First Draft

Non-linear process convolutions for multi-output Gaussian process prediction through Volterra series

Author 1
Institution 1

Author 2
Institution 2

Author 3
Institution 3

Abstract

The paper introduces a non-linear version of the process convolution formalism for building covariance functions for multiple output Gaussian processes. The non-linearity is introduced via Volterra series, one series per each output. We provide closed-form expressions for the mean function and the covariance function of the approximated Gaussian process at the output of the Volterra series. The mean function and covariance function for the joint Gaussian process are derived using formulae for the product moments of Gaussian variables. We compare the performance of the non-linear model against the classical process convolution approach in one synthetic dataset and two real datasets.

1 INTRODUCTION

A multiple-output Gaussian process (MOGP) is a Gaussian process (GP) with a covariance function that accounts for dependencies between multiple and related outputs [Bonilla et al., 2008]. Having models that exploit such dependencies is particularly important when some of the outputs are expensive to measure and the other more inexpensive outputs can be used as surrogates of the expensive output to improve its prediction. A typical example comes from geostatistics, where the accuracy of predicting the concentration of heavy pollutant metals like Lead or Copper (expensive to measure), can be improved by including measurements of pH (less expensive to measure), as secondary variables [Goovaerts, 1997].

One of the challenges in multiple-output GPs is defining a cross-covariance function between outputs that lead to a valid covariance function for the joint GP. There is extensive literature looking at ways to build such types of cross-covariance functions [Alvarez et al., 2012]. One of these approaches is known as process convolutions for which each output is the convolution integral between a smoothing kernel and a latent random process. The approach was introduced by Barry and Ver Hoef [1996] to build covariance functions for single-output GPs, and later for multiple-outputs in Ver Hoef and Barry [1998] Higdon [2002]. The convolution integral linearly transforms the underlying latent process, which is usually assumed to be a Gaussian process. The output process is then a GP with a covariance equal convolution operators acting to modify the covariance function of the latent GP.

The main contribution in this paper is the introduction of non-linear version of the process convolution construction suitable both for single-output and multiple-output GPs. The non-linear model is developed from a Volterra series where the excitation function is a latent random process. The Volterra series have been widely studied in the literature of non-linear dynamical systems [Haber and Keviczky, 1999]. They generalise the Taylor expansion for the case of non-instantaneous input functions. We treat the latent process as a Gaussian process and using formulae for the product moments of Gaussian variables, we provide closed-form expressions for the mean function and covariance function of the output process, that we approximate as a Gaussian process once more using the mean + cov. of the output.

Most attempts to generate non-linear models that involve Gaussian processes come from the alternative representation of the convolution integral based on state space approaches [Hartikainen and Särkkä, 2012, Särkkä et al., 2013]. Exceptions include the works by Lawrence et al. [2007] and Titsias et al. [2009] where the non-linearity is an *static transformation* of the underlying latent GP. We review these ones and other approaches in the Related Work section. Section 4.

We compare the performance of the non-linear model against the classical process convolution approach in one synthetic dataset and two real datasets and show

Preliminary work. Under review by AISTATS 2019. Do not distribute.

that the non-linear version provides better performance than the traditional approach.

2 BACKGROUND

In these section, we briefly review the MOGP with process convolutions. Sometimes we will refer to this particular construction as the convolved MOGP (CMOGP). We also briefly review the Volterra series and the formulae for the product moments of Gaussian variables that we use in a later section.

2.1 Process convolutions for multi-output Gaussian processes

We want to jointly model the set of processes $[f_d(t)]_{d=1}^D$ using a joint Gaussian process. In the process convolution approach to building such Gaussian process, dependencies between different ouputs are introduced by assuming that each output $f_d(t)$ is the convolution integral between a smoothing kernel $G_d(t)$ and a latent function $u(t)$,

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

where the smoothing kernel $G_d(t-\tau)$ should have finite energy. Assuming that the latent process $u(t)$ is a GP with zero mean function and covariance function $k(t,t')$, the set of processes $f_1(t), f_2(t), \cdots, f_D(t)$ are jointly Gaussian with zero mean function and cross-covariance function between $f_d(t)$ and $f_{d'}(t')$ given by

$$k_{d,d'}(t,t') = \mathrm{cov}[f_d(t), f_{d'}(t')] = \int\int G_d(t-\tau_i)G_{d'}(t'-\tau_i)k(\tau_i,\tau_j)\mathrm{d}\tau_i\mathrm{d}\tau_j. \tag{2}$$

The authors in Alvarez et al. [2012] have shown that the covariance function above generalises the well-known linear model of coregionalization, a form of covariance function for multiple outputs commonly used in machine learning and geostatistics.

The expression for $f_d(t)$ in the form of the convolution integral in Eq. (1) is also the representation of a linear dynamical system with impulse response $G_d(t)$. In the context of latent force models, such convolution expressions have been used to compute covariance functions informed by physical systems where the smoothing kernel is related to the so-called Green's function representation of an ordinary differential operator [Álvarez et al., 2013].

2.2 Representing non-linear dynamics with Volterra series

The Taylor series expansion of a time-invariant non-linear system is the polynomial series about a fixed working point:

$$f(t) = \sum_{i=0}^{\infty} c_i u^i(t) = c_0 + c_1 u(t) + c_2 u^2(t) + \dots$$

While the Taylor series is widely used in the approximation of non-linear systems, it can only approximate systems for which input is instantaneous [Haber and Keviczky, 1999].

