Gaussian Process Models for Process Monitoring and Control

Javier Serradilla

Thesis submitted for the degree of
Doctor of Philosophy

School of Mathematics & Statistics
Newcastle University
Newcastle upon Tyne
United Kingdom

November 2012

Petra y Jose Antonio...

...vuestra generosidad y sacrificios diarios siempre han sido mi mejor universidad.

Acknowledgements

With life comes uncertainty, with science comes uncertainty. So, if we are to live with it we had better invest in understanding it. Working towards that endeavour I personally feel the need to mention Dr. J.Q. Shi. During my time in the School of Mathematics & Statistics at Newcastle University I have greatly benefited from his insightful comments and academic guidance; but I have also learned from his constructive criticism. His passion for Statistics is contagious and for that, his invaluable advice and all the time he has committed to me I am very grateful.

A special thought also goes for Nyree who has patiently and stoically waited for me at the end of those very long days in the school. Your resolute support and encouragement has made of this arduous journey a very pleasant walk.

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) and British Petroleum (BP).

Abstract

One problem of special interest both in industry and the engineering community is that of using the enormous amounts of data routinely generated and recorded in efficient process monitoring and control strategies. In statistical terms this is related to identifying those variables which exhibit *unwanted or unusual process variability* so that remedial action can be taken. To this end, a common approach in the literature is to reduce the problem dimensionality by using latent variable models. Customarily, the latent variables are a function of *all* of the original variables and monitoring is carried out in the reduced space.

Within this context, this thesis explores the development of models in which the latent factors are a *function of a subset*, only, of the original observations. By doing that, the advantages of monitoring in a reduced subspace are retained but there there are also additional gains in model interpretability. The idea arises from the *sparse* representation of the mapping matrix between latent and original variables in a linear factor analysis (FA) model. An extension of principal component analysis (PCA) to monitor nonlinear systems is proposed by using a a Gaussian Process Latent Variable model [Lawrence, 2005], GPLVM, as a starting point. Its application in a process control problem is also introduced. Using a Gaussian process, $\mathcal{GP}$, as the backbone, we define a Gaussian Process Functional Factor Analysis model which maps *subsets of the latent variables* to the observed data-space; a study of the model asymptotic properties is given. Several parameter inference methods as well as a model selection procedure via penalty functions are also proposed.

There are several scientific disciplines involved in the problem at hand. Chemical engineers refer to it as a sub-field of Process Control known as *Multivariate Statistical Process Control*. It is also an area of tremendous success in process *Chemometrics* where it has grown very rapidly over the last two decades. In Statistics, it touches the topics of *latent variable models* and *variable selection* methods. And within the Machine Learning community is classified as an *Unsupervised Learning* problem.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today's process industries have at their disposal a wealth of data which is routinely collected from online sensors every few seconds. All this information on dozens, hundreds of variables is stored in large databases and, if properly interpreted, could provide a detailed snapshot of the process behaviour over time. As Kourti [2002] argues, these data sets are often very large in size and contain variables which are generally highly correlated and with low signal-to-noise ratios (i.e. normally the information included in any single variable is small). In the past, rarely would anything be done with all this information due mainly to the intrinsic difficulty in handling it; however, over the last two decades there has been a dramatic increase and urge both in the scientific literature and industry to utilize these databases; the idea is to build data-driven models which, by disregarding the noise in the system, can handle the existing multicollinearity and extract the underlying latent variables which drive the process.

A *Statistical Process Control* (SPC) strategy is concerned with the monitoring of industrial processes over time with the aim to detect disturbances, *special cause* variation, and remove them from the system. It is said that a process is a *state of statistical control* when the only source of variability is *common cause* variation, or, in other words, the sort of variability which is unavoidable, which intrinsically affects the process all the time and cannot be removed [MacGregor and Kourti, 1995]. This variability reduction exercise differs from what is commonly referred to

as *Engineering Process Control* (EPC); the emphasis in EPC is on shifting variability from parts of the process where it could harm product quality/plant performance to those areas of the process where it can be tolerated [Montgomery and Keats, 1994]. These two strategies are not mutually exclusive; on the contrary, they complement one another.

The work presented here concentrates on the SPC side of process control; more specifically, it focuses on *Multivariate Statistical Process Control* (MSPC), where the emphasis is on the monitoring of processes in which several variables are of interest [Bersimis et al., 2006]. In the past, industries would make extensive use of univariate control charts (also known as Shewhart charts) where each variable of interest is monitored independently; there is, however, an inherent problem in treating them as though they were independent when in reality that is not the case and none of those variables define the process/product quality by itself. That not only can lead to poor monitoring strategies but it does neither take advantage of all the available information appropriately. The literature related to MSPC abounds and has grown dramatically over the past two decades. There are many good review papers which give an excellent introduction to the topic including those of Kourti and MacGregor [1995], Qin [2003], Kourti [2003], García-Muñoz et al. [2003] and MacGregor et al. [2005]. The text book by Chiang et al. [2001] offers an extensive account of MSPC and its applications.

As previously stated, the correlation between the process variables is usually of such a high degree that the resulting data matrices have a very low statistical rank. This fact makes latent variable models one of the most appropriate tools to obtain useful and simplified representations of the original data set. In this respect, MSPC not only is a subdiscipline of process control within the Chemical Engineering field. It is also an area of tremendous success in process Chemometrics[1] where it has grown very rapidly over the last two decades. In Statistics it touches the topics of latent variable models and variable selection methods [Hastie et al., 2009]. And within the Machine Learning community may be classified as an unsupervised learning problem [Ghahramani, 2004].

[1]Defined by the International Chemometrics Society as *the science of relating measurements made on a chemical system or process to the state of the system via application of mathematical or statistical methods*. See also Hibbert et al. [2009].

The theme of this thesis revolves around the idea that factor analysis approaches [Tipping and Bishop, 1999] can both be used (1) to speed up process monitoring schemes (fault detection) by constructing latent variables that are a subset of the full original variable space and (2) to facilitate the fault identification phase of the monitoring process. In this respect, the procedure is halfway between a *principal variables* approach [McCabe, 1984], which selects individual variables according to their relative importance and principal component analysis, PCA, based modelling techniques, which builds latent variables as linear combinations of the full original variable set.

In general terms, there is no simple procedure matching data generated in the process industries with a particular model. Notwithstanding the fact that most industrial systems behave non-linearly, linear models like PCA have continued to be used heavily in the area. This not only related to their relative simplicity from an application point of view but also from the pragmatism that arises from observing that non-linearities can be explained by considering additional, minor principal components[Kourti, 2002]. This approach, of course, sacrifices understanding of the data generative process in favour of model applicability.

Both linear and non-linear models will be subject to analysis in subsequent chapters. Non-linear systems will be modelled non-parametrically via a combination of a Gaussian Process and factor analysis-type of model. As for the remaining part of this chapter, it first introduces some notation and explains the workings of PCA focusing on its application to process monitoring; and secondly, it also presents an overview of Gaussian Processes, GPRs, as the backbone of the procedures which are proposed in this thesis.

1.1 Notation

Let D be the dimension of the data space, Q the dimension of the latent space and N the number of observations. The general $(N \times D)$ data matrix of observations will be denoted by $\mathbf{Y}$. The corresponding $(N \times Q)$ data matrix of latent variables will be denoted as $\mathbf{X}$. The i^{th} observation for the j^{th} variable will be written as y_{ij}

for $i = 1, \ldots, N$ and $j = 1, \ldots, D$; thus, we can refer to the whole data matrix as $\mathbf{Y} = (y_{ij})$.

The rows of $\mathbf{Y}$ will be written as $\boldsymbol{y}_1^\intercal, \boldsymbol{y}_2^\intercal, \ldots, \boldsymbol{y}_N^\intercal$. Therefore, $\boldsymbol{y}_i$ is the i^{th} observation for all D-variables and it is written as a column. Likewise, the columns of $\mathbf{Y}$ will be written with subscripts in parentheses as $\boldsymbol{y}_{(1)}, \boldsymbol{y}_{(2)}, \ldots, \boldsymbol{y}_{(D)}$. To summarize:

$$\mathbf{Y} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1D} \\ y_{21} & y_{22} & \cdots & y_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{ND} \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}_1^\intercal \\ \boldsymbol{y}_2^\intercal \\ \vdots \\ \boldsymbol{y}_N^\intercal \end{pmatrix} = (\boldsymbol{y}_{(1)}, \boldsymbol{y}_{(2)}, \ldots, \boldsymbol{y}_{(D)})$$

where

$$\boldsymbol{y}_i = \begin{pmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{iD} \end{pmatrix} \text{ for } (i = 1, \ldots, N), \quad \text{and } \boldsymbol{y}_{(d)} = \begin{pmatrix} y_{1d} \\ y_{2d} \\ \vdots \\ y_{Nd} \end{pmatrix} \text{ for } (d = 1, \ldots, D).$$

The notation for $\mathbf{X}$ is defined similarly:

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1Q} \\ x_{21} & x_{22} & \cdots & x_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NQ} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^\intercal \\ \boldsymbol{x}_2^\intercal \\ \vdots \\ \boldsymbol{x}_N^\intercal \end{pmatrix} = (\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, \ldots, \boldsymbol{x}_{(Q)}).$$

The idea behind the latent variable models that we are seeking to build is that the data we observe is simply the manifestation driven by a core subset Q of latent variables, where $Q \ll D$.

Finally, note that if the observations have been mean-centered, the data covariance matrix, $\mathbf{S}$, can be written as $N^{-1}\mathbf{Y}^\intercal\mathbf{Y}$ or, more generally, as $N^{-1}\mathbf{Y}^\intercal\mathbf{H}\mathbf{Y}$, where $\mathbf{H} = \mathbf{I}_N - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal$ is the centering matrix. Note that $\mathbf{I}_N$ is the $(N \times N)$ identity matrix and $\mathbf{1}$ is a column vector of N ones, Mardia et al. [1979].

1.2 Process monitoring review

Approaches to process monitoring, key variables (or *principal variables* as known in statistics), principal component analysis and performance monitoring charts are introduced here. The section ends with an example which aims at explaining how the process monitoring is carried out in practice in the latent variable space.

1.2.1 Mechanistic versus data-based modelling

There are two main approaches to process modelling. On the one hand, models can be built based on the underlying physics and chemistry laws that govern the behaviour of the process; this is referred to as *mechanistic modelling* and requires a thorough and extensive knowledge about the system under study. Very often, restrictions both in term of cost and time will simply prevent their development. On the other hand, a viable alternative is to use the data that is routinely collected from the process to build a *data-based* model. Whereas these models are much easier to develop, it is also true that the information that can be extracted from them is rather more limited. In many instances, the data-based methodology is used as a *black-box* where the user expects to extract a reliable prediction of how the system is behaving without having to worry about the inner workings of the true generative process.

1.2.2 Key variables

It is undoubtedly very appealing to simply not build a model and monitor the process variables individually. This is an ideal situation as fault detection is almost instantaneous and fault diagnosis is direct in the sense that the variable moving outside its confident limits is the variable developing a fault. But this situation is not practical: today's manufacturing processes measure and log hundreds of variables and therefore individual variable monitoring is unrealistic to say the least; it also ignores the fact that the correct functioning of the process depends on the *joint behaviour* of a set of variables and not on each variable individually [Kourti

and MacGregor, 1995]. Attempts can be made to remove the inessential variables and choose a subset of the original variables that contain, according to a specific criterion, as much information as possible. This gives rise to the concept of *principal variables* as introduced by McCabe [1984]. Exploiting this idea, Srinivasan and Qian [2007] have shown how a multi-state process could be monitored by just focusing on those variables whose behaviour is essential for the smooth running of the process; the variables most important from a monitoring perspective were termed as *key variables* by the authors. While the key-variable approach tackles the issue of dimensionality reduction via variable selection, it does not consider the problem of variable association that could lead to potential departures from normal plant behaviour.

1.2.3 Principal component analysis (PCA)

PCA is arguably the simplest dimensionality-reduction technique that can be applied to a set of correlated data; it is perhaps this simplicity which has contributed to is wide application within the MSPC area. Broadly speaking, the aim of PCA is to reduce the dimensionality of the process data by projecting it down to a latent variable space of lower dimensionality; once this linear transformation has been made, process monitoring is carried out in the reduced latent variable space. The purpose of this section is only to introduce the topic; Wold et al. [1987] provides an excellent explanation from a Chemometrics perspective. For further insights, the interested reader can refer to the monograph by Jolliffe [2002].

Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_D)^\intercal$ be the D-dimensional original variables (of which there are N observations) of process data and $\mathbf{S}$ the sample covariance matrix; let also $\mathbf{S} = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^\intercal$ be its spectral-decomposition where $\mathbf{P} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_D)$ is the matrix of eigenvectors and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, ..., \lambda_D)$ the corresponding matrix of eigenvalues ordered decreasingly, i.e. $\lambda_1 > \lambda_2 > \ldots > \lambda_D$. The basic idea behind PCA is to find a new set of variables $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)^\intercal$ such that the sample variances of the transformation are in decreasing order of magnitude and the $\boldsymbol{x}$-data are uncorrelated. The first principal component of $\boldsymbol{x}$ is $x_1 = \boldsymbol{p}_1^\intercal \boldsymbol{y}$, which is the linear combination of the $\boldsymbol{y}$-variables that has maximal variance amongst all linear combinations subject to the normalization constraint $\|\boldsymbol{p}_1\| = 1$. Likewise, the second principal component

is given by $x_2 = \boldsymbol{p}_2^\intercal \boldsymbol{y}$ and has maximal variance amongst all linear combinations subject to the constraints that it is uncorrelated with x_1 and $\|\boldsymbol{p}_2\| = 1$; additional principal components up to D are defined similarly. Finally, it is easily shown that the variance of the x_j^{th} principal component equals λ_j, i.e. the j^{th} largest eigenvalue of $\mathbf{S}$.

In reality, PCA decomposes the $N \times D$ matrix $\mathbf{Y}$ as the sum of D outer products

$$\mathbf{Y} = \sum_{i=1}^{D} \boldsymbol{x}_{(i)} \boldsymbol{p}_i^\intercal = \mathbf{X}_{N \times D} \mathbf{P}^\intercal,$$

where, as mentioned previously, the $\boldsymbol{x}_{(i)}, \boldsymbol{p}_i$ pairs (known as scores and loading vectors respectively) are ordered by the amount of variance captured. One feature of PCA is that, for linear systems, the less important components in terms of variance are often related to noise in the data. Then, if the process variables are highly correlated, Q principal components ($Q \ll D$) are enough to explain most of the data variability. In those cases, the PCA transformation is truncated after Q components and the remaining small variance factors are consolidated into a residual matrix, $\mathbf{E}$

$$\mathbf{Y} = \sum_{i=1}^{Q} \boldsymbol{x}_{(i)} \boldsymbol{p}_i^\intercal = \mathbf{X}_{N \times Q} \mathbf{P}_Q^\intercal + \mathbf{E},$$

where $\mathbf{P}_Q$ is the $D \times Q$ matrix of loadings vectors retained in the PCA model. From the previous equation, the fitted model values are given by

$$\widehat{\mathbf{Y}} = \mathbf{X}_{N \times Q} \mathbf{P}_Q^\intercal. \tag{1.1}$$

A final important consideration that has to be taken into account is how to determine Q, i.e. the number of principal components that the model is going to include. There is a variety of procedures that could be applied in this respect [Valle et al., 1999]. Very simple methods include the *SCREE test* (which selects k based on what percentage of the total variation is accounted for) and the *average eigenvalue approach* (which takes all those principals components whose eigenvalues are bigger than the average eigenvalue). There are also more complicated methods; for instance, the number of principal components can be selected using other commonly known model selection procedures like *cross-validation* as explained by Wold [1978]; see also Hastie et al. [2009, Chapter 7].

1.2.4 Fault detection: performance monitoring charts

The development of a process monitoring scheme using PCA begins by collecting *nominal process operational data*, i.e. the data generated when it is known that the process was behaving as expected. Once the nominal model has been constructed, new multivariate observations can be projected onto the latent variable subspace using Equation (1.1). Note that, in this respect, the eigenvector matrix $\mathbf{P}_Q$ acts as a linear map projecting the multivariate observations down from $\mathcal{R}^D$ to $\mathcal{R}^Q$. The new latent variables can then be monitored directly, in pairs or by using statistics which are derived from them.

Principal component scores

The principal component scores, $\boldsymbol{x}_i$, are linear combinations of the measurement variables, $\boldsymbol{y}_i$, and should be approximately normally distributed when the original observations are normally distributed. Assuming that the data matrix $\mathbf{Y}$ has been mean-centered, the scores from the PCA decomposition have mean zero with variance equal to their associated eigenvalue, λ_i. With this assumption of normality, upper and lower confidence limits at a significance level α are straightforward and can be calculated as follows

$$\pm z_{\alpha/2} \cdot \sqrt{\lambda_i},$$

where $z_{\alpha/2}$ is the critical value of the standard normal distribution at the $\alpha/2$ significance level. If no assumption is made about the distribution of the observations and the sample size is sufficiently large, confidence limits can also be calculated using the $(\alpha/2, 1-\alpha/2)$ quantiles of the *nominal data* or estimated via kernel density methods.

Bivariate plots of the principal component scores

As derived previously, the first principal component is $\boldsymbol{x}_{(1)} = \mathbf{Y}\boldsymbol{p}_1$ with variance λ_1, the second principal component is $\boldsymbol{x}_{(2)} = \mathbf{Y}\boldsymbol{p}_2$ with variance λ_2 and so on for the k^{th} principal component with variance λ_k. Restricting the analysis now to the first two principal components, an ellipsoidal control limit can be constructed as follows

$$\frac{x_{i1}^2}{\lambda_1} + \frac{x_{i2}^2}{\lambda_2} \leq \chi_2^2(\alpha),$$

which encloses all the pairs (y_{i1}, y_{i2}) whose statistical distance (Mahalanobis) from the mean, $\mathbf{0}$, is less or equal than $\chi^2_2(\alpha)$ with a probability α of committing a type I error [Johnson and Li, 2006]; $\chi^2_2(\alpha)$ is the upper critical value for a χ^2-distribution with two degrees of freedom at the α significance level.

Squared Prediction Error (SPE)

Once a PCA model is available, a future observation, $\boldsymbol{y}_i^*$, can be referenced against it. The j^{th} new principal component score of $\boldsymbol{y}_i^*$ can be easily calculated as $x_{ij}^* = \boldsymbol{p}_j^\intercal \boldsymbol{y}_i^*$. In vector form, $\boldsymbol{x}_i^* = \mathbf{P}_Q^\intercal \boldsymbol{y}_i^*$ which allows to determine the fitted value predicted by the model as $\hat{\boldsymbol{y}}_i^* = \mathbf{P}_Q \boldsymbol{x}_i^*$. Hence, the model residuals are $\boldsymbol{e}_i = \boldsymbol{y}_i^* - \hat{\boldsymbol{y}}_i^*$. Statistically, these errors are well approximated by a multivariate normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_e)$ [Nomikos and MacGregor, 1995].

The SPE[2] is the quadratic form of the error associated with the PCA model, i.e.

$$SPE_i = \boldsymbol{e}_i^\intercal \boldsymbol{e}_i, \tag{1.2}$$

and it is an indication of how well each sample conforms to the PCA model. Box [1954] has shown that this quadratic statistic can be approximated by a weighted chi-squared distribution

$$SPE_\alpha \sim g\chi^2_h$$

where the weight (g) and the degrees of freedom (h) are both functions of the eigenvalues of $\boldsymbol{\Sigma}_e$. An approach to determine g and h is by matching the moments between the $g\chi^2_h$ distribution and the reference distribution of the SPE. The mean and variance of the $g\chi^2_h$ distribution $(\mu = gh, \sigma^2 = 2g^2h)$ are equated to the sample mean (m) and variance (v) of the SPE_i, $i = 1, \ldots, N$; this results in $g = v/(2m)$ and $h = 2m^2/v$. Therefore, the control limit for the SPE is given by

$$SPE_\alpha = \frac{v}{2m}\chi^2_{(2m^2/v),\alpha}$$

with $\chi^2_{2m^2/v,\alpha}$ being the percentile of a chi-squared distribution with $2m^2/v$ degrees of freedom at the α significance level. This method of matching moments is susceptible to error when there are outliers in the data or when the number of observations is

[2] also known as Q-statistic [Jackson, 1991].

small; for those cases the approximation provided by Jackson and Mudholkar [1979] has been shown to be more robust.

Hotelling's statistic

The sum of normalized squared scores (Hotelling's T^2-statistic) is a measure of the variation in each sample *within* the PCA model. It is defined as

$$T_i^2 = \sum_{j=1}^{Q} \frac{x_{ij}^2}{\lambda_j}. \tag{1.3}$$

The upper control limit for the Hotelling's T^2-statistic can be obtained using the empirical reference distribution of the training data or through its relationship with the F-distribution [Jackson, 1991, p. 23], as follows

$$T_{Q,N,\alpha}^2 = \frac{Q\,(N-1)}{N-Q} F_{Q,N-Q,\alpha}.$$

1.2.5 Fault diagnosis: contribution plots

In terms of process faults, there are two kind of abnormalities that can develop in a chemical system, Zhang et al. [1997]. Firstly, the relationship between the process variables could change. What it is expected in this situation is that the difference between the original observations y_d and the model prediction $\hat{y}_d$ would be large. These faults can be detected by monitoring the *Squared Prediction Error*. And secondly, the basic relationship between the process variables could remain unchanged but the process variables could present a variability higher than those in the nominal data. This abnormality would be observable if we were to monitor the latent variables directly. These faults can also be detected by using the Hotelling's T^2 Statistic. A very effective set of multivariate control charts uses therefore the T^2 chart in conjunction with the SPE plot [MacGregor and Kourti, 1995].

Any of the previously described charts can be used in the first stage of the monitoring procedure, *fault detection*. The second stage is related to identifying the root cause responsible for the out-of-control signal; this is commonly referred to as *fault*

identification or *fault diagnosis*. In this respect, the most popular approach is to make of use contribution plots[3] [MacGregor et al., 1994; Miller et al., 1998]. The idea behind them is rather simplistic; it focuses on decomposing the signal which is out of the control limits into its individual constituents so that the variable(s) responsible for the unusual behaviour can be identified.

Let x_{ij} be the j^{th} principal component score for an observation $\boldsymbol{y}_i$ which is defined as a linear combination of all the variables in $\boldsymbol{y}_i$

$$x_{ij} = \boldsymbol{p}_j^{\intercal}\boldsymbol{y}_i = \sum_{d=1}^{D} p_{dj}y_{id} = \sum_{d=1}^{D} \text{cont}_x(y_d), \tag{1.4}$$

where $\text{cont}_x(y_d) = p_{dj}y_{id}$ is the individual contribution of variable y_d to the principal component score x. If it turned out that x_{ij} was a signal out of control then the contributions of each variable, $\text{cont}_x(y_d)$, could be plotted and compared with the contributions for the same variable in the reference (nominal) data set. This comparison coupled with engineering knowledge should help in locating the root cause of the problem. Alternatively, control limits for these contributions could also be used [Conlin et al., 2000].

Similarly, let us assume that a fault is detected in the Q-statistic plot. In this case

$$SPE_i = \|\boldsymbol{e}_i\|^2 = \sum_{d=1}^{D}(y_{id} - \hat{y}_{id})^2 = \sum_{d=1}^{D}[\text{cont}_Q(y_d)]^2, \tag{1.5}$$

where $\text{cont}_Q(y_d) = y_{id} - \hat{y}_{id}$ are the individual contributions of variable y_d to the Q-statistic[4]. As before, plots of $\text{cont}_Q(y_d)$ can be used to establish a visual comparison with the contribution of the variable in the reference data set to help determine the source of the out-of-control signal.

Contributions to the T^2 are not clearly defined in the literature. Although several definitions have been proposed [Qin, 2003] the idea remains the same, namely to identify what variable(s) y_d are responsible for the out-of-control signal.

[3] When a historical data base of common faults is available, an alternative to the contribution plots is to use a reconstruction-based approach; for further details refer to Qin [2003].

[4] Note that subscripts i and j have been omitted in the definitions of $\text{cont}_x(y_d)$ and $\text{cont}_Q(y_d)$ for clarity and to emphasize the dependence of the latent variable with the original observations y_d.

1.2.6 Case study

Let us consider the example proposed by Dong and McAvoy [1996] of a moderately non-linear system with three variables, $D = 3$, but only one underlying latent variable, $Q = 1$. The data is simulated by

$$\begin{aligned} y_1 &= x + \varepsilon_1, \\ y_2 &= x^2 - 3x + \varepsilon_2, \\ y_3 &= -x^3 + 3x^2 + \varepsilon_3. \end{aligned} \tag{1.6}$$

where x is generated from a uniform distribution $\mathcal{U}(1.01, 2)$; the independent noise ε_d is generated from a Gaussian distribution $\mathcal{N}(0, 0.01^2)$ for $d = 1, 2, 3$.

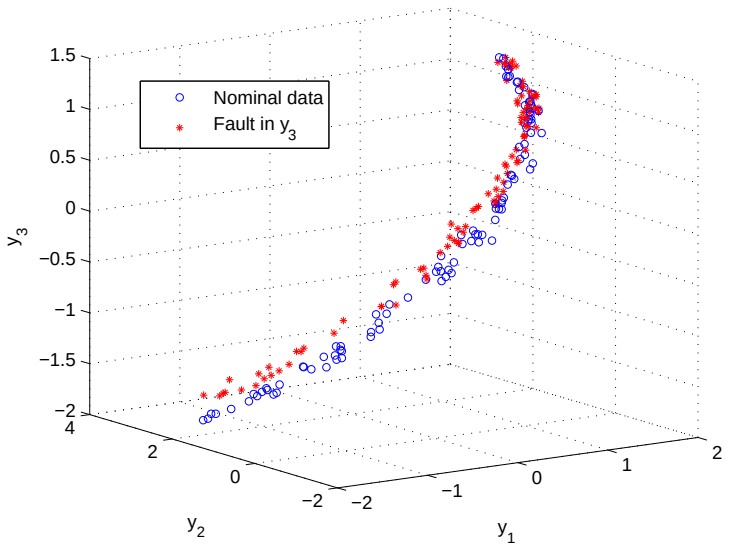

Figure 1.1: Data sets for normal condition (o) and fault condition (*)

The nominal data set (where the process is behaving as intended) is made of 100 observations generated with Equation (1.6). After the first 100 samples, let us assume that a fault has developed in the process affecting only variable y_3. A new data set of 100 faulty data observations are simulated where y_1 and y_2 are obtained as before but with y_3 now given by

$$y_3 = -1.1x^3 + 3.2x^2 + \varepsilon_3. \tag{1.7}$$

The set of faulty data will be used to determine how effective a PCA-based monitoring approach is. All the data has been scaled to zero mean and unit variance as PCA is not scale invariant. As it can be seen in Figure 1.1 there are mild non-linearities. Additionally, the faulty data shows a positive displacement in the y_3 direction but this is not easily identifiable by visual inspection.

One principal component accounts for around 68% of the variance in the data; two principal components account for more than 99% of the total variance, Figure 1.2 (left panel). By splitting the nominal observations into ten blocks and performing cross-validation, a model with two principal components minimizes the root mean square error (RMSE). A practitioner using other more simple methods like the Average Eigenvalue approach would likely select only one principal component. In the latter case, the model would simply fail to account for the non-linearities in the data and would be unable to detect the problem that has developed in the variable y_3, Figure 1.2 (right panel).

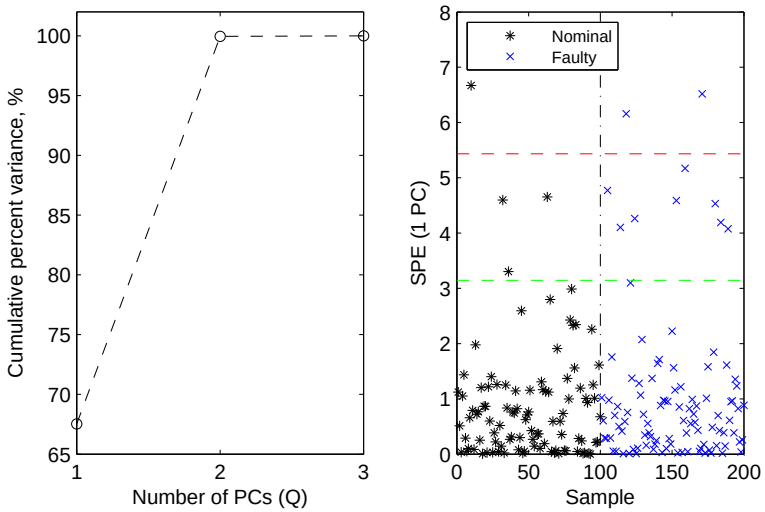

Figure 1.2: Left - cumulative percent variance explained as a function of Q. Right - SPE plot for 1 pricinpal component with 95%,–, and 99%,–, control limits; vertical line at sample 100 separates nominal from faulty observations.

As shown in Figure 1.3 (top panel), the selection of two latent variables renders a model which is successful at detecting the problem in y_3. The next step for the process engineer, knowing that the system has developed a fault, is to find the variable(s) responsible for the consistent out-of-control signals. As mentioned

previously, a way of doing so is to use the variable contributions to the SPE. This is shown in Figure 1.3 (bottom panel), where it can be seen that the contribution of variable y_3 is larger than we would normally expect. Unfortunately, this procedure is not unambiguous and variable y_1 also presents a variability larger than expected. It is now the task of the process engineer, using this information and his knowledge about the system, to carry on with the investigation to be able to discern what variable is, in fact, behaving unexpectedly.

Final remarks

It is hoped that this simple case study both encapsulate the way latent variable models are used in process monitoring and highlight the challenges that these sort

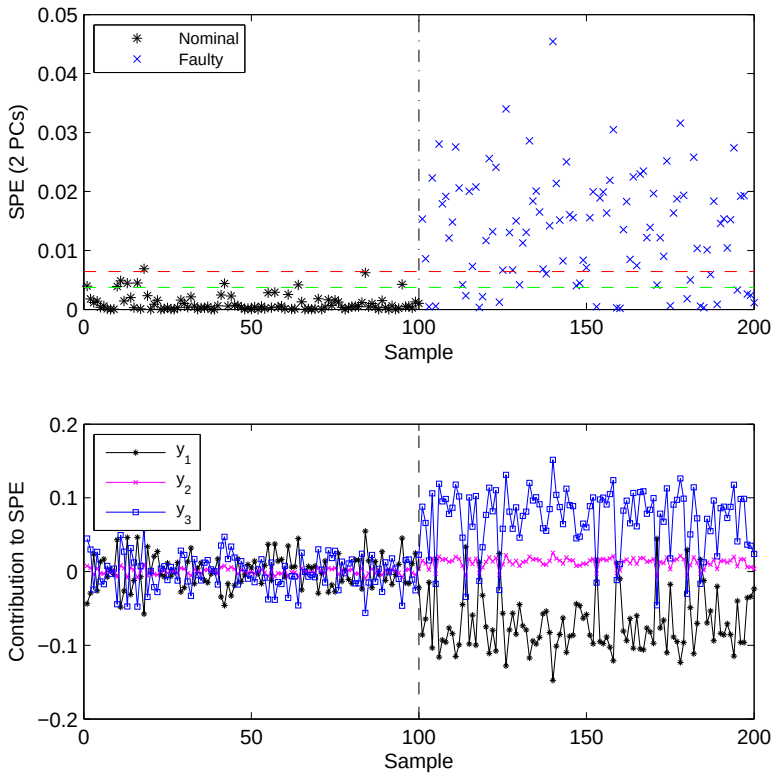

Figure 1.3: Fault identification by using variable contributions to the SPE. Top panel - SPE for a model with two principal components. Bottom panel - variable contributions to the SPE. In both panels, vertical line at sample 100 separates nominal from faulty observations.

of approaches face when they are used in statistical process control. As a way of a summarizing the main ideas which have been and are to be introduced, it is important to take into account the following:

1. The large majority of the data which is generated in the process industries is non-linear; nevertheless, it is rather common for linear latent variable models to be used to model these systems. Leaving aside the argument about the appropriateness of this approach, it is argued that data manipulations such as mean centering or simple transformations as the logarithm contribute to moderate those non-linearities [Kourti, 2002]. Even in those cases, like in the case study just shown, if linear PCA is used to model a non-linear system, it will likely require more latent variables than the underlying dimensionality of the system. That is simply a reflection of a non-parsimonious modelling approach which is unable to reveal the true mechanism generating the data.

2. PCA and PCA-based models build latent variables which are a combination of all of the original variables. The fundamental problem of this approach is that redundant/confounding information is being included as it is rather likely that these latent constructs are not a function of each one of the individual variables that we are choosing to record. Furthermore, it would desirable to have some form of variable selection selection procedure which allowed us to build latent constructs which are only a function of a subset of the original variables.

3. Many different procedures are proposed in the literature to select the required number of principal components. There is no set rule as to which one is the most appropriate and, obviously, different methods will lead to different results. This is rather important if a linear model is to be used to monitor a non-linear process. In many cases the non-linearities will show in components that, a priori, explain very little variance (see, for instance the industrial system studied by Simoglou et al. [2000]); discarding them will simply ignore information that is essential for the correct functioning of the process.

1.3 The Gaussian process regression model

A Gaussian process regression model, GPR, can be used to approximate complex non-linear functions with relative simplicity. Their regression performance is, at least, comparable to that achieved via artificial neural networks (NN) and, in fact, both methods are intrinsically related. They are both non-parametric and, as Neal [1994] has shown, when the number of nodes in the hidden layer of a neural network tends to infinity the NN converges to a Gaussian process.

Whitin the context of regression, Gaussian processes have been widely in use in the field of geostatistics since the 1960's. In this area they are commonly referred to as *kriging*, a term coined by Matheron [1963] in honour of the pioneering work carried out by D. G. Krige, a South African mining engineer; as it would be expected, in spatial statistics the input to the Gaussian process is limited to two or three dimensions.

It is not until the work of O'Hagan [1978] that GPRs were used in statistics to deal more generally with multivariate input regression problems. It can be said, however, that its uptake by the community was fairly slow in subsequent years. It is from the mid-nineties, when Williams and Rasmussen [1996] introduced GPRs in a machine learning context, when there has been a real surge in research activity.

This section intends to provide a short summary about the GPR model. The topic is also discussed in great detail in Rasmussen and Williams [2006] and Shi and Choi [2011].

1.3.1 Gaussian process priors

Let us consider the data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)|_{i=1}^N,\ \boldsymbol{x}_i \in \mathcal{R}^Q, y_i \in \mathcal{R}\}$, i.e. it comprises N pairs of observations each consisting of a Q-*dimensional* input[5] vector $\boldsymbol{x}_i$ and a scalar output y_i. Let also $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^\intercal$ be the $N \times Q$ design matrix with all the input vectors and $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)^\intercal$ the corresponding output vector. The

[5]Note that in the case of Gaussian process regression Q is the dimension of the input variables and it can be high-dimensional.

GPR regression model is defined as follows

$$\begin{aligned} y_i &= f(\boldsymbol{x}_i) + \varepsilon_i, \\ \varepsilon_i &\sim \mathcal{N}(0, \sigma^2) \quad \textit{i.i.d. and} \\ f(\cdot) &\sim \mathcal{GP}(0, k(\cdot, \cdot)), \end{aligned} \tag{1.8}$$

where $\mathcal{GP}(0, k(\cdot, \cdot))$ denotes a Gaussian process prior distribution with zero mean and *covariance function* or *kernel* $k(\cdot, \cdot)$. In other words, we are assuming that y_i is related to $\boldsymbol{x}_i$ non-linearly through an unknown function f, which, in turn, is approximated by a GPR. And by saying that the function f follows a $\mathcal{GP}$ it is meant that, over the finite range of input observations $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$, the vector $\boldsymbol{f} = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_N))^\intercal$ follows a multivariate normal prior distribution. This distribution is commonly specified as having mean zero and an $N \times N$ covariance matrix generated via $k(\cdot, \cdot)$, where the covariance between $f(\boldsymbol{x}_i)$ and $f(\boldsymbol{x}_j)$ is given by $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

1.3.2 Covariance functions

The covariance function (or covariance kernel) allows to write the covariance between the noise-free output, $f(\boldsymbol{x}_i)$, as a function of the input vectors, $\boldsymbol{x}_i$. It is a key part of the GPR as it will govern the properties of the regressed function; it must always generate a positive semi-definite covariance matrix. Throughout this thesis, the *squared exponential* kernel (also known as Radial Basis Function, RBF, or Gaussian kernel) will be used extensively due to its flexibility:

$$\begin{aligned} k_{ij} &= k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) = \operatorname{cov}\left(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)\right) \\ &= b_o + \upsilon_o \exp\left\{-\frac{1}{2}\sum_{q=1}^{Q} w_d \left(\boldsymbol{x}_{iq} - \boldsymbol{x}_{jq}\right)^2\right\}. \end{aligned} \tag{1.9}$$

Let us also define $\mathbf{K}$ as the *covariance* or *kernel matrix* evaluated at all pairs of the N training observations, i.e. $\mathbf{K} = (k_{ij})$.

The squared exponential term in the previous equation captures the idea that vectors close in the input space should give rise to highly correlated outputs. The

term b_o represents a bias controlling the vertical offset of the GPR; v_o controls the vertical scale of the process. Finally, w_d is a weighting on the distance measure for each dimension; hence, if a w_d was to be small, then the i^{th} dimension would be downweighted and would have little effect on the output. Yi et al. [2011] have recently used this idea successfully for variable selection. Note also that these weights are inversely related with the length-scale parameters used to implement *automatic relevance determination* [Neal, 1996] to filter irrelevant inputs out. The effect of all these parameters[6] is best seen by generating sample functions from the prior defined by this kernel, Figure 1.4. Sampling from a GPR is no different from sampling from a multivariate normal distribution as shown in Appendix A.1.

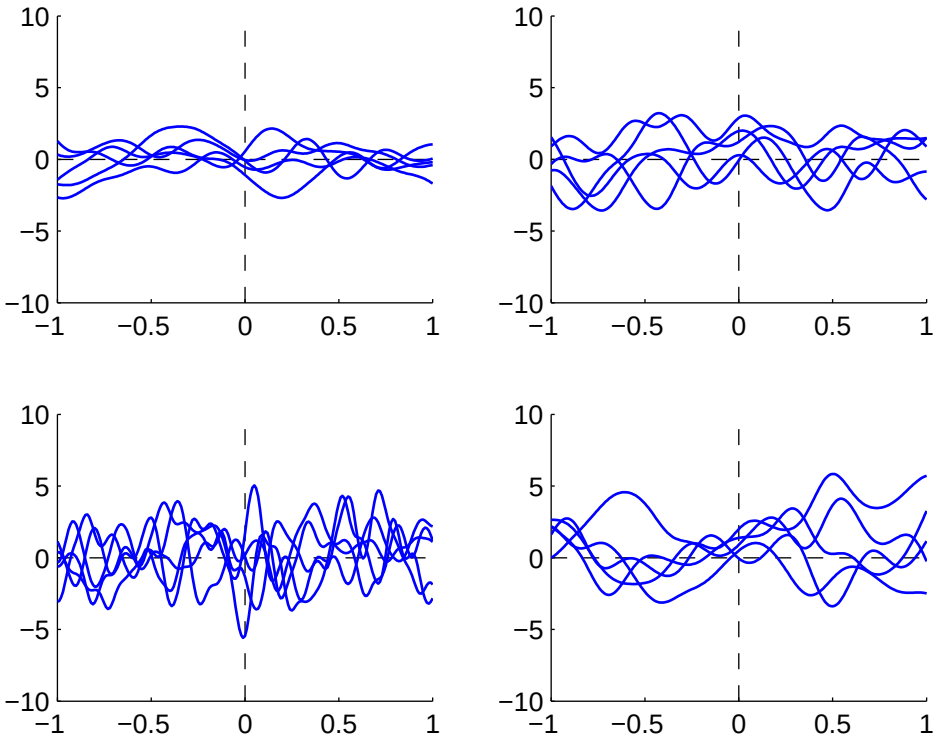

Figure 1.4: Five samples from a $\mathcal{GP}$ with the RBF covariance function and only one input. The GPR parameters, $\log\theta = (\log v_o, \log w_o, \log b_o)$ have the following values: top-left $(0, 4, -3)$, top-right $(1, 4, -3)$, bottom-left $(1, 8, -3)$, bottom-right $(1, 4, 3)$

Furthermore, as it has been mentioned, all the parameters in the model must be positive and therefore it is convenient to reparameterize and consider the parameter

[6] Also known as *hyperparameters* to emphasize that the parameters arise from a prior distribution in Bayesian analysis.

vector in the log-space as explained in Appendix B. That turns the optimisation into an unconstrained problem.

