\mathcal{I} - \mathcal{Q}_t\mathcal{C}_{Y|X})\boldsymbol{\varepsilon}^{-}_t - \mathcal{Q}_t\boldsymbol{\epsilon}_t$, the posterior covariance operator can be reformulated as

$$\begin{aligned}\mathcal{C}^{+}_{XX,t} = &\left(\mathcal{I} - \mathcal{Q}_t\mathcal{C}_{Y|X}\right)\mathcal{C}^{-}_{XX,t}\left(\mathcal{I} - \mathcal{Q}_t\mathcal{C}_{Y|X}\right)^{\intercal} \\ &+ \mathcal{Q}_t\mathcal{R}\mathcal{Q}_t^{\intercal},\end{aligned} \quad (7)$$

where $\mathcal{R}$ is the covariance of the measurement residual. Taking the derivative of the trace of the covariance operator and setting it to zero leads to the following solution for the kernel Kalman gain operator

$$\mathcal{Q}_t = \mathcal{C}^{-}_{XX,t}\mathcal{C}^{\intercal}_{Y|X}(\mathcal{C}_{Y|X}\mathcal{C}^{-}_{XX,t}\mathcal{C}^{\intercal}_{Y|X} + \mathcal{R})^{-1}. \quad (8)$$

From Eqs. 7 and 8, we can also see that it is possible to recursively estimate the covariance embedding operator independently of the mean map or of the observations. This property will allow us later to precompute the covariance embedding operator as well as the kernel Kalman gain operator to further improve the computational complexity of our algorithm.

3.2 Empirical Kernel Kalman Rule

In practice, the conditional observation operator and the kernel Kalman gain operator are estimated from a finite set of samples. Applying the kernel trick (i.e., matrix identities) renders the kernel Kalman rule computationally tractable.

In general, an estimator of the prior mean embedding is given as $\hat{\boldsymbol{\mu}}^-_{X,t} = \boldsymbol{\Upsilon}_x \boldsymbol{m}_t^-$, with weight vector $\boldsymbol{m}_t^-$, and an estimator of the covariance operator is given as $\hat{\mathcal{C}}^-_{XX,t} = \boldsymbol{\Upsilon}_x \boldsymbol{S}_t^- \boldsymbol{\Upsilon}_x^\intercal$, with positive definite weight matrix $\boldsymbol{S}_t^-$.

We can rewrite the kernel Kalman gain operator by substituting with the estimators of the covariance operator and of the conditional operator (c.f. Eq. 3) and obtain

$$\mathcal{Q}_t = \boldsymbol{\Upsilon}_{\boldsymbol{x}} \boldsymbol{S}_t^- \boldsymbol{O}^\intercal \boldsymbol{\Phi}_{\boldsymbol{y}}{}^\intercal (\boldsymbol{\Phi}_{\boldsymbol{y}} \boldsymbol{O} \boldsymbol{S}_t^- \boldsymbol{O}^\intercal \boldsymbol{\Phi}_{\boldsymbol{y}}{}^\intercal + \kappa \mathcal{I})^{-1}, \quad (9)$$

with $\boldsymbol{O} = (\boldsymbol{K}_{xx} + \lambda_{\boldsymbol{O}} \boldsymbol{I}_m)^{-1} \boldsymbol{K}_{xx}$ and with the simplification $\mathcal{R} = \kappa \mathcal{I}$. However, $\mathcal{Q}_t$ still contains the inversion of an infinite dimensional matrix. We can solve this problem by applying the matrix identity $\boldsymbol{A}(\boldsymbol{B}\boldsymbol{A} + \boldsymbol{I})^{-1} = (\boldsymbol{A}\boldsymbol{B} + \boldsymbol{I})^{-1}\boldsymbol{A}$ and arrive at

$$\begin{aligned} \mathcal{Q}_t &= \boldsymbol{\Upsilon}_{\boldsymbol{x}} \boldsymbol{S}_t^- \boldsymbol{O}^\intercal (\boldsymbol{G}_{yy} \boldsymbol{O} \boldsymbol{S}_t^- \boldsymbol{O}^\intercal + \kappa \boldsymbol{I}_m)^{-1} \boldsymbol{\Phi}_{\boldsymbol{y}}{}^\intercal \\ &= \boldsymbol{\Upsilon}_{\boldsymbol{x}} \boldsymbol{Q}_t \boldsymbol{\Phi}_{\boldsymbol{y}}{}^\intercal, \end{aligned} \quad (10)$$