By the Stone-Weierstraß theorem, a given continuous non-linear system with finite-dimensional vector input can be uniformly appoximated by a finite polynomial series [Gallman and Narendra, 1976]. The Volterra series is such a polynomial expansion, describing a series of nested convolution operators:

$$\begin{aligned} f(t) &= \sum_{c=0}^{\infty} \int \dots \int G^{(c)}(t-\tau_1, \dots, t-\tau_u) \prod_{j=1}^{c} u(\tau_j)\mathrm{d}\tau_j \\ &= G^{(0)} + \int G^{(1)}(t-\tau_1)u(\tau_1)\mathrm{d}\tau_1 \\ &+ \iint G^{(2)}(t-\tau_1, t-\tau_2)u(\tau_1)u(\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2 \\ &+ \dots \end{aligned}$$

The leading term $G^{(0)}$ is a constant term, which in practise is assumed to be zero-valued and the series is incremented from $c = 1$. Because of the convolutions involved, the series is no longer modelling instantaneous input at t, giving the series a so-called *memory effect* [Haber and Keviczky, 1999].

As with the Taylor series, the approximant needs a cut-off for the infinite sum, denoted C – a Volterra series with C sum terms is called C-order. The representation of a c^{th} degree kernel, i.e. $G^{(c)}(t-\tau_1, \dots, t-\tau_c)$, can be expressed in different forms, such as in symmetric or triangular form [Haber and Keviczky, 1999]. A common assumption is that the kernels are homogeneous and seperable, such that $G^{(c)}$ is the product of c first degree kernels. The assumption of seperability requires a stronger assumption but reduces the number of unique parameters, which can be very large for a full Volterra series [Schetzen, 1980]. It should also be noted that a truncated Volterra series with seperable homogeneous kernels is equivalent to a Weiner model [Cheng et al., 2017].

2.3 Product moments for multivariate Gaussian random variables

Several of the results that we will present in the following section involve the computation of the expected value of the product of several multivariate Gaussian random variables. For this, we will use results derived in Song and Lee [2015]. We are interested in those

results for which the Gaussian random variables have zero mean. In particular, let $\{X_i\}_{i=1}^c$ be multivariate Gaussian random variables with zero mean values and covariance between X_i and X_j given as ϕ_{ij}. According to Corollary 1 in Song and Lee [2015], the expression for $\mathbb{E}\left[\prod_{i=1}^c X_i^{a_i}\right]$, follows as

$$\mathbb{E}\left[\prod_{i=1}^{c} X_i^{a_i}\right] = \sum_{\mathbf{L}\in T_{\mathbf{a}}} \frac{\prod_{k=1}^{c} a_k!}{2^{\mathrm{tr}[\mathbf{L}]} \prod_{i=1}^{c}\prod_{j=1}^{c} l_{ij}!} \prod_{i=1}^{c}\prod_{j=i}^{c} (\phi_{ij})^{l_{ij}}, \tag{3}$$

where $\mathbf{a} = [a_i]_{i=1}^c$ is a vector consisting of the random variable exponents and $T_{\mathbf{a}}$ is the set of $c \times c$ *symmetric* matrices that meet the condition $L_{\mathbf{a},k} = 0$ for $k = 1, \ldots, c$, as defined by

$$T_{\mathbf{a}} = \left\{ [l_{ij}]_{c\times c} \;\middle|\; \underbrace{a_k - l_{kk} - \sum_{j=1}^{c} l_{jk}}_{L_{\mathbf{a},k}} = 0,\ \forall k \in \{1,\ldots,c\} \right\}. \tag{4}$$

If the sum of exponents, $\sum_{k=1}^c a_k$, is an odd number, then $\mathbb{E}\left\{\prod_{i=1}^c X_i^{a_i}\right\} = 0$ for the zero mean value case (see Corollary 2 in Song and Lee [2015]).

An additional result that we will use later is that if $a_k = 1$, $\forall a_k$ then (3) reduces to (see Remark 5 in Song and Lee [2015]),

$$\mathbb{E}\left[\prod_{i=1}^{c} X_i\right] = \sum_{\mathbf{L}\in T_{\mathbf{a}}} \prod_{i=1}^{c}\prod_{j=i}^{c} \phi_{ij}^{l_{ij}}. \tag{5}$$

3 A NON-LINEAR CMOGP BASED ON VOLTERRA SERIES

We represent a vector-valued non-linear dynamic system with a system of Volterra series of order C. For a given output dimension, d, we approximate the function

$$f_d^{(C)}(t) = \sum_{c=1}^{C} \int \cdots \int G_d^{(c)}(t-\tau_1, \ldots, t-\tau_c) \prod_{j=1}^{c} u(\tau_j) \mathrm{d}\tau_j, \tag{6}$$

where $G_d^{(c)}$ are c^{th} degree Volterra kernels.

Our approach is to assume that $u(t)$, the latent driving function, follows a GP prior. For $C = 1$, we recover the the expression for the process convolution construction of a multi-output GP as defined in (1). In contrast to the linear case, the output $f_d^{(C)}$ is no longer a GP. However, we approximate $f_d^{(C)}$ by a Gaussian process $\tilde{f}_d^{(C)}(t)$ with a mean and covariance function computed from the moments of the output process $f_d^{(C)}$:

$$\tilde{f}_d^{(C)}(t) \sim \mathcal{GP}(\mu_d^{(C)}(t), k_{f_d f_{d'}}^{(C)}(t,t')), \tag{7}$$

where $\mu_d^{(C)}(t) = \mathbb{E}[f_d^{(C)}(t)]$ and $k_{f_d f_{d'}}^{(C)}(t,t') = \mathrm{cov}[f_d^{(C)}(t), f_{d'}^{(C)}(t')]$. Approximating a non-Gaussian distribution with a Gaussian, particularly for non-linear systems, is common in state space modelling, for example in the unscented Kalman filter [Särkkä, 2013]; or as a choice of variational distribution in variational inference [Blei et al., 2017]. We refer to the joint process $\{\tilde{f}_d^{(C)}(t)\}_{d=1}^D$ as the *non-linear convolved multi-output GP* (NCMOGP).

