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 development of Drosophila Melanogaster, and a latent random noise, and $u_n$ is the $n$th row of $U$, which we associate with time $t_n$. This is known as the Kalman filter/smooth.

Normally, the times, $t_n$, are taken to be equally spaced, but more generally we can consider a joint distribution for $p(U | t)$, for a vector of time inputs $t = [t_1 \ldots t_N]^T$, which has the form of a Gaussian process:

$$p(U | t) = \prod_{q=1}^{Q} N(u_q | 0, K_{u_q, u_q}),$$

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 $u_q = [u_q(t_1), \ldots, u_q(t_N)]^T$ represent the values of the function for the $q$th dimension at the times given by $t$. The matrix $K_{u_q, u_q}$ is the covariance function associated with $u_q(t)$ computed at the times given in $t$.

Such a GP can be readily implemented. Given the covariance functions for $\{u_q(t)\}_{q=1}^{Q}$, the implied covariance functions for $\{y(t)\}_{q=1}^{Q}$ 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, $YB = US^T + E$. Here, we have assumed that the forces are acting, for example, through levers, so that we have a matrix of sensitivities, $S \in \mathbb{R}^{D \times Q}$, and a diagonal matrix of spring constants, $B \in \mathbb{R}^{D \times D}$, with elements $\{B_d\}_{d=1}^D$. The original model is recovered by setting $W^T = SB^{-1}$ and $E \sim N(0, B^{-1} \Sigma_d B)$. 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

\[ \dot{Y}_m + YC + YB = US^T + \bar{E}, \]

where \( M \) and \( C \) are diagonal matrices of masses, \( \{M_d\} \), and damping coefficients, \( \{C_d\} \), respectively, \( Y \) is the first derivative of \( Y \) with respect to time (with entries \( \{y_d(t_n)\} \) for \( d = 1, \ldots, D \) and \( n = 1, \ldots, N \)), \( \dot{Y} \) is the second derivative of \( Y \) with respect to time (with entries \( \{\dot{y}_d(t_n)\} \) for \( d = 1, \ldots, D \) and \( n = 1, \ldots, N \)), and \( E \) is once again matrix-variate Gaussian noise. Equation (2) specifies a particular type of interaction between the outputs \( Y \) and the set of latent functions \( U \), namely, that a weighted sum of the second derivative for \( y_d(t) \), \( \dot{y}_d(t) \), the first derivative for \( y_d(t) \), \( \dot{y}_d(t) \), and \( y_d(t) \) is equal to the weighted sum of functions \( \{u_q(t)\} \) 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, \( \{u_q(t)\} \). 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:

\[ \sum_{m=0}^{M} D^m[Y] A_m = US^T + \bar{E}, \]

where \( D^m \) is a linear differential operator such that \( D^m[Y] \) is a matrix with elements given by \( D^m[y_d(t)] = \frac{d^m y_d(t)}{dt^m} \), and \( A_m \) is a diagonal matrix with elements \( A_{md} \) that weight the contribution of \( D^m y_d \).

Each element in expression (3) can be written as

\[ D_0^m y_d = \sum_{n=0}^{M} A_{md} D^m y_d(t) = \sum_{q=1}^{Q} S_{dq} u_q(t) + \epsilon_d(t), \]

where we have introduced a new operator \( D_0^m \) that is equivalent to applying the weighted sum of operators \( D^m \). It is possible to find a linear integral operator \( G_d \) associated with \( D_0^m \) that can be used to solve the nonhomogeneous differential equation in (4). The linear integral operator is defined as

\[ G_d[v](t) = f_d(t,v(t)) = \int_T G_d(t,\tau)v(\tau)d\tau, \]

where \( G_d(t,s) \) is known as the Green’s function associated with the differential operator \( D_0^m \), \( v(t) \) is the input function for the nonhomogeneous differential equation and \( T \) is the input domain. The particular relation between the differential operator and the Green’s function is given by

\[ D_0^m G_d(t,s) = \delta(t-s), \]

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

\[ y_d(t) = f_d(t) + w_d(t) = \sum_{q=1}^{Q} S_{dq} f_d(t, u_q(t)) + w_d(t), \]

