Linear Latent Force Models Using Gaussian Processes

Mauricio A. Álvarez, David Luengo, *Member*, *IEEE*, and Neil D. Lawrence

Abstract—Purely data-driven approaches for machine learning present difficulties when data are scarce relative to the complexity of the model or when the model is forced to extrapolate. On the other hand, purely mechanistic approaches need to identify and specify all the interactions in the problem at hand (which may not be feasible) and still leave the issue of how to parameterize the system. In this paper, we present a hybrid approach using Gaussian processes and differential equations to combine data-driven modeling with a physical model of the system. We show how different, physically inspired, kernel functions can be developed through sensible, simple, mechanistic assumptions about the underlying system. The versatility of our approach is illustrated with three case studies from motion capture, computational biology, and geostatistics.

1 Introduction

Traditionally, the main focus in machine learning has been model generation through a *data-driven paradigm*. In this paradigm, the approach is to combine a dataset with a (typically fairly flexible) class of models and, through judicious use of regularization, make predictions on previously unseen data. There are two key problems with purely data-driven approaches. First, if data are scarce relative to the complexity of the system we may be unable to make accurate predictions on test data. Second, if the model is forced to extrapolate, i.e., make predictions in a regime in which data have not yet been seen, performance can be poor.

In contrast, purely *mechanistic models*, i.e., models that are inspired by the underlying physical knowledge of the system, are common in many domains such as chemistry, systems biology, climate modeling, and geophysical sciences. They normally make use of a fairly well-characterized physical process that underpins the system, often represented with a set of differential equations. The purely mechanistic approach leaves us with a different set of problems to those from the data driven approach. In particular, accurate description of a complex system through a mechanistic modeling paradigm may not be possible. Even if all the physical processes can be adequately described, the resulting model could become extremely complex. Identifying and specifying all the interactions might not be feasible, and we would still be faced with the problem of identifying the parameters of the system.

Despite these problems, physically well-characterized models retain a major advantage over purely data-driven models. A mechanistic model can enable accurate predictions even in regions where there is no available training data. For example, space probes can enter different extraterrestrial orbits regardless of the availability of data for these orbits.

While data-driven approaches do seem to avoid mechanistic assumptions about the data, the regularization which is applied normally encodes some kind of physical intuition, such as the smoothness of the interpolant. This reflects a weak underlying belief about the mechanism that generated the data. In this sense, the data-driven approach can be seen as *weakly mechanistic*, whereas models based on more detailed mechanistic relationships could be seen as *strongly mechanistic*.

The observation that weak mechanistic assumptions underlie a data driven model inspires our approach. We suggest a *hybrid system* that incorporates a (typically overly simplistic) mechanistic model within a data-driven approach. The key is to retain sufficient flexibility in our model to be able to fit the system even when our mechanistic assumptions are not rigorously fulfilled. To illustrate the framework, we will start by considering dynamical systems as latent variable models that incorporate ordinary differential equations (ODEs). In this we follow the work of Lawrence et al. [1], [2], who encoded a first order differential equation in a Gaussian process (GP). Their aim was to construct an accurate model of transcriptional regulation, whereas ours is to make use of the mechanistic model to incorporate salient characteristics of the data (e.g., in a mechanical system *inertia*) without necessarily associating the components of our mechanistic

model with actual physical components of the system. We then show how partial differential equations models can also be used for systems with spatial inputs, thereby extending our framework to multidimensional inputs.

The latent force modeling framework introduced here is related to multiple output Gaussian processes through convolution processes [35], and to collocation methods with Gaussian processes [48]. In multiple output Gaussian processes through convolution processes, the covariance functions usually employed are very general, and do not include any mechanistic assumptions about the data. In collocation methods with Gaussian processes, the interest is toward finding a solution to a linear differential equation, while ours is to develop probabilistic models that incorporate mechanistic ideas in data-driven models. An exhaustive comparison with related work is provided in Section 6.

Part of this work has been previously presented in [10]. The main differences of this paper with [10] include an extended description of the latent force model (LFM), with particular focus on how to obtain the covariance functions involved, additional results in motion capture data, and the formulation of a new spatiotemporal covariance function derived from a partial differential equation.

The paper is organized as follows: In Section 2, we motivate the latent force model starting with a latent variable model. Section 3 defines a latent force model in terms of ordinary and partial differential operators. In Section 4, we provide details for learning a latent force model. We then proceed to show three case studies in Section 5. We use a latent force model based on a second order ordinary differential equation for characterizing motion capture datasets. We also present a latent force model for spatiotemporal domains applied to representing the development of Drosophila Melanogaster, and a latent force model inspired by a diffusion process for explaining the behavior of pollutant metals in the Swiss Jura. Extensive related work is presented in Section 6. Final conclusions are given in Section 7.

2 Motivation: From Latent Variables to Latent Forces

A key challenge in combining the mechanistic and data-driven approaches is how to incorporate the model flexibility associated with the data-driven approach within the mechanism. We choose to do this through latent variables, more precisely, latent functions: unobserved functions from the system. To see how this is possible we first introduce some well-known data-driven models from a mechanistic latent-variable perspective.

Let us assume we wish to summarize a high-dimensional dataset with a reduced dimensional representation. For example, if our data consist of N points in a D-dimensional space we might seek a linear relationship between the data, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_D] \in \mathbb{R}^{N \times D}$ with $\mathbf{y}_d \in \mathbb{R}^{N \times 1}$, and a reduced dimensional representation, $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_Q] \in \mathbb{R}^{N \times Q}$ with $\mathbf{u}_q \in \mathbb{R}^{N \times 1}$, where $Q < D$. From a probabilistic perspective, this involves an assumption that we can represent the data as

$$\mathbf{Y} = \mathbf{U}\mathbf{W}^\top + \mathbf{E}, \tag{1}$$

where $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_D]$ is a matrix-variate Gaussian noise: Each column, $\mathbf{e}_d \in \mathbb{R}^{N \times 1}$ $(1 \leq d \leq D)$, is a multivariate Gaussian with zero mean and covariance $\mathbf{\Sigma}$, i.e., $\mathbf{e}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_d)$. The usual approach, as undertaken in factor analysis and principal component analysis (PCA), to dealing with the unknown latent variables in this model is to integrate out $\mathbf{U}$ under a Gaussian prior and optimize with respect to $\mathbf{W} \in \mathbb{R}^{D \times Q}$ (although it turns out that for a nonlinear variant of the model it can be convenient to do this the other way around; see, for example, [3]). If the data have a temporal nature, then the prior over the latent space could express a relationship between the rows of $\mathbf{U}$, $\mathbf{u}_{t_n} = \mathbf{\Gamma}\mathbf{u}_{t_{n-1}} + \boldsymbol{\eta}$, where $\mathbf{\Gamma}$ is a transformation matrix, $\boldsymbol{\eta}$ is a Gaussian random noise, and $\mathbf{u}_{t_n}$ is the nth row of $\mathbf{U}$, which we associate with time t_n. This is known as the *Kalman filter/smoother*. Normally, the times, t_n, are taken to be equally spaced, but more generally we can consider a joint distribution for $p(\mathbf{U} \mid \mathbf{t})$, for a vector of time inputs $\mathbf{t} = [t_1 \ldots t_N]^\top$, which has the form of a Gaussian process:

$$p(\mathbf{U} \mid \mathbf{t}) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{u}_q \mid \mathbf{0}, \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}),$$

where we have assumed zero mean and independence across the Q dimensions of the latent space. The GP makes explicit the fact that the latent variables are functions, $\{u_q(t)\}_{q=1}^{Q}$, and we have now described them with a process prior. The elements of the vector $\mathbf{u}_q = [u_q(t_1), \ldots, u_q(t_N)]^\top$ represent the values of the function for the qth dimension at the times given by $\mathbf{t}$. The matrix $\mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}$ is the covariance function associated with $u_q(t)$ computed at the times given in $\mathbf{t}$.

Such a GP can be readily implemented. Given the covariance functions for $\{u_q(t)\}_{q=1}^{Q}$, the implied covariance functions for $\{y_d(t)\}_{d=1}^{D}$ are straightforward to derive. In [4], this is known as a semiparametric latent factor model (SLFM), although their main focus is not the temporal case. If the latent functions $u_q(t)$ share the same covariance but are sampled independently, this is known as the multitask Gaussian process prediction model (MTGP) [5], with a similar model introduced in [6]. Historically, the Kalman filter approach has been preferred, perhaps because of its linear computational complexity in N. However, recent advances in sparse approximations have made the general GP framework practical (see [7] for a review).

So far the model described relies on the latent variables to provide the dynamic information. Our main contribution is to include a further dynamical system with a *mechanistic* inspiration. We will make use of a mechanical analogy to introduce it. Consider the following physical interpretation of (1): The latent functions, $u_q(t)$, are Q forces and we observe the displacement of D springs, $y_d(t)$, to the forces. Then, we can reinterpret (1) as the force balance equation, $\mathbf{Y}\mathbf{B} = \mathbf{U}\mathbf{S}^\top + \tilde{\mathbf{E}}$. Here, we have assumed that the forces are acting, for example, through levers, so that we have a matrix of sensitivities, $\mathbf{S} \in \mathbb{R}^{D \times Q}$, and a diagonal matrix of spring constants, $\mathbf{B} \in \mathbb{R}^{D \times D}$, with elements $\{B_d\}_{d=1}^{D}$. The original model is recovered by setting $\mathbf{W}^\top = \mathbf{S}^\top \mathbf{B}^{-1}$ and $\tilde{\mathbf{e}}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{B}^\top \mathbf{\Sigma}_d \mathbf{B})$. With appropriate choice of latent density and noise model this physical model underlies the Kalman filter, PCA,

independent component analysis, and the multi-output Gaussian process models we mentioned above.

