Anomaly Detection and Removal Using Non-Stationary Gaussian Processes

Steven Reece. Roman Garnett, Michael Osborne and Stephen Roberts
Robotics Research Group
Dept. Engineering Science
Oxford University, UK.

I. INTRODUCTION

Gaussian Processes (GPs) are experiencing a resurgence of interest. Current applications are in diverse fields such as geophysics, medical imaging, multi-sensor fusion [5] and sensor placement [2]. A GP is often thought of as a “Gaussian over functions” [7]. It can be used to construct a distribution over functions via a prior on the functions’ values. The prior is specified through a positive-definite kernel, which determines the covariance between two outputs as a function of their corresponding inputs. A GP is fully described by its mean and covariance functions. Suppose we have a set of training data

\[ D = \{(x_1, y_1), \ldots, (x_n, y_n)\} \]

where \( f(x_i) \) is the real process and \( \epsilon_i \) is zero-mean Gaussian with variance \( \sigma^2 \). For convenience both inputs and outputs are aggregated into \( X = \{x_1, \ldots, x_n\} \) and \( Y = \{y_1, \ldots, y_n\} \) respectively. The GP estimates the value of the function \( f \) at sample locations \( X_s = \{x_{s1}, \ldots, x_{sm}\} \). The basic GP regression equations are given in [7]:

\[ \hat{f}_s = K(X_s, X)[K(X, X) + \sigma_n^2 I]^{-1}Y, \]
\[ \text{Cov}(f_s) = K(X_s, X_s) - K(X_s, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X_s, X)^T, \]

where \( \hat{f}_s \) is the marginal posterior mean at \( X_s \) and \( \text{Cov}(f_s) \) is the corresponding covariance. The prior covariance at \( X_s \) is \( K(X_s, X_s) \) where the covariance matrix \( K(X_s, X_s) \) has elements \( K_{ij} = \mathbb{K}(x_i, x_j; \theta_k) \). The function \( \mathbb{K} \) is called the kernel function. The kernel function captures the functions come with parameters, \( \theta_k \), which can be tuned to suit the application. The kernel function is chosen according to known properties of the underlying processes, such smoothness and stationarity. There are many existing kernels to choose from (see, for example, [7]). We will develop a new kernel for our fault recovery algorithm in Section III. The prior mean is traditionally set to zero and we follow this convention. However, the results in this paper can be readily generalised to non-zero prior means. The term \( \sigma_n^2 I \) captures the noise in Eqn 5.

The parameters \( \theta \) (which include \( \sigma_n^2 \) and parameters associated with the kernel function, \( \theta_K \)) are the hyperparameters of the GP. These can be learned using, for example, Bayesian learning techniques:

\[ p(\theta | Y, X) = \frac{p(Y | X, \theta) p(\theta)}{p(Y | X)}. \]

The hyperparameters are usually given a vague prior distribution, \( p(\theta) \).

II. FAULTY SIGNALS

We assume that a faulty signal is the linear composition of the real physical phenomenon of interest and a deterministic offset induced by a fault either in the sensor measurement or in the physical phenomenon itself. Again, the signal is assumed to be drawn from a noisy process:

\[ y_i = f(x_i) + \epsilon(x_i) + \epsilon_i \]

where \( f(x_i) \) is the real process, \( \epsilon(x_i) \) is the fault process and \( \epsilon_i \) is zero-mean Gaussian with variance \( \sigma^2 \).

Again, both inputs and outputs are aggregated into \( X = \{x_1, \ldots, x_n\} \) and \( Y = \{y_1, \ldots, y_n\} \) respectively. The GP regression equations, (2) and (3), are modified so that both the real and fault processes can be inferred:

\[ \hat{f}_s = K_f(X_s, X)[K_s(X, X) + \sigma_n^2 I]^{-1}Y, \]
\[ \text{Cov}(f_s) = K_f(X_s, X_s) - K_f(X_s, X)[K_s(X, X) + \sigma_n^2 I]^{-1}K_f(X_s, X)^T, \]
\[ \hat{\epsilon}_s = K_e(X_s, X)[K_s(X, X) + \sigma_n^2 I]^{-1}Y, \]
\[ \text{Cov}(f_s) = K_e(X_s, X_s) - K_e(X_s, X)[K_s(X, X) + \sigma_n^2 I]^{-1}K_e(X_s, X)^T, \]

where \( K_f(X, X) = K_f(X, X) + K_e(X, X) \). \( \hat{f}_s \) is the marginal posterior mean at \( X_s \) and \( \hat{\epsilon}_s \) is the corresponding covariance of the real process. \( \hat{\epsilon}_s \) and \( \text{Cov}(\epsilon_s) \) are the marginal posterior mean and corresponding covariance of the
fault process, respectively. The prior covariances at \( X_s \) are \( K_f(X_s, X_s) \) and \( K_c(X_s, X_s) \) for the real process and the fault process, respectively, where, again, each covariance matrix is generated by a kernel function. We may assign different kernels to the real and fault processes.

