J. Stefan Institute, Ljubljana, Slovenia

Report **DP-8895**

Incorporating Linear Local Models in Gaussian Process Model

Juš Kocijan[1], Agathe Girard[2], Douglas J. Leith[3]
[1] Jozef Stefan Institute, Ljubljana
and Nova Gorica Polytechnic, Nova Gorica
[2] University of Glasgow, Glasgow
[3] Hamilton Institute, NUI Maynooth

December 2003

Contents

Chapter 1

Introduction

In this report we focus on experimental rather than first principles modelling. Owing to operating and safety constraints, the available measured data from which we are required to construct an empirical model is often concentrated mainly around equilibrium points with only relatively sparse data measured far from equilibrium. A common approach in this situation is to build local models using the data in vicinity of equilibrium points and then blend these models so as to obtain a nonlinear model covering the operating envelope, e.g. [9].

Recently the use of non-parametric Gaussian processes (GP) for modelling dynamic systems has been studied e.g. [5, 6, 2, 4, 7]. This is a probabilistic nonparametric approach to modelling. A key issue in non-parametric GP models is that in their simplest form the computational burden is cubic in the number of data points used. The computational burden is associated with matrix inversion and can be reduced by employing approximate inverses. An alternative approach considered here is to summarise measured data in the vicinity of an equilibrium point by a derivative observation i.e. a local linear model. This not only accords well with engineering practice but has the potential to directly reduce the computational burden.

The purpose of this report is to show how linear local models can be incorporated in GP models and to contribute the derivation of uncertainty propagation through such models. The latter is important for evaluation of GP dynamic systems model analysis.

The report is organised as follows. Gaussian process models are briefly described in the next chapter. The case when input is random is described in the third chapter. The fourth chapter describes derivative observations, their incorporation in GP model and illustrates their use with examples. The case when the model with derivative observations has random inputs is described in the fifth chapter. Conclusions are stated at the end.

Chapter 2

Gaussian process model

2.1 Modelling with a Gaussian Process model

The Gaussian Process (GP) model fits naturally in the Bayesian modelling framework in which the inference of a function $f(\mathbf{x})$ is described by a posterior probability distribution:

$$p(f(\mathbf{x})|\mathbf{t},\mathbf{X}) = \frac{p(\mathbf{t}|f(\mathbf{x}),\mathbf{X})p(f(\mathbf{x}))}{p(\mathbf{t}|\mathbf{X})} \tag{2.1}$$

where $p(\mathbf{t}|f(\mathbf{x}),\mathbf{X})p(f(\mathbf{x}))$ is the probability of the data given and $\mathcal{D} = \{\mathbf{t},\mathbf{X}\}$ are the N input-output data pairs, with $\mathbf{x}_i \in \Re^D$ (where $\mathbf{x}_i$ is a row vector of $\mathbf{X}$ so that $\mathbf{X}$ is the $N \times D$ matrix of inputs) and $t_i \in R$.

The idea of GP modelling is to place a prior directly on the space of admissible functions $p(f(\mathbf{x}))$, instead of parameterizing $f(\mathbf{x})$.

2.1.1 The GP model

The simplest type of priors over functions is the Gaussian one.

A Gaussian process is a Gaussian random function, fully characterized by its mean and covariance function. It can be viewed as a collection of random variables which have a joint multivariate Gaussian distribution:[1] $f(\mathbf{x}_1),\dots,f(\mathbf{x}_n) \sim \mathcal{N}(0,\mathbf{\Sigma})$, where Σ_{ij} gives the covariance between $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ and is a function of the corresponding $\mathbf{x}_i$ and $\mathbf{x}_j$: $\Sigma_{ij} = C(\mathbf{x}_i,\mathbf{x}_j)$. The covariance function $C(.,.)$ can be of any kind, provided that it generates a positive definite covariance matrix Σ. Assuming a stationary process,[2] a common choice of covariance function is

$$C(\mathbf{x}_i,\mathbf{x}_j) \;=\; v \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_i^d - x_j^d)^2\right] \tag{2.2}$$

or in vector form

$$C(\mathbf{x}_i,\mathbf{x}_j) = v \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right] \tag{2.3}$$

[1] For simplicity, we assume a zero-mean process.

[2] The stationarity assumption implies that the covariance between two points depends only on the distance between them and is invariant to translation in the input space.

where D is the input dimension and $v, w_1, \dots, w_D$ are free parameters. Typically, covariance functions are chosen such as (2.3) so that points close together in the input space are more correlated than points far apart (a smoothness assumption). The parameter v controls the vertical scale of variation and the w_i's are inversely proportional to the horizontal length-scale in dimension i ($\lambda_i = 1/\sqrt{w_i}$). Since there is one w parameter for each regressor component, it can be used as a tool to detect the relative importance of the corresponding x_i (Automatic Relevance Detection -ARD- tool introduced by [11]). Other forms of covariance functions are discussed in [1, 13]. Note that the selection of covariance functions suitable for robust generalization in typical dynamic systems applications is still an area open for research.

The use and properties of the Gaussian Process for modelling can be found in [12, 15] and we will now recall the main results.

2.1.2 Inference

Let the input/target relationship be $\mathbf{t} = f(\mathbf{X}) + \epsilon$. We assume an additive white noise with variance v_0,[3] $\epsilon \sim \mathcal{N}(0, v_0)$, and put a GP prior on $f(.)$ with covariance function as (2.3) with unknown parameters. Within this probabilistic framework, we have $t_1, \dots, t_N \sim \mathcal{N}(0, \mathbf{K})$ with $K_{ij} = \Sigma_{ij} + v_0\delta_{ij}$, where $\delta_{ij} = 1$ if $i = j$, 0 otherwise.

Based on a set of N training data pairs, $\{\mathbf{x}_i, t_i\}_{i=1}^{N}$, we wish to find the predictive distribution of y corresponding to a new given input $\mathbf{x}$. We can write

$$\mathbf{t}, t^* \sim \mathcal{N}(0, \mathbf{K}_{N+1}) \quad \text{with} \quad \mathbf{K}_{N+1} = \begin{bmatrix} \begin{bmatrix} \\ \mathbf{K} \\ \\ \end{bmatrix} & \begin{bmatrix} \\ \mathbf{k}(\mathbf{x}) \\ \\ \end{bmatrix} \\ \begin{bmatrix} \mathbf{k}(\mathbf{x})^T \end{bmatrix} & \begin{bmatrix} k(\mathbf{x}) \end{bmatrix} \end{bmatrix} \tag{2.4}$$

We can then divide this joint probability into a marginal and a conditional part. The marginal term gives us the likelihood of the training data: $\mathbf{t}|\mathbf{X} \sim \mathcal{N}(0, \mathbf{K})$, where $\mathbf{t}$ is the $N \times 1$ vector of training targets and $\mathbf{X}$ the $N \times D$ matrix of training inputs.

We need to estimate the unknown parameters of the covariance function, as well as the noise variance v_0. This is done via maximization of the log-likelihood

$$\mathcal{L}(\mathbf{\Theta}) = \log(p(\mathbf{t}|\mathbf{X})) = -\frac{1}{2}\log(|\ \mathbf{K}\ |) - \frac{1}{2}\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t} - \frac{N}{2}\log(2\pi) \tag{2.5}$$

where Θ is the vector of parameters, $\Theta = [w_1 \dots w_D \ v_0 \ v]^T$ and $\mathbf{K}$ is the $N \times N$ training covariance matrix.

The optimization requires the computation of the derivative of $\mathcal{L}$ with respect to each of the parameters:

$$\frac{\partial \mathcal{L}(\mathbf{\Theta})}{\partial \Theta_i} = -\frac{1}{2}\text{trace}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Theta_i}\right) + \frac{1}{2}\mathbf{t}^T\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Theta_i}\mathbf{K}^{-1}\mathbf{t} \tag{2.6}$$

Here, it involves the computation of the inverse of the $N \times N$ covariance matrix $\mathbf{K}$ at every iteration, which can become computationally demanding for large N. An alternative method is, in the Bayesian framework, to put a prior on the parameters and compute their posterior probability (with the integration done using Markov Chain Monte Carlo methods, see [13]).

[3] Correlated noise can also be considered, as shown in [10]

2.1.3 Prediction

The conditional part of (2.4) provides us with the predictive distribution of t^*, $p(t^*|\mathbf{t}, \mathbf{X}, \mathbf{x}) = \frac{p(\mathbf{t},t^*)}{p(\mathbf{t}|\mathbf{X})}$. It can be shown that this distribution is Gaussian with mean and variance

$$\boxed{\mu(\mathbf{x}) \;=\; \mathbf{k}(\mathbf{x})^T\ \mathbf{K}^{-1}\ \mathbf{t}} \tag{2.7}$$

$$\boxed{\sigma^2(\mathbf{x}) \;=\; k(\mathbf{x}) \;-\; \mathbf{k}(\mathbf{x})^T\ \mathbf{K}^{-1}\ \mathbf{k}(\mathbf{x}) + v_0} \tag{2.8}$$

where $\mathbf{k}(\mathbf{x}) = [C(\mathbf{x}_1, \mathbf{x}), \ldots, C(\mathbf{x}_N, \mathbf{x})]^T$ is the $N \times 1$ vector of covariances between the test and training cases and $k(\mathbf{x}) = C(\mathbf{x}, \mathbf{x})$ is the covariance between the test input and itself.

Clearly, what has been presented above is the modelling of a static function. However, it can be readily extended to dynamic systems. Consider the following autoregressive model where the current output depends on delayed outputs and control inputs:

$$\begin{aligned} y(k) \;&=\; f(y(k-1), y(k-2), \ldots, y(k-L), \\ &\quad\; u(k-1), u(k-2), \ldots, u(k-L)) + \epsilon \end{aligned} \tag{2.9}$$

where ϵ is a white noise and k denotes consecutive number of data sample.

Example

Let $\mathbf{x}_k$ denote the state vector at k, composed of the previous outputs y and inputs u, up to a given lag L: $\mathbf{x}_k = [y(k-1), y(k-2), \ldots, y(k-L), u(k-1), u(k-2), \ldots, u(k-L)]$. We wish to model this dynamic system using a Gaussian Process and make multiple-step ahead predictions. A possible choice of input and target vectors from N input and output data pairs for such system where the dimension of GP model D is for example determined by double value of given lag L is as follows.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_k \\ \vdots \\ \mathbf{x}_{N-L-1} \\ \mathbf{x}_{N-L} \end{bmatrix} = \begin{bmatrix} y(L) & y(L-1) & \ldots & y(1) & u(L) & \ldots & u(1) \\ y(L+1) & y(L) & \ldots & y(2) & u(L+1) & \ldots & u(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y(k-1) & y(k-2) & \ldots & y(k-L) & u(k-1) & \ldots & u(k-L) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y(N-2) & y(N-3) & \ldots & y(N-L-1) & u(N-2) & \ldots & u(N-L-1) \\ y(N-1) & y(N-2) & \ldots & y(N-L) & u(N-1) & \ldots & u(N-L) \end{bmatrix} \tag{2.10}$$

for input matrix with $D = 2L$ components (also regressors, columns) and

$$\mathbf{t} = \begin{bmatrix} y(L+1) \\ y(L+2) \\ \vdots \\ y(k+1) \\ \vdots \\ y(N-1) \\ y(N) \end{bmatrix} \tag{2.11}$$

for target vector.

* * *

Multi-step ahead predictions can be achieved by iteratively making repeated one-step ahead predictions up to the desired time span and at the same time feed back the predictive mean (estimate of the output). This approach is of course approximate, because we neglect noise on the lagged outputs on the right-hand side, but is similar to that widely used, for example, when modelling dynamic systems with neural networks or fuzzy models. The obtained variance is still the indicator of regions where model can be more or less trusted, but the values of predicted mean and variance are not correct.

In [2], iterative multi-step ahead prediction is done by feeding back the predictive mean, as well as the predictive variance at each time-step, thus taking the uncertainty attached to each intermediate prediction into account. This means that input at which we wish to predict becomes a normally distributed random variable, which is discussed in the next chapter.

Chapter 3

Prediction at a new random input for a GP model

In this chapter, we are summarising the results of extensions of the GP modelling framework for dealing with random inputs. We first look at making a prediction for a new random input $\mathbf{x}$, when the training inputs are noise-free, a situation that might arise for instance when making multiple-step ahead prediction of a noise-free time-series by propagation of the uncertainty. More elaborate information on this topic can be found in [3].

3.1 Prediction at a random $\mathbf{x}$

In the previous chapter, we saw how based on observed data and on a new input $\mathbf{x}$, the predictive distribution of the corresponding $f(\mathbf{x})$ was readily obtained. We recall that $p(f(\mathbf{x})|\mathcal{D},\mathbf{x})$, where $\mathcal{D} = \{\mathbf{t},\mathbf{X}\}$ is the set of observed targets and corresponding inputs, is Gaussian with mean and variance

$$\mu(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T\boldsymbol{\beta} = \sum_{i=1}^{N} \beta_i C(\mathbf{x},\mathbf{x}_i) \tag{3.1}$$

$$\sigma^2(\mathbf{x}) = C(\mathbf{x},\mathbf{x}) - \mathbf{k}(\mathbf{x})^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}) = C(\mathbf{x},\mathbf{x}) - \sum_{i,j=1}^{N} K_{ij}^{-1} C(\mathbf{x},\mathbf{x}_i)C(\mathbf{x},\mathbf{x}_j) \tag{3.2}$$

where $\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{t}$ and $C(\mathbf{x},\mathbf{x}_i)$ is the covariance between $f(\mathbf{x})$ and $f(\mathbf{x}_i)$.