with $\boldsymbol{Q}_t = \boldsymbol{S}_t^- \boldsymbol{O}^\intercal (\boldsymbol{G}_{yy} \boldsymbol{O} \boldsymbol{S}_t^- \boldsymbol{O}^\intercal + \kappa \boldsymbol{I}_m)^{-1}$ and $\boldsymbol{G}_{yy} = \boldsymbol{\Phi}_y^\intercal \boldsymbol{\Phi}_y$. We can now use this reformulation of the kernel Kalman gain operator and apply it to the update equations for the estimator of the mean embedding (Eq. 5) and the estimator of the covariance operator (Eq. 7). It is now straight forward to obtain the update equations for the weight vector $\boldsymbol{m}_t$ and the weight matrix $\boldsymbol{S}_t$ as

$$\boldsymbol{m}_t^+ = \boldsymbol{m}_t^- + \boldsymbol{Q}_t (\boldsymbol{g}_{:\boldsymbol{y}_t} - \boldsymbol{G}_{yy} \boldsymbol{O} \boldsymbol{m}_t^-), \quad (11)$$

$$\boldsymbol{S}_t^+ = \boldsymbol{S}_t^- - \boldsymbol{Q}_t \boldsymbol{G}_{yy} \boldsymbol{O} \boldsymbol{S}_t^-, \quad (12)$$

where $\boldsymbol{g}_{:\boldsymbol{y}_t} = \boldsymbol{\Phi}_y^\intercal \phi(\boldsymbol{y}_t)$ is the embedding of the measurement at time t. The algorithm requires the inversion of a $m \times m$ matrix in every iteration for computing the kernel Kalman gain matrix $\boldsymbol{Q}_t$. Hence, similar to the kernel Bayes' rule, the computational complexity of a straightforward implementation would scale cubically with the number of data points m. However, in contrast to the KBR, the inverse in $\boldsymbol{Q}_t$ is only dependent on time and not on the estimate of the mean map. The kernel Kalman gain matrix can, thus, be precomputed as it is identical for multiple parallel runs of the algorithm. While many applications do not require parallel state estimations, it is a huge advantage for hyper-parameter optimization as we can evaluate multiple trajectories from a validation set simultaneously. As for most kernel-based methods, hyper-parameter optimization is crucial for scaling the approach to complex tasks. So far, the hyper-parameters of the kernels for the KBF have typically been set by heuristics as optimization would be too expensive.

3.3 The Subspace Kernel Kalman Rule

The computational complexity of non-parametric inference suffers from a polynomial growth with the number of samples used for estimating the embeddings. To overcome this drawback, several approaches exist that aim to find a good trade-off between a compact representation and leveraging from a large data set (Snelson and Ghahramani 2006; Csat and Opper 2002; Smola and Bartlett 2001; Gebhardt, Kupcsik, and Neumann 2015). In the following paragraphs, we will explain how the concept of subspace conditional operators (Gebhardt, Kupcsik, and Neumann 2015) can be applied to the kernel Kalman rule in order to maintain the update rules also for large training sets computationally tractable without ignoring valuable information from the provided training samples.

The subspace conditional operator uses subspace projections as representations of the mean map $\boldsymbol{n}_t = \boldsymbol{\Upsilon}_{\hat{x}}^\intercal \mu_t = \boldsymbol{\Upsilon}_{\hat{x}}^\intercal \boldsymbol{\Upsilon}_x \boldsymbol{m}_t$ and the covariance operator $\boldsymbol{P}_t = \boldsymbol{\Upsilon}_{\hat{x}}^\intercal \mathcal{C}_{XX,t} \boldsymbol{\Upsilon}_{\hat{x}} = \boldsymbol{\Upsilon}_{\hat{x}}^\intercal \boldsymbol{\Upsilon}_x \boldsymbol{S}_t \boldsymbol{\Upsilon}_x^\intercal \boldsymbol{\Upsilon}_{\hat{x}}$, where $\boldsymbol{\Upsilon}_{\hat{x}}$ is the feature matrix of a sparse reference set. The subspace conditional operator of the distribution $P(Y|X)$ is then given as