The use of latent variables means that despite the simplicity of the underlying mechanistic model and the strong associated physical constraints, these models are still powerful enough to be applied to a range of real world datasets. In latent force models, we retain this flexibility by maintaining the latent variables at the heart of the system and introducing richer underlying physical models. For example, we could assume that the springs are acting in parallel with dampers and that the system has mass, allowing us to write

$$\ddot{\mathbf{Y}}\mathbf{M} + \dot{\mathbf{Y}}\mathbf{C} + \mathbf{Y}\mathbf{B} = \mathbf{U}\mathbf{S}^\top + \widehat{\mathbf{E}}, \tag{2}$$

where $\mathbf{M}$ and $\mathbf{C}$ are diagonal matrices of masses, $\{M_d\}_{d=1}^D$, and damping coefficients, $\{C_d\}_{d=1}^D$, respectively, $\dot{\mathbf{Y}}$ is the first derivative of $\mathbf{Y}$ with respect to time (with entries $\{\dot{y}_d(t_n)\}$ for $d = 1, \ldots, D$ and $n = 1, \ldots, N$), $\ddot{\mathbf{Y}}$ is the second derivative of $\mathbf{Y}$ with respect to time (with entries $\{\ddot{y}_d(t_n)\}$ for $d = 1, \ldots D$ and $n = 1, \ldots, N$), and $\widehat{\mathbf{E}}$ is once again matrix-variate Gaussian noise. Equation (2) specifies a particular type of interaction between the outputs $\mathbf{Y}$ and the set of latent functions $\mathbf{U}$, namely, that a weighted sum of the second derivative for $y_d(t)$, $\ddot{y}_d(t)$, the first derivative for $y_d(t)$, $\dot{y}_d(t)$, and $y_d(t)$ is equal to the weighted sum of functions $\{u_q(t)\}_{q=1}^Q$ plus a random noise. The second order mechanical system that this model describes will exhibit characteristics that cannot be accommodated by the standard latent variable set up given in (1), such as inertia and resonance. Of course, the model is not only appropriate for data from mechanical systems. There are many analogous systems that can also be represented by second order differential equations, for example, Resistor-Inductor-Capacitor circuits. A unifying characteristic for all these models is that the system is being forced by latent functions, $\{u_q(t)\}_{q=1}^Q$. Hence, we refer to these models as *latent force models*. The general framework of the latent force model is to combine a mechanistic model with a probabilistic prior over some latent variable or function.

3 Latent Force Models

In the last section, we motivated the latent force model from latent variable models. Here, we look at general characteristics of latent force models. The order of a latent force model is given by the differential equation used to describe the mapping between the latent force and the output functions. A dynamical latent force model of order M employs ordinary differential equations, and the input variable considered is time. In general, we can consider latent force models over multidimensional inputs (e.g., temporospatial systems) through partial differential equations.

3.1 Definition

In general, a dynamical latent force model of order M can be described by the following equation:

$$\sum_{m=0}^{M} \mathcal{D}^m[\mathbf{Y}]\mathbf{A}_m = \mathbf{U}\mathbf{S}^\top + \widehat{\mathbf{E}}, \tag{3}$$

where $\mathcal{D}^m$ is a linear differential operator such that $\mathcal{D}^m[\mathbf{Y}]$ is a matrix with elements given by $\mathcal{D}^m y_d(t) = \frac{\mathrm{d}^m y_d(t)}{\mathrm{d}t^m}$, and $\mathbf{A}_m$ is a diagonal matrix[1] with elements $A_{m,d}$ that weight the contribution of $\mathcal{D}^m y_d$.

Each element in expression (3) can be written as

$$\mathcal{D}_0^M y_d = \sum_{m=0}^{M} A_{m,d}\mathcal{D}^m y_d(t) = \sum_{q=1}^{Q} S_{d,q}u_q(t) + \hat{e}_d(t), \tag{4}$$

where we have introduced a new operator $\mathcal{D}_0^M$ that is equivalent to applying the weighted sum of operators $\mathcal{D}^m$. It is possible to find a linear integral operator $\mathcal{G}_d$ associated with $\mathcal{D}_0^M$ that can be used to solve the nonhomogeneous differential equation in (4). The linear integral operator is defined as

$$\mathcal{G}_d[v](t) = f_d(t, v(t)) = \int_{\mathcal{T}} G_d(t, \tau)v(\tau)\mathrm{d}\tau, \tag{5}$$

where $G_d(t, s)$ is known as the Green's function associated with the differential operator $\mathcal{D}_0^M$, $v(t)$ is the input function for the nonhomogeneous differential equation and $\mathcal{T}$ is the input domain. The particular relation between the differential operator and the Green's function is given by

$$\mathcal{D}_0^M[G_d(t, s)] = \delta(t - s), \tag{6}$$

with s fixed and $\delta(t - s)$ the Dirac delta function [8]. Strictly speaking, the differential operator in (6) is the adjoint for the differential operator appearing in (4). For a more rigorous introduction to Green's functions applied to differential equations refer to [9]. In the signal processing and control theory literatures, the Green's function is known as the impulse response of the system. Without loss of generality, we can set all initial conditions to zero and write the outputs as

$$y_d(t) = f_d(t) + w_d(t) = \sum_{q=1}^{Q} S_{d,q}f_d(t, u_q(t)) + w_d(t),$$

where $f_d(t) = \sum_{q=1}^{Q} S_{d,q}f_d(t, u_q(t))$, $f_d(t, u_q(t)) = \mathcal{G}_d[u_q](t)$, and $w_d(t)$ is an independent process associated with each output. Strictly speaking, the solution of the differential equation implies that $w_d(t) = \mathcal{G}_d[\hat{e}_d](t)$. However, we allow the noise model to be a more general process.

We assume that the latent functions $\{u_q(t)\}_{q=1}^Q$ are independent and each of them follows a Gaussian process prior, that is, $u_q(t) \sim \mathcal{GP}(0, k_{u_q,u_q}(t, t'))$.[2] Due to the linearity of $\mathcal{G}_d$, $\{y_d(t)\}_{d=1}^D$ corresponds to a joint Gaussian process with covariances $k_{y_d,y_{d'}}(t, t') = \mathrm{cov}[y_d(t), y_{d'}(t')]$ given by

$$\mathrm{cov}[f_d(t), f_{d'}(t')] + \mathrm{cov}[w_d(t), w_{d'}(t')]\delta_{d,d'},$$

where $\delta_{d,d'}$ is the Kronecker delta[3] and $\mathrm{cov}[f_d(t), f_{d'}(t')]$ is given by

$$\sum_{q=1}^{Q} S_{d,q}S_{d',q}\mathrm{cov}\left[f_d^q(t), f_{d'}^q(t')\right], \tag{7}$$

1. The matrices $\mathbf{A}_m$ do not need to be diagonal, but for simplicity of derivation we restrict ourselves to this set up in this exposition.

2. Nonzero prior means or correlations between latent functions are also feasible, but again for expositional simplicity we restrict ourselves to these simpler cases.

3. We have used similar notation for the Kronecker delta and the Dirac delta. The particular meaning should be understood from the context.

where we use $f_d^q(t)$ as shorthand for $f_d(t, u_q(t))$. The covariance $\text{cov}[f_d^q(t), f_{d'}^q(t')]$ is equal to

$$\int_{\mathcal{T}} \int_{\mathcal{T}'} G_d(t-\tau) G_{d'}(t'-\tau') k_{u_q,u_q}(\tau,\tau') \mathrm{d}\tau' \mathrm{d}\tau. \quad (8)$$

We alternatively denote $\text{cov}[f_d(t), f_{d'}(t')]$ as $k_{f_d,f_{d'}}(t,t')$, $\text{cov}[f_d^q(t), f_{d'}^q(t')]$ as $k_{f_d^q,f_{d'}^q}(t,t')$, and $\text{cov}[w_d(t), w_{d'}(t')]$ as $k_{w_d,w_{d'}}(t,t')$.

Notice from (8) above that the covariance between $f_d^q(t)$ and $f_{d'}^q(t')$ depends on the covariance $k_{u_q,u_q}(\tau,\tau')$. The form for the covariance $k_{u_q,u_q}(t,t')$ should be such that we can solve both integrals in (8). Two alternatives for $k_{u_q,u_q}(\tau,\tau')$ have been considered before in the context of latent force models. In [10], the covariance $k_{u_q,u_q}(\tau,\tau')$ was considered to follow the squared exponential (SQEXP) form [11]:

$$k_{u_q,u_q}(t,t') = \exp\left(-\frac{(t-t')^2}{\ell_q^2}\right), \quad (9)$$

where ℓ_q is known as the length-scale. In [12], the covariance $k_{u_q,u_q}(\tau,\tau')$ was associated with a Gaussian white noise and therefore followed the form $k_{u_q,u_q}(\tau,\tau') = \sigma_q^2 \delta(\tau-\tau')$, where σ_q^2 stands for the variance of the white noise. As long as the double integral in (8) can be solved analytically, other forms for $k_{u_q,u_q}(\tau,\tau')$ can be taken into account. Possible choices include particular forms of the Matérn class of covariance functions (see the Matérn covariance for $\nu = 3/2$ and $\nu = 5/2$ [11, p. 85]), and the exponential covariance function.

Besides computing the covariance between the outputs, we can also compute the covariance between the outputs and the latent forces. The covariance between $f_d(t)$ and $u_q(t)$, $k_{f_d,u_q}(t,t')$, follows:

$$S_{d,q} \int_{\mathcal{T}} G_d(t-\tau) k_{u_q,u_q}(\tau,t') \mathrm{d}\tau. \quad (10)$$

In Section 5.1, we apply a second order dynamical latent force model to modeling human motion capture data.

3.2 Multidimensional Inputs

In dynamical latent force models, the input variable is one-dimensional (time). For higher-dimensional inputs, $\mathbf{x} \in \mathbb{R}^p$, we can use partial differential equations to establish the dependence relationships between the latent forces, $\{u_q(\mathbf{x})\}_{q=1}^Q$, and the outputs, $\{y_d(\mathbf{x})\}_{d=1}^D$. The initial conditions turn into boundary conditions, specified by a set of functions that are linear combinations of $y_d(\mathbf{x})$ and its lower derivatives, evaluated at a set of specific points of the input space. Once the Green's function associated with the linear partial differential operator has been established, we employ similar equations to (7), (8), and (10) to compute $k_{f_d,f_{d'}}(\mathbf{x},\mathbf{x}')$ and $k_{f_d,u_q}(\mathbf{x},\mathbf{x}')$. Now the covariance for the outputs is written as $k_{y_d,y_{d'}}(\mathbf{x},\mathbf{x}')$, and is given by $k_{f_d,f_{d'}}(\mathbf{x},\mathbf{x}') + k_{w_d,w_d}(\mathbf{x},\mathbf{x}')$, where $k_{w_d,w_d}(\mathbf{x},\mathbf{x}')$ is the covariance for the independent process $w_d(\mathbf{x})$.

In the context of latent force models, choices for $k_{u_q,u_q}(\mathbf{x},\mathbf{x}')$ have included the Gaussian covariance form [10], and a white noise covariance [12]. Alternatives that may be considered include the Matérn class of covariance functions and the exponential covariance function.

We apply latent force models with general higher-dimensional inputs in Section 5.2.

4 Learning Latent Force Models

We have defined latent force models in terms of differential operators and developed a method to encode differential equations in the covariance function. When the latent forces are governed by Gaussian processes, the resulting covariance function can be used for prediction within the GP framework. Here, we describe hyperparameter learning in LFMs, prediction for test cases, and computational complexity. The description is done in terms of the input space $\mathbf{x} \in \mathbb{R}^p$, the dynamical latent force model (where $\mathbf{x} = t$) being a special case.