If we now wish to make a prediction at $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u},\boldsymbol{\Sigma}_x)$, we need to integrate the predictive distribution over the possible $\mathbf{x}$'s, that is

$$p(f(\mathbf{x})|\mathcal{D},\mathbf{u},\boldsymbol{\Sigma}_x) = \int p(f(\mathbf{x})|\mathcal{D},\mathbf{x})p(\mathbf{x})d\mathbf{x} \tag{3.3}$$

where $p(\mathbf{x}) = \mathcal{N}_{\mathbf{x}}(\mathbf{u},\boldsymbol{\Sigma}_x)$ and $p(f(\mathbf{x})|\mathcal{D},\mathbf{x})$ has mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$.

Now, as $p(f(\mathbf{x})|\mathcal{D},\mathbf{x})$ is a nonlinear function of $\mathbf{x}$, this integral cannot be solved analytically without approximation.

3.1.1 Numerical approximation

One way of solving this integral is to go for a numerical approximation, that is

$$p(f(\mathbf{x})|\mathcal{D}, \mathbf{u}, \mathbf{\Sigma}_x) \simeq \frac{1}{T} \sum_{t=1}^{T} p(f(\mathbf{x})|\mathcal{D}, \mathbf{x}^t) \tag{3.4}$$

where $\mathbf{x}^t$ is a sample from $p(\mathbf{x})$. This can be done easily enough using MCMC methods.

3.1.2 Analytical approximation

We are more interested in a Gaussian analytical approximation, that is, in computing the mean and variance of $p(f(\mathbf{x})|\mathcal{D}, \mathbf{u}, \mathbf{\Sigma}_x)$ only.

The expressions for mean and variance are [3]:

$$\boxed{m(\mathbf{u}, \mathbf{\Sigma}_x) = E_{\mathbf{x}}[\mu(\mathbf{x})]} \tag{3.5}$$

where we denote by $m(\mathbf{u}, \mathbf{\Sigma}_x)$ the expectation of $y|\mathcal{D}, \mathbf{u}, \mathbf{\Sigma}_x$ and

$$\boxed{v(\mathbf{u}, \mathbf{\Sigma}_x) = E_{\mathbf{x}}[\sigma^2(\mathbf{x})] + E_{\mathbf{x}}[\mu(\mathbf{x})^2] - (E_{\mathbf{x}}[\mu(\mathbf{x})])^2} \tag{3.6}$$

where $v(\mathbf{u}, \mathbf{\Sigma}_x)$ is the variance of $y|\mathcal{D}, \mathbf{u}, \mathbf{\Sigma}_x$.

3.2 The special case of the Gaussian covariance function

We consider the Gaussian covariance function given by (2.3):

$$C(\mathbf{x}_i, \mathbf{x}_j) = v \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^{-1} (\mathbf{x}_i - \mathbf{x}_j)\right] \tag{3.7}$$

Since it is not really a Gaussian distribution, it is denoted by N for notational convenience (not by $\mathcal{N}$ as Gaussian distributions), because it just denotes some function of the same parametric form. We write it as we would a Gaussian distribution for $\mathbf{x}_i$, centered on $\mathbf{x}_j$:

$$C(\mathbf{x}_i, \mathbf{x}_j) = \tau N_{\mathbf{x}_i}(\mathbf{x}_j, \mathbf{W}) \quad \text{with} \quad \tau = (2\pi)^{D/2} |\mathbf{W}|^{1/2} v \tag{3.8}$$

3.2.1 Prediction at $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma}_x)$

We have seen in section 3.1 that in order to predict at a noisy input, we needed to integrate the predictive distribution over the input distribution (equation (3.3)). Then, a Gaussian analytical approximation of this integral reduced the problem to computing the mean and variance of $p(f(\mathbf{x})|\mathcal{D}, \mathbf{u}, \mathbf{\Sigma}_x)$.

Since $p(\mathbf{x})$ is a Gaussian distribution, if the covariance function $C(.,.)$ happens to be also Gaussian, as given by (3.8), we can use the product of Gaussians (3.9) and solve these integrals exactly.

$$\begin{aligned}
&\mathcal{N}_x(a,A)\mathcal{N}_x(b,B) = z\mathcal{N}_x(c,C)\\
&C = (A^{-1}+B^{-1})^{-1}, \quad c = C(A^{-1}a + B^{-1}b)\\
&z = \mathcal{N}_a(b, A+B) \quad \text{or} \quad z = \mathcal{N}_b(a, A+B)
\end{aligned} \tag{3.9}$$

Note that z is usually found expressed as $z = (2\pi)^{-D/2}|A+B|^{1/2}\exp\left[-\frac{1}{2}(a-b)^T(A+B)^{-1}(a-b)\right]$.

The exact derivations can be found in [3]. Here we are presenting just the final results.

The new predictive mean is equivalent to that obtained for a noise-free test input, except that the covariance between the noisy input and the noise-free training input is computed using a *modified* covariance function which accounts for the uncertainty on the test input. We can write

$$\boxed{m(\mathbf{u},\mathbf{\Sigma}_x) = \textstyle\sum_{i=1}^{N}\beta_i C_{mod_1}(\mathbf{u},\mathbf{x}_i)} \tag{3.10}$$

where

$$C_{mod_1}(\mathbf{u},\mathbf{x}_i) = v|\mathbf{I}+\mathbf{W}^{-1}\mathbf{\Sigma}_x|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{u}-\mathbf{x}_i)^T(\mathbf{W}+\mathbf{\Sigma}_x)^{-1}(\mathbf{u}-\mathbf{x}_i)\right] \tag{3.11}$$

That is to say, the correlation length is 'lengthened' to account for the uncertainty on the new input and the vertical amplitude of variation (formally controlled by v) is accordingly diminished.

The new predictive variance is

$$v(\mathbf{u},\mathbf{\Sigma}_x) = v + \tau^2\sum_{i,j=1}^{N}(\beta_i\beta_j - K_{ij}^{-1})N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W})\mathcal{N}_{\mathbf{u}}\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}, \mathbf{\Sigma}_x + \frac{\mathbf{W}}{2}\right) - m^2(\mathbf{u},\mathbf{\Sigma}_x) \tag{3.12}$$

As in the case of new predictive mean this can be written using modified covariance functions

$$\boxed{v(\mathbf{u},\mathbf{\Sigma}_x) = v + \textstyle\sum_{i,j=1}^{N}(\beta_i\beta_j - K_{ij}^{-1})C_{mod_2}(\mathbf{x}_i,\mathbf{x}_j)C_{mod_3}(\mathbf{u},\mathbf{x}_b) - m^2(\mathbf{u},\mathbf{\Sigma}_x)} \tag{3.13}$$

where $C_{mod_2}(\mathbf{x}_i,\mathbf{x}_j) = \tau N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W})$

$$\begin{aligned}
C_{mod_2}(\mathbf{x}_i,\mathbf{x}_j) &= v_0 2^{-\frac{D}{2}}\exp\left[-\frac{1}{2}(\mathbf{x}_i-\mathbf{x}_j)^T(\frac{\mathbf{W}}{2})^{-1}(\mathbf{x}_i-\mathbf{x}_j)\right]\\
&= v_0 2^{-\frac{D}{2}}\exp\left[-\frac{1}{2}\sum_{d=1}^{D}\frac{w_d}{2}(x_i^d - x_j^d)^2\right]
\end{aligned} \tag{3.14}$$

and $C_{mod_3}(\mathbf{u},\mathbf{x}_b) = \tau\mathcal{N}_{\mathbf{u}}\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_x\right)$.

$$C_{mod_3}(\mathbf{u},\mathbf{x}_b) = v_0\left|\frac{1}{2}\mathbf{I}+\mathbf{W}^{-1}\mathbf{\Sigma}_x\right|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{u}-\mathbf{x}_b)^T\left(\frac{\mathbf{W}}{2}+\mathbf{\Sigma}_x\right)^{-1}(\mathbf{u}-\mathbf{x}_b)\right] \tag{3.15}$$

with $\mathbf{x}_b = \frac{\mathbf{x}_i+\mathbf{x}_j}{2}$.

The examples illustrating simulation with random input can be found in [2, 5, 7].

3.3 The special case of the linear covariance function

Here we give a full derivation for the linear covariance function.

We consider the linear covariance function given by

$$C(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{W} \mathbf{x}_j \tag{3.16}$$

where W is a diagonal matrix of hyperparameters.

The predictive mean for the case when $\mathbf{x}$ is not random is

$$\mu(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T \boldsymbol{\beta} \tag{3.17}$$

$$\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{t} = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1}\mathbf{t} \tag{3.18}$$

$$\mathbf{k}(\mathbf{x}) = C(\mathbf{x}, \mathbf{X}) = \mathbf{X}\mathbf{W}\mathbf{x}^T \tag{3.19}$$

Therefore

$$\mu(\mathbf{x}) = (\mathbf{X}\mathbf{W}\mathbf{x}^T)^T (\mathbf{X}\mathbf{W}\mathbf{X}^T)^{-1}\mathbf{t} \tag{3.20}$$

$$\boxed{\mu(\mathbf{x}) = \mathbf{x}(\mathbf{X}\mathbf{W})^T (\mathbf{X}\mathbf{W}\mathbf{X}^T)^{-1}\mathbf{t}} \tag{3.21}$$

And predictive variance is

$$\sigma^2(\mathbf{x}) = C(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \tag{3.22}$$

$$\boxed{\sigma^2(\mathbf{x}) = \mathbf{x}\mathbf{W}\mathbf{x}^T - (\mathbf{X}\mathbf{W}\mathbf{x}^T)^T (\mathbf{X}\mathbf{W}\mathbf{X}^T)^{-1} (\mathbf{X}\mathbf{W}\mathbf{x}^T)} \tag{3.23}$$

We can write

$$\begin{aligned}
\sigma^2(\mathbf{x}) &= \mathbf{x}\mathbf{W}\mathbf{x}^T - (\mathbf{X}\mathbf{W}\mathbf{x}^T)^T (\mathbf{X}\mathbf{W}\mathbf{X}^T)^{-1} (\mathbf{X}\mathbf{W}\mathbf{x}^T) \\
&= \mathbf{x}\mathbf{W}\mathbf{x}^T - \mathbf{x}(\mathbf{X}\mathbf{W})^T (\mathbf{X}\mathbf{W}\mathbf{X}^T)^{-1} (\mathbf{X}\mathbf{W})\mathbf{x}^T \\
&= \mathbf{x}(\mathbf{W} - (\mathbf{X}\mathbf{W})^T (\mathbf{X}\mathbf{W}\mathbf{X}^T)^{-1} (\mathbf{X}\mathbf{W}))\mathbf{x}^T \\
&= \mathbf{x}\boldsymbol{\alpha}\mathbf{x}^T
\end{aligned} \tag{3.24}$$

where $\boldsymbol{\alpha} = \mathbf{W} - (\mathbf{X}\mathbf{W})^T (\mathbf{X}\mathbf{W}\mathbf{X}^T)^{-1} (\mathbf{X}\mathbf{W})$.

New predictive mean

According to (3.5), we have to compute

$$\begin{aligned}
m(\mathbf{u}, \boldsymbol{\Sigma}_x) &= E_{\mathbf{x}}[\mu(\mathbf{x})] \\
&= \int \mathbf{x}(\mathbf{X}\mathbf{W})^T \boldsymbol{\beta} p(\mathbf{x}) d\mathbf{x} \\
&= \int \mathbf{x} p(\mathbf{x}) d\mathbf{x} (\mathbf{X}\mathbf{W})^T \boldsymbol{\beta}
\end{aligned} \tag{3.25}$$

since $\int \mathbf{x} p(\mathbf{x}) d\mathbf{x} = E_{\mathbf{x}}[\mathbf{x}] = \mathbf{u}$, we can write

$$\boxed{m(\mathbf{u}, \boldsymbol{\Sigma}_x) = \mathbf{u}(\mathbf{X}\mathbf{W})^T \boldsymbol{\beta}} \tag{3.26}$$

It is apparent that $m(\mathbf{u}, \boldsymbol{\Sigma}_x) = \mu(\mathbf{u})$.

New predictive variance

According to (3.6) the variance is given by

$$v(\mathbf{u},\mathbf{\Sigma}_x) = E_{\mathbf{x}}[\sigma^2(\mathbf{x})] + E_{\mathbf{x}}[\mu(\mathbf{x})^2] - (E_{\mathbf{x}}[\mu(\mathbf{x})])^2 \tag{3.27}$$

The last component is

$$(E_{\mathbf{x}}[\mu(\mathbf{x})])^2 = m(\mathbf{u},\mathbf{\Sigma}_x)^2 \tag{3.28}$$

And the others are:

$$\begin{aligned} E_{\mathbf{x}}[\sigma^2(\mathbf{x})] &= \int \sigma^2 p(\mathbf{x}) d\mathbf{x} \\ &= \int \mathbf{x}\boldsymbol{\alpha}\mathbf{x}^T p(\mathbf{x}) d\mathbf{x} \\ &= \int \mathbf{x}\boldsymbol{\alpha}\mathbf{x}^T \mathcal{N}_{\mathbf{x}}(\mathbf{u},\mathbf{\Sigma}_x) d\mathbf{x} \end{aligned} \tag{3.29}$$

Using formula for expectation of a quadratic form under a Gaussian[1] we obtain

$$E_{\mathbf{x}}[\sigma^2(\mathbf{x})] = \mathbf{u}\boldsymbol{\alpha}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\alpha}\mathbf{\Sigma}_{\mathbf{x}}] \tag{3.30}$$

where $\boldsymbol{\alpha} = \mathbf{W} - (\mathbf{XW})^T(\mathbf{XWX}^T)^{-1}(\mathbf{XW})$.