MacKay [1999] provides a comprehensive discussion about what considerations need to be taken into account when choosing a covariance function. Further details about covariance functions properties and how to construct them can also be found in Shawe-Taylor and Cristianini [2004, Chapters 3 and 9].

1.3.3 Posterior distribution

Let the values of the latent function be $f_i = f(\boldsymbol{x}_i)$ and $\boldsymbol{f} = [f_1, \ldots, f_N]^\intercal$. From the model structure defined by Equation (1.8), the conditional distribution of $\boldsymbol{y}|\boldsymbol{f}, \sigma^2$ is multivariate normal

$$\boldsymbol{y}|\boldsymbol{f}, \sigma^2 \sim \mathcal{N}(\boldsymbol{f}, \sigma^2\mathbf{I}).$$

Now, using Bayes' rule, the posterior over the latent function values $\boldsymbol{f}$ is given by

$$p(\boldsymbol{f}|\mathcal{D}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{y}|\boldsymbol{f}, \sigma^2)p(\boldsymbol{f}|\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{f}, \sigma^2)p(\boldsymbol{f}|\boldsymbol{\theta})d\boldsymbol{f}} \propto \varphi(\boldsymbol{y}|\boldsymbol{f}, \sigma^2\mathbf{I})\varphi(\boldsymbol{f}|\mathbf{0}, \mathbf{K}), \tag{1.10}$$

where $\varphi(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the density function of a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. This analytically tractable posterior density is also multivariate normal [Lindley and Smith, 1972] as follows

$$p(\boldsymbol{f}|\mathcal{D}, \boldsymbol{\theta}) \sim \mathcal{N}\left(\mathbf{K}\mathbf{K}_y^{-1}\boldsymbol{y}, \sigma^2\mathbf{K}\mathbf{K}_y^{-1}\right),$$

where $\mathbf{K}_y = \mathbf{K} + \sigma^2\mathbf{I}$. In other words, $\mathbf{K}_y$ is the $N \times N$ covariance matrix whose $(i, j)^{th}$ element is defined as

$$(\mathbf{K}_y)_{ij} = \mathrm{cov}(y_i, y_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sigma^2\delta_{ij}, \tag{1.11}$$

with δ_{ij} being the Kronecker delta. Notice the subtle but important difference between $\mathbf{K}$, the noise-free covariance matrix, and $\mathbf{K}_y$ which incorporates the functional noise along its diagonal.

GPRs provide a straightforward framework to predict the output $f(\boldsymbol{x}^*)$ for a new

input vector $\boldsymbol{x}^*$. The joint distribution of the new enlarged vector of outputs $(y_1, \ldots, y_N, f(\boldsymbol{x}^*))^\intercal$ will still be multivariate normal; the prediction, i.e. $\hat{y}^*$, of $f(\boldsymbol{x}^*)|\mathcal{D}$ is a normal distribution whose mean and variance are given as

$$\begin{aligned} E(f(\boldsymbol{x}^*)|\mathcal{D}) &= \boldsymbol{k}^{*\intercal}\mathbf{K}_y^{-1}\boldsymbol{y}, \\ \text{Var}\,(f(\boldsymbol{x}^*)|\mathcal{D}) &= k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}^{*\intercal}\mathbf{K}_y^{-1}\boldsymbol{k}^*, \end{aligned} \tag{1.12}$$

where $\boldsymbol{k}^* = (k(\boldsymbol{x}^*, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}^*, \boldsymbol{x}_N))^\intercal$ is the vector of covariances between the new input point, $\boldsymbol{x}^*$, and the training data $\boldsymbol{x}_i,\ i = 1, \ldots, N$.

1.3.4 Marginal distribution

The marginal distribution of $\boldsymbol{y}$ can be calculated by integrating out the latent variables from the joint density $p(\boldsymbol{y}, \boldsymbol{f})$, that is

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f}, \sigma^2)p(\boldsymbol{f}|\boldsymbol{\theta})d\boldsymbol{f}. \tag{1.13}$$

This integral is also analytically tractable. Furthermore, as shown in Appendix A.2, it is multivariate normal with the following mean and covariance matrix

$$\boldsymbol{y}|\boldsymbol{\theta} \sim \mathcal{N}_N(\mathbf{0}, \mathbf{K}_y). \tag{1.14}$$

For notational simplicity, in subsequent sections the hyperparameter vector $\boldsymbol{\theta}$ may be loosely overloaded in order to include both the kernel parameters and the functional noise, i.e. $\boldsymbol{\theta} = (w_1, \ldots, w_Q, v_o, b_o, \sigma^2)$.

1.3.5 Empirical Bayes estimation

As Figure 1.4 reveals, what the final regression function looks like is going to be highly dependent on the value of the model hyperparameters $\boldsymbol{\theta}$. A prior distribution could be allocated to each of these hyperparameters and then compute its Bayesian posterior $p(\boldsymbol{\theta}|\mathcal{D})$; this, however, will require a detailed specific knowledge about the system under study which, in most practical circumstances, the modeller will be

lacking. In this case, it is best to use an *empirical Bayes estimate* [Carlin and Louis, 2000, Chapter 3] of the hyperparameters; in other words, the observed data will determine what the most appropriate value should be. Overfitting tends not to be a problem as there are only a small number of unknown parameters governing the final shape of the fitted function.

With the distribution of the training data known as given by Equation (1.14), the log-likelihood function can be easily written as

$$\ell(\boldsymbol{\theta}|\mathcal{D}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{K}_y| - \frac{1}{2}\boldsymbol{y}^\intercal\left(\mathbf{K}_y\right)^{-1}\boldsymbol{y}. \tag{1.15}$$

Training of a Gaussian process involves determining the values of the unknown parameter vector $\boldsymbol{\theta}$ which maximizes the previous cost function[7]. This optimisation is a non-convex optimisation problem and is best carried out using conjugate gradients (CG) minimisers. Full implementation details are given in Appendix B.

As already stated, a full Bayesian approach is also possible but that approach is not pursued in this thesis. Further details are given by Shi and Choi [2011, Chapter 3].

1.4 Contents of this thesis

This chapter has laid the foundations for both the problem and the topic which are to be investigated in the remaining of the thesis; an example using PCA has been given to show how the process monitoring approach is meant to be used.

Keeping on with linear models, Chapter 2 covers the Factor Analysis (FA) model (commonly used in social sciences disciplines) and its applicability in monitoring industrial systems. Our interest in the FA methodology arises from the fact that this model maps subsets of the full covariate space into the observations (also known as *indicators* following FA terminology); by doing that the resulting model gains in interpretability. How such an approach can be used for process monitoring is further highlighted by using simulated data.

[7] This is an example where $\boldsymbol{\theta}$ is overloaded and contains both the kernel hyperparameters and also the functional noise parameter, σ^2.

Chapter 3 commences with the coverage of non-linear models; more specifically, the chapter is concerned with the Gaussian process latent variable model, GPLV, as introduced by Lawrence [2004] and its applicability to fault detection; several examples as to how this methodology can be applied to the monitoring of industrial processes are shown and a new process monitoring approach is also given [Serradilla et al., 2011]. Mirroring the advantages in interpretability which can be attained using a FA approach instead of a PCA model, we introduce in Chapter 4 a new class of models under the name of Gaussian process functional factor analysis model, GPFFA. The idea is to selectively map subsets of the full input space into the observations in a non-linear way taking advantage of the flexibility of Gaussian process priors; full implementation details are given along with worked examples. Moreover, asymptotic properties are discussed at length.

Model selection issues are discussed in Chapter 5; this is an extensive area of research which increases in difficulty due to the latency of the model covariates. We have tackled this by (1) using a Laplace approximation to integrate the latent variables out of the parameters joint density and (2) penalizing the resulting density function in order to carry out simultaneous model selection and parameter estimation. Finally, we conclude in Chapter 6 considering areas of further research.

Chapter 2

Process monitoring using latent factor scores

The aim of this chapter is to introduce the factor analysis (FA) approach to process monitoring in order to tackle the issues of dimensionality reduction and explain variable correlation. FA models are widely in use in the social and behavioural sciences and have been around for a long time [Cudeck and MacCallum, 2007]. There are, however, two main issues that have probably restricted its applicability in other disciplines: namely, the model identification ambiguity and, above all, its limitation to linear systems.

From a monitoring perspective, the procedure is halfway between the *key variables* approach and the PCA-based modelling technique. It has two main advantages arising from the fact that each latent variable or factor is a linear combination of just a subset of the original variables. Therefore fault detection is faster as the confounding effect of redundant variables is eliminated and fault diagnosis becomes easier; in the latter case, if a fault developed in one of the factors, fault diagnosis would become easier as there will be a smaller set of variables which may be responsible for the out-of-control signal.

In the next section a review of the standard linear FA algorithms is made explaining what the differences are between exploratory factor analysis (EFA) and confirmatory

factor analysis (CFA). In Section 2.2, I proceed to discuss those important aspects of how to determine the number of factors and the factor scores. In terms of monitoring statistics, those introduced in Section 1.2.4 can still be used to monitor the new latent variables derived from FA. Finally, two toy models are used to show how this model can be applied in practice.

2.1 Factor analysis models

There is an extensive literature covering the topic of factor analysis. An introduction to the topic can be found in the classic book of Mardia et al. [1979, Chapter 9]. Harman [1976] is wholly devoted to the subject matter; more recent developments and current research topics can be found in Cudeck and MacCallum [2007].

2.1.1 General factor analysis model

Given a set of centered response (or *manifest*) variables, $\{\boldsymbol{y}_i,\ i = 1, \ldots, n\}$, where $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iD})^\intercal$, the basic idea behind factor analysis is relate them to a corresponding set of underlying latent (or *unobserved*) variables, $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iQ})^\intercal$. Ideally $Q \ll D$ and therefore the latent variables will offer a more parsimonious explanation of the dependences between the observations. The latent variables account for the correlation of the response variables or, in other words, given the value of the hidden factors the response variables would be uncorrelated.

In a general form, the linear FA model could also be extended to include non-linear systems [Yalcin and Amemiya, 2001] and expressed as

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\beta}) + \boldsymbol{\varepsilon}, \tag{2.1}$$

where $\boldsymbol{\varepsilon}$ is a $D \times 1$ unobservable vector of errors; $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\beta})$ is a D-variate function of $\boldsymbol{x}$ and the unknown parameter vector, $\boldsymbol{\beta}$, which maps, either linearly or non-linearly, the Q-variate latent vector into the D-variate observation vector. The linear factor analysis model is a special case of Equation (2.1) in which the the function $\boldsymbol{g}(\cdot)$ is

linear, that is

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\beta}) = \boldsymbol{\Lambda}\boldsymbol{x}, \tag{2.2}$$

where $\boldsymbol{\Lambda}$ is a $D \times Q$ parameter matrix of linear mappings normally referred to as *factor loadings*. In what follows of this chapter we will be restricting the attention to these linear models[1].

2.1.2 Linear Factor Analysis model

The (linear) factor analysis model is defined as

$$\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{x}_i + \boldsymbol{\varepsilon}_i, \quad \text{for } i = 1, \ldots, N, \tag{2.3}$$

where $\boldsymbol{\Lambda} \in \mathcal{R}^{D \times Q}$ is a loading or mapping matrix. Normally, it is further assumed that $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_Q)$, i.e. the latent variables are normally distributed, independent and with unit variance; likewise, the error term is also assumed independent and normally distributed as $\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \boldsymbol{\Psi})$ where $\boldsymbol{\Psi} = \text{diag}(\psi_1, \ldots, \psi_D)$ is a D-diagonal matrix. And finally, the latent variables, $\boldsymbol{x}_i$, and the error, $\boldsymbol{\varepsilon}_i$, are assumed to be uncorrelated, that is $\text{cov}(\boldsymbol{x}_i, \boldsymbol{\varepsilon}_i) = \mathbf{0}$.

Note that by constraining the error variance to be a diagonal matrix, the FA model implies that the observed variables $\boldsymbol{y}_i$ are conditionally independent given the latent variables, i.e.

$$\boldsymbol{y}_i | \boldsymbol{\Lambda}, \boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\Lambda}\boldsymbol{x}_i, \boldsymbol{\Psi}).$$

This conditional distribution is meant to show that the correlation between the observations is explained by the *common latent factors* while the error term, ε_{id}, should explain that variability which is unique to a particular observation y_{id}.

For generality, let us denote the model parameters by $\boldsymbol{\theta} = (\boldsymbol{\Lambda}, \boldsymbol{\Psi})$ and drop the subscript from the variables. The marginal distribution of $\boldsymbol{y}$ can now then be calculated

[1] Traditionally in the behavioural sciences, when references are made to factor analysis, the relationship between latent and response variables is assumed to be linear.

by integrating out the latent variables from the joint density

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{\theta}) d\boldsymbol{x}, \tag{2.4}$$

which can be readily worked out as shown in Appendix A.2; hence, the marginal distribution of $\boldsymbol{y}$ is

$$\boldsymbol{y}|\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\intercal + \boldsymbol{\Psi}). \tag{2.5}$$

Within a more general framework, Equation (2.3) can also be thought of as the *measurement submodel* of a bigger class of models known as structural equation models [Bollen, 1989].

Finally, it is worth noting that principal component analysis (PCA) can be thought of as a special case of the FA model defined in Equation (2.3) by further assuming that the noise is isotropic; or, in other words, assuming that each element of $\boldsymbol{\varepsilon}$ has equal variance, i.e. $\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \psi \mathbf{I}_D)$. This induces the following conditional distribution

$$\boldsymbol{y}_i|\boldsymbol{\Lambda}, \boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\Lambda}\boldsymbol{x}_i, \psi \mathbf{I}_D),$$

from which the marginal distribution of $\boldsymbol{y}_i$ follows by integrating out the latent variables

$$\boldsymbol{y}_i|\boldsymbol{\Lambda} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\intercal + \psi \mathbf{I}) \quad \textit{iid for } i = 1, \ldots, N.$$

Tipping and Bishop [1999] named this model *probabilistic PCA*.

2.1.3 Exploratory Factor Analysis (EFA)

It is common to refer to the FA model of Equation (2.3) as EFA when no further assumptions about the structure of $\boldsymbol{\Lambda}$ are made. Let us explicitly write down the relationship between y_j, where $\boldsymbol{y} = (y_1, \ldots, y_j, \ldots, y_D)^\intercal$, and the latent variables

$$y_j = \lambda_{j1} x_1 + \lambda_{j2} x_2 + \ldots + \lambda_{jk} x_k + \varepsilon_j, \tag{2.6}$$

which, when written for $j = 1, \ldots, D$, clearly shows a link between every latent variable and every one of the variables in $\boldsymbol{y}$; what this implies is that an EFA model without further modification will not offer much advantage in terms of process

monitoring over and above what is achieved by constructing a PCA model.

The FA model assumptions need not be as stated previously either. However, from a monitoring perspective, they are the most appropriate as allow for each of the factors to be monitored independently; this is similar to monitoring the principal components which are always built as independent variables. From the marginal distribution of $\boldsymbol{y}$, Equation (2.4), the population covariance matrix is

$$\text{cov}(\boldsymbol{y}) = \boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^{\intercal} + \boldsymbol{\Psi}. \tag{2.7}$$

Hence, if the model holds, $\boldsymbol{\Sigma}$ can be written as a function of the model parameters, that is $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\boldsymbol{\Lambda}, \boldsymbol{\Psi})$. Once the model has been formulated, the main objective in factor analysis is to determine the number of factors Q and the elements of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ given a sample estimate $\mathbf{S}$ of $\boldsymbol{\Sigma}$.

The model as defined by Equation (2.3) is not identified. The latent factors can be transformed via a non-singular orthogonal matrix $\mathbf{Q}$ such that $\boldsymbol{z} = \mathbf{Q}\boldsymbol{x}$. Then $\boldsymbol{z}$ will still be standard normal and Equations (2.3) and (2.7) would then become

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{\Lambda}\mathbf{Q}^{\intercal}\boldsymbol{z} + \boldsymbol{e} \\ \boldsymbol{\Sigma} &= \boldsymbol{\Lambda}\mathbf{Q}^{\intercal}\mathbf{Q}\boldsymbol{\Lambda}^{\intercal} + \boldsymbol{\Psi} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^{\intercal} + \boldsymbol{\Psi} \end{aligned} \tag{2.8}$$

which shows that the latent variables $\boldsymbol{x}$ estimated via the linear factor model are indeterminate up to an orthogonal rotation. From a practical point of view, this indeterminacy in the definition of the factor loadings is resolved by imposing additional constraints to the rotation of the factors loadings [Krzanowski and Marriott, 1995, p. 132]. The rotations need not be restricted to be orthogonal; they can also be *oblique rotations*. The latter, however, will lead to factors which are no longer independent which is not desirable from a process monitoring perspective.

Let us now assume that we have found an initial $D \times Q$ loading matrix $\boldsymbol{\Lambda}$ by solving Equation (2.3) with the necessary constraints. The rotation problem involves finding the $Q \times Q$ matrix $\mathbf{T}$, which produces the following rotated factor matrix

$$\boldsymbol{\Delta} = (\delta_{ij}) = \boldsymbol{\Lambda}\mathbf{T} \tag{2.9}$$

by minimizing a continuous function $f(\boldsymbol{\Delta})$ of the factor loadings. The *orthogonal rotation* will satisfy the constraints

$$\boldsymbol{\Phi} = \mathbf{T}\mathbf{T}^{\intercal} = \mathbf{I}, \tag{2.10}$$

where $\boldsymbol{\Phi}$ is the covariance matrix of the latent variables; hence, this rotation produces latent variables which are uncorrelated and have unit variances. Note how this rotation imposes imposes $\frac{1}{2}Q(Q-1)$ constraints. There are a myriad of rotation criteria in the literature; Browne [2001] provides an excellent review of these. However one of the most used criteria is the *varimax* rotation which belongs to a more general class of methods known as the Crawford-Ferguson [Crawford and Ferguson, 1970] family of rotation criteria.

The emphasis of the varimax method [Kaiser, 1958] is on simplifying the columns of the factor loadings matrix $\boldsymbol{\Lambda}$. The rationale behind the method is to find columns with a few large loadings and as many near-zero loadings as possible. In that sense, Kaiser states that the greatest interpretability will be achieved when the simplicity of a factor j, s_j, is defined as the variance of its squared loadings

$$s_j = \frac{1}{D}\sum_{i=1}^{D}(\delta_{ij}^2)^2 - \frac{1}{D^2}\left(\sum_{i=1}^{D}\delta_{ij}^2\right)^2 \quad \text{for } j = 1, \ldots, Q. \tag{2.11}$$

For the complete factor matrix $\boldsymbol{\Delta}$, the varimax criterion is given as the sum of the simplicities for each individual factor, i.e.

$$f(\boldsymbol{\Delta}) = \sum_{j=1}^{Q}\left[\frac{1}{D}\sum_{i=1}^{D}(\delta_{ij}/h_i)^4 - \frac{1}{D^2}\left(\sum_{i=1}^{D}(\delta_{ij}/h_i)^2\right)^2\right] \tag{2.12}$$

where $h_i = \sum_{j=1}^{Q}\delta_{ij}^2$ is used to normalise each row of the loading matrix. This function weights each variable equally and is normally referred to as the *varimax* criterion.

Both, EFA and a rotation factor transformation like the *varimax* can be used together in a process monitoring setting of a linear system; an example is shown in subsequent sections.

2.1.4 Confirmatory Factor Analysis (CFA)

When no prior knowledge about the model underlying a data set is available we are proposing to use exploratory factor analysis. EFA will answer the question of how many factors are needed to account for the correlation in the observations; once the number of factors has been determined, factor interpretation is achieved by rotating the initial solution. A graphical representation of a the process is shown in Figure 2.1 (these diagrams are normally referred to as *path diagrams* following *structural equation model* terminology):

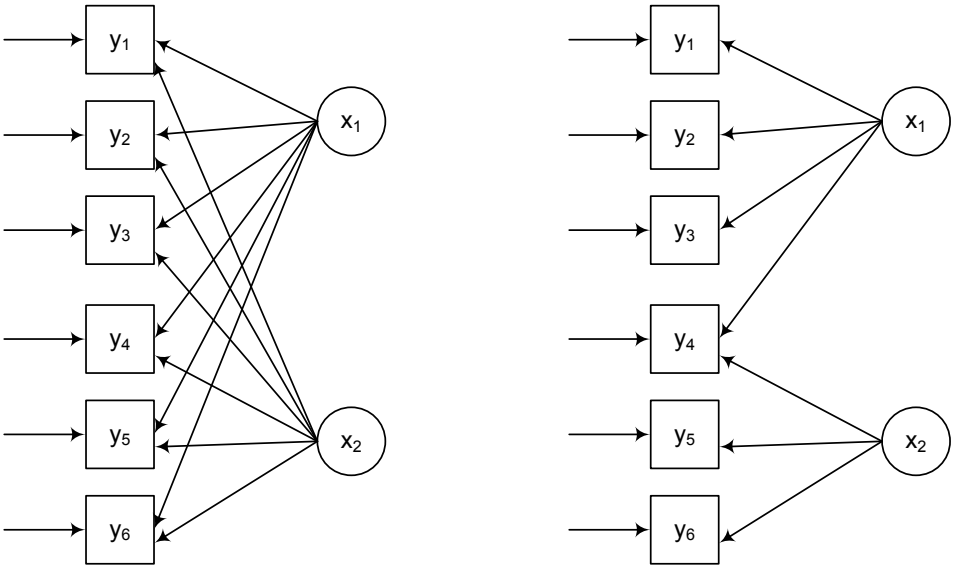

Figure 2.1: Left: *EFA solution (also PCA representation).* Right: *EFA rotated solution.(CFA hypothesised model).*

The plot shows 6 observed variables $y_1, \ldots, y_6$ (enclosed in squares) and two latent variables x_1, x_2 (enclosed in circles); the horizontal arrows pointing to the squares represent variable error terms, $\varepsilon_1, \ldots, \varepsilon_6$. Likewise, the arrows pointing to the variables from the factors are intending to show how each variable loads on each factor. The factor correlation is $\mathbf{\Phi} = \mathbf{I}$, which is represented by the absence of a link between both latent factors. On the left panel, the initial EFA solution is shown; note how every latent construct is related to all of the original variables (this could also be a scaled principal component analysis solution). Once the factors are rotated (right panel) a much simpler structure can be found; note how some of the arrows linking variables with factors do no longer exist. Initially $\mathbf{\Lambda}$ is a full 6×2 matrix with 12 parameters that need determining. Upon rotation, some of the loadings will

no longer be significant which will simplify the ulterior analysis. Mathematically, the gain arises when the full initial loading matrix, $\mathbf{\Lambda}_{\text{initial}}$, is transformed into a sparse matrix $\mathbf{\Lambda}_{\text{rotated}}$:

$$\mathbf{\Lambda}_{\text{initial}} = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \\ \lambda_{41} & \lambda_{42} \\ \lambda_{51} & \lambda_{52} \\ \lambda_{61} & \lambda_{62} \end{pmatrix} \quad \Longrightarrow \quad \mathbf{\Lambda}_{\text{rotated}} = \begin{pmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ \lambda_{41} & \lambda_{42} \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \end{pmatrix}$$

On the other hand, when considerable knowledge about the system is available a confirmatory factor analysis, CFA, would be more appropriate. In that case, a number of factors is hypothesised. Each factor would then be linked to a subset of the original variables with an association only established if there is a significant correlation between the variables. With reference to Figure 2.1 (right panel), variables y_1, y_2, y_3 load on the first factor, x_1, while variables y_5, y_6 load on x_2. According to the model, variable y_4 is an indicator for both x_1 and x_2. As before, the two factors are assumed to be independent. It is the theoretical knowledge about the system what allows us to remove some model parameters by fixing them to zero. Also it is important to realise firstly that no factor rotation is possible, as the only rotation matrix that would retain the zeros in $\mathbf{\Lambda}$ is the identity matrix. And secondly, that rotated factor analysis solution is not necessarily the same as a CFA solution (as implied in Figure 2.1). Confirmatory factor analysis models are treated extensively by Bollen [1989, Chapter 7].

2.2 Model considerations

Before using this new methodology for process monitoring problems there are a few issues remaining, namely (1) how to estimate the model parameters using maximum likelihood; (2) answer the question as to how much variability each factor accounts for; (3) selecting the appropriate number of latent variables, Q; (4) once the model parameters have been estimated, a procedure is needed to determine the factor

scores; (5) how to calculate the standard errors of the model parameters and (6) what statistics can be used for process monitoring of the resulting factor scores. All these topics are treated in the following subsections.

2.2.1 Maximum likelihood estimation

There are several methods available in the literature that can be used to estimate the model parameters in Equation (2.3); see for example [Rencher, 2002, Chapter 5]. However, the usual approach is to proceed by using maximum likelihood. Assuming that $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ are a random sample of size N such that $\boldsymbol{y}_i \sim \mathcal{N}_D(\boldsymbol{\mu}, \boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\intercal + \boldsymbol{\Psi})$ then the likelihood is given by

$$L(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) = (2\pi)^{-\frac{ND}{2}} \, |\boldsymbol{\Sigma}|^{-\frac{N}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{y}_i - \boldsymbol{\mu})^\intercal \, \boldsymbol{\Sigma}^{-1} \, (\boldsymbol{y}_i - \boldsymbol{\mu}) \right)$$

Now, by replacing $\boldsymbol{\mu}$ by its maximum likelihood estimate $\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{y}}$, taking natural logarithms and further standard manipulation, leads to following fitting function

$$\ell(\boldsymbol{\Lambda}, \boldsymbol{\Psi}) = \text{constant} - \frac{N}{2} \log|\boldsymbol{\Sigma}| - \frac{N}{2} \operatorname{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}^*)$$

where $\mathbf{S}^* = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{y}_i - \bar{\boldsymbol{y}})^\intercal (\boldsymbol{y}_i - \bar{\boldsymbol{y}})$ is the sample-biased maximum likelihood estimator of the covariance matrix and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ the model-implied covariance matrix.

Normally, for computational purposes, a slight modification of the previous objective function is optimised

$$F_{ML}(\boldsymbol{\Lambda}, \boldsymbol{\Psi}) = \log|\boldsymbol{\Sigma}| + \operatorname{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}) - \log|\mathbf{S}| - D, \tag{2.13}$$

which aims to find the maximum likelihood estimates $\widehat{\boldsymbol{\Sigma}}$ by minimising the discrepancy between the model-implied covariance matrix and the sample covariance; see, for example, Bollen [1989, Chapter 4]. When the two covariance matrices are equal, both $\log|\widehat{\boldsymbol{\Sigma}}|$ and $\log|\mathbf{S}|$ will be the same whereas $\operatorname{tr}(\widehat{\boldsymbol{\Sigma}}^{-1}\mathbf{S})$ will equal D; hence, the discrepancy function F_{ML} becomes zero. Note also that $\mathbf{S}^* = \frac{(N-1)}{N}\mathbf{S}$ and $\mathbf{S}$ will essentially be the same for large samples.

The maximum likelihood fit function belongs to a general family of fit functions known as *weighted least squares family*. Other members of this family that could be used to fit EFA or CFA models are the *unweighted least squares* and the *generalized least squares* fit functions. For these and yet more additional procedures refer to Jöreskog [2007]. In practice, maximum likelihood is the preferred and most used method. Even though in many occasions the assumption of multivariate normality tends not to hold, the maximum likelihood parameter estimates have been found to be very robust to departures from normality [Boomsma and Hoogland, 2001].

2.2.2 Percentage of total variance explained

The more latent factors included in the model the more will the variance of the original observations be accounted for. This is similar to the idea in principal component analysis where normally a number of principal components are selected such that a given percentage of the original variability is accounted for.

The variance of each response variable y_d is partitioned by the model in equation (2.3) into a part due to the common factors (also known as *communality* or h^2) and a part due uniquely to the variable (error)

$$\text{Var}(y_i) = \sum_{j=1}^{Q} \lambda_{ij}^2 + \psi_i = h_i^2 + \psi_i. \tag{2.14}$$

Therefore, the j^{th} factor contributes λ_{ij}^2 to the total variance of y_i. The total contribution of the j^{th} factor to the sample variance given by $\text{tr}(\mathbf{S})$ is

$$\frac{\sum_{i=1}^{D} \lambda_{ij}^2}{\text{tr}(\mathbf{S})} \tag{2.15}$$

and this result can be used to compare a linear FA model with a PCA model. In general, given a number of principal components (or factors), the total variance accounted for the principal components will be bigger than the variance accounted for the same number of factors; this is to be expected and it is due to some of the factor loadings being zero.

2.2.3 Selecting the number of factors

There are several criteria available to select the number of factors, most of them similar to those used for choosing the number of principal components, [Rencher, 2002, Section 13.4].

(i) Choose Q equal to the number of factors necessary for the variance accounted for to achieve a predetermined percentage, say 80%, of the total variance $\text{tr}(\mathbf{S})$ or $\text{tr}(\mathbf{R})$.

(ii) Choose Q equal to the number of eigenvalues greater than the average eigenvalue.

(iii) Use a scree plot test of the eigenvalues of $\mathbf{S}$ or $\mathbf{R}$.

(iv) Test the hypothesis that m is the correct number of factors, $H_0 : \boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\intercal + \boldsymbol{\Psi}$, where $\boldsymbol{\Lambda}$ is $p \times m$.

Methods (i)-(iii) have their counterpart in PCA. Method (iv) arises when the multinormal distributional assumptions are made about the data.

2.2.4 Factor scores

Once the factor loading matrix have been determined, the objective is to estimate the unobserved factor scores, $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iQ})^\intercal$, $i = 1, \ldots, N$. This is important as monitoring will be based on these estimates or other statistics derived from them. There are several methods available, namely *Thompson's regression* method, *Bartlett's weighted least squares* method and *Anderson-Rubin's* method [Harman, 1976]. Thompson's method is the most popular approach [Rencher, 2002, p. 439]. For completeness, when Thompson's method the factor scores can be found with the following equations

$$\widehat{\mathbf{X}} = \mathbf{Y}_c \mathbf{S}^{-1} \widehat{\boldsymbol{\Lambda}}_Q \tag{2.16}$$

where $\mathbf{Y}_c$ is the $N \times D$ matrix of centred observations. If $\mathbf{R}$ is used instead of $\mathbf{S}$, then

$$\widehat{\mathbf{X}} = \mathbf{Y}_s \mathbf{R}^{-1} \widehat{\mathbf{\Lambda}}_Q \tag{2.17}$$

where $\mathbf{Y}_s$ is the $N \times D$ matrix of standardized observations.

2.2.5 Standard errors of the EFA maximum likelihood estimates

Formulae for the asymptotic standard errors of unrotated EFA loading estimates were originally and systematically developed by Lawley and Maxwell [1971]; these formulae have a slight error subsequently corrected by Jennrich and Thayer [1973]. The initial loading estimates, $\widehat{\mathbf{\Lambda}}$, of the EFA solution are commonly referred to as *unrotated loadings* although they are in reality the solution of maximum-likelihood problem where the loadings are orthogonally rotated[2] to satisfy the $\frac{1}{2}Q(Q-1)$ constraints

$$\mathbf{\Lambda}^\intercal \mathbf{\Psi}^{-1} \mathbf{\Lambda} \text{ is diagonal.} \tag{2.18}$$

These constraints are necessary so that an unique EFA solution can be found as explained in Section (2.1.3).

The standard errors for the MLE's of the factor loadings can also be expressed in terms of the inverse of an augmented information matrix which takes into account the constraints used in the factor rotation [Jennrich, 1974]. Formulae developed this way, although rotation-specific, are easier to handle.

However, it has been known for a long time that solutions where some of the MLE's of the unique variances, $\widehat{\mathbf{\Psi}}$, are zero (*Heywood* case) or near zero [Jöreskog, 1967] are very common. These improper solutions pose a problem for both procedures above as Jennrich and Lawley's formulae involve the reciprocal of the unique variances. That may cause the formulae for the unrotated loadings to break down. Based on a augmented information matrix approach, Hayashi and Bentler [2000] have proposed a modification of Lawley's and Jennrich methods which is based on the following

[2]This rotation is also known as canonical rotation.

alternative rotation constraints

$$\boldsymbol{\Lambda}^{\intercal}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda} \text{ is diagonal,} \tag{2.19}$$

where $\text{cov}(\boldsymbol{y}) = \boldsymbol{\Sigma}$. Both set of constraints, Eq. (2.18) and (2.19), are equivalent but Hayashi's formulation will have the added advantage that can be used regardless of whether or not any element of the unique variances is nearly zero.

2.2.6 Monitoring statistics and fault diagnosis

The monitoring charts proposed in Section 1.2.4 can also be used when the latent variables are estimated using a FA model instead of PCA. To be more specific, monitoring can be done using:

- individual factor scores, x_q for $q = 1, \ldots, Q$, with control limits worked out either theoretically, using the normal distribution, or empirically.
- the squared prediction error.
- Hotelling's statistic.

For the latter two cases, the control limits are the same as those given in Section 1.2.4. The contribution plots introduced in Section 1.2.5 are also applicable for fault identification when a FA model is fitted. Note that there will be a substantial gain when a linear system is modelled using a FA approach, namely that the latent variables will be related only to specific subsets of the observations y_d.

2.3 Numerical examples

In order to show how a linear factor analysis approach can be applied to process monitoring two examples with simulated data are shown. The first example simulates a process where a sudden mean change in one of the process variables develops. In the second one, the same fault arises but develops gradually with time.

2.3.1 Simulation study

Let us assume that we have a 6-variate observation vector $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{i6})$ which is generated from 2 underlying factors $\boldsymbol{x}_i = (x_{i1}, x_{i2})$, according to the linear factor model $\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{x}_i + \boldsymbol{\varepsilon}_i,\ i = 1, \ldots, 100$. The sparse loading matrix mapping the 2-variate factor vector into the 6-variate observation vector is

$$\boldsymbol{\Lambda}^{\mathsf{T}} = \begin{pmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3 & 1 \end{pmatrix}. \tag{2.20}$$

The model assumptions are $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi} = \text{diag}(v_1, \ldots, v_6)$ with $v_i \sim U[0, 0.5)$. The path diagram underlying this system is shown in Figure 2.2. If our knowledge about the system under study is good enough we should be able to hypothesise relationships of this kind.

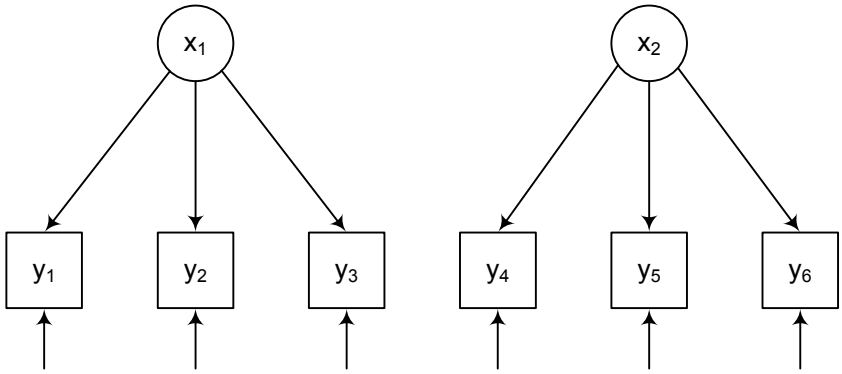

Figure 2.2: Path diagram for the simulated example

Sudden mean change

Normal data are generated following the previous model. Faulty data ($N = 150$) are produced with the same model with the exception of y_5; this variable is generated as $y_5 = (0.3x_2 + \varepsilon_5) + h$, where $h \sim \mathcal{N}(7.5, 1)$ models the process disturbance.

CFA solution

A CFA model can be fitted by minimising Equation (2.13), making provisions for the elements of $\mathbf{\Psi}$ to remain positive; good initial values for the model parameters can be chosen by following the procedure of McDonald and Hartmann [1992]. Alternatively, the *sem package* [Fox, 2006] in R [R Development Core Team, 2009] can also be used.

The variable relationships needed to specify the model are as defined in Figure 2.2. Additionally, for comparison, a 2-principal component model has also been fitted; all the results are shown in Table 2.1; both sets of loadings are based in the sample correlation matrix.

	PC scores		CFA scores	
variable	PC1	PC2	F1	F2
y_1	-0.487	0.295	0.895	0.000
y_2	-0.502	0.306	0.990	0.000
y_3	-0.482	0.311	0.911	0.000
y_4	0.286	0.497	0.000	0.854
y_5	0.307	0.513	0.000	0.987
y_6	0.320	0.462	0.000	0.822
% var	0.474	0.411	0.435	0.397
cum var	0.474	**0.885**	0.435	**0.832**

Table 2.1: PCA and CFA fitted parameters.

There are several important points to take into account in light of the results in Table 2.1:

(a) Principal component analysis is the most efficient way to compress the information of a high-dimensional space [McCabe, 1984]. And this is always the case. As it can be seen in Table 2.1, 2-principal components account for 89% of the information in the original system. However, in this example, it is clear that PCA has been 'too efficient' at doing this and has not only been able to account for the variability in the underlying variables but has also modelled part of the noise built into the system (as the true underlying model is a 2-linear factor model which cannot account for as much variability).

(b) It is worth noting that although the PCA model captures the system variability it does so by sacrificing interpretability. The first principal component contrasts the first 3 variables with the last 3 variables. And the second principal component is simply an average of all the original observations. In other words, all principal component loadings seem to be significant which does not clearly represent the sparse system in Figure 2.2.

(c) Two linear factors account for 83% of the total variation in the system. The original loadings in Equation (2.20) can be recovered by using the fact that the factor analysis model is scale invariant.

(d) As N increases, the CFA scores tend towards the true values in Equation (2.20) with decreasing variance (i.e. as all maximum likelihood estimators, they are asymptotically unbiased and consistent). Regardless of the sample size, the principal components will describe the major direction of variability within the sample. However, as N increases the sample correlation (covariance) matrix becomes more representative of the population correlation matrix. As such, and given the simulated data, the PC loading estimates will gain in interpretability for very large sample sizes (for this particular example N is in the order of 10^4).

Advantages both in terms of fault detection and diagnosis are also to be expected. Any of the monitoring statistics discussed in the introductory chapter, Section 1.2.4, could be used. For the sake of clarity, I am focusing on monitoring the principal component and factor scores. The results are displayed in Figure 2.3.

(a) On the left panel, the scores for the first principal component are plotted. The first 100 samples, in black, correspond to the nominal data. The remaining 150 observations in red, from sample 101 onwards, correspond to faulty data. The best principal component score plot, in terms of detecting samples out of the confidence limits, is principal component 2 due to the fact that the factor loading on x_5 is higher on this component than on principal component 1; however, only 8% of the samples are out of the 99% confidence limit.