Furthermore, we will assume that the c^{th} degree Volterra kernels are separable, such that

$$G_d^{(c)}(t-\tau_1, \ldots, t-\tau_c) = \prod_{i=1}^{c} G_d^{(c,i)}(t-\tau_i),$$

where $G_d^{(c,i)}$ are first degree Volterra kernels.

Using this separable form, we express the output $f_d^{(C)}(t)$ as

$$\sum_{c=1}^{C}\prod_{i=1}^{c}\int_{\tau_i} G_d^{(c,i)}(t-\tau_i)u(\tau_i)\mathrm{d}\tau_i = \sum_{c=1}^{C}\prod_{i=1}^{c} f_d^{(c,i)}(t),$$

where we define

$$f_d^{(c,i)}(t) = \int_{\tau_i} G_d^{(c,i)}(t-\tau_i)u(\tau_i)\mathrm{d}\tau_i.$$

Assuming $u(t)$ has a GP prior with zero mean and covariance $k(t,t')$, and due to the linearity of the expression above, we can compute the corresponding mean and covariance functions for the joint Gaussian process $[f_d^{(c,i)}(t)]_{d=1}^D$. We compute the cross-covariance function between $f_d^{(c,i)}(t)$ and $f_{d'}^{(c',j)}(t')$ using

$$k_{d,d'}^{(c,i),(c',j)}(t,t') = \mathrm{cov}[f_d^{(c,i)}(t), f_{d'}^{(c,j)}(t')] = \int\int G_d^{(c,i)}(t-\tau_i) G_{d'}^{(c',j)}(t'-\tau_i) k(\tau_i,\tau_j)\mathrm{d}\tau_i\mathrm{d}\tau_j. \tag{8}$$

This is a similar expression to the one in (2) for the CMOGP. For some particular forms of the Volterra kernels $G_d^{(c,i)}$ and covariance $k(t,t')$ of the latent process $u(t)$, the covariance $k_{d,d'}^{(c,i),(c',j)}(t,t')$ can be computed analytically.

3.1 NCMOGP with separable Volterra kernels

3.1.1 Mean function

Let us first compute the mean function $\mathbb{E}[f_d^{(C)}(t)]$. Using the definition for the expected value, we get

$$\mathbb{E}[f_d^{(C)}(t)] = \sum_{c=1}^{C} \mathbb{E}\left[\prod_{i=1}^{c} f_d^{(c,i)}(t)\right]. \tag{9}$$

The expected value of the product of the Gaussian processes, $\mathbb{E}[\prod_{i=1}^c f_d^{(c,i)}(t)]$, can be computed using results obtained for the expected value of the product of multivariate Gaussian random variables as introduced in Section 2.3.

Applying the result in (5) to the expression of the expected value in (9), we get

$$\mathbb{E}\left[\prod_{i=1}^{c} f_d^{(c,i)}(t)\right] = \sum_{\mathbf{L}\in T_{\mathbf{a}}} \prod_{i=1}^{c}\prod_{j=i}^{c} \left(k_{d,d}^{(c,i),(c,j)}(t,t)\right)^{l_{ij}}, \tag{10}$$

where

$$k_{d,d}^{(c,i),(c,j)}(t,t) = \text{cov}[f_d^{(c,i)}(t), f_d^{(c,j)}(t)].$$

Notice that only the terms for which c is even are different from zero.

Example 1 To see an example of the kind of expressions that the expected value takes, let us assume that $c = 4$. We then get

$$\mathbb{E}\left[\prod_{i=1}^{4} f_d^{(4,i)}(t)\right] = \sum_{\mathbf{L}\in T_\mathbf{a}} \prod_{i=1}^{4}\prod_{j=i}^{4} \left(k_{d,d}^{(4,i),(4,j)}(t,t)\right)^{l_{ij}},$$

where

$$k_{d,d}^{(4,i),(4,j)}(t,t) = \text{cov}[f_d^{(4,i)}(t), f_d^{(4,j)}(t)].$$

We now need to find the set $T_\mathbf{a}$ containing all $c \times c$ symmetric matrices $\mathbf{L}$, the elements of which meet the conditon described in (4), where $\mathbf{a} = [1\ 1\ 1\ 1]$. This leads to the following system of equations

$$\begin{aligned} 2l_{11} + l_{12} + l_{13} + l_{14} &= 1 \\ l_{12} + 2l_{22} + l_{23} + l_{24} &= 1 \\ l_{13} + l_{23} + 2l_{33} + l_{34} &= 1 \\ l_{14} + l_{24} + l_{34} + 2l_{44} &= 1, \end{aligned}$$

where we have used the symmetry of $\mathbf{L}$, so $l_{ij} = l_{ji}$. It can be seen from the above system that the set $T_\mathbf{a}$ contains three unique symmetric matrices:

$$T_\mathbf{a} = \left\{ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \right\}.$$

