
Article:
Alvarez, M.A. orcid.org/0000-0002-8980-4472, Luengo, D. and Lawrence, N.D. orcid.org/0000-0001-9258-1030 (2013) Linear Latent Force Models Using Gaussian Processes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (11). pp. 2693-2705. ISSN 0162-8828

https://doi.org/10.1109/TPAMI.2013.86

Linear Latent Force Models using Gaussian Processes

Mauricio A. Álvarez[†,‡], David Luengo[♯], Neil D. Lawrence[⋆,○]

† *School of Computer Science, University of Manchester, Manchester, UK M13 9PL.*

‡ *Faculty of Engineering, Universidad Tecnológica de Pereira, Colombia, 660003.*

♯ *Dep. de la Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, 28911 Leganés, España.*

⋆ *School of Computer Science, University of Sheffield, Sheffield, UK S1 4DP.*

○ *The Sheffield Institute for Translational Neuroscience, Sheffield, UK S10 2HQ.*

Abstract

Purely data driven approaches for machine learning present difficulties when data is 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 modelling 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*. The usual approach is to combine a data set 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. Firstly, if data is scarce relative to the complexity of the system we may be unable to make accurate predictions on test data. Secondly, if the model is forced to extrapolate, *i.e.* make predictions in a regime in which data has not yet been seen, performance can be poor.

Purely *mechanistic models*, *i.e.* models which are inspired by the underlying physical knowledge of the system, are common in many domains such as chemistry, systems biology, climate modelling and geophysical sciences, *etc.* 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 modelling 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 prediction even in regions where there is no available training data. For example, Pioneer space probes can enter different extra terrestrial orbits regardless of the availability of data for these orbits.

Whilst data driven approaches do seem to avoid mechanistic assumptions about the data, typically the regularization which is applied encodes some kind of physical intuition, such as the smoothness of the interpolant. This does reflect 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* which involves a (typically overly simplistic) mechanistic model of the system. 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 in practise. To illustrate the framework we will start by considering dynamical systems as latent variable models which incorporate ordinary differential equations. In this we follow the work of Lawrence et al. (2007) and Gao et al. (2008) who encoded a first order differential equation in a Gaussian process (GP). However, 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. For example, for a human motion capture dataset we develop a mechanistic

model of motion capture that does not exactly replicate the *physics* of human movement, but nevertheless captures salient features of the movement. Having shown how linear dynamical systems can be incorporated in a GP, we finally show how partial differential equations can also be incorporated for modelling systems with multiple inputs.

The paper is organized as follows. In section 2 we motivate the latent force model using as an example a latent variable model. Section 3 employs a first order latent force model to describe how the general framework can be used in practise. We then proceed to show three case studies. In section 4 we use a latent force model based on a second order ordinary differential equation for characterizing motion capture datasets. Section 5 presents a latent force model for spatio-temporal domains applied to represent the development of Drosophila Melanogaster, and a latent force model inspired in a diffusion process to explain 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 From latent variables to latent functions

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 data set with a reduced dimensional representation. For example, if our data consists of N points in a D dimensional space we might seek a linear relationship between the data, $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_D] \in \mathbb{R}^{N\times D}$ with $\mathbf{y}_d \in \mathbb{R}^{N\times 1}$, and a reduced dimensional representation, $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_Q] \in \mathbb{R}^{N\times Q}$ 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 $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_D]$ is a matrix-variate Gaussian noise: each column, $\mathbf{e}_d \in \mathbb{R}^{N\times 1}$ ($1 \leq d \leq D$), is a multi-variate Gaussian with zero mean and covariance $\mathbf{\Sigma}$, this is, $\mathbf{e}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_d)$. 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 non-linear variant of the model it can be convenient to do this the other way around, see for example Lawrence (2005)). If the data has a temporal nature, then the Gaussian prior in 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 n-th 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 $p(\mathbf{U}|\mathbf{t})$, for a vector of time inputs $\mathbf{t} = [t_1 \dots t_N]^\top$, which has the form of a Gaussian process,

$$p(\mathbf{U}|\mathbf{t}) = \prod_{q=1}^{Q} \mathcal{N}\left(\mathbf{u}_q|\mathbf{0}, \mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}\right), \tag{2}$$

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 $\mathbf{u}_q = [u_q(t_1), \dots, u_q(t_N)]^\top$, represents the values of the function for the q-th dimension at the times given by $\mathbf{t}$. The matrix $\mathbf{K}_{\mathbf{u}_q,\mathbf{u}_q}$ is the covariance function associated to $u_q(t)$ 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 Teh et al. (2005) this is known as a semi-parametric 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 multi-task Gaussian process prediction model (MTGP) (Bonilla et al., 2008) with a similar model introduced in Osborne et al. (2008). 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 Quiñonero-Candela and Rasmussen (2005) 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, $\mathbf{Y}\mathbf{B} = \mathbf{U}\mathbf{S}^\top + \widetilde{\mathbf{E}}$. 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 $\tilde{\mathbf{e}}_d \sim \mathcal{N}\left(\mathbf{0}, \mathbf{B}^\top\mathbf{\Sigma}_{\boldsymbol{d}}\mathbf{B}\right)$. With appropriate choice of latent density and noise

model this physical model underlies the Kalman filter, PCA, independent component analysis and the multioutput Gaussian process models we mentioned above. The use of latent variables means that despite this strong physical constraint these models are still powerful enough to be applied to a range of real world data sets. We will retain this flexibility by maintaining the latent variables at the heart of the system, but introduce a more realistic system by extending the underlying physical model. Let us assume that the springs are acting in parallel with dampers and that the system has mass, allowing us to write,

$$\ddot{\mathbf{Y}}\mathbf{M} + \dot{\mathbf{Y}}\mathbf{C} + \mathbf{Y}\mathbf{B} = \mathbf{U}\mathbf{S} + \widehat{\mathbf{E}}, \tag{3}$$

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 $d = 1, \dots D$ and $n = 1, \dots, N$), $\ddot{\mathbf{Y}}$ is the second derivative of $\mathbf{Y}$ with respect to time (with entries $\{\ddot{y}_d(t_n)\}$ for $d = 1, \dots D$ and $n = 1, \dots, N$) and $\widehat{\mathbf{E}}$ is once again a matrix-variate Gaussian noise. Equation (3) 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 several characteristics which are impossible to represent in the simpler latent variable model given by (1), such as inertia and resonance. This model is not only appropriate for data from mechanical systems. There are many analogous systems which 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 them as *latent force models* (LFMs). This is our general framework: combine a physical system with a probabilistic prior over some latent variable.

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. Human motion is a high dimensional data set. A skilled puppeteer with a well designed puppet can create a realistic representation of human movement through judicious use of the strings

3 Latent Force Models in Practise

In the last section we provided a general description of the latent force model idea and commented how it compares to previous models in the machine learning and statistics literature. In this section we specify the operational procedure to obtain the Gaussian process model associated to the outputs and different aspects involved in the inference process. First, we illustrate the procedure using a first-order latent force model, for which we assume there are no masses associated to the outputs and the damper constants are equal to one. Then we specify the inference procedure, which involves maximization of the marginal likelihood for estimating hyperparameters. Next we generalize the operational procedure for latent force models of higher order and multidimensional inputs and finally we review some efficient approximations to reduce computational complexity.