Let calculate first

$$\begin{aligned} \mu(\mathbf{x})^2 &= (\mathbf{XWx}^T)^T \boldsymbol{\beta}\boldsymbol{\beta}^T (\mathbf{XWx}^T) \\ &= \mathbf{x}(\mathbf{XW})^T \boldsymbol{\beta}\boldsymbol{\beta}^T (\mathbf{XW})\mathbf{x}^T \\ &= \mathbf{x}\boldsymbol{\gamma}\mathbf{x}^T \end{aligned} \tag{3.31}$$

where $\boldsymbol{\gamma} = (\mathbf{XW})^T \boldsymbol{\beta}\boldsymbol{\beta}^T (\mathbf{XW})$.

Similarly we

$$\begin{aligned} E_{\mathbf{x}}[\mu(\mathbf{x})^2] &= \int \mathbf{x}\boldsymbol{\gamma}\mathbf{x}^T p(\mathbf{x}) d\mathbf{x} \\ &= \int \mathbf{x}\boldsymbol{\gamma}\mathbf{x}^T \mathcal{N}_{\mathbf{x}}(\mathbf{u},\mathbf{\Sigma}_x) d\mathbf{x} \end{aligned} \tag{3.32}$$

and again using formula for expectation of a quadratic form under a Gaussian we obtain

$$E_{\mathbf{x}}[\mu(\mathbf{x})^2] = \mathbf{u}\boldsymbol{\gamma}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\gamma}\mathbf{\Sigma}_{\mathbf{x}}] \tag{3.33}$$

where $\boldsymbol{\gamma} = (\mathbf{XW})^T \boldsymbol{\beta}\boldsymbol{\beta}^T (\mathbf{XW})$.

So the new variance is

$$\begin{aligned} v(\mathbf{u},\mathbf{\Sigma}_x) &= \mathbf{u}\boldsymbol{\alpha}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\alpha}\mathbf{\Sigma}_{\mathbf{x}}] + \mathbf{u}\boldsymbol{\gamma}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\gamma}\mathbf{\Sigma}_{\mathbf{x}}] - m(\mathbf{u},\mathbf{\Sigma}_x)^2 \\ &= \mathbf{u}\boldsymbol{\alpha}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\alpha}\mathbf{\Sigma}_{\mathbf{x}}] + v_0 + \mathbf{u}\boldsymbol{\gamma}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\gamma}\mathbf{\Sigma}_{\mathbf{x}}] - \mathbf{u}(\mathbf{XW})^T \boldsymbol{\beta}\boldsymbol{\beta}^T (\mathbf{XW})\mathbf{u}^T \\ &= \mathbf{u}\boldsymbol{\alpha}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\alpha}\mathbf{\Sigma}_{\mathbf{x}}] + v_0 + \mathbf{u}\boldsymbol{\gamma}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\gamma}\mathbf{\Sigma}_{\mathbf{x}}] - \mathbf{u}\boldsymbol{\gamma}\mathbf{u}^T \end{aligned} \tag{3.34}$$

$$\boxed{v(\mathbf{u},\mathbf{\Sigma}_x) = \mathbf{u}\boldsymbol{\alpha}\mathbf{u}^T + \mathrm{Tr}[\boldsymbol{\alpha}\mathbf{\Sigma}_{\mathbf{x}}] + \mathrm{Tr}[\boldsymbol{\gamma}\mathbf{\Sigma}_{\mathbf{x}}]} \tag{3.35}$$

With a very little calculation it can be shown that $v(\mathbf{u},\mathbf{\Sigma}_x = 0) = \sigma^2(\mathbf{u})$.

[1]
$$\int (\mathbf{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}) \mathcal{N}_{\mathbf{x}}(\mathbf{u},\mathbf{\Sigma}_x) d\mathbf{x} = (\boldsymbol{\mu}-\mathbf{u})^T \mathbf{\Sigma}^{-1} (\boldsymbol{\mu}-\mathbf{u}) \mathrm{Tr}[\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_x]$$

Chapter 4

Incorporating derivative observations

4.1 Derivative observations

The Gaussian process modelling framework is readily extended to include situations where derivatives of a function are observed as well as (or instead of) the values of the function itself. Some results on this topic can be found in [8, 14]. Since differentiation is a linear operation, the derivative of a GP remains a GP. To work with derivative observations we need only replace the Gaussian covariance function in a GP model with the appropriate derivative covariance function. The output (target) vector $\mathbf{t}$ which before consisted solely of output measurements now also contains derivative observations. The corresponding input (training) data are the values of the regressor associated with each derivation observation.

Example

Again consider the following autoregressive model from equation (2.9) where the current output depends on delayed outputs and control inputs:

$$\begin{aligned} y(k) \quad &= \quad f(y(k-1), y(k-2), \ldots, y(k-L), \\ &\qquad u(k-1), u(k-2), \ldots, u(k-L)) + \epsilon \end{aligned} \tag{4.1}$$

where ϵ is a white noise and k denotes consecutive number of data sample.

A possible (but not necessary) choice of data grouping in input matrix with $D = 2L$ components and target vector is as follows.

$$\mathbf{X} = \begin{bmatrix} \mathbf{Y}_{oeq} & \mathbf{U}_{oeq} \\ \mathbf{Y}_{eq} & \mathbf{U}_{eq} \\ \mathbf{Y}_{eq} & \mathbf{U}_{eq} \\ \vdots & \vdots \\ \mathbf{Y}_{eq} & \mathbf{U}_{eq} \\ \vdots & \vdots \end{bmatrix} \qquad \mathbf{t} = \begin{bmatrix} \mathbf{Y}_{oeq1} \\ \mathbf{Y}_{eq} \\ [\frac{\partial f}{\partial y(k)}] \\ \vdots \\ [\frac{\partial f}{\partial u(k)}] \\ \vdots \end{bmatrix} \tag{4.2}$$

where

$\mathbf{Y}_{oeq1}$ is a vector of target response points out of equilibria;

$\mathbf{Y}_{oeq}$ is a vector of input response points out of equilibria;

$\mathbf{U}_{oeq}$ is a matrix of input input points out of equilibria;

$\mathbf{Y}_{eq}$ is a vector of equilibria response points;

$\mathbf{U}_{eq}$ is a matrix of equilibria input points;

$[\frac{\partial f}{\partial y}]$ is a vector of derivative observations of response component (vector of a linear model coefficient in different points);

$[\frac{\partial f}{\partial u}]$ is a vector of derivative observations of input component (vector of a linear model coefficient in different points).

There exist all together D derivative observation vectors ($[\frac{\partial f}{\partial y}]$ and $[\frac{\partial f}{\partial u}]$), one for each component of input matrix. This means that dimensions of input matrix are $(n + D \cdot n_D) \times D$ and dimensions of target vector are $(n + D \cdot n_D) \times 1$ where n is a number of function equilibrium and nonequilibrium observations (input-output data points) and n_D is a number of derivative observations (input-output data points in equilibrium points where derivative observations are derived).

* * *

Since the identification data is changed the covariance matrix must also be changed. We can define the covariance relating any two data points as

$$C(\mathbf{x}_i, \mathbf{x}_j) \;=\; v \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_i^d - x_j^d)^2\right] \tag{4.3}$$

in the case of two functional observations,

$$C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \mathbf{x}_j) = -v w_d (x_i^d - x_j^d) \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_i^d - x_j^d)^2\right] \tag{4.4}$$

in the case of mixed derivative and functional observation and

$$C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \frac{\partial \mathbf{x}_j}{\partial x_j}) = v w_e (\delta_{e,d} - w_d (x_i^e - x_j^e)(x_i^d - x_j^d)) \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_i^d - x_j^d)^2\right] \tag{4.5}$$

in the case of two derivative observations, where $\delta_{e,d}$ is Kronecker operator between e^{th} component derivative in vector $\mathbf{x}_i$ and d^{th} component derivative in vector $\mathbf{x}_j$.

Example

A corresponding ordering of covariance functions in covariance matrix for the data ordering in the previous example would be

$$\mathbf{K} = \begin{bmatrix} \left[C(\mathbf{x}_i, \mathbf{x}_j)\right] & \left[C(\mathbf{x}_i, \frac{\partial \mathbf{x}_j}{\partial x_j})\right]_{d=1} & \cdots & \left[C(\mathbf{x}_i, \frac{\partial \mathbf{x}_j}{\partial x_j})\right]_{d=D} \\ \left[C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \mathbf{x}_j)\right]_{d=1} & \left[C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \frac{\partial \mathbf{x}_j}{\partial x_j})\right]_{e=1,d=1} & \cdots & \left[C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \frac{\partial \mathbf{x}_j}{\partial x_j})\right]_{e=1,d=D} \\ \vdots & \vdots & \vdots & \vdots \\ \left[C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \mathbf{x}_j)\right]_{d=D} & \left[C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \frac{\partial \mathbf{x}_j}{\partial x_j})\right]_{e=D,d=1} & \cdots & \left[C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \frac{\partial \mathbf{x}_j}{\partial x_j})\right]_{e=D,d=D} \end{bmatrix} \tag{4.6}$$

$$\mathbf{k}(\mathbf{x}) = \begin{bmatrix} \big[C(\mathbf{x}_i, \mathbf{x})\big] \\ \\ \Big[C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \mathbf{x})\Big]_{d=1} \\ \vdots \\ \Big[C(\frac{\partial \mathbf{x}_i}{\partial x_i}, \mathbf{x})\Big]_{d=D} \end{bmatrix} \tag{4.7}$$

$$k(\mathbf{x}) = \big[C(\mathbf{x}, \mathbf{x})\big] = v \tag{4.8}$$

* * *

The GP model acts to integrate and smooth the noisy derivative observations. Derivative observations around an equilibrium point can be interpreted as observations of a local linear model about this equilibrium point. This means that the derivative observations can be synthesised using standard linear regression. Such synthetic derivative observations can then be used to summarise training points in the vicinity of equilibrium points, thereby effectively reducing the number of data points in the model for computational purposes. It is important to note that a local linear input-output model such as a transfer function model only specifies a derivative observation up to a co-ordinate transformation. In this paper we always use lagged inputs and outputs as our state co-ordinates for simplicity but of course other choices are possible.

Input data may or may not contain information about noise. If noise information for function points is available it is added to covariance matrix diagonal elements corresponding to these data. For the points with no information about noise output signal variance hyperparameter v_0 is learned as in Chapter 2.

When standard identification methods are used for derivative observation, noise information for each local model is also obtained. The covariance matrices of each linear local model obtained at identification (see [14]) are added to overall covariance matrix for the corresponding derivative component.

The predictive distribution has mean and variance respectively given by equations (2.7) and (2.8).

Example

Modelling

Consider the nonlinear dynamic system described by equation

$$y(k+1) = 0.5y(k) + \tanh(y(k) + u^3(k)) \tag{4.9}$$

We are interested in exploring the potential for achieving an accurate model using derivative observations at equilibrium points plus a small number of function observations at off-equilibrium points. We selected ten equilibrium points uniformly spanning the operating region of interest. At each equilibrium point we applied a small-scale pseudo random binary signal with mean 0 and magnitude 0.03; the corresponding output signal is contaminated with normally distributed measurement noise in the range [-0.001,0.001]. A linear approximation to the local dynamics at the equilibrium point was identified using the Matlab algorithm IV4. In addition to this equilibrium information, a small, sparse set of off-equilibrium input-output data consisting of only 6 data points was selected (larger numbers of off-equilibrium observations were also studied and this number was chosen as a compromise between the accuracy of achieved fit and number of data points used). A nonparametric Gaussian process prior model was then constructed. To summarise, the model made use of the following training information:

- Ten equilibrium input-output values spanning the operating region of interest.

- The set of coefficients of the identified linear models representing partial derivatives of the output.
- The six input-output values that were sampled out of equilibrium points.

