Fast Kernel Approximations for Latent Force Models and Convolved Multiple-Output Gaussian processes

Cristian Guarnizo
Faculty of Engineering
Universidad Tecnológica de Pereira
Pereira, Colombia, 660003

Mauricio A. Álvarez
Department of Computer Science
The University of Sheffield
Sheffield, UK, S1 4DP

Abstract

A latent force model is a Gaussian process with a covariance function inspired by a differential operator. Such covariance function is obtained by performing convolution integrals between Green's functions associated to the differential operators, and covariance functions associated to latent functions. In the classical formulation of latent force models, the covariance functions are obtained analytically by solving a double integral, leading to expressions that involve numerical solutions of different types of error functions. In consequence, the covariance matrix calculation is considerably expensive, because it requires the evaluation of one or more of these error functions. In this paper, we use random Fourier features to approximate the solution of these double integrals obtaining simpler analytical expressions for such covariance functions. We show experimental results using ordinary differential operators and provide an extension to build general kernel functions for convolved multiple output Gaussian processes.

1 INTRODUCTION

Latent force models (LFMs) [Álvarez et al., 2009] are a type of multiple-output Gaussian processes (GPs) where the covariance function has been derived from physical models. In particular, LFMs assume that each output $\{f_d(t)\}_{d=1}^D$ can be expressed as the convolution integral of a latent function $u(t)$, and a Green's function $G_d(t)$ associated to a linear dynamical system, one per output, $f_d(t) = \int_0^t G_d(t-\tau)u(\tau)d\tau$. Such representation for $f_d(t)$ introduces a dependency between outputs $f_d(t)$ and $f_{d'}(t)$. For example, if we assume that $u(t)$ follows a Gaussian process prior with zero mean function and covariance $k(t,t')$, due to the linearity of the integral transform, $f_d(t)$ and $f_{d'}(t)$ are jointly Gaussian with a cross-covariance function given as $k_{f_d,f_{d'}}(t,t') = \int_0^t G_d(t-\tau)\int_0^{t'} G_{d'}(t'-\tau')k(\tau,\tau')\mathrm{d}\tau'\mathrm{d}\tau$.

LFMs have been used for uncovering the dynamics of transcription factors in a gene network [Gao et al., 2008], for extrapolating human motion from motion capture data [Álvarez et al., 2013], for segmenting motor primitives in humanoid robotics [Álvarez et al., 2011], for modeling the thermal properties of buildings [Ghosh and et al., 2015], among several other applications for which prior knowledge of a mechanistic model can be coded in the covariance function of a GP. By including physics in the covariance function of a GP, we grant extrapolation abilities to an otherwise interpolation only-model.

In a classical latent force model, the covariance of the latent function $k(t,t')$ follows an Exponentiated Quadratic (EQ) form, leading to analytical solutions for the cross-covariances $k_{f_d,f_{d'}}(t,t')$. However, these solutions are computationally expensive since they involve calculating functions that can only be obtained by numerical methods. For example, using the second order LFM introduced in Álvarez et al. [2009], involves computing the error function $\mathrm{erf}(\cdot)$ with a complex argument or the Faddeeva function, that require the evaluation of numerical integrals that are expensive to compute.

In this work, we use random Fourier features (RFF) [Rahimi and Recht, 2008] to reduce the mathematical complexity of the expressions involved in the covariance functions of the LFM. In particular, we approximate the calculation of the EQ kernel, with a representation that involves its probability density via the Bochner's theorem. Such representation for the covariance of $k(\tau,\tau')$ transforms the double integral for $k_{f_d,f_{d'}}(t,t')$ into two separate integrals that can easily be solved using the Laplace or Fourier transforms. Once the inner integrals are solved (the integrals that depend on τ and τ'), the remaining integral is solved using a Monte Carlo approximation with S samples. The quality of the approximation of the

cross-covariances $k_{f_d,f_{d'}}(t,t')$ will depend, then, on the number of samples S used. Additionally, by representing the latent force model kernel using a sum of basis functions, we are able to reduce the computational complexity of inverting the $ND \times ND$ kernel matrix obtained from the multiple outputs, assuming that each output has N data observations.

Following a similar procedure, we also introduce a random Fourier feature approximation for the more general convolved multiple output Gaussian process kernel, a model that can be used for multiple-output with no particular known dynamics.

2 LATENT FORCE MODELS

Latent force models are Gaussian processes for multiple outputs with the characteristic that their covariance function involves ordinary or partial differential equations. In particular, LFMs assume that each output $\{f_d(t)\}_{d=1}^D$ can be described using

$$\mathcal{D}_d\{f_d(t)\} = u(t),$$

where $\mathcal{D}_d$ is the differential operator associated to a linear ordinary differential equation (ODE) or a linear partial differential equation (PDE), and $u(t)$ is the excitation function. LFMs assume that $u(t)$ is unknown and place a Gaussian process prior over it. The solution for $f_d(t)$ follows as

$$f_d(t) = \int_0^t G_d(t-\tau)u(\tau)d\tau, \tag{1}$$

where $G_d(\cdot)$ corresponds to the Green's function associated to the differential operator $\mathcal{D}_d$. The latent force or function $u(t)$ is unobserved, and follows a Gaussian process prior with zero mean function, and covariance function given by $k(t,t')$. Since $u(t)$ is being transformed by a linear operator, $f_d(t)$ also follows a Gaussian process with covariance function $k_{f_d,f_d}(t,t')$. Furthermore, since all $f_d(t)$ have a common input $u(t)$, it is also possible to compute a cross-covariance function between $f_d(t)$, and $f_{d'}(t')$, $k_{f_d,f_{d'}}(t,t')$.

Equation (1) can be extended to include additional latent functions with different characteristics, leading to express each output as

$$f_d(t) = \sum_{q=1}^{Q} S_{d,q} \int_0^t G_d(t-\tau)u_q(\tau)d\tau,$$

where there are Q latent functions or forces $\{u_q(t)\}_{q=1}^Q$, and $S_{d,q}$ is a sensitivity parameter that accounts for the influence of force $u_q(t)$ over output d. Assuming the independence of these latent forces and that they all follow Gaussian process priors with covariance functions $k_q(t,t')$, it is possible to compute the cross-covariance functions $k_{f_d,f_{d'}}(t,t')$, $\forall\, d,d' = 1\ldots,D$. The following general expression can be used to build the covariance $k_{f_d,f_{d'}}(t,t')$ of a LFM

$$\sum_{q=1}^{Q} S_{d,q}S_{d',q} \int_0^t G_d(t-\tau) \int_0^{t'} G_{d'}(t'-\tau')\times k_q(\tau,\tau')\mathrm{d}\tau'\mathrm{d}\tau. \tag{2}$$

Depending on the form for the covariance function for $k_q(t,t')$, it is possible to find a closed-form expression for $k_{f_d,f_{d'}}(t,t')$. A common option for $k_q(\tau,\tau')$ is the Exponentiated Quadratic form

$$k_q(\tau,\tau') = \exp\left[-\frac{(\tau-\tau')^2}{\ell_q^2}\right],$$

where ℓ_q is known as the length-scale parameter.