We now have an expression for the expected value, by (10):

$$\begin{aligned} \mathbb{E}\left[\prod_{i=1}^{4} f_d^{(4,i)}(t)\right] &= k_{d,d}^{(4,1),(4,2)}(t,t)k_{d,d}^{(4,3),(4,4)}(t,t) \\ &+ k_{d,d}^{(4,1),(4,3)}(t,t)k_{d,d}^{(4,2),(4,4)}(t,t) \\ &+ k_{d,d}^{(4,1),(4,4)}(t,t)k_{d,d}^{(4,2),(4,3)}(t,t). \end{aligned}$$

3.1.2 Cross-covariance function

For computing the covariance function, we first need to compute the second moment between $f_d^{(C)}(t)$ and $f_{d'}^{(C)}(t')$. The second moment is given as

$$\begin{aligned} &\mathbb{E}\left[f_d^{(C)}(t) f_{d'}^{(C)}(t')\right] \\ &= \mathbb{E}\left[\sum_{c=1}^{C}\prod_{i=1}^{c} f_d^{(c,i)}(t) \sum_{c'=1}^{C}\prod_{j=1}^{c'} f_{d'}^{(c',j)}(t')\right] \\ &= \sum_{c=1}^{C}\sum_{c'=1}^{C} \mathbb{E}\left[\prod_{i=1}^{c}\prod_{j=1}^{c'} f_d^{(c,i)}(t) f_{d'}^{(c',j)}(t')\right] \\ &= \sum_{c=1}^{C}\sum_{c'=1}^{C} \mathbb{E}\left[\prod_{i=1}^{c+c'} \bar{f}_{d;i}(t)\right], \end{aligned} \tag{11}$$

where $\bar{f}_{d;i}$ is the i^{th} output of a vector-valued function consisting of all functions in the product

$$\bar{\mathbf{f}}_d(t) = \left[f_d^{(c,1)}(t)\ \ldots\ f_d^{(c,c)}(t)\ f_{d'}^{(c',1)}(t')\ \ldots\ f_{d'}^{(c',c')}(t')\right]^\top.$$

We have assumed that both $f_d^{(C)}(t)$ and $f_{d'}^{(C)}(t')$ share the same value of C, although a more general expression can be obtained where each output can have its own C value. We can apply the expressions in Song and Lee [2015] to the above moment of the product of Gaussian random variables as we did for computing the mean function in Section 3.1.1. Using the expression for the expected value of the product of Gaussian variables, we get

$$\mathbb{E}\left[\prod_{i=1}^{c+c'} \bar{f}_{d;i}(t)\right] = \sum_{\mathbf{L}\in T_\mathbf{a}} \prod_{i=1}^{c+c'}\prod_{j=i}^{c+c'} \left(\text{cov}\left[\bar{f}_{d;i}(t), \bar{f}_{d;j}(t)\right]\right)^{l_{ij}},$$

where the covariance element is defined in (8) as the cross-covariance of two latent functions.

Example 2 For illustration purposes, let us assume that $c = 3$ and $c' = 1$. In this case, we would have

$$\begin{aligned} &\mathbb{E}\left[\prod_{i=1}^{3} f_d^{(3,i)}(t) f_{d'}^{(1,1)}(t')\right] \\ &= \sum_{\mathbf{L}\in T_\mathbf{a}} \prod_{i=1}^{4}\prod_{j=i}^{4} \left(\text{cov}\left[\bar{f}_{d;i}(t), \bar{f}_{d;j}(t)\right]\right)^{l_{ij}}, \end{aligned}$$

where we have defined $\bar{\mathbf{f}}_d(t)$ as

$$\bar{\mathbf{f}}_d(t) = \left[f_d^{(3,1)}(t)\quad f_d^{(3,2)}(t)\quad f_d^{(3,3)}(t)\quad f_{d'}^{(1,1)}(t')\right]^\top.$$

The set $T_\mathbf{a}$ contains the same matrices as we found in Example 1 in Section 3.1.1 leading to

$$\begin{aligned} &\mathbb{E}\left[\prod_{i=1}^{3} f_d^{(c,i)}(t) f_{d'}^{(1,1)}(t')\right] \\ &= k_{d,d}^{(3,1),(3,2)}(t,t)k_{d,d'}^{(3,3),(1,1)}(t,t') \\ &+ k_{d,d}^{(3,1),(3,3)}(t,t)k_{d,d'}^{(3,2),(1,1)}(t,t') \\ &+ k_{d,d'}^{(3,1),(1,1)}(t,t')k_{d,d}^{(3,2),(3,3)}(t,t). \end{aligned}$$

The cross-covariance function between $f_d^{(C)}(t)$ and $f_{d'}^{(C)}(t')$ is then computed using

$$\begin{aligned} &\text{cov}[f_d^{(C)}(t), f_{d'}^{(C)}(t')] \\ &= \mathbb{E}[f_d^{(C)}(t) f_{d'}^{(C)}(t')] - \mathbb{E}[f_d^{(C)}(t)]\,\mathbb{E}[f_{d'}^{(C)}(t')]. \end{aligned}$$

We have expressions for both $\mathbb{E}[f_d^{(C)}(t) f_{d'}^{(C)}(t')]$ and $\mathbb{E}[f_d^{(C)}(t)]$ by (9) and (11) respectively.

3.2 NCMOGP with separable and homogeneous Volterra kernels

In the section above, we introduced a model that allows for different first-order Volterra kernels $G_d^{(c,i)}(x)$, when building the general Volterra kernel of order c. A further assumption

would be to set all first-order Volterra kernels to be the same, this is,

$$G_d^{(c,i)}(t-\tau_i) = G_d(t-\tau), \quad \forall c,\ \forall i,$$

meaning that $f_d^{(c,i)}(t) = f_d(t)$, for all c and for all i. We will refer to this as the separable and homogeneous version of the NCMOGP.