4.1 Hyperparameter Learning

Gaussian processes allow us to trivially marginalize the effect of the latent forces, $\{u_q(\mathbf{x})\}_{q=1}^Q$, by focusing only on the covariance for the outputs, $k_{y_d,y_{d'}}(\mathbf{x},\mathbf{x}')$. Given a set of inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ and the parameters θ of the covariance function,[4] the marginal likelihood for the outputs can be written as

$$p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y} \mid \mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma}), \quad (11)$$

where $\mathbf{y} = \text{vec}\mathbf{Y}$,[5] $\mathbf{K}_{\mathbf{f},\mathbf{f}} \in \mathbb{R}^{ND \times ND}$ with each element given by $\text{cov}[f_d(\mathbf{x}_n), f_{d'}(\mathbf{x}'_{n'})]$ for $n = 1, \ldots, N$ and $n' = 1, \ldots, N$, and $\boldsymbol{\Sigma}$ represents the covariance associated with the independent processes $w_d(\mathbf{x})$.

In general, the vector of hyperparameters $\boldsymbol{\theta}$ is unknown, so we estimate it by maximizing the logarithm of the marginal likelihood of (11). This type of estimation is known as type II maximum likelihood, empirical Bayes, or the evidence approximation [13]. The maximization is performed numerically by using a gradient descent method.

4.2 Predictive Distribution and Posterior over the Latent Forces

Prediction for a set of test inputs $\mathbf{X}_*$ is done using standard Gaussian process regression techniques. The predictive distribution is given by

$$p(\mathbf{y}_* \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}_* \mid \boldsymbol{\mu}_*, \mathbf{K}_{\mathbf{y}_*,\mathbf{y}_*}),$$

with $\boldsymbol{\mu}_* = \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma})^{-1}\mathbf{y}$ and $\mathbf{K}_{\mathbf{y}_*,\mathbf{y}_*} = \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma})^{-1}\mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^\top + \boldsymbol{\Sigma}_*$, where we have used $\mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*}$ to represent the evaluation of $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ at the input set $\mathbf{X}_*$. The same meaning is given to the covariance matrix $\mathbf{K}_{\mathbf{f}_*,\mathbf{f}}$.

As part of the inference process, we are also interested in the posterior distribution for the set of latent forces:

$$p(\mathbf{u} \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{u} \mid \boldsymbol{\mu}_{\mathbf{u}|\mathbf{y}}, \mathbf{K}_{\mathbf{u}|\mathbf{y}}),$$

with $\boldsymbol{\mu}_{\mathbf{u}|\mathbf{y}} = \mathbf{K}_{\mathbf{f},\mathbf{u}}^\top(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma})^{-1}\mathbf{y}$ and $\mathbf{K}_{\mathbf{u}|\mathbf{y}} = \mathbf{K}_{\mathbf{u},\mathbf{u}} - \mathbf{K}_{\mathbf{f},\mathbf{u}}^\top(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma})^{-1}\mathbf{K}_{\mathbf{f},\mathbf{u}}$, where $\mathbf{u} = \text{vec}\mathbf{U}$, $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ is a block-diagonal matrix with blocks given by $\mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}$. In turn, the elements of $\mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}$ are given by $k_{u_q,u_q}(\mathbf{x},\mathbf{x}')$. Also, $\mathbf{K}_{\mathbf{f},\mathbf{u}}$ is a matrix with blocks $\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_q}$, where $\mathbf{K}_{\mathbf{f}_d,\mathbf{u}_q}$ has entries given by $k_{f_d,u_q}(\mathbf{x},\mathbf{x}')$.

4. Also known as hyperparameters [11, see 20].

5. $\mathbf{x} = \text{vec}\,\mathbf{X}$ is the vectorization operator that transforms the matrix $\mathbf{X}$ into a vector $\mathbf{x}$. The vector is obtained by stacking the columns of the matrix.

4.3 Efficient Approximations

Learning the hyperparameter vector $\boldsymbol{\theta}$ through the maximization of the logarithm of the marginal likelihood in (11) involves the inversion of the matrix $\mathbf{K}_{\mathrm{f,f}} + \boldsymbol{\Sigma}$, inversion that scales as $\mathcal{O}(D^3N^3)$. For the single output case, this is $D = 1$; different efficient approximations have been introduced in the machine learning literature to reduce computational complexity, including [7], [11], [14], [15], [16], [17]. Recently, [18] introduced an efficient approximation for the case $D > 1$. It is based on the assumption that if only a few number $K < N$ of values of $u(\mathbf{x})$ are known, then the set of outputs $f_d(\mathbf{x}, u(\mathbf{x}))$ are uniquely determined. The approximation obtained shares characteristics with the Partially Independent Training Conditional (PITC) approximation introduced in [7] and the authors of [18] refer to the approximation as the PITC approximation for multiple-outputs. The set of values $\{u(\mathbf{z}_k)\}_{k=1}^K$ are known as inducing variables, and the corresponding set of inputs, $\{\mathbf{z}_k\}_{k=1}^K$, are known as inducing inputs. This terminology has been used before in the case $D = 1$.

A different type of approximation was presented in [12] based on variational methods. It is a generalization of [17] for multiple-output Gaussian processes. The approximation establishes a lower bound on the marginal likelihood and reduces computational complexity to $\mathcal{O}(DNK^2)$. The authors call this approximation Deterministic Training Conditional Variational (DTCVAR) approximation for multiple-output GP regression, borrowing ideas from [7] and [17].

5 Applications

Sections 3 and 4 introduced the basic aspects of latent force models required for using them in practice. In this section, we will illustrate the performance of latent force models in three different real-world applications: modeling time-course data in human-motion datasets, describing the spatiotemporal evolution of gene products in Drosophila, and predicting heavy metal concentrations in a geostatistics application. For all these applications, we will focus on latent force covariances, $k_{u_q,u_q}(\mathbf{x}, \mathbf{x}')$, that follow the squared exponential form. This covariance function leads to latent forces that are infinitely differentiable, along with their corresponding outputs.[6]

5.1 Second Order Dynamical System

One analogy for our model comes through puppetry. A marionette is a representation of a human (or animal) controlled by a limited number of inputs through strings (or rods) attached to the character. In a puppet show, these inputs are the unobserved latent functions, while the movement of the joints in the marionette is the observed output functions. A skilled puppeteer with a well-designed puppet can create a realistic representation of human movement through judicious use of the strings.

Human motion capture data consists of a skeleton and multivariate time courses of angles that summarize the motion. This motion can be modeled with a set of second order differential equations which, due to variations in the centers of mass induced by the movement, are nonlinear. The simplification we consider for the latent force model is to linearize these differential equations, resulting in the following second order system:

$$M_d \frac{\mathrm{d}^2 y_d(t)}{\mathrm{d}t^2} + C_d \frac{\mathrm{d}y_d(t)}{\mathrm{d}t} + B_d y_d(t) = \sum_{q=1}^{Q} S_{d,q} u_q(t) + \hat{e}_d(t).$$

While the above equation is not the correct physical model for our system, it will still be helpful when extrapolating predictions across different motions, as we shall see in the experimental results. The dynamic behavior of this system can exhibit inertia and resonance. Note that the system is overparameterized, and we can assume, without loss of generality, that the masses are equal to one.

For the motion capture data, $y_d(t)$ corresponds to a given observed angle over time, and its derivatives represent angular velocity and acceleration. The system is fully characterized by the undamped natural frequency, $\omega_{0d} = \sqrt{B_d}$, and the damping ratio, $\zeta_d = \frac{1}{2} C_d / \sqrt{B_d}$. Systems with a damping ratio greater than 1 are said to be overdamped, whereas underdamped systems exhibit resonance and have a damping ratio less than 1. For critically damped systems, $\zeta_d = 1$. Undamped systems (i.e., no friction) have $\zeta_d = 0$.

Ignoring the initial conditions, the solution of the second order differential equation is given by the integral operator of (5), with Green's function

$$G_d(t, s) = \frac{1}{\omega_d} \exp(-\alpha_d(t - s)) \sin(\omega_d(t - s)),$$

where $\omega_d = \sqrt{4B_d - C_d^2}/2$ and $\alpha_d = C_d/2$.

According to the general framework described in Section 3, the covariance function between the outputs is obtained by solving expression (8), where $k_{u_q,u_q}(t, t')$ follows the SQEXP form in (9). Solution for $k_{f_d^q, f_{d'}^q}(t, t')$ is then given by [10]

$$\begin{aligned} K_0 \big[& h_q(\widetilde{\gamma}_{d'}, \gamma_d, t, t') + h_q(\gamma_d, \widetilde{\gamma}_{d'}, t', t) + h_q(\gamma_{d'}, \widetilde{\gamma}_d, t, t') \\ & + h_q(\widetilde{\gamma}_d, \gamma_{d'}, t', t) - h_q(\widetilde{\gamma}_{d'}, \widetilde{\gamma}_d, t, t') - h_q(\widetilde{\gamma}_d, \widetilde{\gamma}_{d'}, t', t) \\ & - h_q(\gamma_{d'}, \gamma_d, t, t') - h_q(\gamma_d, \gamma_{d'}, t', t) \big], \end{aligned}$$

where $K_0 = \ell_q \sqrt{\pi}/8\omega_d\omega_{d'}$, $\gamma_d = \alpha_d + j\omega_d$, and $\widetilde{\gamma}_d = \alpha_d - j\omega_d$ and the functions $h_q(\widetilde{\gamma}_{d'}, \gamma_d, t, t')$ follow:

$$h_q(\gamma_{d'}, \gamma_d, t, t') = \frac{\Upsilon_q(\gamma_{d'}, t', t) - e^{-\gamma_d t} \Upsilon_q(\gamma_{d'}, t', 0)}{\gamma_d + \gamma_{d'}},$$

with $\Upsilon_q(\gamma_{d'}, t, t')$:

$$\begin{aligned} & 2e^{\left(\frac{\ell_q^2 \gamma_{d'}^2}{4}\right)} e^{-\gamma_{d'}(t - t')} - e^{\left(-\frac{(t - t')^2}{\ell_q^2}\right)} \mathrm{w}(j z_{d',q}(t)) \\ & - e^{\left(-\frac{(t')^2}{\ell_q^2}\right)} e^{(-\gamma_{d'} t)} \mathrm{w}(-j z_{d',q}(0)), \end{aligned} \tag{12}$$

6. We can apply the framework for less smooth covariance functions (such as the Matérn covariance) but when comparisons between convolved and nonconvolved approaches were made we would need some way of accounting for that smoothness (as measured by the differentiability). The latent force model outputs are convolved versions of the underlying latent function and would therefore always be differentiable one more time than the latent function for the nonconvolved. The squared exponential gives latent functions that are infinitely differentiable. This ensures the smoothness characteristics are the same for both the convolved and nonconvolved models.

and $z_{d',q}(t) = (t-t')/\ell_q - (\ell_q \gamma_{d'})/2$. Note that $z_{d',q}(t) \in \mathbb{C}$, and $\mathrm{w}(jz)$ in (12), for $z \in \mathbb{C}$, denotes Faddeeva's function $\mathrm{w}(jz) = \exp(z^2)\mathrm{erfc}(z)$, where $\mathrm{erfc}(z)$ is the complex version of the complementary error function, $\mathrm{erfc}(z) = 1 - \mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_z^\infty \exp(-v^2) \mathrm{d}v$. Faddeeva's function is usually considered the complex equivalent of the error function because $|\mathrm{w}(jz)|$ is bounded whenever the imaginary part of jz is greater or equal than zero. Using Faddeeva's function is the key to achieving a good numerical stability when computing (12) and its gradients.