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

We assume that the latent functions \( \{u_q(t)\} \) are independent and each of them follows a Gaussian process prior, that is, \( u_q(t) \sim GP(0, k_{u_q}(t,t')) \). Due to the linearity of \( y_d \), \( \{y_d(t)\} \) corresponds to a joint Gaussian process with covariances \( k_{y_d}(t,t') = \text{cov}[y_d(t), y_d(t')] \) given by

\[ \text{cov}[f_d(t), f_d(t')] + \text{cov}[w_d(t), w_d(t')] \delta_{d,d'}, \]

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

\[ \sum_{q=1}^{Q} S_{dq} S_{dq'} \text{cov}[f_d(t), f_d(t')]. \]

1. The matrices \( A_m \) do not need to be diagonal, but for simplicity of derivation we restrict ourselves to this setup 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 \( f_d^0(t) \) as shorthand for \( f_d(t, u_d(t)) \). The covariance \( \text{cov}[f_d^0(t), f_d^0(t')] \) is equal to

\[
\int_T \int_T G_d(t - \tau)G_d(t' - \tau')k_{u_d u_d}(\tau, \tau')d\tau d\tau'.
\] (8)

We alternatively denote \( \text{cov}[f_d(t), f_d(t')] \) as \( k_{f_d f_d}(t, t') \), \( \text{cov}[f_d^0(t), f_d^0(t')] \) as \( k_{f_d f_d^0}(t, t') \), and \( \text{cov}[u_d(t), w_d(t')] \) as \( k_{u_d w_d}(t, t') \).

Notice from (8) above that the covariance between \( f_d^0(t) \) and \( f_d^0(t') \) depends on the covariance \( k_{u_d u_d}(\tau, \tau') \). The form for the covariance \( k_{u_d u_d}(t, t') \) should be such that we can solve both integrals in (8). Two alternatives for \( k_{u_d u_d}(\tau, \tau') \) have been considered before in the context of latent force models. In [10], the covariance \( k_{u_d u_d}(\tau, \tau') \) was considered to follow the squared exponential (SQEXP) form [11]:

\[
k_{u_d u_d}(t, t') = \exp \left( -\frac{(t - t')^2}{\ell^2} \right),
\] (9)

where \( \ell \) is known as the length-scale. In [12], the covariance \( k_{u_d u_d}(\tau, \tau') \) was associated with a Gaussian white noise and therefore followed the form \( k_{u_d u_d}(\tau, \tau') = \sigma^2_d \delta(\tau - \tau') \), where \( \sigma^2_d \) stands for the variance of the white noise. As long as the double integral in (8) can be solved analytically, other forms for \( k_{u_d u_d}(\tau, \tau') \) 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 \( u_d(t) \), \( k_{f_d u_d}(t, t') \), follows:

\[
S_d \int_T G_d(t - \tau)k_{u_d u_d}(\tau, t')d\tau.
\] (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, \( x \in \mathbb{R}^p \), we can use partial differential equations to establish the dependence relationships between the latent forces, \( \{u_d(x)\}_{d=1}^D \), and the outputs, \( \{y_d(x)\}_{d=1}^D \). The initial conditions turn into boundary conditions, specified by a set of functions that are linear combinations of \( y_d(x) \) 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 \( k_{f_d u_d}(x, x') \) and \( k_{f_d f_d}(x, x') \). Now the covariance for the outputs is written as \( k_{u_d u_d}(x, x') \), and is given by \( k_{f_d f_d}(x, x') + k_{f_d u_d}(x, x') \). The covariance for the independent process \( u_d(x) \) is given by \( k_{u_d u_d}(x, x') \).

In the context of latent force models, choices for \( k_{u_d u_d}(x, x') \) 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 LFMds, prediction for test cases, and computational complexity. The description is done in terms of the input space \( x \in \mathbb{R}^p \), the dynamical latent force model (where \( x = t \)) being a special case.