The response on validation data together with process response is given in Figure 4.1. We assess the goodness of the fit of the validation signal by computing the following cost functions:

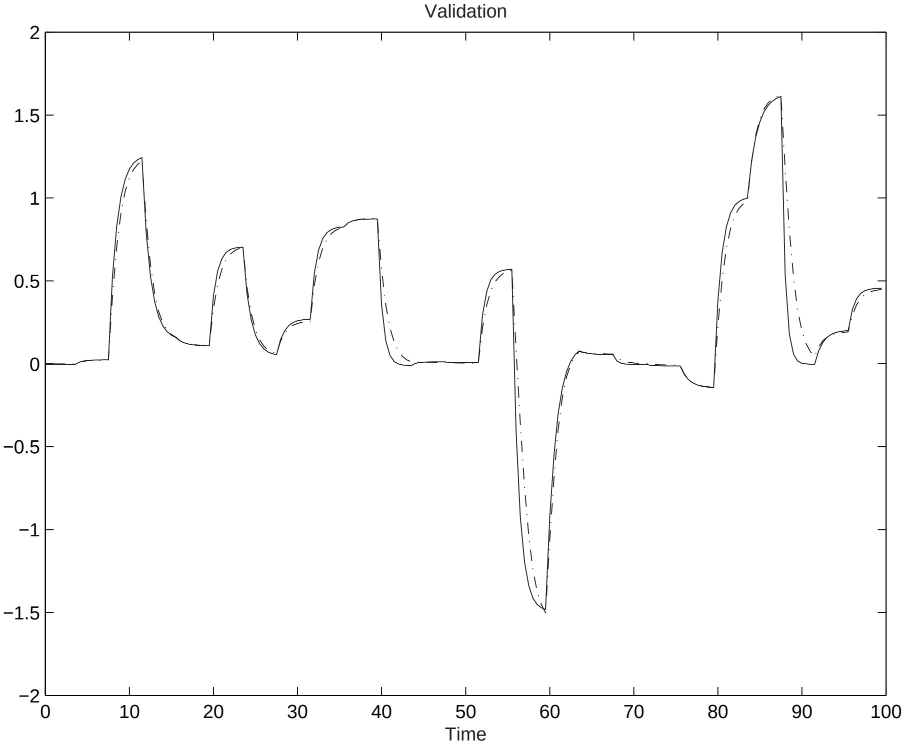

Figure 4.1: Response on validation data: GP model response - dash-dot line, process response - solid line

- average absolute test error
$$AE = \frac{1}{N}\sum \mid \hat{y} - y \mid = 0.0467 \tag{4.10}$$
where N is the number of validation points, y the process response (target) and $\hat{y}$ is the model output (predictive mean);
- average squared test error
$$SE = \frac{1}{N}\sum (\hat{y} - y)^2 = 0.0124 \tag{4.11}$$
- minus log-predictive density error
$$\begin{aligned} LD &= \frac{1}{2N}\sum (\log(2\pi) + \log(\sigma^2) + \frac{(\hat{y} - y)^2}{\sigma^2}) \\ &= -0.826 \end{aligned} \tag{4.12}$$
where σ^2 is the predictive variance.

The response is comparable to that in [5], where a GP model without derivative observations based on 200 data points was constructed. More descriptive of quality of the nonparametric model here (that contains only 36 points) is the error between the process and the GP model in the operating region $u(k) \in [-1, 1], y(k) \in [-2, 2]$, see Figure 4.2. It can be seen from the figure that the model covers the surprisingly wide area out of equilibrium locus.

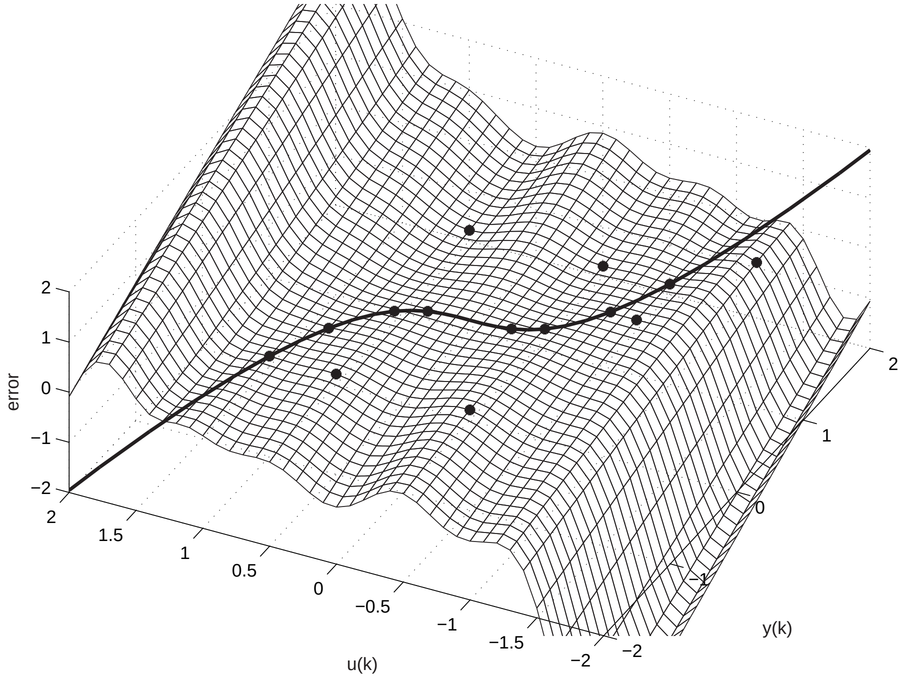

Figure 4.2: Error between GP model and process where solid line presents equilibrium locus and dots are training points

Control

The nonlinear model predictive control approach considered here is a receding horizon strategy which is essentially the same as in [7] (where a Gaussian process model without derivative observations was used). The model used for control is fixed, identified off-line, which means that the control algorithm is not an adaptive one. The moving-horizon minimisation problem is of the form [7]

$$\min_{\mathbf{U}(k)} \; [r(k+P) - \hat{y}(k+P)]^2 \tag{4.13}$$

where r is reference trajectory, $\mathbf{U}(k) = [u(k) \ldots u(k+P)]$ is input signal, P is the coincidence point (the point where a match between output and reference value is expected). The optimisation algorithm, which uses Matlab Optimization toolbox routine for unconstrained optimisation, is solved at each sample time over a prediction horizon of length P, for a series of moves which equals to control horizon N_u. The process model is the Gaussian process model obtained in previous step, which includes derivative observations. As with other nonlinear predictive control algorithms the computation burden is significant. This and other issues of interest for applied NMPC are discussed in the paper [7].

In our case the reference trajectory r is defined so that it approaches the set-point exponentially from the current output value. This means that the closed-loop system should behave close to the first order system when the process model is a good description of the process itself. The coincidence point for the chosen MPC was selected as $P = 8$ and the control horizon $N_u = 1$. The set-point for the closed-loop system was chosen in such a way that it covers a large portion of the operating region, forcing the closed-loop system to exercise far from equilibrium. The closed-loop response of the unconstrained control is given in Figure 4.3. It can be seen from Figure 4.3 that the closed-loop response follows the desired set point very well including during demands for large set point changes that take the system far from equilibrium point (but remaining within the operating envelope $u(k) \in [-1, 1], y(k) \in [-2, 2]$).

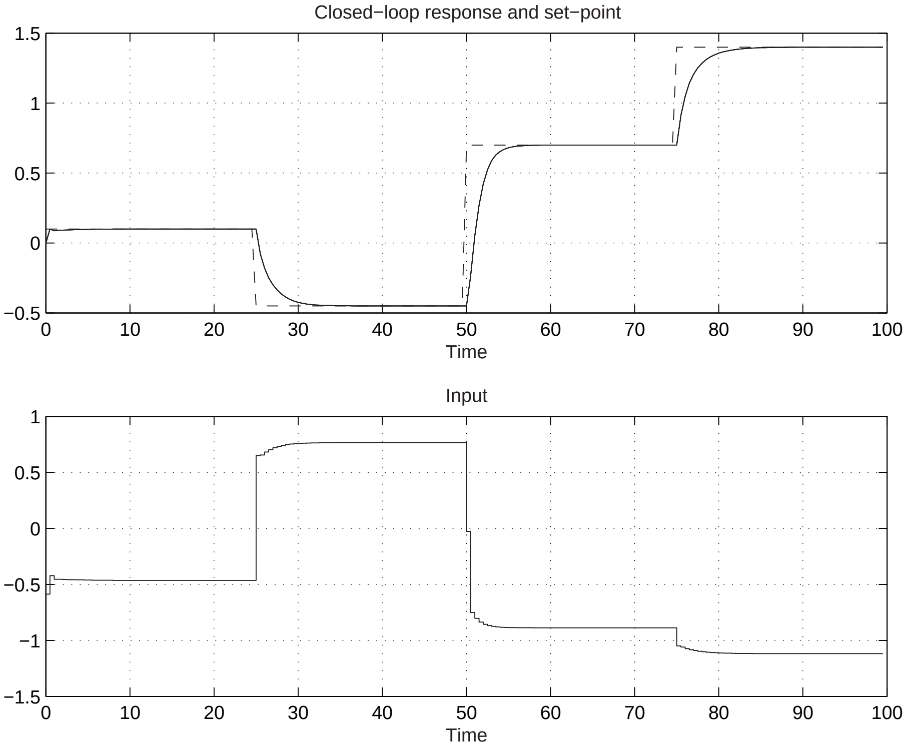

Figure 4.3: Response of GP model based predictive control - solid line and set point - dashed line (upper figure) and control signal (bottom figure)

The example shows that a grey-box model consisting of local linear models obtained from data around equilibrium points, the corresponding equilibrium points and a very small number of data out of equilibrium points can be effectively used for predictive control. Moreover, the data used to obtain the grey-box model is well suited to the kind of data usually available in practice when carrying out experimental modelling. The model obtained is relatively small in comparison with a GP model that does not make use of derivative observations, while the model quality is comparable. This makes it very suitable for applications.

In the case information about variance at random input is given this would enable calculation of propagated model variance and construction of robust predictive control as it is possible with GP model that does not include derivative observations [7].

* * *

Chapter 5

Prediction at a new random input for GP model with derivative observations

Investigation to derive propagation of model uncertainties of the GP model which contains derivative observations is presented in this chapter.

5.1 Introduction

When derivative observations are used for system modelling, they are always used in combination with function observations (points). This means that input data is combination of function and derivative observations and also that prediction and variance of output have to be calculated taking both sorts of data in account.

As shown previously predicted mean value and variance at random input can be obtained as

$$m(\mathbf{u}, \mathbf{\Sigma}_x) = E_{\mathbf{x}}[\mu(\mathbf{x})] \tag{5.1}$$

$$v(\mathbf{u}, \mathbf{\Sigma}_x) = E_{\mathbf{x}}[\sigma^2(\mathbf{x})] + E_{\mathbf{x}}[\mu(\mathbf{x})^2] - E_{\mathbf{x}}[\mu(\mathbf{x})]^2 \tag{5.2}$$

As we have already said the input matrix is put together from function points and derivative points and corresponding target from corresponding targets.

For $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^D$, the covariance between $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ is given by

$$\mathrm{Cov}[f(\mathbf{x}_i), f(\mathbf{x}_j)] = C(\mathbf{x}_i, \mathbf{x}_j) = v_0 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d(x_i^d - x_j^d)^2\right] \tag{5.3}$$

or

$$C(\mathbf{x}_i, \mathbf{x}_j) = v_0 \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right] \tag{5.4}$$

where $\mathbf{W}^{-1} = \mathrm{diag}[w_1 \ldots w_D]$ or $\mathbf{W} = \mathrm{diag}[1/w_1 \ldots 1/w_D]$. Which we can write

$$C(\mathbf{x}_i, \mathbf{x}_j) = \tau N_{\mathbf{x}_i}(\mathbf{x}_j, \mathbf{W}) \tag{5.5}$$

with $\tau = (2\pi)^{D/2}|\mathbf{W}|^{1/2}v_0$.

If we are now looking at the covariance between the derivative point $f'_d(\mathbf{x}_i) = \frac{\partial f(\mathbf{x}_i)}{\partial x_i^d}$, i.e. d^{th} component of the derivative at $\mathbf{x}_i$, and $f(\mathbf{x}_j)$, we have

$$\mathrm{Cov}[f'_d(\mathbf{x}_i), f(\mathbf{x}_j)] = \frac{\partial C(\mathbf{x}_i, \mathbf{x}_j)}{\partial x_i^d} = -w_d(x_i^d - x_j^d)C(\mathbf{x}_i, \mathbf{x}_j) \tag{5.6}$$

That we can also write

$$C'_{x_i^d}(\mathbf{x}_i, \mathbf{x}_j) = -\tau w_d(x_i^d - x_j^d)N_{\mathbf{x}_i}(\mathbf{x}_j, \mathbf{W}) \tag{5.7}$$

5.1.1 Predictive mean and variance corresponding to a new x for function points

The case with the Gaussian covariance function has already been described and results were as follows.