LFMs have mostly being used for multiple output regression. In this case, the observed output d, $y_d(t)$, is assumed to follow a Gaussian likelihood, $y_d(t) = f_d(t)+\epsilon_d$, where $\epsilon_d \sim \mathcal{N}(0,\sigma_d^2)$.

3 FEATURE EXPANSIONS FOR KERNELS DERIVED FROM LATENT FORCE MODELS

In order to scale kernel machines, Rahimi and Recht [2008] introduced the idea of random Fourier features to approximate a kernel function using inner products between basis functions. Parameters of these basis functions are sampled from a distribution associated to the kernel function. We are particularly interested in the approximation for the EQ kernel, which has been commonly used in LFMs. The idea is to replace the EQ kernel that is usually assumed for $k_q(\tau,\tau')$ by providing a random Fourier feature representation for it via the Bochner's theorem,

$$k_q(\tau,\tau') = e^{-\frac{(\tau-\tau')^2}{\ell_q^2}} = \int p(\lambda)e^{j(\tau-\tau')\lambda}d\lambda, \tag{3}$$

where $p(\lambda) = \mathcal{N}(\lambda|0,\frac{2}{\ell_q^2})$. A key insight from Rahimi and Recht [2008] was to use a finite approximation for $k_q(\tau,\tau')$ by using Monte Carlo sampling to solve the above integral over λ,

$$k_q(\tau,\tau') \approx \frac{1}{S}\sum_{s=1}^{S} e^{j\lambda_s\tau}e^{-j\lambda_s\tau'},$$
$$= \frac{1}{S}\sum_{s=1}^{S} v(\tau,\lambda_s)v^*(\tau,\lambda_s),$$

where S is the number of Monte Carlo samples, $v(\tau, \lambda_s)$ is a basis function with parameter λ_s, $v^*(\tau, \lambda_s)$ is the complex conjugate of $v(\tau, \lambda_s)$, and $\lambda_s \sim p(\lambda)$. Since the kernel function is a real function, the real part of the product $v(\tau, \lambda_s)v^*(\tau, \lambda_s)$ is used instead.

Using the expression for $k_q(\tau, \tau')$ in Eq. (3) inside the expression for the cross-covariance function for the LFM, $k_{f_d f_{d'}}(t, t')$, we get

$$\sum_{q=1}^{Q} S_{d,q} S_{d',q} \int_0^t G_d(t-\tau) \int_0^{t'} G_{d'}(t'-\tau') \times \int p(\lambda) e^{j(\tau-\tau')\lambda} d\lambda \mathrm{d}\tau' \mathrm{d}\tau.$$

Organizing the above expression we obtain

$$\sum_{q=1}^{Q} S_{d,q} S_{d',q} \int p(\lambda) v_d(t, \theta_d \lambda) v^*_{d'}(t', \theta_{d'}, \lambda) d\lambda, \quad (4)$$

with

$$v_d(t, \theta_d, \lambda) = \int_0^t G_d(t-\tau) e^{j\lambda\tau} \mathrm{d}\tau,$$

where θ_d makes reference to the parameters of the Green's function $G_d(\cdot)$. Also, $v^*_{d'}(t', \theta_{d'}, \lambda)$ is the complex conjugate for $v_{d'}(t', \theta_{d'}, \lambda)$. The integrals over t and t' above can be solved using the Laplace transform $\mathcal{L}\{\cdot\}$

$$\begin{aligned} v_d(t, \theta_d, \lambda) &= \mathcal{L}^{-1}\mathcal{L}\left\{ \int_0^t G_d(t-\tau) e^{j\lambda\tau} \mathrm{d}\tau \right\} \\ &= \mathcal{L}^{-1}\left\{ \mathcal{G}_d(s) \mathcal{L}\{e^{j\lambda\tau}\} \right\}, \end{aligned}$$

where $\mathcal{G}_d(s)$ is the Laplace transform for $G_d(t)$. The operator $\mathcal{L}^{-1}\{\cdot\}$ refers to the inverse Laplace transform. Furthermore, notice that when $G_{d'}(\cdot)$ is a real function, we can compute $v^*_{d'}(t', \theta_{d'}, \lambda) = v_{d'}(t', \theta_{d'}, -\lambda)$.

Similarly to Rahimi and Recht [2008], we use Monte Carlo sampling to approximate the integral over λ in Eq. (4), leading to

$$\sum_{q=1}^{Q} \frac{S_{d,q} S_{d',q}}{S} \left[\sum_{s=1}^{S} v_d(t, \theta_d, \lambda_s) v^*_{d'}(t', \theta_{d'}, \lambda_s) \right],$$

where $\lambda_s \sim p(\lambda)$.

The steps to compute a RFF approximation of the LFM kernel are

1. Compute $v_d(t, \theta_d, \lambda) = \int_0^t G_d(t-\tau) e^{j\lambda\tau} \mathrm{d}\tau$ using the Laplace transform.
2. Compute the RFF approximation for the LFM covariance function $k_{f_d f_{d'}}(t, t')$ using

$$\sum_{q=1}^{Q} \frac{S_{d,q} S_{d',q}}{S} \left[\sum_{s=1}^{S} v_d(t, \theta_d, \lambda_s) v^*_{d'}(t', \theta_{d'}, \lambda_s) \right],$$

where $\lambda_s \sim p(\lambda)$. The distribution we use to sample from, $p(\lambda)$, depends on the kernel assumed for the latent forces $u_q(t)$.

Interestingly, $v_d(t, \theta_d, \lambda)$ represents the response of the dynamical system to the excitation $e^{j\lambda t}$ up to time t. We will occasionally refer to this random feature as a *random Fourier response feature* (RFRF).