$$\mathcal{C}^{\mathrm{s}}_{Y|X} = \boldsymbol{\Phi}_{\boldsymbol{y}} \boldsymbol{K}_{x\hat{x}} (\boldsymbol{K}_{\hat{x}x} \boldsymbol{K}_{x\hat{x}} + \lambda_{\hat{x}} \boldsymbol{I}_n)^{-1} \boldsymbol{\Upsilon}_{\hat{\boldsymbol{x}}}{}^\intercal, \quad (13)$$

where $\boldsymbol{K}_{\hat{x}x} = \boldsymbol{\Upsilon}_{\hat{x}}^\intercal \boldsymbol{\Upsilon}_x \in \mathbb{R}^{n \times m}$ is the kernel matrix between the sparse subset and the full training data. Using the subspace conditional operator and the subspace representation of the covariance embedding in the kernel Kalman gain operator, we arrive at the following finite representation of the subspace kernel Kalman gain operator

$$\boldsymbol{Q}^{\mathrm{s}}_t = \boldsymbol{P}_t^- \boldsymbol{L}^{\mathrm{s}} \left(\boldsymbol{K}_{\hat{x}x} \boldsymbol{G}_{yy} \boldsymbol{K}_{x\hat{x}} \boldsymbol{L}^{\mathrm{s}} \boldsymbol{P}_t^- \boldsymbol{L}^{\mathrm{s}} + \kappa \boldsymbol{I}_n \right)^{-1} \boldsymbol{K}_{\hat{x}x},$$

with $\boldsymbol{L}^{\mathrm{s}} = \left(\boldsymbol{K}_{\hat{x}x} \boldsymbol{K}_{x\hat{x}} + \lambda \boldsymbol{I}_n \right)^{-1}$. For a detailed derivation of $\boldsymbol{Q}^{\mathrm{s}}_t$, we refer to the supplement. By analogously applying the subspace representations to the update equations for the mean map and the covariance operator, we obtain

$$\boldsymbol{n}_t^+ = \boldsymbol{n}_t^- + \boldsymbol{Q}^{\mathrm{s}}_t \left(\boldsymbol{g}(y_t) - \boldsymbol{G}_{yy} \boldsymbol{K}_{x\hat{x}} \boldsymbol{L}^{\mathrm{s}} \boldsymbol{n}_t^- \right), \quad (14)$$

$$\boldsymbol{P}_t^+ = \boldsymbol{P}_t^- - \boldsymbol{Q}^{\mathrm{s}}_t \boldsymbol{G}_{yy} \boldsymbol{K}_{x\hat{x}} \boldsymbol{L}^{\mathrm{s}} \boldsymbol{P}_t^-. \quad (15)$$

3.4 Recovering the State-Space Distribution

Recovering a distribution in the state space that is a valid preimage of a given mean map is still a topic of ongoing research. There are several approaches to this problem, such as fitting a Gaussian mixture model (McCalman, O 'Callaghan, and Ramos 2013), or sampling from the embedded distribution by optimization (Chen, Welling, and Smola 2010). In the experiments conducted for this paper, we approached the preimage problem by matching a Gaussian distribution, which is a reasonable choice if the recovered distribution is unimodal. Since the belief state of the kernel Kalman rule is a mean map as well as a covariance operator, obtaining the mean and covariance of a Gaussian approximation is done by simple matrix manipulations. For more details we refer to the supplement. However, also any other approach from the literature can be used with the KKR.