We shall consider two kinds of faults in this paper, drift and bias (see Figure 1). Although, our approach can be extended to arbitrary fault types. Both faults are temporary, so they have a start time and an end time. The fault process is assumed to be zero outside of this time interval. Drift faults are gradual. Starting from zero error they grow over time and then either shrink back to zero gradually or snap back to zero instantaneously. Bias faults are severe and immediate. They induce a significant error in the signal at onset which persists until the fault subsides at which point the signal snaps back onto the real process. The drift fault is continuous at the start whereas the bias fault is discontinuous.

![Fig. 1. Typical real process and faulty observations for both bias and drift faults.](image)

In order to model the drift fault we develop a kernel which guarantees that the fault is continuous at onset. We call this kernel the continuous, conditionally independent (or CCI) kernel. The CCI kernel has applications outside of fault recovery. This non-stationary kernel splices two or more locally stationary kernels together whilst preserving function continuity throughout. Consequently, we present this kernel thoroughly for the first time.

### III. Non-Stationary Kernels

Many applications use stationary covariance functions for which the kernel is a function of the distance between the input points. Stationary covariance functions are appealing due to their intuitive interpretation and their relative ease of construction. Unfortunately, stationary GP functions are not applicable in applications where there are input-dependent variations in the model hyperparameters (e.g. length-scale, amplitude) and kernel families. Consequently, non-stationary GP functions have been proposed, such as the neural network kernel [11] and the Gibbs kernel [1].

Methods for deriving non-stationary kernels from stationary kernels have also been proposed. Perhaps the earliest approach was to assume a stationary random field within a moving window [3]. This approach works well when the non-stationarity is smooth and gradual. It fails when sharp changes in the kernel structure occur. An alternative solution is to introduce an arbitrary non-linear mapping (or warping) \( u(x) \) of the input \( x \) and then apply a stationary covariance function in the \( u \)-space [9]. Unfortunately, this approach does not handle sharp changes between different locally applied kernels very well [4]. The mixture of GPs [10] approach uses the EM algorithm to simultaneously assign GP mixtures to locations and optimise their hyperparameters. Although the mixture approach can use arbitrary local GP kernels, it does not guarantee function continuity over GP kernel transitions. Paciorek [6] proposes a non-stationary GP kernel which guarantees continuity over region boundaries. Unfortunately, this approach requires that the local, stationary kernels belong to the same family.

#### A. Example: The Gibbs Non-Stationary Kernel

Gibbs [1], [7] derived the covariance function:

\[
K(x, x') = \prod_{d=1}^{D} \left( \frac{2l_d(x)l_d(x')}{l_d^2(x) + l_d^2(x')} \right)^{1/2} \exp \left( -\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{l_d^2(x) + l_d^2(x')} \right)
\]

where each length-scale \( l_i(x) \) is an arbitrary positive function of \( x \) and \( D \) is the dimensionality of \( x \). If the length-scale varies rapidly then the covariance drops off quite sharply due to the pre-factor in Eqn 6. As a consequence the inferred function estimate can be quite uncertain at length-scale boundaries. This is demonstrated in Figure 2(a) for which the length-scale changes from \( l(x) = 35 \) for \( x \leq 130 \) to \( l(x) = 15 \) for \( x > 130 \). Further, the Gibbs kernel does not guarantee that functions generated by the kernel are continuous. Figure 2(b) shows a typical sample drawn from the posterior Gaussian distribution represented in Figure 2(a).

![Fig. 2. Part (a) shows a function and its estimate obtained using the Gibbs Non-stationary Gaussian Process kernel. Also, (b) a random sample drawn from the Gibbs non-stationary kernel typically showing a discontinuity where the length-scale changes.](image)

#### B. Example: Warping of the Input Space

This example demonstrates the limitation of modelling piece-wise stationary functions by warping the input space as proposed by [9]. Figures 3 and 4 show a continuous wedge function with low and high signal-to-noise (SNR) respectively. The figures also show the mean and first standard deviation of two models:

- a warped squared exponential
- two squared exponential functions joined, using the new kernel, at \( x = 100 \).