3.1 First-order Latent Force Model

Assume a simplified latent force model, for which only the first derivative of the outputs is included. This is a particular case of equation (3), with masses equal to zero and damper constants equal to one. With these assumptions, equation (3) can be written as

$$\dot{\mathbf{Y}} + \mathbf{Y}\mathbf{B} = \mathbf{U}\mathbf{S} + \widehat{\mathbf{E}}. \tag{4}$$

Individual elements in equation (4) follow

$$\frac{\mathrm{d}y_d(t)}{\mathrm{d}t} + B_d y_d(t) = \sum_{q=1}^{Q} S_{d,q} u_q(t) + \hat{e}_d(t). \tag{5}$$

Given the parameters $\{B_d\}_{d=1}^D$ and $\{S_{d,q}\}_{d=1,q=1}^{D,Q}$, the uncertainty in the outputs is given by the uncertainty coming from the set of functions $\{u_q(t)\}_{q=1}^Q$ and the noise $\hat{e}_d(t)$. Strictly speaking, this equation belongs to a more general set of equations known as *stochastic differential equations* (SDE) that are usually solved using special techniques from stochastic calculus (Øksendal, 2003). The representation used in equation (5) is more common in physics, where it receives the name of *Langevin equations* (Reichl, 1998). For the simpler equation (5), the solution is found using standard calculus techniques and is given by

$$y_d(t) = y_d(t_0)e^{-B_d t} + \sum_{q=1}^{Q} S_{d,q}\mathcal{G}_d[u_q](t) + \mathcal{G}_d[\hat{e}_d](t), \tag{6}$$

where $y_d(t_0)$ correspond to the value of $y_d(t)$ for $t = t_0$ (or the initial condition) and $\mathcal{G}_d$ is a linear integral operator that follows

$$\mathcal{G}_d[v](t) = f_d(t, v(t)) = \int_0^t e^{-B_d(t-\tau)} v(\tau) \mathrm{d}\tau.$$

Our noise model $\mathcal{G}_d[\hat{e}_d](t)$ has a particular form depending on the linear operator $\mathcal{G}_d$. For example, for the equation in (6) and assuming a white noise process prior for $e_d(t)$, it can be shown that the process $\mathcal{G}_d[\hat{e}_d](t)$ corresponds to the Ornstein-Uhlenbeck (OU) process (Reichl, 1998). In what follows, we will allow the noise model to be a more general process and we denote it by $w_d(t)$. Without loss of generality, we also assume that the initial conditions $\{y_d(t_0)\}_{d=1}^D$ are zero, so that we can write again equation (6) as

$$y_d(t) = \sum_{q=1}^{Q} S_{d,q} \mathcal{G}_d[u_q](t) + w_d(t). \tag{7}$$

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

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

where we use $f_d^q(t)$ as a shorthand for $f_d(t, u_q(t))$. Furthermore, for the latent force model in equation (4), the covariance $\mathrm{cov}[f_d^q(t), f_{d'}^q(t')]$ is equal to

$$\int_0^t e^{-B_d(t-\tau)} \int_0^{t'} e^{-B_{d'}(t'-\tau')} k_{u_q,u_q}(\tau,\tau') \mathrm{d}\tau' \mathrm{d}\tau. \tag{8}$$

Notice from the equation above that the covariance between $f_d^q(t)$ and $f_{d'}^q(t')$ depends on the covariance $k_{u_q,u_q}(\tau,\tau')$. We alternatively denote $\mathrm{cov}[f_d(t), f_{d'}(t')]$ as $k_{f_d,f_{d'}}(t,t')$ and $\mathrm{cov}[f_d^q(t), f_{d'}^q(t')]$ as $k_{f_d^q,f_{d'}^q}(t,t')$. The form for the covariance $k_{u_q,u_q}(t,t')$ is such that we can solve both integrals in equation (8) and find an analytical expression for the covariance $k_{f_d,f_{d'}}(t,t')$. In the rest of the paper, we assume the covariance for each latent force $u_q(t)$ follows the squared-exponential (SE) form (Rasmussen and Williams, 2006)

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

where ℓ_q is known as the length-scale. We can compute the covariance $k_{f_d^q,f_{d'}^q}(t,t')$ obtaining (Lawrence et al., 2007)

$$k_{f_d^q,f_{d'}^q}(t,t') = \frac{\sqrt{\pi}\ell_q}{2}[h_{d',d}(t',t) + h_{d,d'}(t,t')], \tag{10}$$

where

$$h_{d',d}(t',t) = \frac{\exp(\nu_{q,d'}^2)}{B_d + B_{d'}} \exp(-B_{d'}t') \Bigg\{ \exp(B_{d'}t) \left[\mathrm{erf}\left(\frac{t'-t}{\ell_q} - \nu_{q,d'}\right) + \mathrm{erf}\left(\frac{t}{\ell_q} + \nu_{q,d'}\right)\right] - \exp(-B_d t) \left[\mathrm{erf}\left(\frac{t'}{\ell_q} - \nu_{q,d'}\right) + \mathrm{erf}(\nu_{q,d'})\right] \Bigg\}, \tag{11}$$

where $\mathrm{erf}(x)$ is the real valued error function, $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-y^2)\mathrm{d}y$, and $\nu_{q,d} = \ell_q B_d/2$. The covariance function in equation (10) is nonstationary. For the stationary regime, the covariance function can be obtained by writing $t' = t + \tau$ and taking the limit as t tends to infinity. This is, $k_{f_d^q,f_{d'}^q}^{\mathrm{STAT}}(\tau) = \lim_{t\to\infty} k_{f_d^q,f_{d'}^q}(t, t+\tau)$. The stationary covariance could

[1] We can allow a mean prior different from zero.

also be obtained making use of the power spectral density for the stationary processes $u_q(t)$, $U_q(\omega)$ and the transfer function $H_d(\omega)$ associated to $h_d(t-s) = e^{-B_d(t-s)}$, the impulse response of the first order dynamical system. Then applying the convolution property of the Fourier transform to obtain the power spectral density of $f_d^q(t)$, $F_d^q(\omega)$, and finally using the *Wiener-Khinchin theorem* to find the solution for $f_d^q(t)$ (Shanmugan and Breipohl, 1988).
As we will see in the following section, for computing the posterior distribution for $\{u_q(t)\}_{q=1}^Q$, we need the cross-covariance between the output $y_d(t)$ and the latent force $u_q(t)$. Due to the independence between $u_q(t)$ and $w_d(t)$, the covariance reduces to $k_{f_d,u_q}(t,t')$, given by

$$k_{f_d,u_q}(t,t') = \frac{\sqrt{\pi}\ell_q S_{d,q}}{2}\exp(\nu_{q,d}^2)\exp(-B_d(t-t'))\left[\operatorname{erf}\left(\frac{t-t'}{\ell_q}-\nu_{q,d}\right)+\operatorname{erf}\left(\frac{t'}{\ell_q}+\nu_{q,d}\right)\right]. \tag{12}$$

3.2 Hyperparameter learning

We have implicitly marginalized out the effect of the latent forces using the Gaussian process prior for $\{u_q(t)\}_{q=1}^Q$ and the covariance for the outputs after marginalization is given by $k_{y_d,y_{d'}}(t,t')$. Given a set of inputs $\mathbf{t}$ and the parameters of the covariance function,[2] $\boldsymbol{\theta} = (\{B_d\}_{d=1}^D, \{S_{d,q}\}_{d=1,q=1}^{D,Q}, \{\ell_q\}_{q=1}^Q)$, the marginal likelihood for the outputs can be written as

$$p(\mathbf{y}|\mathbf{t},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma}), \tag{13}$$

where $\mathbf{y} = \operatorname{vec}\mathbf{Y}$,[3] $\mathbf{K}_{\mathbf{f},\mathbf{f}} \in \mathbb{R}^{ND\times ND}$ with each element given by $\operatorname{cov}[f_d(t_n), f_{d'}(t'_{n'})]$ for $n = 1,\dots,N$ and $n' = 1,\dots,N$ and $\boldsymbol{\Sigma}$ represents the covariance associated with the independent processes $w_d(t)$. In general, the vector of parameters $\boldsymbol{\theta}$ is unknown, so we estimate it by maximizing the marginal likelihood.
For clarity, we assumed that all outputs are evaluated at the same set of inputs $\mathbf{t}$. However, due to the flexibility provided by the Gaussian process formulation, each output can have associated a specific set of inputs, this is $\mathbf{t}_d = [t_1^d,\dots,t_{N_d}^d]$.
Prediction for a set of input test $\mathbf{t}_*$ is done using standard Gaussian process regression techniques. The predictive distribution is given by

$$p(\mathbf{y}_*|\mathbf{y},\mathbf{t},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}_*|\boldsymbol{\mu}_*, \mathbf{K}_{\mathbf{y}_*,\mathbf{y}_*}), \tag{14}$$

with

$$\begin{aligned}\boldsymbol{\mu}_* &= \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}\left(\mathbf{K}_{\mathbf{f},\mathbf{f}}+\boldsymbol{\Sigma}\right)^{-1}\mathbf{y},\\ \mathbf{K}_{\mathbf{y}_*,\mathbf{y}_*} &= \mathbf{K}_{\mathbf{f}_*,\mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*,\mathbf{f}}\left(\mathbf{K}_{\mathbf{f},\mathbf{f}}+\boldsymbol{\Sigma}\right)^{-1}\mathbf{K}_{\mathbf{f}_*,\mathbf{f}}^\top + \boldsymbol{\Sigma}_*,\end{aligned}$$