In different applications of LFMs, we need to perform inference over the latent forces $u_q(t)$. Inference over $u_q(t)$ requires the evaluation of the cross-covariance functions $k_{f_d, u_q}(t, t')$. Such cross-covariances are also important in schemes that reduce computational complexity in convolved multiple output Gaussian processes, where the underlying process $u_q(t)$ evaluated at a discrete set of input locations serve the purpose of *inducing variables* [Álvarez et al., 2010, Álvarez and Lawrence, 2011]. The approximation of $k_{f_d, u_q}(t, t')$ using RFFs is given by

$$k_{f_d, u_q}(t, t') = \frac{1}{S} \sum_{s=1}^{S} v_d(t, \theta_d, \lambda_s) e^{-j\lambda_s t'}.$$

4 HYPERPARAMETER SELECTION AND COMPUTATIONAL COMPLEXITY

Let us assume, we are given observations $\{\mathbf{y}, \mathbf{X}\} = \{\mathbf{y}_d, \mathbf{X}_d\}_{d=1}^{D}$ (each $\mathbf{y}_d \in \mathbb{R}^N$ and $\mathbf{X}_d \in \mathbb{R}^{N \times p}$), and we want to learn the hyperparameters of the kernel function, $\{\{\theta_d, \sigma_d^2\}_{d=1}^{D}, \{\ell_q\}_{q=1}^{Q}\}$, that allow us to explain $\mathbf{y}$. With that in mind, the hyperparamters can be learned from the log-marginal likelihood [Rasmussen and Williams, 2006]

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}) = &- \frac{ND}{2} \log(2\pi) - \frac{1}{2} \mathbf{y}^\top (\mathbf{K}_{\mathbf{f},\mathbf{f}} + \mathbf{\Sigma})^{-1} \mathbf{y} \\ &- \frac{1}{2} \log |\mathbf{K}_{\mathbf{f},\mathbf{f}} + \mathbf{\Sigma}|, \end{aligned} \quad (5)$$

where $\mathbf{\Sigma}$ is a diagonal matrix containing the variances of the noise level per output, and $\mathbf{K}_{\mathbf{f},\mathbf{f}} \in \mathbb{R}^{ND \times ND}$ is a block-wise matrix with blocks calculated using (2). As it is usual, we can use a gradient-based optimization procedure to estimate the hyperparameters that maximize the log-marginal likelihood leading to the infamous computational complexity of $\mathcal{O}(D^3 N^3)$.

However, notice that by the elegance of the RFF representation, the covariance matrix can instead be approximated

as $\mathbf{K}_{\mathbf{f},\mathbf{f}} = \mathbb{R}\left\{\mathbf{\Phi}\mathbf{\Phi}^{\mathsf{H}}\right\}$, where $\mathbf{\Phi} \in \mathbb{C}^{ND\times QS}$ has entries $v_d(t,\theta_d,\lambda_s)$, and $\mathbf{\Phi}^{\mathsf{H}}$ is the conjugate transpose of $\mathbf{\Phi}$. Furthermore, the covariance matrix can be re-written as $\mathbf{K}_{\mathbf{f},\mathbf{f}} = \mathbf{\Phi}_c\mathbf{\Phi}_c^\top$, with $\mathbf{\Phi}_c = [\mathbb{R}\{\mathbf{\Phi}\}\ \ \mathbb{I}\{\mathbf{\Phi}\}] \in \mathbb{C}^{ND\times 2QS}$. Using the matrix inversion and determinant lemmas, we express the log-marginal likelihood as

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\log|\mathbf{\Sigma}| - \frac{1}{2}\left(\mathbf{y}^\top\mathbf{\Sigma}^{-1}\mathbf{y} - \boldsymbol{\alpha}^\top\mathbf{A}^{-1}\boldsymbol{\alpha}\right) - \frac{1}{2}\log|\mathbf{A}| - \frac{ND}{2}\log(2\pi), \tag{6}$$

with $\mathbf{A} = \mathbf{I} + \mathbf{\Phi}_c^\top\mathbf{\Sigma}^{-1}\mathbf{\Phi}_c$ and $\boldsymbol{\alpha} = \mathbf{\Phi}_c^\top\mathbf{\Sigma}^{-1}\mathbf{y}$, effectively reducing computational complexity from $\mathcal{O}(D^3N^3)$ to $\mathcal{O}(DNQ^2S^2)$, which is now linear with respect to the data size.

Alternatively, one could couple the computation of the kernel functions $k_{f_d,f_{d'}}(t,t')$ and $k_{f_d,u_q}(t,t')$ through random Fourier response features, with *(i)* any of the different computationally efficient approximations for optimizing the log-marginal likelihood in convolved multiple-output Gaussian process [Álvarez and Lawrence, 2011], or *(ii)* a lower bound on the log-marginal likelihood through a variational approximation [Álvarez et al., 2010]. Both styles of approximations require the specification of K inducing variables.

5 FAST KERNEL BUILDING FROM ORDINARY DIFFERENTIAL EQUATIONS

Let us assume we are interested in analyzing an ODE of order P given as

$$\mathcal{D}_d^{(P)}\{f_d(t)\} = \sum_{q=1}^{Q} S_{d,q}u_q(t),$$

where the differential operator $\mathcal{D}_d^{(P)}$ is defined as

$$\mathcal{D}_d^{(P)} = a_0\frac{\mathrm{d}^P}{\mathrm{d}t^P} + a_1\frac{\mathrm{d}^{P-1}}{\mathrm{d}t^{P-1}} + \ldots + a_{P-1}\frac{\mathrm{d}}{\mathrm{d}t} + a_P.$$

The Laplace transform of the Green's function $G_d(t)$ for the above ODE can be found as

$$\begin{aligned}\mathcal{G}_d(s) &= \frac{1}{a_0}\frac{1}{s^P + \frac{a_1}{a_0}s^{P-1} + \ldots + \frac{a_P}{a_0}} \\ &= \frac{1}{a_0}\frac{1}{(s-s_1)(s-s_2)\ldots(s-s_P)},\end{aligned} \tag{7}$$

where the s_i's represent the roots of the polynomial given in the denominator of (7). Additionally, the Laplace transform for $\mathcal{L}\{e^{j\lambda\tau}\} = \frac{1}{s-j\lambda}$. We can use a partial-fraction expansion for $\mathcal{G}_d(s)$, and then apply the inverse Laplace transform over the product $\mathcal{G}_d(s)\mathcal{L}\{e^{j\lambda\tau}\}$ to find $v_d(t,\theta_d,\lambda)$.

Interestingly, if all the roots $s_1,\ldots,s_P$ are distinct real or distinct complex, and $s_{P+1} = j\lambda$ (the additional root obtained from $\mathcal{L}\{e^{j\lambda\tau}\}$), the random Fourier response feature $v_d(t,\theta_d,\lambda)$ can be expressed as

$$\frac{1}{a_0}\mathcal{L}^{-1}\left\{\sum_{p=1}^{P+1}\frac{A_p}{(s-s_p)}\right\} = \frac{1}{a_0}\sum_{p=1}^{P+1}A_p e^{s_p t},$$

where each coefficient A_p is calculated as

$$A_p = \frac{1}{\prod_{\forall i\neq p}(s_p - s_i)}, \tag{8}$$

and, as before, $s_{P+1} = j\lambda$.