For low SNR the warping approach can smooth over features of high curvature such as the wedge apex. For high SNR
In many applications a completely different GP function may be required to model different regions within the space of interest and the family of non-stationary covariance functions in the literature may be too restrictive to model these problems especially when there are function continuity conditions at region boundaries. We will show how arbitrary stationary GP kernels can be combined to form non-stationary GP covariance priors which preserve function continuity. We shall call the new kernel the Markov Region Link (MRL).

The paper is organised as follows. Section IV presents the problem description as a piece-wise stationary problem with boundary constraints. Then Section V presents the MRL kernel for functions which are continuous boundaries and this is extended to cases where function derivatives are continuous at region boundaries in Section VI. In Sections VII and VIII we demonstrate the efficacy of our approach on simulated data from a faulty sensor target estimation problem as well as a dataset involving EOG artifact corrupted EEG signals. Finally, we conclude in Section IX.

IV. PROBLEM DESCRIPTION

We will assume that a domain can be partitioned such that within regions (tiles) the process is stationary [4]. However, each region may be modelled by kernels from different families. For example, one region may be modelled by a Matérn kernel whereas a neighbouring region may be modelled by a mixture of squared exponential and period kernels. We do not assume that the functions generated by these kernels are independent between regions and, although we desire sharply changing GP kernels or hyperparameters at region boundaries, we would also like to preserve function continuity at the boundaries.

Two regions are labelled $R_1$ and $R_2$ and collectively they form the global region $R$. A function over $R$ is inferred at sample locations $X_*$ given training data at locations $X = \{x_1, \ldots, x_n\}$. However, the training data locations are partitioned between the regions and the region boundary. Let $X_r$ be the locations internal to region $R_r$ and let $X_{rb}$ be the locations on the boundary. Then:

$$ X = X_1 \cup X_{rb} \cup X_2 . $$

We will assume that the function can be modelled using a stationary GP in each region and endeavour to design a global GP covariance prior which preserves the individual region kernels. We will also endeavour to preserve function continuity where desired across region boundaries including, for example, function continuity or continuity of function derivatives. Thus, the necessary conditions are:

1) The global kernel $K$ preserves the individual region kernels, $K_r$. That is, $K(X, X) = K_r(X, X)$ for all $X \subseteq X_r \cup X_{rb}$ and all regions $r$.

2) The global kernel preserves function continuity, or derivative continuity, at the boundary.

Proposition 1: If two regions, labelled 1 and 2, are joined at the boundary $X_{rb}$ and a function defined over the regions is modelled by $K_1$ in region $R_1$ and $K_2$ in $R_2$ and the function is continuous at the boundary then:

$$ K_1(X_{rb}, X_{rb}) = K_2(X_{rb}, X_{rb}) = K_{rb} $$

\[^1\text{In 1D problems the region boundary is often referred to as the changepoint.}\]
The boundary covariance $K_B$ is a hyperparameter which can be learned from the training data.

V. THE MARKOV LINK KERNEL

We assume that the processes internal to each region are conditionally independent given the process at the boundary $B$. The corresponding graphical model is shown in Figure 5 and $f(X_1)$ and $f(X_2)$ are the processes internal to the regions labelled 1 and 2 and $f(X_B)$ is the process at the boundary. The process in region 1 and at the boundary is modelled using the GP kernel $K_1$. The rows and columns of $K_1$ correspond to the stacked vector $O_1 = (X_1, X_B)$:

$$K_1 = \begin{pmatrix} K_1(X_1, X_1) & K_1(X_1, X_B) \\ K_1(X_1, X_B) & K_1(X_B, X_B) \end{pmatrix}.$$ 

Similarly, the process in region 2 and at the boundary is modelled using the GP kernel $K_2$ where the row and columns correspond to the stacked vector $O_2 = (X_B, X_2)$:

$$K_2 = \begin{pmatrix} K_2(X_B, X_B) & K_2(X_B, X_2) \\ K_2(X_2, X_B) & K_2(X_2, X_2) \end{pmatrix}.$$ 