3.5 The Kernel Kalman Filter

Similar to the kernel Bayes' filter (Fukumizu, Song, and Gretton 2013; Song, Fukumizu, and Gretton 2013), we can combine the kernel Kalman rule with the kernel sum rule to formulate the *kernel Kalman filter*. We assume a data set $\{(\bar{x}_1, x_1, y_1), \ldots, (\bar{x}_m, x_m, y_m)\}$ consisting of triples with preceding state $\bar{x}_i$, state x_i, and measurement y_i as given. Based on this data set we can define the feature matrices $\boldsymbol{\Upsilon}_x = [\boldsymbol{\varphi}(x_1), \ldots, \boldsymbol{\varphi}(x_m)]$, $\boldsymbol{\Upsilon}_{\bar{x}} = [\boldsymbol{\varphi}(\bar{x}_1), \ldots, \boldsymbol{\varphi}(\bar{x}_m)]$, and $\boldsymbol{\Phi}_y = [\boldsymbol{\phi}(y_1), \ldots, \boldsymbol{\phi}(y_m)]$. We represent the belief state as a mean map $\hat{\mu}_{X,t} = \boldsymbol{\Upsilon}_x \boldsymbol{m}_t$ and a covariance operator $\hat{\mathcal{C}}_{XX,t} = \boldsymbol{\Upsilon}_x \boldsymbol{S}_t \boldsymbol{\Upsilon}_x^\intercal$. We can learn the estimators of the

conditional embeddings for the distributions $P(X|\bar{X})$ and $P(Y|X)$ as

$$\mathcal{C}_{X|\bar{X}} = \boldsymbol{\Upsilon}_x(\boldsymbol{K}_{\bar{x}\bar{x}} + \lambda_{\boldsymbol{T}}\boldsymbol{I})^{-1}\boldsymbol{\Upsilon}_{\bar{x}}^\intercal \quad \text{and} \tag{16}$$

$$\mathcal{C}_{Y|X} = \boldsymbol{\Phi}_y(\boldsymbol{K}_{xx} + \lambda_{\boldsymbol{O}}\boldsymbol{I})^{-1}\boldsymbol{\Upsilon}_x^\intercal, \tag{17}$$

respectively. We can propagate the posterior belief state at time t to the prior belief state at time $t+1$ time by applying the kernel sum rule to the mean embedding and the covariance embedding and obtain the following update rules

$$\boldsymbol{m}_{t+1}^- = \boldsymbol{T}\boldsymbol{m}_t^+, \quad \boldsymbol{S}_{t+1}^- = \boldsymbol{T}\boldsymbol{S}_t^+\boldsymbol{T}^\intercal + \boldsymbol{V}, \tag{18}$$

with $\boldsymbol{T} = (\boldsymbol{K}_{\bar{x}\bar{x}} + \lambda_{\boldsymbol{T}}\boldsymbol{I})^{-1}\boldsymbol{K}_{\bar{x}x}$. Subsequently, we can use the kernel Kalman rule to obtain a posterior belief from the prior embedding conditioned on a new measurement following the steps described in Section 3.2.

4 Experimental Results

In this section, we give a concise description of the experiments we conducted to evaluate the performance of the kernel Kalman rule. In a first experiment, we compare only the kernel Kalman rule to the kernel Bayes' rule by applying them to a simple estimation task. In the further experiments, we compare the kernel Kalman filter to other filtering methods. For these evaluations we use time series from two simulated environments, a simple pendulum and a quad-link, and time series from a human motion tracking data set (Wojtusch and von Stryk 2015). For all kernel based methods, we use the squared exponential kernel. We choose the kernel bandwidths according to the *median trick* (Jaakkola, Diekhans, and Haussler 1999) and scale the found median distances with a single parameter that is optimized.

4.1 Estimation of an Expectation of a Gaussian Distribution

In a first simple experiment, we compare the kernel Kalman rule to the kernel Bayes' rule. We train the models with data sets consisting of hidden states, sampled uniformly from the interval $[-2.5, 2.5]$, and the corresponding measurements, where we add Gaussian noise with standard deviation $\sigma = 0.3$. The training set consists of 100 samples for the KKR and KBR and 500 samples for the subspace KKR, where we still use 100 samples as reference set. We estimate the hidden state by a conditional operator that maps from the kernel space back to the original state. Figure 1 shows the mean squared error (MSE) of the estimated hidden state to

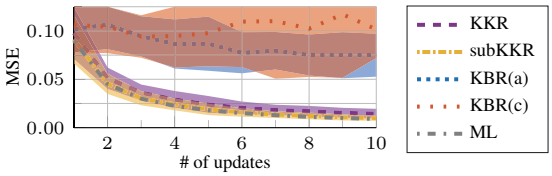


Figure 1: Estimating the expectation of a Gaussian distribution with KKR and KBR. The reference is the expected error of a maximum likelihood estimator.