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 $\mathbf{t}_*$. 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,

$$p(\mathbf{u}|\mathbf{y},\mathbf{t},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{u}|\boldsymbol{\mu}_{\mathbf{u}|\mathbf{y}}, \mathbf{K}_{\mathbf{u}|\mathbf{y}}), \tag{15}$$

with

$$\begin{aligned}\boldsymbol{\mu}_{\mathbf{u}|\mathbf{y}} &= \mathbf{K}_{\mathbf{f},\mathbf{u}}^\top\left(\mathbf{K}_{\mathbf{f},\mathbf{f}}+\boldsymbol{\Sigma}\right)^{-1}\mathbf{y},\\ \mathbf{K}_{\mathbf{u}|\mathbf{y}} &= \mathbf{K}_{\mathbf{u},\mathbf{u}} - \mathbf{K}_{\mathbf{f},\mathbf{u}}^\top\left(\mathbf{K}_{\mathbf{f},\mathbf{f}}+\boldsymbol{\Sigma}\right)^{-1}\mathbf{K}_{\mathbf{f},\mathbf{u}},\end{aligned}$$

where $\mathbf{u} = \operatorname{vec}\mathbf{U}$, $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ 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 $k_{u_q,u_q}(t,t')$ in equation (9), for $\{t_n\}_{n=1}^N$. 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 $k_{f_d,u_q}(t,t')$ in equation (12).

3.3 Higher-order Latent Force Models

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

$$\sum_{m=0}^{M}\mathcal{D}^m[\mathbf{Y}]\mathbf{A}_m = \mathbf{U}\mathbf{S}^\top + \widehat{\mathbf{E}}, \tag{16}$$

[2] Also known as hyperparameters.
[3] $\mathbf{x} = \operatorname{vec}\mathbf{X}$ 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.

where $\mathcal{D}^m$ is a linear differential operator such that $\mathcal{D}^m[\mathbf{Y}]$ is a matrix with elements given by $\mathcal{D}^m y_d(t) = \frac{\mathrm{d}^m y_d(t)}{\mathrm{d}^m t}$ and $\mathbf{A}_m$ is a diagonal matrix with elements $A_{m,d}$ that weights the contribution of $\mathcal{D}^m y_d$.
We follow the same procedure described in section 3.1 for the model in equation (16) with $M = 1$. Each element in expression (16) can be written as

$$\mathcal{D}_0^M y_d = \sum_{m=0}^{M} A_{m,d} \mathcal{D}^m y_d(t) = \sum_{q=1}^{Q} S_{d,q} u_q(t) + \hat{e}_d(t), \tag{17}$$

where we have introduced a new operator $\mathcal{D}_0^M$ that is equivalent to apply the weighted sum of operators $\mathcal{D}^m$. For a homogeneous differential equation in (17), this is $u_q(t) = 0$ for $q = 1, \ldots, Q$ and $e_d(t) = 0$, and a particular set of initial conditions $\{\mathcal{D}^m y_d(t_0)\}_{m=0}^{M-1}$, it is possible to find a linear integral operator $\mathcal{G}_d$ associated to $\mathcal{D}_0^M$ that can be used to solve the non-homogeneous differential equation. The linear integral operator is defined as

$$\mathcal{G}_d[v](t) = f_d(t, v(t)) = \int_{\mathcal{T}} G_d(t, \tau) v(\tau) \mathrm{d}\tau, \tag{18}$$

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

$$\mathcal{D}_0^M [G_d(t, s)] = \delta(t - s), \tag{19}$$

with s fixed, $G_d(t, s)$ a fundamental solution that satisfies the initial conditions and $\delta(t-s)$ the Dirac delta[4] function (Griffel, 2002). Strictly speaking, the differential operator in equation (19) is the adjoint for the differential operator appearing in equation (17). For a more rigorous introduction to Green's functions applied to differential equations, the interested reader is referred to Roach (1982). In the signal processing and control theory literature, the Green's function is known as the impulse response of the system. Following the general latent force model framework, we write the outputs as

$$y_d(t) = \sum_{q=1}^{Q} S_{d,q} \mathcal{G}_d[u_q](t) + w_d(t), \tag{20}$$

where $w_d(t)$ is again an independent process associated to each output. We assume once more that the latent forces follow independent Gaussian process priors with zero mean and covariance $k_{u_q,u_q}(t, t')$. The covariance for the outputs $k_{y_d,y_{d'}}(t, t')$ is given by $k_{f_d,f_{d'}}(t, t') + k_{w_d,w_{d'}}(t, t')\delta_{d,d'}$, with $k_{f_d,f_{d'}}(t, t')$ equal to

$$\sum_{q=1}^{Q} S_{d,q} S_{d',q} k_{f_d^q,f_{d'}^q}(t, t'), \tag{21}$$

and $k_{f_d^q,f_{d'}^q}(t, t')$ following

$$\int_{\mathcal{T}} \int_{\mathcal{T}'} G_d(t - \tau) G_{d'}(t' - \tau') k_{u_q,u_q}(\tau, \tau') \mathrm{d}\tau' \mathrm{d}\tau. \tag{22}$$

Learning and inference for the higher-order latent force model is done as explained in subsection 3.2. The Green's function is described by a parameter vector $\boldsymbol{\psi}_d$ and with the length-scales $\{\ell_q\}_{q=1}^{Q}$ describing the latent GPs, the vector of hyper-parameters is given by $\boldsymbol{\theta} = \{\{\boldsymbol{\psi}_d\}_{d=1}^{D}, \{S_{d,q}\}_{d=1,q=1}^{D,Q}, \{\ell_q\}_{q=1}^{Q}\}$. The parameter vector $\boldsymbol{\theta}$ is estimated by maximizing the logarithm of the marginal likelihood in equation (13), where the elements of the matrix $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ are computed using expression (21) with $k_{f_d^q,f_{d'}^q}(t, t')$ given by (22). For prediction we use expression (14) and the posterior distribution is found using expression (15), where the elements of the matrix $\mathbf{K}_{\mathbf{f},\mathbf{u}}$, $k_{f_d,u_q}(t, t') = k_{f_d^q,u_q}(t, t')$, are computed using

$$S_{d,q} \int_{\mathcal{T}} G_d(t - \tau) k_{u_q,u_q}(\tau, t') \mathrm{d}\tau. \tag{23}$$

In section 4, we present in detail a second order latent force model and show its application in the description of motion capture data.

[4] We have used the same notation for the Kronecker delta and the Dirac delta. The particular meaning should be understood from the context.

3.4 Multidimensional inputs

In the sections above we have introduced latent force models for which the input variable is one-dimensional. For higher-dimensional inputs, $\mathbf{x} \in \mathbb{R}^p$, we can use linear partial differential equations to establish the dependence relationships between the latent forces and the outputs. 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. Inference and learning is done in a similar way to the one-input dimensional latent force model. Once the Green's function associated to the linear partial differential operator has been established, we employ similar equations to (22) and (23) to compute $k_{f_d,f_d'}(\mathbf{x},\mathbf{x}')$ and $k_{f_d,u_q}(\mathbf{x},\mathbf{x}')$ and the hyperparameters appearing in the covariance function are estimated by maximizing the marginal likelihood. In section 5, we will present examples of latent force models with spatio-temporal inputs and a basic covariance with higher-dimensional inputs.