Let the vector of covariances between function points and test points be denoted as $\mathbf{k}_\mathbf{x}$ with its i^{th} component $k_{i\mathbf{x}}(\mathbf{x})$, then the predictive mean corresponding to a new $\mathbf{x}$ is given by

$$\mu_1(\mathbf{x}) = \sum_i \beta_i k_{i\mathbf{x}}(\mathbf{x}) = \tau \sum_i \beta_i N_\mathbf{x}(\mathbf{x}_i, \mathbf{W}) \tag{5.8}$$

and the predictive variance by

$$\begin{aligned} \sigma_1^2(\mathbf{x}) &= k(\mathbf{x}) - \sum_{i,j} K_{ij}^{-1} k_{i\mathbf{x}}(\mathbf{x}) k_{j\mathbf{x}}(\mathbf{x}) && (5.9) \\ &= v - \tau^2 \sum_{i,j}^N K_{ij}^{-1} N_\mathbf{x}(\mathbf{x}_i, \mathbf{W}) N_\mathbf{x}(\mathbf{x}_j, \mathbf{W}) && (5.10) \end{aligned}$$

5.1.2 Predictive mean and variance corresponding to a new x for derivative points

If the training points consist of the d^{th} components of the derivatives at $\mathbf{x}_1, \ldots, \mathbf{x}_N$, $\{f'_d(\mathbf{x}_i)\}_{i=1}^N$, then, the i^{th} component of the vector $\mathbf{k}'_\mathbf{x}$, giving the covariances between these training points and the test input, is given by

$$k'_{i\mathbf{x}}(\mathbf{x}) = C'_{x_i^d}(\mathbf{x}_i, \mathbf{x}) = -\tau w_d(x_i^d - x^d)N_\mathbf{x}(\mathbf{x}_i, \mathbf{W}) \tag{5.11}$$

and the covariance between the new test input and itself is

$$k(\mathbf{x}) = C'_{x_i^d}(\mathbf{x}, \mathbf{x}) = 0 \tag{5.12}$$

Therefore, the predictive mean is given by

$$\mu_d(\mathbf{x}) = \sum_i \beta_i k'_{i\mathbf{x}}(\mathbf{x}) = -\tau \sum_{d=1}^D w_d \sum_i \beta_i (x_i^d - x^d) N_\mathbf{x}(\mathbf{x}_i, \mathbf{W}) \tag{5.13}$$

and the predictive variance by

$$\begin{aligned}\sigma_d^2(\mathbf{x}) &= k(\mathbf{x}) - \sum_{i,j} K_{ij}^{-1} k'_{i\mathbf{x}}(\mathbf{x}) k'_{j\mathbf{x}}(\mathbf{x}) & (5.14)\\ &= -\tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} K_{ij}^{-1}(x_i^e - x^e)(x_j^d - x^d) N_{\mathbf{x}}(\mathbf{x}_i, W) N_{\mathbf{x}}(\mathbf{x}_j, W) & (5.15)\end{aligned}$$

where e and d denote indices of different derivative components. This results from a product of two covariance vectors each containing covariances for all derivative components.

5.2 Predictive mean for a new $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma}_x)$

We need to compute $m(\mathbf{u}, \mathbf{\Sigma}_x) = E_{\mathbf{x}}[\mu(\mathbf{x})]$, that is,

$$m(\mathbf{u}, \mathbf{\Sigma}_x) = E_{\mathbf{x}}[\mu(\mathbf{x})] = \int \mathbf{k}(\mathbf{x})^T \boldsymbol{\beta} p(\mathbf{x}) d\mathbf{x} \tag{5.16}$$

where $\mathbf{k}(\mathbf{x})$ is covariance between entire input data and test vector and can be written as

$$\mathbf{k}(\mathbf{x}) = \begin{bmatrix} \mathrm{Cov}(f(\mathbf{x}_i), f(\mathbf{x})) \\ \mathrm{Cov}(f'(\mathbf{x}_i), f(\mathbf{x})) \end{bmatrix} = \begin{bmatrix} \mathbf{k}_{\mathbf{x}} \\ \mathbf{k}'_{\mathbf{x}} \end{bmatrix} \tag{5.17}$$

where $\mathbf{k}'_{\mathbf{x}}$ denotes covariance between input and test points for all derivative components.

$$m(\mathbf{u}, \mathbf{\Sigma}_x) = E_{\mathbf{x}}[\mu_1(\mathbf{x})] + E_{\mathbf{x}}[\mu_d(\mathbf{x})] = \begin{bmatrix} \int \mathbf{k}_{\mathbf{x}}^T \boldsymbol{\beta}_{(1)} p(\mathbf{x}) d\mathbf{x} \\ \int \mathbf{k}_{\mathbf{x}}'^T \boldsymbol{\beta}_{(d)} p(\mathbf{x}) d\mathbf{x} \end{bmatrix} \tag{5.18}$$

where $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_{(1)} \\ \boldsymbol{\beta}_{(d)} \end{bmatrix}$

Let us solve the integrals separately. The indices $_{(\cdot)}$ will be from now on left out for notational convenience.

The solution for the part representing function points is known

$$\boxed{E_{\mathbf{x}}[\mu_1(\mathbf{x})] = m_1(\mathbf{u}, \mathbf{\Sigma}_x) = \textstyle\sum_i \beta_i C_{mod_1}(\mathbf{u}, \mathbf{x}_i)} \tag{5.19}$$

And for the part representing derivative points is as follows.

We need to compute $m_d(\mathbf{u}, \mathbf{\Sigma}_x) = E_{\mathbf{x}}[\mu_d(\mathbf{x})]$, that is,

$$\begin{aligned}E_{\mathbf{x}}[\mu_d(\mathbf{x})] &= -\tau \sum_{d=1}^{D} w_d \sum_i \beta_i \int (x_i^d - x^d) N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} \\ &= -\tau \sum_{d=1}^{D} w_d \sum_i \beta_i \left[x_i^d l_i^1 - l_i^2 \right] & (5.20)\end{aligned}$$

with

$$\begin{aligned} l_i^1 &= \int N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} & (5.21)\\ l_i^2 &= \int x^d N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} & (5.22)\end{aligned}$$

where $p(\mathbf{x}) = \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x)$.

For both integrals, we need to compute the product of $N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W})$ with $p(\mathbf{x}) = \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x)$. We have

$$N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W})\mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x) = \mathbf{z}_i N_{\mathbf{x}}(\mathbf{c}_i, \mathbf{C}) \tag{5.23}$$

with

$$\begin{aligned} \mathbf{C} &= (\mathbf{W}^{-1} + \boldsymbol{\Sigma}_x^{-1})^{-1} \\ \mathbf{c}_i &= \mathbf{C}(\mathbf{W}^{-1}\mathbf{x}_i + \boldsymbol{\Sigma}_x^{-1}\mathbf{u}) \\ \mathbf{z}_i &= \mathcal{N}_{\mathbf{u}}(\mathbf{x}_i, \mathbf{W} + \boldsymbol{\Sigma}_x) \end{aligned} \tag{5.24}$$

Programming of expression for $\mathbf{c}_i$ is made easier and avoids cases when $\boldsymbol{\Sigma}_x$ is rank deficient if Matrix Inversion Lemma ($(\mathbf{A}+\mathbf{X}\mathbf{B}\mathbf{X}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{X}(\mathbf{B}^{-1}+\mathbf{X}^T\mathbf{A}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{A}^{-1}$) and relation $(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ are used.

$$\begin{aligned} \mathbf{c}_i &= (\mathbf{x}_i\mathbf{W}^{-1} + \mathbf{u}\boldsymbol{\Sigma}_x^{-1})\mathbf{C} && (5.25) \\ &= (\mathbf{x}_i\mathbf{W}^{-1} + \mathbf{u}\boldsymbol{\Sigma}_x^{-1})(\mathbf{W}^{-1} + \boldsymbol{\Sigma}_x^{-1})^{-1} && (5.26) \\ &= \mathbf{x}_i\mathbf{W}^{-1}(\mathbf{W}^{-1} + \boldsymbol{\Sigma}_x^{-1})^{-1} + \mathbf{u}\boldsymbol{\Sigma}_x^{-1}(\mathbf{W}^{-1} + \boldsymbol{\Sigma}_x^{-1})^{-1} && (5.27) \\ &= \mathbf{x}_i\mathbf{W}^{-1}(\mathbf{W} - \mathbf{W}(\mathbf{W}+\boldsymbol{\Sigma})^{-1}\mathbf{W}) + \mathbf{u}(\mathbf{W}^{-1}\boldsymbol{\Sigma} + \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma})^{-1} && (5.28) \\ &= \mathbf{x}_i(\mathbf{I} - (\mathbf{W}+\boldsymbol{\Sigma})^{-1}\mathbf{W}) + \mathbf{u}(\mathbf{W}^{-1}\boldsymbol{\Sigma} + \mathbf{I})^{-1} && (5.29) \end{aligned}$$

So that

$$\begin{aligned} l_i^1 &= \mathbf{z}_i \int N_{\mathbf{x}}(\mathbf{c}_i, \mathbf{C})d\mathbf{x} = \mathbf{z}_i && (5.30) \\ l_i^2 &= \mathbf{z}_i \int x^d N_{\mathbf{x}}(\mathbf{c}_i, \mathbf{C})d\mathbf{x} = \mathbf{z}_i c_i^d && (5.31) \end{aligned}$$

where c_i^d is the d^{th} component of $\mathbf{c}_i$.

We then have

$$E_{\mathbf{x}}[\mu_d(\mathbf{x})] = -\tau \sum_{d=1}^{D} w_d \sum_i \beta_i \left[x_i^d l_i^1 - l_i^2\right] = -\tau \sum_{d=1}^{D} w_d \sum_i \beta_i \mathbf{z}_i [x_i^d - c_i^d] \tag{5.32}$$

Replacing $\mathbf{z}_i$ by its expression,

$$m_d(\mathbf{u}, \boldsymbol{\Sigma}_x) = -\tau \sum_{d=1}^{D} w_d \sum_i \beta_i \mathcal{N}_{\mathbf{u}}(\mathbf{x}_i, \mathbf{W} + \boldsymbol{\Sigma}_x)[x_i^d - c_i^d] \tag{5.33}$$

or

$$\boxed{E_{\mathbf{x}}[\mu_d(\mathbf{x})] = -\textstyle\sum_{d=1}^{D} w_d \sum_i \beta_i (x_i^d - c_i^d) C_{mod_1}(\mathbf{x}_i, \mathbf{u})} \tag{5.34}$$

with c_i^d, d^{th} component of $(\mathbf{W}^{-1} + \boldsymbol{\Sigma}_x^{-1})^{-1}(\mathbf{W}^{-1}\mathbf{x}_i + \boldsymbol{\Sigma}_x^{-1}\mathbf{u})$ and $C_{mod_1}(\mathbf{x}_i, \mathbf{u}) = \tau\mathbf{z}_i$, that is

$$C_{mod_1}(\mathbf{x}_i, \mathbf{u}) = v_0 |\mathbf{I} + \mathbf{W}^{-1}\boldsymbol{\Sigma}_x|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{u})^T(\mathbf{W} + \boldsymbol{\Sigma}_x)^{-1}(\mathbf{x}_i - \mathbf{u})\right] \tag{5.35}$$

Case of diagonal $\boldsymbol{\Sigma}_x$ If $\boldsymbol{\Sigma}_x = \text{diag}[v_{x1} \ldots v_{xD}]$, we have $\mathbf{W} + \boldsymbol{\Sigma}_x = \text{diag}[w_1^{-1} + v_{x1} \ldots w_D^{-1} + v_{xD}]$, so that we can write

$$C_{mod_1}(\mathbf{x}_i, \mathbf{u}) = v_0 \left(\prod_{d=1}^{D} (1 + w_d v_{xd}) \right)^{-1/2} \exp \left[-\frac{1}{2} \sum_{d=1}^{D} (w_d^{-1} + v_{xd})^{-1} (x_i^d - u^d)^2 \right] \tag{5.36}$$

The final expression for $m(\mathbf{u}, \boldsymbol{\Sigma}_x)$ is

$$\boxed{m(\mathbf{u}, \boldsymbol{\Sigma}_x) = E_{\mathbf{x}}[\mu_1(\mathbf{x})] + E_{\mathbf{x}}[\mu_d(\mathbf{x})]} \tag{5.37}$$

5.3 Predictive variance for a new $\mathbf{x} \sim \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_x)$

It is given by

$$v(\mathbf{u}, \boldsymbol{\Sigma}_x) = E_{\mathbf{x}}[\sigma^2(\mathbf{x})] + E_{\mathbf{x}}[\mu(\mathbf{x})^2] - E_{\mathbf{x}}[\mu(\mathbf{x})]^2 \tag{5.38}$$

and we already have $E_{\mathbf{x}}[\mu(\mathbf{x})]^2 = m(\mathbf{u}, \boldsymbol{\Sigma}_x)^2$.