KKR	subKKR	KBR(a)	KBR(b)	KBR(c)
0.0813 s	0.1074 s	1.2149 s	0.8643 s	0.5655 s

Table 1: Mean time consumed for performing 10 updates on 100 estimates (in parallel) over 20 runs.

the true hidden state for 10 iterative updates with new measurements. The graphs depict mean and 2σ intervals over 20 evaluations. We conducted this experiment with all three versions of the KBR, however, version (b) was numerically instable which led to MSEs that did not fit into the plot anymore. We can see that the estimates from the KKR and the subKKR are at the level of the ML estimator, while the KBR has a worse performance which is not improving with the number of updates. In addition, we limit the values in the $\boldsymbol{D}$ matrix for the KBRs to only positive values in order to get reasonable results.

In Table 1, we compare the computational efficiency of the KKR to the KBR. The times shown are the mean times for 10 updates of 100 estimates. While the single updates could be performed in parallel with the KKR and the subKKR, they have to be performed for each estimate independently for the KBR. It can be clearly seen that the KKR outperforms the KBR in the order of a magnitude. Furthermore, the subKKR maintains computational efficiency, while using 5 times more data points for learning the models.

4.2 Pendulum

In this experiment, we use a simulated pendulum as system dynamics. It is uniformly initialized in the range $[0.1\pi, 0.4\pi]$ with a velocity sampled from the range $[-0.5\pi, 0.5\pi]$. The observations of the filters were the joint positions with additive Gaussian noise sampled from $\mathcal{N}(0, 0.01)$. We compare the KKF, the subspace KKF (subKKF) and the KKF learned with the full dataset (fullKKF) to version (a) of the kernel Bayes filter (Song, Fukumizu, and Gretton 2013), the kernel Kalman filter with covariance embedding operator (KKF-CEO) (Zhu, Chen, and Principe 2014), as well as to standard filtering approaches such as the EKF (Julier and Uhlmann 1997) and the UKF (Wan and Van Der Merwe 2000). Both, the EKF and the UKF require a model of the system dynamics. KBF(b) was numerically too instable and KBF(c) already yielded worse results in the previous experiment. To learn the models, we simulated episodes with a length of 30 steps (3s). The results are shown in Figure 2. The KKF and subKKF show clearly better results than all other non-parametric filtering methods and reach a performance level close to the EKF and UKF.

4.3 Quad-Link

In this experiment, we used a simulated 4-link pendulum where we observe the 2-D end-effector positions. The state of the pendulum consists of the four joint angles and joint velocities. We evaluate the prediction performance of the subKKF in comparison to the KKF-CEO, the EKF and the UKF. All other non-parametric filtering methods could not achieve a good performance or broke down due to the high computation times. As the subKKF outperformed the KKF

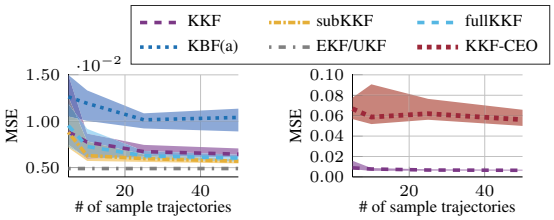
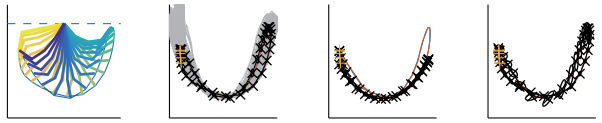


Figure 2: Comparison of KKF to KBF(a), KKF-CEO, EKF and UKF. All kernel methods (except fullKKF) use kernel matrices of 100 samples. The subKKF method uses a subset of 100 samples and the whole dataset to learn the conditional operators. Depicted is the median MSE to the ground-truth of 20 trials with the [0.25 0.75] quantiles.



(a) anim. QL (b) MCF (c) UKF (d) subKKF

Figure 3: Example trajectory of the quad-link end-effector. The filter outputs in black, where the ellipses enclose 90% of the probability mass. All filters were updated with the first five measurements (yellow marks) and predicted the following 30 steps. Figure (a) is an animation of the trajectory.

in the previous experiments and is also computationally much cheaper, we skip the comparison to the standard KKF in this and the subsequent experiments.