Next, we show some examples of the expressions obtained for the random Fourier response features associated to the ODE of first and second orders. Besides, for all ODE experiments the hyperparameters are learned using the variational approach described in Álvarez et al. [2010] and they were carried out using a single core of an AMD FX-8350 @ 4.0 GHz. We also include measures of the time required to evaluate the objective function and its gradients to compare the time cost induced by the evaluation of the different covariance functions. Code to replicate the following experiments is available at `github.com/cdguarnizo/kff_lfm`.

5.1 FIRST-ORDER MODEL (ODE1)

For the first-order ODE we have the following equation

$$\mathcal{D}_d^{(1)}\{f_d(t)\} = \frac{\mathrm{d}f_d(t)}{\mathrm{d}t} + \gamma_d f_d(t) = \sum_{q=1}^{Q}u_q(t),$$

from which the Laplace transform is given by $\mathcal{G}_d(s) = \frac{1}{s+\gamma_d}$. We then have $s_1 = -\gamma_d$, and $s_2 = j\lambda$. The random Fourier response feature for the d-th output function of a first-order ODE is obtained as

$$\begin{aligned}v_d^{(1)}(t,\theta_d,\lambda) &= A_1e^{s_1t} + A_1e^{s_2t} \\ &= -\frac{e^{-\gamma_d t}}{\gamma_d + j\lambda} + \frac{e^{j\lambda t}}{\gamma_d + j\lambda} \\ &= \frac{e^{j\lambda t} - e^{-\gamma_d t}}{\gamma_d + j\lambda}.\end{aligned}$$

Next, we compare the performance of the first order ODE described in Gao et al. [2008] with the kernel obtained by using the above random Fourier response feature for interpolation of Air temperature.

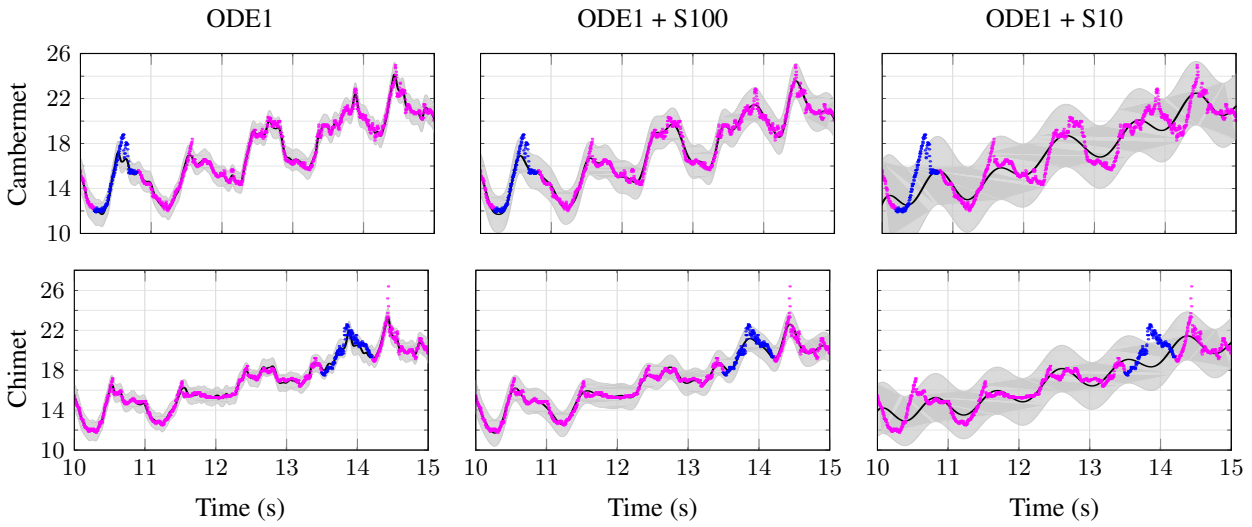

Figure 1: Comparison of the predictive GPs, for the air temperature experiment, using the standard LFM (first column) and the RFRF approximation for $S = 100$ (second column) and $S = 10$ samples (third column). Training data is represented using red dots and Test data using blue dots. The black line in the mean over the predictive GP function, and the shaded region denotes two times the standard deviation.

Air temperature Here, we consider the problem of modeling and predicting air temperature time series from a network sensor located at the south coast of England. The dataset consists of temperature measurements at four locations known as Bramblemet, Sotonmet, Cambermet and Chimet. [1] The air temperatures are measured during the period from July 10 to July 15, 2013. Specifically, we adopt the same experiment (train and test data) used in Nguyen and Bonilla [2014] and described in Tab. 1. The variational approach is configured with 200 inducing variables, six latent forces and the maximum number of iterations for the optimization procedure is set to 500.

Table 1: Number of training and test data-points considered on the air temperature experiment.

#	Name	Training	Test
1	Bramblemet	1425	0
2	Cambermet	1268	173
3	Chimet	1235	201
4	Sotonmet	1097	0

Table 2 reports the predictive performance using the covariance functions build from the LFM and the proposed RFRF. Note that for a low number of samples S, the proposed approach presents the worst performance. This is because the more samples we use the better the mean of predictive GP is able to fit the coarse behavior from the observed data, as shown in figure 1. Interestingly, the RFRF starts to outperform the standard one, using only 50 or 100 samples with about half of the time required by the original covariance function.

[1] Weather data can be found in `http://www.bramblemet.co.uk`.

Table 2: Results on air temperature data.