#### 4.1 Hyperparameter Learning

Gaussian processes allow us to trivially marginalize the effect of the latent forces, \( \{u_d(x)\}_{d=1}^D \), by focusing only on the covariance for the outputs, \( k_{u_d u_d}(x, x') \). Given a set of inputs \( X = \{x_n\}_{n=1}^N \) and the parameters \( \theta \) of the covariance function, the marginal likelihood for the outputs can be written as

\[
p(y \mid X, \theta) = \mathcal{N}(y \mid 0, K_{t \mid t'} + \Sigma),
\] (11)

where \( y = \text{vec}Y \), \( K_{t \mid t'} = \mathbb{E}[y_T y'] \) with each element given by \( \text{cov}[f_d(x_n), f_d(x_{n'})] \) for \( n = 1, \ldots, N \) and \( n' = 1, \ldots, N \), and \( \Sigma \) represents the covariance associated with the independent processes \( u_d(x) \).

In general, the vector of hyperparameters \( \theta \) 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 \( X_t \), is done using standard Gaussian process regression techniques. The predictive distribution is given by

\[
p(y_t \mid y, X, \theta) = \mathcal{N}(y_t \mid \mu_{y_t}, K_{y_t y_t}),
\]

where \( \mu_{y_t} = K_{t \mid t'}(K_{t \mid t'} + \Sigma)^{-1}y \) and \( K_{y_t y_t} = K_{t \mid t'} - K_{t \mid t'}(K_{t \mid t'} + \Sigma)^{-1}K_{t \mid t'} \). Here, we have used \( K_{t \mid t'} \) to represent the evaluation of \( K_{t \mid t'} \) at the input set \( X_t \).

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

\[
p(u \mid y, X, \theta) = \mathcal{N}(u \mid \mu_{u y}, K_{u y}),
\]

where \( \mu_{u y} = K_{u y}^T(K_{t \mid t'} + \Sigma)^{-1}y \) and \( K_{u y} = K_{u u} - K_{t \mid t'}K_{t \mid t'}(K_{t \mid t'} + \Sigma)^{-1}K_{t \mid t'} \), where \( u = \text{vec}U \). \( K_{u u} \) is a block-diagonal matrix with blocks given by \( K_{u_d u_d} \). In turn, the elements of \( K_{u_d u_d} \) are given by \( k_{u_d u_d}(x, x') \). Also, \( K_{u u} \) is a matrix with blocks \( K_{f_d u_d} \), where \( K_{f_d u_d} \) has entries given by \( k_{f_d u_d}(x, x') \).

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

5. \( x = \text{vec}X \) is the vectorization operator that transforms the matrix \( X \) into a vector \( x \). The vector is obtained by stacking the columns of the matrix.
4.3 Efficient Approximations

Learning the hyperparameter vector $\theta$ through the maximization of the logarithm of the marginal likelihood in (11) involves the inversion of the matrix $K_{ff} + \Sigma$, inversion that scales as $O(D^3 N^3)$. 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 $u(x)$ are known, then the set of outputs $f_i(x, u(x))$ 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 $\{u(z_k)\}_{k=1}^K$ are known as inducing variables, and the corresponding set of inputs, $\{z_k\}_{k=1}^K$ 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 $O(DNK^2)$. 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 course data in human-motion datasets, describing the dynamic behavior of this system 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, $\gamma_d(t)$ 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, $\omega_d = \sqrt{B_d/C_d}$ and the damping ratio, $\zeta_d = \frac{1}{2\sqrt{B_d/C_d}}$. 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

$$G_d(t, s) = \frac{1}{\omega_d^2} \exp(-\alpha_d(t - s)) \sin(\omega_d(t - s)),$$

where $\omega_d = \sqrt{4B_d - C_d/\alpha_d}$ 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 $k_{\gamma_d, \gamma_d}(t, t')$ follows the SQEXP form in (9). Solution for $k_{\gamma_d, \gamma_d}(t, t')$ is then given by [10]

$$k_{\gamma_d, \gamma_d}(t, t') = \frac{\alpha_d \gamma_d}{\gamma_d + \alpha_d} T_d(\gamma_d, t', t) - e^{-\gamma_d t} T_d(\gamma_d, t', 0),$$

with $T_d(\gamma_d, t', t)$:

$$T_d(\gamma_d, t', t) = \begin{cases} \frac{1}{\gamma_d + \alpha_d} \left( \frac{1}{\beta_d} \right)^{\frac{\gamma_d^2}{2}} \exp(-\frac{\gamma_d^2}{4}) w(j \gamma_d \phi(t)) & \text{if } t < t' \\ \frac{1}{\gamma_d + \alpha_d} \left( \frac{1}{\beta_d} \right)^{\frac{\gamma_d^2}{2}} \exp(-\frac{\gamma_d^2}{4}) w(-j \gamma_d \phi(0)) & \text{if } t > t' \end{cases},$$