Similarly, the cross covariance between latent functions and outputs in (10) is given by $k_{f_d^q,u_q}(t,t') = \frac{\ell_q S_{d,q} \sqrt{\pi}}{j4\omega_d} [\Upsilon_q(\widetilde{\gamma}_d, t, t') - \Upsilon_q(\gamma_d, t, t')]$.

5.1.1 Motion Capture Data

We use motion capture data to illustrate the performance of the second order latent force model. Our motion capture dataset is from the CMU motion capture database.[7] We considered two different categories of movement: golf-swing and walking. For golf-swing, we take subject 64, motions 1, 2, 3, and 4, and for walking we take subject 35, motions 2 and 3; subject 10, motion 4; subject 12, motions 1, 2, and 3; subject 16, motions 15 and 21; subject 7, motions 1 and 2; and subject 8, motions 1 and 2. We refer to the pair subject and motion by the notation $A(B)$, where A refers to the subject and B to the particular motion. Original capture is at 120 frames per second (fps). We downsampled by 4 to obtain 30 fps.[8] Although each movement is described by time courses of 62 angles, we selected only the outputs whose signal-to-noise ratio was over 20 dB. To compute the signal-to-noise ratio, we train a GP regressor for each output, employing a covariance function that is the sum of a squared exponential kernel and a white Gaussian noise, $\sigma_S^2 \exp[-\frac{(\mathrm{x}-\mathrm{x}')^2}{2\ell^2}] + \sigma_N^2 \delta(\mathrm{x}, \mathrm{x}')$, where σ_S^2 and σ_N^2 are variance parameters. For each output, we compute the signal-to-noise ratio as $10 \log_{10}(\sigma_S^2/\sigma_N^2)$. After this preprocessing step, we end up with 50 outputs for the golf-swing example and 33 outputs for the walking example.

For each movement category (walk, golf-swing) the subject repeats the motion several times. We refer to each repeat as an individual "motion." We train on a subset of the motions for each movement and test on a different subset of motions for the same movement category to assess the model's ability to extrapolate.[9] For testing, we condition on time as an input and the following outputs: the three positions associated with the root nodes across all time and the initial position and final position of the figure (we used five frames from both the initial and final positions). For the golf-swing, we use leave-one-out cross-validation, in which one of the $64(B)$ movements is left aside (with $B = 1, 2, 3$ or 4) for testing, while we use the other three for training. For the walking example, we train using motions 35(2), 10(4), 12(1), and 16(15) and validate over all the other motions (8 in total).

TABLE 1
RMSE and R^2 for Golf-Swing and Walking

Movement	Method	RMSE	R^2 (%)
Golf swing	IND GP	21.55 ± 2.35	30.99 ± 9.67
	MTGP	21.19 ± 2.18	45.59 ± 7.86
	SLFM	21.52 ± 1.93	49.32 ± 3.03
	LFM	$\mathbf{18.09 \pm 1.30}$	$\mathbf{72.25 \pm 3.08}$
Walking	IND GP	8.03 ± 2.55	30.55 ± 10.64
	MTGP	7.75 ± 2.05	37.77 ± 4.53
	SLFM	7.81 ± 2.00	36.84 ± 4.26
	LFM	$\mathbf{7.23 \pm 2.18}$	$\mathbf{48.15 \pm 5.66}$

We use the above setup to train an LFM, an MTGP, and an SLFM. We set $Q = 2$ for all models[10] for our comparisons. We use the DTCVAR efficient approximation with $K = 30$ and fixed inducing-points placed equally spaced in the input interval (varying K between 20 and 50 for the walking example did not change results significantly). We also considered a regression model that directly predicts the angles of the body given the orientation of three root nodes using standard independent GPs with SQEXP covariance functions.[11] We determined hyperparameters through maximum likelihood for each model independently. Results for all methods are summarized in Table 1 in terms of root-mean-square error (RMSE) and percentage of explained variance (R^2). In the table, the measure shown is the mean of the measure in the validation set, plus and minus one standard deviation.

The LFM outperforms the other methods both in terms of RMSE and R^2. This is particularly true for the R^2 performance measure, indicating that the LFM generates more realistic motions.[12]

5.2 Partial Differential Equations

In Section 5.1, we considered dynamical latent force models which lead to multi-output Gaussian processes with a single input variable: time. We now apply the methodology alongside partial differential equations to recover multi-output Gaussian processes that are functions of several inputs. We first show an example of spatiotemporal covariance obtained from the latent force model idea and then an example of a covariance function that, using a simplified version of the diffusion equation, allows an expression for higher-dimensional inputs.

5.2.1 Gap-Gene Network of Drosophila Melanogaster

The Gap-gene network in Drosophila Melanogaster is associated with segmentation in early organism development. It is a spatiotemporal system where the expression of

7. The CMU Graphics Lab Motion Capture Database was created with funding from NSF EIA-0196217 and is available at http://mocap.cs.cmu.edu.

8. We selected specific frame intervals for each motion. For 64(1), frames $[120, 400]$; for 64(2), frames $[170, 420]$; for 64(3), frames $[100, 300]$; and for 64(4), frames $[80, 315]$. For 35(2), frames $[55, 338]$; for 10(4), frames $[222, 499]$; for 12(1), frames $[22, 328]$; and for 16(15), frames $[62, 342]$. For all other motions, we use all the frames.

9. We use "to train" or "training" to refer to hyperparameter estimation.

10. We also ran MTGP and SLFM with higher values of Q and obtained similar results.

11. Such a model does not have a concept of time, so it is not possible to augment predictions by including the initial pose and the final pose.

12. For both examples, golf-swing and walking, values of the damping ratios obtained for each output, ζ_d, are sensible. For golf-swing, the range of damping-ratio values is $(0, 0.9)$, and for walking, the range is $(0.1, 0.8)$. These values correspond to underdamped systems, that is, systems that exhibit oscillations. Oscillations appear naturally both in golf-swings and walking.

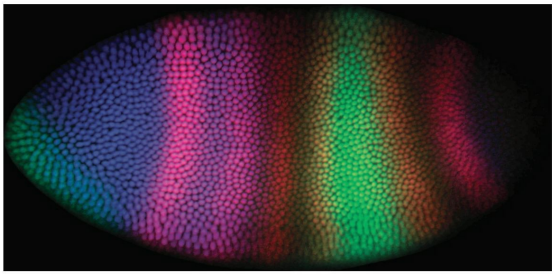

Fig. 1. Drosophila body segmentation genes. Blue stripes correspond to hunchback, green stripes to knirps, and red stripes to eve-skipped at cleavage cycle 14A, temporal class 3.

proteins evolves with time. During the blastoderm stage of the Drosophila development, different maternal gradients determine the polarity of the embryo along its anterior-posterior (A-P) axis.

Maternal gradients interact with the so-called trunk gap genes, including *hunchback (hb), Krüppel (Kr), giant (gt)*, and *knirps (kni)*, and this network of interactions establishes the patterns of segmentation of the Drosophila.

Fig. 1 shows the gene expression of the hunchback, the knirps, and the eve-skipped genes in a color-scale intensity image. The image corresponds to cleavage cycle 14A, temporal class 3.[13]

The gap-gene network dynamics is usually represented using a set of coupled nonlinear partial differential equations [22], [23]:

$$\frac{\partial y_d(x,t)}{\partial t} = \zeta(t)P_d(y(x,t)) - \lambda_d y_d(x,t) + \mathrm{D}_d \frac{\partial^2 y_d(x,t)}{\partial x^2},$$

where $y_d(x,t)$ denotes the relative concentration of gap protein of the dth gene at the space point x and time point t. The term $P_d(y(x,t))$ accounts for production and it is a function, usually nonlinear, of production of all other genes. The parameter λ_d represents the decay and D_d the diffusion rate. The function $\zeta(t)$ accounts for changes occurring during the mitosis in which the transcription is off [22].

We linearize the equation above by replacing the nonlinear term $\zeta(t)P_d(y(x,t))$ with the linear term $\sum_{q=1}^{Q} S_{d,q}u_q(x,t)$, where $S_{d,q}$ are sensitivities that account for the influence of the latent force $u_q(x,t)$ over the quantity of production of gene d. In this way, the new diffusion equation is given by[14]

$$\frac{\partial y_d(x,t)}{\partial t} = \sum_{\forall q} S_{d,q}u_q(x,t) - \lambda_d y_d(x,t) + \mathrm{D}_d \frac{\partial^2 y_d(x,t)}{\partial x^2}.$$

This expression corresponds to a second order nonhomogeneous partial differential equation. It is also parabolic with one space variable and constant coefficients. The exact solution of this equation is subject to particular initial and boundary conditions. For a first boundary value problem with domain $0 \leq x \leq l$, initial condition $y_d(x, t=0)$ equal to zero, and boundary conditions $y_d(x=0,t)$ and $y_d(x=l,t)$ both equal to zero, the solution to this equation is [24], [25], [26]:

$$y_d(x,t) = \sum_{q=1}^{Q} S_{d,q} \int_0^t \int_0^l u_q(\xi,\tau) G_d(x,\xi,t-\tau)\mathrm{d}\xi\mathrm{d}\tau,$$

where the Green's function $G_d(x,\xi,t)$ is

$$\frac{2}{l}e^{-\lambda_d t}\sum_{n=1}^{\infty}\sin\left(\frac{n\pi x}{l}\right)\sin\left(\frac{n\pi\xi}{l}\right)e^{\left(-\frac{\mathrm{D}_d n^2\pi^2 t}{l^2}\right)}.$$

We assume that the latent forces $u_q(x,t)$ follow a Gaussian process with covariance function that factorizes across input dimensions, i.e.,

$$k_{u_q,u_q}(x,t,x',t') = \exp\left(-\frac{(t-t')^2}{\left(\ell_q^t\right)^2}\right)\exp\left(-\frac{(x-x')^2}{\left(\ell_q^x\right)^2}\right),$$

where ℓ_q^t represents the length-scale along the time-input dimension and ℓ_q^x the length-scale along the space input dimension. The covariances $k_{f_d^q,f_{d'}^q}(x,t,x',t')$ are computed using the expression for the Green's function and the expression for the covariance of the latent forces, in a similar fashion to (8), leading to

$$\frac{4}{\ell^2}\sum_{n=1}^{\infty}\sum_{m=1}^{\infty} k^t_{f_d^q,f_{d'}^q}(t,t')k^x_{f_d^q,f_{d'}^q}(x,x'), \tag{13}$$

where $k^t_{f_d^q,f_{d'}^q}(t,t')$ and $k^x_{f_d^q,f_{d'}^q}(x,x')$ are also kernel functions that depend on the indexes n and m. The kernel function $k^t_{f_d^q,f_{d'}^q}(t,t')$ is given by

$$k^t_{f_d^q,f_{d'}^q}(t,t') = \frac{\sqrt{\pi}\ell_q^t}{2}[h_{d',d}(t',t) + h_{d,d'}(t,t')], \tag{14}$$

where

$$\begin{aligned} h_{d',d}(t',t) = \frac{\exp\left(\nu_{q,d'}^2\right)}{\beta_d+\beta_{d'}}\exp(-\beta_{d'}t')\Bigg\{ & \exp(\beta_{d'}t) \\ & \times\left[\mathrm{erf}\left(\frac{t'-t}{\ell_q^t}-\nu_{q,d'}\right)+\mathrm{erf}\left(\frac{t}{\ell_q^t}+\nu_{q,d'}\right)\right] \\ & -\exp(-\beta_d t)\left[\mathrm{erf}\left(\frac{t'}{\ell_q^t}-\nu_{q,d'}\right)+\mathrm{erf}(\nu_{q,d'})\right]\Bigg\}, \end{aligned}$$

where $\mathrm{erf}(x)$ is the real valued error function, $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-y^2)\mathrm{d}y$, $\beta_d = \lambda_d + D_d\omega_n^2$, $\beta_{d'} = \lambda_{d'} + D_{d'}\omega_m^2$, $\omega_n = \frac{n\pi}{\ell}$, $\omega_m = \frac{m\pi}{\ell}$, and $\nu_{q,d} = \ell_q^t\beta_d/2$.