(b) On the right panels, factor scores are monitored. Note that the sudden mean change was introduced in x_5. In relation to Figure 2.2, x_5 is only related to the second factor; therefore, we expect that the fault manifest itself in that second

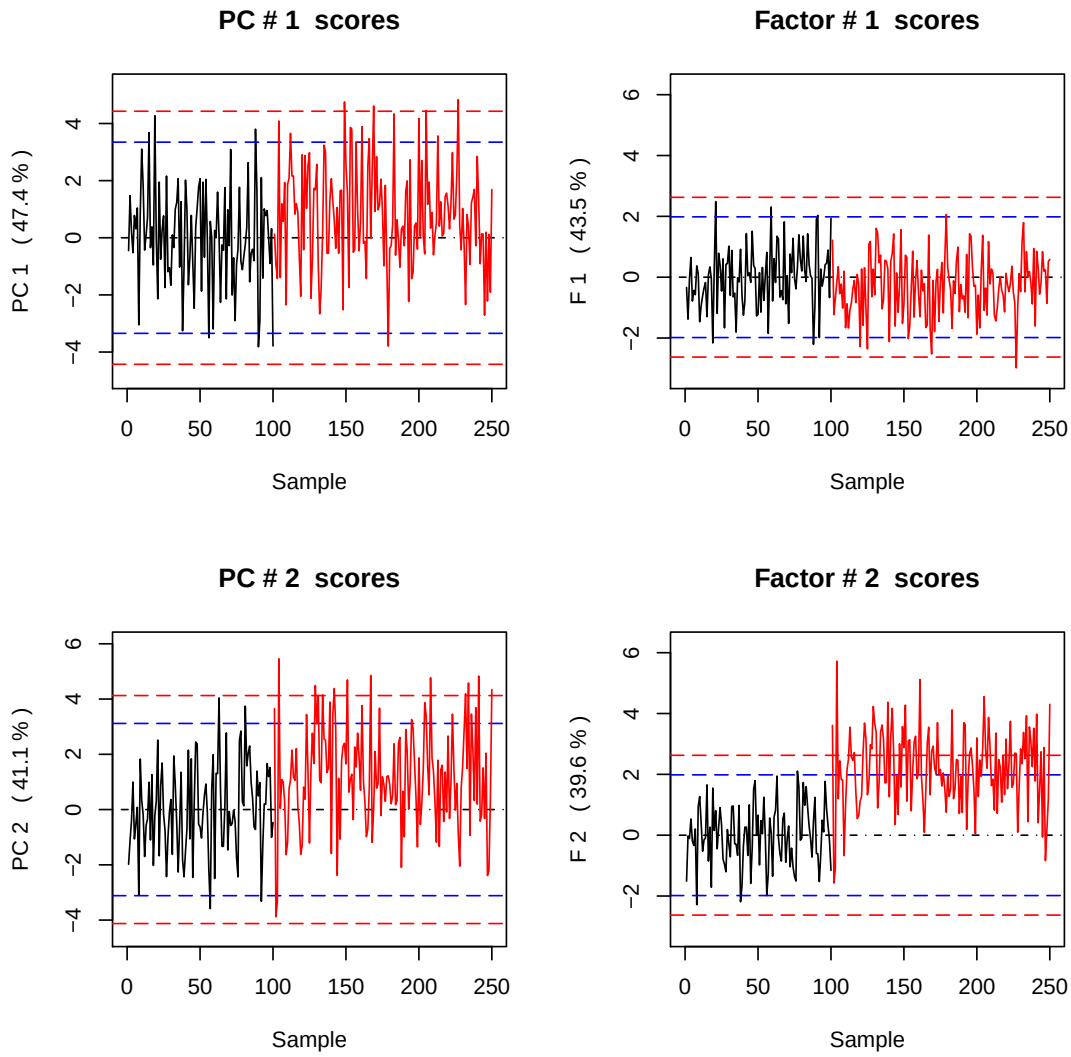

Figure 2.3: Latent factor scores monitoring. Left panel: principal component scores. Right panel: factor scores. Dashed blue line is the 95% confidence limit. Dashed red line is the 99% confidence limit.

factor score. In fact, more than 39% of the samples are out of the 99% confidence limit in the second factor score. The mean change is also very conspicuous from sample 101 onwards.

(c) Fault diagnosis could now be carried out by using contribution plots [Miller et al., 1998] on the second factor score as explained in Section 1.2.5. Note that as that factor is only related to variables $y_4, y_5,$ and y_6 finding the root cause of the problem becomes much simpler.

EFA solution

It is also interesting to compare the results obtained in Table 2.1 by using CFA with the results that would be obtained when our knowledge about the system under study is limited or non-existent. In that case, the approach is first to extract the maximum likelihood factors; subsequently, they can be rotated by using the varimax criterion, Equation (2.12). Having determined the rotated factor loadings, one can calculate asymptotic standard errors [Archer and Jennrich, 1973] and test the hypothesis $H_0 : \lambda_{ij} = 0$ for each coefficient (i refers to variable and j to factor). As, asymptotically, $\hat{\lambda}_{ij} \sim \mathcal{N}\left(\lambda_{ij}, \mathrm{SE}^2(\hat{\lambda}_{ij})\right)$, the null hypothesis H_0 can be evaluated by using the test statistic

$$Z = \frac{\hat{\lambda}_{ij}}{\mathrm{SE}(\hat{\lambda}_{ij})} ,$$

where SE is the asymptotic standard error of the parameter. Then, for a significance level α, one rejects H_0 when $|Z| \geq Z_\alpha$; in this case Z_α is the $100(1 - \frac{1}{2}\alpha)\%$ quantile of the standard normal distribution. Confidence intervals for $\hat{\lambda}_{ij}$ can be constructed in a similar fashion.

Table 2.2: EFA-varimax fitted parameters.

	EFA extraction		EFA-varimax					
Variable	F1	F2	F1	SE	Z-stat.	F2	SE	Z-stat.
x_1	0.889	0.100	0.893*	0.023	38.82	-0.061	0.059	-1.04
x_2	0.983	0.118	0.988*	0.012	82.35	-0.060	0.051	-1.18
x_3	0.904	0.121	0.911*	0.020	45.33	-0.044	0.058	-0.76
x_4	-0.147	0.842	0.007	0.065	0.11	0.854*	0.033	25.89
x_5	-0.166	0.972	0.012	0.055	0.22	0.986*	0.022	44.84
x_6	-0.196	0.800	0.048	0.067	-0.72	0.823*	0.037	22.23

Note. Estimates significant at $\alpha = 0.05, (Z_\alpha = 1.96)$ are marked with an asterisk.

Results of this EFA-varimax solution are shown in Table 2.2. There are two points worth noting regarding them. Firstly, the rotation algorithm has worked by increasing those loadings that were originally closer to one and decreasing the ones closer to zero. And, secondly, the EFA-varimax solution is able to recover a similar solution as the CFA approach with the added advantage that no knowledge about the system was used to model the data.

Gradual mean change

Normal data, as before, is generated following the model in Section 2.3.1. 150 observations of faulty data are also generated with the previous model where, now, the fault in y_5 is introduced as a gradual mean change; it is assumed that at each time unit t, the mean of the variable, $\bar{y}_5$, changes by $t/20$ units. Therefore every 20 sample points, $\bar{x}_5$ will have change by *1 unit*. Mathematically:

$$y_5 = (0.3x_2 + e_5) + h$$

where $h \sim \mathcal{N}(t/20, 1)$. Hence, at $t = 0$, $\bar{y}_5 = 0$ and at $t = 150$, $\bar{y}_5 = 7.5$ as in the previous simulation.

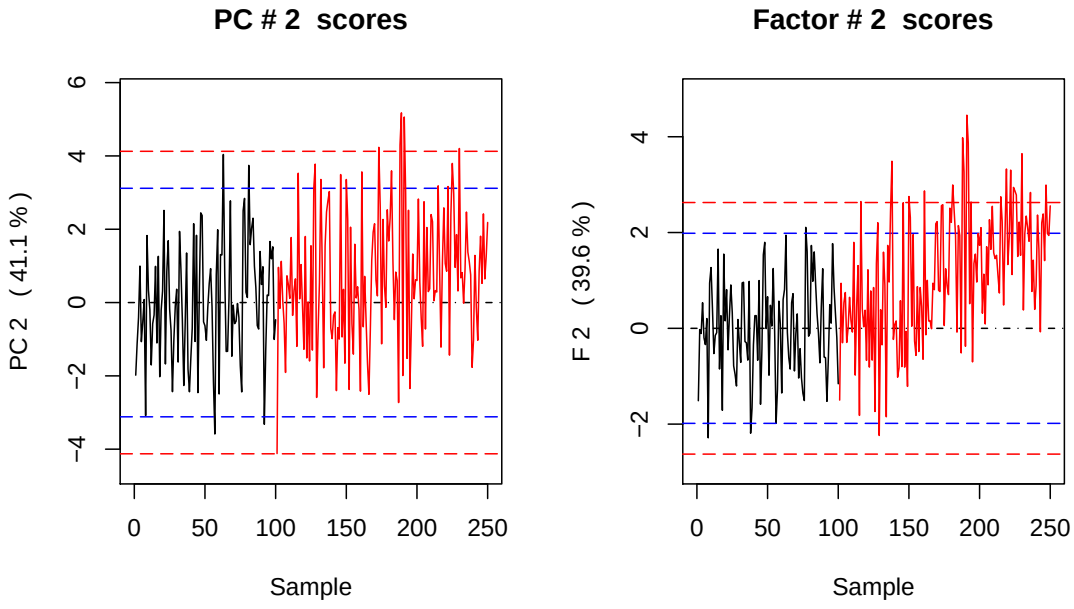

Figure 2.4: Latent variable monitoring. Left panel: principal component 1 score. Right panel: factor 1 score. Dashed blue line is the 95% confidence limit. Dashed red line is the 99% confidence limit.

The results of monitoring the second principal and factor scores are shown in Figure 2.4. Although the procedure is not as effective as before due to the slow change in the mean, it still outperforms a PCA approach. Only 3% of the faulty data points are out of the 99% confidence interval for principal component 2, whereas 13% are out of control in the case of the factor analysis model. Additionally, the

gradual mean change in the monitoring graph can be readily spotted in the case of the second factor score.

2.4 Chapter summary

Factor analysis is a well known statistical technique widely used in the social sciences. The technique has been introduced in this chapter as an alternative to PCA when modelling linear systems. This could be done in two different ways:

- When there is enough knowledge about the industrial system, a theoretical model may be hypothesized and then a CFA could be fitted.
- In case of limited knowledge, EFA can be used be to extract the factors; then an orthogonal rotation can be applied to the extracted factors in order to simplify the model structure.

There are gains in interpretability when a FA model can be fitted in lieu of PCA. Parameter fitting, however, requires constrained non-linear optimization routines which are very sensitive to the initial values; as a result, blindly trying to use the model for non-linear data is not as straightforward as PCA and may not be always possible.

Implicitly, both FA and PCA are best suited for linear processes in steady state, i.e. such that the observations are *independent* from one another. Additionally, FA requires the observations to be *identically distributed*. Whereas there are no explicit distributional assumptions with PCA, there is a requirement in the monitoring stage for the residuals to be independent and normally distributed; in turn, for these latter assumptions to be met, both independence and some degree of normality will be needed in the original observations [3].

In dynamic processes, the current values of the variables will depend on the past values and, therefore, observations will no longer be *independent*. Neither of the models discussed so far are able to account for this time-dependency and, if applied

[3]This is related to the Central Value Theorem. Linear combinations of variables which are not normally distributed will tend towards a normal distribution.

to dynamic systems, they will result in residuals with structure. There exist variants of PCA which are able to account for this time-dependency; the idea is to expand, column-wise, the observations matrix by appending time-shifted versions of itself at different lags; this approach that has referred to as Dynamic PCA in the literature [Ku et al., 1995]. In the remaining chapters emphasis is shifted towards models which can handle both observational-dependency and non-linearities. Much of the data generated in industrial systems are characterized by these two features.

Chapter 3

Process monitoring with Gaussian process latent variable models

Part of Chapter 1 has devoted its attention to look at linear latent variable models in the context of process monitoring. Linear PCA models are extensively used, regardless of whether the process is linear or not; they have the advantage that models can be fitted straightforwardly but also the dangers of failing to capture the underlying process non-linearities. As an alternative to that approach we proposed in Chapter 2 that, as long as the the process under study is reasonably linear, a factor analysis model could also be fitted. While fitting this model is not as effortless, the FA approach brings about advantages of being able to relate the latent variables with a subset of the original variables by exploiting their correlation.

With the majority of industrial processes behaving non-linearly, the question that arises at this stage is about whether feasible non-linear alternatives can be developed; those procedures should also retain the advantages of dimensionality reduction achieved both with PCA and FA but also the partial *variable selection gain* attained when fitting a FA model. And, above all, they should target at finding the underlying latent variables which are driving the observations.

There exist several *PCA extensions* to non-linear systems in the literature; a brief summary is given in the next section. The focus on this chapter is, by building

on the Gaussian Process Latent Variable (GPLV) model developed by Lawrence [2004, 2005], to develop yet another one. As the name implies, the backbone of the procedure is a Gaussian process regression, GPR, model. The most usual setting under which Gaussian processes are used, as introduced in Section 1.3, is to map a multivariate input into a univariate response. If, given the inputs, the multivariate outputs can be assumed independent, Lawrence's approach relies on combining several GPRs so that multivariate responses can be jointly modelled. This idea is very powerful as it allows GPRs to approximate complex multivariate non-linear systems with relative simplicity.

The regression performance of GPRs is, at least, comparable to that achieved via artificial neural networks (NN) and, in fact, both methods are intrinsically related. As Neal [1994] has shown, when the number of nodes in the hidden layer of a neural network tends to infinity the NN converges to a Gaussian process. There exist applications of neural networks to *fault detection and diagnosis* as shown by Tan and Mavrovouniotis [1995] which are successful at modelling non-linear systems within a process monitoring scheme. This chapter aims to show how GPRs can also be used for the same purpose; the advantage being that a lesser number of model parameters are needed to build the nonlinear map between the process inputs and outputs.

There are some recent applications of the GPLV model to process monitoring in chemical engineering [Ge and Song, 2010]. We review that existing approach, highlight its limitations and compare the procedure performance against other well used nonparametric methods. In addition, we propose a new procedure to the way in which new observations are mapped into the non-linear latent space determined by the GPLV model; this whole chapter is based on the work of Serradilla et al. [2011].

3.1 Nonparametric approaches to process monitoring

The fact that PCA has been widely used to model non-linear systems is perhaps related to is intrinsic simplicity. Implicitly, in doing so, the PCA model is used

as a black-box or dimensionality-reduction artefact where the number of principal components retained has no resemblance with the real underlying dimensionality of the problem. A very clear example of this is given by Simoglou et al. [2000], who managed to identify a problem in an industrial system by looking at principal components that were explaining very little of the total variance in the system covariance matrix.

If we are to capture process non-linearities efficiently more complex models are needed. A way of doing so, while still using PCA, is through what Gnanadesikan [1977] defines as *generalized PCA*. The idea is to extend the Q-dimensional vector $\boldsymbol{x}$ into a new input vector $\boldsymbol{x}'$ which, while still containing the original variables in $\boldsymbol{x}$, is enlarged by using non-linear functions of those variables. Subsequently, linear PCA is performed in the augmented input space. The key to this approach is to decide on the appropriate dimensionality of $\boldsymbol{x}'$ as well as the non-linear relationships between the original variables needed to describe the system. This drawback can be removed by using a function $\Phi : \boldsymbol{x} \in \mathcal{R}^Q \mapsto \boldsymbol{x}' \in \mathcal{R}^F$ which automatically carries out the non-linear mapping of the input space into an arbitrarily high-dimensional space (or *feature space*, as known in the machine learning community), where $F \gg Q$. It turns out that this mapping can be performed implicitly by using kernel covariance functions and therefore Φ does not need to be specified [Schölkopf et al., 1998]; this approach is known as *kernel PCA* and has been shown to have an excellent performance in the monitoring of non-linear systems [Choi et al., 2005]. Nevertheless, there is a cost incurred in achieving such performance and that comes in terms of the lack in model interpretability.

It is possible to achieve performances similar to those of kernel PCA algorithms by using GPLVM-based approaches. Briefly, the idea is to consider a set of GPRs to map the input space variables, $\boldsymbol{x} \in \mathcal{R}^Q$, into the observational space, $\boldsymbol{y} \in \mathcal{R}^D$. Note that *a priori* the input positions $\boldsymbol{x}$ are unknown and therefore need to be determined. In a second step, when new observations become available, we first project them onto the latent space and subsequently onto the original observational space. This approach shares similarities to the non-linear principal component analysis based on principal curves, NLPCA, developed by Dong and McAvoy [1996]. Let $\mathbf{Y} \in \mathcal{R}^{N \times D}$ be our original observations and $\mathbf{X} \in \mathcal{R}^{N \times Q}$ the corresponding latent variable

representation. Dong and McAvoy's approach relies on an additive model, i.e.

$$\mathbf{Y} = \sum_{i=1}^{Q} f_i(x_i) + \mathbf{E}$$

where $\mathbf{E}$ is a matrix of model errors and f_i a non-linear function of the input variables. This model assumes that the original observations are generated as a linear combination of $Q-$univariate non-linear functions; the latent variables must therefore be determined one at a time. The GPLV model, on the other hand, is not restricted to additive models and can account for multiplicative effects as all the latent variables are determined simultaneously. The GPLV model is also closely related to the concept of Input-Training neural network, IT-net, proposed by Tan and Mavrovouniotis [1995]. The idea is that the net input variables are not fixed but adjusted along with internal network parameters so that it can reproduce the net output more efficiently. Jia et al. [1998] have shown how a process fault can successfully be detected using the IT-net to map the latent variables into the observations that have been compressed via PCA. For a given prediction performance, an advantage of using the GPLV model over the IT-net is that it requires a substantially lower number of parameters; it is also a full probabilistic model where prediction uncertainty and hypothesis testing can be carried out if necessary.

The GPLVM is first described in next section. We then describe what approaches could be taken to project new observations onto the model space, Section 3.3; this is a crucial step in a process monitoring and control scheme. Finally, both a simulation example and a real application are given in Section 3.5..

3.2 Gaussian process latent variable models

There are two main differences between a *normal* GPR and a GPLV model. Firstly, in the latter, the input positions, $\boldsymbol{x}$, are not given; and, secondly, it can also be used to model a multivariate output. Therefore, when working with GPLV models, the purpose of the inference procedure is not only to determine the best value of $\boldsymbol{\theta}$, the covariance function hyper-parameters, but also the best value of the latent input positions, $\mathbf{X}$.

Let us consider a new dataset $\mathcal{D} = \{\boldsymbol{y}_i|_{i=1}^N,\ \boldsymbol{y}_i \in \mathcal{R}^D\}$, which is made of N D-dimensional observations. Instead of a collection of N-observations, the dataset can also be thought of a collection of D-variables, i.e. $\mathcal{D} = \{\boldsymbol{y}_{(d)}|_{d=1}^D,\ \boldsymbol{y}_{(d)} = (y_{1d}, \ldots, y_{Nd}) \in \mathcal{R}^N\}$. A Gaussian process latent variable model [Lawrence, 2005] is defined as

$$\begin{aligned} y_{dn} &= f_d(\boldsymbol{x}_n) + \varepsilon_{dn}, \\ f_d(\boldsymbol{x})|\boldsymbol{x} &\sim \mathcal{GP}_d(0, k(\boldsymbol{\theta}_d); \boldsymbol{x}), \\ \varepsilon_{dn} &\sim \mathcal{N}(0, \sigma_d^2) \end{aligned} \tag{3.1}$$

for $d = 1, \ldots, D$ and and $n = 1, \ldots, N$. Here, $\boldsymbol{\theta}_d$ are the parameters involved in the d^{th} $\mathcal{GP}$ for y_d; in this chapter we are assuming that they are all the same, i.e. $\boldsymbol{\theta} = \boldsymbol{\theta}_1 = \ldots = \boldsymbol{\theta}_d$. Therefore, the model is simply a stochastic mapping, using the same GPR, between $\boldsymbol{x}$, the Q-dimensional latent space, and each output dimension y_d. It is relatively straightforward to extend the model to the case in which the GPR parameters, $\boldsymbol{\theta}_d$, are not the same.

In the context of a monitoring scheme, we observe $\mathcal{D}$ and aim to build a map to the unobserved $\mathbf{X}$ in the Q-dimensional latent space ($Q \ll D$); this latent space is subsequently used for fault detection and diagnosis.

3.2.1 GPLV model inference

Training of the GPLV model is the procedure whereby both the latent variables, $\mathbf{X}$, and the GPR parameters, $\boldsymbol{\theta}$, are determined. In order to do that, firstly, the joint marginal distribution for $\mathbf{Y}$, the $N \times D$ matrix of observations, can be written as

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) \sim \prod_{d=1}^{D} \varphi(\boldsymbol{y}_{(d)}; \mathbf{0}, \mathbf{K}_y)$$

where $p(\cdot)$ denotes the probability density function and $\varphi(\cdot; \mathbf{0}, \mathbf{K}_y)$ is the Gaussian density with its corresponding mean and covariance matrix. The associated log-

likelihood can then be expressed as

$$\ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y}) = -\frac{D}{2}\log|\mathbf{K}_y| - \frac{1}{2}\operatorname{tr}(\mathbf{K}_y^{-1}\mathbf{Y}\mathbf{Y}^\intercal) \tag{3.2}$$

where the constant terms have been omitted. Maximization of the previous function is, however, not possible without additional identifiability constraints. By giving a Gaussian prior distribution to each latent variable, $\boldsymbol{x}_i \sim \mathcal{N}(0, \mathbf{I}_Q)$, then $\mathbf{X} \sim \prod_{i=1}^{N} \mathcal{N}(0, \mathbf{I}_Q)$. Hence

$$p(\mathbf{X}) \propto \exp\left\{-\frac{1}{2}\operatorname{tr}(\mathbf{X}\mathbf{X}^\intercal)\right\}$$

and the posterior distribution is given by:

$$p(\mathbf{X}, \boldsymbol{\theta}|\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})p(\mathbf{X})p(\boldsymbol{\theta}). \tag{3.3}$$

We can then calculate the *maximum a posteriori* (MAP) solution with respect to the latent factor scores, $\mathbf{X}$, and the unknown parameters, $\boldsymbol{\theta}$, by maximizing the following log-likelihood

$$\ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y})_{MAP} = \ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y}) - \frac{1}{2}\operatorname{tr}(\mathbf{X}\mathbf{X}^\intercal) \tag{3.4}$$

where constant terms have been omitted and a non-informative prior for $\boldsymbol{\theta}$ has been used.

The *empirical Bayes estimate* solution for the GPLV model can be found by jointly maximizing Equation (3.4) with respect to $\mathbf{X}$ and $\boldsymbol{\theta}$. The model log-likelihood is both non-linear and *non-convex*. Due to the high-dimensionality of the problem, a global solution cannot be guaranteed and multiple local maxima may occur. Further details about the solution procedure of the model are given in Appendix B.2.1.

3.2.2 GPLV model prediction

The GPLV model prediction for a new but known input vector $\boldsymbol{x}^*$ is an extension of Equation (1.12) to every output variable $\boldsymbol{y}_{(d)}$. Let us define $f_M(\boldsymbol{x}^*) = (f_1(\boldsymbol{x}^*), \ldots, f_D(\boldsymbol{x}^*))^\intercal$. The joint distribution of the new enlarged matrix of out-

puts $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N, f_M(\boldsymbol{x}^*))^\intercal$ will still be multivariate normal; the prediction, $\widehat{\boldsymbol{y}}^*$, of $f_M(\boldsymbol{x}^*)|\mathcal{D}$ is also a multivariate normal distribution whose mean and common variance are given as

$$\begin{aligned} E(f_M(\boldsymbol{x}^*)|\mathcal{D}) &= \mathbf{Y}^\intercal \mathbf{K}_y^{-1} \boldsymbol{k}^*, \\ \text{Var}\left(f_M(\boldsymbol{x}^*)|\mathcal{D}\right) &= \left(k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}^{*\intercal} \mathbf{K}_y^{-1} \boldsymbol{k}^*\right) \mathbf{I}_D \end{aligned} \tag{3.5}$$

where, as before, $\boldsymbol{k}^* = (k(\boldsymbol{x}^*, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}^*, \boldsymbol{x}_N))^\intercal$ is the vector of covariances between the new input point, $\boldsymbol{x}^*$, and the training data $\boldsymbol{x}_i$, $i = 1, \ldots, N$; $\mathbf{I}_D$ is the D-dimensional identity matrix.

3.2.3 Big sample sizes: the active set

Numerically, irrespective of whether a full or an empirical Bayes approach is used to obtain the model estimates $\widehat{\mathbf{X}}$ and $\widehat{\boldsymbol{\theta}}$, the inverse of the covariance matrix, $\mathbf{K}_y^{-1}$, is involved in Equation (3.4). The cost of the log-likelihood evaluations is, hence, of order $O(N^3)$, where N is the sample size. As N increases, model training slows down as a result of the cost of the calculations but also due to the increased dimensionality of the problem. This, in turn, may render the algorithm impractical for many of the data sets available in industry.

As it is the case with a GPR, the model hyper-parameters must be positive. Furthermore, there is an identifiability problem in the model log-likelihood which is further explained in Section 4.5. This makes necessary to introduce additional numerical constraints to prevent the kernel hyper-parameters from becoming excessively large; otherwise the optimization becomes unstable. Alternatively, an informative prior for $\boldsymbol{\theta}$ can be introduced in Equation (3.4) which has a penalty-like effect, discouraging large values.

For those cases where the nominal data set is substantially large[1] , training of the GPLV model can be sped up by selecting a subset $\mathcal{I}$ of size m, with $m \ll N$, from the original data set $\mathcal{D}$. Let us denote the remaining (unselected observations) as

[1] What is meant by *large* depends on the computing power available. As a practical rule, samples where $N \simeq 2 \cdot 10^2$ may start slowing the optimization down considerably.

$\mathcal{J}$. By replacing $\mathcal{D}$ with $\mathcal{I}$, computational efficiencies are gained as the cost of the likelihood calculation will be of order $O(m^3)$ rather than $O(N^3)$. $\mathcal{I}$ is normally referred to as the *active set* and, obviously, its selection causes a reduction in the information available for inference [Shi and Choi, 2011, Section 3.3]. What it is expected is that, if a good subset selection is made, most of the information will be kept. There are several criteria that can be used to partition $\mathcal{D}$ into $\mathcal{I}$ and $\mathcal{J}$. The most popular ones are probably based on the Kullback-Leibler divergence criterion and the process entropy. The latter criterion is used by the *Informative Vector Machine*, IVM, algorithm [Lawrence et al., 2003] which sequentially selects the points in $\mathcal{I}$ according to the reduction in the process' entropy that they cause. An IVM implementation of the GPLV model can be found in Lawrence [2005].

3.3 Projecting new observations onto the latent space

Given a training (nominal) set of D-dimensional observations $\mathbf{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)^\intercal$, their representation in the latent space can be found by maximizing Equation (3.4). In other words, both the latent variables $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^\intercal$ and $\boldsymbol{\theta}$, the GPR parameters, can be considered known once the optimization is completed. The model prediction, $\widehat{\mathbf{Y}}$, can then be easily found by applying Equation (3.5).

Let us now say that a new observation $\boldsymbol{y}_j = (y_{j1}, \ldots, y_{jD})^\intercal$ becomes available (for notational convenience, we use $\boldsymbol{y}_j$ instead of $\boldsymbol{y}^*$). The problem of projecting that observation onto the latent space is concerned with finding $\boldsymbol{x}_j$, its associated latent variable representation. We provide two possible ways of doing so.

3.3.1 MAP projection

Equation (1.12) is a standard result from nonparametric Gaussian process regression. For clarity, it can also be expressed as

$$\boldsymbol{y}_j | \boldsymbol{x}_j, \mathbf{X}, \boldsymbol{\theta} \sim \mathcal{N}_D(\widehat{\boldsymbol{y}}_j, s_j^2 \mathbf{I}_D) \tag{3.6}$$

where

$$\begin{aligned} \widehat{\boldsymbol{y}}_j &= \mathbf{Y}^\intercal \mathbf{K}_y^{-1} \boldsymbol{k}_j, \\ s_j^2 &= k(\boldsymbol{x}_j, \boldsymbol{x}_j) - \boldsymbol{k}_j^\intercal \mathbf{K}_y^{-1} \boldsymbol{k}_j + \sigma^2 \end{aligned} \tag{3.7}$$

and $\boldsymbol{k}_j = (k(\boldsymbol{x}_j, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}_j, \boldsymbol{x}_N))^\intercal$. Note that, as we observe $\boldsymbol{y}_j$ and not $f(\boldsymbol{x}_j)$, the uncertainty is higher and reflected via σ^2.

In Equation (3.6), $\mathbf{X}$ and $\boldsymbol{\theta}$ are treated as given and evaluated at their MAPs as discussed in previous sections. Thus, the log-likelihood in terms of $\boldsymbol{x}_j$ can be written as

$$\begin{aligned} \ell(\boldsymbol{x}_j; \boldsymbol{y}_j, \mathbf{X}, \boldsymbol{\theta}) = &- \frac{D}{2} \log(2\pi) - \frac{D}{2} \log(s_j^2) \\ &- \frac{1}{2(s_j^2)} (\boldsymbol{y}_j - \widehat{\boldsymbol{y}}_j)^\intercal (\boldsymbol{y}_j - \widehat{\boldsymbol{y}}_j). \end{aligned} \tag{3.8}$$

Additionally, by giving a Gaussian prior distribution to the latent variable $\boldsymbol{x}_j$, that is $\boldsymbol{x}_j \sim \mathcal{N}(0, \mathbf{I}_Q)$, then

$$p(\boldsymbol{x}_j) \propto \exp\left(-\frac{1}{2} \boldsymbol{x}_j^\intercal \boldsymbol{x}_j\right).$$

The MAP can therefore be found by maximizing the following log-likelihood function

$$\ell_{MAP}(\boldsymbol{x}_j; \boldsymbol{y}_j, \mathbf{X}, \boldsymbol{\theta}) = \ell(\boldsymbol{x}_j; \boldsymbol{y}_j, \mathbf{X}, \boldsymbol{\theta}) - \frac{1}{2} \boldsymbol{x}_j^\intercal \boldsymbol{x}_j \tag{3.9}$$

where constant terms have been omitted.

The same *scaled conjugate gradient* optimiser described in Appendix B.2.1 can be employed to determine $\boldsymbol{x}_j$; now the objective function to maximize is given by Equation (3.9) and the gradients thereof with respect to $\boldsymbol{x}_j$ are given in Appendix B.2.3. This is the method used both by Lawrence [2005] and Ge and Song [2010]; we should, however, be cautious when using it as the objective function given by Equation (3.9) is non-convex. A procedure must be put in place to make sure that the global maximum is chosen when projecting every new observation. While this is relatively simple when the underlying dimensionality of the latent space is low, the problem is far from trivial when this is not the case. Likewise, this approach becomes more

uncertain when applied to fault detection since new observations may come from a (faulty) system which might be different from the system we used to train the model. This problem will be further explained in Section 3.5 via a simulation example; refer also to Figure 3.4.

3.3.2 Neural Network (NN) projection

The procedure we prefer to follow in order to use the GPLV model for process monitoring is to build two neural network models; a similar idea has been used by Dong and McAvoy [1996] who based their method on the principal curves algorithm proposed by Hastie and Stuetzle [1989]. By doing this we avoid dealing with the non-convexity problem altogether.

The first NN, *Net 1* as shown schematically in Figure 3.1, is used to map the standardized D-dimensional input observations onto the underlying Q-dimensional latent variables as determined by the GPLV model. The second NN, referred to as *Net 2* in Figure 3.1, maps the Q-dimensional latent variables onto the D-dimensional GPLV model prediction, $\widehat{\boldsymbol{y}}$, as given by Equation (3.5). Hence, model learning in both neural networks is based on the observed data, $\mathcal{D}$, and the related latent variables, $\mathbf{X}$, estimated as described in Section 3.2.

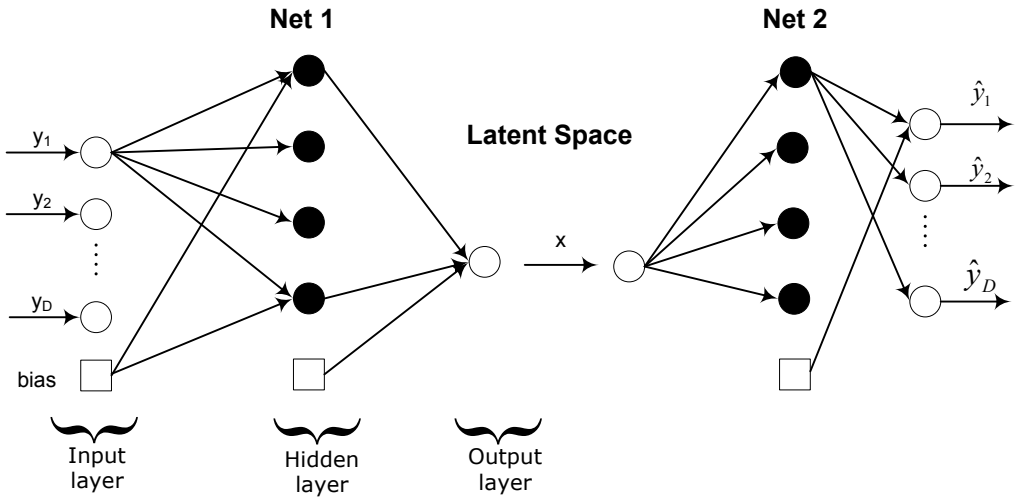

Figure 3.1: Architecture of the neural networks needed for process monitoring; only 1 latent variable.

Feed-forward neural networks architectures [Bishop, 2006, Section 5.1] with one hidden layer in both networks are appropriate to carry out the mappings. Hyperbolic tangent sigmoid transfer functions were used as activation functions in the hidden layer and the identity transfer function was chosen for the output layer in the examples that follow. Network training was carried out using a *scaled conjugate gradient backpropagation* algorithm implemented in MATLAB [2010]. Once the GPLV model has been fitted and both networks trained, the only remaining unknown in the training process is M, the number of nodes in the hidden layers. This parameter is adjusted in order to achieve the best predictive performance; it controls the total number of network parameters (model complexity) so we can expect an optimum value to exist giving the best generalisation performance.

Bishop [2006, Section 5.5] cites different procedures that could be used for this purpose. The method we have followed to control network complexity is *early-stopping*. The available data is divided into three subsets. The first subset is the *training set*, used to compute gradients and the network parameters. The second subset is the *validation set* whose error is monitored during the training process. The training set error is a non-increasing function of the iteration index. On the other hand, the validation set error normally decreases during the initial phase of training; however, as the network begins to overfit the training data, the error of the validation data set will typically begin to rise. When this latter error increases during six consecutive iterations, training is stopped and the network parameters at the minimum of the validation error are adopted. The third subset is the *test set*, which it is only used to assess the generalization performance of the network.

Prediction is straightforward once both networks have been fully trained. For a new observation $\boldsymbol{y}_j$, *Net 1* will output the corresponding latent variable $\boldsymbol{x}_j$; this will then be used as the input for *Net 2* which will, in turn, output the model prediction $\widehat{\boldsymbol{y}}_j$.

In fact, projections between the original, the latent spaces and vice versa need not be restricted to neural networks. Other nonparametric approaches would also be suitable to build the links; in this case, Gaussian processes do provide an excellent alternative as it can be seen in Figure 3.6.

3.4 Monitoring strategy

In the examples that follow a comparison will be made between GPLV-based models and kernel PCA. It is relevant to emphasize what the main difference between these nonparametric methods are. In that respect, the idea behind kernel PCA is similar to that of a generalized linear model which uses a nonparametric link function: an appropriate kernel function needs to be chosen so that the process can be properly modelled. By contrast, the GPLV-based model aims at describing the non-linear relationships directly; and, it does so by seeking the process underlying dimensionality. As a monitoring method it should, therefore, be more flexible and suitable in modelling any type of non-linear stochastic system.

The monitoring statistics that will be used to monitor the process were described in Section 1.2.4; in particular, we will be using plots of the *squared prediction error* (SPE) as the fault introduced in the simulated case study changes the (non-linear) relationship between the process variables following the fault classification made by Zhang et al. [1997].

The monitoring strategy can be summarized as follows:

A.- **Nominal model**

i. Select the nominal data $\mathbf{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)^\intercal$ from observations where the process is known to be behaving as intended.

ii. Select the number of latent variables Q. This value could be set using the user's theoretical knowledge of the system under study if available. Alternatively, it could be based on a desired percentage of the variance explained (see e.g. Table 3.2).

iii. Build the GPLV model. The outputs from this model will be the latent variables, $\mathbf{X}$, as well as the GPR model parameters, $\boldsymbol{\theta}$.

iv. Use the fitted model to find the confidence limits for the SPE or any other statistics used.

B.- **New Observations**

Once new observations become available they can be projected onto the reduced subspace by any of the following methods:

i. *Method 1:* MAP projection. Caution must be exercised before using this procedure following the discussion given at the end of Section 3.3.1.

ii. *Method 2:* NN projection. This requires the construction of two auxiliary neural network models. The mappings are as follows:

$$\textbf{Net-1: } \mathbf{Y} \in \mathcal{R}^D \mapsto \mathbf{X} \in \mathcal{R}^Q$$
$$\textbf{Net-2: } \mathbf{X} \in \mathcal{R}^Q \mapsto \widehat{\mathbf{Y}} \in \mathcal{R}^D$$

iii. For every new observation j, calculate SPE_j or any other statistic that is being used to monitor the process.

3.5 Case studies

The performance of the GPLV-based model is analysed in this section with two examples. The first looks at simulated data that have appeared in the literature and will be used to compare the method with some of its nonparametric peers. The second example refers to data coming from a continuous stirred tank reactor (CSTR) that has also been widely used in the chemical engineering literature.

3.5.1 Simulation example

This first example refers to the system presented by Choi et al. [2005]. There are three variables, $D = 3$, but only one underlying latent variable, $Q = 1$. The data is simulated by

$$\begin{aligned} y_1 &= x + \varepsilon_1, \\ y_2 &= x^2 - 3x + \varepsilon_2, \\ y_3 &= -x^3 + 3x^2 + \varepsilon_3. \end{aligned} \tag{3.10}$$

where x is generated from a uniform distribution $\mathcal{U}(0.01, 1)$; the independent noise ε_d is generated from a Gaussian distribution $\mathcal{N}(0, 0.01^2)$ for $d = 1, 2, 3$.

The nominal data set is made of 100 observations generated with Equation (3.10). As an independent data set to test type I errors (false alarms) 100 additional observations (samples 101-200) of normal operating data are also generated from the same equations. A final data set of 100 faulty data observations (samples 201-300) is also simulated where y_1 and y_2 are obtained as before but with y_3 now given by

$$y_3 = -1.1x^3 + 3.2x^2 + \varepsilon_3. \tag{3.11}$$

The set of faulty data will be used to determine type II errors (missing alarms). For analysis purposes, we consider that an alarm is triggered when the SPE statistic has a value higher than the 99% control limit. Figure 3.2, panels (a)-(c), shows the data sets for both, the normal and fault conditions as 2-D plots for every combination of the dependent variables and from one realization of this system. Notice the similarity

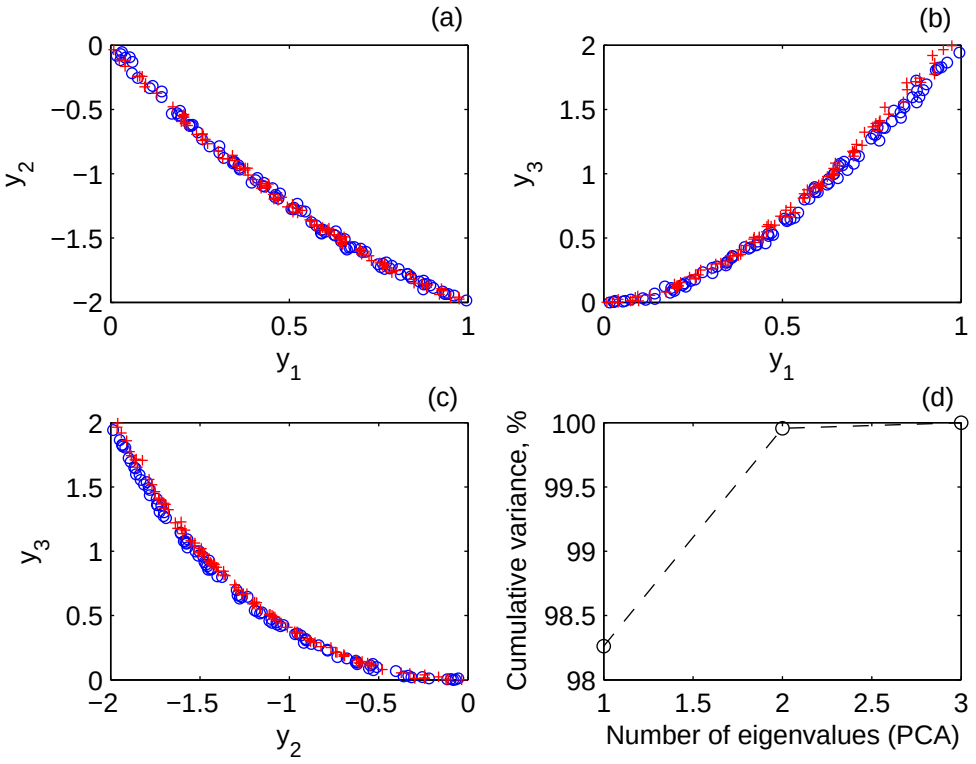

Figure 3.2: Data sets for normal condition (o) and fault condition (+): (a) y_1 vs. y_2, (b) y_1 vs. y_2 and (c) y_2 vs. y_3. Panel (d) represents the cumulative variance accounted for the linear principal components.

between them; likewise, the fault in the y_3 direction is not easily identifiable by visual inspection. Within the range for the independent variable, data have also low noise and mild non-linearities which explains why one principal component accounts for more than 98% of the total variance, Figure 3.2, panel (d). All the data has been scaled to zero mean and unit variance.