3.5 Efficient approximations

Learning the parameter vector $\boldsymbol{\theta}$ through the maximization of expression (13) involves the inversion of the matrix $\mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma}$, 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 Csató and Opper (2001); Seeger et al. (2003); Quiñonero-Candela and Rasmussen (2005); Snelson and Ghahramani (2006); Rasmussen and Williams (2006); Titsias (2009). Recently, Álvarez and Lawrence (2009) introduced an efficient approximation for the case $D > 1$, which exploits the conditional independencies in equation (18): assuming that only a few number $K < N$ of values of $v(t)$ are known, then the set of outputs $f_d(t, v(t))$ are uniquely determined. The approximation obtained shared characteristics with the Partially Independent Training Conditional (PITC) approximation introduced in Quiñonero-Candela and Rasmussen (2005) and the authors of Álvarez and Lawrence (2009) refer to the approximation as the PITC approximation for multiple-outputs. The set of values $\{v(t_k)\}_{k=1}^K$ are known as inducing variables, and the corresponding set of inputs, inducing inputs. This terminology has been used before for the case in which $D = 1$.
A different type of approximation was presented in Álvarez et al. (2010) based on variational methods. It is a generalization of Titsias (2009) for multiple-output Gaussian processes. The approximation establishes a lower bound on the marginal likelihood and reduce 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 Quiñonero-Candela and Rasmussen (2005) and Titsias (2009).

4 Second Order Dynamical System

In Section 1 we introduced the analogy of a marionette's motion being controlled by a reduced number of forces. Human motion capture data consists of a skeleton and multivariate time courses of angles which summarize the motion. This motion can be modelled with a set of second order differential equations which, due to variations in the centers of mass induced by the movement, are non-linear. The simplification we consider for the latent force model is to linearize these differential equations, resulting in the following second order system,

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

Whilst 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 next section. Note also that, although similar to (5), the dynamic behavior of this system is much richer than that of the first order system, since it can exhibit inertia and resonance. In what follows, we will 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 summarized 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 one are said to be overdamped, whereas underdamped systems exhibit resonance and have a damping ratio less than one. For critically damped systems $\zeta_d = 1$, and finally, for undamped systems (i.e. no friction) $\zeta_d = 0$.
Ignoring the initial conditions, the solution of the second order differential equation is given by the integral operator of equation (18), with Green's function

$$G_d(t,s) = \frac{1}{\omega_d} \exp(-\alpha_d(t-s)) \sin(\omega_d(t-s)), \tag{24}$$

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

According to the general framework described in section 3.2, the covariance function between the outputs is obtained by solving expression (22), where $k_{u_q,u_q}(t,t')$ follows the SE form in equation (9). Solution for $k_{f_d^q,f_{d'}^q}(t,t')$ is then given by (Álvarez et al., 2009)

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

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

$$\Upsilon_q(\gamma_{d'},t,t')=2e^{\left(\frac{\ell_q^2\gamma_{d'}^2}{4}\right)}e^{-\gamma_{d'}(t-t')}-e^{\left(-\frac{(t-t')^2}{\ell_q^2}\right)}\mathrm{w}(jz_{d',q}(t))-e^{\left(-\frac{(t')^2}{\ell_q^2}\right)}e^{(-\gamma_{d'}t)}\mathrm{w}(-jz_{d',q}(0)),\qquad(25)$$

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 (25), 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, since $|\mathrm{w}(jz)|$ is bounded whenever the imaginary part of jz is greater or equal than zero, and is the key to achieving a good numerical stability when computing (25) and its gradients.

Similarly, the cross-covariance between latent functions and outputs in equation (23) is given by

$$k_{f_d^q,u_q}(t,t')=\frac{\ell_q S_{d,q}\sqrt{\pi}}{j4\omega_d}[\Upsilon_q(\widetilde{\gamma}_d,t,t')-\Upsilon_q(\gamma_d,t,t')],$$

Motion Capture data

Our motion capture data set is from the CMU motion capture data base.[5] We considered two different types of movements: golf-swing and walking. For golf-swing we consider subject 64 motions 1, 2, 3 and 4, and for walking we consider 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. Subsequently, we will refer to the pair subject and motion by the notation $X(Y)$, where X refers to the subject and Y to the particular motion. The data was down-sampled by 4.[6] 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 as explained in appendix A, ending up with 50 outputs for the golf-swing example and 33 outputs for the walking example.

We were interested in training on a subset of motions for each movement and testing on a different subset of motions for the same movement, to assess the model's ability to extrapolate. For testing, we condition on three angles associated to the root nodes and also on the first five and the last five output points of each other output. For the golf-swing, we use leave-one out cross-validation, in which one of the $64(Y)$ movements is left aside (with $Y=1,2,3$ 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).

We use the above setup to train a LFM model with $Q=2$. We compare our model against MTGP and SLFM, also with $Q=2$. For these three models, we use the DTCVAR efficient approximation with $K=30$ and fixed inducing-points placed equally spaced in the input interval. 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 SE covariance functions. 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.

We notice from table 1 that 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 the ability that the LFM has for generating more realistic motions.

5 Partial Differential Equations and Latent Forces

So far we have considered dynamical latent force models based on ordinary differential equations, leading to multioutput Gaussian processes which are functions of a single variable: time. As mentioned before, the methodology can also be

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

[6] 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.

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	$\mathbf{7.23 \pm 2.18}$	$\mathbf{48.15 \pm 5.66}$

Table 1: RMSE and R^2 for golf swing and walking

applied in the context of partial differential equations to recover multioutput Gaussian processes which are functions of several inputs. We first show an example of spatio-temporal 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.

5.1 Gap-gene network of Drosophila melanogaster

In this section we show an example of a latent force model for a spatio-temporal domain. For illustration, we use gene expression data obtained from the Gap-gene network of the Drosophila melanogaster. We propose a linear model that can account for the mechanistic behavior of the gene expression.
The gap gene network is responsible for the segmented body pattern of the *Drosophila melanogaster*. During the blastoderm stage of the development of the body, different maternal gradients determine the polarity of the embryo along its anterior-posterior (A-P) axis (Perkins et al., 2006).

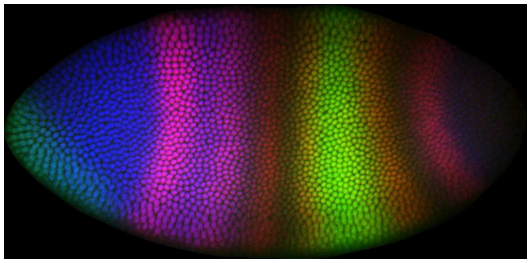

Figure 1: Drosophila body segmentation genes. Blue stripes correspond to hunchback, green stripes to knirps and red stripes to eve-skipped at cleavage cycle 14A, temporal class 3.