In a first qualitative evaluation, we compare the long-term prediction performance of the subKKF in comparison to the UKF, the EKF and the Monte-Carlo filter (MCF) as a baseline (see Appendix). This evaluation can be seen in Figure 3. The UKF is not able to predict the movements of the quad-link end-effector due to the high non-linearity, while the subKKF is able to predict the whole trajectory.

We also compared the 1, 2 and 3-step prediction performance of the subKKF to the KKF-CEO, EKF and UKF (Fig. 4). The KKF-CEO provides poor results already for the filtering task. The EKF performs equally bad, since the observation model is highly non-linear. The UKF already yields a much better performance as it does not suffer from the linearization of the system dynamics. The subKKF outperformed the UKF.

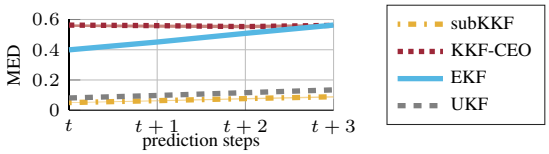


Figure 4: 1, 2 and 3 step prediction performances in mean euclidean distances (MED) to the true end-effector positions of the quad-link.

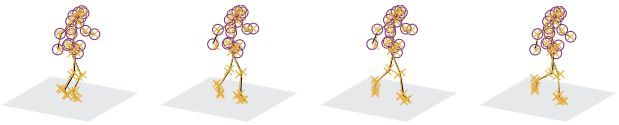


Figure 5: Example sequence of 4 postures. The markers of the upper body (violet circles) were observed and the positions of all markers (yellow crosses) and of all joints (yellow skeleton) were estimated. The black skeleton is the ground-truth from the dataset.

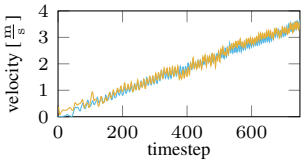


Figure 6: Estimated subject velocity of the human motion dataset by the subKKF (orange) and ground-truth (blue).

4.4 Human Motion Dataset

We used walking and running motions captured from one subject from the HuMoD dataset (Wojtusch and von Stryk 2015). The dataset comprised the x, y and z locations of the 36 markers attached to the subject's body as well as the velocity of the subject in x-direction, computed from the markers at the left and right anterior-superior iliac spine. We used walking and running motions at different constant speeds for training one model. For the evaluation of this model, we used a test data-set in which the subject accelerates from 0m/s to 4m/s.

In a first experiment, we learned the subKKF with a kernel size of 800 samples, where we used data windows of size 3 with the 3D positions of all 36 markers as state representation and the current 3D positions of all markers as observations. In this task, we want to predict the subject's velocity. To do so, we learned a conditional operator from the state representation to the observed velocities in the training set. Subsequently, we applied our model to the transition motion dataset to reconstruct the subject's velocity. The results of this experiment are depicted in Figure 6.

In a second experiment, we learned the subKKF with a kernel size of 2000 samples and used data windows of size 4 with all 36 marker positions as state representations. However, this time we observed only the markers on the upper body and used all marker positions as well as the joint positions as prediction targets. Figure 5 shows a sequence of 4 postures, where the blue circles denote the observation, the red crosses the reconstructed positions of the markers, the red skeleton is drawn with the reconstructed positions of the joints and the black skeleton is drawn with the joint positions in the dataset.

5 Conclusions

In this paper, we proposed the kernel Kalman rule (KKR) for Bayesian inference with nonparametric representations of probability distributions. We showed that the KKR, as an alternative to the kernel Bayes' rule (KBR), is computationally more efficient, numerically more stable and follows from a clear optimization objective. We combined the

KKR as Bayesian update with the kernel sum rule to formulate the kernel Kalman filter (KKF) that can be applied to nonlinear filtering tasks. In difference to existing kernel Kalman filter formulations, the KKF provides a more general formulation that is much closer to the original Kalman filter equations and can also be applied to partially observable systems. Future work will concentrate on the use of hyper-parameter optimization with more complex kernels and learning the transition dynamics in the RKHS with an expectation-maximization algorithm in case of missing information about the latent state.

Acknowledgments

This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No #645582 (RoMaNS).