The expression for overall $E_{\mathbf{x}}[\sigma^2(\mathbf{x})]$ can be written as

$$\begin{aligned}
E_{\mathbf{x}}[\sigma^2(\mathbf{x})] &= v - \int \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\
&= v - \int \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} \\
&= v - \int \begin{bmatrix} \mathbf{k}_{\mathbf{x}}^T \mathbf{K}_{ij(1)}^{-1} \mathbf{k}_{\mathbf{x}} & \mathbf{k}_{\mathbf{x}}^T \mathbf{K}_{ij(1d)}^{-1} \mathbf{k}'_{\mathbf{x}} \\ \mathbf{k}_{\mathbf{x}}^{T'} \mathbf{K}_{ij(1d)}^{-1} \mathbf{k}_{\mathbf{x}} & \mathbf{k}_{\mathbf{x}}^{T'} \mathbf{K}_{ij(d)}^{-1} \mathbf{k}'_{\mathbf{x}} \end{bmatrix} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} \\
&= v - \begin{bmatrix} \int \mathbf{k}_{\mathbf{x}}^T \mathbf{K}_{ij(1)}^{-1} \mathbf{k}_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} & \int \mathbf{k}_{\mathbf{x}}^T \mathbf{K}_{ij(1d)}^{-1} \mathbf{k}'_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} \\ \int \mathbf{k}_{\mathbf{x}}^{T'} \mathbf{K}_{ij(1d)}^{-1} \mathbf{k}_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} & \int \mathbf{k}_{\mathbf{x}}^{T'} \mathbf{K}_{ij(d)}^{-1} \mathbf{k}'_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} \end{bmatrix}
\end{aligned} \tag{5.39}$$

where $\mathbf{K}^{-1} = \begin{bmatrix} \mathbf{K}_{ij(1)}^{-1} & \mathbf{K}_{ij(1d)}^{-1} \\ \mathbf{K}_{ij(1d)}^{-1} & \mathbf{K}_{ij(d)}^{-1} \end{bmatrix}$

As we again calculate the integrals separately we will be, from the reason of convenience, using the following notation

$$E_{\mathbf{x}}[\sigma_1^2(\mathbf{x})] = -\int \mathbf{k}_{\mathbf{x}}^T \mathbf{K}_{ij(1)}^{-1} \mathbf{k}_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} \tag{5.40}$$

$$E_{\mathbf{x}}[\sigma_{1d}^2(\mathbf{x})] = -\int \mathbf{k}_{\mathbf{x}}^T \mathbf{K}_{ij(1d)}^{-1} \mathbf{k}'_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} \tag{5.41}$$

$$E_{\mathbf{x}}[\sigma_d^2(\mathbf{x})] = -\int \mathbf{k}_{\mathbf{x}}^{T'} \mathbf{K}_{ij(d)}^{-1} \mathbf{k}'_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \boldsymbol{\Sigma}_{\mathbf{x}}) d\mathbf{x} \tag{5.42}$$

Be aware that $\mathbf{K}_{ij(\cdot)}^{-1}$ means only the corresponding part of the overall inverse covariance matrix $\mathbf{K}^{-1}$. The indices $_{(\cdot)}$ will be from now on left out for notational convenience.

The expression for $\underline{E_{\mathbf{x}}[\sigma_1^2(\mathbf{x})]}$ is

$$\boxed{E_{\mathbf{x}}[\sigma_1^2(\mathbf{x})] = \tau^2 \textstyle\sum_{i,j} K_{ij}^{-1} \left[N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) \mathcal{N}_{\mathbf{u}} \left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \boldsymbol{\Sigma}_x + \frac{\mathbf{W}}{2} \right) \right]} \tag{5.43}$$

Replacing $\underline{\sigma_d^2(\mathbf{x})}$ by its expression (equation (5.15)) and taking care of the case when we are looking at the covariance of different d^{th} components, we need to compute

$$\begin{aligned}
E_{\mathbf{x}}[\sigma_d^2(\mathbf{x})] &= -\tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} K_{ij}^{-1} \int (x_i^e - x^e)(x_j^d - x^d) N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} \\
&= -\tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} K_{ij}^{-1} \int (x_i^e x_j^d - x_i^e x^d - x^e x_j^d + x^e x^d) N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} \\
&= -\tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} K_{ij}^{-1} [x_i^e x_j^d L_{ij}^1 - x_i^e L_{ij}^{2d} - x_j^d L_{ij}^{2e} + L_{ij}^3] \qquad (5.44)
\end{aligned}$$

with

$$\begin{aligned}
L_{ij}^1 &= \int N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} && (5.45) \\
L_{ij}^{2d} &= \int x^d N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} && (5.46) \\
L_{ij}^3 &= \int x^e x^d N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} && (5.47)
\end{aligned}$$

Using (3.9), we have $N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) = N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{x}} \left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} \right)$. And again, the product $N_{\mathbf{x}} \left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} \right)$ with $p(\mathbf{x}) = \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma}_x)$ is

$$N_{\mathbf{x}} \left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} \right) \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma}_x) = \mathbf{z}_{ij} N_{\mathbf{x}}(\mathbf{c}_{ij}, \mathbf{C}) \qquad (5.48)$$

with

$$\begin{aligned}
\mathbf{z}_{ij} &= N_{\mathbf{u}} \left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_x \right) \\
\mathbf{c}_{ij} &= \mathbf{C} \left(\left(\frac{\mathbf{W}}{2} \right)^{-1} \frac{\mathbf{x}_i + \mathbf{x}_j}{2} + \mathbf{\Sigma}_x^{-1} \mathbf{u} \right) \qquad (5.49) \\
\mathbf{C} &= \left(\left(\frac{\mathbf{W}}{2} \right)^{-1} + \mathbf{\Sigma}_x^{-1} \right)^{-1}
\end{aligned}$$

Again, as in the expression (5.29) we can simplify calculation of c_{ij} to

$$c_{ij} = \frac{\mathbf{x}_i + \mathbf{x}_j}{2} (\mathbf{I} - (\frac{\mathbf{W}}{2} + \mathbf{\Sigma})^{-1} \frac{\mathbf{W}}{2}) + \mathbf{u}((\frac{\mathbf{W}}{2})^{-1} \mathbf{\Sigma} + \mathbf{I})^{-1} \qquad (5.50)$$

So that we have

$$\begin{aligned}
L_{ij}^1 &= N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) \mathbf{z}_{ij} \int N_{\mathbf{x}}(\mathbf{c}_{ij}, \mathbf{C}) d\mathbf{x} = N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) \mathbf{z}_{ij} && (5.51) \\
L_{ij}^{2d} &= N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) \mathbf{z}_{ij} \int x^d N_{\mathbf{x}}(\mathbf{c}_{ij}, \mathbf{C}) d\mathbf{x} = N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) \mathbf{z}_{ij} c_{ij}^d && (5.52) \\
L_{ij}^3 &= N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) \mathbf{z}_{ij} \int x^e x^d N_{\mathbf{x}}(\mathbf{c}_{ij}, \mathbf{C}) d\mathbf{x} = N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) \mathbf{z}_{ij} (C_{ed} + c_{ij}^e c_{ij}^d) && (5.53)
\end{aligned}$$

where c_{ij}^d is the d^{th} component of $\mathbf{c}_{ij}$ with regard to index i and C_{ed} is the (e, d) entry of the $D \times D$ matrix $\mathbf{C}$ with regard to temporary indices.

Therefore,

$$\begin{aligned} E_{\mathbf{x}}[\sigma_d^2(\mathbf{x})] &= -\tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} K_{ij}^{-1}[x_i^e x_j^d L_{ij}^1 - (x_i^e + x_j^d)L_{ij}^2 + L_{ij}^3] \\ &= -\tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} K_{ij}^{-1} N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W})\mathbf{z}_{ij}[x_i^e x_j^d - (x_i^e c_{ij}^d + x_j^d c_{ij}^e) + C_{ed} + c_{ij}^e c_{ij}^d] \end{aligned} \tag{5.54}$$

and replacing $\mathbf{z}_{ij}$ by its expression, we have

$$\begin{aligned} &E_{\mathbf{x}}[\sigma_d^2(\mathbf{x})] = -\tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} K_{ij}^{-1} N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}}\left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_x\right) \\ &[x_i^e x_j^d - (x_i^e c_{ij}^d + x_j^d c_{ij}^e) + C_{ed} + c_{ij}^e c_{ij}^d] \end{aligned} \tag{5.55}$$

Let $C_{mod_2}(\mathbf{x}_i, \mathbf{x}_j) = \tau N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W})$. We have

$$\begin{aligned} C_{mod_2}(\mathbf{x}_i, \mathbf{x}_j) &= v_0 2^{-\frac{D}{2}} \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T (\frac{\mathbf{W}}{2})^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right] \\ &= v_0 2^{-\frac{D}{2}} \exp\left[-\frac{1}{2}\sum_{d=1}^{D} \frac{w_d}{2}(x_i^d - x_j^d)^2\right] \end{aligned} \tag{5.56}$$

Also, let $C_{mod_3}(\mathbf{u}, \mathbf{x}_b) = \tau N_{\mathbf{u}}\left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_x\right)$, that is

$$C_{mod_3}(\mathbf{u}, \mathbf{x}_b) = v_0 \left|\frac{1}{2}\mathbf{I} + \mathbf{W}^{-1}\mathbf{\Sigma}_x\right|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{u} - \mathbf{x}_b)^T \left(\frac{\mathbf{W}}{2} + \mathbf{\Sigma}_x\right)^{-1} (\mathbf{u} - \mathbf{x}_b)\right] \tag{5.57}$$

where $\mathbf{x}_b = \frac{\mathbf{x}_i + \mathbf{x}_j}{2}$.

Note that in the special case where $\mathbf{\Sigma}_x$ is diagonal, we have

$$C_{mod_3}(\mathbf{u}, \mathbf{x}_b) = v_0 \left(\prod_{d=1}^{D}\left(\frac{1}{2} + w_d v_{xd}\right)\right)^{-1/2} \exp\left[-\frac{1}{2}\sum_{d=1}^{D}(w_d^{-1}/2 + v_{xd})^{-1}(u^d - x_b^d)^2\right] \tag{5.58}$$

We can then write

$$\boxed{\begin{aligned} &E_{\mathbf{x}}[\sigma_d^2(\mathbf{x})] = \\ &-\sum_{e,d=1}^{D} w_e w_d \sum_{i,j} K_{ij}^{-1}(x_i^e x_j^d - (x_i^e c_{ij}^d + x_j^d c_{ij}^e) + C_{ed} + c_{ij}^e c_{ij}^d) C_{mod_2}(\mathbf{x}_i, \mathbf{x}_j) C_{mod_3}(\mathbf{u}, \mathbf{x}_b) \end{aligned}} \tag{5.59}$$

with C_{ed} the (e, d) entry of $\left(\left(\frac{\mathbf{W}}{2}\right)^{-1} + \mathbf{\Sigma}_x^{-1}\right)^{-1}$, that is $C_{ed} = (2w_d + v_{xd})^{-1}$ and c_{ij}^d is the d^{th} element of $\mathbf{C}\left(\left(\frac{\mathbf{W}}{2}\right)^{-1} \frac{\mathbf{x}_i + \mathbf{x}_j}{2} + \mathbf{\Sigma}_x^{-1}\mathbf{u}\right)$ with $\mathbf{C} = \text{diag}[(2w_1 + v_{x1})^{-1} \dots (2w_D + v_{xD})^{-1}]$, in the case of a diagonal $\mathbf{\Sigma}_x$.

We approach the computation of $\underline{E_{\mathbf{x}}[\sigma_{1d}^2(\mathbf{x})]}$ in the similar way as before

$$\begin{aligned}
E_{\mathbf{x}}[\sigma_{1d}^2(\mathbf{x})] &= \tau^2 \sum_{d=1}^{D} w_d \sum_{i,j} K_{ij}^{-1} \int (x_i^d - x^d) N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} \\
&= \tau^2 \sum_{d=1}^{D} w_d \sum_{i,j} K_{ij}^{-1} \Bigg[x_i^d \int N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} \\
&\quad - \int x^d N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W}) N_{\mathbf{x}}(\mathbf{x}_j, \mathbf{W}) p(\mathbf{x}) d\mathbf{x} \Bigg] \\
&= \tau^2 \sum_{d=1}^{D} w_d \sum_{i,j} K_{ij}^{-1} N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) \mathbf{z}_{ij} [x_i^d - c_{ij}^d] \qquad (5.60)
\end{aligned}$$

and replacing $\mathbf{z}_{ij}$ by its expression, we have

$$E_{\mathbf{x}}[\sigma_{1d}^2(\mathbf{x})] = \tau^2 \sum_{d=1}^{D} w_d \sum_{i,j} K_{ij}^{-1} N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}}(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_x)[x_i^d - c_{ij}^d] \qquad (5.61)$$

Let $C_{mod_2}(\mathbf{x}_i, \mathbf{x}_j) = \tau N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W})$ as in equation (5.56) and $C_{mod_3}(\mathbf{x}_i, \mathbf{x}_j) = \tau N_{\mathbf{u}}(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}, \frac{\mathbf{W}}{2} + \mathbf{\Sigma}_x)$ as in equation (5.57).

We can then write

$$\boxed{E_{\mathbf{x}}[\sigma_{1d}^2(\mathbf{x})] = \textstyle\sum_{d=1}^{D} w_d \sum_{i,j} K_{ij}^{-1} [x_i^d - c_{ij}^d] C_{mod_2}(\mathbf{x}_i, \mathbf{x}_j) C_{mod_3}(\mathbf{u}, \mathbf{x}_b)} \qquad (5.62)$$

where $\mathbf{x}_b = \frac{\mathbf{x}_i + \mathbf{x}_j}{2}$.