Maternal gradient interact with the so called trunk gap genes, including *hunchback (hb)*, *Krüppel (Kr)*, *giant (gt)*, and *knirps (kni)*, and this network of interactions establish the patterns of segmentation of the Drosophila.
Figure 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.[7]
The gap-gene network dynamics is usually represented using a set of coupled non-linear partial differential equations (Perkins et al., 2006; Gursky et al., 2004)

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

where $y_d(x,t)$ denotes the relative concentration of gap protein of the d-th 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 non-linear, 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 (Perkins et al., 2006).
We linearize the equation above by replacing the non-linear 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 which account for the influence of the latent force $u_q(x,t)$ over the quantity of production of

[7]The embryo name is dm12 and the image was taken from `http://urchin.spbcas.ru/flyex/`.

gene d. In this way, the new diffusion equation is given by

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

This expression corresponds to a second order non-homogeneous 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 given by Polyanin (2002); Butkovskiy and Pustyl'nikov (1993); Stakgold (1998)

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

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

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

We assume that the latent forces $u_q(x, t)$ follow a Gaussian process with covariance function that factorizes across inputs dimensions, this is

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

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 covariance for the outputs $y_d(x, t)$, $k_{f_d^q,f_{d'}^q}(x,t,x',t')$, is computed using the expressions for the Green's function and the covariance of the latent forces, in a similar fashion to equation (22), leading to

$$k_{f_d^q,f_{d'}^q}(x,t,x',t') = \frac{4}{\ell^2} \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} k^t_{f_d^q,f_{d'}^q}(t,t') k^x_{f_d^q,f_{d'}^q}(x,x'), \tag{26}$$

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 expression (10) and $k^x_{f_d^q,f_{d'}^q}(x,x')$ is given by

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

where $\omega_n = \frac{n\pi}{\ell}$ and $\omega_m = \frac{m\pi}{\ell}$. 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 included in appendix B.
For completeness, we also include the cross-covariance between the outputs and the latent functions, which follows as

$$k_{f_d,u_q}(x,t,x',t') = \frac{2S_{d,q}}{l} \sum_{n=1}^{\infty} k^t_{f_d,u_q}(t,t') k^x_{f_d,u_q}(x,x'),$$

where $k^t_{f_d,u_q}(t,t')$ is given by expression (12), and $k^x_{f_d,u_q}(x,x')$ follows

$$k^x_{f_d,u_q}(x,x') = \sin(w_n x)\, C(x',n,\ell_q^x),$$

where $C(x', n, \ell_q^x)$ is a function of x', the index n and the length-scale of the space input dimension. The expression for $C(x', n, \ell_q^x)$ is included in appendix C.

Prediction of gene expression data

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 and compare against the multi-task Gaussian process (MTGP) framework already mentioned in section 2. Covariance for the MTGP is a particular

case of the latent force model covariance in equations (21) and (22). If we make $G_d(t-\tau) = \delta(t-\tau)$ in equation (21), and $k_{u_q,u_q}(t,t') = k_{u,u}(t,t')$ for all values of q, we get

$$k_{f_d,f_{d'}}(t,t') = \sum_{q=1}^{Q} S_{d,q} S_{d',q} k_{u,u}(t,t').$$

Our purpose is to compare the prediction performance of the covariance above and the DROS covariance function.
We use data from Perkins et al. (2006), in particular, we have quantitative wild-type concentration profiles for the protein products of giant and knirps at 9 time points and 58 spatial locations. We work with a gene at a time and assume that the outputs correspond to the different time points. This setup is very common in computer emulation of multivariate codes (see Conti and O'Hagan (2010); Osborne et al. (2008); Rougier (2008)) in which the MTGP model is heavily used. For the DROS kernel, we use 30 terms in each sum involved in its definition, in equation (26).
We randomly select 20 spatial points for training the models, this is, for finding hyperparameters according to the description of subsection 3.2. 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 a traditional approach like MTGP, for this particular task.

Gene	Method	RMSE	$\mathbf{R^2}$ (%)
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	$\mathbf{3.01 \pm 0.81}$	$\mathbf{99.60 \pm 0.01}$

Table 2: RMSE and R^2 for protein data prediction

5.2 Diffusion in the Swiss Jura

The Jura data is 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 data set. We use a simplified version of the heat equation of p variables. The p-dimensional non-homogeneous heat equation is represented as

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

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 $\mathbb{R}^p = \{-\infty < x_j < \infty; j = 1,\dots,p\}$ and initial condition prescribed by the set of latent forces,

$$u(\mathbf{x}) = \sum_{q=1}^{Q} S_{d,q} u_q(\mathbf{x}), \quad \text{at } t = 0,$$

the solution to the system (Polyanin, 2002) is then given by

$$y_d(\mathbf{x},t) = \int_0^t \int_{\mathbb{R}^p} G_d(\mathbf{x},\mathbf{x}',t,\tau)\Phi(\mathbf{x}',\tau)\mathrm{d}\mathbf{x}'\mathrm{d}\tau + \int_{\mathbb{R}^p} G_d(\mathbf{x},\mathbf{x}',t,0)u(\mathbf{x}')\mathrm{d}\mathbf{x}', \tag{27}$$

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

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

with $T_{d,j}(t,\tau) = \kappa_{d,j}(t-\tau)$. The covariance function we propose here is derived as follows. In equation (27), we assume that the external force $\Phi(\mathbf{x},t)$ is zero, following

$$y_d(\mathbf{x},t) = \sum_{q=1}^{Q} S_{d,q} \int_{\mathbb{R}^p} G_d(\mathbf{x},\mathbf{x}',t,0)u_q(\mathbf{x}')\mathrm{d}\mathbf{x}'. \tag{28}$$

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

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

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

$$y_d(\mathbf{x}) = \sum_{q=1}^{Q} S_{d,q} \int_{\mathbb{R}^p} \widetilde{G}_d(\mathbf{x}, \mathbf{x}') u_q(\mathbf{x}') \mathrm{d}\mathbf{x}', \tag{29}$$

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 $\ell_{d,j}$. 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, 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

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

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 to the Gaussian covariance of the latent force Gaussian process prior. The covariance function between the output and latent functions, $k_{f_d^q, u_q}(\mathbf{x}, \mathbf{x}')$, is given by

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

where $\mathbf{P}_d^q = \mathbf{P}_d^{-1} + \mathbf{\Lambda}_q^{-1}$.

Prediction of Metal Concentrations

We used our model to replicate the experiments described in Goovaerts (pp. 248,249 1997) 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).[8] Figure 2 shows an example of the prediction problem. 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, as it is shown in the figure by the squared regions. In geostatistics, this configuration of sample locations is known as *undersampled* or *heterotopic*, 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 multi-task Gaussian process and the semiparametric latent factor model. For our experiments we made use of ten repeats to report standard deviations. For each repeat, the data is 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 render better results than multi-task GPs (in this case, the multi-task GP is equivalent to the semiparametric latent factor model). However, when increasing the value of the latent forces to two ($Q = 2$), 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.

[8] Data available at `http://www.ai-geostats.org/`.

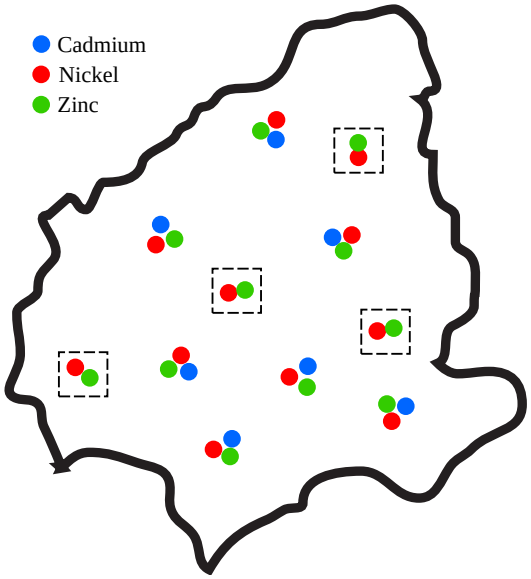

Figure 2: Sketch of the topsoil of Swiss Jura. Secondary variables like nickel and zinc help in the prediction of the primary variable cadmium, in the squared-regions.

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 3: RMSE 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

Table 4: R^2 for pollutant metal prediction

6 Related work

Differential equations are the cornerstone in a diverse range of engineering fields and applied sciences. However, their use for inference in statistics and machine learning has been less studied. The main field in which they have been used is known as *functional data analysis* (Ramsay and Silverman, 2005).