Kernel	Cambermet		Chimet		Time
	NMSE	NLPD	NMSE	NLPD	[s]
ODE1+S10	0.74	3.26	0.58	1.53	1.89
ODE1+S20	0.45	1.95	0.93	1.75	2.09
ODE1+S50	**0.08**	**1.10**	0.21	1.08	2.68
ODE1+S100	0.12	1.18	**0.12**	**0.82**	3.93
ODE1	0.11	1.37	0.19	0.99	6.28

5.2 SECOND-ORDER MODEL (ODE2)

As a second example of a random Fourier feature representation of a LFM, we use a second-order ordinary differential operator $\mathcal{D}_d^{(2)}\{\cdot\}$ that represents, e.g., a mass-spring-damper system. The second-order operator is given as

$$\mathcal{D}_d^{(2)} = m_d \frac{d^2}{dt^2} + c_d \frac{d}{dt} + b_d,$$

where m_d, c_d and b_d are the mass, damper and spring constants, respectively. From the above equation, we obtain the Laplace transform of the Green's function as

$$\mathcal{G}_d(s) = \frac{1}{m_d} \frac{1}{s^2 + \frac{c_d}{m_d} s + \frac{b_d}{m_d}}.$$

Following the procedure described above, it can be shown that the random Fourier response feature for the d-th out-

put is given by

$$v_d^{(2)}(t,\theta_d,\lambda) = \frac{1}{m_d}\left[A_1 e^{s_1 t} + A_2 e^{s_2 t} + A_3 e^{s_3 t}\right],$$

where

$$s_1, s_2 = -\frac{c_d}{2m_d} \pm \sqrt{\frac{c_d^2}{4m_d^2} - \frac{b_d}{m_d}},$$

are the roots of the polynomial obtained from the second-order ODE, and $s_3 = j\lambda$ corresponds to the root induced by the excitation $e^{j\lambda t}$. Note that the coefficients A_1 and A_2 were calculated using (8). Furthermore, if $c_d^2 > 4m_d b_d$ then the roots s_1 and s_2 are real, and the model's response is known as "overdamped". When $c_d^2 < 4m_d b_d$ the roots are a pair of complex conjugates, and the response is known as "underdamped".

Figure 2 shows the covariance matrices for a two-output LFM using the standard expression for the covariance function in Álvarez et al. [2009], and the kernel obtained by using the random Fourier response features for the ODE2, $v_d^{(2)}(t,\theta_d,\lambda)$, based on $S = 100$ samples. In this example, we consider that the first output follows an overdamped response, while the second output has an underdamped response. Additionally, the input times comprises 100 values in the range from 0s to 3s for each output. Just to have a quantitative measure of the approximation obtained by the RFRF approach, the Frobenius norm between the covariance matrices shown in figure 2 is 239.1. However, for $S = 10^5$ samples, the Frobenius norm is 5.8, which states that we are able to reduce the approximation error by the cost of increasing the number of samples. Note that the covariance values are similar,

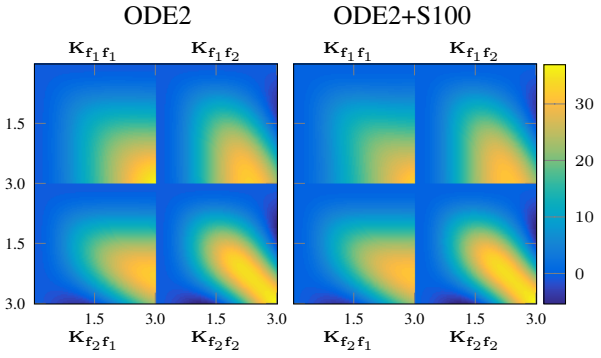

Figure 2: Comparison of the covariance matrix evaluation using the standard LFM and the RFRF.

indicating that the correlation between the outputs and within each output is preserved and well approximated by the inner products of the random features $v_d^{(2)}(t,\theta_d,\lambda)$.

For the following experiments, we consider two motion capture (MOCAP) datasets, [2] which consist of measured joint angles from different types of motions. Additionally, the variational approach is configured with 25 inducing variables, six latent forces and the maximum number of iterations set to 500.

MOCAP - Golf swing In this experiment, we consider the movement "Golf swing" performed by subject 64 motion 01. From the 62 available channels, we selected 56 each having 448 samples, except for two outputs where 81 consecutive samples were considered for testing purposes. The complete dataset for training consists of 24926 datapoints.

Table 3: Results for Golf Swing dataset.

Kernel	root-Ypos		lowerback-Yrot		Time
	NMSE	NLPD	NMSE	NLPD	[s]
ODE2+S10	0.39	-2.23	0.98	2.69	2.20
ODE2+S20	0.24	-2.35	1.49	4.30	3.02
ODE2+S50	0.17	-2.39	**0.27**	**1.17**	4.59
ODE2+S100	0.12	**-2.45**	0.32	1.34	9.31
ODE2	**0.11**	-2.39	3.19	7.26	28.96

Table 3 reports the predictive performance using the covariance functions built from the LFM and the proposed RFRF. In this experiment, the RFRF approximations fit better the testing data for output "lowerback-Yrot", as shown in figure 3. In contrast, output "root-Ypos" testing data is best fitted by the standard LFM. In summary, the models learned using 50 and 100 samples not only performed better than the standard LFM, but also their cost time is reduced by a fraction of three and six, respectively.

MOCAP - Walk For this experiment, we consider the movement "walk" from subject 02 motion 01. From the 62 available channels, we selected 48 each having 343 samples, except for 121 and 105 consecutive samples of two outputs that were considered for testing purposes. The complete dataset for training consists of 16238 datapoints.

Table 4 reports the predictive performance for the testing data used in "walk" experiment. Output "lowerback-Yrot" missing data is best fitted by the standard LFM. However, the testing data for output "lradius-Xrot" is best fitted by the proposed RFRF approach, as shown in figure 3. Interestingly, for this experiment, the observed data are smooth, which can be fitted with adequate accuracy using 10 or 20 samples using the RFRF approach.

[2] MOCAP datasets are available at `http://mocap.cs.cmu.edu/`.