where $\phi(t)$ represents the movement through judicious use of the strings.
and $z_{\delta d}(t) = (t - t')/\ell_\delta - (\ell_\delta \gamma_d)/2$. Note that $z_{\delta d}(t) \in \mathbb{C}$, and $w(jz)$ in (12), for $z \in \mathbb{C}$, denotes Faddeeva's function $w(jz) = \exp(z^2) \text{erfc}(z)$, where $\text{erfc}(z)$ is the complex version of the complementary error function, $\text{erfc}(z) = 1 - \text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_z^{\infty} \exp(-t^2) dt$. Faddeeva's function is usually considered the complex equivalent of the error function because $|w(jz)|$ is bounded whenever the imaginary part of $z$ 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 $k_{f_{q_1}y}(t, t') = \frac{4\ell_\delta}{\ell_\delta + \ell'_\delta} \left[ 1 - \frac{t + t'}{2\ell_\delta} \right]$. This is particularly true for the $R^2$ performance measure, indicating that the LFM generates more realistic motions.

### 5.3 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

<table>
<thead>
<tr>
<th>Movement</th>
<th>Method</th>
<th>RMSE</th>
<th>$R^2$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golf swing</td>
<td>IND GP</td>
<td>21.55 ± 2.35</td>
<td>30.99 ± 9.67</td>
</tr>
<tr>
<td></td>
<td>MTGP</td>
<td>21.19 ± 2.18</td>
<td>45.59 ± 7.86</td>
</tr>
<tr>
<td></td>
<td>SLFM</td>
<td>21.52 ± 1.93</td>
<td>49.32 ± 3.03</td>
</tr>
<tr>
<td></td>
<td>LFM</td>
<td><strong>18.09 ± 1.30</strong></td>
<td><strong>72.25 ± 3.08</strong></td>
</tr>
<tr>
<td>Walking</td>
<td>IND GP</td>
<td>8.03 ± 2.55</td>
<td>30.55 ± 10.64</td>
</tr>
<tr>
<td></td>
<td>MTGP</td>
<td>7.75 ± 2.05</td>
<td>37.77 ± 4.53</td>
</tr>
<tr>
<td></td>
<td>SLFM</td>
<td>7.81 ± 2.00</td>
<td>36.84 ± 4.26</td>
</tr>
<tr>
<td></td>
<td>LFM</td>
<td><strong>7.23 ± 2.18</strong></td>
<td><strong>48.15 ± 5.66</strong></td>
</tr>
</tbody>
</table>

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, $\zeta$, 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.
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 knirsps, and the eve-skipped genes in a color-scale intensity image. The image corresponds to cleavage cycle 14A, temporal class 3.

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

\[ \frac{\partial y_d(x, t)}{\partial t} = \zeta(t) P_d(y(x, t)) - \lambda_d y_d(x, t) + D_d \frac{\partial^2 y_d(x, t)}{\partial x^2}, \]

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 nonlinear, of production of all other genes. The parameter \( \lambda_d \) 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

\[ \frac{\partial y_d(x, t)}{\partial t} = \sum_{q} S_{d,q} u_q(x, t) - \lambda_d y_d(x, t) + D_d \frac{\partial^2 y_d(x, t)}{\partial x^2}. \]

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 \( 0 < x < l \), 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]:

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

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

\[ \frac{2}{t} e^{-\lambda_d t} \sum_{n=1}^{\infty} \sin \left( \frac{n \pi x}{l} \right) \sin \left( \frac{n \pi \xi}{l} \right) e^{-\left( \frac{n \pi^2}{l^2} \right)} \exp \left( -\frac{(t - \tau)^2}{\ell_\tau^2} \right) \exp \left( -\frac{(x - \xi)^2}{\ell_x^2} \right), \]

where \( \ell_\tau \) represents the length-scale along the time-input dimension and \( \ell_x \) the length-scale along the space input dimension. The covariances \( k_{q,j,q'}(x, x') \) 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