Of course, if the kernels both accurately model the prior covariance of the process at the boundary then:

$$K_1(X_B, X_B) = K_2(X_B, X_B) = K_B.$$ 

So we condition both $K_1$ and $K_2$ on $K_B$ to yield $K_1^*$ and $K_2^*$ respectively:

$$K_1^*(X_1, X_2) = K_1(X_1, X_1) + G_1[K_B - K_1(X_B, X_B)]G_1^T,$$

$$K_2^*(X_2, X_2) = K_2(X_2, X_2) + G_2[K_B - K_2(X_B, X_B)]G_2^T,$$

where $G_1 = K_1(X_1, X_B)K_1(X_B, X_B)^{-1}$ and $G_2 = K_2(X_2, X_B)K_2(X_B, X_B)^{-1}$. The global prior covariance is then:

$$K = \begin{pmatrix} K_1^*(X_1, X_1) & K_1^*(X_1, X_B)D \\ K_1^*(X_1, X_B) & K_2^*(X_2, X_B) \\ D^T & K_2^*(X_2, X_B) \end{pmatrix}.$$ 

where the rows and columns correspond to the stacked vector $O = (X_1, X_B, X_2)$. The cross-terms, $D$, are:

$$D \triangleq \text{Cov}(f_2^*(X_1), f_2^*(X_2))$$

where $f_1^*$ and $f_2^*$ are the region function values conditioned on the function at the boundary:

$$f_1^*(X_1) = K_1(X_1, X_B)K_B^{-1}f(X_B),$$

$$f_2^*(X_2) = K_2(X_2, X_B)K_B^{-1}f(X_B).$$

Since $\text{Cov}(f(X_B), f(X_B)) = K_B$ then:

$$D = G_1K_BG_2^T.$$

As a corollary of this approach we can derive a Gaussian Process kernel for 1D signals with a change point at $x_B$:

**Corollary 1**: If $K_1$ and $K_2$ are two stationary GP kernels (not necessarily from the same family) which model region 1 and region 2 respectively, and $\theta_1$ and $\theta_2$ are their hyperparameters, then:

$$K(x_1, x_2; \theta_1, \theta_2) = \begin{cases} K_1(x_1, x_2; \theta_1) + g_1(x_1)[K_B - K_1(X_B, X_B; \theta_1)]g_1(x_2)^T & \text{if } x_1, x_2 < x_B \\ K_2(x_1, x_2; \theta_2) + g_2(x_1)[K_B - K_2(X_B, X_B; \theta_2)]g_2(x_2)^T & \text{if } x_1, x_2 \geq x_B \\ g_1(x_1)K_Bg_2(x_2)^T & \text{otherwise}. \end{cases}$$

where:

$$g_1(x_1) = K_1(x_1, X_B; \theta_1)K_1(X_B, X_B)^{-1},$$

$$g_2(x_2) = K_2(x_2, X_B; \theta_2)K_2(X_B, X_B; \theta_2)^{-1}.$$

To demonstrate the new kernel we return to the problem in Figure 2. Using identical hyperparameters and observations as in Figure 2 the function estimate obtained using the Markov Region Link approach is shown in Figure 6.

VI. DERIVATIVE BOUNDARY CONDITIONS

So far, we have developed covariance functions which preserve function continuity at the boundary. The approach can be extended to assert function derivative continuity at the boundary. The covariance between a function and any of its
derivatives can be determined from the GP kernel [7]. For example, the prior covariance between the function and its first derivative is:

\[ [\partial K(X, Y)]_{ij} \triangleq \text{Cov} \left( \frac{\partial f(x_i)}{\partial x_i}, f(x_j) \right) = \frac{\partial^2 K(x_i, x_j)}{\partial x_i \partial x_j} . \]

where \( x_i \in X \) and \( x_j \in Y \). The covariance between the derivatives is:

\[ [\partial \partial K(X, Y)]_{ij} \triangleq \text{Cov} \left( \frac{\partial f(x_i)}{\partial x_i}, \frac{\partial f(x_j)}{\partial x_j} \right) = \frac{\partial^2 K(x_i, x_j)}{\partial x_i \partial x_j} . \quad (9) \]

The derivative variance at \( x_i \) can be obtained by setting \( x_j \) to \( x_i \) in Eqn 9. In our notation \( \partial K(X, Y) \) denotes partial differentiation with respect to the first parameter, in this case \( X \) and \( \partial \partial K(X, Y) \) denotes double differentiation with respect to both \( X \) and then \( Y \).