Appendix

We use a Monte Carlo approach to compare the KKF to a baseline in the quadlink prediction task (c.f. Sec. 4.3). We sample 10^5 episodes from the known system equations and use the knowledge about the observation noise to compute the likelihood of the observations w.r.t. each of the sampled episodes. We can then obtain a posterior of each sampled episode given the observations by normalizing the likelihood. We weigh each of the sampled episodes with its posterior probability and obtain finally the Monte Carlo filter output as the sum of these weighted episodes.

References

Aronszajn, N. 1950. Theory of reproducing kernels. *Transactions of the American Mathematical Society* 68(3):337–404.

Boots, B.; Gretton, A.; and Gordon, G. J. 2013. Hilbert space embeddings of predictive state representations. In *Proceedings of the 29th International Conference on Uncertainty in Artificial Intelligence (UAI)*.

Chen, Y.; Welling, M.; and Smola, A. J. 2010. Supersamples from kernel-herding. In *Proceedings of the 26th International Conference on Uncertainty in Artificial Intelligence (UAI)*.

Csat, L., and Opper, M. 2002. Sparse on-line gaussian processes. *Neural computation* 14(3):641–668.

Fukumizu, K.; Song, L.; and Gretton, A. 2013. Kernel bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research* 14(1):3683–3719.

Gauss, C. F. 1823. *Theoria combinationis observationum erroribus minimis obnoxiae*. H. Dieterich.

Gebhardt, G. H. W.; Kupcsik, A.; and Neumann, G. 2015. Learning subspace conditional embedding operators. *Workshop on Large-Scale Kernel Learning at ICML 2015*.

Hsu, D.; Kakade, S. M.; and Zhang, T. 2012. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences* 78(5):1460–1480.

Jaakkola, T.; Diekhans, M.; and Haussler, D. 1999. Using the fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, 149–158.

Jaeger, H. 2000. Observable operator models for discrete stochastic time series. *Neural Computation* 12(6):1371–1398.

Julier, S. J., and Uhlmann, J. K. 1997. A new extension of the kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3, 3–2.

Kalman, R. E. 1960. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering* 82(1):35–45.

McCalman, L.; O 'Callaghan, S.; and Ramos, F. 2013. Multi-modal estimation with kernel embeddings for learning motion models. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2845–2852. Karlsruhe: IEEE.

McElhoe, B. A. 1966. An assessment of the navigation and course corrections for a manned flyby of mars or venus. *IEEE Transactions on Aerospace and Electronic Systems* AES-2.

Ralaivola, L., and d'Alche Buc, F. 2005. Time series filtering, smoothing and learning using the kernel kalman filter. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* 3:1449–1454.

Schölkopf, B.; Herbrich, R.; and Smola, A. 2001. A generalized representer theorem. *Computational Learning Theory* (2111):416–426.

Simon, D. 2006. *Optimal State Estimation*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Smith, G. L.; Schmidt, S. F.; and McGee, L. A. 1962. *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. National Aeronautics and Space Administration.

Smola, A. J., and Bartlett, P. 2001. Sparse greedy gaussian process regression. *Advances in Neural Information Processing Systems* 13:619–625.

Smola, A. J.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A hilbert space embedding for distributions. In *Proceedings of the 18th international conference on Algorithmic Learning Theory*, volume 4754, 13–31.

Snelson, E., and Ghahramani, Z. 2006. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, 1257–1264.

Song, L.; Boots, B.; Siddiqi, S. M.; Gordon, G. J.; and Smola, A. J. 2010. Hilbert space embeddings of hidden markov models. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 991–998.

Song, L.; Fukumizu, K.; and Gretton, A. 2013. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine* 30(4):98–111.

Sorenson, H. W. 1970. Least-squares estimation: from gauss to kalman. *Spectrum, IEEE* 7(7):63–68.

Wan, E., and Van Der Merwe, R. 2000. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 153–158. IEEE.

Wojtusch, J., and von Stryk, O. 2015. Humod - a versatile and open database for the investigation, modeling and simulation of human motion dynamics on actuation level. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE.

Zhu, P.; Chen, B.; and Principe, J. C. 2014. Learning nonlinear generative models of time series with a kalman filter in rkhs. *IEEE Transactions on Signal Processing* 62(1):141–155.