The covariance $k^x_{f_d^q,f_{d'}^q}(x,x')$ is given by

$$k^x_{f_d^q,f_{d'}^q}(x,x') = C(n,m,\ell_q^x)\sin(\omega_n x)\sin(\omega_m x').$$

The term $C(n,m,\ell_q^x)$ represents a function that depends on the indexes n and m and on the length-scale of the space-input dimension. The expression for $C(n,m,\ell_q^x)$ is

$$C(n,m,\ell_q^x) = \int_0^l\int_0^l \sin(w_n\xi)\sin(w_m\xi')e^{\left[-\frac{(\xi-\xi')^2}{(\ell_q^x)^2}\right]}\mathrm{d}\xi'\mathrm{d}\xi.$$

The solution of this double integral depends upon the relative values of n and m. If $n \neq m$ and n and m are both

13. The embryo name is dm12 and the image was taken from http://urchin.spbcas.ru/flyex/ [19], [20], [21].

14. For convenience, we've dropped the noise term, $\hat{e}_d(t)$. However, we include the contribution of the independent process $w_d(x,t)$ with a separate covariance $k_{w_d,w_{d'}}(x,t,x',t')$.

TABLE 2
RMSE and R^2 for Protein Data Prediction

Gene	Method	RMSE	R^2 (%)
giant	MTGP	26.56 ± 0.30	81.12 ± 0.01
	DROS	$\mathbf{2.00 \pm 0.35}$	$\mathbf{99.78 \pm 0.01}$
knirps	MTGP	16.14 ± 8.44	91.18 ± 2.77
	DROS	$\mathbf{3.01 \pm 0.81}$	$\mathbf{99.60 \pm 0.01}$

even or both odd, then the analytical expression for $C(n, m, \ell_q^x)$ is

$$\left(\frac{\ell_q^x l}{\sqrt{\pi}(m^2 - n^2)}\right)\{n\mathcal{I}[\mathcal{W}(m, \ell_q^x)] - m\mathcal{I}[\mathcal{W}(n, \ell_q^x)]\},$$

where $\mathcal{I}[\cdot]$ is an operator that takes the imaginary part of the argument and $\mathcal{W}(m, \ell_q^x)$ is given by

$$\mathcal{W}(m, \ell_q^x) = \mathrm{w}(jz_1^{\gamma_m}) - e^{-\left(\frac{l}{\ell_q^x}\right)^2} e^{-\gamma_m l}\mathrm{w}(jz_2^{\gamma_m}),$$

being $z_1^{\gamma_m} = \frac{\ell_q^x \gamma_m}{2}$, $z_2^{\gamma_m} = \frac{l}{\ell_q^x} + \frac{\ell_q^x \gamma_m}{2}$, and $\gamma_m = j\omega_m$.

The term $C(n, m, \ell_q^x)$ is zero if, for $n \neq m$, n is even and m is odd or vice versa.

Furthermore, when $n = m$, the expression for $C(n, n, \ell_q^x)$ follows as

$$\frac{\ell_q^x \sqrt{\pi}\, l}{2}\left\{\mathcal{R}[\mathcal{W}(n, \ell_q^x)] - \mathcal{I}[\mathcal{W}(n, \ell_q^x)]\left[\frac{(\ell_q^x)^2 n\pi}{2l^2} + \frac{1}{n\pi}\right]\right\} + \frac{(\ell_q^x)^2}{2}\left[e^{-(\frac{l}{\ell_q^x})^2}\cos(n\pi) - 1\right],$$

where $\mathcal{R}[\cdot]$ is an operator that takes the real part of the argument.

The cross covariance between the outputs and the latent functions can be computed using (10).

Results and Discussion. We want to assess the contribution that a simple mechanistic assumption might bring to the prediction of gene expression data when compared to a covariance function that does not imply mechanistic assumptions

We refer to the covariance function obtained in the section before as the Drosophila (DROS) kernel. We use the DROS kernel as the covariance of a GP, and compare its performance against the multitask Gaussian process (MTGP) framework already mentioned in Section 2.

We use data from [22], in particular, we have quantitative wild-type concentration profiles for the protein products of giant and knirps at nine time points and 58 spatial locations. Since there are a fixed number of time points for each protein, we can build a model with a fixed number of outputs and associate each output with a time point. This setup is very common in computer emulation of multivariate codes (see [6], [27], [28]) in which the MTGP model is heavily used. For the DROS kernel, we use 30 terms in each sum involved in its definition in (13).

We randomly select 20 spatial points for training the models, that is, for finding hyperparameters according to the description of Section 4.1. The other 38 spatial points are used for validating the predictive performance. Results are shown in Table 2 for five repetitions of the same experiment. It can be seen that the mechanistic assumption included in the GP model considerably outperforms MTGP for this particular task.

5.2.2 Diffusion in the Swiss Jura

The Jura data are a set of measurements of concentrations of several heavy metal pollutants collected from topsoil in a 14.5 km^2 region of the Swiss Jura. We consider a latent function that represents how the pollutants were originally laid down. As time passes, we assume that the pollutants diffuse at different rates, resulting in the concentrations observed in the dataset. We use a simplified version of the heat equation of p variables. The p-dimensional nonhomogeneous heat equation is represented as[15]

$$\frac{\partial y_d(\mathbf{x}, t)}{\partial t} = \sum_{j=1}^{p} \kappa_{d,j} \frac{\partial^2 y_d(\mathbf{x}, t)}{\partial x_j^2} + \Phi(\mathbf{x}, t),$$

where $p = 2$ is the dimension of $\mathbf{x}$, the measured concentration of each pollutant over space and time is given by $y_d(\mathbf{x}, t)$, $\kappa_{d,j}$ is the diffusion constant of output d in direction p, and $\Phi(\mathbf{x}, t)$ represents an external force, with $\mathbf{x} = \{x_j\}_{j=1}^p$. Assuming the domain $\mathbb{R}^p = \{-\infty < x_j < \infty; j = 1, \ldots, p\}$, and initial condition prescribed by the set of latent forces $u(\mathbf{x}) = \sum_{q=1}^{Q} S_{d,q} u_q(\mathbf{x})$, at $t = 0$, the solution to the system [24] is then given by

$$y_d(\mathbf{x}, t) = \int_0^t \int_{\mathbb{R}^p} G_d(\mathbf{x}, \mathbf{x}', t, \tau)\Phi(\mathbf{x}', \tau)\mathrm{d}\mathbf{x}'\mathrm{d}\tau + \int_{\mathbb{R}^p} G_d(\mathbf{x}, \mathbf{x}', t, 0)u(\mathbf{x}')\mathrm{d}\mathbf{x}', \quad (15)$$

where $G_q(\mathbf{x}, \mathbf{x}', t, \tau)$ is the Green's function given by

$$G_d(\mathbf{x}, \mathbf{x}', t, \tau) = \frac{1}{K_\pi \sqrt{\prod_{j=1}^p T_{d,j}}} \exp\left[-\sum_{j=1}^{p} \frac{(x_j - x_j')^2}{4T_{d,j}}\right],$$

with $K_\pi = 2^p \pi^{p/2}$ and $T_{d,j}(t, \tau) = \kappa_{d,j}(t - \tau)$. The covariance function we propose here is derived as follows: In (15), we assume that the external force $\Phi(\mathbf{x}, t)$ is zero, following:

$$y_d(\mathbf{x}, t) = \sum_{q=1}^{Q} S_{d,q} \int_{\mathbb{R}^p} G_d(\mathbf{x}, \mathbf{x}', t, 0) u_q(\mathbf{x}')\mathrm{d}\mathbf{x}'.$$

We can again write the expression for the Green's function as

$$G_d(\mathbf{x}, \mathbf{x}', t) = \frac{1}{(2\pi)^{\frac{p}{2}} \sqrt{\prod_{j=1}^p \ell_{d,j}}} \exp\left[-\sum_{j=1}^{p} \frac{(x_j - x_j')^2}{2\ell_{d,j}}\right],$$

where $\ell_{d,j} = 2T_{d,j} = 2\kappa_{d,j}t$. The coefficient $\ell_{d,j}$ is a function of time. In our model for the diffusion of the pollutant metals, we think of the data as a snapshot of the diffusion process. Consequently, we consider the time instant of this snapshot as a parameter to be estimated. In other words, the measured concentration is given by

$$y_d(\mathbf{x}) = \sum_{q=1}^{Q} S_{d,q} \int_{\mathbb{R}^p} \widetilde{G}_d(\mathbf{x}, \mathbf{x}') u_q(\mathbf{x}')\mathrm{d}\mathbf{x}',$$

15. For simplicity, we again omit the noise term $\hat{e}_d(t)$.

TABLE 3
RMSE for Pollutant Metal Prediction

Method	Cadmium (Cd)	Cobalt (Co)	Copper (Cu)	Lead (Pb)
IND GP	0.8353 ± 0.0898	2.2997 ± 0.1388	18.9616 ± 3.4404	28.1768 ± 5.8005
MTGP ($Q = 1$)	0.7638 ± 0.1016	2.2892 ± 0.1792	14.4179 ± 2.7119	21.5861 ± 4.1888
HEATK ($Q = 1$)	0.6773 ± 0.0628	2.06 ± 0.0887	13.1788 ± 2.6446	17.9839 ± 2.9450
MTGP ($Q = 2$)	0.6980 ± 0.0832	2.1299 ± 0.1983	12.7340 ± 2.2104	17.9399 ± 1.9981
SLFM ($Q = 2$)	0.6941 ± 0.0834	2.172 ± 0.1204	12.8935 ± 2.6125	17.9024 ± 2.0966
HEATK ($Q = 2$)	$\mathbf{0.6759 \pm 0.0623}$	$\mathbf{2.0345 \pm 0.0943}$	$\mathbf{12.5971 \pm 2.4842}$	$\mathbf{17.5571 \pm 2.6076}$