These relationships can be used to define non-stationary GP covariance priors which impose continuous function derivatives at region boundaries. The prior mean and covariance for both the regional and global priors are augmented to include the function derivative. For example, if the first derivative is added to the prior then the prior covariances for regions \( R_1 \) and \( R_2 \) become: \(^2\)

\[
K_1' = \begin{pmatrix}
K_1(X_1, X_1) & K_1(X_1, X_B) & [\partial K_1(X_B, X_1)]^T \\
K_1(X_B, X_1) & K_1(X_B, X_B) & [\partial K_1(X_B, X_B)]^T \\
\partial K_1(X_B, X_1) & \partial K_1(X_B, X_B) & \partial^2 K_1(X_B, X_B)
\end{pmatrix}
\]

and:

\[
K_2' = \begin{pmatrix}
K_2(X_B, X_B) & [\partial K_2(X_B, X_B)]^T & K_2(X_B, X_2) \\
\partial K_2(X_B, X_B) & \partial^2 K_2(X_B, X_B) & \partial K_2(X_B, X_2) \\
[\partial K_2(X_B, X_B)]^T & [\partial^2 K(X_B, X_2)]^T & K_2(X_2, X_2)
\end{pmatrix}
\]

The rows and columns in \( K_1' \) correspond to the stacked vector \( O_1 = (X_1, X_B, D(X_B)) \) where \( D(X_B) \) denotes the function derivative at \( X_B \). Similarly, the rows and columns in \( K_2' \) correspond to the stacked vector \( O_2 = (X_B, D(X_B), X_2) \). We have defined the ordering in this way so that the prior covariances can be slotted into the global covariance prior whose rows and columns represent that stacked vector \( O = (X_1, X_B, D(X_B), X_2) \): the function at \( X_1 \); then the function at \( X_B \), then the function derivatives at \( X_B \) and finally the function at \( X_2 \). Consequently, we can use the approach outlined in Section V to construct a global prior for processes which are conditionally independent in each region. This is done by defining \( K_B \) as follows:

\[
K_B = \begin{pmatrix}
K_1(X_B, X_B) & [\partial K_1(X_B, X_B)]^T \\
\partial K_1(X_B, X_B) & \partial^2 K_1(X_B, X_B)
\end{pmatrix}
\]

and using \( K_1' \) and \( K_2' \) defined above in place of \( K_1 \) and \( K_2 \) in Section V.

If it is not desirable to retain estimates for the region boundary derivatives then the corresponding rows and columns can be deleted from the global prior mean and covariance.

Figure 7 shows the effect that the derivative constraint can have on the function estimate. Two stationary Gibbs kernels are used with \( l(x) = 35 \) for \( x \leq 130 \) and \( l(x) = 15 \) for \( x > 130 \) as in Figure 2. Clearly, the approach which imposes a continuous first derivative on the GP model produces a tighter function estimate at \( x = 130 \).

VII. APPLICATION 1: TARGET ESTIMATION WITH FAULTY SENSORS

We shall use a GP to estimate a target’s position over time \( t \) as it is tracked by a simple sensor. However, the sensor is faulty and outputs readings which have either drifted gradually from the truth or undergone a sudden jolt away from the truth resulting in a fixed bias in the reading (see Figure 1).

The proposed algorithm operates on-line and infers a posterior distribution for the current target’s position using observations of its previous locations. Smoothing from future observations is not considered.

The target’s trajectory is described by the process \( f \) and the sensor is subject to occasional faults. The sensor’s fault process \( e \) can be either a short term fixed bias or it can drift over a period of time. The, possibly faulty, observation at time \( t_i \) is:

\[
y_i = f(t_i) + e(t_i) + \epsilon_i
\]

where \( \epsilon_i \) is zero mean, Gaussian with variance \( \sigma^2 = 0.001 \). We wish to estimate target location \( f(t) \) over time.

The processes \( f \) and \( e \) are described by GP kernels \( K_f \) and \( K_e \). We will assume that \( K_f \) is stationary and we will use a simple squared exponential kernel to model the target dynamics. However, the fault is intermittent and it starts at time \( t = T_0 \) and ends at \( t = T_1 \). We model the fault process

\(^2\)We shall use prime to denote the augmented covariances.
using a non-stationary kernel. Firstly, \( e(t) \) is zero over times, \( t < T_0 \) and \( t > T_1 \), for which there is no fault. For a bias fault, we assume that the bias is a fixed offset and thus assert:

\[
K_{\text{bias}}(t_i, t_j) = \mu \text{ for all } T_0 \leq t_i, t_j \leq T_1.
\]

where \( \mu \) is a scale parameter representing the magnitude of the bias. We assume that the drift is gradual and describe it via a squared exponential kernel:

\[
K_{\text{drift}}(t_i, t_j) = \mu \exp \left(-\frac{(t_i - t_j)^2}{L^2}\right)
\]

where \( \mu \) and \( L \) are scale and length parameters, respectively, and again, \( T_0 \leq t_i, t_j \leq T_1 \). For simplicity, the scale parameters are assumed known. However, the time parameters and fault type \( ft \in \{\text{bias, drift}\} \) are inferred using a simple Bayesian multi-hypothesis approach which is outlined later in this section.