\[ \frac{4}{\ell_x^2} \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} k_{l_x/q, l'_x/q'}(t, t') k_{l_x/q, l'_x/q'}(x, x'), \quad (13) \]

where \( l_x/q \) and \( l'_x/q' \) are also kernel functions that depend on the indexes \( n \) and \( m \). The kernel function \( k_{l_x/q, l'_x/q'}(t, t') \) is given by

\[ k_{l_x/q, l'_x/q'}(t, t') = \frac{\sqrt{\pi \ell_t}}{2} \left[ h_{x,t}(t', t) + h_{d,e}(t, t') \right], \quad (14) \]

where

\[ h_{x,t}(t', t) = \exp \left( \frac{t'}{\ell_t} \right) \exp \left( -\frac{t'}{\ell_t} \right) \exp \left( -\frac{t}{\ell} \right) \exp \left( -\frac{t}{\ell} \right) \exp \left( -\frac{t}{\ell} \right) \exp \left( -\frac{t}{\ell} \right), \]

where \( \text{erf}(x) \) is the real valued error function, \( \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-y^2) dy \), \( \beta_d = \lambda_d + D_d \omega_m^2 \), \( \beta_e = \lambda_e + D_e \omega_m^2 \), \( \omega_m = \ell_x/\ell_t \), \( \omega_m = \ell_x/\ell_t \), and \( \beta_m = \ell_x/\ell_t \).

The covariance \( k_{l_x/q, l'_x/q'}(x, x') \) is given by

\[ k_{l_x/q, l'_x/q'}(x, x') = C(n, m, \ell_x/q') \sin(\omega_m x) \sin(\omega_m x'). \]

The term \( C(n, m, \ell_x/q') \) represents a function that depends on the indexes \( n \) and \( m \) and on the length-scale of the space-input dimension. The expression for \( C(n, m, \ell_x/q') \) is

\[ C(n, m, \ell_x/q') = \int_0^1 \int_0^1 \sin(w_m \xi) \sin(w_m \xi') e^{-\left( \frac{(w_m \xi)^2}{\ell_x^2} \right)} dw_m \xi, \]

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 $C(n, m, \ell_q^p)$ is

$$
\mathcal{I} \left[ \ell_q^p \right] = \frac{\ell_q^p}{2} \left( R \left[ W(n, \ell_q^p) \right] - I \left[ W(n, \ell_q^p) \right] \right) + \frac{(\ell_q^p)^2}{2} e^{-\gamma_m^q} \cos(n\pi) - 1,
$$

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

$$
W(n, \ell_q^p) = w(jz_n^m) - e^{-\frac{(\ell_q^p)^2}{4}} e^{-\gamma_m^q} w(jz_n^m),
$$

being $z_n^m = \frac{\ell_q^p}{2}, z_n^{m-1} = \frac{\ell_q^p}{2}$, and $\gamma_m^q = j\omega_m^q$.

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

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

$$
\frac{\ell_q^p}{2} \mathcal{I} \left[ \ell_q^p \right] = \frac{1}{2} \left( R \left[ W(n, \ell_q^p) \right] - I \left[ W(n, \ell_q^p) \right] \right) + \frac{(\ell_q^p)^2}{2} e^{-\gamma_m^q} \cos(n\pi) - 1,
$$

where $\mathcal{R} \left[ \cdot \right]$ 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 simulation of multivariate codes (see [6], [27], [28]).

In 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.

### Table 2
**RMSE and $R^2$ for Protein Data Prediction**

<table>
<thead>
<tr>
<th>Gene</th>
<th>Method</th>
<th>RMSE (mean ± SD)</th>
<th>$R^2$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>giant</td>
<td>MTGP</td>
<td>26.56 ± 0.30</td>
<td>81.12 ± 0.01</td>
</tr>
<tr>
<td></td>
<td>DROS</td>
<td><strong>2.00 ± 0.35</strong></td>
<td><strong>99.78 ± 0.01</strong></td>
</tr>
<tr>
<td>knirps</td>
<td>MTGP</td>
<td>16.14 ± 8.44</td>
<td>91.18 ± 2.77</td>
</tr>
<tr>
<td></td>
<td>DROS</td>
<td><strong>3.01 ± 0.81</strong></td>
<td><strong>99.60 ± 0.01</strong></td>
</tr>
</tbody>
</table>

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 km² 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

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

where $p = 2$ is the dimension of $x$, the measured concentration is given by $y_d(x, t)$, $\kappa_{d,j}$ is the diffusion constant of output $d$ in direction $p$, and $\Phi(x, t)$ represents an external force, with $x = \{x_j\}_{j=1}^{p}$.