Firstly, we show how to train the GPLV model for one simulation and highlight the problem that arises when projecting new observations using the MAP projection method. Then a graphical comparison based on this unique simulation is made among the non-linear GPLV model, with both MAP and NN projections, and linear PCA. Secondly, a more complete analysis of robustness (false alarm rate) and sensitivity (missing alarm rate) is carried out by looking at type I and type II errors respectively based on 200 simulations; a comparison with kernel PCA is also performed.

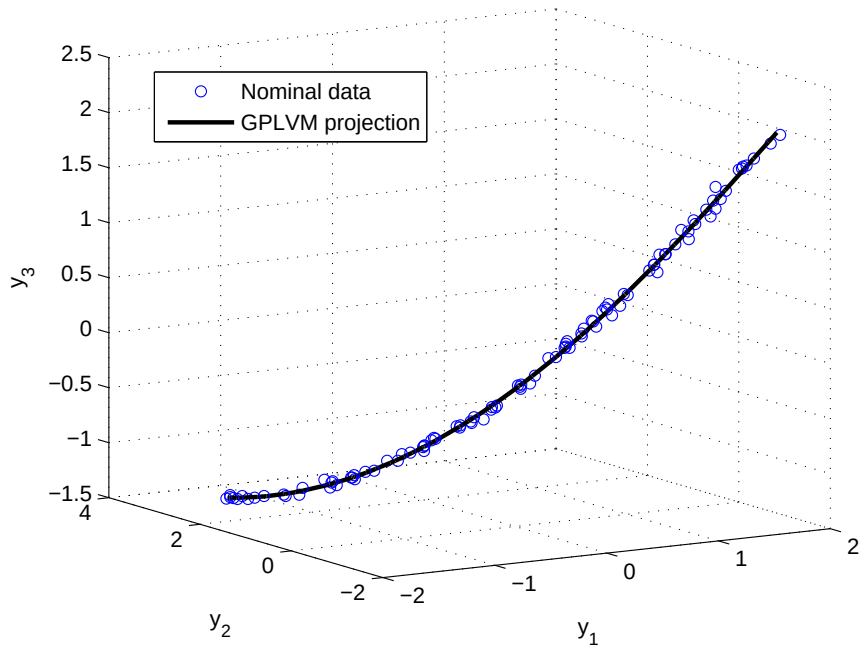

Figure 3.3: Normalized *nominal data* and the GPLV model prediction.

Model training

The objective is to model the *nominal data* in the previous system by using the non-linear GPLV model defined in Section 3.2. Latent positions were initialized using linear PCA while the GPR parameters were given random positive values. The prediction for the training data, as given by Equation (3.5), is shown in Figure 3.3. As it can be seen the GPRs do provide an excellent and smooth approximation to the data.

One of the advantages of using this simulation is that the generating latent variable, $\boldsymbol{x}$, is fully known. It can therefore be compared with its estimate, $\widehat{\boldsymbol{x}}$, obtained by fitting the GPLV model. The correlation coefficient is $\text{cor}(\boldsymbol{x}, \widehat{\boldsymbol{x}}) = 0.999$, thus also showing the suitability of the proposed model for this non-linear system. Due to the low levels of noise in the system, this latent variable represents 99.9% of the total variance (as opposed to the 98.3% variance accounted for one principal component).

New observations: MAP and NN projections

We have generated 100 samples of independent data and 100 samples with a known fault in y_3. Before projecting every observation onto the latent space, let us focus on any single one of the independent samples, which we denote as $\boldsymbol{y}_j$. The aim of the MAP projection is to determine the latent variable $\boldsymbol{x}_j$ associated with the available observation. As previously explained, this can be done by maximizing Equation (3.9) with respect to $\boldsymbol{x}_j$. In this case, as the latent space is mono-dimensional, the log-likelihood can also be visualized for different values of $\boldsymbol{x}_j$ as shown in Figure 3.4, left panel. What the plot highlights is that the objective function is not convex and, for this particular case, three maxima occur. Although not shown, the shape of this log-likelihood function is very sensitive to the value of $\boldsymbol{y}_j$ to the point that for some faulty observations it occurs that the global maximum switches between the middle and the left/right side maxima.

Figure 3.4, right panel, shows the result obtained for $\boldsymbol{x}_j$ by carrying out a blind optimization where the initial values of the latent variable $\boldsymbol{x}_j$ were set randomly by using a standard normal distribution. As it can be seen, the blind optimization leads to projections clustered in three groups which depend on the starting point chosen to initiate the algorithm; that, in turn, leads to an unacceptable number of type I errors (and to a spurious increase in type II errors); refer to the MAP-1 model in Figure 3.5 for further details. For an one-dimensional problem there is no complication in finding the global maximum; we simply choose several random starting points and select the one with the highest value of the target function. However, it is important to notice that for multivariate optimization problems, where very little information is available about the shape of the log-likelihood function, we will not be able to guarantee the fact that the global maximum is chosen systematically at all times. In this sense, caution must be exercised if the monitoring method proposed by Ge and Song [2010] was to be used.

For comparison purposes, Figure 3.5 shows the SPE for a linear PCA model (1 latent variable), GPLV-based models with MAP projections where either the optimization has been carried out blindly (MAP-1) or the global maximum has been chosen (MAP-2) and a GPLV model with a NN projection. In the case of PCA, there are no false alarms in the independent data and the number of missing alarms for faulty

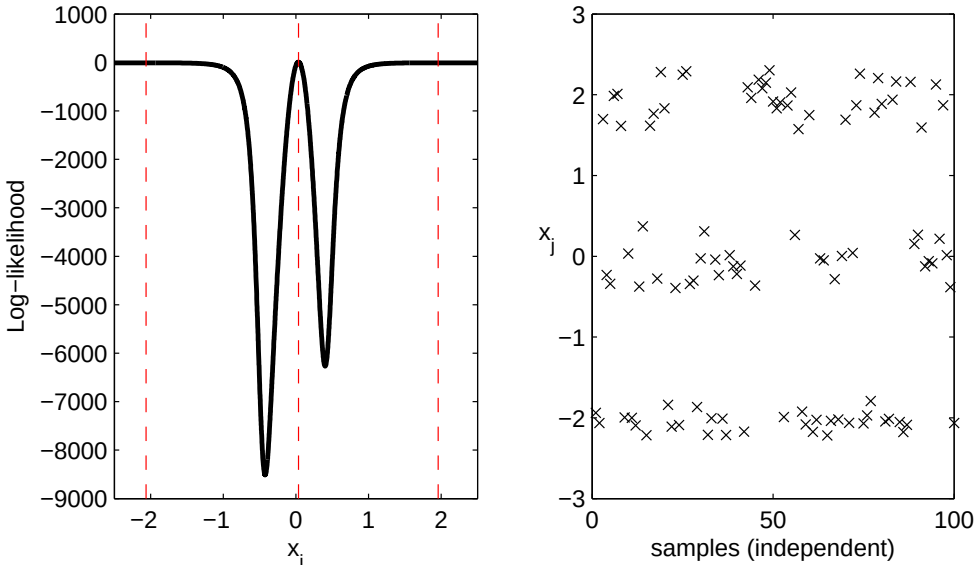

Figure 3.4: Left panel: log-likelihood (projection of an *independent observation*). Global maximum located at $x_j = 0.0347$; red vertical lines indicate the location of the maxima. Right panel: blind optimization (where no attempt to find the global maximum has been made) results of the independent samples.

data is 97. Visually, all the three regions (nominal, independent and faulty) are very similar clearly indicating that a linear PCA model would not be appropriate for a system of these characteristics. Note how a blind optimization leads to a MAP projection where the number of type I faults is inadmissibly high (74) for the method to be used. For the MAP-2 model, the number of false alarms is 1 and the number of missing alarms is 80; finally, for the NN projection the number of type I errors is 1 and the number of type II errors is 72. The fact that the percentage of missing alarms is relatively high in the last two cases is related to the fault being somewhat subtle as shown in Figure 3.2. However, a visual inspection of the SPE plots clearly reveals that the 'faulty' region is different from the rest which should help identify the problem in the plant; the SPE with a NN projection is clearly the best performer.

Note also the that fault does not show in the corresponding T^2 plots (not shown). That is related to the type of fault being analysed; the fault represented in Equation (3.11) has changed the relationship between the process variables and therefore it is expected that deviations from the model be mostly detected by the SPE statistic [Zhang et al., 1996].

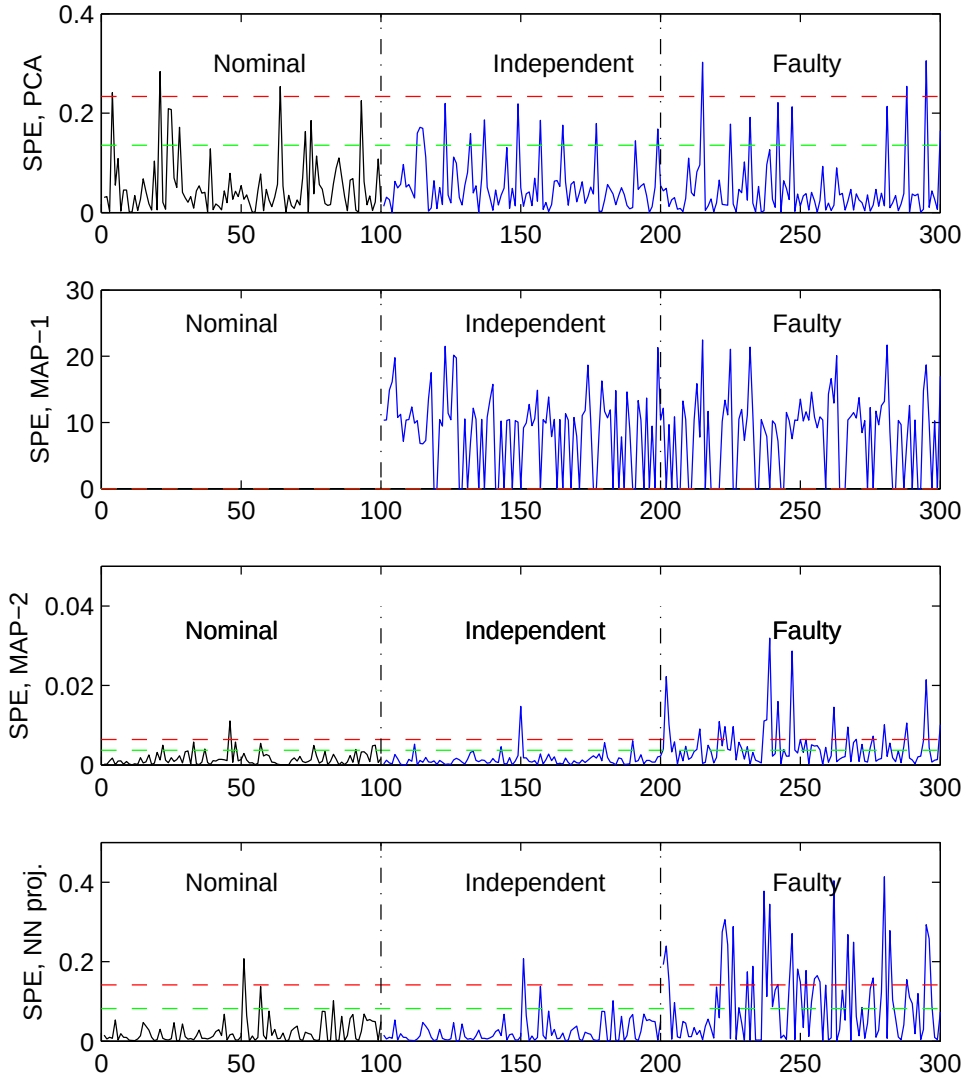

Figure 3.5: SPE for nominal, independent and faulty observations: 1^{st} panel - PCA model (1 PC), 2^{nd} panel - GPLV model with an MAP projection (MAP-1, blind optimization), 3^{rd} panel - GPLV model with an MAP projection (MAP-2, global maximum found) and 4^{th} panel - GPLV model with a NN projection. Dashed horizontal lines are the 95% and 99% confidence intervals.

Full simulation

In order to perform a full robustness and sensitivity study, 200 runs were carried out. GPLV models with MAP and NN projections are considered. Additionally, their performance is compared with that of kernel PCA. Full results are given in Table 3.1 and Figure 3.6. The results are quite similar to those given in the previous section. PCA hardly raises any alarm for faulty data. By contrast, kernel PCA gives the smallest type II error for faulty data; however, it also produces the highest type I error, 16.3%, for the independent data which is not acceptable in practice. This is evidence that kernel PCA is failing to properly model the non-linear relationships between the observed variables. The GPLV method with MAP and NN projections performs very well in terms of both types of error.

Method	Type I error (%)			Type II error (%)		
	IQR	Median	Mean	IQR	Median	Mean
LPCA	3.0	3.0	3.3	5.0	94.0	93.4
MAP proj.	4.0	3.0	3.4	13.0	64.0	65.4
GP proj.	4.0	3.0	3.3	13.0	64.0	65.8
NN proj.	4.0	4.0	4.2	7.0	81.0	79.3
KPCA	8.0	16.0	16.3	12.0	57.0	57.2

Table 3.1: Type I and type II error rates

Results presented in Table 3.1 and Figure 3.6 for the GPLV model with the MAP projection are based on the ones where the global maximum has been chosen systematically; these are achieved by using different starting values and by checking the values of the objective function. When a global maximum cannot be found (this would be the usual case when two or more latent variables are used), the MAP method will lead to an unreasonably high number of type I faults and then a NN projection should be the preferred method.

As discussed in Section 3.3.2, other non-linear methods could be used to map the real observations with the latent variables. Results in Table 3.1 and Figure 3.6 also include those in which the mapping has been carried out with GPR models; its performance is rather similar to that of the GPLV model with MAP and NN projections.

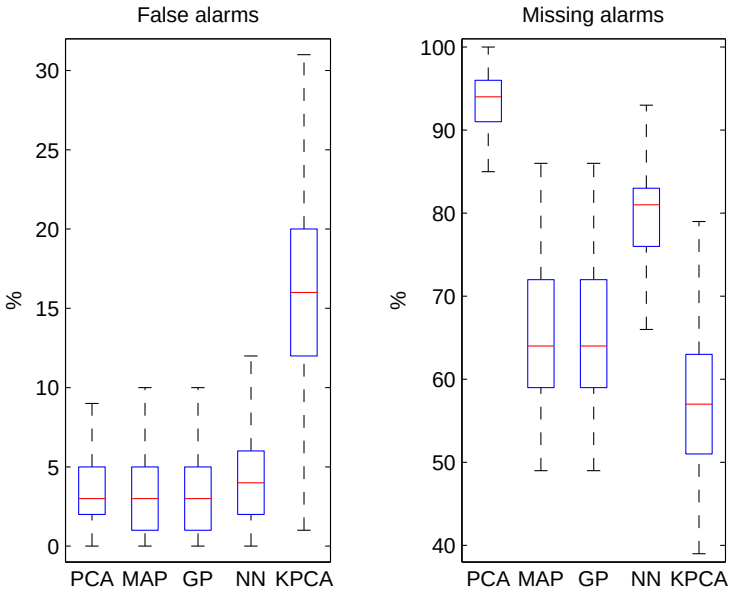

Figure 3.6: Full simulation results based on 200 runs. The numerical results are presented in Table 3.1. 1 PC selected in LPCA and 23 PC's for KPCA.

3.5.2 CSTR process

In this example, data is generated using the model for a non-isothermal continuous stirred tank reactor (CSTR); full details for it are provided in Appendix E. This example has been widely used in the literature to test other nonparametric methods; see for example, Lee et al. [2004], Yoon and MacGregor [2004], Choi et al. [2008] and Alcala and Qin [2010] amongst others.

Complex fault generation

Yoon and MacGregor [2001] categorize abnormal operating conditions as either *simple* or *complex* faults; in the former case, a fault occurring in one variable does not propagate into other variables whereas in the latter situation, the effect of the fault is seen by other process variables. To clarify this, let us generate 100 observations from the CSTR process and introduce a complex fault at $t = 50$ minutes; the fault is simply a bias of 1°C in the outlet temperature sensor. A time series plot of both

T and F_c is given in Figure 3.7. Note that as the outlet temperature is the controlled variable, the feedback controller will act to remove this bias at the expense of increasing the cooling water flow rate.

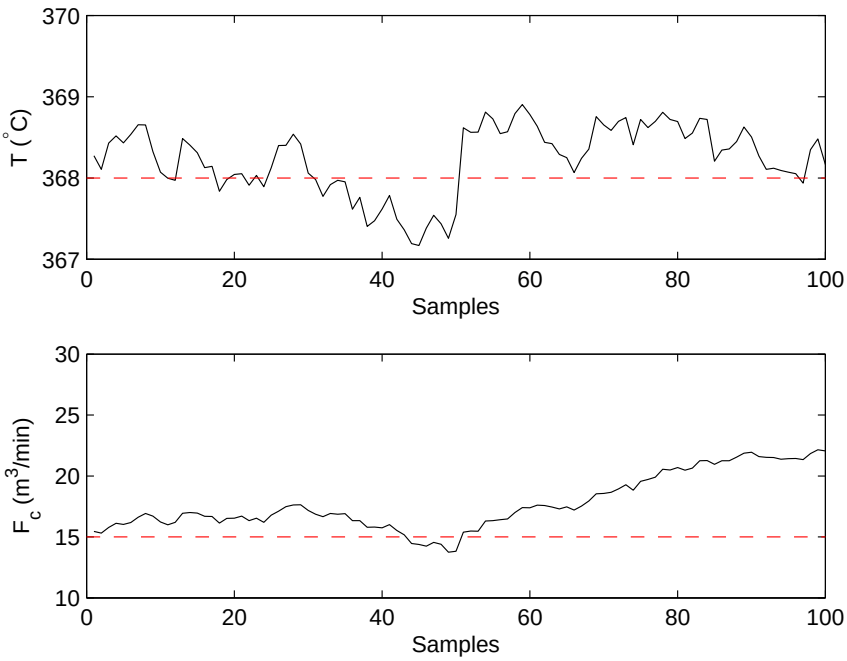

Figure 3.7: Bias fault of 1°C in the outlet temperature sensor occurring at $t = 50$ min.; controller set point at 368°C.

Complex fault detection

The training data is obtained by simulating the CSTR process for 200 minutes. A further 100 observations are generated containing the 1°C permanent bias in the outlet temperature sensor. The data has been mean centered and scaled to unit variance.

Two GPLV models, each one with the 200 observations from the training data, with one ($Q = 1$) and two latent variables ($Q = 2$) respectively, have also been built. To monitor the process, we have then used two *feed-forward* neural network models. The first network builds the map from $\mathbf{Y} \mapsto \widehat{\mathbf{X}}$, with 20 nodes in the hidden layer, while the second network takes back the observations from the latent space into their original dimensionality, i.e. $\widehat{\mathbf{X}} \mapsto \widehat{\mathbf{Y}}$ (25 nodes in the hidden layer).

Latent variables	Explained variance (%)	
	PCA	GPLV model
1	35.6	84.8
2	55.3	96.0
3	71.4	-

Table 3.2: Results for PCA and GPLV models

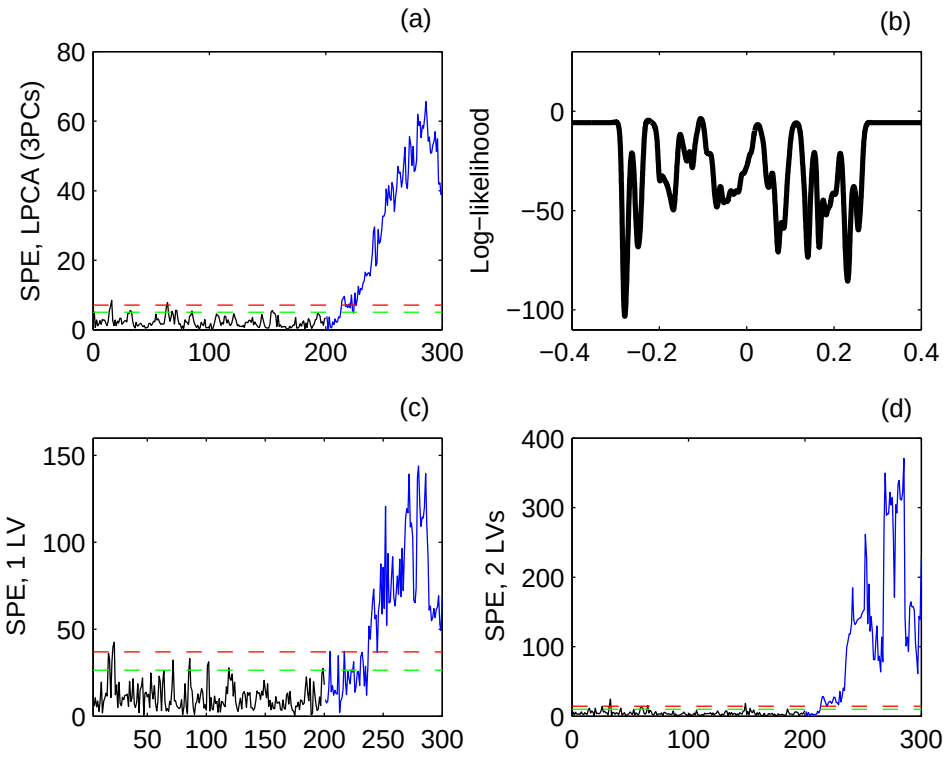

Figure 3.8: Panels (a): SPE for a linear PCA model with 3 PCs; (b) log-likelihood for a faulty observation (1 LV); (c) SPE for a GPLV-NN model with 1 LV; (d) SPE for a GPLV-NN model with 2 LVs. Horizontal dashed lines correspond to the 95% and 99% confidence limits.

Let us first consider the case with only one underlying dimension. As shown in Table 3.2, this latent variable is able to account for around 85% of the original variance. Figure 3.8, panel (c), shows the SPE for the GPLV model with a NN projection when $Q = 1$. As it can be seen, shortly after sample 200, the SPE starts moving pretty abruptly outside the confidence limits as a result of the bias

fault being introduced. To give some perspective into the problem of using the MAP projection, the log-likelihood given by Equation (3.9) for one of the faulty observations has been plotted in Figure 3.8, panel (b). If we were to use the MAP method, we would be dealing with the maximization of a similar-shaped function for every new sample that we wanted to project into the latent space.

The second GPLV model, with $Q = 2$, is able to explain about 96% of the variation in the original data; it, therefore, seems that $Q = 2$ should be very close to the true dimensionality of this non-linear system. As before, the SPE has been calculated and plotted in Figure 3.8, panel (d). It is very obvious, even by a visual comparison, that this latter model is far more sensitive than the model with only one latent variable. Not only the magnitude of the SPE for the training data reduces as a result of having an improved model but also the range in the faulty data SPE increases quite dramatically.

As a comparison, a linear PCA model with three principal components was selected by using 10-fold cross-validation. The percentage of variance explained as a function of the number of principal components kept in the model is given in Table 3.2. A linear model built this way would be able to detect the bias fault as seen in the SPE plot of Figure 3.8, panel (a), which is similar to the GPLVM with one latent variable. This shows clearly that a non-linear model should be used in this example.

3.6 Chapter summary

Ge and Song [2010] have proposed a GPLV-based model using a MAP projection as a way to monitor industrial processes. That approach, however, when used on fault detection tasks, is prone to the potentially serious pitfall of having to determine the global maximum of a likelihood function which is nonconvex (e.g. Figures 3.4 and 3.8, panel (b)). As the dimensionality of the latent space becomes larger, that problem becomes less and less trivial due to the non-convexity problem of the likelihood function when projecting new observations into the latent space.

To deal with the aforementioned problem, the key step in fault detection, we propose the use of two additional nonparametric models. Figure 3.1 displays this idea by using two NN models; this is in line with previous approaches that have been successfully applied. Other nonparametric projection methods such as a Gaussian process regression could also be employed in that step. The modelling of non-linear relationships between process variables is still a challenging problem when we have very little prior knowledge. There exist some non-linear methods, for example kernel PCA [Lee et al., 2004] and NLPCA [Dong and McAvoy, 1996]; or, alternatively, as we are proposing in this thesis, a GPLV-based model. By using simulated data we have shown how this class of models can unravel complicated non-linear relationships and find the underlying latent variables driving the process; the models have also shown high robustness and a good balance between robustness and sensitivity.

Stationary processes are characterized by observations which are independent from one another. That feature, however, is lost in dynamic systems. $\mathcal{GP}$-based models are able to account for variable dependency in a natural fashion; further ability to model time dependency could be explicitly incorporated into the model via kernel parameters which depend on time, i.e. $\boldsymbol{\theta}(t)$, or through a mixture model type of formulation. The latter avenue has not been pursued in this thesis. Likewise, only Gaussian-noise distributions have been covered. It could be argued that, from a process monitoring perspective, this is a requirement so that the proposed residual and model-based statistics can be used. But the Gaussian process latent variable model need not be restricted to Gaussian noise and could be extended to account for other non-Gaussian distributions. Similar extensions have already been proposed in the literature for Gaussian process regression models [Wang and Shi, 2011].

The pairing of the GPLVM/Projection algorithms can be seen as an extension to non-linear systems of the PCA idea for linear systems: we now have latent variables which are *non-linear combinations* of all of the original observations. Although these new latent variables could be representative of the underlying dimensionality of the system, they however lack physical interpretation which simply makes the problem of fault diagnosis more demanding. Or, in other words, we are able to non-linearly model the industrial system but still have an unresolved problem with variable selection. Regarding the latter, the idea that we are to develop in the next chapter is that of portraying the GPLV model as the building block of a bigger class of models denoted as *Gaussian process functional factor analysis models*, GPFFA. As the name implies, what we are looking to achieve non-linearly is to retain the interpretability advantages produced in linear systems when a FA model is used against a PCA approach.

Chapter 4

Gaussian Process Functional Factor Analysis model

The previous two chapters have set out the framework for a hybrid model which will borrow ideas from both, factor analysis- and GPLV-based models. The aim in this chapter is to develop such a model. The two main properties sought are that the model must be able to handle non-linear systems and, at the same time, able to establish relationships only between those observations which are *somehow* related; we explain futher what we mean by *somehow*.

The conventional FA approach relies on variable correlation to answer the question of how the original variables are linked with one another; subsequently, it builds latent variables (also known as *common factors*) which are a function only of those original variables amongst which there is a relationship. Once in the territory of non-linear systems, variable correlation loses its meaning; in other words, there does not exist a non-linear surrogate for linear correlation. One potential way to counteract that fact would be to use engineering knowledge about the process to propose a model from which the latent variables could be derived (in a similar fashion as confirmatory factor analysis). Or, alternatively, the analyst could resort to implementing automatic variable selection techniques (e.g. penalty functions). Combining those ideas with the capabilities of a GPLV model to build a nonparametric and non-linear map between the latent and observational spaces results in a model which is more

meaningful from a process monitoring perspective.

The GPLV model is defined by Equation (3.1). To put the model into perspective, its structure assumes a relationship between $\boldsymbol{x} \in \mathcal{R}^Q$ and all of the variables in $\boldsymbol{y} \in \mathcal{R}^D$ (refer to the left panel of Figure 4.1); hopefully, $Q \ll D$ and then the analysis will render a more parsimonious representation of the data. In doing so, the latent variables are able to keep *most of the information* included in $\boldsymbol{y}$. But, is there a significant cost building a model this way? In short, the answer is yes; whereas it is highly advantageous representing the observations in a lower dimensional space, the price to pay is that the latent variables will lack physical interpretation. In this regard, and leaving the model structure aside, PCA, generalised PCA [Gnanadesikan, 1977] and kernel PCA [Schölkopf et al., 1998] all share a similar principle. The latent variables in PCA are a linear combination of all of the original observations; in generalised PCA, the latent variables are still a linear combination of a *finitely-enlarged* dimensional space which hopefully is able to capture non-linearities in the system; and finally, in KPCA, the latent variables are a linear combination of a *infinitely-enlarged* dimension which is conceptually archived via kernel functions.

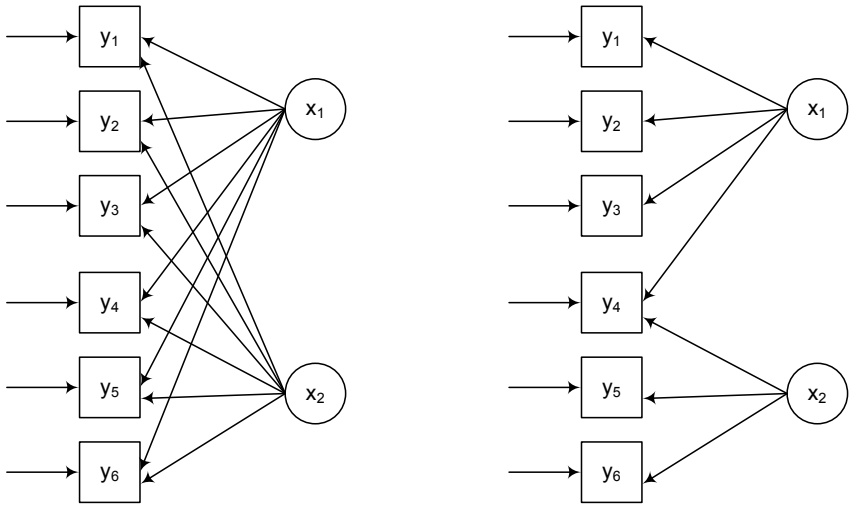

Figure 4.1: Model dependencies. Left: *GP latent variable model*. Right: *GP functional factor analysis model*.

4.1 The model

A way to strike a balance between representativeness and physical interpretability is to extend to non-linear systems the idea of factor analysis. It is desirable that the derived latent variables still keep as much as possible of the information in $\boldsymbol{y}$ while, at the same time, irrelevant relationships are removed. This is shown in the right panel of Figure 4.1 where the latent variable x_1 is the common factor associated with variables y_1, y_2, y_3 and y_4; likewise, x_2 is the common factor associated with variables y_4, y_5, y_6. Being able to remove associations (pictorially represented with an arrow) which are not significant should bring about physical interpretability to the latent constructs. That, in turn, is key in process monitoring applications.

Mathematically, we define the Gaussian Process Functional Factor Analysis model, GPFFA, as follows

$$\begin{aligned} y_{dn} = f_d(\boldsymbol{x}_n^{(d)}) + \varepsilon_{dn}; \quad \varepsilon_{dn} \sim \mathcal{N}(0, \sigma_d^2), \\ f_d(\boldsymbol{x}^{(d)})|\boldsymbol{x} \sim \mathcal{GP}_d(0, k(\boldsymbol{\theta}_d); \boldsymbol{x}^{(d)}), \\ \boldsymbol{x}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_Q) \end{aligned} \tag{4.1}$$

for $d = 1, \ldots, D$ and $n = 1, \ldots, N$. Likewise $\boldsymbol{x}_n^{(d)}$ is a subset of $\boldsymbol{x}_n$ and includes only those covariates which are associated with a given y_d; for example, with reference to the right panel of Figure 4.1, these D subsets are as follows

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(2)} = \boldsymbol{x}^{(3)} = x_1, \quad \boldsymbol{x}^{(4)} = (x_1, x_2)^\intercal, \quad \boldsymbol{x}^{(5)} = \boldsymbol{x}^{(6)} = x_2.$$

In order to account for this characteristic of the model, we introduce indicator variables. Let us define

$$\boldsymbol{i}_d = (i_{d1}, \ldots, i_{dQ})^\intercal \tag{4.2}$$

as the indicator vector for variable y_d such that $i_{dq} = 1$ iff x_q is included in $\boldsymbol{x}^{(d)}$ and $i_{dq} = 0$ otherwise. Therefore, we can write

$$\boldsymbol{x}^{(d)} = (i_{d1}x_1, \ldots, i_{dQ}x_Q)^\intercal = \boldsymbol{i}_d \odot \boldsymbol{x}.$$

With this in mind, $\mathcal{GP}_d(0, k(\boldsymbol{\theta}_d); \boldsymbol{x}^{(d)})$ represents a Gaussian process regression

model whose kernel covariance function can be written as

$$k_d(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_j^{(d)}; \boldsymbol{\theta}_d) = \upsilon_{d0} \exp\left\{ -\frac{1}{2} \sum_{q=1}^{Q} i_{dq} w_{dq} \left(x_{iq} - x_{jq}\right)^2 \right\}, \tag{4.3}$$

where $\boldsymbol{\theta}_d = (\upsilon_{d0}, w_{d1}, \ldots, w_{dq})^\intercal$ is the vector of hyper-parameters related to the d^{th} Gaussian process[1]. Note that the inverse of the weight parameters, $(w_{d1}, \ldots, w_{dq})^{-1}$ are the *characteristic length-scales* of the *squared exponential* covariance function as defined by Rasmussen and Williams [2006, p.106]; finally, $i, j = 1, \ldots, N$.

The main features of the above model are:

(a) In a similar fashion to defining the structure of the loading matrix $\boldsymbol{\Lambda}$ in the linear FA model, Equation (2.3), we can define the relationship between latent and original observations in the GPFFA model. By using Equation (4.1) in conjunction with Equation (4.3), each output variable y_d can be linked with specific latent variables x_q. Or, leaving aside causality, each latent variable will be a *non-linear combination* of a subset of the observed variables.

(b) The model offers similar advantages to that of the linear factor model in terms of interpretability, as each latent variable is a combination of a subset of the original variables. Similarly, the model will offer improvements in fault detection and diagnosis as the confounding effect of unrelated variables is removed. More generally, the model can also be thought as being a part of a structural equation model [Bollen, 1989] and hence, extended in a similar manner.

(c) It is a nonparametric model that can model complex non-linear relationships via a $\mathcal{GP}$ prior with a relatively small number of parameters in comparison to other existent nonparametric models.

(d) The observed variables y_d can be thought of as functional or longitudinal, $y_d(t)$, data and the observations at every time point t_i, $y_{di} = y_d(t_i)$ for $i = 1, \ldots, N$ could be dependent. The independence assumption for different observations is essential in the conventional factor analysis model, Equation (2.2.1); in this respect, the GPFFA model clearly differs from the FA approach.

[1] Generally the vector $\boldsymbol{x}^{(d)}$ will no longer be $Q-$dimensional; we represent this by writing w_{dq} instead of w_{dQ}.

(e) In the particular case where $\boldsymbol{x}_n^{(1)} = \boldsymbol{x}_n^{(2)} = \ldots = \boldsymbol{x}_n^{(D)} = \boldsymbol{x}_n$ in Equation (4.3) the GPFFA model simplifies to the general GPLVM made up of D *independent* GPRs as described in Chapter 3. Furthermore, if additionally $k_1(\cdot,\cdot;\boldsymbol{\theta}_1) = \ldots = k_D(\cdot,\cdot;\boldsymbol{\theta}_D) = k(\cdot,\cdot;\boldsymbol{\theta})$ the model will simplify to the GPLVM extensively described by Lawrence [2005] formed by D *independent and identically distributed* GPRs.

4.2 Inference

The building block of a *GPLVM* is a GPR model. In a similar way, a *GPLVM* is the building block of the *GPFFA* model, which, sitting at the top of the hierarchy, is the most general and flexible of the three models.

In a GPFFA model, for every y_d, the model parameters involved from the kernel function are $\boldsymbol{\theta}_d = (\upsilon_{d0}, w_{d1}, \ldots, w_{dq})^\intercal$. As we have done in previous chapters, for notational simplicity $\boldsymbol{\theta}_d$ may also loosely include σ_d^2, the variance of the independent errors; the context of the problem will determine whether that is the case. Additionally, the dimensionality of the latent variables associated with each y_d will vary and therefore the vectors $\boldsymbol{\theta}_d$, for each d, may not all have the same dimension (i.e. the vector of weight parameters linked to the latent variables will differ). Let us also define $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_D\}$ as the vector containing all the hyperparamenters of the D covariance functions. In terms of dimensionality, $\boldsymbol{\theta} \in \mathcal{R}^h$ where $h \leq D(Q+2)$; the equality only holds when a Gaussian process latent variable model is considered.

4.2.1 Estimation of model hyperparameters

From Equation (4.1) and the definition of a $\mathcal{GP}$ prior, we have that

$$p(\boldsymbol{y}_{(d)}|\boldsymbol{X},\boldsymbol{\theta}_d) = \int p(\boldsymbol{y}_{(d)}|\boldsymbol{f},\boldsymbol{X},\boldsymbol{\theta}_d)p(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{\theta}_d)d\boldsymbol{f}.$$

This equation is analytically tractable (Appendix A.2) and hence the marginal density of $\boldsymbol{y}_{(d)}|\boldsymbol{X}, \boldsymbol{\theta}_d$ is given by

$$\boldsymbol{y}_{(d)}|\boldsymbol{X}, \boldsymbol{\theta}_d \sim \mathcal{N}_N(\mathbf{0}, \mathbf{K}_d) \tag{4.4}$$

where

$$\mathbf{K}_d = \mathbf{K}_{d,f} + \sigma_d^2 \mathbf{I}_N \tag{4.5}$$

and $\mathbf{K}_{d,f}$ is the noise-free covariance matrix whose $(i,j)^{th}$ element can be calculated according to Equation (4.3). The assumption of the GPFFA model is that there are *D-independent* multivariate normal observations distributed according to Equation (4.4). Then, the joint marginal density for $\mathbf{Y} = (\boldsymbol{y}_{(1)}, \ldots, \boldsymbol{y}_{(D)})$ follows as

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) &= \prod_{d=1}^{D} \mathcal{N}_N(\mathbf{0}, \mathbf{K}_d) \\ &= \prod_{d=1}^{D} \left[(2\pi)^{-\frac{N}{2}} |\mathbf{K}_d|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \boldsymbol{y}_{(d)}{}^{\intercal} \mathbf{K}_d^{-1} \boldsymbol{y}_{(d)} \right) \right]. \end{aligned} \tag{4.6}$$

Hence, the associated log-likelihood can be written as

$$\ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y}) = \sum_{d=1}^{D} \ell_d = \sum_{d=1}^{D} \left[-\frac{N}{2} \log(2\pi) - \frac{1}{2} \log|\mathbf{K}_d| - \frac{1}{2} \operatorname{tr}\left(\mathbf{K}_d^{-1} \boldsymbol{y}_{(d)} \boldsymbol{y}_{(d)}{}^{\intercal} \right) \right]. \tag{4.7}$$

Note that the dimensions of the latent variables $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^{\intercal}$ are $N \times Q$ which is very large. They can, however, be integrated out of the joint density in order to obtain the marginal density for the observations as follows

$$\begin{aligned} p(\mathbf{Y}|\boldsymbol{\theta}) &= \int p(\mathbf{Y}|\boldsymbol{X}, \boldsymbol{\theta}) p(\mathbf{X}) d\mathbf{X} \\ &= \int \prod_{d=1}^{D} p(\boldsymbol{y}_{(d)}|\boldsymbol{X}, \boldsymbol{\theta}_d) \prod_{n=1}^{N} p(\boldsymbol{x}_n) d\mathbf{X}. \end{aligned} \tag{4.8}$$

The *empirical Bayes estimates* of $\boldsymbol{\theta}$ could then be found by maximizing the log-likelihood function, $\ell(\boldsymbol{\theta}|\mathbf{Y})$, related to this density. Unfortunately, the calculation of the above integral is not analytically tractable and approximation methods will have to be used; Laplace and profile log-likelihood approximations will be discussed in the next chapter.