Similarly for overall $E_{\mathbf{x}}[\mu(\mathbf{x})^2]$, we have

$$\begin{aligned}
E_{\mathbf{x}}[\mu(\mathbf{x})^2] &= \int \mathbf{k}(\mathbf{x})^T \boldsymbol{\beta}\boldsymbol{\beta}^T \mathbf{k}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\
&= \int \mathbf{k}(\mathbf{x})^T \boldsymbol{\beta}\boldsymbol{\beta}^T \mathbf{k}(\mathbf{x}) \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} \\
&= \begin{bmatrix} \mathbf{k}_{\mathbf{x}}^T \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1)}^T \mathbf{k}_{\mathbf{x}} & \mathbf{k}_{\mathbf{x}}^T \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1d)}^T \mathbf{k}'_{\mathbf{x}} \\ \mathbf{k}_{\mathbf{x}}^{T'} \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1d)}^T \mathbf{k}_{\mathbf{x}} & \mathbf{k}_{\mathbf{x}}^{T'} \boldsymbol{\beta}\boldsymbol{\beta}_{ij(d)}^T \mathbf{k}'_{\mathbf{x}} \end{bmatrix} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} \\
&= \begin{bmatrix} \int \mathbf{k}_{\mathbf{x}}^T \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1)}^T \mathbf{k}_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} & \int \mathbf{k}_{\mathbf{x}}^T \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1d)}^T \mathbf{k}'_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} \\ \int \mathbf{k}_{\mathbf{x}}^{T'} \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1d)}^T \mathbf{k}_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} & \int \mathbf{k}_{\mathbf{x}}^{T'} \boldsymbol{\beta}\boldsymbol{\beta}_{ij(d)}^T \mathbf{k}'_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} \end{bmatrix} \qquad (5.63)
\end{aligned}$$

where $\boldsymbol{\beta}\boldsymbol{\beta}^T = \begin{bmatrix} \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1)}^T & \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1d)}^T \\ \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1d)}^T & \boldsymbol{\beta}\boldsymbol{\beta}_{ij(d)}^T \end{bmatrix}$

And we again calculate the integrals separately we will be, from the reason of convenience, using the following notation

$$\begin{aligned}
E_{\mathbf{x}}[\mu_1(\mathbf{x})^2] &= \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1)}^T \int \mathbf{k}_{\mathbf{x}}^T \mathbf{k}_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} \qquad (5.64) \\
E_{\mathbf{x}}[\mu_{1d}(\mathbf{x})^2] &= \boldsymbol{\beta}\boldsymbol{\beta}_{ij(1d)}^T \int \mathbf{k}_{\mathbf{x}}^T \mathbf{k}'_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} \qquad (5.65) \\
E_{\mathbf{x}}[\mu_d(\mathbf{x})^2] &= \boldsymbol{\beta}\boldsymbol{\beta}_{ij(d)}^T \int \mathbf{k}_{\mathbf{x}}^{T'} \mathbf{k}'_{\mathbf{x}} \mathcal{N}_{\mathbf{x}}(\mathbf{u}, \mathbf{\Sigma_x}) d\mathbf{x} \qquad (5.66)
\end{aligned}$$

Be aware that $\boldsymbol{\beta}\boldsymbol{\beta}_{ij(\cdot)}^T$ again means only the corresponding part of the overall $\boldsymbol{\beta}\boldsymbol{\beta}^T$. The indices $_{(\cdot)}$ will be from now on left out for notational convenience.

Expression for $\underline{E_{\mathbf{x}}[\mu_1(\mathbf{x})^2]}$ is

$$\boxed{E_{\mathbf{x}}[\mu_1(\mathbf{x})^2] = \tau^2 \textstyle\sum_{i,j} \beta_i\beta_j \left[N_{\mathbf{x}_i}(\mathbf{x}_j, 2\mathbf{W}) N_{\mathbf{u}}\left(\frac{\mathbf{x}_i+\mathbf{x}_j}{2}, \mathbf{\Sigma}_x + \frac{\mathbf{W}}{2}\right)\right]} \tag{5.67}$$

$\underline{E_{\mathbf{x}}[\mu_d(\mathbf{x})^2]}$ can be calculated as

$$\begin{aligned} E_{\mathbf{x}}[\mu_d(\mathbf{x})^2] &= \tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} \beta_i\beta_j \int (x_i^e - x^e)(x_j^d - x^d) N_{\mathbf{x}}(\mathbf{x}_i, W) N_{\mathbf{x}}(\mathbf{x}_j, W) p(\mathbf{x}) d\mathbf{x} \\ &= \tau^2 \sum_{e,d=1}^{D} w_e w_d \sum_{i,j} \beta_i\beta_j [x_i^e x_j^d L_{ij}^1 - x_i^e L_{ij}^{2d} - x_j^d L_{ij}^{2e} + L_{ij}^3] \end{aligned} \tag{5.68}$$

$$\boxed{\begin{aligned} &E_{\mathbf{x}}[\mu_d(\mathbf{x})^2] = \sum_{e,d=1}^{D} w_e w_d \\ &\sum_{i,j} \beta_i\beta_j (x_i^e x_j^d - (x_i^e c_{ij}^d + x_j^d c_{ij}^e) + C_{ed} + c_{ij}^e c_{ij}^d) C_{mod_2}(\mathbf{x}_i, \mathbf{x}_j) C_{mod_3}(\mathbf{u}, \mathbf{x}_b) \end{aligned}} \tag{5.69}$$

Similarly for $\underline{E_{\mathbf{x}}[\mu_{1d}(\mathbf{x})^2]}$, we have

$$E_{\mathbf{x}}[\mu_{1d}(\mathbf{x})^2] = -\tau^2 \sum_{d=1}^{D} w_d \sum_{i,j} \beta_i\beta_j \int (x_i^d - x^d) N_{\mathbf{x}}(\mathbf{x}_i, W) N_{\mathbf{x}}(\mathbf{x}_j, W) p(\mathbf{x}) d\mathbf{x} \tag{5.70}$$

$$\boxed{E_{\mathbf{x}}[\mu_{1d}(\mathbf{x})^2] = -\textstyle\sum_{d=1}^{D} w_d \sum_{i,j} \beta_i\beta_j [x_i^d - c_{ij}^d] C_{mod_2}(\mathbf{x}_i, \mathbf{x}_j) C_{mod_3}(\mathbf{u}, \mathbf{x}_b)} \tag{5.71}$$

So that finally, the predictive variance is expressed as

$$\boxed{\begin{aligned} v(\mathbf{u}, \mathbf{\Sigma}_x) = v &+ E_{\mathbf{x}}[\sigma_1^2(\mathbf{x})] + 2E_{\mathbf{x}}[\sigma_{1d}^2(\mathbf{x})] + E_{\mathbf{x}}[\sigma_d^2(\mathbf{x})] \\ &+ E_{\mathbf{x}}[\mu_1(\mathbf{x})^2] + 2E_{\mathbf{x}}[\mu_{1d}(\mathbf{x})^2] + E_{\mathbf{x}}[\mu_d(\mathbf{x})^2] - m(\mathbf{u}, \mathbf{\Sigma}_x)^2 \end{aligned}} \tag{5.72}$$

5.4 Application to multi-step ahead iterative prediction

We wish to apply these results to the multi-step ahead prediction task of dynamic system response. We focus on the iterative approach, which consists in making repeated one-step ahead predictions up to the desired horizon. As we do so, we also suggest to propagate the uncertainty (induced by the successive predictions) as we predict ahead in time.

In the following, we consider the time series $y_{t_1}, \dots, y_t$ and assume the following state-space model

$$\begin{aligned} y_t &= f(\mathbf{x}_t) + \epsilon \quad \text{with} \\ \mathbf{x}_t &= [y_{t-1}, \dots, y_{t-L}]^T \end{aligned} \tag{5.73}$$

where ϵ is a white noise with variance v.

Assuming the time-series is known up to time t, we wish to predict k steps ahead: that is to say, to find the predictive distribution of y_{t+k} corresponding to $\mathbf{x}_{t+k} = [y_{t+k-1}, \dots, y_{t+k-L}]^T$.

The naive way of doing so is by considering $\mathbf{x}_{t+k} = [\hat{y}_{t+k-1}, \dots, \hat{y}_{t+k-L}]^T$, where $\hat{y}_{t+k-i}$ is the point estimate of y_{t+k-i}.

Using the results derived in the previous section, we propose to predict k-steps by propagating the uncertainty as we predict ahead in time so that now, each y_{t+k-i} is a Gaussian random variable, with known mean and variance, so that we have an $L \times 1$ random state $\mathbf{x}_{t+k} = [y_{t+k-1}, \dots, y_{t+k-L}]^T$, which correponds to $\mathbf{x} \sim \mathcal{N}(\mathbf{u}, \mathbf{\Sigma}_x)$ of the previous section.

Here is a sketch of what we do:

- $t+1$: $\mathbf{x}_{t+1} = [y_t, \dots, y_{t-L}]^T$. Since the time-series is known up to time t, the predictive mean and variance of the corresponding y_{t+1} are simply given by $\mu(\mathbf{x}_{t+1})$ and $\sigma^2(\mathbf{x}_{t+1})$, computed using (5.13) and (5.14) resp.
- $t+2$: $\mathbf{x}_{t+2} = [y_{t+1}, y_t, \dots, y_{t+1-L}]^T$. Now, we have $y_{t+1} \sim \mathcal{N}(\mu(\mathbf{x}_{t+1}), \sigma^2(\mathbf{x}_{t+1}))$ so that we have

$$\mathbf{x}_{t+2} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_{t+1}) \\ y_t \\ \vdots \\ y_{t+1-L} \end{bmatrix}, \begin{bmatrix} \sigma^2(x_{t+1}) & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} \right)$$

  Now, the predictive mean and variance of the corresponding y_{t+2} are given by $m(\mathbf{x}_{t+2})$ and $v(\mathbf{x}_{t+2})$, computed using (5.34) and (5.72) resp.
- $t+3$: $\mathbf{x}_{t+3} = [y_{t+2}, y_{t+1}, \dots, y_{t+2-L}]^T$, with $y_{t+1} \sim \mathcal{N}(\mu(\mathbf{x}_{t+1}), \sigma^2(\mathbf{x}_{t+1}))$ and $y_{t+2} \sim \mathcal{N}(m(\mathbf{x}_{t+2}), v(\mathbf{x}_{t+2}))$. We have

$$\mathbf{x}_{t+3} \sim \mathcal{N}\left(\begin{bmatrix} m(x_{t+2}) \\ \mu(x_{t+1}) \\ y_t \\ \vdots \\ y_{t+2-L} \end{bmatrix}, \begin{bmatrix} v(x_{t+2}) & \mathrm{Cov}[y_{t+2}, y_{t+1}] & 0 & \dots & 0 \\ \mathrm{Cov}[y_{t+1}, y_{t+2}] & \sigma^2(x_{t+1}) & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} \right)$$

  And the predictive mean and variance of the corresponding y_{t+3} are given by $m(\mathbf{x}_{t+3})$ and $v(\mathbf{x}_{t+3})$, computed using (5.34) and (5.72) resp.

etc, etc.

Finally, at $t+k$, we have $\mathbf{x}_{t+k} = [y_{t+k-1}, y_{t+k-2}, \dots, y_{t+k--L}]^T$ such that

$$x_{t+k} \sim \mathcal{N}\left(\begin{bmatrix} m(x_{t+k-1}) \\ m(x_{t+k-2}) \\ \dots \\ m(x_{t+k-L}) \end{bmatrix}, \begin{bmatrix} v(x_{t+k-1}) & \mathrm{Cov}[y_{t+k-1}, y_{t+k-2}] & \dots & \mathrm{Cov}[y_{t+k-1}, y_{t+k-L}] \\ \mathrm{Cov}[y_{t+k-2}, y_{t+k-1}] & v(x_{t+k-2}) & \dots & \mathrm{Cov}[y_{t+k-2}, y_{t+k-L}] \\ \dots & \dots & \dots & \dots \\ \mathrm{Cov}[y_{t+k-L}, y_{t+k-1}] & \mathrm{Cov}[y_{t+k-L}, y_{t+k-2}] & \dots & v(x_{t+k-L}) \end{bmatrix} \right)$$

and we finally get the corresponding $m(\mathbf{x}_{t+k})$ and $v(\mathbf{x}_{t+k})$, again computed using (5.34) and (5.72) resp.

5.4.1 Cross-covariance terms

As seen, at time $t+l$, we have the random input vector $x_{t+l} = [y_{t+l-1}, \dots, y_{t+l-L}]^T \sim \mathcal{N}(\mathbf{u}, \mathbf{\Sigma}_x)$ with mean $\mathbf{u}$ formed by the predictive mean of the lagged outputs $y_{t+l-\tau}$, $\tau = 1, \dots, L$, given by (5.13), or by

(5.34), depending on l, and the diagonal elements of the $L \times L$ input covariance matrix $\mathbf{\Sigma}_x$ contain the corresponding predictive variances.

We now need to compute the cross-covariance terms $\mathrm{Cov}[y_{t+l-i}, y_{t+l-j}]$ for $i, j = 1 \dots L$ with $i \neq j$. This corresponds to computing $\mathrm{Cov}[y_{t+l}, \mathbf{x}_{t+l}]$, disregarding the last (*oldest*) element of $\mathbf{x}_{t+l}$.