From the frequentist statistics point of view, the literature in functional data analysis has been concerned with the problem of parameter estimation in differential equations (Poyton et al., 2006; Ramsay et al., 2007): given a differential equation with unknown coefficients $\{\mathbf{A}_m\}_{m=0}^{M}$, how do we use data to fit those parameters? Notice that 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 might even be non-linear (Perkins et al., 2006)), 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. Having said that, we review some of 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 (references Ramsay et al. (2007) and Brewer et al. (2008) provide reviews and detailed descriptions of additional methods).
The solution by numerical approximations include 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. For exposition, we assume in equation (17) that $D = 1$, $Q = 1$ and $S_{1,1} = 1$. We are interested in finding the solution $y(t)$ to the following differential equation, with unknown parameters $\boldsymbol{\theta} = \{A_m\}_{m=0}^{M}$,

$$\mathcal{D}_0^M y(t) = \sum_{m=0}^{M} A_m \mathcal{D}^m y(t) = u(t),$$

In the classical approach, we assume that we have access to a vector of initial conditions, $\mathbf{y}_0$ and data for $u(t)$, $\mathbf{u}$. We start with an initial guess for the parameter vector $\boldsymbol{\theta}_0$ and solve numerically the differential equation to find a solution $\widetilde{\mathbf{y}}$. An updated parameter vector $\widetilde{\boldsymbol{\theta}}$ is obtained by minimizing

$$E(\boldsymbol{\theta}) = \sum_{n=1}^{N} \|\widetilde{y}(t_n) - y(t_n)\|,$$

through any gradient descent method. To use any of those methods, we must be able to compute $\partial E(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$, which is equivalent to compute $\partial y(t)/\partial \boldsymbol{\theta}$. In general, when we do not have access to $\partial y(t)/\partial \boldsymbol{\theta}$, we can compute it using what is known as the *sensitivity equations* (see Bard (1974), chapter 8, for detailed explanations), which are solved along with the ODE equation that provides the partial solution $\widetilde{\mathbf{y}}$. Once a new parameter vector $\widetilde{\boldsymbol{\theta}}$ has been found, the same steps are repeated until some convergence criterion is satisfied. If the initial conditions are not available, they can be considered as additional elements of the parameter vector $\boldsymbol{\theta}$ and optimized in the same gradient descent method.
In collocation methods, the solution of the differential equation is approximated using a set of basis functions, $\{\phi_i(t)\}_{i=1}^{J}$, this 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, this 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) (Ramsay, 1996) is one example of a collocation method in which the basis functions are *splines*. In PDA, the parameters of the differential equation are obtained by minimizing the squared residuals of the higher order derivative $\mathcal{D}^M y(t)$ and the weighted sum of derivatives $\{\mathcal{D}^m y(t)\}_{m=0}^{M-1}$, instead of the squared residuals between the approximated solution and the observed data.

An example of a collocation method augmented with Gaussian process priors was introduced by Graepel (2003). Graepel starts with noisy observations, $\widehat{y}(t)$, of the differential equation $\mathcal{D}_0^M y(t)$, such that $\widehat{y}(t) \sim \mathcal{N}(\mathcal{D}_0^M y(t), \sigma_y)$. The solution $y(t)$ is expressed using a basis representation, $y(t) = \sum \beta_i \phi_i(t)$. A Gaussian prior is placed over $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_J]$, and its posterior computed under the above likelihood. With the posterior over $\boldsymbol{\beta}$, the predictive distribution for $\widehat{y}(t_*)$ can be readily computed, being a function of the matrix $\mathcal{D}_0^M \boldsymbol{\Phi}$ with elements $\{\mathcal{D}_0^M \phi_i(t_n)\}_{n=1,i=1}^{N,J}$. It turns out that products $\mathcal{D}_0^M \boldsymbol{\Phi} \left(\mathcal{D}_0^M \boldsymbol{\Phi}\right)^\top$ that appear in this predictive distribution have individual elements that can be written using the sum $\sum_{i=1}^{J} \mathcal{D}_0^M \phi_i(t_n) \mathcal{D}_0^M \phi_i(t_{n'}) = \mathcal{D}_{0,t}^M \mathcal{D}_{0,t'}^M \sum_{i=1}^{J} \phi_i(t_n)\phi_i(t_{n'})$ or, using a kernel representation for the inner products $k(t_n, t_{n'}) = \sum_{i=1}^{J} \phi_i(t_n)\phi_i(t_{n'})$, as $k_{\mathcal{D}_{0,t}^M, \mathcal{D}_{0,t'}^M}(t_n, t_{n'})$, 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'. In other words, the result of the differential equation $\mathcal{D}_0^M y(t)$ is assumed to follow a Gaussian process prior with covariance $k_{\mathcal{D}_{0,t}^M, \mathcal{D}_{0,t'}^M}(t, t')$. An approximated solution $\widetilde{y}(t)$ can be computed through the expansion $\widetilde{y}(t) = \sum_{n=1}^{N} \alpha_n k_{\mathcal{D}_{0,t'}^M}(t, t_n)$, where α_n is an element of the vector $(\mathbf{K}_{\mathcal{D}_{0,t}^M, \mathcal{D}_{0,t'}^M} + \sigma_y \mathbf{I}_N)^{-1} \widehat{\mathbf{y}}$, 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 $\widehat{\mathbf{y}}$ are noisy observations of $\mathcal{D}_0^M y(t)$.

Although, we presented the above methods in the context of linear ODEs, solutions by numerical approximations and collocation methods are applied to non-linear ODEs as well.

Gaussian processes have been used as models for systems identification (Solak et al., 2003; Kocijan et al., 2005; Calderhead et al., 2009; Thompson, 2009). In Solak et al. (2003), a non-linear dynamical system is linearized around an equilibrium

point by means of a Taylor series expansion (Thompson, 2009),

$$y(t) = \sum_{j=0}^{\infty} \frac{y^{(j)}(a)}{j!}(t-a)^j,$$

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 derivative 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. An important consequence of including derivative information in the inference process is that the uncertainty in the posterior prediction is reduced as compared to using only function observations. This aspect of derivative information have been exploited in the theory of computer emulation to reduce the uncertainty in experimental design problems (Morris et al., 1993; Mitchell and Morris, 1994).

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 (Ljung, 1999). The particular dependency $y(t) = g(\{y(t-t_{k-l})\}_{l=1}^{L})$, where $g(\cdot)$ is a general non- linear function, is modelled using a Gaussian process prior and the predicted value for the output $y_*(t_k)$ is used as a new input for multi-step ahead prediction at times t_j, with $j > k$ (Kocijan et al., 2005). Uncertainty about $y_*(t_k)$ can also be incorporated for predictions of future output values (Girard et al., 2003).

On the other hand, multivariate systems with Gaussian process priors have been thoroughly studied in the spatial analysis and geostatistics literature (Higdon, 2002; Boyle and Frean, 2005; Journel and Huijbregts, 1978; Cressie, 1993; Goovaerts, 1997; Wackernagel, 2003). In short, a valid covariance function for multi-output 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 (Teh et al., 2005) and the multi-task GP (Bonilla et al., 2008) can be seen as particular cases of the LMC (Álvarez et al., 2011b). Higdon (2002) proposed the direct use of a expression (18) 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 single output GP had already been proposed by Barry and Ver Hoef (1996); Ver Hoef and Barry (1998). Calder and Cressie (2007) reviews several extensions of the single process convolution covariance. Boyle and Frean (2005) introduced the process convolution idea for multiple outputs to the machine learning audience. Boyle (2007) developed the idea of using impulse responses of filters to represent $G_d(t,s)$, assuming the process $v(t)$ was white Gaussian noise. Independently, Murray-Smith and Pearlmutter (2005) also introduced the idea of transforming a Gaussian process prior using a discretized version of the integral operator of equation (18). Such 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 (Shi et al., 2005).

There has been a recent interest in introducing Gaussian processes in the state space formulation of dynamical systems (Ko et al., 2007; Deisenroth et al., 2009; Turner et al., 2010) for the representation of the possible nonlinear relationships between the latent space and between the latent space and the observation space. Going back to the formulation of the dimensionality reduction model, we have

$$\begin{aligned}\mathbf{u}_{t_n} &= \mathbf{g}_1(\mathbf{u}_{t_{n-1}}) + \boldsymbol{\eta},\\ \mathbf{y}_{t_n} &= \mathbf{g}_2(\mathbf{u}_{t_n}) + \boldsymbol{\epsilon},\end{aligned}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are noise processes and $\mathbf{g}_1(\cdot)$ and $\mathbf{g}_2(\cdot)$ are general non-linear functions. Usually $\mathbf{g}_1(\cdot)$ and $\mathbf{g}_2(\cdot)$ are unknown, and research on this area has focused on developing a practical framework for inference when assigning Gaussian process priors to both functions.