Assuming the domain $\mathbb{R}^p = (-\infty < x_j < \infty; j = 1, \ldots, p)$, and initial condition prescribed by the set of latent forces $u(x) = \sum_{q=1}^{Q} S_q u_q(x)$, at $t = 0$, the solution to the system [24] is then given by

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

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

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

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

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

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

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

where $T_{d,j} = 2\kappa_{d,j}$.

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

$$
y_d(x) = \sum_{q=1}^{Q} S_q \int_{\mathbb{R}^p} \tilde{G}_d(x, x') u_q(x') dx',
$$

15. For simplicity, we again omit the noise term $\delta_d(t)$. 
TABLE 3
RMSE for Pollutant Metal Prediction

<table>
<thead>
<tr>
<th>Method</th>
<th>Cadmium (Cd)</th>
<th>Cobalt (Co)</th>
<th>Copper (Cu)</th>
<th>Lead (Pb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND GP</td>
<td>0.8353 ± 0.0898</td>
<td>2.2997 ± 0.1388</td>
<td>18.9616 ± 3.4404</td>
<td>28.1768 ± 5.8005</td>
</tr>
<tr>
<td>MTGP (Q = 1)</td>
<td>0.7638 ± 0.1016</td>
<td>2.2892 ± 0.1792</td>
<td>14.4179 ± 2.7119</td>
<td>21.5861 ± 4.1888</td>
</tr>
<tr>
<td>HEATK (Q = 1)</td>
<td>0.6773 ± 0.0628</td>
<td>2.06 ± 0.0887</td>
<td>13.1788 ± 2.6446</td>
<td>17.9630 ± 2.9450</td>
</tr>
<tr>
<td>MTGP (Q = 2)</td>
<td>0.6960 ± 0.0832</td>
<td>2.1299 ± 0.1983</td>
<td>12.7340 ± 2.2104</td>
<td>17.9399 ± 1.9881</td>
</tr>
<tr>
<td>SLFM (Q = 2)</td>
<td>0.6941 ± 0.0834</td>
<td>2.172 ± 0.1204</td>
<td>12.8935 ± 2.6125</td>
<td>17.9024 ± 2.0966</td>
</tr>
<tr>
<td>HEATK (Q = 2)</td>
<td>0.6758 ± 0.0623</td>
<td>2.0345 ± 0.0943</td>
<td>12.5971 ± 2.4842</td>
<td>17.5671 ± 2.6076</td>
</tr>
</tbody>
</table>

TABLE 4
R² for Pollutant Metal Prediction

<table>
<thead>
<tr>
<th>Method</th>
<th>Cadmium (Cd)</th>
<th>Cobalt (Co)</th>
<th>Copper (Cu)</th>
<th>Lead (Pb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND GP</td>
<td>15.07 ± 7.43</td>
<td>57.81 ± 7.19</td>
<td>25.84 ± 7.54</td>
<td>23.48 ± 10.40</td>
</tr>
<tr>
<td>MTGP (Q = 1)</td>
<td>27.25 ± 5.89</td>
<td>58.45 ± 5.71</td>
<td>58.84 ± 8.35</td>
<td>56.85 ± 11.60</td>
</tr>
<tr>
<td>HEATK (Q = 1)</td>
<td>43.83 ± 8.71</td>
<td>66.19 ± 4.60</td>
<td>65.55 ± 8.21</td>
<td>71.45 ± 5.78</td>
</tr>
<tr>
<td>MTGP (Q = 2)</td>
<td>40.30 ± 5.17</td>
<td>64.13 ± 5.10</td>
<td>67.51 ± 8.36</td>
<td>69.70 ± 6.90</td>
</tr>
<tr>
<td>SLFM (Q = 2)</td>
<td>40.97 ± 5.15</td>
<td>62.49 ± 5.41</td>
<td>67.35 ± 8.29</td>
<td>70.21 ± 6.04</td>
</tr>
<tr>
<td>HEATK (Q = 2)</td>
<td>43.94 ± 6.56</td>
<td>67.17 ± 4.30</td>
<td>68.40 ± 6.48</td>
<td>70.55 ± 6.88</td>
</tr>
</tbody>
</table>