We then have

$$\mathrm{Cov}[y_{t+l}, \mathbf{x}_{t+l}) = E[y_{t+l}\mathbf{x}_{t+l}] - E[y_{t+l}]E[\mathbf{x}_{t+l}] \tag{5.74}$$

with $E[y_{t+l}] = m(\mathbf{x}_{t+l})$ given by (5.34) and we denote $E[\mathbf{x}_{t+l}]$ by $\mathbf{u}_{t+l}$, formed of the lagged predictive means. We have

$$\begin{aligned}
E[y_{t+l}\mathbf{x}_{t+l}] &= \int\int y_{t+l}\mathbf{x}_{t+l}p(y_{t+l}, \mathbf{x}_{t+l})dy_{t+l}d\mathbf{x}_{t+l} \\
&= \int\int y_{t+l}\mathbf{x}_{t+l}p(y_{t+l}|\mathbf{x}_{t+l})p(\mathbf{x}_{t+l})dy_{t+l}d\mathbf{x}_{t+l} \\
&= \int \mathbf{x}_{t+l}\mu(\mathbf{x}_{t+l})p(\mathbf{x}_{t+l})d\mathbf{x}_{t+l}
\end{aligned}$$

with

$$\begin{aligned}
\mu(\mathbf{x}_{t+l}) &= \sum_{i=1}^{N}\beta_i k_i(\mathbf{x}_{t+l}) + \sum_{i=N+1}^{N^1}\beta_i k_i^1(\mathbf{x}_{t+l}) + \cdots + \sum_{i=N^{D-1}+1}^{N^D}\beta_i k_i^D(\mathbf{x}_{t+l}) \\
&= \tau\sum_{i=1}^{N}\beta_i N_{\mathbf{x}_{t+l}}(\mathbf{x}_i, \mathbf{W}) - \tau\sum_{d=1}^{D} w_d \sum_{i=N^{d-1}+1}^{N^d}\beta_i(x_i^d - x_{t+l}^d)N_{\mathbf{x}_{t+l}}(\mathbf{x}_i, \mathbf{W})
\end{aligned} \tag{5.75}$$

Again we will not indicate the ranges of the i, d indices anymore. Always refer to (5.75). Let $y = y_{t+l}$ and $\mathbf{x} = \mathbf{x}_{t+l}$ for notational convenience. We need to solve

$$\begin{aligned}
E[y\mathbf{x}] &= \tau\sum_{i=1}^{N}\beta_i\int \mathbf{x}N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W})p(\mathbf{x})d\mathbf{x} - \tau\sum_{d=1}^{D} w_d\sum_{i=1}^{N}\beta_i\int \mathbf{x}(x_i^d - x^d)N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W})p(\mathbf{x})d\mathbf{x} \\
&= \tau\sum_{i=1}^{N}\beta_i l_i - \tau w_d\sum_{i=1}^{N}\beta_i(x_i^d l_i - ll_i)
\end{aligned}$$

with

$$l_i = \int \mathbf{x}N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W})p(\mathbf{x})d\mathbf{x} \tag{5.76}$$

$$ll_i = \int \mathbf{x}x^d N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W})p(\mathbf{x})d\mathbf{x} \tag{5.77}$$

As already seen in the previous section, we have $N_{\mathbf{x}}(\mathbf{x}_i, \mathbf{W})p(\mathbf{x}) = \mathbf{z}_i N_{\mathbf{x}}(\mathbf{c}_i, \mathbf{C})$ with $\mathbf{z}_i, \mathbf{c}_i, \mathbf{C}$ given by (5.24). So that we have

$$l_i = \mathbf{z}_i\int \mathbf{x}N_{\mathbf{x}}(\mathbf{c}_i, \mathbf{C})d\mathbf{x} = \mathbf{c}_i\mathbf{z}_i \tag{5.78}$$

$$ll_i = \mathbf{z}_i\int \mathbf{x}x^d N_{\mathbf{x}}(\mathbf{c}_i, \mathbf{C})d\mathbf{x} = (\mathbf{C}^{[d]} + \mathbf{c}^d)\mathbf{z}_i \tag{5.79}$$

where $\mathbf{C}^{[d]}$ is d^{th} column of matrix $\mathbf{C}$ and $\mathbf{c}^d$ such that i^{th} entry of $\mathbf{c}^d$ is $\mathbf{c}_i^d = \mathbf{c}_i c_i^d$;

$$\begin{aligned} E[y\mathbf{x}] &= \tau \sum_i \beta_i l_i \mathbf{c}_i - \tau \sum_d w_d \sum_i \beta_i (\mathbf{z}_i \mathbf{c}_i x_i^d - \mathbf{z}_i (\mathbf{C}^{[d]} + \mathbf{c}^d)) \\ &= \tau \sum_i \beta_i z_i \mathbf{c}_i - \tau \sum_d w_d \sum_i \beta_i \mathbf{z}_i (\mathbf{c}_i x_i^d - (\mathbf{C}^{[d]} + \mathbf{c}_i c_i^d)) \\ &= \sum_i \beta_i \mathbf{c}_i C_{mod_1} - \sum_d w_d \sum_i \beta_i (\mathbf{c}_i x_i^d - (\mathbf{C}^{[d]} + \mathbf{c}_i c_i^d)) C_{mod_1} \end{aligned} \tag{5.80}$$

and finally

$$\mathrm{Cov}(y_{T+k}, \mathbf{x}_{T+k}) = (\sum_i \beta_i \mathbf{c}_i C_{mod_1} - \sum_d w_d \sum_i \beta_i (\mathbf{c}_i x_i^d - (\mathbf{C}^{[d]} + \mathbf{c}_i c_i^d)) C_{mod_1}) - m(k-1)\mathbf{u}_{T+k} \tag{5.81}$$

Example

We use the same example as in the previous chapter. This time the input has double magnitude in comparison with before. The results can be seen in Figure 5.1. Propagation of uncertainty and mean causes that standard deviations are larger in some areas and mean value is also effected by the propagation as expected.

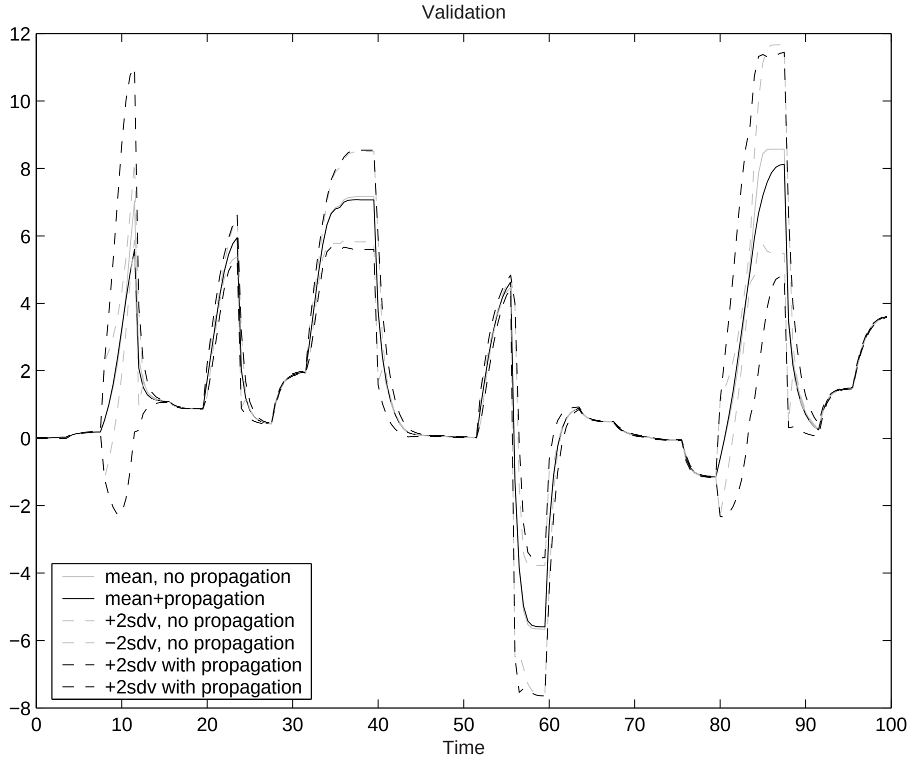

Figure 5.1: Response on validation data: GP model response without propagation of uncertainty - grey lines, GP model response with propagation of uncertainty - black lines

Chapter 6

Conclusions

This report shows how linear local models can be incorporated in GP models and contributes the derivation of uncertainty propagation through such models. Incorporation of derivative observations, obtained as coefficients of local linear models in equilibrium points with regular linear regression method, means joining local linear models and GP models. Local model networks have problem retaining local information when optimised to fit the process globally. GP models have problem with model dimensions when a lot of data is used for identification. Joining this two approaches can result in global models containing global and local information, of acceptable GP model dimensions and suited to the kind of data usually available in practice when carrying out experimental modelling (a lot of data in vicinity of equilibria points and few data far from equilibria).

The main contribution of our work is the derivation of prediction at a new random input for GP model with derivative observations which enables multi-step-ahead prediction (simulation) of such models. The obtained results still need to be validated with MCMC method and on examples of dynamic systems with simulation and experimentally.

Presented work is a part of widespread research activities on GP for dynamic systems in the world. These activities are at present mainly as follows:

- modelling with mixtures of models - local GP models with linear covariance functions and their optimisation based on blended (readily available) local variance information and not, as common, on data recorded out of equilibria points (at University College Cork, Cork);
- evaluation of GP models experimentally on measured data (Jozef Stefan Institute, Ljubljana and University of Glasgow, Glasgow);
- investigation of GP modelling tools for nonlinear systems structure identification - scheduling variable identification (at Hamilton Institute, Maynooth and University of Strathclyde, Glasgow) and identification of local models blending functions (at University College Cork, Cork);
- control using GP models and incorporating variance directly in cost function e.g. as a soft constraint (at University of Glasgow, Glasgow and Jozef Stefan Institute, Ljubljana);
- reinforcement learning of GP models for control, calculation of gradients for more efficient optimisation etc. (at Max Planck Institute, Tübingen);
- GP models predictive control and its semi-industrial application (Jozef Stefan Institute).

Acknowledgement

This work was made possible by EC funded Multi-Agent Control Research Training Network HPRN-CT-1999-00107 and Enterprise Ireland International Collaboration Grant IC/2003/25. Authors are grateful to William Leithead and Roderick Murray-Smith for very useful discussions.

Bibliography

[1] P. Abrahamsen, "A Review of Gaussian Random Fields and Correlation Functions", Report No. 917, http://publications.nr.no/917_Rapport.pdf, Norvegian Computer Centre, Oslo, 1997.

[2] A. Girard, C.E. Rasmussen, J. Quiñonero Candela, R. Murray-Smith, "Multi-step ahead prediction for non linear dynamic sytems - A Gaussian Process treatment with propagation of the uncertainty", *NIPS* 15, Vancouver, Canada, MIT Press, http://books.nips.cc/papers/files/nips15/AA06.pdf, 2003.

[3] A. Girard, "Gaussian process priors: analytical solutions for the prediction at a noisy input", DCS Technical Report, Department of Computing Science, Glasgow University, to appear, 2004.

[4] G. Gregorčič, G. Lightbody, "Internal model control based on a Gaussian process prior model", *Proceedings of ACC'2003*, Denver, CO, 4981-4986, 2003.

[5] J. Kocijan, A. Girard, B. Banko, R. Murray-Smith, "Dynamic Systems Identification with Gaussian Processes", *Proceedings of 4th Mathmod*, Vienna, 776-784, 2003, expanded version submitted for publication to *Mathematical and Computer Modelling of Dynamic Systems.*

[6] J. Kocijan, B. Likar, B. Banko, A. Girard, R. Murray-Smith, C.E. Rasmussen, "A case based comparison of identification with neural network and Gaussian process models", *Preprints of IFAC ICONS Conference*, Faro, 137-142, http://www.kyb.tuebingen.mpg.de/publication.html?user=carl, 2003.

[7] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, B. Likar, "Predictive control with Gaussian process models", In: *Eurocon*, Ljubljana, http://www.kyb.tuebingen.mpg.de/publication.html?user=carl, 2003.

[8] D. J. Leith, W. E. Leithead, E. Solak, R. Murray-Smith, "Divide & conquer identification: Using Gaussian process priors to combine derivative and non-derivative observations in a consistent manner", *Conference on Decision and Control 2002*, Las Vegas, 2002.

[9] R. Murray-Smith, T. A. Johansen, R. Shorten, "On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures", *European Control Conference*, Karlsruhe, BA-14, http://hamilton.may.ie/db/uploads/1002637263_link_f10207.pdf, 1999.

[10] R. Murray-Smith, A. Girard, "Gaussian Process priors with ARMA noise models", *Irish Signals and Systems Conference*, Maynooth, , 147-152, http://www.dcs.gla.ac.uk/%7Eagathe/reports.html, 2001.

[11] R.M. Neal, *Bayesian learning for neural networks*, Lecture notes in statistics, Springer Verlag, New York, 1996.

[12] A. O'Hagan, "On curve fitting and optimal design for regression (with discussion)", *Journal of the Royal Statistical Society B*, 40, 1-42, 1978.

[13] C.E.Rasmussen, "Evaluation of Gaussian Processes and other Methods for Non-Linear Regression", Ph.D. Disertation, Graduate department of Computer Science, University of Toronto, Toronto, http://www.kyb.tuebingen.mpg.de/publication.html?user=carl, 1996.

[14] E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, C. E. Rasmussen, "Derivative observations in Gaussian Process models of dynamic systems", *NIPS* 15, Vancouver, Canada, MIT Press, `http://books.nips.cc/papers/files/nips15/AA70.pdf`, 2003.

[15] C.K.I. Williams, "Prediction with Gaussian processes: From linear regression to linear prediction and beyond", In: *Learning in Graphical Models* (Edt.: Jordan, M.I.),Kluwer Academic, Dordrecht, 599-621, `http://www.dai.ed.ac.uk/homes/ckiw/online_pubs.html`, 1998.