TABLE 4
R^2 for Pollutant Metal Prediction

Method	Cadmium (Cd)	Cobalt (Co)	Copper (Cu)	Lead (Pb)
IND GP	15.07 ± 7.43	57.81 ± 7.19	25.84 ± 7.54	23.48 ± 10.40
MTGP ($Q = 1$)	27.25 ± 5.89	58.45 ± 5.71	58.84 ± 8.35	56.85 ± 11.60
HEATK ($Q = 1$)	43.83 ± 8.71	66.19 ± 4.60	65.55 ± 8.21	$\mathbf{71.45 \pm 5.78}$
MTGP ($Q = 2$)	40.30 ± 5.17	64.13 ± 5.10	67.51 ± 8.36	69.70 ± 6.90
SLFM ($Q = 2$)	40.97 ± 5.15	62.49 ± 5.41	67.35 ± 8.29	70.21 ± 6.04
HEATK ($Q = 2$)	$\mathbf{43.94 \pm 6.56}$	$\mathbf{67.17 \pm 4.30}$	$\mathbf{68.40 \pm 6.46}$	70.55 ± 6.88

where $\widetilde{G}_d(\mathbf{x}, \mathbf{x}')$ is the Green's function $G_d(\mathbf{x}, \mathbf{x}', t)$ that considers the variable t as a parameter to be estimated through $\ell_{d,j}$. The expression for $\widetilde{G}_d(\mathbf{x}, \mathbf{x}')$ corresponds to a Gaussian smoothing kernel, with diagonal covariance. This is

$$\widetilde{G}_d(\mathbf{x}, \mathbf{x}') = \frac{|\mathbf{P}_d|^{1/2}}{(2\pi)^{p/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top \mathbf{P}_d (\mathbf{x} - \mathbf{x}')\right],$$

where $\mathbf{P}_d$ is a precision matrix, with diagonal form and entries $\{p_{d,j} = \frac{1}{\ell_{d,j}}\}_{j=1}^{p}$.

If we take the latent function to be given by a GP with the Gaussian covariance function that follows the same form as $\widetilde{G}_d(\mathbf{x}, \mathbf{x}')$, we can compute the multiple output covariance functions analytically. The covariance function between the output functions, $k_{f_d^q, f_{d'}^q}(\mathbf{x}, \mathbf{x}')$, is obtained as

$$\frac{1}{(2\pi)^{p/2}|\mathbf{P}_{d,d'}^q|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top (\mathbf{P}_{d,d'}^q)^{-1}(\mathbf{x} - \mathbf{x}')\right],$$

where $\mathbf{P}_{d,d'}^q = \mathbf{P}_d^{-1} + \mathbf{P}_{d'}^{-1} + \mathbf{\Lambda}_q^{-1}$, and $\mathbf{\Lambda}_q$ is the precision matrix associated with the Gaussian covariance of the latent force Gaussian process prior. The covariance function between the output and latent functions can be computed using (10).

Results and Discussion. We used our model to replicate the experiments described in [29, pp. 248, 249] in which a *primary variable* (cadmium, cobalt, copper, and lead) is predicted in conjunction with some *secondary variables* (nickel and zinc for cadmium and cobalt; copper, nickel, and zinc for copper and lead).[16] For several sample locations, we have access to the primary variable, for example, cadmium, and the secondary variables, nickel and zinc. These sample locations are usually referred to as the *prediction set*. At some other locations, we only have access to the secondary variables. In geostatistics, this configuration of sample locations is known as *undersampled* or *heterotopic* [29], where usually a few expensive measurements of the attribute of interest are supplemented by more abundant data on correlated attributes that are cheaper to sample.

By conditioning on the values of the secondary variables at the prediction and validation sample locations and the primary variables at the prediction sample locations, we can improve the prediction of the primary variables at the validation locations. We compare results for the heat kernel with results from prediction using independent GPs for the metals, the multitask Gaussian process, and the semiparametric latent factor model. For our experiments, we made use of 10 repeats to report standard deviations. For each repeat, the data are divided into a different prediction set of 259 locations and different validation set of 100 locations. Root mean square errors and percentage of explained variance are shown in Tables 3 and 4, respectively.

Note from both tables that all methods outperform independent Gaussian processes, in terms of RMSE and explained variance. For one latent function ($Q = 1$), the Gaussian process with Heat kernel renders better results than multitask GPs (in this case, the multitask GP is equivalent to the semiparametric latent factor model). However, when increasing the value of the latent forces to 2 ($Q = 2$), performances for all methods are quite similar. There is a still a gain in performance when using the Heat kernel, although the results are within the standard deviation. Also, when comparing the performances for the GP with Heat kernel using one and two latent forces, we notice that both measures are quite similar. In summary, the heat kernel provides a simplified explanation for the outputs in the sense that, using only one latent force, we provide better performances in terms of RMSE and explained variance.

6 Related Work

When a Gaussian process is used to represent the latent forces and the mechanistic models are linear differential

16. Data available at http://www.ai-geostats.org/.

equations, our framework results in a multiple output Gaussian process with a covariance function that encodes the interactions between the different mechanistic models. By using the marginal likelihood to estimate the hyperparameters $\boldsymbol{\theta}$ of the covariance function embedded in the latent force model, we are estimating the parameters of differential equations.

The related work can be seen from different perspectives. We focus on three: Gaussian processes for multiple outputs, parameter estimation in differential equations, and Gaussian processes for systems identification.

6.1 Gaussian Processes for Multiple Outputs

Gaussian process priors for multiple outputs have been thoroughly studied in the spatial analysis and geostatistics literature [29], [30], [31], [32], [33], [34]. A valid covariance function for multioutput processes can be generated using the linear model of coregionalization (LMC). In the LMC, each output $y_d(t)$ is represented as a linear combination of a series of basic processes $\{u_q\}_{q=1}^{Q}$, some of which share the same covariance function $k_{u_q,u_q}(t,t')$. Both, the semiparametric latent factor model [4] and the multitask GP [5] can be seen as particular cases of the LMC [35]. Higdon [30] proposed the direct use of (5) to obtain a valid covariance function for multiple outputs, and referred to this kind of construction as process convolutions. Process convolutions for constructing covariances for a single output GP had already been proposed by Barry and Hoef [36], [37]. Calder and Cressie [38] review several extensions of the single process convolution covariance. It has been used, for example, to develop nonstationary covariance functions by Paciorek and Schervish [39]. Boyle and Frean [31] introduced the process convolution idea for multiple outputs to the machine learning audience. Boyle [40] suggested the idea of using impulse responses of filters to represent $G_d(t,s)$, assuming the process $v(t)$ was white Gaussian noise. The latent force model generalizes this idea to allow more general covariance functions for the latent processes. Independently, [41] also introduced the idea of transforming a Gaussian process prior using a discretized version of the integral operator of (5). Such a transformation could be applied for the purposes of fusing the information from multiple sensors (a similar setup to the latent force model but with a discretized convolution), for solving inverse problems in reconstruction of images, or for reducing computational complexity working with the filtered data in the transformed space [42].

It is important to emphasize that latent force models are part of the wider process convolution framework for constructing covariance functions. The main difference from previous approaches, including MTGP, SLFM, and LMC, is the inclusion of physical models for constructing the covariance function.

6.2 Parameter Estimation in Differential Equations

Differential equations are the cornerstone of a diverse range of engineering fields and applied sciences. However, combination with probabilistic models and use within machine learning and statistics is less explored. We now briefly review the most significant related works in this area, which fall within a field generally known as *functional data analysis* [43].

From the frequentist statistics point of view functional data analysis has been concerned with the problem of parameter estimation in differential equations [44], [45]: Given a differential equation with unknown coefficients $\{\mathbf{A}_m\}_{m=0}^{M}$, how do we use data to fit those parameters? There is a subtle difference between those techniques and the latent force model. While these parameter estimation methods start with a very accurate description of the interactions in the system via the differential equation (the differential equation is often nonlinear [22]), in the latent force model we use the differential equation as part of the modeling problem: The differential equation is used as a way to introduce prior knowledge over a system for which we do not know the real dynamics, but for which we hope some important features of that dynamics could be expressed. Still, we briefly review the parameter estimation methods because they also deal with differential equations with an uncertainty background.

Classical approaches to fit parameters $\boldsymbol{\theta}$ of differential equations to observed data include numerical approximations of initial value problems and collocation methods ([45] and [46] provide reviews and detailed descriptions of additional methods).

The solution by numerical approximations includes an iterative process in which, given an initial set of parameters $\boldsymbol{\theta}_0$ and a set of initial conditions $\mathbf{y}_0$, a numerical method is used to solve the differential equation. The parameters of the differential equation are then optimized by minimizing an error criterion between the approximated solution and the observed data.

In collocation methods, the solution of the differential equation is approximated using a set of basis functions, $\{\phi_i(t)\}_{i=1}^{J}$, that is, $y(t) = \sum_{i=1}^{J} \beta_i \phi_i(t)$. The basis functions must be sufficiently smooth so that the derivatives of the unknown function appearing in the differential equation can be obtained by differentiation of the basis representation of the solution, that is, $\mathcal{D}^m y(t) = \sum \beta_i \mathcal{D}^m \phi_i(t)$. Collocation methods also use an iterative procedure for fitting the additional parameters involved in the differential equation. Once the solution and its derivatives have been approximated using the set of basis functions, minimization of an error criteria is used to estimate the parameters of the differential equation. Principal differential analysis (PDA) [47] is one example of a collocation method in which the basis functions are *splines*.