4.2.2 Joint estimation of model hyperparameters and latent variables

An alternative to integrating out the latent variables $\mathbf{X}$ and then estimate $\boldsymbol{\theta}$ is to infer both set of parameters jointly. The joint posterior distribution of $(\mathbf{X}, \boldsymbol{\theta})$ is given by

$$p(\mathbf{X}, \boldsymbol{\theta}|\mathbf{Y}) \propto p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta})p(\mathbf{X})p(\boldsymbol{\theta}).$$

Maximum a posteriori (MAP) parameter estimates can be obtained by finding the mode of this posterior density. The log-likelihood of the model parameters is given by Equation (4.7). The prior for $\mathbf{X}$ is normal as for the model specification, Equation (4.1). The corresponding log-likelihood function of this posterior density can then be expressed as

$$\begin{aligned} \ell_{MAP}(\mathbf{X}, \boldsymbol{\theta}) &= \sum_{d=1}^{D} \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) + \sum_{n=1}^{N} \log p(\boldsymbol{x}_n) \\ &= \sum_{d=1}^{D} \left[-\frac{N}{2} \log(2\pi) - \frac{1}{2} \log|\mathbf{K}_d| - \frac{1}{2} \operatorname{tr}\left(\mathbf{K}_d^{-1} \boldsymbol{y}_{(d)} \boldsymbol{y}_{(d)}^{\intercal}\right) \right] - \frac{1}{2} \operatorname{tr}(\mathbf{X}\mathbf{X}^{\intercal}), \end{aligned} \tag{4.9}$$

where a non-informative prior has been allocated to $\boldsymbol{\theta}$. The joint maximisation of this target function with respect $\mathbf{X}$ and $\boldsymbol{\theta}$ will produce the estimates $\widehat{\mathbf{X}}$ and $\widehat{\boldsymbol{\theta}}$ sought. This joint estimate is similar to the MAP solution for the GPLVM proposed by Lawrence [2005].

When Equation (4.8) is used to find the estimates for $\boldsymbol{\theta}$, the factor scores $\mathbf{X}$ can still be calculated by using the MAP procedure. In those circumstances, the log-likelihood would be given by

$$\ell(\mathbf{X}; \mathbf{Y}) = \sum_{d=1}^{D} \log p(\mathbf{Y}|\mathbf{X}) + \sum_{n=1}^{N} \log p(\boldsymbol{x}_n), \tag{4.10}$$

which, upon maximisation, will produce an estimate for $\mathbf{X}$. Once the estimates of $(\mathbf{X}, \boldsymbol{\theta})$ are available, the unknown function values $f_d(\cdot)$ can also be estimated following the usual procedure in a $\mathcal{GP}$ regression model; refer to Section 1.3.3.

There are cases where, numerically, it is advantageous to further penalize the log-

likelihood as follows

$$\ell_{MAP}(\mathbf{X}, \boldsymbol{\theta}) = \ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y}) - \frac{1}{2}\operatorname{tr}(\mathbf{X}\mathbf{X}^{\intercal}) - \sum_{d=1}^{D}\sum_{j=1}^{h}\log\theta_{dj}. \tag{4.11}$$

If we were to minimize the negative form of the previous equation[2], it is clear that the term $\sum_{d=1}^{D}\sum_{j=1}^{h}\log\theta_{dj}$ acts as a penalty or regularizer which discourages large values of θ_{dj}. Numerically, this has been implemented by Lawrence [2004] for a GPLV model.

4.2.3 Model simplifications: grouping *iid* GPRs

The general GPFFA model given by Equation (4.1) assumes that the joint density of $\mathbf{Y}$ is generated by D independent GPRs; each GPR is parametrized by $\boldsymbol{\theta}_d$ in the kernel covariance function which, in turn, increases the dimensionality of the problem.

While possible, in general such complexity is not required; in other words, we can assume that some of the observations are generated by the same GPR (one could say that they belong to the same *group*, g) and are therefore identically distributed. In such case, the model log-likelihood simplifies slightly as

$$\ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y}) = \sum_{g=1}^{G}\left[-\frac{ND_g}{2}\log(2\pi) - \frac{D_g}{2}\log|\mathbf{K}_g| - \frac{1}{2}\operatorname{tr}\left(\mathbf{K}_g^{-1}\mathbf{Y}^{(g)}\mathbf{Y}^{(g)\intercal}\right)\right], \tag{4.12}$$

where D_g is the number of variables in group g; all variables within each group are generated by independent, identically distributed GPRs with $g = 1, ..., G$. $\mathbf{Y}^{(g)}$ represents the $N \times D_g$ matrix grouping all these variables column-wise.

Now it is easier to see the $GPLV$ model as the limiting case of the more general $GPFFA$ model when $G = 1$, i.e. $D_g = D$ and $\boldsymbol{x}_i^d = \boldsymbol{x}_i$ (i.e. all the latent variables are assumed to be linked to each one of the observations). Then, the log-likelihood

[2]The numerical advantage is related to the underspecification of the model as discussed in a subsequent section. This log-likelihood equation could be thought of as the MAP solution where each hyperparameter is given an uniform distribution i.e. $\theta_{dj} = \mathcal{U}[0, \theta_{dj}]$. Then $p(\theta_{dj}) = \frac{1}{\theta_{dj}}$ and, upon taking the natural logarithm, Equation (4.11) is produced.

simplifies as

$$\ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y}) = \left[-\frac{ND}{2} \log(2\pi) - \frac{D}{2} \log|\mathbf{K}| - \frac{1}{2} \operatorname{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^{\intercal}\right) \right],$$

which is the same as Equation (3.2).

4.3 Numerical implementation

This implementation refers to the joint estimation of the model hyperparameters and the latent variables (MAP solution) as discussed in Section 4.2.2. Details about the Laplace approximation to Equation (4.8) are given in the next chapter. Firstly, the mathematical formulation of the problem is presented; this can can be tackled either as a constrained optimisation problem or can be reformulated as an unconstrained problem.

Secondly, the algorithm followed to solve the *GPFFA* model is summarised. The algorithm makes use of a non-linear optimiser which only requires of the analytical gradients of the negative log-likelihood. These gradients will be briefly introduced in the last section and expanded in Appendix B.2.2.

4.3.1 Problem formulation

In order to find a solution to the GPFFA model, Equation (4.9) need to be maximised with respect to the model parameters. Note that this is equivalent to minimising the negative log-likelihood.

Let $\boldsymbol{x}_v = \operatorname{vec}(\mathbf{X})$ be the vector containing all the latent variables. The optimisation problem can be stated as follows:

$$(\widehat{\boldsymbol{x}}_v, \hat{\boldsymbol{\theta}}) = \arg \min_{(\boldsymbol{x}_v, \boldsymbol{\theta})} [-\ell(\boldsymbol{x}_v, \boldsymbol{\theta}; \mathbf{Y})_{MAP}] \quad \text{subject to} \quad \boldsymbol{\theta} > \mathbf{0}$$

where the constraints in the hyperparameters are imposed in order to make sure

that the kernel covariance function, Equation (4.3), generates a positive definite matrix. For numerical stability, additional constraints may also be needed in order to ensure the kernel function parameters do not become arbitrarily large[3].We are therefore dealing with a *constrained optimisation* problem. The objective function can be reformulated in terms of the hyperparameters in the logarithmic space so that the problem becomes an *unconstrained optimisation*:

$$(\widehat{\boldsymbol{x}}_v, \log \hat{\boldsymbol{\theta}}) = \arg \min_{(\boldsymbol{x}_v, \log \boldsymbol{\theta})} [-\ell(\boldsymbol{x}_v, \log \boldsymbol{\theta}; \mathbf{Y})_{MAP}] \tag{4.13}$$

which is the route followed in this thesis.

4.3.2 Optimising

The optimization of the GPFFA model requires of the first derivatives of the objective function with respect to the unknown parameters. The derivation is related to that given in Appendix B.2 for the GPLV model with two important caveats: firstly, the assumption of identically distributed GPRs has been dropped. And, secondly, the indicator variables defining the latent variable subsets need to be taken into consideration. Full details are provided in Appendix C.1.

The log-likelihood is a non-linear function of $\boldsymbol{x}_v$ and $\boldsymbol{\theta}$ and suffers from the same *non-convexity* problems associated with the GPLV model. The scaled conjugate gradient algorithm described in Appendix B.2.1 is used to find the MAP estimate of the model parameters. A high-level summary of the algorithm that is followed in subsequent sections is presented in Algorithm 1.

[3]This is related to the problem needing further identifiability constraints.

Algorithm 1: GPFFA model optimiser

1. Define model structure. Input:
   - G, number of independent GPRs.
   - D_g, number of identically distributed GPRs for each g.
   - Q, number of latent variables.
   - $\boldsymbol{i}_g$ in Equation (4.2), which define latent variable subsets.
2. Generate data following the model structure.
3. Initialise $\mathbf{X}$ and $\boldsymbol{\theta}$; let $\boldsymbol{x}_0 = \{\boldsymbol{x}_{v0}, \boldsymbol{\theta}_0\}$
   - $\mathbf{X}_0$, initialised with PCA; add random noise $\mathcal{N}(0, 0.005^2)$.
   - $\boldsymbol{\theta}_0$ is initialised randomly.
4. Optimisation step
   - Set maximum number of iterations and termination criteria.
   - Run optimizer until convergence. Terminate if
     (a) Maximum number of iterations is reached or,
     (b) Distance moved in search direction/change in function value is less than tolerance, i.e.

$$\|\boldsymbol{x}_v^{(t+1)} - \boldsymbol{x}_v^{(t)}\| + \|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\| \leq 10^{-4} \quad \text{and}$$
$$|\ell_{MAP}^{(t+1)} - \ell_{MAP}^{(t)}| \leq 10^{-4}$$

5. Repeat steps 3 and 4, each for different starting value of $\boldsymbol{x}_0$.
6. Final solution is the one with the smallest value of $-\ell_{MAP}$.

4.4 Numerical examples

Four different examples are reported. In all, the idea is first to generate the data, $\mathbf{Y}$, given the latent variables $\mathbf{X}$ and by assuming that $\mathbf{Y}$ and $\mathbf{X}$ are non-linearly related. Then, Gaussian process priors are allocated to these non-linear functions and inference is carried out using a GPFFA model. As a measure of goodness of fit, the correlation between the generating latent variables, $\mathbf{X}$, and the model estimates, $\widehat{\mathbf{X}}$, is used.

Example 1

The system under consideration has eight variables, $D = 8$, and two underlying latent variables, i.e. $Q = 2$. The relationship between the latent and original space is as shown in the path diagram in Figure 4.2

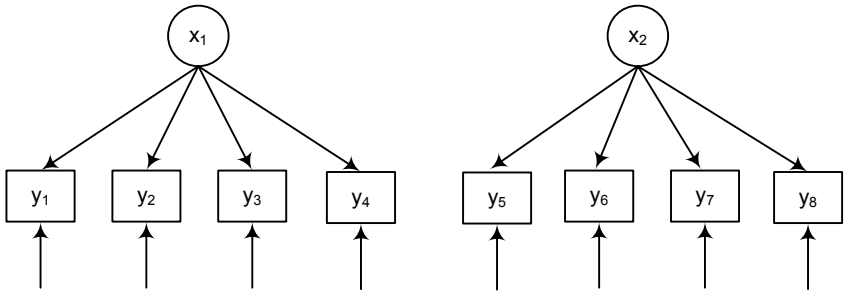

Figure 4.2: Relationship between the latent and original spaces (example 1).

The data is generated as follows:

1. The mathematical representation of the model in Figure 4.2 is as follows

$$\begin{aligned} y_d &= f_d(x_1) + e_d, \quad d = 1, \ldots, 4; \\ y_d &= f_d(x_2) + e_d, \quad d = 5, \ldots, 8 \end{aligned} \tag{4.14}$$

where f_d, for $d = 1, \ldots, 8$, are a mix of linear/non-linear functions, $x_1 \sim \mathcal{N}(0, 1)$ and $x_2 \sim \mathcal{N}(5, 9)$.

2. Generate errors are $e_d \sim \mathcal{N}(0, \sigma_d^2)$ such that $\sigma_d \sim U[0, 1]$ for $d = 1, \ldots, 8$.

3. Generate $N = 100$ observations of $y_1, \ldots, y_8$ with the following equations

$$\begin{aligned} y_1 &= x_1 + e_1 & y_5 &= \sin(x_2) + e_5 \\ y_2 &= x_1^2 + e_2 & y_6 &= \cos(x_2) + e_6 \\ y_3 &= 3 + x_1^3 + e_3 & y_7 &= 0.5x_2 + e_7 \\ y_4 &= e^{(0.7x_1)} + e_4 & y_8 &= 0.5x_2^2 + e_8 \end{aligned} \tag{4.15}$$

The aim is to model the data produced with Equation (4.15) and which have the functional structure defined by Equation (4.14); there are 8 non-linear functions which are to be approximated with two different GPR models.

Run	$-\ell_{MAP}$	$\text{corr}(x_1, \hat{x}_1)$	$\text{corr}(x_2, \hat{x}_2)$	*Iterations*
1	57.14	0.9018	0.9261	629
2	59.24	0.8944	0.9260	644
3	57.13	0.9018	0.9261	592
4	57.14	0.9019	0.9261	592
5	57.11	0.9016	0.9260	598
6	40.96	0.9008	0.9952	559

Table 4.1: Example 1: minimisation results using Equation (4.11).

Note the following:

(a) The two GPRs in this system use the same form of the covariance function, Equation (4.3), but different hyperparameters $\boldsymbol{\theta}_d$. The first four variables, $y_1, \ldots, y_4$, are mapped by four *i.i.d* GPR_1's with covariance function $k_1(\boldsymbol{\theta_1}; x_1)$. Likewise, variables $y_5, \ldots, y_8$ are mapped by four *i.i.d* GPR_2's with covariance function $k_2(\boldsymbol{\theta_2}; x_2)$.

(b) For comparison purposes, both Equation (4.9) and Equation (4.11) will be minimised; the only difference between both equations is that the latter has the

extra term $\sum_{j=1}^{h} \log \theta_j^{(d)}$. The effect of this term (or penalty) is in discouraging large values of the hyperparameters in both Gaussian processes.

The results shown in Tables (4.1) and (4.2) refer to the minimisation carried out using Equation (4.11). The algorithm was run 6 times (for different seeds) with run number 6 being the case where the latent variables are initialised with their true values instead of using PCA.

	GPR_1			GPR_2		
Run	w_1	v_0	σ^2	w_2	v_0	σ^2
1	14.79	92.65	0.27	26.31	201.05	0.25
2	15.84	94.13	0.27	26.66	218.40	0.25
3	15.06	94.48	0.27	26.47	207.01	0.25
4	14.82	93.59	0.27	26.14	205.24	0.25
5	15.55	92.18	0.26	27.04	207.16	0.25
6	17.67	99.86	0.27	22.78	228.41	0.24

Table 4.2: Example 1: MAP estimates using Equation (4.11).

The best solution achieved in this case corresponds to run number 6. In all of the cases, however, the correlation between the MAP estimates of the latent variables and the true values are very high; this is to be expected due to relative simplicity of the model structure. The estimated values of the model hyperparameters for both GPR models are shown in Table 4.2.

In order to establish a comparison, the data have also been fitted by using Equation (4.9) as the objective function. The minimisation results are displayed in Table 4.3. The first conclusion that can be drawn from it is that the algorithm takes longer to converge. This can be understood by the fact that the search space for each of the GPR hyperparameters is unrestricted in the range $(0, +\infty)$ as opposed to the minimisation carried out using Equation (4.11) where the hyperparameters have been given an uniform prior.

And secondly, the effect of removing the extra term, $\sum_{j=1}^{h} \log \theta_j^{(d)}$, from the objective function is in eliminating the constraints that were shrinking the GPR hyperparameters in the previous simulation. Generally, as shown in Table 4.4, the model

Run	$-\ell_{MAP}$	$\text{corr}(x_1, \hat{x}_1)$	$\text{corr}(x_2, \hat{x}_2)$	*Iterations*
1	41.99	0.9005	0.9264	1025
2	42.65	0.7915	0.9262	1067
3	42.07	0.9008	0.9262	833
4	42.00	0.9012	0.9263	857
5	42.11	0.9004	0.9262	1025
6	23.17	0.9001	0.9951	831

Table 4.3: Example 1: minimisation results using Equation (4.9).

hyperparameter estimates are now sensibly larger than before.

	GPR_1			GPR_2		
Run	w_1	v_0	σ^2	w_2	v_0	σ^2
1	55.19	107.11	0.27	97.80	243.31	0.25
2	103.24	88.16	0.26	91.95	225.90	0.25
3	50.00	104.69	0.27	88.85	249.91	0.25
4	54.37	105.55	0.27	97.03	242.88	0.25
5	46.16	108.37	0.27	84.53	243.60	0.25
6	66.72	96.60	0.27	83.55	280.56	0.24

Table 4.4: Example 1: MAP estimates using Equation (4.9).

As in the previous example, the best result is achieved when the latent variables are initialised with their true values (lowest log-likelihood). Again, the correlation between true latent variables and their estimates are very high. Figure 4.3 is a plot of the standardised original variables *versus* their standardised estimates for run number 1. The standardisation is necessary as the scales of the variables will differ; as the problem has been set up, not enough identifiability constraints have been imposed. This issue will be further discussed in a subsequent section.

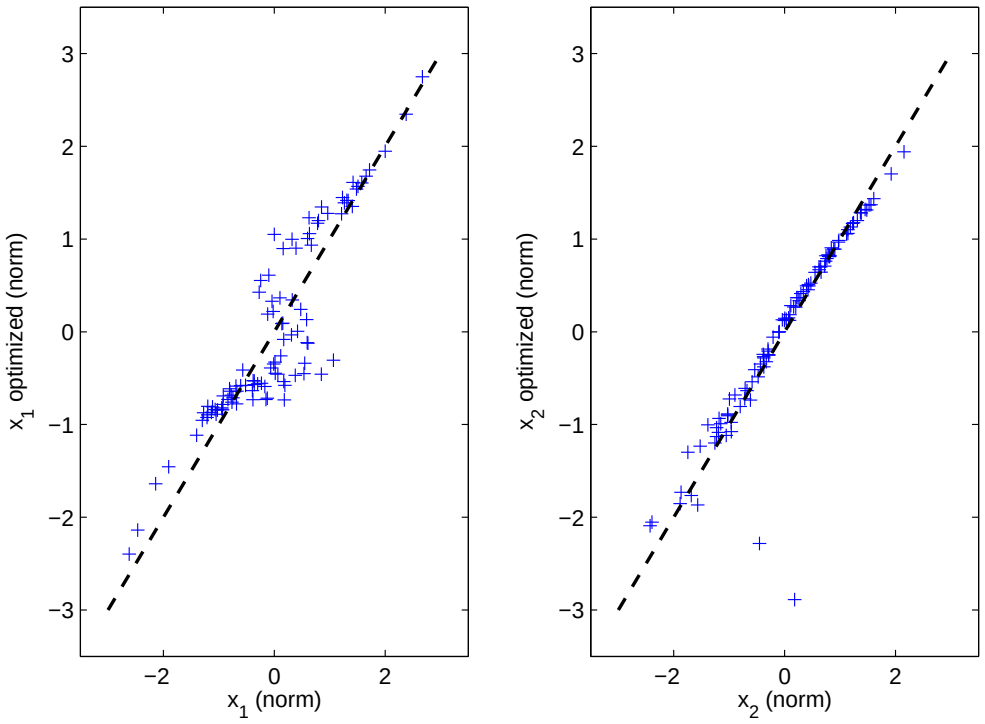

Figure 4.3: Example 1: original latent variables (standardised) versus their estimates (standardised) for *run 1* in Table 4.3. Dashed line (−−) is a 45° reference line.

Example 2

The system under consideration has eight variables, $D = 8$, and two underlying latent variables, i.e. $Q = 2$. The relationship between the latent and original space is as shown in the path diagram in Figure 4.4; in this case, the relationship between the variables is more complex than in the previous example, namely due to the dependencies of variables y_4, y_5 on both x_1 and x_2.

The data is generated according to the following steps:

1. The relationship between latent and manifest variables in the path diagram of

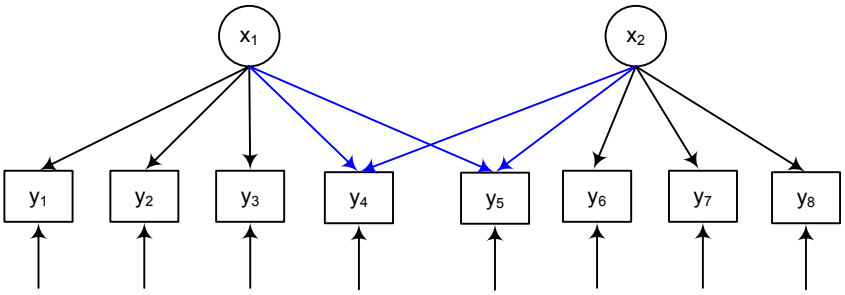

Figure 4.4: Relationship between the latent and original space (example 2).

Figure 4.4 is as follows

$$\begin{aligned} y_d &= f_d(x_1) + e_d, \quad d = 1, \ldots, 3; \\ y_d &= f_d(x_1, x_2) + e_d, \quad d = 4, 5; \\ y_d &= f_d(x_2) + e_d, \quad d = 6, \ldots, 8 \end{aligned} \tag{4.16}$$

where $x_1 \sim \mathcal{N}(0, 1)$ and $x_2 \sim \mathcal{N}(5, 9)$ and $f_d \quad (d = 1, \ldots, 8)$ are non-linear functions which will be approximated with Gaussian process priors having the following structure

$$\begin{aligned} y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_1)|x_1), \quad d = 1, \ldots, 3; \\ y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_2)|x_1, x_2), \quad d = 4, 5; \\ y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_3)|x_2), \quad d = 6, \ldots, 8. \end{aligned} \tag{4.17}$$

2. Generate errors $e_d \sim \mathcal{N}(0, \sigma_d^2)$ such that $\sigma_d \sim U[0, 1]$ for $d = 1, \ldots, 8$.

3. Generate $N = 100$ observations of the data $y_1, \ldots, y_8$ such that

$$\begin{aligned} y_1 &= x_1 + e_1 & y_5 &= x_1 + \sin(x_2) + e_5 \\ y_2 &= x_1^2 + e_2 & y_6 &= \cos(x_2) + e_6 \\ y_3 &= 3 + x_1^3 + e_3 & y_7 &= 0.5x_2 + e_7 \\ y_4 &= e^{(0.7x_1)} + x_2 + e_4 & y_8 &= 0.5x_2^2 + e_8 \end{aligned} \tag{4.18}$$

The $N = 100$ observations of data are then fitted by using a GPFFA model with

the $\mathcal{GP}$ priors structure defined in Equation (4.17). The model hyperparameters are given non-informative priors and therefore Equation (4.9) is minimised.

Run	$-\ell_{MAP}$	$\text{corr}(x_1, \hat{x}_1)$	$\text{corr}(x_2, \hat{x}_2)$	*Iterations*
1	112.44	0.7267	0.7440	2284
2	62.99	0.9097	0.7510	3152
3	108.47	0.7075	0.7324	3638
4	125.14	0.7271	0.7035	2740
5	138.40	0.7719	0.7063	2953
6	43.37	0.9291	0.9911	1052

Table 4.5: Example 2: minimisation results using Equation (4.9).

	GPR_1			GPR_2				GPR_3		
Run	w_1	υ_0	σ^2	w_1	w_2	υ_0	σ^2	w_2	υ_0	σ^2
1	236.79	52.37	0.18	2.88	26.21	45.41	0.46	69.41	263.29	0.22
2	26.39	104.99	0.16	3.33	37.84	44.29	0.62	69.18	268.66	0.22
3	111.18	69.18	0.17	5.25	47.90	38.98	0.48	93.86	244.01	0.24
4	630.19	39.20	0.14	2.82	5.00	81.74	0.33	66.88	307.84	0.29
5	268.54	51.07	0.18	5.43	6.16	57.43	0.52	86.91	266.75	0.24
6	40.78	99.22	0.17	2.82	53.40	51.21	0.58	40.58	376.56	0.23

Table 4.6: Example 2: MAP estimates using Equation (4.9).

The results are shown in Table 4.5. Note the following

(a) The algorithm was run 6 times with the last run corresponding to the case where the latent variables were initialised with their true values; in that case, the algorithm converges quicker and achieves the higher correlations as it would intuitively be expected.

(b) In general, when comparing these results with those in Table 4.3, the correlations in the current example are lower. The functional relationships are the same for variables y_1, y_2, y_3 which provide information to determine x_1; likewise y_6, y_7, y_8 provide information about the parameter x_2. However, y_4, y_5 provide information both about x_1 and x_2 at the same time; this can be interpreted as not being as informative as having both y_4 and y_5 provide information individually about the latent variables.

(c) The convergence times (see number of *iterations*) are also higher. This is due to both, having additional model parameters as well as having assumed a more intricate model.

The MAP estimates of the parameters are shown in Table 4.6. This model is more complex and has four parameters more than the corresponding model in example 1.

Example 3

In this example the structure of the previous simulation is maintained while increasing the number of variables providing information about the latent constructs; the idea is that these additional variables will add extra information from which the latent variables can be learnt. Let us consider a system with ten variables, $D = 10$, and two underlying latent variables, i.e. $Q = 2$. The relationship between the latent and original space is as shown in the path diagram of Figure 4.5

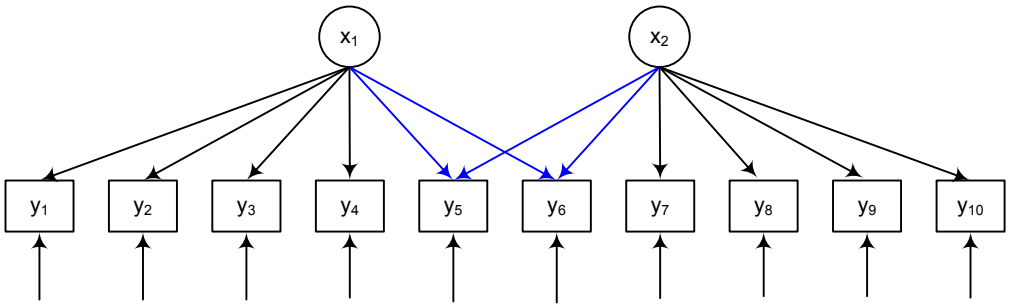

Figure 4.5: Example 3: relationship between the latent and original space.

The data is generated as follows:

(a) The relationship between latent and manifest variables in the model in Figure 4.5 is of the form:

$$\begin{aligned} y_d &= f_d(x_1) + e_d, \quad d = 1, \ldots, 4; \\ y_d &= f_d(x_1, x_2) + e_d, \quad d = 5, 6; \\ y_d &= f_d(x_2) + e_d, \quad d = 7, \ldots, 10, \end{aligned} \tag{4.19}$$

where $x_1 \sim \mathcal{N}(0,1)$ and $x_2 \sim \mathcal{N}(5,9)$ and $f_d \quad (d = 1,\ldots,10)$ are non-linear functions that we will be approximated with ten GPR models. These $\mathcal{GP}$ priors have the following structure:

$$\begin{aligned} y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_1)|x_1), \quad d = 1,\ldots,4; \\ y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_2)|x_1, x_2), \quad d = 5,6; \\ y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_3)|x_2), \quad d = 7,\ldots,10. \end{aligned} \tag{4.20}$$

(b) Errors are generated as in example 2. $N = 100$ observations of data are produced with the following equations (y_d's have been renumbered to account for the extra two variables)

$$\begin{aligned} y_1 &= x_1 + e_1 & y_6 &= x_1 + \sin(x_2) + e_6 \\ y_2 &= x_1^2 + e_2 & y_7 &= \sin(x_2) + e_7 \\ y_3 &= 3 + x_1^3 + e_3 & y_8 &= \cos(x_2) + e_8 \\ y_4 &= e^{(0.7x_1)} + e_4 & y_9 &= 0.5x_2 + e_9 \\ y_5 &= e^{(0.7x_1)} + x_2 + e_5 & y_{10} &= 0.5x_2^2 + e_{10} \end{aligned} \tag{4.21}$$

As in the previous examples, the algorithm was run 6 times; the last run corresponds to the case where the latent variables are initialised with the true values which, again, results in the best model (lowest negative log-likelihood). The results of the runs, where Equation (4.9) is minimised, are shown in Table 4.7.

Run	$-\ell_{MAP}$	$\text{corr}(x_1, \hat{x}_1)$	$\text{corr}(x_2, \hat{x}_2)$	*Iterations*
1	265.85	0.9204	0.7007	1695
2	223.27	0.9375	0.7444	2111
3	221.91	0.9357	0.7378	2338
4	241.99	0.8807	0.6720	1898
5	238.94	0.8998	0.6969	2321
6	155.55	0.9305	0.9936	1682

Table 4.7: Example 3: minimisation results using Equation (4.9).

By comparing the results in Table 4.7 with those in Table 4.5 it can be seen that correlations between the latent and the true generating variables have, generally,

increased. That is expected as there is an additional variable for each latent variable from which information can be learnt.

Point-estimates of the model hyperparameters are shown in Table 4.8.

	GPR_1			GPR_2				GPR_3		
Run	w_1	v_0	σ^2	w_1	w_2	v_0	σ^2	w_2	v_0	σ^2
1	27.54	78.81	0.28	1.67	66.62	31.43	0.79	103.65	142.18	0.34
2	22.54	116.90	0.29	1.36	122.99	32.01	0.48	80.18	148.32	0.33
3	19.64	115.37	0.29	1.17	120.11	31.13	0.48	89.53	139.39	0.33
4	31.87	90.22	0.29	3.86	86.54	27.04	0.56	128.71	132.06	0.32
5	28.50	89.54	0.29	3.25	101.36	26.97	0.54	138.78	136.09	0.32
6	40.08	95.97	0.28	2.48	60.07	39.20	0.52	59.83	287.55	0.31

Table 4.8: Example 3: MAP estimates using Equation (4.9).

Example 4

The system under consideration has twelve variables, $D = 12$, and three underlying latent variables, i.e. $Q = 3$. Note that by adding an additional latent variable to the problem the number of unknown parameters in the model increase quite dramatically by N, the sample size; additionally, further hyperparameters will be needed to model the bigger problem complexity. The relationship between the latent and original space is as shown in the path diagram in Figure 4.6.

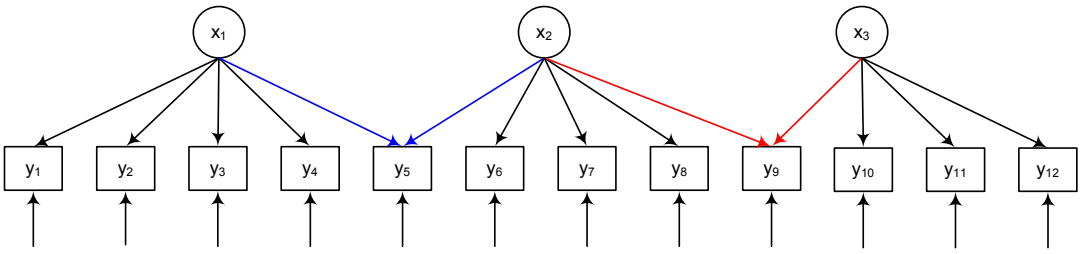

Figure 4.6: Example 4: relationship between the latent and original space.

The data is generated as follows:

(a) The relationship between latent and manifest variables in the model in Figure 4.6 of the following form

$$\begin{aligned} y_d &= f_d(x_1) + e_d, \quad d = 1, \ldots, 4; \\ y_5 &= f_5(x_1, x_2) + e_d, \\ y_d &= f_d(x_2) + e_d, \quad d = 6, \ldots, 8; \\ y_9 &= f_9(x_2, x_3) + e_d, \\ y_d &= f_d(x_3) + e_d, \quad d = 10, \ldots, 12, \end{aligned} \tag{4.22}$$

where $x_1 \sim \mathcal{N}(0, 1)$, $x_2 \sim \mathcal{N}(5, 9)$ and $x_3 \sim \mathcal{N}(2, 4)$; $f_d \quad (d = 1, \ldots, 12)$ are non-linear functions. These are to be approximated by a GPFFA model with 12 GPR models having priors defined according the following structure:

$$\begin{aligned} y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_1)|x_1), \quad d = 1, \ldots, 4; \\ y_5 &\sim \text{GPR}(0, k(\boldsymbol{\theta}_2)|x_1, x_2), \\ y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_1)|x_2), \quad d = 6, \ldots, 8; \\ y_9 &\sim \text{GPR}(0, k(\boldsymbol{\theta}_4)|x_2, x_3), \\ y_d &\overset{\text{ind}}{\sim} \text{GPR}(0, k(\boldsymbol{\theta}_5)|x_3), \quad d = 10, \ldots, 12. \end{aligned} \tag{4.23}$$

(b) Generate errors $e_d \sim \mathcal{N}(0, \sigma_d^2)$ such that $\sigma_d \sim U[0, 1]$ for $i = 1, \ldots, 12$.

(c) Generate $N = 100$ observations of the data $(y_1, \ldots, y_{12})$ with the following functional form

$$\begin{aligned} y_1 &= x_1 + e_1 & y_5 &= x_1 + \cos(x_2) + e_5 & y_9 &= \sin(x_2) + 0.5x_3 + e_9 \\ y_2 &= x_1^2 + e_2 & y_6 &= x_2^2 + e_6 & y_{10} &= \cos(x_3) + e_{10} \\ y_3 &= x_1^3 + e_3 & y_7 &= 2 + x_2 + e_7 & y_{11} &= 0.5x_3 + e_{11} \\ y_4 &= e^{0.7x_1} + e_4 & y_8 &= \cos(x_2) + e_8 & y_{12} &= x_3^2 + e_{12} \end{aligned} \tag{4.24}$$

As in the previous examples, the algorithm was run for 6 times with the last run corresponding to the case where the latent variables were initialised using the true values. The number of model parameters that need determining is $3N + h(Q + 2)$;

in this example $h = 5$ as there are 5 different $\mathcal{GP}$ priors in Equation (4.23). Full results are given in Table 4.9.

Run	$-\ell_{MAP}$	$\text{corr}(x_1, \hat{x}_1)$	$\text{corr}(x_2, \hat{x}_2)$	$\text{corr}(x_3, \hat{x}_3)$	*Iterations*
1	563.25	-0.9543	0.8147	-0.5125	2128
2	236.43	-0.8308	0.9235	-0.5054	7001
3	598.52	-0.9572	0.8053	-0.5341	7001
4	574.95	-0.9613	0.6991	-0.5236	4478
5	220.56	-0.9541	0.9508	-0.5579	7001
6	-165.95	0.9015	0.9925	0.9773	7001

Table 4.9: Example 4: minimisation results using Equation (4.9).

The best result is achieved when the algorithm is started with the true values for the latent variables, run number 6, as it could be expected. The solution seems to get trapped in local minima much more easily than in the previous cases; that, in turn, translates into more variable results. The computation burden increases also substantially as the as the joint parameter space is rather large.

4.5 Identifiability considerations

While a joint estimate of $(\mathbf{X}, \boldsymbol{\theta})$ may be found using an unconstrained optimiser when the problem is formulated in terms of the likelihood function given in Equation (4.6), the model is underspecified. Generally, the problem arises as not enough identifiability constraints to define the model parameters uniquely have been imposed. For two latent variables, the kernel covariance function is given by

$$v_0 \exp\left\{-\frac{1}{2}\left[w_1 \left(x_{i1} - x_{j1}\right)^2 + w_2 \left(x_{i2} - x_{j2}\right)^2\right]\right\} \tag{4.25}$$

where it can be seen that without *'fixing'* any of the (v_0, w_q, x_{iq}) there will many combinations leading to the same solution.

In the general context of *structural equation models*, necessary and sufficient identification rules are provided by Bollen [1989, Table 4.1]; the (linear) Factor Analysis

model can be seen as a submodel within the structural equation models architecture. Specifically, in relation to the FA submodel, underspecification problems have been widely documented in the literature. For instance, Jöreskog and Sörbom [1997, p.133] propose two different ways of dealing with this when handling the linear model $\boldsymbol{y}_i = \boldsymbol{\Lambda}\boldsymbol{x}_i + \boldsymbol{\varepsilon}_i$. In short, both solutions are as follows:

(a) *Reference variables solution.* In this case, a value of λ_{ij} is *fixed* to 1 for every column of $\boldsymbol{\Lambda}$. In turn, this places the unknown latent variables $\boldsymbol{x}_i$ in the same scale of measurement as the observations $\boldsymbol{y}_i$.

(b) *standardized solution.* Here, the scale of the latent variables is standardized by fixing the diagonal elements of the latent variables covariance matrix to one.

In the framework of GPLV models, the solution adopted by Lawrence [2005] is to give a prior to the latent variables resulting in a log-likelihood function which is comparable to the MAP solution of Equation (4.9). Adding $-\frac{1}{2}\operatorname{tr}(\mathbf{X}\mathbf{X}^\intercal)$ to the target function prevents the latent variables from becoming excessively large when optimising. Likewise, further adding a penalty (which, equivalently, can be seen as allocating a prior) to the kernel hyperparameters discourages solutions where the estimated values of $\boldsymbol{\theta}$ are very large. This is the effect achieved using Equation (4.11) and can be seen by comparing the results in Table 4.2 with those in Table 4.4. A similar approach has been taken in this chapter to deal with this identifiability issue.

4.6 Posterior consistency

The majority of the discussion offered in this section is not intended to present rigorous demonstrations; the exception is the first case, where the asymptotic properties of $\boldsymbol{\theta}$ are discussed. The main purpose is to provide an indication as to where and how the model unknowns obtain information from the observations when both, the sample size, N, and the dimensionality of the problem, D, increase.

Ideally, as more data become available, what we are expecting is for the posterior distribution of the model unknowns to concentrate around their true distribution; in broad terms, this is what is meant by posterior consistency.

4.6.1 Consistency of model hyperparameters

$\boldsymbol{\theta}$ contains all the GPFFA model hyperparameters and the corresponding functional noise; that is $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta_d}, \ldots, \boldsymbol{\theta_D}, \sigma_1^2, \ldots, \sigma_D^2\}$ where $\boldsymbol{\theta}_d$ is defined in Equation (4.3) and σ_d^2 is given by Equation (4.1). The marginal density for each variable y_d is given by Equation (4.4) which states that $\boldsymbol{y}_{(d)} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_d)$, with $\mathbf{K}_d$ given in Equation (4.5). Likewise, the joint marginal likelihood for all the observations, $\mathbf{Y}$, once all the latent variables have been integrated out, is that given in Equation (4.8). The MLE estimates of $\boldsymbol{\theta}$ are based on maximizing this likelihood.

Posterior consistency for $\boldsymbol{\theta}$ is achieved based both in Equations (4.4) and (4.8) when N is sufficiently large. Detailed proofs are given in Appendix D.

4.6.2 Consistency of X

The information for each x_{qn} is mainly provided by those y_{dn} for $d = 1, \ldots, D$ which are associated with x_{qn}. For example, in relation to the right panel in Figure 4.1, x_{1n} is associated with y_{1n} to y_{4n} and x_{2n} is associated with y_{4n} to y_{6n}. Additionally, the observations in the neighbourhood of the n^{th} observation may also provide some information due to the dependency of observations.