We then get

$$f_d^{(C)}(t) = \sum_{c=1}^{C}\prod_{i=1}^{c} f_d^{(c,i)}(t) = \sum_{c=1}^{C}\prod_{i=1}^{c} f_d(t) = \sum_{c=1}^{C} f_d^c(t),$$

where

$$f_d(t) = \int G_d(t-\tau)u(\tau)\mathrm{d}\tau,$$

and $u(t) \sim \mathcal{GP}(0, k(t,t'))$. The cross-covariance function between $f_d(t)$ and $f_{d'}(t')$ is again $k_{d,d'}(t,t')$ as in (??).

As we did in Section 3.1, we can compute the mean function for $f_d^{(C)}(t)$ and cross-covariance functions between $f_d^{(C)}(t)$ and $f_{d'}^{(C)}(t')$.

3.3 Mean function

Using expression (3), the mean function $\mathbb{E}[f_d^{(C)}(t)]$ follows as

$$\mathbb{E}[f_d^{(C)}(t)] = \mathbb{E}\left[\sum_{c=1}^{C} f_d^c(t)\right] = \sum_{c=1}^{C}\mathbb{E}\left[f_d^c(t)\right]$$
$$= \sum_{c=1}^{C}\frac{c!}{2^{l_{11}}l_{11}!}(k_{d,d}(t,t))^{l_{11}},$$

where l_{11} is such that $2l_{11} = c$. This means that the above expression only has solutions for c even, and $l_{11} = c/2$. Then

$$\mathbb{E}[f_d^{(C)}(t)] = \sum_{c=1}^{C}\frac{c!}{2^{c/2}\left(\frac{c}{2}\right)!}(k_{d,d}(t,t))^{c/2},$$

for c even and $C \geq 2$.

3.4 Cross-covariance function

We can compute the second moment $\mathbb{E}[f_d^{(C)}(t)f_{d'}^{(C)}(t')]$ using

$$\mathbb{E}[f_d^{(C)}(t)f_{d'}^{(C)}(t')] = \mathbb{E}\left[\sum_{c=1}^{C} f_d^c(t)\sum_{c'=1}^{C} f_{d'}^{c'}(t')\right]$$
$$= \sum_{c=1}^{C}\sum_{c'=1}^{C}\mathbb{E}\left[f_d^c(t)f_{d'}^{c'}(t')\right].$$

Once again we use expression (3) to compute $\mathbb{E}\left[f_d^c(t)f_{d'}^{c'}(t')\right]$, leading to

$$\mathbb{E}\left[f_d^c(t)f_{d'}^{c'}(t')\right] =$$
$$\sum_{\mathbf{L}\in T_{\mathbf{a}}} A_{c,c',\mathbf{L}}(k_{d,d}(t,t))^{l_{11}}(k_{d,d'}(t,t'))^{l_{12}}(k_{d',d'}(t',t'))^{l_{22}},$$

where $A_{c,c',\mathbf{L}}$ is defined as

$$A_{c,c',\mathbf{L}} = \frac{c!c'!}{2^{l_{11}+l_{22}}l_{11}!l_{12}!l_{22}!}$$

and $k_{d,d'}(t,t')$ is defined in (??). For avoiding computer overflow due to the factorial operators when computing $A_{c,c',\mathbf{L}}$, we compute $\exp(\log(A_{c,c',\mathbf{L}}))$ instead.

As stated previously, the expected value will be 0 if $c+c'$ is not even.

Example 3 Let as assume that $c = 3$ and $c' = 3$. The expected value $\mathbb{E}\left[f_d^3(t)f_{d'}^3(t')\right]$ follows as

$$\sum_{\mathbf{L}\in T_{\mathbf{a}}} A_{3,3,\mathbf{L}}(k_{d,d}(t,t))^{l_{11}}(k_{d,d'}(t,t'))^{l_{12}}(k_{d',d'}(t',t'))^{l_{22}},$$

where $A_{3,3,\mathbf{L}} = (3!3!)/2^{l_{11}+l_{22}}l_{11}!l_{12}!l_{22}!$. To find the elements in $\mathbf{L} \in T_{\mathbf{a}}$, we need to solve similar equations to the ones in Example 1. We would have

$$2l_{11} + l_{12} = c = 3$$
$$l_{12} + 2l_{22} = c' = 3.$$

We can see that there are two valid solutions for $\mathbf{L}$:

$$T_{\mathbf{a}} = \left\{\begin{bmatrix}0 & 3\\ 3 & 0\end{bmatrix}, \begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}\right\}.$$

The expression for $\mathbb{E}\left[f_d^3(t)f_{d'}^3(t')\right]$ is thus

$$6k_{d,d'}^3(t,t') + 9k_{d,d}(t,t)k_{d,d'}(t,t')k_{d',d'}(t',t').$$

Again, we compute the covariance $\mathrm{cov}[f_d^{(C)}(t), f_{d'}^{(C)}(t')]$ now that we have expressions to compute the mean for $f_d^{(C)}(t)$ and the expected value for the product between $f_d^{(C)}(t)$ and $f_{d'}^{(C)}(t')$.