The bias fault causes a jump in the observation sequence when the fault sets in at \( t = T_0 \). The drift fault is gradual and \( e(T) \) is zero at \( t = T_0 \) thus causing the combined process \( f(t) + e(t) \) to be continuous (see Figure 1).

Thus, for the drift fault, \( e(t) \) is continuous at \( T_0 \), discontinuous at \( T_1 \) when the fault disappears and the rate of change of \( e \) will be discontinuous at both \( T_0 \) and \( T_1 \). We use the Markov Region Link kernel to construct the fault process prior covariance \( K_{\text{drift}}^* \) from \( K_{\text{drift}} \) and impose the continuity boundary condition at \( T_0 \). Using the approach set out in Section V the prior covariance for the drift fault becomes the block matrix:

\[
K_{\text{drift}}^* = \begin{pmatrix}
0 & 0 & 0
0 & K_{\text{drift}}^*(X_f, T_0) & 0
0 & 0 & 0
\end{pmatrix}
\]

The first row and column are zero matrices for times less than \( T_0 \), corresponding to the period before the fault starts. The last row and column are zero matrices for times greater than \( T_1 \), corresponding to times after the fault has stopped. The central row and column blocks are prior covariances over time samples \( X_f \) during which the sensor is faulty:

\[
X_f = \{t \mid T_0 < t \leq T_1\}.
\]

Continuity of the fault process at \( T_0 \) imposes \( K_{\text{drift}}^*(T_0, T_0) = 0 \). Values for \( K_{\text{drift}}^* \) are obtained using Corollary 1 in Section V with \( X_B = T_0 \), \( K_1 = 0 \), \( K_B = K_{\text{drift}}^*(T_0, T_0) = 0 \) and \( K_2 = K_{\text{drift}} \).

The bias kernel is more straightforward:

\[
K_{\text{bias}} = \begin{pmatrix}
0 & 0 & 0 & 0
0 & \mu & 0 & 0
0 & \mu & \mu & 0
0 & 0 & 0 & 0
\end{pmatrix}
\]

with the rows and columns interpreted as for \( K_{\text{drift}}^* \).

We assume that the sensor fault \( e \) and target trajectory \( f \) processes are independent. We note that the trajectory process, \( f \), is hidden and thus we use the following slightly modified GP equations to infer the individual processes:

\[
\hat{f}_e = K_f(X_e, X)K_f(X_e, X) + K_e(X_e, X) + \sigma^2 I^{-1}Y
\]

\[
\text{Cov}(\hat{f}_e) = K_f(X_e, X) - K_f(X_e, X)K_e(X_e, X)X \times [K_f(X_e) + K_e(X_e) + \sigma^2 I]^{-1}K_f(X_e, X)^T
\]

where \( K_e = K_{\text{bias}}^* \) or \( K_e = K_{\text{bias}}^* \).

A distribution over the parameters \( \theta = \{ft, T_0, T_1\} \) is determined using the procedure outlined in Section I. The likelihood function used in 4 is:

\[
p(Y \mid X, \theta) = \mathcal{N}(Y; 0, K_f(X, X) + K_e(X, X) + \sigma^2 I)
\]

where \( \mathcal{N} \) is the multi-variate normal distribution. The hyper-parameters are marginalised using Monte-Carlo integration.

Figure 8 shows typical tracks, observations and GP track estimates. The target trajectory and observation sequence are randomly generated, \( f \sim \mathcal{N}(0, K_f) \) and \( y \sim \mathcal{N}(0, K_e + \sigma^2 I) \). Notice that the algorithm has successfully corrected for the faulty observations.

Parallel faults may also be modelled using our approach. A parallel fault comprises more than one basic fault occurring at the same time. In the following example the sensor measurement drifts to a new fixed bias offset. The fault can be modelled using a drift fault, without fault correction, followed immediately by a biased fault.

A SICK laser range sensor is used to track a person. The sensor is subject to a knock as it tracks the person. This knock results in the sensor drifting for a period before coming to rest. The sensor data thus exhibits a drift followed by a constant bias. Figure 9 shows a typical 180 bearing range scan at a single time instance during the tracking procedure. Figure 10 shows the data obtained from our sensor (transformed to the laboratory centred Cartesian coordinates) and also the target’s true trajectory obtained by extrapolating the SICK data using stationary objects in the environment.

We use the MRL to splice together both the drift kernel and the bias kernel and assert that the target observations are continuous at the transition from drift to bias. Both the transition time and the kernel induced variance, \( K_B \), at that time are hyperparameters of our model.