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

\[
\tilde{G}_d(x, x') = \frac{1}{(2\pi)^{d/2}} \exp \left[ -\frac{1}{2} (x - x')^T P_d(x - x') \right],
\]

where \( P_d \) is a precision matrix, with diagonal form and entries \( \{ p_{d,d} = \frac{1}{\ell_{d,t}} \}_{t=1}^T \).

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

\[
\frac{1}{(2\pi)^{d/2}} | P_{d,d}^{\theta} |^{1/2} \exp \left[ -\frac{1}{2} (x - x')^T (P_{d,d}^{\theta})^{-1} (x - x') \right],
\]

where \( P_{d,d}^{\theta} = P_d^{-1} + \sum_{q=1}^Q \Lambda_q^{-1} \), and \( \Lambda_q \) 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 \( Q = 2 \), 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
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 $\theta$ 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 $(u_d^{Q_1})$, some of which share the same covariance function $k_{u_u}(t, t')$. 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 $G(t, s)$, 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) could be applied for the purposes of fusing the information from multiple sensors (a similar setup to the latent force model).

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 $\{A_n\}_{n=0}^M$, 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 $\theta$ 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 $\theta_0$ and a set of initial conditions $y_{0}$, 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, $y(t) = \sum_{j=1}^J \beta_j \phi_j(t)$. 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, $D^m y(t) = \sum_{j=1}^J \beta_j D^m \phi_j(t)$. 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, $\bar{y}(t)$, of the differential equation $D^m_y(t)$ such that $\bar{y}(t) \sim N(D^m_0 y(t), \sigma^2_y)$. The solution of the differential equation $D^m_0 y(t)$ is assumed to follow a Gaussian process prior with covariance $k_{D^m_0 y_D} (t, t')$, where this covariance is obtained by taking $D^m_0$ derivatives of $k(t, t')$ with respect to $t$ and $D^m_0$ derivatives with respect to $t'$. The covariance $k(t, t')$ is freely chosen. An approximated solution $\bar{y}(t)$ can then be computed through the expansion $\bar{y}(t) = \sum_{n=1}^N \alpha_n k_{D^m_0 y_D} (t, t_n)$, where $\alpha_n$ is an element of the vector $(K_{D^m_0 y_D, D^m_0 y_D} + \sigma^2_y I_N)^{-1} \bar{y}$.
where $K_{D^y,D^y}$ is a matrix with entries $k_{D^y,D^y}(t_n, t_{n'})$, and $\hat{y}$ is a vector of noisy observations of $D^y_0 y(t)$. An important difference between this method and the latent force model is that we do not assume we have access to noisy observations for $D^y_0 y(t)$, 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, $T(x)$, and piezometric head, $\phi(x)$, 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 $T(x)$ and $\phi(x)$ 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 $T(x)$ and $\phi(x)$. For details, the reader is referred to [51, see chapter 9] and [53, pp. 637-643]. It turns out that 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 $T(x)$ and $\phi(x)$ 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], $y(t) = \sum_{i=0}^{\infty} \frac{y^{(i)}(0)}{i!} (t - a)^i$, 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 $y^{(i)}(a)$ as regression coefficients. The derivatives are assumed to follow a Gaussian process prior with a covariance function that is obtained as $k_{1\rightarrow j}(t, t')$, where the superscript $j$ indicates how many derivatives of $k(t, t')$ are taken with respect to $t$ and the superscript $i$ 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 \({y(t - t_{k-1})|_{t_{k-1}}}{^T}\), a common setup in the classical theory of systems identification [58]. The particular dependency $y(t) = g(\{y(t - t_{k-1})|_{t_{k-1}}\})$, 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 $u_{\theta}(t)$ 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 $u_{\theta}(t)$ governed by Gaussian processes with covariance function (14).

- **Switching Dynamical Latent Force Models.** A further extension of the LFM framework allows the parameter vector $\theta$ 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 CONSO-LIDER-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