4 RELATED WORK

In the works by Lawrence et al. [2007], non-linear dynamics are introduced with a LFM prior within a non-linear function, which are inferred using the Laplace approximation, with the convolution operator itself approximated as a discrete sum. Similarly, Titsias et al. [2009] approximate the posterior to the non-linear LFM using an MCMC sampling approach. Approaches to non-linear likelihoods with GP priors, not limited to LFMs, include the warped GP model [Lázaro-Gredilla, 2012] and chained GPs [Saul et al., 2016] which make use of variational approximations. Techniques from state space modelling, including Taylor series linearisation and sigma points used to approximate Gaussians in the extended and unscented Kalman filters respectively have been applied to non-linear Gaussian processes, both for single output and multitask learning [Steinberg and Bonilla, 2014, Bonilla et al., 2016].

An alternative perspective to the linear LFM model is to construct it as a continuous-discrete state space model (SSM) driven by white noise [Hartikainen and Särkkä, 2012, Särkkä et al., 2013]. Hartikainen et al. [2012] use this approach for the general case non-linear Weiner system,

approximating the posterior with an unscented Kalman filter and Rauch-Tung-Streibel smoother. The SSM approach benefits from inference being performed in linear time, but relies on certain constraints on the nature of the underlying covariance functions. In particular, a kernel must have a rational power spectrum to be used in exact form, which precludes the use of, for example, the Gaussian RBF kernel for exact Gaussian process regression without introducing additional approximation error [Särkkä et al., 2013]. ? also use a state space representation to approximate Hammerstein-Weiner models, albeit with sequential Monte Carlo and a maximum-likelihood approach.

5 IMPLEMENTATION

Multi-output regression with NCMOGP In this paper, we are interested in the multi-output regression case. Therefore, we restrict the likelihood models for each output to be Gaussian. In particular, we assume that each observed output $y_d(t)$ follows

$$y_d(t) = f_d^{(C)}(t) + w_d(t),$$

where $w_d(t)$ is a white Gaussian noise process with covariance function $\sigma_d^2 \delta_{t,t'}$. Other types of likelihoods are possible as for example in Moreno-Muñoz et al. [2018].

Kernel functions In all the experiments, we use an squared-exponentiated quadratic (SEQ) form for the smoothing kernels $G_d(t-\tau)$ and an SEQ form for the kernel of the latent functions $u(t)$. With these forms, the kernel $k_{d,d'}(t,t')$ also follows an SEQ form. We use the expressions for $k_{d,d'}(t,t')$ obtained by Álvarez and Lawrence [2011].

High-dimensional inputs The resulting mean function $\mathbb{E}[f_d^{(C)}(t)]$ and covariance function $\text{cov}[f_d^{(C)}(t), f_{d'}^{(C)}(t')]$ assume that the input space is one-dimensional. We can extend the approach to high-dimensional inputs, $\mathbf{x} \in \mathbb{R}^p$ by assuming that both the mean function and covariance function factorise across the input dimension, and using the same expressions for the kernels for each factorised dimension.

Hyperparameter learning We optimise the log-marginal likelihood for finding point estimates of the hyperparameters $\boldsymbol{\theta}$ of the NCMOGP. Hyperparameters include the parameters for the smothing kernels $G_d(\cdot)$, the kernel function $k(t,t')$ and the variances for the white noise processes $w_d(t)$, σ_d^2. For simplicity in the notation, we assume that all the outputs are evaluated at the same set of inputs $\mathbf{t} = \{t_n\}_{n=1}^N$. Let $\mathbf{y} = [\mathbf{y}_1^\top, \cdots, \mathbf{y}_D^\top]^\top$, with $\mathbf{y}_d = [y_d(t_1), \cdot, y_D(t_N)]^\top$. The log-marginal likelihood $\log p(\mathbf{y}|\mathbf{t})$ is then given as

$$-\frac{ND}{2}\log(2\pi) - \frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}^{(C)})^\top (\mathbf{K}_{\mathbf{f}^{(C)},\mathbf{f}^{(C)}} + \boldsymbol{\Sigma})^{-1} \times (\mathbf{y}-\boldsymbol{\mu}^{(C)}) - \frac{1}{2}\log\left|\mathbf{K}_{\mathbf{f}^{(C)},\mathbf{f}^{(C)}} + \boldsymbol{\Sigma}\right|,$$

where $\boldsymbol{\mu}^{(C)}$ has entries given by $\mu_d^{(C)}(t)$, $\mathbf{K}_{\mathbf{f}^{(C)},\mathbf{f}^{(C)}} \in \mathbb{R}^{ND\times ND}$ has entries computed using $k_{f_d^{(C)},f_{d'}^{(C)}}(t,t')$ and $\boldsymbol{\Sigma}$ is a diagonal matrix containing the variances of the noise processes per output. As it is usual, we use a gradient-based optimization procedure to estimate the hyperparameters that maximize the log-marginal likelihood.