Let us only consider the special case where $\boldsymbol{\theta}$ is given and $\boldsymbol{x}_n = (x_{1n}, \ldots, x_{Qn})$ is estimated merely[4] from $\boldsymbol{y}_n = (y_{1n}, \ldots, y_{Dn})$ by maximizing the following likelihood

$$l(\boldsymbol{x}_n) = \sum_{d=1}^{D} [\log p(y_{dn}|\boldsymbol{x}_n)] + \log p(\boldsymbol{x}_n), \tag{4.26}$$

where $p(\boldsymbol{x}_n) \sim \mathcal{N}_Q(\mathbf{0}, \boldsymbol{I}_Q)$ and $p(y_{dn}|\boldsymbol{x}_n)$ is a univariate normal distribution derived from Equation (4.4) (i.e. its n^{th} element). The first part of Equation (4.26) is the log-likelihood of the data whereas the second term is related to the prior of the latent variables. The asymptotic properties of $\hat{\boldsymbol{x}}_n$ do not depend on the prior; that is, as D increases the role of the log-likelihood becomes more dominant over the role

[4]This will provide a conservative result; as it has been mentioned, x_{qn} could get additional information for from observations y_{dn^*} where n^* are observations in the neighbourhood of n.

of the prior.

When D is sufficiently large, $(y_{1n}, \ldots, y_{dn}, \ldots, y_{Dn})|\boldsymbol{x}_n$ for $d = 1, \ldots, D$ are independent observations (although not identically distributed). General regularity conditions from independent observations (see, for example, Lehmann and Casella [1998]) could then be applied in order to study the consistency of $\hat{\boldsymbol{x}}_n$ given $\boldsymbol{\theta}$. Asymptotically, the variance of $\hat{\boldsymbol{x}}_n$ will be given by the inverse of the second derivatives, that is

$$\text{Var}(\hat{\boldsymbol{x}}_n) = -\left(\frac{\partial^2 l(\boldsymbol{x}_n)}{\partial \boldsymbol{x}_n \boldsymbol{x}_n^\intercal}\right)^{-1} = -\left(\sum_{d=1}^{D} \frac{\partial^2 l_d(\boldsymbol{x}_n)}{\partial \boldsymbol{x}_n \boldsymbol{x}_n^\intercal}\right)^{-1},$$

As $D \to \infty$ then $\text{Var}(\hat{\boldsymbol{x}}_n) \to 0$. In a practical problem, D is limited and therefore the accuracy of the estimate of $\hat{\boldsymbol{x}}_n$ will be approximately determined by the value of this second order derivative.

In a more general case, given $\boldsymbol{\theta}$, $\mathbf{X}$ will be estimated by maximizing Equation (4.10). As mentioned previously, the accuracy of the estimates will be given by the inverse of the second derivative of Equation (4.10) with respect to $\mathbf{X}$. These derivatives are provided in Appendix C.2.

4.6.3 Consistency of the regression function given X

What does it mean that the GPFFA model leads to consistent estimates of the regression function, f_d? Loosely answered, the concept relates to how f_d updates as the sample size increases; if its posterior distribution is consistent then the regression function will concentrate around its true value, $f_{d,0}$.

Shi and Choi [2011] discuss this problem widely in the context of the GPR model when the value of the covariates, $\boldsymbol{x}_n$, is fixed and known. The authors' **Theorem 2.1** provides a proof that *almost sure* consistency can be achieved for the true regression function when Q, the dimensionality of the latent variables, is 1; the consistency achieved for unidimensional covariates can also be achieved for multidimensional cases but further considerations need to be made in order to deal with the dimensionality problem: (i) either bigger sample sizes are needed (which is not desirable from a practical point of view) or (ii) stronger assumptions for the regression

function need to be made; for further details see also Choi and Schervish [2007].

In the case of the GPFFA model, a similar proof can be develop by imposing the additional assumption that $\hat{\boldsymbol{x}}_n$ be good estimator of $\boldsymbol{x}_n$. Then, given $\hat{\boldsymbol{x}}_n$

$$f_d(\boldsymbol{x}_n^{(d)})|\boldsymbol{x}_n \sim \mathrm{GPR}_d(0, k(\boldsymbol{\theta}_d); \boldsymbol{x}_n^{(d)}),$$

and hence, a similar outcome in terms of consistency can be achieved for the GPPFA model when Shi and Choi's **Theorem 2.1** is applied to f_d for $d = 1, \ldots, D$.

4.6.4 General consistency theory

In general terms, the problem needs to consider the consistency of $f_d(\cdot)$, $\boldsymbol{\theta}$ and $\boldsymbol{X}$ simultaneously. Whereas this is an interesting problem it is considerably more demanding and will require further development.

4.7 Chapter summary

Current non-linear models dealing with latent variables tend to focus primarily on prediction while sidestepping model interpretability; while this may be appropriate in those applications where the latent embedding of the data is not of interest, in applications of process control it is of particular importance. With physical interpretability in mind, in this chapter we have introduced and defined a new class of nonparametric models, the Gaussian process functional factor analysis model. Its main characteristic is that it allows maps to be built between subsets of the latent variables and the dependent observations. We have further proposed a method of estimation for the unknown parameters and also discussed the model asymptotic properties.

The next natural step is towards model selection. In relation to the right panel of Figure 4.1, model selection is related to establishing the links (represented pictorially by arrows) between latent variables and what is observed. In (linear) factor analysis

a model is normally hypothesized based on theoretical knowledge; then it is all left to the data to further support (or not) the initial theory. In an engineering setting, while that approach is still possible, it is generally harder to pursue and a different methodology will be needed. These and other aspects will be discussed further in the next chapter.

Chapter 5

Model Selection

In so far as model selection is concerned, there are two main questions that need to be taken into account:

(i) How many different Gaussian process priors are needed to model the data appropriately? This is related to the way the output dimensions are grouped together as briefly discussed in Section 4.2.3. One potential way of doing this is to use any knowledge that we may have about the system. For instance, if we had temperatures in a distillation tower or other related equipment, there are explicit relationships amongst them all arising from physical/chemical laws and therefore they should all probably be modelled together.

(ii) Once a decision has been made as to how the observations should be grouped, the second question we face is related to the way the latent variables and their indicators $y_{(d)}$ are linked together.

This chapter assumes that a decision has been made about (i). Then, an automated way of letting the data decide about (ii) is sought.

The parameter vector $\boldsymbol{\theta}$ is key to any proposal for model selection. In this respect, two approaches are considered. Firstly, a *profile log likelihood* can be written by considering that $\boldsymbol{\theta}$ is the vector of *parameters of interest* and $\mathbf{X}$ is a matrix of *nuisance parameters*. And, secondly, the latent variables $\mathbf{X}$ are to be integrated out of the joint density of $\mathbf{Y}$ and $\mathbf{X}$. Under any of these two scenarios, the resulting

profile/marginal density will allow the associated likelihood function to be written as a function of the kernel hyperparameters only, $\ell(\boldsymbol{\theta})$. If that is feasible, $\boldsymbol{\theta}$ can then be penalized using an appropriate penalty function and both, variable selection and model parameter estimation could be carried out simultaneously; in this regard, the theory developed by Yi et al. [2011] for penalized Gaussian processes can be extended and adapted for the problem at hand.

Before providing any more details, it is worth highlighting what it is required in order to integrate the latent variables out of the joint density. The starting point is the marginal distribution of $\boldsymbol{Y}$ given by

$$p(\boldsymbol{Y}|\boldsymbol{\theta}) = \int p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) p(\mathbf{X}|\boldsymbol{\theta}) d\boldsymbol{X} = \int p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) p(\mathbf{X}) d\boldsymbol{X}. \tag{5.1}$$

Unfortunately the $N \times Q$ dimension of $\boldsymbol{X}$ is very large and the calculation of this integral is not tractable. This problem has similarities to that arising in binary Gaussian process classification where the latent function needs to be integrated out. In that specific case several approximations have been provided in the literature; Kuss and Rasmussen [2005] review and compare the results using a *Laplace's approximation* (LA) with an *Expectation-Propagation* (EP) algorithm. Their work has been subsequently extended by Nickisch and Rasmussen [2008] who provide a very comprehensive review including additional approximations like the *Kullback-Leibler* (KL) divergence minimization and *Variational Bayes*(VB) approaches; all those results are compared against a gold standard based on a *Markov chain Monte Carlo* (MCMC) sampling procedure.

In this chapter, the following three ideas will be developed:

(a) Can a profile log likelihood approach be used to estimate the model parameters and carry out model selection?

(b) How feasible it is to use a Laplace approximation to solve the numerical integration problem posed in Equation (5.1).

(c) Can the resulting profile/marginal likelihood be penalized in order to automate the variable selection problem?

5.1 Profile log likelihood

As it has been introduced, from a model selection perspective the parameters vector $\boldsymbol{\theta}$ is of central interest whereas the matrix $\mathbf{X}$ plays a secondary role; this matrix is more like a nuisance term. In this respect, the *profile log likelihood* [Davison, 2003, chapter 4] for the GPFFA model can be expressed as

$$\ell_{\text{prof}}(\boldsymbol{\theta}) = \max_{\mathbf{X}} \ell_{MAP}(\mathbf{X}, \boldsymbol{\theta}) = g(\widehat{\mathbf{X}}_{\boldsymbol{\theta}}, \boldsymbol{\theta}), \tag{5.2}$$

where $\widehat{\mathbf{X}}_{\boldsymbol{\theta}}$ is the maximum likelihood estimate for a known $\boldsymbol{\theta}$ and $\ell_{MAP}(\mathbf{X}, \boldsymbol{\theta})$ is given by Equation (4.9), that is[1]

$$g(\mathbf{X}, \boldsymbol{\theta}) = \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) + \log p(\mathbf{X}|\boldsymbol{\theta}). \tag{5.3}$$

Let us now define $\boldsymbol{x} = \text{vec}(\mathbf{X})$ and $n^* = N \cdot Q$. The function $\ell_{\text{prof}}(\boldsymbol{\theta})$ can now be optimized w.r.t the hyperparameters, $\boldsymbol{\theta}$. In order to do that, the derivatives of $\ell_{\text{prof}}(\boldsymbol{\theta})$ are needed. On the one hand, the derivatives can be obtained numerically; this is a quick but computationally intensive process as for every hyperparameter, θ_j, a numerical optimization must be carried out in order to find the maximum of $g(\mathbf{X}, \boldsymbol{\theta})$.

Alternatively, the derivatives can be worked out analytically. The covariance matrix $\mathbf{K}$ is an explicit function of the hyperparameters but also, implicitly, $\widehat{\mathbf{X}}$ is a function of $\boldsymbol{\theta}$, as when the hyperparameters change, the optimum of $g(\mathbf{X}, \boldsymbol{\theta})$ also changes (see also Rasmussen and Williams [2006, p. 125] for a similar problem). Hence

$$\frac{\partial \ell_{\text{prof}}(\boldsymbol{\theta})}{\partial \theta_j} = \left.\frac{\partial g(\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_j}\right|_{\text{explicit}, \widehat{\mathbf{X}}_{\boldsymbol{\theta}}} + \left(\frac{\partial g(\mathbf{X}, \boldsymbol{\theta})}{\partial \boldsymbol{x}}\right)^{\intercal} \left.\frac{\partial \boldsymbol{x}}{\partial \theta_j}\right|_{\text{implicit}, \widehat{\mathbf{X}}_{\boldsymbol{\theta}}} \tag{5.4}$$

Note that $\frac{\partial g(\mathbf{X}, \boldsymbol{\theta})}{\partial \theta_j}$ is given in Appendix C.1 whereas the second term in the previous expression vanishes as $\frac{\partial g(\mathbf{X}, \boldsymbol{\theta})}{\partial \boldsymbol{x}} = \mathbf{0}$ at $\boldsymbol{x} = \hat{\boldsymbol{x}}_{\boldsymbol{\theta}}$.

Finally, by further penalizing $\ell_{\text{prof}}(\boldsymbol{\theta})$, a model selection approach could subsequently be implemented.

[1] The change from ℓ_{MAP} to g is only for notational convenience.

5.2 Laplace approximation

The idea behind the Laplace Approximation is to approximate the non-Gaussian posterior distribution with a Gaussian approximation which is tractable. Williams and Barber [1998] and Rasmussen and Williams [2006, Section 3.4] provide further details as to how the approximation works in a binary Gaussian process classification problem. In the case of the Gaussian process factor analysis model, we have

$$p(\mathbf{Y}|\mathbf{X},\boldsymbol{\theta}) = \prod_{d=1}^{D} \varphi(\boldsymbol{y}_{(d)}; \mathbf{0}, \mathbf{K}_d)$$

being $\varphi(\boldsymbol{y}_{(d)}; \mathbf{0}, \mathbf{K}_d)$ a Gaussian density. Also

$$p(\mathbf{X}) = \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{x}_i|\mathbf{0}, \mathbf{I}_Q) \propto \exp\left(-\frac{1}{2}\operatorname{tr}\mathbf{X}\mathbf{X}^\intercal\right).$$

Now, taking logarithms

$$\ell(\boldsymbol{\theta}) = \log\left(p(\boldsymbol{Y}|\boldsymbol{\theta})\right) = \log\int e^{\log\left(p(\mathbf{Y}|\mathbf{X},\boldsymbol{\theta})p(\mathbf{X}|\boldsymbol{\theta})\right)}d\boldsymbol{X} = \log\int e^{\log g(\mathbf{X},\boldsymbol{\theta})}d\boldsymbol{X},$$

The Laplace approximation requires the first two derivatives of $g(\mathbf{X},\boldsymbol{\theta})$ with respect to $\boldsymbol{x}$. The first derivative is given as follows:

$$\nabla g(\boldsymbol{x})_{n^*\times 1} = \frac{\partial\log\left(p(\boldsymbol{Y}|\mathbf{X},\boldsymbol{\theta})\right)}{\partial\boldsymbol{x}} + \frac{\partial\log p(\mathbf{X}|\boldsymbol{\theta})}{\partial\boldsymbol{x}} = \frac{\partial\log\left(p(\boldsymbol{Y}|\mathbf{X},\boldsymbol{\theta})\right)}{\partial\boldsymbol{x}} - \boldsymbol{x} \tag{5.5}$$

where the elements of the vector $\frac{\partial\log\left(p(\boldsymbol{Y}|\mathbf{X},\boldsymbol{\theta})\right)}{\partial\boldsymbol{x}}$ are as given by Equation (C.6).

Likewise, the $n^* \times n^*$ Hessian or matrix of second derivatives is given by

$$\begin{aligned}\nabla^2 g(\boldsymbol{x})_{n^*\times n^*} &= \frac{\partial^2\log\left(p(\boldsymbol{Y}|\mathbf{X},\boldsymbol{\theta})\right)}{\partial\boldsymbol{x}\partial\boldsymbol{x}^\intercal} + \frac{\partial^2\log p(\mathbf{X}|\boldsymbol{\theta})}{\partial\boldsymbol{x}\partial\boldsymbol{x}^\intercal} \\ &= \frac{\partial^2\log\left(p(\boldsymbol{Y}|\mathbf{X},\boldsymbol{\theta})\right)}{\partial\boldsymbol{x}\partial\boldsymbol{x}^\intercal} - \mathbf{I}_{n^*\times n^*}.\end{aligned} \tag{5.6}$$

where an element-wise calculation of $\frac{\partial^2\log\left(p(\boldsymbol{Y}|\mathbf{X},\boldsymbol{\theta})\right)}{\partial\boldsymbol{x}\partial\boldsymbol{x}^\intercal}$ can be obtained using Equation (C.9).

Therefore, by using the Laplace method the log-likelihood of the marginal distribution can be approximated as

$$\ell(\boldsymbol{\theta}) = h(\widehat{\mathbf{X}}_{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \log \int e^{g(\mathbf{X}, \boldsymbol{\theta})} d\boldsymbol{X} \approx \frac{N}{2} \log(2\pi) + g(\widehat{\mathbf{X}}_{\boldsymbol{\theta}}, \boldsymbol{\theta}) - \frac{1}{2} \log|\mathbf{H}| \tag{5.7}$$

where $\mathbf{H} = -\nabla^2 g(\boldsymbol{x})$ and $\widehat{\mathbf{X}}_{\boldsymbol{\theta}}$ is the maximizer of $g(\mathbf{X}, \boldsymbol{\theta})$ for a given $\boldsymbol{\theta}$. As $g(\mathbf{X}, \boldsymbol{\theta})$ also depends on $\boldsymbol{\theta}$, in order to compute the previous log-likelihood a two-stage algorithm is needed:

1. For a given $\boldsymbol{\theta}$, find $\widehat{\mathbf{X}}_{\boldsymbol{\theta}}$ by maximizing $g(\mathbf{X}, \boldsymbol{\theta})$, the unnormalized posterior density of the latent variables.
2. Update $\boldsymbol{\theta}$ by maximizing Equation (5.7) given $\widehat{\mathbf{X}}_{\boldsymbol{\theta}}$.

5.3 Approximation for big sample sizes

For big samples, computation of Equation (5.7) slows considerably not only due to the increased number of parameters in the model but mainly to the problem of having to compute the Hessian matrix $\mathbf{H}$ and the corresponding determinant. As discussed in Section 3.2.3 when dealing with the GPLV model, a possible solution to treat big sample sizes is to use the *active set* approach; this works by selecting a subset of the original observations containing as much information as possible in some statistical sense.

Taking into consideration the discussion in Section 4.6 about posterior consistency, however, a different approach can also be tried. For a given observation i, the latent variables $\boldsymbol{x}_i$ obtain most of the information from the associated indicators $\boldsymbol{y}_i$. Therefore, a way of speeding up the calculation would be to partition the available data set into J smaller subsets, that is

$$\{\mathbf{Y}\} = \{\{\mathbf{Y}_1\}, \ldots, \{\mathbf{Y}_j\}, \ldots, \{\mathbf{Y}_J\}\}$$

with corresponding latent variables

$$\{\mathbf{X}\} = \{\{\mathbf{X}_1\}, \ldots, \{\mathbf{X}_j\}, \ldots, \{\mathbf{X}_J\}\}.$$

If we further assume that the subsets $\{\mathbf{Y}_j\}$ are independent, then the log marginal likelihood in Equation (5.7) can be written as

$$\begin{aligned}\ell(\boldsymbol{\theta}) &= \sum_{j=1}^{J} h(\widehat{\mathbf{X}}_j, \boldsymbol{\theta}) \\ &\approx \sum_{j=1}^{J} \left\{ g(\widehat{\mathbf{X}}_j, \boldsymbol{\theta}) - \frac{1}{2} \log|\mathbf{H}|_{\widehat{\mathbf{X}}_j, \boldsymbol{\theta}} \right\}. \end{aligned} \tag{5.8}$$

This equation can be used as an alternative to Equation (5.7) to deal with big sample sizes in order to speed up the calculations.

The model parameters can also be found by using Equation (5.2) directly. As argued previously, for bigger sample sizes the profile log likelihood calculation can be rewritten as

$$\ell_{\text{prof}}(\boldsymbol{\theta}) = g(\widehat{\mathbf{X}}_{\boldsymbol{\theta}}, \boldsymbol{\theta}) \approx \sum_{j=1}^{J} \left\{ g(\widehat{\mathbf{X}}_j, \boldsymbol{\theta}) \right\}, \tag{5.9}$$

which will speed up the calculations. While the loss of information is minimal as shown in Table 5.1, it will be bigger for those observations closer to the end of the intervals which have been chosen to partition the original sample. Additionally, note that Equation (5.9) is easier to compute as, unlike Equation (5.7), it does not require the Hessian.

Finally, as proposed in the previous chapter, $\mathbf{X}$ and $\boldsymbol{\theta}$ can be estimated jointly. If the original sample was to be split, the resulting likelihood function could then be expressed as

$$\begin{aligned}\ell_{MAP}(\mathbf{X}, \boldsymbol{\theta}) &= \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) + \log p(\mathbf{X}|\boldsymbol{\theta}) \\ &\approx \sum_{j=1}^{J} \left(\log p(\mathbf{Y}_j|\mathbf{X}_j, \boldsymbol{\theta}) + \log p(\mathbf{X}_j|\boldsymbol{\theta}) \right). \end{aligned} \tag{5.10}$$

5.4 Numerical example

The purpose of this example is to compare the parameter estimates ($\boldsymbol{\theta}$ and $\mathbf{X}$) using the following 4 scenarios:

1. Joint estimation (JL) using Equation (4.9).
2. Estimation using the profile log likelihood (PL), Equation (5.2).
3. Estimation using the profile log likelihood and splitting the sample (PL/split), Equation (5.9).
4. Estimation using a Laplace approximation where the sample size has been split (LA/split), Equation (5.8).

Run	Initial solution		Method	Final correlation	
	$\text{corr}(x_1, \hat{x}_1)$	$\text{corr}(x_2, \hat{x}_2)$		$\text{corr}(x_1, \hat{x}_1)$	$\text{corr}(x_2, \hat{x}_2)$
1	0.5704	0.5141	JL	0.7267	0.7440
			PL	0.8580	0.6980
			PL/split	0.7818	0.7574
			LA/split	0.5449	0.4826
2	0.6996	0.6462	JL	0.9184	0.7738
			PL	0.9922	0.7940
			PL/split	0.8208	0.9347
			LA/split	0.5781	0.6995
3	0.5459	0.4631	JL	0.7683	0.8290
			PL	0.9243	0.9019
			PL/split	0.6624	0.8655
			LA/split	0.5858	0.4722
4	0.6510	0.5902	JL	0.7362	0.7978
			PL	0.7608	0.6850
			PL/split	0.7257	0.9793
			LA/split	0.4763	0.6492
5	0.5668	0.6072	JL	0.6842	0.9336
			PL	0.8785	0.9790
			PL/split	0.8866	0.7040
			LA/split	0.5036	0.5341

Table 5.1: Comparison of results using 4 different methods to estimate the latent variables. Method refers to: (JL) - joint estimation, (PL) - profile log likelihood, (PL/split) - profile log likelihood with split sample and (LA/split) - Laplace approximation with split sample.

In all cases, the relationship between the latent and their indicator variables is that portrayed in Figure 4.4, with data generated using Equation (4.18). The sample size is $N = 100$ with subsamples of 20 observations for those cases where the sample size has been split. The results are shown in Table 5.1. As in the previous chapter, the correlation between the true latent variables and their estimates is reported as an empirical measure of goodness of fit. Five different data samples have been generated (*runs*). The *Initial solution* refers to the correlation between the true latent variables and their initial estimates using PCA. Likewise, *final correlation* refers to the correlation between the true latent variables and their estimates, once the optimization procedure converges.

Although this example is limited, the results in Table 5.1 point towards the following findings: (1) the final correlations are generally the highest when the profile log likelihood (PL) is used to make parameter inference; (2) when the profile log likelihood is used but the sample size is partitioned (PL/split), the correlations generally decrease in relation to the PL method but are comparable to those obtained when both $\boldsymbol{\theta}$ and $\mathbf{X}$ are estimated jointly (JL); (3) the final correlations obtained using the Laplace approximation are the lowest of the four methods.

5.5 Variable selection via penalty functions

Let us assume we have D observations $\boldsymbol{y}_n = (y_{n1}, \ldots, y_{nD})^\intercal$, for $n = 1, \ldots, N$, and that each observation $\boldsymbol{y}_n$ has been generated by, at most, Q ($Q < D$) latent variables $\boldsymbol{x}_n = (x_{n1}, \ldots, x_{nQ})^\intercal$. If we have an extensive knowledge about the system under investigation, for instance via a deterministic model, we might be able to establish the theoretical relationship between the latent and the observational variables; in other words, we may be able to write $y_{dn} = f_d(\boldsymbol{x}_n^{(d)})$, where f_d is the unknown function we are trying to estimate and $\boldsymbol{x}_n^{(d)}$ is a subset of $\boldsymbol{x}_n$, i.e. $\boldsymbol{x}_n^{(d)} \subseteq \boldsymbol{x}_n$. We could then proceed to fit a GPFFA model directly as indicated in Chapter 4.

However, in most of the cases, the physical relationship will be unclear or simply unknown, and a different procedure is needed in order to establish the link between the response variables y_{dn} and the latent factors $\boldsymbol{x}_n^{(d)}$. In statistical terms, this is a

variable selection problem.

5.5.1 The base model

The starting point for the model selection problem is a general Gaussian process latent variable model made up of D-independent Gaussian processes. Implicitly, we are assuming that that the latent variable dimensionality, Q, is known although it can be considered a part of the problem. Let us rewrite it as follows

$$\begin{aligned} y_{dn} &= f_d(\boldsymbol{x}_n) + \varepsilon_{dn}, \text{ with } \varepsilon_{dn} \sim \mathcal{N}(0, \sigma_d^2) \text{ and} \\ f_d(\boldsymbol{x})|\boldsymbol{x} &\sim \mathcal{GP}_d(0, k(\boldsymbol{\theta}_d); \boldsymbol{x}). \end{aligned} \tag{5.11}$$

$\boldsymbol{\theta}_d = [w_{d1}, \ldots, w_{dQ}, \upsilon_{0d}, \sigma_d^2]$ are the hyperparameters of the covariance function. Let us also recall $k(\boldsymbol{\theta}_d)$, the covariance function of the Gaussian process prior

$$\begin{aligned} \operatorname{cov}(y_{id}, y_{jd}) &= k_d(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}_d) \\ &= \upsilon_{0d} \exp\left\{-\frac{1}{2}\sum_{q=1}^{Q} w_{dq}\left(x_{iq} - x_{jq}\right)^2\right\} + \sigma_d^2 \delta_{ij} \end{aligned} \tag{5.12}$$

The key to variable selection, according to this covariance function, are the regression coefficients w_{dq}; they determine how relevant an input is. The larger the value of w_{dq} the more relevant the corresponding x_{iq} input is in predicting y_{id}. And conversely, the smaller the value the more irrelevant the input variable will be [Neal, 1994]. Taking this to the limit, if $w_{dq} = 0$ simply indicates that x_{iq} and y_{id} are independent. This is the kind of selective relationship suitable for process monitoring; by setting to zero some of the regression coefficients, unrelated input variables are eliminated resulting in an improved model both in terms of interpretability and prediction accuracy.

The log-likelihood associated with this base model, Equation (5.11), is

$$\ell_{MAP}(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y}) = \sum_{d=1}^{D}\left[-\frac{1}{2}\log|\mathbf{K}_d| - \frac{1}{2}\operatorname{tr}\left(\mathbf{K}_d^{-1}\boldsymbol{y}_{(d)}\boldsymbol{y}_{(d)}^{\intercal}\right)\right] - \frac{1}{2}\operatorname{tr}(\mathbf{X}\mathbf{X}^{\intercal}), \tag{5.13}$$

and the total number of hyperparameters w_{dq} which are directly related to the latent

variables is $p = D \cdot Q$ (excluding σ_d^2 and v_{0d} for $d = 1, \ldots, D$). Let us now define $\mathcal{A}$ as the subset of those parameters which are different from zero in the true model, that is $\mathcal{A} = \{w_{dq} \neq 0\}$. By defining a GPFFA model, what we are assuming is that the cardinality of $\mathcal{A}$ is $|\mathcal{A}| = p_0 < p$. In other words, the true model depends only on a subset of the predictors.

5.5.2 Penalized GP latent variable model(p-GPLV)

Yi [2009] and Yi et al. [2011] have carried out extensive and successful variable selection studies with GPR models. A similar approach can be applied to the GPLV model; there is, however, an added complexity in terms of the problem dimensionality as the latent variables are unknown. The idea is to introduce a suitable penalty in the log likelihood in order to selectively remove those predictors which are irrelevant to the response variables. In general terms, the penalized log-likelihood is defined as [Fan and Li, 2001]

$$\ell_p = -\ell(\boldsymbol{\theta}; \mathbf{Y}) + N \sum_{q,d} p_\lambda(w_{dq}) \tag{5.14}$$

where $p_\lambda(w_{dq})$ is the penalty term which is allowed to depend on λ, the regularization or tuning parameter; this, in turn, controls the size of the penalty. N is the sample size and $\ell(\boldsymbol{\theta}; \mathbf{Y})$ can be the log likelihood derived from the profile/marginal densities.

In recent years, there has been an enormous amount of research activity devoted to regularization methods and, therefore, quite a large selection of penalty functions have been proposed. A summary of the most well known penalties is given in Table 5.2; namely, the Bridge penalty [Frank and Friedman, 1993], the LASSO or 'least absolute shrinkage and selection operator' [Tibshirani, 1996], the Elastic-net [Zou and Hastie, 2005], the Ridge [Hoerl and Kennard, 1970], the Adaptive LASSO [Zou, 2006] and the SCAD or Smoothly Clipped Absolute Deviation Penalty [Fan and Li, 2001].

The weight parameters associated with the latent variables are non-negative and therefore $|w_{dq}|$ in any of the penalty functions of Table 5.2 can simply be expressed as w_{dq}. The column labelled as *Singular* indicates whether the penalty function

Name	$p_\lambda(w_{dq})$	Parameters	Singular
Bridge	$\lambda\|w_{dq}\|^\gamma$	$\lambda, 0 < \gamma < 1$	Yes
LASSO	$\lambda\|w_{dq}\|$	λ_n	Yes
Elastic-net	$\lambda\left((1-\alpha)w_{dq}^2 + \alpha\|w_{dq}\|\right)$	$\lambda, 0 \leq \alpha \leq 1$	Yes
Ridge	λw_{dq}^2	λ	No
Adaptive LASSO	$\lambda\beta_{dq}\|w_{dq}\|$	λ, β_{dq}	Yes
SCAD	$\begin{cases} \lambda w_{dq} & \text{if } 0 \leq w_{dq} \leq \lambda \\ -\frac{w_{dq}^2 - 2a\lambda w_{dq} + \lambda^2}{2(a-1)} & \text{if } \lambda < w_{dq} \leq a\lambda \\ \frac{(a+1)\lambda^2}{2} & \text{if } w_{dq} > a\lambda \end{cases}$	λ, a	Yes

Table 5.2: Penalty functions

can be used as a variable selection tool. The only penalty unsuitable for variable selection is the Ridge penalty as, although it shrinks the regression coefficients, it is unable to set them to zero regardless of the value of λ.

To test this numerically, a LASSO-penalized GPLV log-likelihood will be implemented. This function, using the profile log likelihood and the lasso penalty can be written as follows

$$\ell_p = -\ell_{MAP}(\boldsymbol{\theta}, \widehat{\mathbf{X}}_{\boldsymbol{\theta}}) + N\lambda \sum_{q,d} w_{dq}, \tag{5.15}$$

with $\lambda \geq 0$ being the tuning parameter. As its value increases, the values of w_{dq} will start shrinking towards zero. As it carries on increasing it will progressively set the values of those w_{dq} unrelated to the response as zero. In the limit, when λ dominates the log-likelihood, all the weight parameters will be set to zero. The value of λ is critical and will need to be chosen adaptively. There is a further complication here in that conventional approaches (e.g. cross-validation) will not be applicable due to the latency of the covariates. The emphasis in process monitoring is in having a good representation of the latent variables. In that respect, the correlation between the true generating latent variables, $\boldsymbol{x}$, and those given by the model parameters, $\hat{\boldsymbol{x}}$, are a way of choosing λ; how that would work in practice will be shown with a numerical example.

5.6 Numerical example

Figure 5.1 shows the path diagram for the true model (in solid black arrows) that we are considering. There are four variables, $D = 4$, and two underlying latent variables, $Q = 2$.

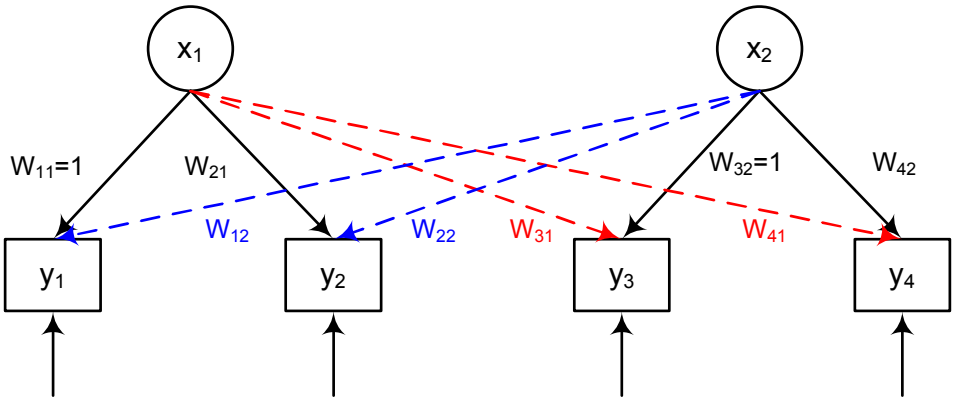

Figure 5.1: True model (solid black arrows) and inexistent functional relationships (red and blue dashed arrows).

The data, $N = 100$, has been generated with the following equations

$$\begin{aligned} y_1 &= x_1 + e_1 & y_3 &= x_2 + e_3 \\ y_2 &= e^{(0.7x_1)} + e_2 & y_4 &= 0.7x_2^2 + e_4 \end{aligned} \tag{5.16}$$

where $x_1 \sim \mathcal{N}(0, 1)$ and $x_2 \sim \mathcal{N}(3, 2)$ and the error terms are $e_d \sim \mathcal{N}(0, \sigma_d^2)$ such that $\sigma_d \sim U[0, 1]$ for $d = 1, \ldots, 4$. Note that the mathematical representation of the true model is

$$\begin{aligned} y_d &= f_d(x_1) + e_d, \quad d = 1, 2; \\ y_d &= f_d(x_2) + e_d, \quad d = 3, 4 \end{aligned} \tag{5.17}$$

Hence, in relation to Figure 5.1, both the red and blue dashed arrows are spurious relationships which we would expect to be removed by penalizing the profile log likelihood. To model the functional relationships in Equation (5.16) four different Gaussian process priors have been chosen, one for each output variable.

Parameter identifiability has a bigger impact when a penalty is imposed on the log

likelihood and some of the model parameters need to be kept fixed[2] in order for a suitable solution to be found. Numerically, a constraint optimization problem is set up by fixing the values of the parameters governing the variance of the unknown functions, υ_{0d}, as $\upsilon_{01} = 23.28$, $\upsilon_{02} = 1.27$, $\upsilon_{03} = 25.01$, $\upsilon_{04} = 40.04$ as well as setting $w_{11} = w_{32} = 1$. The values of υ_{0d} were chosen by optimizing first the true model.

λ	w_{12}	w_{21}	w_{22}	w_{31}	w_{41}	w_{42}	$\text{cor}(x_1, \hat{x}_1)$	$\text{cor}(x_2, \hat{x}_2)$
0.00	0.0000	1.6181	0.0068	0.0001	0.0291	0.7183	0.734	0.720
0.05	0.0000	0.5936	0.1761	0.0000	0.0048	0.2461	0.676	0.813
0.10	0.0000	0.3110	0.0058	0.0000	0.0410	0.1724	0.724	0.803
0.50	0.0000	0.1469	0.0000	0.0000	0.0015	0.0893	0.730	0.867
1.00	0.0000	0.0863	0.0000	0.0000	0.0006	0.0571	0.725	0.866
2.00	0.0000	0.0595	0.0000	0.0000	0.0005	0.0344	0.717	0.865
3.00	0.0000	0.0453	0.0000	0.0000	0.0004	0.0252	0.722	0.865
4.00	0.0000	0.0377	0.0000	0.0000	0.0003	0.0200	0.721	0.865

Table 5.3: Penalized profile log likelihood estimates of the weight parameters.

There are two main conclusions that can be drawn in light of the results shown in Table 5.3:

- As λ increases, all the parameters shrink as expected (apart from σ_d^2 $d = 1, \ldots, 4$ which have not been penalized - not shown.). As represented in Figure 5.1, in the true model w_{12}, w_{22}, w_{31} and w_{41} are all zero. Note how the penalty imposed on the log likelihood is successful at detecting those. It is, however, worth realising that w_{22} and w_{41} start with relatively high values and then shrink towards zero rather quickly. The situation with w_{12} and w_{31} is different in the sense that their starting values are very low; this is related both to Equation (5.16) used to generate the data and the conditions set initially to solve the constraint optimization. These latter two parameters may need to be set to zero in other problems for a feasible solution to be found.

- Pragmatically, by looking at the correlations, $\text{cor}(x_1, \hat{x}_1)$ and $\text{cor}(x_2, \hat{x}_2)$, the most suitable model would be one where λ is between 0.50 and 1.00 as those values render the bigger correlations. In both cases, as shown in Table 5.3, the estimates of w_{12}, w_{22}, w_{31} and w_{41} are shrunk to zero.

[2] This is very much related to the problem discussed in Section 4.5.

5.7 Chapter summary

Model selection is a complex problem but a very important one for a GPFFA model to be implemented successfully. Joint inference of $\mathbf{X}$ and $\boldsymbol{\theta}$, as proposed in Chapter 4, is not suitable for the implementation of a penalized model selection approach. This chapter has discussed two possible alternatives where the likelihood function is written as a function of the model hyperparameters only; namely (1) a profile log likelihood implementation and (2) a Laplace approximation. Whereas parameter estimation via the profile log likelihood produces results at least comparable to those obtained with a joint estimation, the results from a Laplace approximation are rather unsatisfactory and computationally highly demanding (to a large extent, this is related to the calculation of a Hessian matrix).

Building on the previous findings, the weight parameters ($w_{dq},\ d = 1, \ldots, D$ and $q = 1, \ldots, Q$) in the profile log likelihood can be penalized with relative ease. A LASSO penalty has been used in this thesis, but there are several others that could also be implemented (see Table 5.2). Numerical results with a relatively simple example have been produced and appear to be promising despite the latency of the model covariates. A suitable solution, however, requires of a well defined constrained optimization problem.

There are several issues that have arisen during the course of the chapter, which remain open and where further research work should be directed in the future. These will be further discussed in the following and last chapter which will also provide a final overview of the work presented in this thesis.

Chapter 6

Conclusions and further work

The main idea behind this thesis was to propose a model which can be used for fault detection in industrial systems while retaining as much physical interpretability as possible. In the area of Multivariate Statistical Process Control (MSPC), interpretability not only translates generally into a more parsimonious model but also into a model where fault diagnosis is easier to perform. Generally, the bigger the industrial process the bigger the number of variables which need to be considered from a monitoring perspective. To the limit, when all of the variables in the system are considered and monitored simultaneously and individually, there are no interpretability issues. This, however, might not be practical for two reasons; namely (1) there may be far too many variables in the process to take account of and, more importantly, (2) many of those variables will either be duplicated (correlated) or might simply be irrelevant to our purpose (nuisance variables). An early strategy to deal with this problem has been to select only those variables which are relevant to the control purpose, the Principal variables [McCabe, 1984], using statistical principles. No further issues about fault diagnosis remain as the monitoring is still carried out on individual variables. On the opposite limit, on the other side of the spectrum, latent variable models have also been developed in order to construct new variables which could summarize the variability of the process. The most remarkable cases are those models built using principal component analysis (PCA). Such has been their success, that the methodology has not only been used to model linear systems but also the more predominant non-linear processes. It is in these cases where

fault diagnosis, or interpretability, is the most difficult: each principal component is a a linear combination of every observation in the system which compounds the problem of identifying what individual variable(s) are responsible for any potential departures from expected behaviour.

Halfway between the previous two approaches, this thesis proposes and defines what we have named as the Gaussian process functional factor analysis (GPFFA) model. If factor analysis builds linear latent variables which are a combination of a subset of the variables in the system, the GPFFA model aims to achieve the same goal while capturing complex non-linear relationships. Fault diagnosis then reduces to a subset of the original variables which brings, as a result, important interpretability gains. This is an unsupervised learning problem in a high dimensional space: parameters are not only the sought latent variables, the target from a process monitoring perspective, but also the hyperparameters of the Gaussian processes that we have chosen to model the unknown functional relationships between the latent and response variables.

6.1 Summary of thesis and main contributions

A summary of what it has been covered in this thesis as well as the main contributions are better highlighted chapter by chapter:

Chapter 1 provides a review to the topic of fault detection and diagnosis in industrial systems. An outline is also provided as to what statistics are more useful for monitoring purposes. An example is shown in which a latent variable model is built using PCA; this is subsequently used in a toy problem. The chapter also illustrates how Gaussian processes are used in regression problems. Together, the review of this two areas serves as a platform motivating the rest of the thesis.