Figure 11 shows the ground truth and the one standard deviation estimate if no fault is assumed as the person moves from right to left. The magnitude of the error is underestimated.

Figure 12 shows the GP on-line estimated trajectory as well as an estimate of the fault. Both estimates are correctly determined in this case.
VIII. APPLICATION 2: EOG ARTIFACT REMOVAL FROM EEG SIGNALS

In this example we use our approach to recover from a drift type fault, specifically to tracking EEG signals and detecting and removing EOG artifacts. The recovery of the EEG signal is often treated as a blind source separation problem [8] where ICA identifies the separate artifact free EEG signal (which we refer to as EEG*) and the EOG signal. We propose an alternative approach which uses mixtures of Gaussian Processes. Our approach allows us to encode any information we have about the shape of the component signals including signal smoothness and continuity at change points. We explicitly encode models for the EEG* and EOG signal and explicitly stipulate that these signals are independent.

The “observed” EEG signal, $y$, is a mixture of EEG* and EOG artifact signals. The EEG* signal, $s_{\text{eeg}*}$, is modelled as the combination of a smooth function $m_{\text{eeg}*}$ (generated by a GP with prior covariance $K_{\text{eeg}*}$) and residuals $r_{\text{eeg}*}$. The EOG artifact, $s_{\text{eog}}$, is modelled as a piece-wise smooth function, $m_{\text{eog}}$ (generated from a GP “fault” model with prior covariance $K_{\text{eog}}$) and, again, with residuals $r_{\text{eog}}$.

$$s_{\text{eeg}*} = m_{\text{eeg}*} + r_{\text{eeg}*} ,$$
$$s_{\text{eog}} = m_{\text{eog}} + r_{\text{eog}} ,$$
$$y = s_{\text{eeg}*} + s_{\text{eog}} .$$

where $m_{\text{eeg}*} \sim \mathcal{N}(0, K_{\text{eeg}*})$, $r_{\text{eeg}*} \sim \mathcal{N}(0, R_{\text{eeg}*} I)$, $m_{\text{eog}} \sim \mathcal{N}(0, K_{\text{eog}})$ and $r_{\text{eog}} \sim \mathcal{N}(0, R_{\text{eog}} I)$. The random vectors $m_{\text{eeg}*}$, $m_{\text{eog}}$, $r_{\text{eeg}*}$ and $r_{\text{eog}}$ are assumed to be mutually independent. The residuals, $r_{\text{eeg}*}$ and $r_{\text{eog}}$, are considered to be part of the signals and are therefore not treated as noise and are not filtered out.

We use a simple squared exponential to model the EEG* signal. As for the EOG model, typically the EOG signal is zero everywhere except within a small time window. Within this window the EOG artifact can be modelled as two smooth functions (not necessarily monotonic) which join at a spike near the...
centre of the window. Thus, the EOG’s prior covariance $K_{eog}$ is chosen to be zero everywhere except between the artifacts start and end times, $T_s$ and $T_e$. We use the methods outlined in Section V to build the EOG artifact prior covariance. Between the start and end times the EOG artifact signal is modelled by two piece-wise squared exponential kernels joined mid-way between $T_s$ and $T_e$ so that they are continuous at the midpoint and also at $T_s$ and $T_e$.

The following GP equations determine the mean and covariance for the hidden variable, $m_{eegs}$:

$$m_{eegs}(x_s) = [K_{eegs}(x_s, X)] \times [K_{eegs}(X, X) + K_{eog}(X, X) + \sigma^2 I]^{-1} y(X)$$

$$\text{Cov}_{eegs}(x_s) = K_{eegs}(x_s, x_s) - K_{eegs}(x_s, X) \times [K_{eegs}(X, X) + K_{eog}(X, X) + \sigma^2 I]^{-1} \times K_{eegs}(x_s, X)^T$$

where $\sigma^2 = R_{eegs} + R_{eog}$. Similar expressions can be obtained for $m_{eog}$.