Predictive distribution The predictive distribution is the same one used for the single-output case. Let $\mathbf{t}_* = \{t_{n,*}\}_{n=1}^{N_*}$ be the input test set. The predictive distribution follows as $p(\mathbf{y}_*|\mathbf{y}) = \mathcal{N}(\mathbf{y}_*|\boldsymbol{\mu}_{\mathbf{y}_*|\mathbf{y}}, \mathbf{K}_{\mathbf{y}_*|\mathbf{y}})$, with $\boldsymbol{\mu}_{\mathbf{y}_*|\mathbf{y}} = \boldsymbol{\mu}_*^{(C)} + \mathbf{K}_{\mathbf{f}_*^{(C)},\mathbf{f}^{(C)}}\widetilde{\mathbf{K}}_{\mathbf{f}^{(C)},\mathbf{f}^{(C)}}^{-1}\boldsymbol{\mu}^{(C)}$ and $\mathbf{K}_{\mathbf{y}_*|\mathbf{y}} = \mathbf{K}_{\mathbf{f}_*^{(C)},\mathbf{f}_*^{(C)}} - \mathbf{K}_{\mathbf{f}_*^{(C)},\mathbf{f}^{(C)}}\widetilde{\mathbf{K}}_{\mathbf{f}^{(C)},\mathbf{f}^{(C)}}^{-1}\mathbf{K}_{\mathbf{f}_*^{(C)},\mathbf{f}^{(C)}}^\top + \boldsymbol{\Sigma}_*$, where $\widetilde{\mathbf{K}}_{\mathbf{f}^{(C)},\mathbf{f}^{(C)}} = \mathbf{K}_{\mathbf{f}^{(C)},\mathbf{f}^{(C)}} + \boldsymbol{\Sigma}$. In these expressions, $\boldsymbol{\mu}_*^{(C)}$ has entries given by $\mu^{(C)}(t_{n,*})$; $\mathbf{K}_{\mathbf{f}_*^{(C)},\mathbf{f}_*^{(C)}}$ has entries given by $k_{f_d^{(C)},f_{d'}^{(C)}}(t_{n,*},t_{m,*})$; and $\mathbf{K}_{\mathbf{f}_*^{(C)},\mathbf{f}^{(C)}}$ has entries given by $k_{f_d^{(C)},f_{d'}^{(C)}}(t_{n,*},t_m)$.

6 EXPERIMENTAL RESULTS

Experimental results are provided for the NCMOGP with homogeneous and separable kernels. In all the experiments that follow, hyperparameter estimation is done through maximization of the log-marginal likelihood as explained in Section 5. We use the Normalised Mean Squared Error (NMSE) and the Negative Log-Predictive Density (NLPD) to assess the performance.

6.1 Toy example

We set a problem of $D = 3$ outputs where the smoothing kernels are given as $G_d(t-\tau) = S_d \exp(-P_d(t-\tau)^2)$, with parameters $S_1 = 5, S_2 = 1, S_3 = 2$, and $P_1 = 200, P_2 = 0.1$ and $P_3 = 100$. The latent function follows as $u(t) = \sum_{i=1}^4 \left(\frac{1}{i^2}\right)\cos(2it)$. We then numerically solve the convolution integral $f_d(t) = \int_0^t G_d(t-\tau)u(\tau)d\tau$ for 200 datapoints in the input range $[0,1]$. We compute $f_d^{(C=4)}(t) = \sum_{c=1}^4 f_d^c(t)$ and the observed data is obtained by adding Gaussian noise with a variance of $\sigma_d^2 = 0.005 \times \text{var}\left[f_d^{(C=3)}(t)\right]$. We randomly split the dataset into a train set of $N = 50$ per output and the rest of the datapoints are used for assesing the performance. Table 3

Table 1: Results for the Toy example.

C	NMSE	NLPD
1	0.0142 ± 0.0022	-2.6868 ± 0.0794
2	0.0075 ± 0.0012	-2.9641 ± 0.0665
3	$\mathbf{0.0071 \pm 0.0009}$	$\mathbf{-2.9780 \pm 0.0513}$
4	0.0072 ± 0.0010	-2.9546 ± 0.0960
5	0.0071 ± 0.0011	-2.9428 ± 0.0956

shows the NMSE and the NLPD for different values of C for twenty different partitions of the original dataset into training and testing. We show the average of the metric plus-minus a standard deviation. The performance is similar for the non-linear models with $C \geq 3$, although the model for $C = 3$ shows the lowest standard deviation.

6.2 Weather data

We use the air temperature dataset considered previously by Nguyen and Bonilla [2014] and used in other multiple outputs GP papers. The dataset contains air temperature measurements at four spatial locations in the south coast of

England: Bramble Bank, Southhampton Dockhead, Chichester Harbour and Chichester Bar, usually refer to by the names Bramblemet, Sotonmet, Cambermet and Chimet, respectively. The measurements correspond to July 10 to July 15, 2013. The prediction problem as introduced in Nguyen and Bonilla [2014] corresponds to predict consecutive missing data in the outputs Cambermet and Chimet, 173 and 201 observations, respectively, using observed data from all the stations, 1425 observations for Bramblemet, 1268 observations for Cambermet, 1235 for Chimet and 1097 for Sotonmet. The missing data was artificially removed and we have access to the ground-truth measurements.

Table 2: Results for the Weather dataset.

C	NMSE	NLPD
1	0.6581	1.9235
2	**0.4795**	**1.6671**
3	1.2489	2.4843
4	1.1176	2.1692
5	0.6009	2.1257

Table 2 reports the mean predictive performance for five random initialisations of the models for the missing observations. We averaged over the two outputs with missing data. There is a reduction in the NMSE and an improvement in the NLPD for the non-linear models with $C = 2$ and $C = 5$ compared to the linear model.

6.3 A high-dimensional input example

The NCMOGP can also be applied for datasets with an input dimension greater than one. We use a subset of the SARCOS dataset[1] for illustration purposes. The prediction problem corresponds to map from positions, velocities and accelerations to the joint torques in seven degrees-of-freedom SARCOS anthropomorphic robot arm. This is, the datasets contains $D = 7$ outputs and the dimension of the input space is $p = 21$, corresponding to seven positions, seven velocities, and seven accelerations. The kernels used follow the idea described in Section 5 for higher-dimensional inputs.

Our setup is as follows: from the file `sarcos_inv.mat` we randonly select $N = 500$ for each output for training and from the file `sarcos_inv_test.mat`, we randomly select another 500 observations for testing. And we repeat the experiment ten times for different training and testing sets taken from the same two files.