Factor analysis (FA) is a model that is heavily used in social sciences disciplines. Unlike PCA, FA constructs latent variables which are linear combinations of a *subset of the full variable space*; this is highly appealing from a process monitoring point of view. In fact, there is a close relationship between FA and PCA as shown by Tipping and Bishop [1999]. Chapter 2 proposes two different approaches whereby

this model can be used as an alternative to PCA in linear systems; namely (1) combining exploratory factor analysis (EFA) with an orthogonal rotation such as VARIMAX in order to produce a simpler structure in the loading matrix; and (2) using confirmatory factor analysis (CFA) directly. This second alternative requires more theoretical knowledge about the system under investigation which might not be the case when a decision has been made to use a data-based approach to monitor the process. An example is also provided showing how this methodology could be used in practice. FA is a linear model-based approach: an iterative optimization must be carried out in order to minimize a target function which is based on the assumed model for the data. This is important because any non-linear relationships between the variables will not be reflected in the sample covariance matrix and therefore will not be modelled correctly.

The Gaussian process latent variable (GPLV) model [Lawrence, 2005] is discussed in Chapter 3. PCA is to linear systems as the GPLV model is to non-linear processes. This thesis argues that if the model is to be used successfully to monitor industrial systems, it will require two auxiliary models: firstly, a model is needed to map the observations into the latent space; and, secondly, an additional model is then required to map the scores back from the latent space into the original space. We propose this in Serradilla et al. [2011] where examples of the methodology applied to real data are also provided.

A natural extension of the linear FA approach and the GPLV methodology leads, in Chapter 4, to the Gaussian process functional factor analysis (GPFFA) model. A full model description is provided and inference, based on a joint optimization of the model parameters, is discussed at length. Several examples, building in complexity, are also explored; given the high dimensionality of the problem[1], the model does remarkably well in uncovering the hidden latent variables in our simulations. Conditions for asymptotic posterior consistency of the model parameters are also examined in this chapter.

Finally, in Chapter 5, the focus is turned towards model selection. This topic is complex in nature; more so in the case of the GPFFA model where the latency of the input variables makes the process computationally expensive as discussed in

[1]The total number of model parameters is $D(Q+2)+NQ$ where D is the number of observed variables, Q the number of latent variables and N the sample size.

Section 5.3. Expressions are given both, for a marginal likelihood where the latent variables are integrated out using a Laplace approximation and for a profile log likelihood; both could be used as a alternative to the joint estimation of the model parameters proposed in Chapter 3. An approximation for big sample sizes is also developed. Based on the successful implementation of Yi et al. [2011] with Gaussian process regression problems, a penalized approach for the profile log likelihood is proposed as a way to carry out model selection.

6.2 Future research work

The introduction of the GPFFA model is a wholly new research area both in *Statistics* and *Machine Learning*. As such, there are several topics where further research should be warranted and where more results would help with model consolidation.

Algorithm speed is an important consideration with GPR models, where speed and efficiency are dictated by the need to invert a large sample covariance matrix. This is even more important with the GPFFA model, where several Gaussian process priors are combined together and where the input variables are latent. While several inference procedures have been proposed in this thesis, further work should include investigating other methods. For instance, in binary Gaussian process classification Nickisch and Rasmussen [2008] argue that the *Expectation-Propagation* (EP) algorithm is almost always the method of preference to determine the marginal log likelihood. It would be of interest to use the EP algorithm in the context of the GPFFA model.

Further research into the topics of *asymptotic theory* and *model identifiability* is also of interest. More specifically, this should result in a general consistency theory for the unknown regression function, $f_d(\cdot)$, $\boldsymbol{\theta}$ and $\mathbf{X}$ as well as a formal set of conditions to ensure that model parameters are identifiable under any set of circumstances.

Finally, from a *model selection* perspective, bigger and more complex simulations studies will help harness the applicability of the model not only in the field of process monitoring but also into other areas of science.

Appendix A

Mathematical miscellanea

A.1 Simulation from a multivariate normal distribution

Simulation of a full realisation of a $\mathcal{GP}$ is not possible. We can, however, sample the $\mathcal{GP}$ at a finite set of points where the function is defined. It suffices that we are able to sample from a multivariate normal distribution:

1. Set n, the number of points where the function is defined. Then

$$f(\mathbf{X}) \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma} = k(\mathbf{X}, \mathbf{X})).$$

2. Calculate the $n \times n$ covariance matrix, $\boldsymbol{\Sigma} = k(\mathbf{X}, \mathbf{X})$.

3. Compute the square root of $\boldsymbol{\Sigma}$, for instance, by computing the Cholesky decomposition [1]:

$$\mathbf{R}^\intercal \mathbf{R} = \boldsymbol{\Sigma}, \text{ where } \mathbf{R} \text{ is an upper triangular matrix}$$

[1] Or, alternatively, we could use the spectral decomposition, i.e. $\boldsymbol{\Sigma}^{\frac{1}{2}} = \mathbf{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{U}^\intercal$, where $\mathbf{U}$ is an orthogonal matrix with the eigenvectors of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with its leading eigenvalues. However, the Cholesky factor should be used when possible as it is numerically very stable and faster than alternative methods [Press et al., 2007, Section 2.9].

4. Then $f(\mathbf{X}) = \mathbf{R}^\intercal \boldsymbol{z} + \boldsymbol{\mu}, \quad \boldsymbol{z} \sim \mathcal{N}_n(\mathbf{0}, \mathbf{I})$.

5. This works as $f(\mathbf{X})$ is a linear combination of normal variables and

$$\begin{aligned} \mathrm{E}[f(\mathbf{X})] &= \mathrm{E}[\mathbf{R}^\intercal \boldsymbol{z} + \boldsymbol{\mu}] = \boldsymbol{\mu}, \\ \mathrm{Var}[f(\mathbf{X})] &= \mathrm{Var}[\mathbf{R}^\intercal \boldsymbol{z} + \boldsymbol{\mu}] = \mathbf{R}^\intercal \mathbf{I} \mathbf{R} = \boldsymbol{\Sigma}. \end{aligned}$$

A.2 Gaussian identities

When working out the marginal distribution involving Gaussian distributions only, the following result is useful. Given the following conditional distribution

$$p(\boldsymbol{y}|\boldsymbol{f}, \sigma^2) = \mathcal{N}(\mathbf{A}_1 \boldsymbol{f}, \mathbf{C}_1),$$

such that the prior distribution of $\boldsymbol{f}$ is

$$p(\boldsymbol{f}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \mathbf{C}_2).$$

Then, as shown by Lindley and Smith [1972], the marginal density of $\boldsymbol{y}$ is given by

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{f}, \sigma^2) p(\boldsymbol{f}|\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{f} = \mathcal{N}\left(\mathbf{0}, \mathbf{C}_1 + \mathbf{A}_1 \mathbf{C}_2 \mathbf{A}_1^\intercal\right). \tag{A.1}$$

A.3 Matrix derivatives

Suppose that $\mathbf{K}$ is a $N \times N$ matrix whose elements are a function of θ. The following derivatives are useful and are used throughout this thesis [Rasmussen and Williams, 2006, p. 202]

$$\frac{\partial}{\partial \theta} \mathbf{K}^{-1} = -\mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta} \mathbf{K}^{-1}, \tag{A.2}$$

and if $\mathbf{K}$ is a positive definite symmetric matrix (like a variance-covariance matrix) then the derivative of the log determinant is given by

$$\frac{\partial}{\partial \theta} \log|\mathbf{K}| = \mathrm{tr}\left(\mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta}\right). \tag{A.3}$$

Appendix B

Optimisation miscellanea

B.1 Optimising a Gaussian process

To re-state the result given in Equation (1.15), the marginal log-likelihood of the Gaussian process regression model is given by

$$\ell(\boldsymbol{\theta}|\mathcal{D}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{K}_y| - \frac{1}{2}\boldsymbol{y}^\intercal\mathbf{K}_y^{-1}\boldsymbol{y}, \tag{B.1}$$

where $\mathbf{K}_y = \mathbf{K} + \sigma^2\mathbf{I}$ is the covariance matrix for the noisy observations, $\boldsymbol{y}$. Using the matrix identities in Appendix A.3, the gradient of the marginal log-likelihood w.r.t. the hyperparameters are

$$\begin{aligned}\frac{\partial\ell(\boldsymbol{\theta})}{\partial\theta_j} &= -\frac{1}{2}\operatorname{tr}\left(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\right) + \frac{1}{2}\left(\boldsymbol{y}^\intercal\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\boldsymbol{y}\right) \\ &= \frac{1}{2}\operatorname{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal - \mathbf{K}_y^{-1})\frac{\partial\mathbf{K}_y}{\partial\theta_j}\right) \quad \text{where} \quad \boldsymbol{\alpha} = \mathbf{K}_y^{-1}\boldsymbol{y}.\end{aligned} \tag{B.2}$$

The maximum a posteriori (MAP) of the parameters, $\hat{\boldsymbol{\theta}}$, can be calculated as

$$\hat{\boldsymbol{\theta}} : \arg\min_{\boldsymbol{\theta}} -\ell(\boldsymbol{\theta}|\boldsymbol{\mathcal{D}}). \tag{B.3}$$

Optimisation can be carried out by evaluating Equations (B.2), (B.1) and (B.6) in conjunction with a non-linear minimiser. The main burden in computing the expressions above is dominated by the need to invert $\mathbf{K}_y$ and therefore gradient based optimisers are preferred as argued by Rasmussen and Williams [2006, Section 5.4].

B.1.1 Kernel derivatives

The central calculation in Equation (B.2) is the derivative of the kernel with respect to each hyperparameter i.e., $\frac{\partial \mathbf{K}_y}{\partial \theta_j}$. Let us now assume that the bias term b_o in the covariance function, Equation (1.9), is zero[1] and re-write it as

$$\begin{aligned} k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) &= \upsilon_o \exp\left\{-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{iq} - x_{jq})^2\right\} \\ &= \upsilon_o \exp\left\{-\frac{1}{2}\sum_{q=1}^{Q} w_q d^2_{q,jq}\right\}, \end{aligned} \tag{B.4}$$

where $d^2_{q,ij} = (x_{iq} - x_{jq})^2$ is simply the squared euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ along the q dimension. Let us also define $\mathbf{D}_q = \left(d^2_{q,ij}\right)$, i.e. the $N \times N$ matrix of squared euclidean distances only along the q dimension.

Now, every $\left(\frac{\partial \mathbf{K}_y}{\partial \theta_j}\right)$ is an $N \times N$ matrix given as follows:

$$\begin{aligned} \left(\frac{\partial \mathbf{K}_y}{\partial \upsilon_o}\right) &= \frac{1}{\upsilon_o}\mathbf{K}, \\ \left(\frac{\partial \mathbf{K}_y}{\partial w_q}\right) &= \left(-\frac{1}{2}\right)\mathbf{D}_q \odot \mathbf{K}, \\ \left(\frac{\partial \mathbf{K}_y}{\partial \sigma^2}\right) &= \mathbf{I}. \end{aligned} \tag{B.5}$$

In the previous expressions $\mathbf{I}$ is the N-dimensional identity matrix, $\odot$ represents the Hadamard (element-wise) product and $\mathbf{K}$ represents the noise-free covariance matrix.

[1] This is only done for clarity as it simplifies the final form of the gradients.

A further constraint in the GPR model is that all the elements of $\boldsymbol{\theta}$ must be positive. In order to achieve that, it is better to reparametrize and carry out the optimization in the log-space, i.e.

$$\log \theta = (\log w_1, \ldots, \log w_d, \log \upsilon_o, \log \sigma^2).$$

That can be easily achieved combining the gradients in Equation (B.5) with the chain rule, that is

$$\frac{\partial \mathbf{K}_y}{\partial \log \theta_j} = \frac{\partial \mathbf{K}_y}{\partial \theta_j} \frac{\partial \, \mathrm{e}^{(\log \theta_j)}}{\partial \log \theta_j} = \frac{\partial \mathbf{K}_y}{\partial \theta_j} \theta_j. \tag{B.6}$$

B.1.2 Second derivatives

The second partial derivative of the marginal log-likelihood, Equation (B.1), w.r.t. the hyperparameters is given as

$$\frac{\partial}{\partial \theta_i} \left[\frac{\partial \ell(\boldsymbol{\theta})}{\partial \theta_j} \right] = \frac{1}{2} \operatorname{tr} \left[\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal - \mathbf{K}^{-1} \right) \left(\frac{\partial^2 \mathbf{K}}{\partial \theta_i \theta_j} - \mathbf{A}_{ij} \right) - \boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal \mathbf{A}_{ji} \right], \tag{B.7}$$

where $\mathbf{A}_{ij} = \frac{\partial \mathbf{K}}{\partial \theta_i} \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_j}$ and $\boldsymbol{\alpha} = \mathbf{K}^{-1} \boldsymbol{y}$.

The proof is rather lengthy but it is shown below for completeness

$$
\begin{aligned}
\frac{\partial}{\partial\theta_i}\left[\frac{\partial\ell(\boldsymbol{\theta})}{\partial\theta_j}\right] &= -\frac{1}{2}\operatorname{tr}\left(\frac{\partial\mathbf{K}_y^{-1}}{\partial\theta_i}\frac{\partial\mathbf{K}_y}{\partial\theta_j}+\mathbf{K}_y^{-1}\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\right)+\frac{1}{2}\left(\boldsymbol{y}^\intercal\frac{\partial\mathbf{K}_y^{-1}}{\partial\theta_i}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\boldsymbol{y}\right)\\
&\quad+\frac{1}{2}\left(\boldsymbol{y}^\intercal\mathbf{K}_y^{-1}\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\mathbf{K}_y^{-1}\boldsymbol{y}+\boldsymbol{y}^\intercal\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\frac{\partial\mathbf{K}_y^{-1}}{\partial\theta_i}\boldsymbol{y}\right)\\
&= \frac{1}{2}\operatorname{tr}\left(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}-\mathbf{K}_y^{-1}\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\right)\\
&\quad-\frac{1}{2}\left(\boldsymbol{y}^\intercal\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\boldsymbol{y}\right)\\
&\quad+\frac{1}{2}\left(\boldsymbol{y}^\intercal\mathbf{K}_y^{-1}\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\mathbf{K}_y^{-1}\boldsymbol{y}-\boldsymbol{y}^\intercal\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\mathbf{K}_y^{-1}\boldsymbol{y}\right)\\
&= \frac{1}{2}\operatorname{tr}\left(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}-\mathbf{K}_y^{-1}\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\right)\\
&\quad-\frac{1}{2}\left(\boldsymbol{\alpha}^\intercal\frac{\partial\mathbf{K}_y}{\partial\theta_i}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}\boldsymbol{\alpha}-\boldsymbol{\alpha}^\intercal\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\boldsymbol{\alpha}+\boldsymbol{\alpha}^\intercal\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\boldsymbol{\alpha}\right)\\
&= \frac{1}{2}\operatorname{tr}\left(\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}-\mathbf{K}_y^{-1}\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\right)\\
&\quad-\frac{1}{2}\left(\boldsymbol{\alpha}^\intercal\left[\frac{\partial\mathbf{K}_y}{\partial\theta_i}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_j}-\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}+\frac{\partial\mathbf{K}_y}{\partial\theta_j}\mathbf{K}_y^{-1}\frac{\partial\mathbf{K}_y}{\partial\theta_i}\right]\boldsymbol{\alpha}\right)\\
&= \frac{1}{2}\operatorname{tr}\left(\mathbf{K}_y^{-1}\mathbf{A}_{ij}-\mathbf{K}_y^{-1}\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\right)-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal\left[\mathbf{A}_{ij}+\mathbf{A}_{ji}-\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\right]\right)\\
&= \frac{1}{2}\operatorname{tr}\left[\left(\mathbf{K}_y^{-1}-\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal\right)\left(\mathbf{A}_{ij}-\frac{\partial^2\mathbf{K}_y}{\partial\theta_i\theta_j}\right)-\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal\mathbf{A}_{ji}\right].
\end{aligned}
$$

B.2 Optimising the GPLV model

B.2.1 Learning algorithm

The log-likelihood of the GPLVM, Equation (3.4), is the product of D-independent GPR models. Model fitting or learning is carried out by maximizing that cost function with respect to $\mathbf{X}$ and $\boldsymbol{\theta}$. This optimisation will produce the *Empirical Bayes estimate* of the parameters; however, due to the log-likelihood being non-convex, the algorithm suffers from local optima. As it is common in those cases, we randomly start the algorithm at different points and select the solution with the highest likelihood.

Among the array of non-linear optimisers that can be used, *conjugate gradient methods* [Nocedal and Wright, 2006, Section 5.2] have been the suggested choice in the numerical analysis community when dealing with these specific problems. In broad terms, the *conjugate gradient method with line search (CGL)* works by iteratively computing search directions which are conjugate with respect the Hessian matrix (or an approximation thereof). Once the search direction has been found, a unidimensional line search with respect to the step size is carried out along the conjugate direction in order to determine a new approximation to the local minimum of the objective function. Note that conjugate gradient methods avoid having to calculate the Hessian matrix. Rasmussen [1996] uses the Polak and Ribière [1969] version of the CGL in the context of neural networks training and Gaussian process regression. Yi [2009] uses the same procedure in the context of penalised Gaussian processes.

In the specific area of Gaussian process latent variable models, Lawrence [2005] uses the *scaled conjugate gradient (SCG)* method proposed by Møller [1993]. This is also the optimiser used in this thesis for both the GPLV and GPFFA models. Without trying to offer a rigorous analysis of the performance of the different optimisation procedures, Møller's method presents several advantages over the CGL, namely

(a) It does not require of any user-dependent parameters which are critical for the method successful performance.

(b) It avoids the line search step altogether by computing a finite differences ap-

proximation to the Hessian matrix. It is worth noting that line searches will be costly as they will require possibly several function evaluations each of which makes use of the inverse of the kernel matrix.

(c) The positive-definiteness of the Hessian is controlled by using a *scale parameter* at each point. Both, the approximate Hessian and the scale parameter are subsequently used to compute the step size. The scale parameter is approximately inversely proportional to the step size so large scale parameters will correspond to small step sizes.

(d) I use the SCG implementation of Nabney [2002] which uses the Polak and Ribière [1969] formulae to update the search direction at every iteration.

An alternative solution to the *Empirical Bayes estimate* can be found by using a Markov Chain Monte Carlo algorithm. Full implementation details are given by Shi and Choi [2011, Section 8.2].

B.2.2 GPLV model derivatives

Training of the GPLV model requires the maximization of the log-likelihood function given by Equation (3.4). The analytical derivatives of this function with respect to the latent positions, $\mathbf{X}$, and the $\mathcal{GP}$ regression model parameters, $\boldsymbol{\theta}$, are also needed for the SCG optimizer. Note that we refer to every element of $\boldsymbol{\theta}$ as θ_j.

These gradients can be calculated using the chain rule as follows

$$\begin{aligned}\frac{\partial \ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y})}{\partial x_{iq}} &= \operatorname{tr}\left[\left(\frac{\partial \ell}{\partial \mathbf{K}_y}\right)^{\intercal}\left(\frac{\partial \mathbf{K}_y}{\partial x_{iq}}\right)\right] \\ \frac{\partial \ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y})}{\partial \theta_j} &= \operatorname{tr}\left[\left(\frac{\partial \ell}{\partial \mathbf{K}_y}\right)^{\intercal}\left(\frac{\partial \mathbf{K}_y}{\partial \theta_j}\right)\right]\end{aligned} \tag{B.8}$$

The common derivative, i.e. the $N \times N$ gradient of the log-likelihood with respect to the kernel matrix, is independent of the chosen covariance function and is given by

$$\left(\frac{\partial \ell}{\partial \mathbf{K}_y}\right) = -\frac{D}{2}\mathbf{K}_y^{-1} + \frac{1}{2}\mathbf{K}_y^{-1}\mathbf{Y}\mathbf{Y}^{\intercal}\mathbf{K}_y^{-1} \tag{B.9}$$

Hyperparameters derivatives

The kernel matrix $\mathbf{K}_y$ is a function of $\boldsymbol{\theta}$ as shown by Equation (1.11). Simulations using the GPLVM use a simplified version of the covariance function given by Equation (1.9) in which all of the variable weights are assumed to be equal, that is $w_1 = \ldots = w_D = \gamma$. With this in mind, the kernel function can be written as

$$\begin{aligned} k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) &= \upsilon_o \exp\left\{-\frac{1}{2}\sum_{q=1}^{Q}\gamma(x_{iq} - x_{jq})^2\right\} \\ &= \upsilon_o \exp\left\{-\frac{1}{2}\gamma d_{ij}^2\right\} \end{aligned} \tag{B.10}$$

where $d_{ij}^2 = \sum_{q=1}^{Q}(x_{iq} - x_{jq})^2$ is simply the squared euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Let us also define $\mathbf{D} = \left(d_{ij}^2\right)$, i.e. the $N \times N$ matrix of squared euclidean distances.

Every $\left(\frac{\partial \mathbf{K}_y}{\partial \theta_j}\right)$ is an $N \times N$ matrix given as follows:

$$\begin{aligned} \left(\frac{\partial \mathbf{K}_y}{\partial \upsilon_o}\right) &= \frac{1}{\nu_0}\mathbf{K} \\ \left(\frac{\partial \mathbf{K}_y}{\partial \gamma}\right) &= \left(-\frac{1}{2}\right)\mathbf{D} \odot \mathbf{K} \\ \left(\frac{\partial \mathbf{K}_y}{\partial \sigma^2}\right) &= \mathbf{I} \end{aligned} \tag{B.11}$$

where $\mathbf{I}$ is the N-dimensional identity matrix and $\odot$ represents the Hadamard (element-wise) product.

As it is the case with GPR models, a further constraint in the GPLV model is that all the elements of $\boldsymbol{\theta}$ must be positive and therefore optimisation is best done in the log-space, Equation (B.6).

Latent positions derivatives

Finally $\left(\frac{\partial \mathbf{K}_y}{\partial x_{iq}}\right)$ is a $N \times N$ symmetric matrix of all zeros but the i^{th} row/column. The elements of this row/column are given by

$$\left(\frac{\partial \mathbf{K}_y}{\partial x_{iq}}\right)_i = -\gamma \begin{pmatrix} (x_{iq} - x_{1q})k(\boldsymbol{x}_1, \boldsymbol{x}_i) \\ (x_{iq} - x_{2q})k(\boldsymbol{x}_2, \boldsymbol{x}_i) \\ \vdots \\ (x_{iq} - x_{Nq})k(\boldsymbol{x}_N, \boldsymbol{x}_i) \end{pmatrix}$$

where, notationally, the subscript i in the right hand-side of the equation is included to refer only to the elements in the i^{th} row/column of the gradient matrix.

Furthermore, there is an extra term in Equation (3.4) which is independent of the kernel matrix, $\frac{1}{2}\text{tr}(\mathbf{X}\mathbf{X}^\intercal)$. As $\left(\frac{\partial}{\partial \mathbf{X}} \text{tr}(\mathbf{X}^\intercal \mathbf{X})\right) = 2\mathbf{X}$ it finally follows that

$$\left(\frac{\partial \ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y})}{\partial \mathbf{X}}\right)_{MAP} = \left(\frac{\partial \ell(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y})}{\partial \mathbf{X}}\right) - \mathbf{X} \tag{B.12}$$

B.2.3 MAP projection gradients

The first derivatives of the log-likelihood, Equation (3.8), with respect the new latent variables can be found by applying the chain rule as

$$\frac{\partial \ell(\boldsymbol{x}_j; \boldsymbol{y}_j, \mathbf{X}, \boldsymbol{\theta})}{\partial x_{jq}} = \left[\left(\frac{\partial \ell}{\partial \boldsymbol{k}_j} \right)^\intercal \left(\frac{\partial \boldsymbol{k}_j}{\partial x_{jq}} \right) \right]. \tag{B.13}$$

Let us first re-express the log-likelihood as

$$\begin{aligned} \ell(\boldsymbol{x}_j; \boldsymbol{y}_j, \mathbf{X}, \boldsymbol{\theta}) &= -\frac{D}{2}\log(s_j^2) \\ &\quad - \frac{1}{2(s_j^2)}(\boldsymbol{y}_j - \widehat{\boldsymbol{y}}_j)^\intercal(\boldsymbol{y}_j - \widehat{\boldsymbol{y}}_j) \\ &= -\frac{D}{2}\log(s_j^2) - \frac{1}{2(s_j^2)}\boldsymbol{e}_j^\intercal \boldsymbol{e}_j. \end{aligned}$$

where $\boldsymbol{e}_j = \boldsymbol{y}_j - \widehat{\boldsymbol{y}}_j$. Therefore

$$\begin{aligned} \left(\frac{\partial \ell}{\partial \boldsymbol{k}_j} \right) &= \left(\frac{D}{2} \right) \frac{1}{(s_j^2)} \left(2\mathbf{K}_y^{-1}\boldsymbol{k}_j \right) \\ &\quad - \frac{1}{2(s_j^2)^2} \left[-2\mathbf{K}_y^{-1}\mathbf{Y}\boldsymbol{e}_j(s_j^2) - \boldsymbol{e}_j^\intercal \boldsymbol{e}_j(-2\mathbf{K}_y^{-1}\boldsymbol{k}_j) \right] \\ &= \frac{D\mathbf{K}_y^{-1}\boldsymbol{k}_j}{s_j^2} + \frac{\mathbf{K}_y^{-1}\mathbf{Y}\boldsymbol{e}_j}{s_j^2} - \frac{\boldsymbol{e}_j^\intercal \boldsymbol{e}_j \mathbf{K}_y^{-1}\boldsymbol{k}_j}{(s_j^2)^2} \end{aligned}$$

and

$$\left(\frac{\partial \ell}{\partial \boldsymbol{k}_j} \right)^\intercal = \frac{D\boldsymbol{k}_j^\intercal \mathbf{K}^{-1} + \boldsymbol{e}_j^\intercal \mathbf{Y}^\intercal \mathbf{K}^{-1}}{s_j^2} - \frac{\boldsymbol{e}_j^\intercal \boldsymbol{e}_j \boldsymbol{k}_j^\intercal \mathbf{K}^{-1}}{(s_j^2)^2}.$$

Finally $\left(\frac{\partial \boldsymbol{k}_j}{\partial x_{jq}} \right)$ is the following $N \times 1$ vector

$$\left(\frac{\partial \boldsymbol{k}_j}{\partial x_{jq}} \right) = -\gamma \begin{pmatrix} (x_{jq} - x_{1q})k(\boldsymbol{x}_1, \boldsymbol{x}_j) \\ (x_{jq} - x_{2q})k(\boldsymbol{x}_2, \boldsymbol{x}_j) \\ \vdots \\ (x_{jq} - x_{Nq})k(\boldsymbol{x}_N, \boldsymbol{x}_j) \end{pmatrix}.$$

And, as $\frac{\partial}{\partial \boldsymbol{x}_j}\left(\frac{1}{2}\boldsymbol{x}_j^\intercal \boldsymbol{x}_j \right) = \boldsymbol{x}_j$, we finally have the gradient of Equation (3.9) with

respect to $\boldsymbol{x}_j$ as

$$\left(\frac{\partial \ell(\boldsymbol{x}_j; \boldsymbol{y}_j, \mathbf{X}, \boldsymbol{\theta})}{\partial \boldsymbol{x}_j}\right)_{MAP} = \left(\frac{\partial \ell(\boldsymbol{x}_j; \boldsymbol{y}_j, \mathbf{X}, \boldsymbol{\theta})}{\partial \boldsymbol{x}_j}\right) - \boldsymbol{x}_j.$$

Appendix C

GPFFA model gradients

The covariance matrix of the *GPFFA* model is given by

$$(\mathbf{K}_d)_{ij} = \text{cov}(y_{id}, y_{jd}) = (\mathbf{K}_{d,f})_{ij} + \sigma_d^2 = k_d(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_j^{(d)}; \boldsymbol{\theta}_d) + \sigma_d^2 \tag{C.1}$$

where $k_d(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_j^{(d)}; \boldsymbol{\theta}_d)$ is given by Equation (4.3) and written as a function of the indicator variable vector, $\boldsymbol{i}_d$. While the derivatives of this kernel matrix with respect to the model hyperparameters and the latent variables are related to those developed for the *GPLVM* in Appendix B.2, there are some important differences.

C.1 GPFFA model: first derivatives

Let θ_{kj} denote any of the model hyperparameters (for $k = 1, \ldots, D$ and $j = 1, \ldots, Q + 2$). The gradients of the log-likelihood in the log-space can be calcu-

lated by applying the chain rule as follows

$$\begin{aligned}\frac{\partial}{\partial \log \theta_{kj}} \ell_{MAP}(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y}) &= \sum_{d=1}^{D} \left\{ \frac{\partial}{\partial \log \theta_{kj}} \ell_d(\mathbf{X}, \boldsymbol{\theta_d}; \boldsymbol{y}_{(d)}) \right\} \\ &= \operatorname{tr}\left[\sum_{d=1}^{D} \left\{ \left(\frac{\partial \ell_d}{\partial \mathbf{K}_d} \right)^{\intercal} \left(\frac{\partial \mathbf{K}_d}{\partial \theta_{kj}} \right) \theta_{kj} \right\} \right] \\ &= \operatorname{tr}\left[\left(\frac{\partial \ell_k}{\partial \mathbf{K}_k} \right)^{\intercal} \left(\frac{\partial \mathbf{K}_k}{\partial \theta_{kj}} \right) \theta_{kj} \right], \end{aligned} \tag{C.2}$$

as $\left(\frac{\partial \mathbf{K}_d}{\partial \theta_{kj}}\right) = 0$ for all $d \neq k$ and $\left(\frac{\partial \mathbf{K}_d}{\partial \log \theta_{dj}}\right) = \left(\frac{\partial \mathbf{K}_d}{\partial \theta_{dj}}\right) \theta_{dj}$.

Whereas the derivative of the log-likelihood with respect to the kernel matrix, which is kernel-independent, is given by

$$\begin{aligned}\left(\frac{\partial \ell_d}{\partial \mathbf{K}_d} \right) &= -\frac{1}{2} \frac{\partial}{\partial \mathbf{K}_d} \log |\mathbf{K}_d| - \frac{1}{2} \frac{\partial}{\partial \mathbf{K}_d} \operatorname{tr}(\mathbf{K}_d^{-1} \boldsymbol{y}_{(d)} \boldsymbol{y}_{(d)}^{\intercal}) \\ &= -\frac{1}{2} \left(\mathbf{K}_d^{-1} \right)^{\intercal} + \frac{1}{2} \left(\mathbf{K}_d^{-1} \boldsymbol{y}_{(d)} \boldsymbol{y}_{(d)}^{\intercal} \mathbf{K}_d^{-1} \right)^{\intercal}. \end{aligned} \tag{C.3}$$

In the case where we have G groups ($G < D$), that is some of the y_d's are assumed to be generated by independent and identically distributed GPR models, the log-likelihood will be given by Equation (4.12) and hence

$$\left(\frac{\partial \ell_g}{\partial \mathbf{K}_g} \right) = -\frac{n_g}{2} \left(\mathbf{K}_g^{-1} \right)^{\intercal} + \frac{1}{2} \left(\mathbf{K}_g^{-1} \mathbf{Y}^{(g)} \mathbf{Y}^{(g)\intercal} \mathbf{K}_g^{-1} \right)^{\intercal}. \tag{C.4}$$

In contrast with the derivatives with respect to the hyperparameters, the latent variables x_{ij} enter the log-likelihood through several $\mathbf{K}_d$ and therefore, the derivative will still be the sum of D terms, that is

$$\begin{aligned}\frac{\partial \ell_{MAP}(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y})}{\partial x_{ij}} &= \sum_{d=1}^{D} \left\{ \frac{\partial}{\partial x_{ij}} \ell_d(\mathbf{X}, \boldsymbol{\theta_d}; \boldsymbol{y}_{(d)}) \right\} \\ &= \operatorname{tr}\left[\sum_{d=1}^{D} \left\{ \left(\frac{\partial \ell_d}{\partial \mathbf{K}_d} \right)^{\intercal} \left(\frac{\partial \mathbf{K}_d}{\partial x_{ij}} \right) \right\} \right] - x_{ij}. \end{aligned} \tag{C.5}$$

Combining the previous result with Equation (C.3) leads to

$$\begin{aligned}\frac{\partial \ell_{MAP}(\mathbf{X}, \boldsymbol{\theta}; \mathbf{Y})}{\partial x_{ij}} &= \operatorname{tr}\left[\sum_{d=1}^{D}\left\{\left(\frac{\partial \ell_d}{\partial \mathbf{K}_d}\right)^{\intercal}\left(\frac{\partial \mathbf{K}_d}{\partial x_{ij}}\right)\right\}\right] - x_{ij} \\ &= \operatorname{tr}\left[\sum_{d=1}^{D}\left\{\frac{1}{2}\left(\mathbf{K}_d^{-1}\boldsymbol{y}_{(d)}\boldsymbol{y}_{(d)}^{\intercal}\mathbf{K}_d^{-1} - \mathbf{K}_d^{-1}\right)\left(\frac{\partial \mathbf{K}_d}{\partial x_{ij}}\right)\right\}\right] - x_{ij} \\ &= \sum_{d=1}^{D}\left[\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{\alpha}_d\boldsymbol{\alpha}_d^{\intercal} - \mathbf{K}_d^{-1}\right)\frac{\partial \mathbf{K}_d}{\partial x_{ij}}\right)\right] - x_{ij} \end{aligned} \tag{C.6}$$

where $\boldsymbol{\alpha}_d = \mathbf{K}_d^{-1}\boldsymbol{y}_{(d)}$. To complete the calculation, both the derivatives $\left(\frac{\partial \mathbf{K}_d}{\partial \theta_{dj}}\right)$ and $\left(\frac{\partial \mathbf{K}_d}{\partial x_{ij}}\right)$ are also needed. Their calculation is shown in the next subsections.

Hyperparameters derivatives

The hyperparameters enter the log-likelihood function, Equation (4.9), through the kernel covariance matrix as $\boldsymbol{\theta}_d = [\upsilon_{d0}, w_{d1}, \ldots, w_{dQ}, \sigma_d^2]$.

The derivatives $\left(\frac{\partial \mathbf{K}_d}{\partial \theta_{dj}}\right)$ are $N \times N$ dimensional and are as follows

$$\begin{aligned}\left(\frac{\partial \mathbf{K}_d}{\partial \upsilon_{d0}}\right) &= \frac{1}{\upsilon_{d0}}\mathbf{K}_{d,f} \\ \left(\frac{\partial \mathbf{K}_d}{\partial \sigma_d^2}\right) &= \mathbf{I}_N \\ \left(\frac{\partial \mathbf{K}_d}{\partial w_{dq}}\right) &= -\frac{1}{2}\mathbf{D}_{dq} \odot \mathbf{K}_{d,f} \end{aligned} \tag{C.7}$$

where $\odot$ represents the Hadamard (element-wise) product and $\mathbf{D}_{dq}$ is also an $N \times N$ matrix whose ij^{th} element is given by $i_{dq}(x_{iq} - x_{jq})^2$.

Latent variables derivatives

$\left(\frac{\partial \mathbf{K}_d}{\partial x_{iq}}\right)$ is an $N \times N$ sparse matrix of all zeros but the i^{th} row and column; the $(i,i)^{th}$ position is also zero, that is

$$\left(\frac{\partial \mathbf{K}_d}{\partial x_{iq}}\right) = \begin{pmatrix} & \cdots & * & & \cdots & \\ & & c_{it} & & & \\ & 0 & * & & 0 & \\ * & c_{it} & 0 & * & * & * \\ & & * & & & \\ & 0 & * & & 0 & \end{pmatrix} \tag{C.8}$$

The elements c_{it} of the i^{th} row/column are given by

$$c_{it} = \left(\frac{\partial \mathbf{K}_d(i,t)}{\partial x_{iq}}\right) = -i_{dq} w_{dq} (x_{iq} - x_{tq}) k_{d,f}(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_t^{(d)})$$

where $k_{d,f}(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_t^{(d)}) = \upsilon_0^{(d)} \exp\left\{-\frac{1}{2}\sum_{q=1}^{Q} i_{dq} w_{dq} \left(x_{iq} - x_{tq}\right)^2\right\}$ is the functional part of the kernel covariance function.

Given the special structure of $\left(\frac{\partial \mathbf{K}_d}{\partial x_{iq}}\right)$, the trace calculation in Equation (C.6) simplifies somewhat. Let us see that with a specific example how to calculate $\text{tr}(\mathbf{AB})$. The matrices $\mathbf{A}$ and $\mathbf{B}$ are symmetric and the latter is also sparse as already discussed. Then,

$$\begin{aligned} &\text{tr}\left[\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2N} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3N} \\ \vdots & & \vdots & & \vdots \\ a_{N1} & a_{N2} & a_{N3} & \cdots & a_{NN} \end{pmatrix} \begin{pmatrix} 0 & 0 & b_1 & \cdots & 0 \\ 0 & 0 & b_2 & \cdots & 0 \\ b_1 & b_2 & 0 & \cdots & b_N \\ \vdots & & \vdots & & \vdots \\ 0 & 0 & b_N & \cdots & 0 \end{pmatrix}\right] \\ &= a_{13}b_1 + a_{23}b_2 + (a_{31}b_1 + a_{32}b_2 + 0 + \ldots + a_{3N}b_N) + \ldots + a_{N3}b_N \\ &= (a_{13} + a_{31})b_1 + (a_{23} + a_{32})b_2 + 0 + \ldots + (a_{3N} + a_{N3})b_N \\ &= 2(a_{13}b_1 + a_{23}b_2 + 0 + \ldots + a_{3N}b_N) \end{aligned}$$

C.2 GPFFA model: second derivatives

The second derivatives are considerably more involved. The calculations are as follows

$$\frac{\partial^2 \ell_{MAP}}{\partial x_{jk} \partial x_{iq}} = \sum_{d=1}^{D} \left[\frac{1}{2} \operatorname{tr} \left(\left(\boldsymbol{\alpha}_d \boldsymbol{\alpha}_d^\intercal - \mathbf{K}_d^{-1} \right) \left(\frac{\partial^2 \mathbf{K}_d}{\partial x_{jk} x_{iq}} - \mathbf{A}_{jk,iq}^{(d)} \right) - \boldsymbol{\alpha}_d \boldsymbol{\alpha}_d^\intercal \mathbf{A}_{iq,jk}^{(d)} \right) \right] - \delta_{ij} \delta_{qk} \tag{C.9}$$

where $\mathbf{A}_{jk,iq}^{(d)} = \frac{\partial \mathbf{K}_d}{\partial x_{jk}} \mathbf{K}_d^{-1} \frac{\partial \mathbf{K}_d}{\partial x_{iq}}$. Note that $i, j = 1, \ldots, N$ and $q, k = 1, \ldots, Q$.

Calculation of $\operatorname{tr} \left(\frac{1}{2} \left(\boldsymbol{\alpha}_d \boldsymbol{\alpha}_d^\intercal - \mathbf{K}_d^{-1} \right) \left(\frac{\partial^2 \mathbf{K}_d}{\partial x_{jk} x_{iq}} \right) \right)$

For $i = j$ and for the cases where $q = k$ or $q \neq k$ (i.e main and minor diagonal elements), note that the matrix $\frac{\partial^2 \mathbf{K}_d}{\partial x_{jk} x_{iq}}$ has the same special structure as Equation (C.8) and therefore the calculation simplifies in the same way.

For $i \neq j$ and for the cases where $q = k$ or $q \neq k$ (i.e off-diagonal elements), note that the matrix $\frac{\partial^2 \mathbf{K}_d}{\partial x_{jk} x_{iq}}$ has the same special structure as Equation (C.8) and therefore the calculation simplifies in the same way.