An example of a collocation method augmented with Gaussian process priors was introduced by Graepel in [48]. Graepel starts with noisy observations, $\widehat{y}(t)$, of the differential equation $\mathcal{D}_0^M y(t)$ such that $\widehat{y}(t) \sim \mathcal{N}(\mathcal{D}_0^M y(t), \sigma_y^2)$. The solution of the differential equation $\mathcal{D}_0^M y(t)$ is assumed to follow a Gaussian process prior with covariance $k_{\mathcal{D}_{0,t}^M, \mathcal{D}_{0,t'}^M}(t,t)$, where this covariance is obtained by taking $\mathcal{D}_0^M$ derivatives of $k(t,t')$ with respect to t and $\mathcal{D}_0^M$ derivatives with respect to t'. The covariance $k(t,t')$ is freely chosen. An approximated solution $\widetilde{y}(t)$ can then be computed through the expansion $\widetilde{y}(t) = \sum_{n=1}^{N} \alpha_n k_{\mathcal{D}_{0,t}^M, \mathcal{D}_{0,t'}^M}(t, t_n)$, where α_n is an element of the vector

$$\left(\mathbf{K}_{\mathcal{D}_{0,t}^M, \mathcal{D}_{0,t'}^M} + \sigma_y^2 \mathbf{I}_N\right)^{-1} \widehat{\mathbf{y}},$$

where $\mathbf{K}_{\mathcal{D}_{0,t}^M,\mathcal{D}_{0,t'}^M}$ is a matrix with entries $k_{\mathcal{D}_{0,t}^M,\mathcal{D}_{0,t'}^M}(t_n,t_{n'})$, and $\widehat{\mathbf{y}}$ is a vector of noisy observations of $\mathcal{D}_0^M y(t)$. An important difference between this method and the latent force model is that we do not assume we have access to noisy observations for $\mathcal{D}_0^M y(t)$, but noisy observations for the outputs. The LFM is also typically intended for multiple outputs.

Gaussian processes and differential equations have also been used simultaneously in hydrogeology [49], [50], [51], [52], particularly for cokriging using flow equations [53]. The relationship between transmissivity, $T(\mathbf{x})$, and piezometric head, $\phi(\mathbf{x})$, in an aquifer or reservoir is modeled by a nonlinear partial differential equation derived from the conservation of mass and Darcy's law. In a practical setting, there are plenty of measurements for piezoelectric head, but only few measurements for transmissivity [52, see chapter 8]. Cokriging [29] can be used to estimate the amount of transmissivity using piezoelectric head as an auxiliary variable. Using cokriging, though, requires the covariances for $T(\mathbf{x})$ and $\phi(\mathbf{x})$ and the cross covariance between them. An alternative for computing these covariances consists of employing a linear version for the partial differential equation obtained through a small perturbation approximation for $T(\mathbf{x})$ and $\phi(\mathbf{x})$. For details, the reader is referred to [51, see chapter 9] and [53, pp. 637-643]. It turns out that, given a covariance for $\phi(\mathbf{x})$, the covariance for $T(\mathbf{x})$ and the cross covariance for $T(\mathbf{x})$ and $\phi(\mathbf{x})$ can be computed analytically in a similar way to (8) and (10), where the Green's function is obtained from the linear approximation for the partial differential equation. A key difference with latent force models is that, usually, we do not have access to data for the latent forces, in contrast to the method described above in which data for $T(\mathbf{x})$ and $\phi(\mathbf{x})$ is usually at hand.

6.3 Gaussian Processes for Systems Identification

In control engineering, systems identification refers to a set of techniques used for representing a dynamical system by a mathematical model (mostly a linear model). A detailed description of the dynamical system is usually unknown, and parameters of the surrogate model are estimated from measured data.

Gaussian processes have been used as models for systems identification [54], [55], [56], [57]. In [54], a nonlinear dynamical system is linearized around an equilibrium point by means of a Taylor series expansion [57], $y(t) = \sum_{j=0}^{\infty} \frac{y^{(j)}(a)}{j!}(t-a)^j$, with a the equilibrium point. For a finite value of terms, the linearization above can be seen as a regression problem in which the covariates correspond to the terms $(t-a)^j$ and the derivatives $y^{(j)}(a)$ as regression coefficients. The derivatives are assumed to follow a Gaussian process prior with a covariance function that is obtained as $k^{(j,j')}(t,t')$, where the superscript j indicates how many derivatives of $k(t,t')$ are taken with respect to t and the superscript j' indicates how many derivatives of $k(t,t')$ are taken with respect to t'. Derivatives are then estimated a posteriori through standard Bayesian linear regression.

Gaussian processes have also been used to model the output $y(t)$ at time t_k as a function of its L previous samples $\{y(t-t_{k-l})\}_{l=1}^L$, a common setup in the classical theory of systems identification [58]. The particular dependency $y(t) = g(\{y(t-t_{k-l})\}_{l=1}^L)$, where $g(\cdot)$ is a general nonlinear function, is modeled using a Gaussian process prior and the predicted value for the output $y_*(t_k)$ is used as a new input for multistep ahead prediction at times t_j, with $j > k$ [55]. Uncertainty about $y_*(t_k)$ can also be incorporated for predictions of future output values [59].

It is worth mentioning that there has been recent interest in introducing Gaussian processes in the state space formulation of dynamical systems [60], [61], [62] for the representation of the possible nonlinear relationships between the latent space and between the latent space and the observation space.

An important difference of all the above methods for systems identification and the latent force model framework is that we are interested in describing multidimensional outputs. Furthermore, we are interested in constructing powerful covariance functions that can be used within a Gaussian process. The approaches described above are all black-box methods.

Since the original submission of this paper, work by Hartikainen and Särkkä [63], [64] has considered latent force models from a state space modeling perspective, leading to significant improvements in computational complexity for temporal datasets.

7 Conclusion

In this paper, we have presented the latent force model: a hybrid approach to modeling that sits between a fully mechanistic and a data-driven approach. We used Gaussian process priors and linear differential equations to model interactions between different variables. The result is the formulation of a probabilistic model, based on a kernel function, that encodes the coupled behavior of several dynamical systems and allows for more accurate predictions. Linear latent force models explored in this paper can be extended in several ways, including:

Nonlinear Latent Force Models. If the likelihood function is not Gaussian or the differential equation is nonlinear, the inference process is not generally analytic and approximations must be used such as Laplace's approximation [1] or sampling [65].

Cascaded Latent Force Models. Latent forces $u_q(t)$ could be the outputs of another latent force model. For example, in Honkela et al. [66], the authors use a cascaded system to describe gene expression data for which a first order linear system has inputs $u_q(t)$ governed by Gaussian processes with covariance function (14).

Switching Dynamical Latent Force Models. A further extension of the LFM framework allows the parameter vector $\boldsymbol{\theta}$ to have discrete changes as function of the input time. In [67], this model was used for the segmentation of movements performed by a Barrett WAM robot as haptic input device.

Acknowledgments

David Luengo was partly financed by Comunidad de Madrid (project PRO-MULTIDIS-CM, S-0505/TIC/0233),

and by the Spanish government (research grant JC2008-00219, CICYT project TEC2009-14504-C02-01 and CONSOLIDER-INGENIO Project CSD2008-00010). Mauricio A. Álvarez and Neil D. Lawrence were financed by a Google Research Award and EPSRC Grant No EP/F005687/1 "Gaussian Processes for Systems Identification with Applications in Systems Biology." Neil D. Lawrence was also funded by the EU FP7 BioPreDyn award no 289434. Mauricio A. Álvarez also acknowledges support from the Overseas Research Student Award Scheme (ORSAS), from the School of Computer Science of the University of Manchester, and from the Universidad Tecnológica de Pereira, Colombia. The main body of this work was completed while Mauricio A. Álvarez was still a PhD student at The University of Manchester. The authors also thank the reviewers for their helpful comments.

REFERENCES

[1] N.D. Lawrence, G. Sanguinetti, and M. Rattray, "Modelling Transcriptional Regulation Using Gaussian Processes," *Proc. Advances in Neural Information Processing Systems*, B. Schölkopf, J. C. Platt, and T. Hofmann, eds., vol. 19, pp. 785-792, 2007.

[2] P. Gao, A. Honkela, M. Rattray, and N.D. Lawrence, "Gaussian Process Modelling of Latent Chemical Species: Applications to Inferring Transcription Factor Activities," *Bioinformatics*, vol. 24, pp. i70-i75, 2008.

[3] N.D. Lawrence, "Probabilistic Nonlinear Principal Component Analysis with Gaussian Process Latent Variable Models," *J. Machine Learning Research*, vol. 6, pp. 1783-1816, Nov. 2005.

[4] Y.W. Teh, M. Seeger, and M.I. Jordan, "Semiparametric Latent Factor Models," *Proc. Workshop Artificial Intelligence and Statistics*, R.G. Cowell and Z. Ghahramani, eds., pp. 333-340, Jan. 2005.

[5] E.V. Bonilla, K.M. Chai, and C.K.I. Williams, "Multi-Task Gaussian Process Prediction," *Proc. Neural Information Processing Systems*, J.C. Platt, D. Koller, Y. Singer, and S. Roweis, eds., vol. 20, 2008.

[6] M.A. Osborne, A. Rogers, S.D. Ramchurn, S.J. Roberts, and N.R. Jennings, "Towards Real-Time Information Processing of Sensor Network Data Using Computationally Efficient Multi-Output Gaussian Processes," *Proc. Int'l Conf. Information Processing in Sensor Networks*, 2008.

[7] J. Quiñonero-Candela and C.E. Rasmussen, "A Unifying View of Sparse Approximate Gaussian Process Regression," *J. Machine Learning Research*, vol. 6, pp. 1939-1959, 2005.

[8] D.H Griffel, *Applied Functional Analysis*, reprinted ed. Dover Publications Inc., 2002.

[9] G.F Roach, *Green's Functions*, second ed. Cambridge Univ. Press, 1982.

[10] M.A. Álvarez, D. Luengo, and N.D Lawrence, "Latent Force Models," *Proc. 12th Int'l Conf. Artificial Intelligence and Statistics*, D. van Dyk and M. Welling, eds., pp. 9-16, Apr. 2009.

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

[12] M.A. Álvarez, D. Luengo, M.K Titsias, and N.D Lawrence, "Efficient Multioutput Gaussian Processes through Variational Inducing Kernels," *Proc. 13th Int'l Conf. Artificial Intelligence and Statistics*, Y.W. Teh and M. Titterington, eds., pp. 25-32, May 2010.

[13] C.M Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[14] L. Csató and M. Opper, "Sparse Representation for Gaussian Process Models," *Proc. Advances in Neural Information Processing Systems*, T.K Leen T.G Dietterich and V. Tresp, eds., vol. 13, pp. 444-450, 2001.

[15] M. Seeger, C.K.I Williams, and N.D Lawrence, "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression," *Proc. Ninth Int'l Workshop Artificial Intelligence and Statistics*, C.M. Bishop and B.J. Frey, eds., Jan. 2003.

[16] E. Snelson and Z. Ghahramani, "Sparse Gaussian Processes Using Pseudo-Inputs," *Proc. Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J.C Platt, eds., vol. 18, 2006.

[17] M.K. Titsias, "Variational Learning of Inducing Variables in Sparse Gaussian Processes," *Proc. 12th Int'l Conf. Artificial Intelligence and Statistics*, D. van Dyk and M. Welling, eds., vol. 5, pp. 567-574, Apr. 2009.

[18] M.A. Álvarez and N.D Lawrence, "Sparse Convolved Gaussian Processes for Multi-Output Regression," *Proc. Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds., vol. 21, pp. 57-64, 2009.