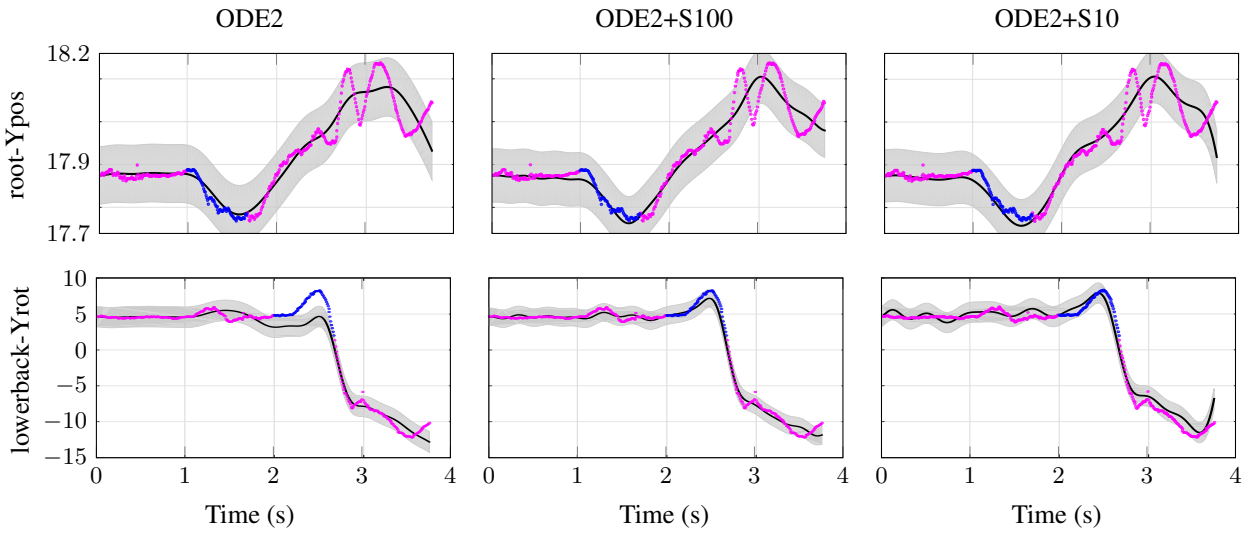

Figure 3: Comparison of the predictive GPs, for the Golf swing experiment, using the standard LFM (first column) and the RFRF approximation for $S = 100$ (second column) and $S = 10$ samples (third column) . Training data is represented using red dots and Test data using blue dots. The black line in the mean over the predictive GP function, and the shaded region denotes two times the standard deviation.

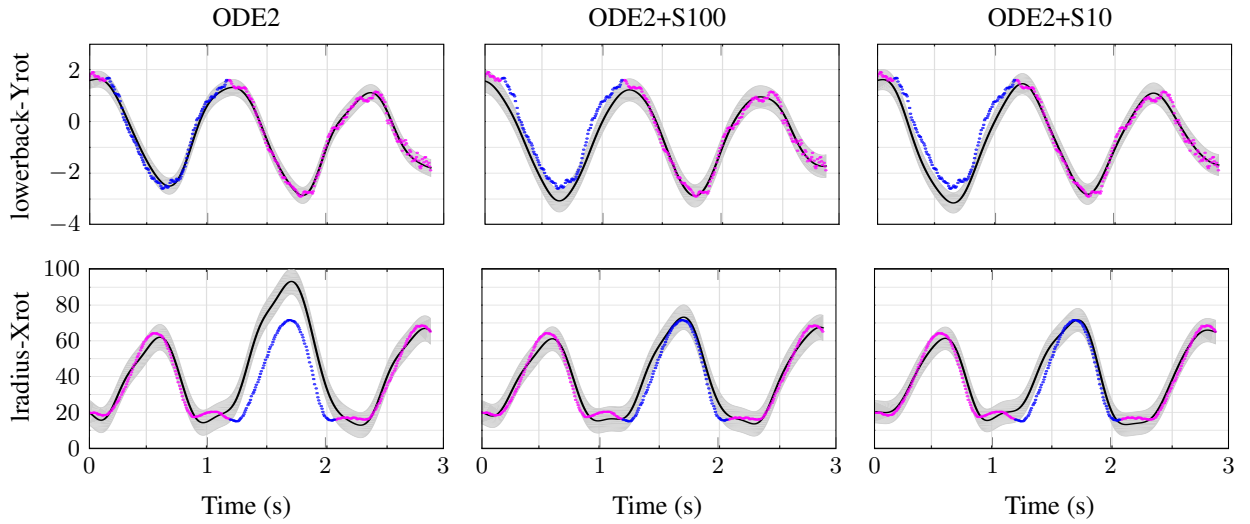

Figure 4: Comparison of the predictive GPs obtained for the the motion "Walk" using the standard LFM (first column) and the RFF approximation for $S = 100$ (third column) and $S = 10$ samples (third column) . Training data is represented using red dots and Test data using blue dots. The black line in the mean over the predictive GP function, and the shaded region denotes two times the standard deviation.

We remark that the evaluation of the covariance function ODE2 is the most expensive one because it requires the evaluation of the Faddeeva function. Hence, the computation time per iteration is reduced using the inner product of $v_d^{(2)}(t, \theta_d, \lambda)$.

6 RANDOM FOURIER FEATURES FOR CONVOLVED MULTIPLE OUTPUT GAUSSIAN PROCESSES

Convolution processes can be used to build kernels for vector-valued functions, as reviewed in Álvarez and Lawrence [2011]. Following similar expressions to the ones in section 3, an output $f_d(\mathbf{x})$, with $\mathbf{x} \in \mathbb{R}^p$, can be modeled as a convolution integral of general smooth-

Table 4: Results for Walk Dataset.

Kernel	lowerback-Yrot		lradius-Xrot		Time
	NMSE	NLPD	NMSE	NLPD	[s]
ODE2+S10	0.21	5.05	0.12	1.06	1.45
ODE2+S20	0.22	2.09	0.49	**0.87**	2.04
ODE2+S50	0.22	4.77	0.19	5.28	3.24
ODE2+S100	0.18	3.35	**0.09**	3.86	6.09
ODE2	**0.02**	**-0.10**	0.99	19.63	19.67

ing kernels $\{G^i_{d,q}(\cdot)\}^{D,Q,R_q}_{d=1,q=1,i=1}$, and latent processes $\{u^i_q(\mathbf{x})\}^{Q,R_q}_{q=1,i=1}$

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q}\sum_{i=1}^{R_q}\int_{\mathcal{X}} G^i_{d,q}(\mathbf{x}-\mathbf{z})u^i_q(\mathbf{z})d\mathbf{z},$$

where, according to Álvarez and Lawrence [2011], the variable R_q makes reference to the number of latent functions u_q that share the same covariance function $k_q(x, x')$, although are sampled independently. Granted that the $u^i_q(\mathbf{x})$ are independent GPs with zero mean and covariance functions $\text{cov}[u^i_q(\mathbf{x}), u^j_{q'}(\mathbf{x}')] = k_q(\mathbf{x}, \mathbf{x}')\delta_{q,q'}\delta_{i,j}$, where $\delta_{q,q'}$ and $\delta_{i,j}$ are Kronecker deltas, the cross-covariance between $f_d(\mathbf{x})$, and $f_{d'}(\mathbf{x}')$, $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$, follows a familiar form

$$\sum_{q=1}^{Q}\sum_{i=1}^{R_q}\int_{\mathcal{X}} G^i_{d,q}(\mathbf{x}-\mathbf{z})\int_{\mathcal{X}} G^i_{d',q}(\mathbf{x}'-\mathbf{z}')k_q(\mathbf{z},\mathbf{z}')d\mathbf{z}d\mathbf{z}'.$$

This covariance function subsumes several other covariance functions proposed in the literature for multiple output GPs, including the linear model of coregionalization [Álvarez and Lawrence, 2011].