Finally, it is important to highlight the work of Calder (Calder, 2003, 2007, 2008) as an alternative to multiple-output modeling. Her work can be seen in the context of state-space models,

$$\begin{aligned}\mathbf{u}_{t_n} &= \mathbf{u}_{t_{n-1}} + \boldsymbol{\eta},\\ \mathbf{y}_{t_n} &= \mathbf{G}_{t_n}\mathbf{u}_{t_n} + \boldsymbol{\epsilon},\end{aligned}$$

where $\mathbf{y}_{t_n}$ and $\mathbf{u}_{t_n}$ are related through a discrete convolution over an independent spatial variable. This is, for a fixed t_n, $y_d^{t_n}(\mathbf{s}) = \sum_q \sum_i G_d^{t_n}(\mathbf{s}-\mathbf{z}_i)u_q^{t_n}(\mathbf{z}_i)$ for a grid of I spatial inputs $\{\mathbf{z}_i\}_{i=1}^{I}$.

7 Conclusion

In this paper we have presented a hybrid approach to modelling 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. The implementation of latent force models introduced in this paper can be extended in several ways, including:
Non-linear Latent Force Models. If the likelihood function is not Gaussian the inference process has to be accomplished in a different way, through a Laplace approximation (Lawrence et al., 2007) or sampling techniques (Titsias et al., 2009).
Cascaded Latent Force Models. For the above presentation of the latent force model, we assumed that the covariances $k_{u_q,u_q}(t,t')$ were squared-exponential. However, more structured covariances can be used. For example, in Honkela et al. (2010), the authors use a cascaded system to describe gene expression data for which a first order system, like the one presented in subsection 3.1, has as inputs $u_q(t)$ governed by Gaussian processes with covariance function (10).
Stochastic Latent Force Models. If the latent forces $u_q(t)$ are white noise processes, then the corresponding differential equations are stochastic, and the covariances obtained in such cases lead to stochastic latent force models. In Álvarez et al. (2010), a first-order stochastic latent force model is employed for describing the behavior of a multivariate financial dataset: the foreign exchange rate with respect to the dollar of ten of the top international currencies and three precious metals.
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 Álvarez et al. (2011a) this model was used for the segmentation of movements performed by a Barrett WAM robot as haptic input device.

Acknowledgments

DL has been partly financed by Comunidad de Madrid (project PRO-MULTIDIS-CM, S-0505/TIC/0233), and by the Spanish government (CICYT project TEC2006-13514-C02-01 and researh grant JC2008-00219). MA and NL have been financed by a Google Research Award and EPSRC Grant No EP/F005687/1 "Gaussian Processes for Systems Identification with Applications in Systems Biology". MA also acknowledges the 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.

A Preprocessing for the mocap data

For selecting the subset of angles for each of the motions for the golf-swing movement and the walking movement, we use as performance measure the signal-to-noise ratio obtained in the following way. 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,

$$\sigma_S^2 \exp\left[-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\ell^2}\right] + \sigma_N^2 \delta(\mathbf{x},\mathbf{x}'),$$

where σ_S^2 and σ_N^2 are variance parameters, and $\delta(\mathbf{x},\mathbf{x}')$ is the Dirac delta function. For each output, we compute the signal-to-noise ratio as $10\log_{10}(\sigma_S^2/\sigma_N^2)$.

B Expression for $C(n,m,\ell_x)$

For simplicity, we assume $Q=1$ and write $\ell_q^x = \ell_x$. The expression for $C(n,m,\ell_x)$ is given by

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

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

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

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

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

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

The term $C(n, m, \ell_x)$ is zero if, for $n \neq m$, n is even and m is odd or viceversa.

Finally, when $n = m$, the expression for $C(n, n, \ell_x)$ follows as

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

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

C Expression for $C(x', n, \ell_x)$

As in appendix B, we assume $Q = 1$ and $\ell_q^x = \ell_x$. The expression $C(x', n, \ell_x)$ is as

$$\frac{\ell_x\sqrt{\pi}}{2}\mathcal{I}\left[e^{-\left(\frac{x'-l}{\ell_x}\right)^2} e^{\gamma_n l}\mathrm{w}(jz_2^{\gamma_n, x'}) - e^{-\left(\frac{x'}{\ell_x}\right)^2}\mathrm{w}(jz_1^{\gamma_n, x'})\right],$$

with $z_1^{\gamma_n, x'} = \frac{x'}{\ell_x} + \frac{\ell_x \gamma_n}{2}$, $z_2^{\gamma_n, x'} = \frac{x'-l}{\ell_x} + \frac{\ell_x \gamma_n}{2}$, $\gamma_n = j\omega_n$ and $\mathcal{I}[\cdot]$ is an operator that takes the imaginary part of the argument.

References

Mauricio A. Álvarez and Neil D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. In Koller et al. (2009), pages 57–64.

Mauricio A. Álvarez, David Luengo, and Neil D. Lawrence. Latent Force Models. In van Dyk and Welling (2009), pages 9–16.

Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In Teh and Titterington (2010), pages 25–32.

Mauricio A. Álvarez, Jan Peters, Bernhard Schölkopf, and Neil D. Lawrence. Switched latent force models for movement segmentation. In *NIPS*, volume 24, pages 55–63. MIT Press, Cambridge, MA, 2011a.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: a review. Technical report, University of Manchester, Massachusetts Institute of Technology and University of Sheffield, 2011b. Available at `http://arxiv.org/pdf/1106.6251v1`.

Yonathan Bard. *Nonlinear Parameter Estimation*. Academic Press, first edition, 1974.

Ronald Paul Barry and Jay M. Ver Hoef. Blackbox kriging: spatial prediction without specifying variogram models. *Journal of Agricultural, Biological and Environmental Statistics*, 1(3):297–322, 1996.

Sue Becker, Sebastian Thrun, and Klaus Obermayer, editors. *NIPS*, volume 15, Cambridge, MA, 2003. MIT Press.

Edwin V. Bonilla, Kian Ming Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In John C. Platt, Daphne Koller, Yoram Singer, and Sam Roweis, editors, *NIPS*, volume 20, Cambridge, MA, 2008. MIT Press.

Phillip Boyle. *Gaussian Processes for Regression and Optimisation*. PhD thesis, Victoria University of Wellington, Wellington, New Zealand, 2007.

Phillip Boyle and Marcus Frean. Dependent Gaussian processes. In Lawrence Saul, Yair Weiss, and Léon Bouttou, editors, *NIPS*, volume 17, pages 217–224, Cambridge, MA, 2005. MIT Press.

Daniel Brewer, Martino Barenco, Robin Callard, Michael Hubank, and Jaroslav Stark. Fitting ordinary differential equations to short time course data. *Philosophical Transactions of the Royal Society A*, 366:519–544, 2008.

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

Catherine A. Calder. Dynamic factor process convolution models for multivariate space-time data with application to air quality assessment. *Environmental and Ecological Statistics*, 14(3):229–247, 2007.

Catherine A. Calder. A dynamic process convolution approach to modeling ambient particulate matter concentrations. *Environmetrics*, 19:39–48, 2008.

Catherine A. Calder. *Exploring latent structure in spatial temporal processes using process convolutions*. PhD thesis, Institute of Statistics and Decision Sciences, Duke University, Durham, NC, USA, 2003.

Catherine A. Calder and Noel Cressie. Some topics in convolution-based spatial modeling. In *Proceedings of the 56th Session of the International Statistics Institute*, August 2007.

Ben Calderhead, Mark Girolami, and Neil D Lawrence. Accelerating bayesian inference over nonlinear differential equations with gaussian processes. In Koller et al. (2009), pages 217–224.