[19] D. Kosman, J. Reinitz, and D.H Sharp, "Automated Assay of Gene Expression at Cellular Resolution," *Proc. Pacific Symp. Biocomputing*, R. Altman, K. Dunker, L. Hunter, and T. Klein, eds., pp. 6-17, 1999.

[20] D. Kosman, S. Small, and J. Reinitz, "Rapid Preparation of a Panel of Polyclonal Antibodies to Drosophila Segmentation Proteins," *Development Genes Evolution*, vol. 208, pp. 290-294, 1998.

[21] S. Surkova et al., "Characterization of the Drosophila Segment Determination Morphome," *Developmental Biology*, vol. 313, no. 2, pp. 844-862, 2008.

[22] T.J. Perkins, J. Jaeger, J. Reinitz, and L. Glass, "Reverse Engineering the Gap Gene Network of Drosophila Melanogaster," *PLOS Computational Biology*, vol. 2, no. 5, pp. 417-427, 2006.

[23] V.V. Gursky, J. Jaeger, K.N. Kozlov, J. Reinitz, and A. Samsonov, "Pattern Formation and Nuclear Divisions Are Uncoupled in Drosophila Segmentation: Comparison of Spatially Discrete and Continuous Models," *Physica D*, vol. 197, pp. 286-302, 2004.

[24] A.D Polyanin, *Handbook of Linear Partial Differential Equations for Engineers and Scientists*. Chapman & Hall/CRC Press, 2002.

[25] A. Butkovskiy and L. Pustyl'nikov, *Characteristics of Distributed-Parameter Systems*. Kluwer Academic Publishers, 1993.

[26] I. Stakgold, *Green's Functions and Boundary Value Problems*, second ed. John Wiley & Sons, Inc., 1998.

[27] S. Conti and A. O'Hagan, "Bayesian Emulation of Complex Multi-Output and Dynamic Computer Models," *J. Statistical Planning and Inference*, vol. 140, no. 3, pp. 640-651, 2010.

[28] J. Rougier, "Efficient Emulators for Multivariate Deterministic Functions," *J. Computational and Graphical Statistics*, vol. 17, no. 4, pp. 827-834, 2008.

[29] P. Goovaerts, *Geostatistics for Natural Resources Evaluation*. Oxford Univ. Press, 1997.

[30] D.M Higdon, "Space and Space-Time Modelling Using Process Convolutions," *Quantitative Methods for Current Environmental Issues*, C. Anderson, V. Barnett, P. Chatwin, and A. El-Shaarawi, eds., pp. 37-56, Springer-Verlag, 2002.

[31] P. Boyle and M. Frean, "Dependent Gaussian Processes," *Proc. Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bouttou, eds., vol. 17, pp. 217-224, 2005.

[32] A.G. Journel and C.J Huijbregts, *Mining Geostatistics*. Academic Press, 1978.

[33] N.A.C. Cressie, *Statistics for Spatial Data*, revised ed. John Wiley & Sons, 1993.

[34] H. Wackernagel, *Multivariate Geostatistics*. Springer-Verlag, 2003.

[35] M.A. Álvarez, L. Rosasco, and N.D. Lawrence, "Kernels for Vector-Valued Functions: A Review," *Foundations and Trends in Machine Learning*, vol. 4, no. 3, pp. 195-266, 2012.

[36] R.P. Barry and J.M. Ver Hoef, "Blackbox Kriging: Spatial Prediction without Specifying Variogram Models," *J. Agricultural, Biological, and Environmental Statistics*, vol. 1, no. 3, pp. 297-322, 1996.

[37] J.M. Ver Hoef and R.P. Barry, "Constructing and Fitting Models for Cokriging and Multivariable Spatial Prediction," *J. Statistical Planning and Inference*, vol. 69, pp. 275-294, 1998.

[38] C.A. Calder and N. Cressie, "Some Topics in Convolution-Based Spatial Modeling," *Proc. 56th Session Int'l Statistics Inst.*, Aug. 2007.

[39] C.J. Paciorek and M.J Schervish, "Nonstationary Covariance Functions for Gaussian Process Regression," *Proc. Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, eds., 2004.

[40] P. Boyle, "Gaussian Processes for Regression and Optimisation," PhD dissertation, Victoria Univ. of Wellington, Wellington, New Zealand, 2007.

[41] R. Murray-Smith and B.A Pearlmutter, "Transformation of Gaussian Process Priors," *Proc. First Int'l Conf. Deterministic and Statistical Methods in Machine Learning*, J. Winkler, M. Niranjan, and N. Lawrence, eds., pp. 110-123, 2005.

[42] J.Q Shi, R. Murray Smith, D. Titterington, and B. Pearlmutter, "Learning with Large Data Sets Using Filtered Gaussian Process Priors," *Proc. Hamilton Summer School on Switching and Learning in Feedback systems*, R. Murray-Smith and R. Shorten, eds., pp. 128-139, 2005.

[43] J.O. Ramsay and B.W Silverman, *Functional Data Analysis*, second ed. Springer, 2005.

[44] A. Poyton, M.S. Varziri, K.B. McAuley, J. McLellan, and J.O. Ramsay, "Parameter Estimation in Continuous-Time Dynamic Models Using Principal Differential Analysis," *Computers and Chemical Eng.*, vol. 30, pp. 698-708, 2006.

[45] J.O. Ramsay, G. Hooker, D. Campbell, and J. Cao, "Parameter Estimation for Differential Equations: A Generalized Smoothing Approach," *J. Royal Statistical Soc. Series B (Methodological)*, vol. 69, no. 5, pp. 741-796, 2007.

[46] D. Brewer, M. Barenco, R. Callard, M. Hubank, and J. Stark, "Fitting Ordinary Differential Equations to Short Time Course Data," *Philosophical Trans. Royal Soc. A*, vol. 366, pp. 519-544, 2008.

[47] J.O. Ramsay, "Principal Differential Analysis: Data Reduction by Differential Operators," *J. Royal Statistical Soc. Series B (Methodological)*, vol. 58, no. 3, pp. 495-508, 1996.

[48] T. Graepel, "Solving Noisy Linear Operator Equations by Gaussian Processes: Application to Ordinary and Partial Differential Equations," *Proc. 20th Int'l Conf. Machine Learning*, T. Fawcett and N. Mishra, eds., pp. 234-241, Aug. 2003.

[49] P.K. Kitanidis and E.G. Vomvoris, "A Geostatistical Approach to the Inverse Problem in Groundwater Modeling (Steady State) and One-Dimensional Simulations," *Water Resources Research*, vol. 19, no. 3, pp. 677-690, 1983.

[50] T.A. Task Committee on Geostatistical Techniques in Geohydrology of the Ground Water Hydrology Committee of the ASCE Hydraulics Division, "Review of Geostatistics in Geohydrology. I: Basic Concepts," *J. Hydraulic Eng.*, vol. 116, no. 5, pp. 612-632, 1990.

[51] P.K Kitanidis, *Introduction to Geostatistics: Applications in Hydrogeology*. Cambridge Univ. Press, 1997.

[52] J.P. Chilès and P. Delfiner, *Geostatistics: Modeling Spatial Uncertainty*. Wiley, 1999.

[53] T.A. Task Committee on Geostatistical Techniques in Geohydrology of the Ground Water Hydrology Committee of the ASCE Hydraulics Division, "Review of Geostatistics in Geohydrology. II: Applications," *J. Hydraulic Eng.*, vol. 116, no. 5, pp. 633-658, 1990.

[54] E. Solak, R. Murray-Smith, W.E Leithead, D.J Leith, and C.E Rasmussen, "Derivative Observations in Gaussian Process Models of Dynamic Systems," *Proc. Conf. Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, eds., vol. 15, pp. 1033-1040, 2003.

[55] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith, "Dynamic Systems Identification with Gaussian Processes," *Math. and Computer Modelling of Dynamical Systems*, vol. 11, no. 4, pp. 411-424, 2005.

[56] B. Calderhead, M. Girolami, and N.D Lawrence, "Accelerating Bayesian Inference over Nonlinear Differential Equations with Gaussian Processes," *Proc. Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds., vol. 21, pp. 217-224, 2009.

[57] K.R Thompson, "Implementation of Gaussian Process Models for Nonlinear System Identification," PhD dissertation, Dept. of Electronics and Electrical Eng., Univ. of Glasgow, United Kingdom, 2009.

[58] L. Ljung, *System Identification: Theory for the User*, second ed. Prentice Hall PTR, 1999.

[59] A. Girard, C.E Rasmussen, J.Q Candela, and R. Murray Smith, "Gaussian Process Priors with Uncertain Inputs—Application to Multiple-Step Ahead Time Series Forecasting," *Proc. Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, eds., vol. 15, pp. 529-536, 2003.

[60] J. Ko, D.J Klein, D. Fox, and D. Haehnel, "GP-UKF: Unscented Kalman Filters with Gaussian Process Prediction and Observation Models," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, pp. 1901-1907, 2007.

[61] M.P Deisenroth, M.F Huber, and U.D Hanebeck, "Analytic Moment-Based Gaussian Process Filtering," *Proc. 26th Int'l Conf. Machine Learning*, pp. 225-232, 2009.

[62] R. Turner, M.P Deisenroth, and C.E Rasmussen, "State-Space Inference and Learning with Gaussian Processes," *Proc. 13th Int'l Conf. Artificial Intelligence and Statistics*, Y.W. Teh and M. Titterington, eds., pp. 868-875, May 2010.

[63] J. Hartikainen and S. Särkkä, "Sequential Inference for Latent Force Models," *Proc. 27th Conf. Uncertainty in Artificial Intelligence*, pp. 311-318, 2011.

[64] J. Hartikainen, M. Seppänen, and S. Särkkä, "State-Space Inference for Nonlinear Latent Force Models with Application to Satellite Orbit Prediction," *Proc. 29th Int'l Conf. Machine Learning*, 2012.

[65] M. Titsias, N.D Lawrence, and M. Rattray, "Efficient Sampling for Gaussian Process Inference Using Control Variables," *Proc. Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds., vol. 21, pp. 1681-1688, 2009.

[66] A. Honkela, C. Girardot, E.H. Gustafson, Y.-H. Liu, E.E.M. Furlong, N.D. Lawrence, and M. Rattray, "Model-Based Method for Transcription Factor Target Identification with Limited Data," *Proc. Nat'l Academy of Sciences USA*, vol. 107, no. 17, pp. 7793-7798, 2010.

[67] M.A. Álvarez, J. Peters, B. Schölkopf, and N.D Lawrence, "Switched Latent Force Models for Movement Segmentation," *Proc. Conf. Neural Information Processing Systems*, vol. 24, pp. 55-63, 2011.

[68] *Proc. Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds., vol. 21, 2009.

[69] *Proc. Artificial Intelligence and Statistics*, D. van Dyk and M. Welling, eds., Apr. 2009.

[70] *Proc. Artificial Intelligence and Statistics*, Y.W. Teh and M. Titterington, eds., May 2010.

[71] *Proc. Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, eds., vol. 15, 2003.