A general purpose expression for $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$ can be obtained by assuming that both $G^i_{d,q}(\cdot)$ and $k_q(\cdot,\cdot)$ follow Gaussian forms. The cross-covariance $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$ would then also follow a Gaussian form after solving the double integration for $\mathcal{X} = \mathbb{R}^p$. The authors in Álvarez and Lawrence [2011] provided a closed-form expression for $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$ for this case, when $R_q = 1$.

We can also use random Fourier features for $k_q(\cdot,\cdot)$ in the expression above. For the Gaussian case, since the integrations are over $\mathbb{R}^p$, we use a Fourier transform instead of a Laplace transform as it was the case for the LFM. Let us assume that both $G_{d,q}(\cdot)$ and $k_q(\cdot,\cdot)$ follow Gaussian forms,

$$G_{d,q}(\boldsymbol{\tau}) = \exp\left[-\frac{P_d}{2}\boldsymbol{\tau}^\top\boldsymbol{\tau}\right],$$
$$k_q(\mathbf{z},\mathbf{z}') = \exp\left[-\frac{1}{\ell_q^2}(\mathbf{z}-\mathbf{z}')^\top(\mathbf{z}-\mathbf{z}')\right],$$

where P_d is the inverse-width associated to the smoothing kernel for output d, and ℓ_q is the length-scale for the kernel of the latent function. The cross-covariance $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$ follows as

$$\sum_{q=1}^{Q} S_{d,q}S_{d,q'}\int_{\mathcal{X}}\int_{\mathcal{X}}\exp\left[-\frac{P_d}{2}(\mathbf{x}-\mathbf{z})^\top(\mathbf{x}-\mathbf{z})\right]$$
$$\times\exp\left[-\frac{P_{d'}}{2}(\mathbf{x}'-\mathbf{z}')^\top(\mathbf{x}'-\mathbf{z}')\right]k_q(\mathbf{z},\mathbf{z}')d\mathbf{z}d\mathbf{z}'.$$

Using again the Bochner's theorem for $k_q(\mathbf{z}, \mathbf{z}')$,

$$k_q(\mathbf{z},\mathbf{z}') = \int p(\boldsymbol{\lambda})\exp(j\boldsymbol{\lambda}^\top(\mathbf{z}-\mathbf{z}'))d\boldsymbol{\lambda}.$$

Placing this form for $k_q(\mathbf{z}, \mathbf{z}')$ inside the expression for $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$, and solving the integral over $\boldsymbol{\lambda}$ using Monte Carlo, we get that $k_{f_d,f_{d'}}(\mathbf{x}, \mathbf{x}')$ follows

$$\sum_{q=1}^{Q}\frac{S_{d,q}S_{d,q'}}{S}\boldsymbol{\phi}_d^\top(\mathbf{x}, P_d, \boldsymbol{\Lambda}_q)\boldsymbol{\phi}^*_{d'}(\mathbf{x}', P_{d'}, \boldsymbol{\Lambda}_q),$$

where

$$\boldsymbol{\phi}_d(\mathbf{x}, P_d, \boldsymbol{\Lambda}_q) = \exp\left[-\frac{1}{2P_d}\mathbf{b}_q + j\boldsymbol{\Lambda}_q\mathbf{x}\right],$$

with $\mathbf{b}_q = \sum_j(\boldsymbol{\Lambda}_q \odot \boldsymbol{\Lambda}_q)_{i,j} \in \mathbb{R}^{S\times 1}$, being $\odot$ the Hadamard product, and $\boldsymbol{\Lambda}_q = \frac{1}{\ell_q}\mathbf{Z} \in \mathbb{R}^{S\times p}$, where the entries of the matrix $\mathbf{Z}$ are sampled from $\mathcal{N}(0, 1)$. Hyperparameters θ_d and ℓ_q can be estimated using similar procedures to the ones described in section 4.

SARCOS As an illustration of the use of the kernel above, we performed an experiment on a subset of the SARCOS dataset described in the book by Rasmussen and Williams [2006].[3] We use a subset of the data in the file `sarcos_inv.mat`. In particular, we randomly select 10000 data observations that include two outputs, corresponding to the first two joint torques, and the first seven inputs, corresponding to the joint positions. We then randomly select 1000 observations for the second output as the test data. We use the remaining 19000 for training, this is, for hyperparameter optimization. We compare the performance between the kernel proposed in Álvarez and Lawrence [2011] (CMOC) and the kernel obtained using the random Fourier response features for different values of S. For the CMOC we optimize the marginal likelihood as in Eq. (5), whereas for the RFRF, we use the marginal likelihood as in Eq. (6). Table 5 reports the NMSE and NLPD for the 1000 test observations for the second output. These experiments were carried out using a single core of an Intel Xeon E5-2630v3 @ 2.4 GHz.

[3] Available at `http://www.gaussianprocess.org/gpml/data/`

Table 5: Results for the Sarcos Experiment.

Kernel	NMSE	NLPD	Time [s]
RFF+GG+S50	0.34	3.58	10.14
RFF+GG+S100	0.30	3.52	18.47
RFF+GG+S200	0.26	3.44	38.55
RFF+GG+S500	0.24	3.41	64.62
RFF+GG+S1000	0.22	3.36	85.00
CMOC	**0.19**	**3.21**	353.00

We notice that the performance of the approximation increases with S, and approaches the performance of CMOC, keeping the computation time per iteration to a fraction of the original one. As it was also expected, in higher dimensions, we need a larger number of random features to approach the performance of the CMOC.

7 RELATED WORK

Random Fourier features have been used in the literature for Gaussian processes before. For example, in Bonilla et al. [2016], the authors use RFFs in order to propose a multi-task GP model that circumvents the scalability problem of the GPs. Their model for the multiple outputs uses an affine transformation of the random features, whereas we use a non-instantaneous transformation via the Green's functions. Also in Yang et al. [2015], the authors use a faster approximation of random Fourier features via the FastFood kernels [Le et al., 2013], for approximating the kernel functions of a GP. Their method is not used for multiple outputs, nor does include dynamical systems.

Latent force models have been also studied using a state-space formulation [Hartikainen and Särkkä, 2011] and in that line of research, low-rank approximations for computing features have also been introduced [Solin and Särkkä, 2014]. Specifically, this work approximates the covariance function using the Laplace operator eigenvalues and eigenfunctions. This formulation has been used in Svensson et al. [2016] to approximate the GP priors that are placed over the functions that transform the state vector in the update state and observation equations. Thus, it has not been considered to approximate the GP model of the excitation function.

Brault et al. [2016] directly build random Fourier features for vector-valued kernels using an operator-valued version of Bochner's theorem. The construction is applied to the decomposable kernel, the curl-free kernel and the div-free kernel. In our construction, rather than starting with a fixed form for the operator-valued kernel, we use a general mechanism used to build valid operator kernel functions and apply linear operators over the random Fourier features defined for single output kernels.