Table 3: Results for a subset of the SARCOS dataset.

C	NMSE	NLPD
1	0.0497 ± 0.0252	1.4292 ± 1.1080
2	$\mathbf{0.0478 \pm 0.0238}$	$\mathbf{1.4067 \pm 1.1032}$
3	0.0571 ± 0.0377	1.4164 ± 1.1401
4	0.0720 ± 0.0855	1.4294 ± 1.0787
5	0.0830 ± 0.0809	1.4674 ± 1.1449

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

Table 3 shows the averaged NMSE and averaged NLPD for the ten repetitions plus-minus a standard deviation. As for the Weather dataset, the non-linear model yields better averaged performance than the linear model, even for $C = 2$. However, when looking at the predictive performance we notice that each outputs is usually better predicted by different values of C. For example, in terms of NLPD, the best order to predict outputs $d = 1, 2, 3, 7$ would be a non-linear model with $C = 4$. The best model for predicting outputs $d = 4, 6$ would be $C = 3$, and the best model for predicting output $d = 5$ would be $C = 5$.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a non-linear extension of the process convolution formalism to build multiple-output Gaussian processes. We derived a novel mean function and covariance function from the non-linear operations introduced by the transformations in a Volterra series and showed experimental results that corroborate that these non-linear models have indeed an added benefit in real-world datasets.

We envision several paths for future work. The most pressing one is extending the framework to make it suitable for larger datasets. We can use similar ideas to the ones presented in Moreno-Muñoz et al. [2018] to establish a stochastic variational lower bound for the model introduced in this paper. Exploring the non-linear models for the case of latent force models is also an interesting venue. In this paper we use an smoothing kernel with an SQ form, but it is also possible to use smoothing kernels that correspond to Green's function in dynamical systems. Automatically learning the smoothing kernel from data is also an alternative as for example in Guarnizo and Álvarez [2017].

An observation from the SARCOS experiment, one that could have been expected, is that not all ouputs provide the best predictions for the same value of C. A potential extension of our model would be to allow the automatic learning of the order C per output dimension, say C_d. The kernels we considered were derived assuming that the Volterra kernels were separable and homogeneous. Relaxing both assumptions is yet another path for future research.

References

M. A. Álvarez and N. D. Lawrence. Computationally Efficient Convolved Multiple Output Gaussian Processes. *Journal of Machine Learning Research*, 12:1425–1466, 2011.

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

Mauricio Á Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

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.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Edwin Bonilla, Daniel Steinberg, and Alistair Reid. Extended and unscented kitchen sinks. In *International Conference on Machine Learning*, pages 1651–1659, 2016.

Edwin V. Bonilla, Kian Ming Chai, and Christopher K. I. Williams. Multi-task Gaussian process prediction. In *NIPS 2007*, pages 153–160, 2008.

C M Cheng, Z K Peng, W M Zhang, and G Meng. Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review. *Mechanical Systems and Signal Processing*, 87:340–364, 2017.

Philip G Gallman and K S Narendra. Representations of nonlinear systems via the Stone-Weierstraß theorem. *Automatica*, 12(6):619–622, 1976.

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

Christian Guarnizo and Mauricio A. Álvarez. Impulse Response Estimation of Linear Time-Invariant systems using Convolved Gaussian Processes and Laguerre functions. In Marcelo Mendoza and Sergio Velastín, editors, *CIARP 2017, Valparaiso, Chile, November 7-10, 2017, Proceedings*, pages 635–642. Springer International Publishing, 2017.

Robert Haber and László Keviczky. *Nonlinear System Identification - Input-Output Modeling Approach*. Kluwer Academic, 1999.

Jouni Hartikainen and Simo Särkkä. Sequential inference for latent force models. *arXiv preprint arXiv:1202.3730*, 2012.

Jouni Hartikainen, Mari Seppänen, and Simo Särkkä. State-space inference for non-linear latent force models with application to satellite orbit prediction. pages 723–730, 2012.

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, 2002.

Neil D Lawrence, Guido Sanguinetti, and Magnus Rattray. Modelling transcriptional regulation using Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 785–792, 2007.

Miguel Lázaro-Gredilla. Bayesian warped Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1619–1627, 2012.

Pablo Moreno-Muñoz, Antonio Artés-Rodríguez, and Mauricio A. Álvarez. Heterogeneous Multi-output Gaussian Process Prediction. https://arxiv.org/abs/1805.07633, 2018.

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

Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

Simo Särkkä, Arno Solin, and Jouni Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.

Alan D Saul, James Hensman, Aki Vehtari, and Neil D Lawrence. Chained Gaussian processes. In *Artificial Intelligence and Statistics*, pages 1431–1440, 2016.

Martin Schetzen. *The Volterra and Wiener theories of nonlinear systems*. Wiley, 1980.

Lickho Song and Seungwon Lee. Explicit formulae for product moments of multivariate Gaussian random variables. *Statistics & Probability Letters*, 100:27 – 34, 2015. ISSN 0167-7152. doi: https://doi.org/10.1016/j.spl.2015.01.030. URL http://www.sciencedirect.com/science/article/pii/S016771521500036X.

Daniel M Steinberg and Edwin V Bonilla. Extended and unscented Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1251–1259, 2014.

Michalis K Titsias, Neil D Lawrence, and Magnus Rattray. Efficient sampling for Gaussian process inference using control variables. In *Advances in Neural Information Processing Systems*, pages 1681–1688, 2009.

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.