To track the EEG signal our algorithm determines $s_{eegs}$ sequentially over time. When $x_s$ is the current time and $X$ is the previous times at which data points were obtained then:

$$p(s_{eegs}(x_s), s_{eog}(x_s) | y(x_s), \hat{m}_{eegs}(x_s), \hat{m}_{eog}(x_s))$$

$$\propto p(y(x_s) | s_{eegs}(x_s), s_{eog}(x_s), \hat{m}_{eegs}(x_s), \hat{m}_{eog}(x_s))$$

$$\times p(s_{eegs}(x_s), s_{eog}(x_s) | \hat{m}_{eegs}(x_s), \hat{m}_{eog}(x_s))$$

$$\propto \delta(y(x_s), s_{eegs}(x_s) + s_{eog}(x_s))$$

$$\times p(s_{eegs}(x_s) | \hat{m}_{eegs}(x_s))$$

$$\times p(s_{eog}(x_s) | \hat{m}_{eog}(x_s)).$$

Marginalising $s_{eog}$:

$$p(s_{eegs}(x_s) | y(x_s), \hat{m}_{eegs}(x_s), \hat{m}_{eog}(x_s))$$

$$\propto p(y(x_s) - s_{eegs}(x_s) | \hat{m}_{eog}(x_s))$$

$$\times p(s_{eegs}(x_s) | \hat{m}_{eegs}(x_s)) .$$

In general, when $s$ is Gaussian distributed then its mean, $\hat{s}$, is the solution to:

$$\frac{\partial \log p(s | \cdot)}{\partial s} = 0$$

and its variance, $\text{Var}(s)$, is given by:

$$\text{Var}(s) = -E \left[ \frac{\partial^2 \log p(s | \cdot)}{\partial s^2} \right]^{-1} .$$

Thus, defining $P_{eegs}^*(x_s) \triangleq \text{Cov}_{eegs}(x_s) + R_{eegs}$ and $P_{eog}^*(x_s) \triangleq \text{Cov}_{eog}(x_s) + R_{eog}$:

$$\hat{s}_{eegs}(x_s) = \frac{P_{eegs}^*(x_s)(y(x_s) - \hat{m}_{eog}(x_s)) + P_{eog}^*(x_s)\hat{m}_{eegs}(x_s)}{P_{eegs}^*(x_s) + P_{eog}^*(x_s)}$$

and:

$$\text{Var}_{eegs}(x_s) = \frac{P_{eegs}^*(x_s) + P_{eog}^*(x_s)}{P_{eegs}^*(x_s)P_{eog}^*(x_s)} .$$

Similar reasoning leads us to similar expressions for the EOG artifact signal:

$$\hat{s}_{eog}(x_s) = \frac{P_{eog}^*(x_s)(y(x_s) - \hat{m}_{eegs}(x_s)) + P_{eegs}^*(x_s)\hat{m}_{eog}(x_s)}{P_{eegs}^*(x_s) + P_{eog}^*(x_s)}$$

and:

$$\text{Var}_{eog}(x_s) = \frac{P_{eegs}^*(x_s) + P_{eog}^*(x_s)}{P_{eegs}^*(x_s)P_{eog}^*(x_s)} .$$

These expressions for $\hat{s}_{eegs}$ and $\hat{s}_{eog}$ determine the proportion of the EEG signal residual that is assigned to the EOG artifact signal and also to the artifact free EEG signal ($\text{EEG}^*$).

Our model requires eight hyperparameters, collectively referred to as $\theta$; the scale heights and scale lengths for the GP models (we assume that both parts of the EEG model have the same scale heights and lengths); the artifact start and end times and also the residual variances $R_{eegs}$ and $R_{eog}$. The likelihood used in Equation 4 to determine a distribution over the hyperparameter values is given by:

$$p(y(x_s) \mid \theta) = \mathcal{N}[y(x_s); \hat{m}_{eegs, \theta}(x_s), \hat{m}_{eog, \theta}(x_s), P_{eegs, \theta}^*(x_s) + P_{eog, \theta}^*(x_s)].$$

Again, the hyperparameters are marginalised using Monte-Carlo sampling.

Figure 13 shows a typical EEG signal which is corrupted by EOG artifacts. It also shows the one standard error confidence interval for the artifact free EEG* signal and the EEG artifact obtained using our algorithm. Figure 14 shows the mean difference between the original EEG signal and the inferred EEG* signal, indicating the expected proportion of the original signal that is retained in the EEG*.

![Fig. 13. EEG signal (crosses) and 1 standard error confidence intervals for the EEG* (left panel) and EOG (right panel) signals obtained using the GP mixture model approach.](image)

IX. Conclusions

This paper has presented an approach to building piece-wise stationary prior covariance matrices from stationary Gaussian Process kernels. Where appropriate the approach asserts function continuity or continuity of any function derivative at the region boundaries. The approach has been successfully demonstrated on sensor fault detection and recovery and also on EEG signal tracking.
Fig. 14. Original EEG signal (dots) and difference (line) between original signal and the mean EEG* obtained using the GP mixture model approach.

REFERENCES