Kernel matrix second derivatives

For the case of the second derivatives of the kernel there are 4 possible situations to consider

	$q = k$	$q \neq k$
$i = j$	$\left(\frac{\partial^2 \mathbf{K}_d(i,j)}{\partial {x_{iq}}^2} \right)$	$\frac{\partial}{\partial x_{ik}} \left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{iq}} \right)$
$i \neq j$	$\frac{\partial}{\partial x_{jq}} \left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{iq}} \right)$	$\frac{\partial}{\partial x_{jk}} \left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{iq}} \right)$

When $i = j$ and $q = k$ (DIAGONAL ELEMENTS)

The matrix of second derivates has the same form as Equation (C.8) but with elements given by

$$\frac{\partial}{\partial x_{iq}} c_{it} = \frac{\partial}{\partial x_{iq}} \left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{iq}} \right) = \begin{pmatrix} & \cdots & * & & \cdots & \\ & & \frac{\partial}{\partial x_{iq}} c_{it} & & & \\ & 0 & * & & 0 & \\ * & \frac{\partial}{\partial x_{iq}} c_{it} & 0 & * & * & * \\ & & * & & & \\ & 0 & * & & 0 & \end{pmatrix}$$

where the non-zero elements of the matrix are given by

$$\frac{\partial}{\partial x_{iq}} c_{it} = -i_{dq} w_{dq} k_{d,f}(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_t^{(d)}) \left(1 - i_{dq} w_{dq} (x_{iq} - x_{tq})^2\right)$$

When $i = j$ and $q \neq k$ (MINOR DIAGONAL)

As before, the matrix of second derivates has the same form as Equation (C.8) but with elements given by

$$\frac{\partial}{\partial x_{ik}} c_{it} = \frac{\partial}{\partial x_{ik}} \left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{iq}} \right) = \begin{pmatrix} & \cdots & * & & \cdots & \\ & & \frac{\partial}{\partial x_{ik}} c_{it} & & & \\ & 0 & * & & 0 & \\ * & \frac{\partial}{\partial x_{ik}} c_{it} & 0 & * & * & * \\ & & * & & & \\ & 0 & * & & 0 & \end{pmatrix}$$

where the non-zero elements of the matrix are given by

$$\frac{\partial}{\partial x_{ik}} c_{it} = \left[i_{dq} w_{dq} (x_{iq} - x_{tq}) k_{d,f}(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_t^{(d)}) \right] \left[i_{dk} w_{dk} (x_{ik} - x_{tk}) \right]$$

When $i \neq j$ and $q = k$

The matrix of second derivates only has the $(i,j)^{th}$ and $(j,i)^{th}$ positions different from zero (i.e. only the i^{th} element in both the j^{th} row/column are non-zero)

$$\frac{\partial}{\partial x_{jk}}\left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{iq}}\right) = \begin{pmatrix} & \cdots & 0 & & \cdots & \\ & & * & & & \\ & 0 & 0 & & 0 & \\ 0 & * & 0 & 0 & 0 & 0 \\ & & 0 & & & \\ & 0 & 0 & & 0 & \end{pmatrix}$$

where the non-zero elements of the matrix are given by

$$\begin{aligned}\left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{jq}\partial x_{iq}}\right) &= i_{dq}w_{dq}k_{d,f}(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_j^{(d)}) - i_{dq}^2 w_{dq}^2 (x_{iq} - x_{jq})^2 k_{d,f}(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_j^{(d)}) \\ &= i_{dq}w_{dq}k_{d,f}(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_j^{(d)})\left(1 - i_{dq}w_{dq}(x_{iq} - x_{jq})^2\right)\end{aligned}$$

Note that the difference with the case where $i = j$ is a minus sign.

When $i \neq j$ and $q \neq k$

As before, there only two elements that are non-zeros and are given by

$$\left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{jk}\partial x_{iq}}\right) = -\left[i_{dq}w_{dq}(x_{iq} - x_{jq})k_{d,f}(\boldsymbol{x}_i^{(d)}, \boldsymbol{x}_j^{(d)})\right]\left[i_{dk}w_{dk}(x_{ik} - x_{jk})\right]$$

where, again, there is a minus sign difference with regards to the case $i = j$.

Notes about computation

1. The derivatives of the kernel $\mathbf{K}_d$ with respect to the latent variables are all $N \times N$ matrices; in total there are $n^* = N \cdot Q$ of them. They all are sparse

matrices made up by only one non-zero vector (either the row or the column - they are the same). Therefore all these derivates can be stored in an $n^* \times N$ super-matrix where each row is the non-zero vector of derivatives extracted from $\left(\frac{\partial \mathbf{K}_d(i,j)}{\partial x_{iq}}\right)$.

2. The Hessian has $\frac{N(N+1)}{2}$ distinct elements and $\frac{N(N-1)}{2}$ non-distinct elements. The distinct elements can be stored in an upper (-lower) triangular matrix, $\mathbf{L}$, and the symmetric Hessian can then be returned as $\mathbf{B} = \mathbf{L} + \mathbf{L}^\intercal - \text{diag}(\mathbf{L})$.

3. When computing the trace, there is a considerable difference in speed if

   (a) $\text{tr}(\mathbf{A} \cdot \mathbf{B})$ - not very efficient calculation.

   (b) $\text{sum}(\text{sum}(\mathbf{A}^\intercal \cdot \mathbf{B}), 2)$ - efficient.

   (c) $\text{vec}(\mathbf{A}^\intercal)^\intercal \cdot \text{vec}(\mathbf{B})$ - efficient.

Appendix D

Asymptotic results

The observations $(y_{d1}, \ldots, y_{dN})$ of the GPFFA model are neither independent nor identically distributed. Regularity conditions for cases covering independent observations (see, e.g., Lehmann and Casella [1998]) are therefore not applicable here.

D.1 Posterior consistency

Firstly, regularity conditions and the resulting consistency theorems of the maximum likelihood estimator when the likelihood is based on dependent observations, as formulated in Basawa and Prakasa Rao [1980], are discussed. This derivation also builds on that given by Shi and Choi [2011, App. (A.6)].

Let $\boldsymbol{Y}^n = (Y_1, \ldots, Y_n)$, $n \geq 1$ be a sequence of random samples with density $p(\boldsymbol{y}^n; \theta) = p(y_1, \ldots, y_n; \theta)$. Also let θ_0 be the true value of θ. Let us define the conditional density

$$p_k(\theta) = p(\boldsymbol{y}^k; \theta)/p(\boldsymbol{y}^{k-1}; \theta)$$

for every $k \geq 1$. Assume that the function $p_k(\theta)$ is twice differentiable with respect to θ for all θ in a neighbourhood I of θ_0 and all $\boldsymbol{y}^k$. Further assume that the support of $p(\boldsymbol{y}^n; \theta)$ is independent of $\theta \in I$. Define $\phi_k(\theta) = \log p_k(\theta)$ and let $\dot{\phi}_k(\theta)$ be the $p \times 1$ vector whose ith component is $\dot{\phi}_{k,i} = \frac{\partial}{\partial \theta_i}\phi_k(\theta)$ and $\ddot{\phi}_k(\theta)$ be the $p \times p$ matrix whose

$(ij)^{th}$ component is $\ddot{\phi}_{k,i,j} = \frac{\partial^2}{\partial\theta_i \partial\theta_j}\phi_k(\theta)$. For simplicity, we formulate the regularity conditions for the one-dimensional case. Denote

$$U_k(\theta) = \dot{\phi}_k(\theta),\ V_k(\theta) = \ddot{\phi}_k,\ U_k = U_k(\theta_0),\ V_k = V_k(\theta_0).$$

Let $L_n(\theta) = \log p(\boldsymbol{y}^n; \theta)$. Let $\mathcal{F}_n$ be the σ-field generated by Y_j, $1 \leq j \leq n$ and $\mathcal{F}_0$ be the trivial σ-field. Assume that the following conditions are satisfied:

(C1) $\phi_k(\theta)$ is thrice differentiable with respect to θ for all $\theta \in I$. Let $W_k(\theta) = \dddot{\phi}_k(\theta)$ be the third derivative of $\phi_k(\theta)$ with respect to θ.

(C2) Double differentiation of $p(\boldsymbol{y}^n; \theta)$ with respect to θ under the integral sign is permitted for $\theta \in I$ in $\int p(\boldsymbol{y}^n; \theta) d\mu^n(\boldsymbol{y}^n)$.

(C3) $\mathrm{E}|V_k| < \infty$, $\mathrm{E}|Z_k| < \infty$ where $Z_k = V_k + U_k^2$.

Let us define the random variables $i_k(\theta_0) = \mathrm{Var}[U_k|\mathcal{F}_{k-1}] = \mathrm{E}[U_k^2|\mathcal{F}_{k-1}]$ and $I_n(\theta_0) = \sum_{k=1}^n i_k(\theta_0)$. Let $S_n = \sum_{i=1}^n U_k$ and $S_n^* = \sum_{i=1}^n V_k + I_n(\theta_0)$.

In addition to (C1)–(C3), assume that the following condition holds.

(C4) There exists a sequence of constants $K(n) \to \infty$ as $n \to \infty$ such that

(i) $\{K(n)\}^{-1} S_n \xrightarrow{p} 0$.

(ii) $\{K(n)\}^{-1} S_n^* \xrightarrow{p} 0$.

(iii) There exists $a(\theta_0) > 0$ such that for every $\epsilon > 0$ $P[\{K(n)\}^{-1} I_n(\theta_0) \geq 2a(\theta_0)] \geq 1 - \epsilon$ for all $n \geq N(\epsilon)$, and

(iv) $\{K(n)\}^{-1} \sum_{k=1}^n \mathrm{E}|W_k(\theta)| < M < \infty$ for all $\theta \in I$ and for all n.

Lemma 1. Basawa and Prakasa Rao [1980, Theorem 2.1, p. 121] *Under regularity conditions (C1)–(C4) in this section, the likelihood equation has a root $\hat{\theta}_n$ with P_{θ_0}-probability*[1] *approaching* 1 *that is consistent for θ_0 as $n \to \infty$.*

Assume now that we have observed a set of data $y_i, i = 1, \ldots, N$. The Gaussian process regression model has been defined in Equation (1.8); let us recall it (with a

[1] In other words, this is *convergence in probability*: $\lim_{n\to\infty} P(|\hat{\theta}_n - \theta_0| \leq \epsilon) \to 1\ \forall\ \epsilon > 0$.

slight change of notation)

$$y_i|f_i \sim g(f_i) \text{ independently, and} \tag{D.1}$$

$$\boldsymbol{f} = (f_1, \ldots, f_n)^\intercal \sim \mathcal{GP}(0, k(\cdot, \cdot; \boldsymbol{\theta})) \text{ or } \boldsymbol{f} \sim N(\mathbf{0}, \boldsymbol{K}), \tag{D.2}$$

where the $(i,j)^{th}$ element of $\boldsymbol{K}$ is given by the covariance kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta})$. When y_i is assumed to have a normal distribution, $g(f_i)$ is the density of a normal distribution. For the above GPR model, the marginal distribution of $\boldsymbol{y}$ is still normal

$$\boldsymbol{y}|\boldsymbol{\theta} \sim \mathcal{N}_N(\mathbf{0}, \mathbf{C}^{\boldsymbol{\theta}}_{N\times N}). \tag{D.3}$$

where $\mathbf{C} = \mathbf{K} + \sigma^2\mathbf{I}$ and the marginal log-likelihood of the hyperparameters, $\boldsymbol{\theta}$, is given by

$$l_n(\boldsymbol{\theta}) = \log p(\mathcal{D}|\boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{C}_{N\times N}(\boldsymbol{\theta})| - \frac{1}{2}\boldsymbol{y}^\intercal \boldsymbol{C}_{N\times N}(\boldsymbol{\theta})^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi. \tag{D.4}$$

The empirical Bayesian approach chooses the value of $\boldsymbol{\theta}$ which maximises the above marginal log-likelihood.

Corollary 1. *Under regularity conditions (C1)–(C4) in this section, the likelihood equation in (D.4) has a root $\hat{\boldsymbol{\theta}}_n$ with P_{θ_0}-probability approaching 1 which is consistent for θ_0 as $n \to \infty$. In addition, there exists a sequence r_n such that $r_n \to \infty$ as $n \to \infty$ and*

$$r_n^{-1} l_n'(\boldsymbol{\theta}) = \mathcal{O}_p(1) \text{ and } \|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta_0}\| = \mathcal{O}_p(r_n^{-1}). \tag{D.5}$$

Proof of Lemma 1. Notice that as in Equation (D.3), the marginal distribution of $\mathbf{Y}^n = (Y_1, \ldots, Y_n)^T$, $n \geq 1$ has a multivariate normal distribution $\mathcal{N}_n(\mathbf{0}_n, \boldsymbol{C}^{\boldsymbol{\theta}}_{n\times n})$. Additionally, also note that $\mathbf{Y}^k$ has a non-singular $\mathcal{N}_k(\mathbf{0}_k, \boldsymbol{C}^{\boldsymbol{\theta}}_{k\times k})$ distribution. Thus, from the standard theory of multivariate normal distribution, $p_k(\boldsymbol{\theta})$, the conditional probability density of $\mathbf{Y}^k$ given $\mathbf{Y}^{k-1}$ is also a normal density with mean $m_k(\boldsymbol{\theta})$ and variance $v_k(\boldsymbol{\theta})$, where $m_k(\boldsymbol{\theta})$ and $v_k(\boldsymbol{\theta})$ are some functions of $\boldsymbol{\theta}$, determined by the linear combination of the matrices of $\boldsymbol{C}^{\boldsymbol{\theta}}_{k\times k}$ and its inverse.

Thus, without loss of generality, assuming that θ is a scalar, $\phi_k(\theta)$ and its derivatives

are given by

$$\begin{aligned}
\phi_k(\theta) &= -\log(\sqrt{2\pi m_k(\theta)}) - \left\{\frac{1}{2v_k(\theta)}(y_k - m_k(\theta))^2\right\} \\
\dot{\phi}_k(\theta) &= -\frac{m_k'(\theta)}{m_k(\theta)} + \frac{v_k'(\theta)}{2v_k(\theta)^2}(y_k - m_k(\theta))^2 - \frac{(y_k - m_k(\theta))}{v_k(\theta)}m_k'(\theta) \\
\ddot{\phi}_k(\theta) &= A_k(\theta)(y_k - m_k(\theta))^2 + B_k(\theta)(y_k - m_k(\theta)) + C_k(\theta),
\end{aligned}$$

where $A_k(\theta), B_k(\theta), C_k(\theta)$ are some functions of θ which are based on the first two derivatives of $m_k(\theta)$ and $v_k(\theta)$.

Notice that $z_k = (y_k - m_k(\theta))/\sqrt{v_k(\theta)}$ has a standard normal distribution and its square has a χ^2 distribution, given y^{k-1}. Therefore, it follows that $(C1)$–$(C3)$ hold under the non-singular normal distribution with suitable mean and variance, thrice differentiable with respect to θ. In addition, since the conditional distribution of z_k is a non degenerate normal distribution, there exist constants $M_1 > 0$ and $m_1 > 0$ such that $i_k(\theta_0) = m_1 \leq \text{Var}[U_k|\mathcal{F}_{k-1}] < M_1$. Since the distribution of z_k is determined independently of k, the constants M_1 and m_1 are achieved uniformly on k.

Let us now define $K(n) = I_n(\theta_0)$. Then, it follows that $K(n) = \mathcal{O}(n)$ which satisfies $(i) - (iii)$ in (C4). In addition, the third derivative of ϕ_k, $\dddot{\phi}_k(\theta)$ is also given based on the first, the second and the third derivatives of $m_k(\theta)$ and $v_k(\theta)$. Note that as $m_k(\theta)$ and $v_k(\theta)$ are thrice differentiable with respect to θ for all $\theta \in I$, it is clear that the condition (iv) of $(C4)$ also holds. Therefore, the solution of the likelihood equation $\hat{\boldsymbol{\theta}}_n$ is consistent for $\boldsymbol{\theta_0}$ by Lemma 1.

In order to check the asymptotic normality, additional conditions for asymptotic normality given by Basawa and Prakasa Rao [1980] need to be verified. However, since convergence in probability implies convergence in distribution, it is certain that there exists a sequence r_n such that

$$r_n^{-1} l_n'(\boldsymbol{\theta}) = \mathcal{O}_p(1) \text{ and } \|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta_0}\| = \mathcal{O}_p(r_n^{-1}).$$

The proof for the posterior consistency of the empirical Bayes estimates of the GPR model is completed.

We recall Equation (4.8). There exists $\boldsymbol{X}^*$ such that [2]

$$\begin{aligned} p(\boldsymbol{Y}|\boldsymbol{\theta}) &= \int p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})p(\boldsymbol{X})d\boldsymbol{X} \\ &\propto p(\boldsymbol{Y}|\boldsymbol{X}^*,\boldsymbol{\theta})p(\boldsymbol{X}^*) \\ &= \prod_{d=1}^{D} p(\boldsymbol{y}_{(d)}|\boldsymbol{X}^*,\boldsymbol{\theta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n^*). \end{aligned}$$

Let us consider the special case

$$p(\boldsymbol{Y}|\boldsymbol{X}^*,\boldsymbol{\theta}) = \prod_{d=1}^{D} p(\boldsymbol{y}_{(d)}|\boldsymbol{X}^*,\boldsymbol{\theta}_d),$$

where all the elements in $\boldsymbol{\theta}_d$ are distinct for different d. Thus, Lemma 1 and Corollary 1 can be used to

$$l_{d,n} = \log p(\boldsymbol{y}_{(d)}|\boldsymbol{X}^*,\boldsymbol{\theta}_d) \tag{D.6}$$

for $d = 1,\ldots,D$. This leads to the following theorem.

Theorem 1. *Under regularity conditions (C1)–(C4), the likelihood equation in (D.6) has a root $\hat{\boldsymbol{\theta}}_{d,n}$ with P_{θ_0}-probability approaching one which is consistent for $\theta_{d,0}$ as $n \to \infty$, where $\theta_{d,0}$ is the true value.*

Furthermore, we can define

$$l_n = \sum_{i=1}^{d} \log p(\boldsymbol{y}_{(d)}|\boldsymbol{X}^*,\boldsymbol{\theta}_d). \tag{D.7}$$

We have the following result:

Corollary 2. *Under regularity conditions (C1)–(C4), the likelihood equation in (D.7) has a root $\hat{\boldsymbol{\theta}}_n$ with P_{θ_0}-probability approaching one which is consistent for θ_0 as $n \to \infty$.* In addition, there exists a sequence γ_n such that $\gamma_n \to \infty$ as $n \to \infty$ and

$$\gamma_n^{-1} l_n'(\boldsymbol{\theta}) = \mathcal{O}_p(1) \text{ and } \|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta_0}\| = \mathcal{O}_p(\gamma_n^{-1}). \tag{D.8}$$

[2] This is related to the *Mean value theorem for definite integrals*. Note that despite the fact the integral is indefinite, the function will be $p(\boldsymbol{Y},\boldsymbol{X}|\boldsymbol{\theta})$ will be zero when $\mathbf{X}$ takes extreme values (either positive or negative); therefore the indefinite integral can be subdivided and thought of as a definite integral where the mean value theorem applies.

The proof is similar to Corollary 1.

Appendix E

Continuous stirred tank reactor (CSTR) model

The process flow is depicted in Figure E.1. The reaction, $A \rightarrow B$, is irreversible, exothermic and takes place in liquid phase. A feed stream of reactant A with flow rate F_a is premixed with a solvent stream flowing at a rate F_s; the concentration of reactant A in both streams is C_{aa} and C_{as} respectively. This premixed stream, with reactant concentration C_i and flow rate F, is then fed into the jacketed reactor where the reaction takes place.

A summary of the process variables and simulation parameters is given in Table E.1; note that variables a_1 and a_2 are used to simulate degradation in the reaction rate due to impurities and fouling of the water-cooled heat exchanger respectively.

Variable type	
Controlled variable:	T
Manipulated variable:	F_c
Disturbances:	$C_{aa}, C_{as}, F_s, T_i, T_{ci}, a_1, a_2$
Measured variables:	$T_{ci}, T_i, C_{aa}, C_{as}, F_s, F_c, C, T$

Table E.1: CSTR process variables summary

The system has only a PI control loop whose aim is to maintain the outlet temper-

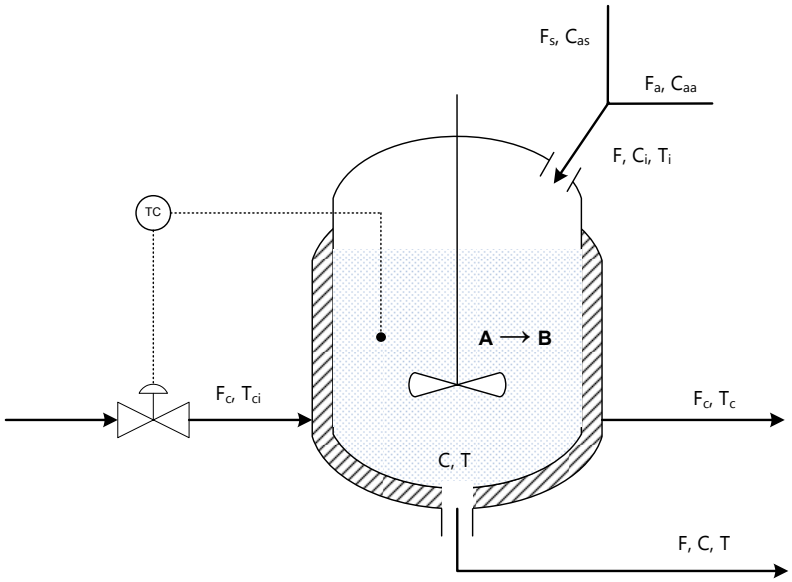

Figure E.1: Process flow diagram of the non-isothermal CSTR system; C_i and C refer to the concentration of reactant A.

ature T at a set value. This is done by controlling the flow of cooling water, F_c, which enters the reactor jacket at a temperature T_{ci} and leaves at a temperature T_c. The model assumes perfect mixing, constant physical properties and negligible shaft work. The dynamic behaviour of this process is governed by two ordinary differential equations(ODE). Firstly, the mass balance for reactant A

$$V\frac{dC}{dt} = F(C - C_i) - Vr \tag{E.1}$$

where V is the volume of reacting liquid; r is an Arrhenius-type reaction rate given as $r = k_0 e^{-E/RT} C$ with k_0 being the pre-exponential factor and R the gas constant.

The second ODE is the global energy balance written as follows:

$$\begin{aligned} V\rho c_p\frac{dT}{dt} = \rho c_p F(T_i - T) - \frac{aF_c^{b+1}}{F_c + F_c^b/2\rho_c c_{pc}}(T - T_{ci}) \\ + (-\Delta H_r)Vr \end{aligned} \tag{E.2}$$

where ρ and ρ_c are the densities of the reacting mixture and the cooling water, respectively, whereas c_p and c_{pc} as their specific heat capacities; ΔH_r is the heat of the reaction.

Simulation parameters
$V = 1\text{m}^3; \rho = 10^6\text{g/m}^3; \rho_c = 10^6\text{g/m}^3; \frac{E}{R} = 8330.1\text{K}; c_p = 1\text{cal/(g} \cdot \text{K)};$ $c_{pc} = 1\text{cal/(g} \cdot \text{K)}; k_0 = 10^{10}(\text{m}^3/\text{kmol min}); a = 1.678 \cdot 10^6(\text{cal/min K});$ $b = 0.5; \Delta H_r = -1.3 \cdot 10^7\text{cal/kmol}$
Initial conditions
$T_i = 370\text{K}; T_{ci} = 365\text{K}; T = 368.25\text{K}; F_c = 15\text{m}^3/\text{min}; F_s = 0.9\text{m}^3/\text{min};$ $F_a = 0.1\text{m}^3/\text{min}; C_i = 0.8\text{Kmol/m}^3; C_{as} = 0.1\text{Kmol/m}^3; C_{aa} = 19.1\text{Kmol/m}^3$
PI controller
$K_c = -1.5; T_I = 5.0$

Table E.2: CSTR simulation parameters

All process disturbances are simulated as first order autoregressive processes with the following equation:

$$x_t = \phi x_{t-1} + e_t,$$

where $e_t \sim \mathcal{N}(0, \sigma_e^2)$ and x_t refer to *process disturbances* as shown in Table E.1; allocated values of σ_e^2 are given in Table E.3. Finally, this table also shows the measurement noise, $e_M \sim \mathcal{N}(0, \sigma_M^2)$, that is added to all measured variables.

Variable	*Measurement noise,* σ_M^2	*Process noise,* σ_e^2	*AR coefficients,* ϕ
T	$4 \cdot 10^{-4}$	-	-
C	$2.5 \cdot 10^{-5}$	-	-
F_c	$1.0 \cdot 10^{-2}$	-	-
T_{ci}	$2.5 \cdot 10^{-3}$	$0.475 \cdot 10^{-1}$	0.9
T_i	$2.5 \cdot 10^{-3}$	$0.475 \cdot 10^{-1}$	0.9
C_{aa}	$1.0 \cdot 10^{-2}$	$0.475 \cdot 10^{-1}$	0.9
F_a	$2.5 \cdot 10^{-3}$	-	-
C_{as}	$2.5 \cdot 10^{-5}$	$1.875 \cdot 10^{-3}$	0.5
F_s	$4.0 \cdot 10^{-6}$	$0.19 \cdot 10^{-2}$	0.9
a_1	-	$0.19 \cdot 10^{-2}$	0.9
a_2	-	$0.0975 \cdot 10^{-2}$	0.95

Table E.3: CSTR measurement and process noise

The model was originally proposed by Yoon and MacGregor [2001]; this is a slight modification from the original which does not have the outlet concentration controller.

Bibliography

Alcala, C. F. and Qin, S. J. (2010). Reconstruction-based contribution for process monitoring with kernel principal component analysis. *Industrial & Engineering Chemistry Research*, 49(17):7849–7857. Cited on page 64.

Archer, C. and Jennrich, R. (1973). Standard errors for rotated factor loadings. *Psychometrika*, 38(4):581–592. Cited on page 40.

Basawa, I. V. and Prakasa Rao, B. (1980). *Statistical Inference for Stochastic Processes*. Academic Press, London. Cited on pages 136, 137, and 139.

Bersimis, S., Psarakis, S., and Panaretos, J. (2006). Multivariate statistical process control charts: An overview. *Quality and Reliability Engineering International*, 23(5):517–543. Cited on page 2.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer. Cited on page 54.

Bollen, K. A. (1989). *Structural Equations with Latent Variables*. Wiley, New York. Cited on pages 26, 30, 31, 73, and 92.

Boomsma, A. and Hoogland, J. (2001). The robustness of lisrel modeling revisited. In *Structural equation modeling: Present and future*, pages 139–168. Lincolnwood, IL: Scientific Software International. Cited on page 32.

Box, G. E. P. (1954). Some theorems on quadratic forms applied in the study of analysis of variance problems: effect of inequality of variance in one-way classification. *Ann. Math. Stat.*, 25:290–302. Cited on page 9.

Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate Behavioral Research*, 36:111–150. Cited on page 28.

Carlin, B. and Louis, T. (2000). *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall/CRC, 2^{nd} edition. Cited on page 21.

Chiang, L., Russell, E., and Braatz, R. (2001). *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag, New York. Cited on page 2.

Choi, S. W., Lee, C., Lee, J.-M., Hyun Park, J., and Lee, I.-B. (2005). Fault detection and identification of nonlinear processes based on kernel pca. *Chemometrics and Intelligent Laboratory Systems*, 75:55–67. Cited on pages 46 and 57.

Choi, S. W., Morris, J., and Lee, I.-B. (2008). Nonlinear multiscale modelling for fault detection and identification. *Chemical Engineering Science*, 63:2252–2266. Cited on page 64.

Choi, T. and Schervish, M. J. (2007). On posterior consistency in nonparametric regression problems. *J. Multivar. Anal.*, 98:1969–1987. Cited on page 96.

Conlin, A., Martin, E., and Morris, A. (2000). Confidence limits for contribution plots. *Journal of Chemometrics*, 14(5-6):725–736. Cited on page 11.

Crawford, C. and Ferguson, G. (1970). A general rotation criterion and its use in orthogonal rotation. *Psychometrika*, 35:321–332. Cited on page 28.

Cudeck, R. and MacCallum, R. C. (2007). *Factor Analysis at 100: Historical Developments and Future Directions*. Lawrence Erlbaum Associates, New Jersey. Cited on pages 23 and 24.

Davison, A. C. (2003). *Statistical Models*. Cambridge University Press, Cambridge. Cited on page 100.

Dong, D. and McAvoy, T. (1996). Nonlinear principal component analysis based on principal curves and neural networks. *Computers and Chemical Engineering*, 20(1):65–78. Cited on pages 12, 46, 53, and 68.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360. Cited on page 107.

Fox, J. (2006). Teacher's corner: Structural equation modeling with the sem package in r. *Structural Equation Modeling: A Multidisciplinary Journal*, 13(3):465–486. Cited on page 37.

Frank, I. and Friedman, J. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35:109–135. Cited on page 107.

García-Muñoz, S., Kourti, T., MacGregor, J., Mateos, A., and Murphy, G. (2003). Troubleshooting of an industrial batch process using multivariate methods. *Industrial and Engineering Chemistry Research*, 42(15):3592–3601. Cited on page 2.

Ge, Z. and Song, Z. (2010). Nonlinear probabilistic monitoring based on the Gaussian process latent variable model. *Industrial & Engineering Chemistry Research*, 49(10):4792–4799. Cited on pages 45, 52, 60, and 68.

Ghahramani, Z. (2004). Unsupervised learning. In Bousquet, O., Raetsch, G., and von Luxburg, U., editors, *Advanced Lectures on Machine Learning*, volume 3176, pages 72–112. Springer-Verlag. Cited on page 2.

Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations*. Wiley, New York. Cited on pages 46 and 71.

Harman, H. H. (1976). *Modern Factor Analysis*. University of Chicago Press, Chicago, 3^{rd} edition. Cited on pages 24 and 33.

Hastie, T. and Stuetzle, W. (1989). Principal curves. *J. Am. Statistical Assoc.*, 84:502–516. Cited on page 53.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer-Verlag, 2^{nd} edition. Cited on pages 2 and 7.

Hayashi, K. and Bentler, P. (2000). The asymptotic covariance matrix of maximum-likelihood estimates in factor analysis: the case of nearly singular matrix of estimates of unique variances. *Linear Algebra and its Applications*, 321:153–173. Cited on page 34.

Hibbert, D. B., Minkkinen, P., Faber, N., and Wise, B. M. (2009). Iupac project: A glossary of concepts and terms in chemometrics. *Analytica Chimica Acta*, 642(1-2):3 – 5. Cited on page 2.

Hoerl, A. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67. Cited on page 107.

Jackson, J. E. (1991). *A User's Guide to Principal Components*. John Wiley and Sons, NJ. Cited on pages 9 and 10.

Jackson, J. E. and Mudholkar, G. S. (1979). Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349. Cited on page 10.

Jennrich, R. (1974). Simplified formulae for standard errors in maximum-likelihood factor analysis. *British Journal of Mathematical and Statistical Psychology*, 27:122–131. Cited on page 34.

Jennrich, R. and Thayer, D. (1973). A note on lawley's formulas for standard errors in maximum likelihood factor analysis. *Psychometrika*, 38(4):571–580. Cited on page 34.

Jia, F., Martin, E. B., and Morris, A. J. (1998). Non-linear principal components analysis for process fault detection. *Computers & Chemical Engineering*, 22(Supplement 1):S851 – S854. European Symposium on Computer Aided Process Engineering-8. Cited on page 47.

Johnson, R. A. and Li, R. (2006). Multivariate statistical process control schemes for controlling a mean. In Pham, H., editor, *Springer Handbook of Engineering Statistics*, pages 327–344. Springer-Verlag, London. Cited on page 9.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, 2^{nd} edition. Cited on page 6.

Jöreskog, K. G. (1967). Some contributions to maximum likelihood factor analysis. *Psychometrika*, 32(4):443–482. Cited on page 34.

Jöreskog, K. G. (2007). Factor analysis and its extensions. In *Factor Analysis at 100: Historical Developments and Future Directions*, pages 47–77. Lawrence Erlbaum Associates, New Jersey. Cited on page 32.

Jöreskog, K. G. and Sörbom, D. (1997). *LISREL 8: User's Reference Guide*. Scientific Software. Cited on page 93.

Kaiser, H. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23:187–200. Cited on page 28.

Kourti, T. (2002). Process analysis and abnormal situation detection: from theory to practice. *Control Systems Magazine, IEEE*, 22(5):10 – 25. Cited on pages 1, 3, and 15.

Kourti, T. (2003). Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *Journal of Chemometrics*, 17(1):93–109. Cited on page 2.

Kourti, T. and MacGregor, J. (1995). Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28(1):3 – 21. Cited on pages 2 and 5.

Krzanowski, W. and Marriott, F. (1995). *Multivariate Analysis Part 2: Classification, Covariance Structures and Repeated Measurements*. Kendall's Library of Statistics, London: Arnold. Cited on page 27.

Ku, W., Storer, R. H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30(1):179–196. Cited on page 43.

Kuss, M. and Rasmussen, C. (2005). Assessing approximate inference for binary Gaussian process classification. *Journal of Machine Learning Research*, 6:1679–1704. Cited on page 99.

Lawley, D. N. and Maxwell, A. E. (1971). *Factor Analysis as a Statistical Method (Butterworths Mathematical Texts)*. Butterworths. Cited on page 34.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816. Cited on pages i, 45, 48, 51, 52, 74, 76, 93, 114, and 122.

Lawrence, N. D. (2004). Gaussian process models for visualisation of high dimensional data. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems*, pages 329–336. MIT Press, Cambridge, MA. Cited on pages 22, 45, and 77.

Lawrence, N. D., Seeger, M., and Herbrich, R. (2003). Fast sparse Gaussian process methods: The informative vector machine. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems*, volume 15, pages 625–632. MIT Press, Cambridge, MA. Cited on page 51.

Lee, J. M., Yoo, C. K., Choi, S. W., Vanrolleghem, P. A., and Lee, I. B. (2004). Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 59(1):223–234. Cited on pages 64 and 68.

Lehmann, E. L. and Casella, G. (1998). *Theory of Point Estimation (Springer Texts in Statistics)*. Springer, New York, 2^{nd} edition. Cited on pages 95 and 136.

Lindley, D. V. and Smith, A. F. M. (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(1):1–41. Cited on pages 19 and 117.

MacGregor, J., Jaeckle, C., Kiparissides, C., and Koutoudi, M. (1994). Process monitoring and diagnosis by multiblock pls methods. *AIChE J.*, 40(5):826828. Cited on page 11.

MacGregor, J. and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403 – 414. Cited on pages 1 and 10.

MacGregor, J., Yu, H., García-Muñoz, S., and Flores-Cerrillo, J. (2005). Data-based latent variable methods for process analysis, monitoring and control. *Computers and Chemical Engineering*, 29(6):1217 – 1223. Cited on page 2.

MacKay, D. J. C. (1999). Introduction to Gaussian processes. Technical report, Cambridge University. Cited on page 19.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. Academic Press, London. Cited on pages 4 and 24.

Matheron, G. (1963). *Traité de geostatistique appliquée: Le Krigéage*. Number 2 in Mémoires du Bureau de recherches géologiques et minières. Ed. B.R.G.M., Paris. Cited on page 16.

MATLAB (2010). *Neural Network Toolbox, version 6.0.4 (Matlab R2010a)*. The MathWorks Inc., Natick, Massachusetts. Cited on page 54.

McCabe, G. P. (1984). Principal variables. *Technometrics*, 26(2):137–144. Cited on pages 3, 6, 37, and 112.

McDonald, R. and Hartmann, W. (1992). A procedure of obtaining initial values of parameters in the ram model. *Multivariate Behavioral Research*, 27(1):57–76. Cited on page 37.

Miller, P., Swanson, R., and Heckler, C. (1998). Contribution plots: a missing link in multivariate quality control. *Int. J. Appl. Math. Comput. Sci.*, 8:775–792. Cited on pages 11 and 39.

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533. Cited on page 122.

Montgomery, D. and Keats, J. (1994). Integrating statistical process control and engineering process control. *Journal of Quality Technology*, 26:79–87. Cited on page 2.

Nabney, I. (2002). *NETLAB: algorithms for pattern recognition*. Advances in pattern recognition. Springer-Verlag, London, Berlin, Heidelberg. Cited on page 123.

Neal, R. M. (1994). *Bayesian Learning for Neural Networks*. PhD thesis, Dept. of Computer Science, University of Toronto. Cited on pages 16, 45, and 106.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics No. 118. Springer-Verlag, New York. Cited on page 18.

Nickisch, H. and Rasmussen, C. (2008). Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078. Cited on pages 99 and 115.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer, 2^{nd} edition. Cited on page 122.

Nomikos, P. and MacGregor, J. F. (1995). Multivariate spc charts for monitoring batch processes. *Technometrics*, 37(1):41–59. Cited on page 9.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society B*, 40:1–42. Cited on page 16.

Polak, E. and Ribière, G. (1969). Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle*, 3(1):35–43. Cited on pages 122 and 123.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes, 3rd Edition: The Art of Scientific Computing*. Cambridge University Press. Cited on page 116.

Qin, S. (2003). Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17:480–502. Cited on pages 2 and 11.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. Cited on page 37.

Rasmussen, C. E. (1996). *Evaluation of Gaussian Processes and other Methods for Non-linear Regression*. PhD thesis, Dept. of Computer Science, University of Toronto. Cited on page 122.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA. Cited on pages 16, 73, 100, 101, 117, and 119.

Rencher, A. (2002). *Methods of Multivariate Analysis*. John Wiley and Sons, 2^{nd} edition. Cited on pages 31 and 33.

Schölkopf, B., Smola, A., and Muller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319. Cited on pages 46 and 71.

Serradilla, J., Shi, J. Q., and Morris, J. (2011). Fault detection based on Gaussian process latent variable models. *Chemometrics and Intelligent Laboratory Systems*, 109(1):9–21. Cited on pages 22, 45, and 114.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press. Cited on page 19.

Shi, J. Q. and Choi, T. (2011). *Gaussian Process Regression Analysis for Functional Data*. Chapman & Hall/CRC, London. Cited on pages 16, 21, 51, 95, 123, and 136.

Simoglou, A., Martin, E. B., and Morris, A. J. (2000). Multivariate statistical process control of an industrial fluidised-bed reactor. *Control Engineering Practice*, 8(8):893–909. Cited on pages 15 and 46.

Srinivasan, R. and Qian, M. (2007). State-specific key variables for monitoring multi-state processes. *Chemical Engineering Research and Design*, 85(12):1630–1644. Cited on page 6.

Tan, S. and Mavrovouniotis, M. L. (1995). Reducing data dimensionality through optimizing neural network inputs. *AIChE Journal*, 41:1471 – 1480. Cited on pages 45 and 47.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288. Cited on page 107.

Tipping, M. and Bishop, C. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622. Cited on pages 3, 26, and 113.

Valle, S., Li, W., and Qin, S. (1999). Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. *Industrial and Engineering Chemistry Research*, 38(11):4389–4401. Cited on page 7.

Wang, B. and Shi, J. (2011). Generalized Gaussian process regression model for non-Gaussian functional data. *Article in revission*. Cited on page 68.

Williams, C. and Barber, D. (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1342–1351. Cited on page 101.

Williams, C. and Rasmussen, C. (1996). Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8:514–520. Cited on page 16.

Wold, S. (1978). Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–405. Cited on page 7.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal components analysis. *Chemometr. Intell. Lab. System 2*, pages 37–52. Cited on page 6.

Yalcin, I. and Amemiya, Y. (2001). Nonlinear factor analysis as a statistical method. *Statistical Science*, 16(3):275–294. Cited on page 24.

Yi, G. (2009). *Variable selection with penalized Gaussian Process regression models*. PhD thesis, School of Mathematics and Statistics, Newcastle University. Cited on pages 107 and 122.

Yi, G., Shi, J. Q., and Choi, T. (2011). Penalized Gaussian process regression and classification for high-dimensional nonlinear data. *Biometrics*. Cited on pages 18, 99, 107, and 115.

Yoon, S. and MacGregor, J. F. (2001). Fault diagnosis with multivariate statistical models, part i: using steady state fault signatures. *Journal of Process Control*, 11:387–400. Cited on pages 64 and 145.

Yoon, S. and MacGregor, J. F. (2004). Principal component analysis of multiscale data for process monitoring and fault diagnosis. *AIChE Journal*, 50(11):2891–2903. Cited on page 64.

Zhang, J., Martin, E., and Morris, A. (1996). Fault detection and diagnosis using multivariate statistical techniques. *Transactions of the Institution of Chemical Engineers*, 74:89–96. Cited on page 61.

Zhang, J., Martin, E., and Morris, A. (1997). Process monitoring using non-linear statistical techniques. *Chemical Engineering Journal*, 67(3):181 – 189. Cited on pages 10 and 55.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429. Cited on page 107.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal Of The Royal Statistical Society Series B*, 67(2):301–320. Cited on page 107.