8 CONCLUSIONS AND FUTURE WORK

We have shown in this paper how to use random Fourier features for easing the computation of the kernel functions associated to LFMs. As a by-product, we have also reduced the computational complexity of working in multiple-output GPs from $\mathcal{O}(D^3N^3)$ to $\mathcal{O}(DNQ^2S^2)$. We showed experiments over datasets of different sizes for which results with LFM are slow to compute. Our random Fourier response features reduce computational time without compromising performance. Also, notice that by having decoupled the solution of the convolution integrals from the particular form for the kernel of the latent functions, we now can easily build kernels for latent force models with different kernel functions in the GPs of the latent functions, just by changing the distribution $p(\lambda)$ from which we sample from.

These novel representations of latent force models open the path for different types of future work: the application of random Fourier response features for building more efficient versions of sequential LFM [Álvarez et al., 2011] and hierarchical LFM [Honkela and et al., 2010]; the use of physically inspired Fourier features in other Gaussian process models, particularly, deep models [Cutajar et al., 2017]; the use of more efficient sampling techniques for obtaining the Fourier features, e.g. Quasi-Monte Carlo sampling [Avron et al., 2016]. With a more efficient way to compute kernels for multiple-outputs, we can also use more expensive model selection approaches, for example, those based on automatic composition of kernel functions [Duvenaud and et al., 2013], for building more complex covariance functions, e.g. combinations of first order models and second order models, as sums of kernels or as products of kernels. For the case of convolved multiple outputs GPs where the input dimension is greater than three (compared to typical LFMs), the computation of dense Gaussian matrices can be replaced by the product between Hadamard matrices and diagonal Gaussian matrices, which are faster to compute [Le et al., 2013].

Acknowledgments

CG would like to thank to Convocatoria 567 from Administrative Department of Science, Technology and Innovation of Colombia (COLCIENCIAS) for the support. MAA has been financed by the Engineering and Physical Research Council (EPSRC) Research Project EP/N014162/1.

References

M. A. Álvarez and N. D. Lawrence. Computationally Efficient Convolved Multiple Output Gaussian Processes.

Journal of Machine Learning Research, 12:1425–1466, 2011.

M. A. Álvarez, D. Luengo, and N. D. Lawrence. Latent Force Models. In David van Dyk and M. Welling, editors, *Proceedings of AISTATS 2009*, pages 9–16, Clearwater Beach, Florida, 16-18 April 2009. JMLR W&CP 5.

M. A. Álvarez, D. Luengo, M. Titsias, and N. D. Lawrence. Efficient Multioutput Gaussian Processes through Variational Inducing Kernels. In Y.-W. Teh and M. Titterington, editors, *Proceedings of AISTATS 2010*, volume 9 of *Proceedings of Machine Learning Research*, pages 25–32, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010.

M. A. Álvarez, J. Peters, B. Schölkopf, and N. D. Lawrence. Switched Latent Force Models for Movement Segmentation. In J. Shawe-Taylor, R. Zemel, C. Williams, and J. Lafferty, editors, *Advances in Neural Information Processing Systems 24*, pages 55–63. MIT, 2011.

M. A. Álvarez, D. Luengo, and N. D. Lawrence. Linear Latent Force Models using Gaussian Processes. *IEEE TPAMI*, 35(11):2693–2705, 2013.

H. Avron, V. Sindhwani, J. Yang, and M. W. Mahoney. Quasi-Monte Carlo Feature Maps for Shift-Invariant Kernels. *Journal of Machine Learning Research*, 17 (120):1–38, 2016.

E. Bonilla, D. Steinberg, and A. Reid. Extended and Unscented Kitchen Sinks. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of ICML 2016*, volume 48 of *Proceedings of Machine Learning Research*, pages 1651–1659, New York, New York, USA, 20–22 Jun 2016.

R. Brault, M. Heinonen, and F. Buc. Random Fourier Features For Operator-Valued Kernels. In Robert J. Durrant and Kee-Eung Kim, editors, *Proceedings of The 8th Asian Conference on Machine Learning*, volume 63 of *Proceedings of Machine Learning Research*, pages 110–125, 2016.

K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone. Random Feature Expansions for Deep Gaussian Processes. In Doina Precup and Yee Whye Teh, editors, *Proceedings of ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 884–893, International Convention Centre, Sydney, Australia, 06–11 Aug 2017.

D. Duvenaud and et al. Structure Discovery in Nonparametric Regression through Compositional Kernel Search. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of ICML 2013*, volume 28 of *Proceedings of Machine Learning Research*, pages 1166–1174, Atlanta, Georgia, USA, 17–19 Jun 2013.

P. Gao, A. Honkela, M. Rattray, and N. D. Lawrence. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, 24(16):i70–i75, 2008.

S. Ghosh and et al. Modeling the thermal dynamics of buildings: A latent-force- model-based approach. *ACM Trans. Intell. Syst. Technol.*, 6(1):7:1–7:27, March 2015.

J. Hartikainen and S. Särkkä. Sequential Inference for Latent Force Models. In *Proceedings of UAI 2011*, pages 311–318, 2011.

A. Honkela and et al. Model-based method for transcription factor target identification with limited data. *Proc. Natl. Acad. Sci.*, 107(17):7793–7798, 2010.

Q. Le, T. Sarlós, and A. Smola. Fastfood: Approximating Kernel Expansions in Loglinear Time. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of ICML 2013*, volume 28 of *Proceedings of Machine Learning Research*, pages 244–252, Atlanta, Georgia, USA, 17–19 Jun 2013.

Trung V. Nguyen and Edwin V. Bonilla. Collaborative Multi-output Gaussian Processes. In *Proceedings of UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*, pages 643–652, 2014.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. ISBN 0-262-18253-X.

Arno Solin and Simo Särkkä. Hilbert Space Methods for Reduced-Rank Gaussian Process Regression. `https://arxiv.org/pdf/1401.5508.pdf`, 2014.

Andreas Svensson, Arno Solin, Simo Särkkä, and Thomas Schön. Computationally Efficient Bayesian Learning of Gaussian Process State Space Models. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of AISTATS 2016*, volume 51 of *Proceedings of Machine Learning Research*, pages 213–221, Cadiz, Spain, 09–11 May 2016. PMLR.

Z. Yang, A. Wilson, A. Smola, and Le Song. À la Carte – Learning Fast Kernels. In G. Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of AISTATS 2015*, volume 38 of *Proceedings of Machine Learning Research*, pages 1098–1106, San Diego, California, USA, 09–12 May 2015.