Stefano Conti and Anthony O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3):640–651, 2010.

Noel A. C. Cressie. *Statistics for Spatial Data*. John Wiley & Sons (Revised edition), USA, 1993.

Lehel Csató and Manfred Opper. Sparse representation for Gaussian process models. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS*, volume 13, pages 444–450, Cambridge, MA, 2001. MIT Press.

Marc Peter Deisenroth, Marco F. Huber, and Uwe D. Hanebeck. Analytic moment-based Gaussian process filtering. In *Proceedings of the 26th International Conference on Machine Learning (ICML 2009)*, pages 225–232. Omnipress, 2009.

Pei Gao, Antti Honkela, Magnus Rattray, and Neil D. Lawrence. Gaussian process modelling of latent chemical species: Applications to inferring transcription factor activities. *Bioinformatics*, 24:i70–i75, 2008. doi: 10.1093/bioinformatics/btn278.

Agathe Girard, Carl Edward Rasmussen, Joaquin Quiñonero Candela, and Roderick Murray-Smith. Gaussian process priors with uncertain inputs – application to multiple-step ahead time series forecasting. In Becker et al. (2003), pages 529–536.

Pierre Goovaerts. *Geostatistics For Natural Resources Evaluation*. Oxford University Press, USA, 1997.

Thore Graepel. Solving noisy linear operator equations by Gaussian processes: Application to ordinary and partial differential equations. In Tom Fawcett and Nina Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 234–241, Washington, DC, USA, August 21-24 2003. AAAI Press.

D. H. Griffel. *Applied Functional Analysis*. Dover publications Inc., Mineola, New York, reprinted edition, 2002.

Vitaly V. Gursky, Johannes Jaeger, Konstantin N. Kozlov, John Reinitz, and Alexander Samsonov. Pattern formation and nuclear divisions are uncoupled in Drosophila segmentation: comparison of spatially discrete and continuous models. *Physica D*, 197:286–302, 2004.

David M. Higdon. Space and space-time modelling using process convolutions. In C. Anderson, V. Barnett, P. Chatwin, and A. El-Shaarawi, editors, *Quantitative methods for current environmental issues*, pages 37–56. Springer-Verlag, 2002.

Antti Honkela, Charles Girardot, E. Hilary Gustafson, Ya-Hsin Liu, Eileen E. M. Furlong, Neil D. Lawrence, and Magnus Rattray. Model-based method for transcription factor target identification with limited data. *Proc. Natl. Acad. Sci.*, 107 (17):7793–7798, 2010.

Andre G. Journel and Charles J. Huijbregts. *Mining Geostatistics*. Academic Press, London, 1978. ISBN 0-12391-050-1.

Jonathan Ko, Daniel J. Klein, Dieter Fox, and Dirk Haehnel. GP-UKF: Unscented Kalman filters with Gaussian process prediction and observation models. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1901–1907, San Diego, CA, USA, 2007.

Juš Kocijan, Agathe Girard, Blaž Banko, and Roderick Murray-Smith. Dynamic systems identification with Gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424, 2005.

Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors. *NIPS*, volume 21, Cambridge, MA, 2009. MIT Press.

Neil D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, Nov. 2005.

Neil D. Lawrence, Guido Sanguinetti, and Magnus Rattray. Modelling transcriptional regulation using Gaussian processes. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *NIPS*, volume 19, pages 785–792. MIT Press, Cambridge, MA, 2007.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, Upper Saddle River, New Jersey, second edition, 1999.

Toby J. Mitchell and Max Morris. Asymptotically optimum experimental designs for prediction of deterministic functions given derivative information. *Journal of Statistical Planning and Inference*, 41:377–389, 1994.

Max D. Morris, Toby J. Mitchell, and Donald Ylvisaker. Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics*, 35(3):243–255, 1993.

Roderick Murray-Smith and Barak A. Pearlmutter. Transformation of Gaussian process priors. In Joab Winkler, Mahesan Niranjan, and Neil Lawrence, editors, *Deterministic and Statistical Methods in Machine Learning*, pages 110–123. LNAI 3635, Springer-Verlag, 2005.

Bert Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, Germany, sixth edition, 2003.

Michael A. Osborne, Alex Rogers, Sarvapali D. Ramchurn, Stephen J. Roberts, and Nicholas R. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN 2008)*, 2008.

Theodore J. Perkins, Johannes Jaeger, John Reinitz, and Leon Glass. Reverse engineering the gap gene network of Drosophila melanogaster. *PLOS Comput Biol*, 2(5):417–427, 2006.

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

A.A. Poyton, M. Saeed Varziri, Kim B. McAuley, James McLellan, and Jim O. Ramsay. Parameter estimation in continuous-time dynamic models using principal differential analysis. *Computers and Chemical Engineering*, 30:698–708, 2006.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

Jim O. Ramsay. Principal differential analysis: Data reduction by differential operators. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(3):495–508, 1996.

Jim O. Ramsay and Bernard W. Silverman. *Functional Data Analysis*. Springer Series in Statistics, New York, NY, (USA), second edition, 2005.

Jim O. Ramsay, G. Hooker, D. Campbell, and J. Cao. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society. Series B (Methodological)*, 69(5):741–796, 2007.

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

Linda E. Reichl. *A modern course in statistical physics*. John Wiley & Sons, United States of America, second edition, 1998.

Gary F. Roach. *Green's functions*. Cambridge University Press, Cambridge, UK, second edition, 1982.

Jonathan Rougier. Efficient emulators for multivariate deterministic functions. *Journal of Computational and Graphical Statistics*, 17(4):827–834, 2008.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL, 3–6 Jan 2003.

Sam K. Shanmugan and Arthur M. Breipohl. *Random signals: detection, estimation and data analysis*. John Wiley & Sons, United States of America, first edition, 1988.

Jian Qing Shi, Roderick Murray-Smith, D.M. Titterington, and Barak Pearlmutter. Learning with large data sets using filtered Gaussian process priors. In R. Murray-Smith and R. Shorten, editors, *Proceedings of the Hamilton Summer School on Switching and Learning in Feedback systems*, pages 128–139. LNCS 3355, Springer-Verlag, 2005.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Yair Weiss, Bernhard Schölkopf, and John C. Platt, editors, *NIPS*, volume 18, Cambridge, MA, 2006. MIT Press.

Ercan Solak, Roderick Murray-Smith, William E. Leithead, Douglas J. Leith, and Carl E. Rasmussen. Derivative observations in Gaussian process models of dynamic systems. In Becker et al. (2003), pages 1033–1040.

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

Yee Whye Teh and Mike Titterington, editors. *AISTATS*, Chia Laguna, Sardinia, Italy, 13-15 May 2010. JMLR W&CP 9.

Yee Whye Teh, Matthias Seeger, and Michael I. Jordan. Semiparametric latent factor models. In Robert G. Cowell and Zoubin Ghahramani, editors, *AISTATS 10*, pages 333–340, Barbados, 6-8 January 2005. Society for Artificial Intelligence and Statistics.

Keith R. Thompson. *Implementation of Gaussian Process Models for nonlinear system Identification*. PhD thesis, Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow, UK, 2009.

Michalis Titsias, Neil D Lawrence, and Magnus Rattray. Efficient sampling for Gaussian process inference using control variables. In Koller et al. (2009), pages 1681–1688.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In van Dyk and Welling (2009), pages 567–574.

Ryan Turner, Marc Peter Deisenroth, and Carl Edward Rasmussen. State-space inference and learning with Gaussian processes. In Teh and Titterington (2010), pages 868–875.

David van Dyk and Max Welling, editors. *AISTATS*, Clearwater Beach, Florida, 16-18 April 2009. JMLR W&CP 5.

Jay M. Ver Hoef and Ronald Paul Barry. Constructing and fitting models for cokriging and multivariable spatial prediction. *Journal of Statistical Plannig and Inference*, 69:275–294, 1998.

Hans Wackernagel. *Multivariate Geostatistics*. Springer-Verlag Heidelberg New york, 2003.