Statistical inverse identification for nonlinear train dynamics using a surrogate model in a Bayesian framework

David Lebel, Christian Soize, Christine Fünfschilling, Guillaume Perrin

Statistical inverse identification for nonlinear train dynamics using a surrogate model in a Bayesian framework

D. Lebel[a,b], C. Soize[a], C. Fünfschilling[b], G. Perrin[c]

[a] *Université Paris-Est Marne-la-Vallée, MSME, UMR 8208 CNRS, 5 Bd Descartes 77454 Marne-la-Vallée Cedex 2, France*
[b] *SNCF Innovation & Recherche, 40 Av des Terroirs de France, 75012 Paris, France*
[c] *CEA DAM DIF, Bruyères-le-Chatel, 91297 Arpajon Cedex, France*

Abstract

This paper presents a Bayesian calibration method for a simulation-based model with stochastic functional input and output. The originality of the method lies in an adaptation involving the representation of the likelihood function by a Gaussian process surrogate model, to cope with the high computational cost of the simulation, while avoiding the surrogate modeling of the functional output. The adaptation focuses on taking into account the uncertainty introduced by the use of a surrogate model when estimating the parameters posterior probability distribution by MCMC. To this end, trajectories of the random surrogate model of the likelihood function are drawn and injected in the MCMC algorithm. An application on a train suspension monitoring case is presented.

Keywords:

1. Introduction

The goal of this work is to perform the inverse identification of the input parameters of an expensive computer code with functional input excitation and output response, under uncertainty, from experimental data. The industrial application motivating the work presented in this paper consists in monitoring high-speed train suspensions using acceleration measurements for maintenance purposes.

This identification problem can be seen as a calibration problem. The main purpose of calibration is to fit a model to experimental data, in order to provide a good

prediction for untested conditions. As long as the quality of the prediction is satisfying, the calibrated value of the parameters may not correspond to the ”real” physical value, in the case when the parameters have a physical meaning. The problem studied in this paper differs in this point because the ”real” physical value of the parameters is precisely what we are looking for.

Because the problem is affected by various sources of uncertainty (noise on the measurements, model error, lack of knowledge about the model parameters), a probabilistic model of the system response is built based on the simulation code. The Bayesian framework is well suited to combine a probabilistic model with experimental data to obtain information about the model input parameters. A Bayesian calibration approach is adopted.

However, except a few particular analytical cases, Bayesian calibration methods generally require the computation of the model response for numerous values of the input parameters. Surrogate modeling is a solution to circumvent the numerical cost induced by the numerous calls to an expensive computer code. It consists in replacing the latter by an efficient algebraic approximation of the output response. Although it is a classical approach for simulations with scalar output, surrogate modeling of functional output remains a complex task. Because we want to keep the whole information provided by the measurements, we ruled out the possibility of condensing the output into a few indicators of interest. The solution we propose in this paper is, instead of relying on a surrogate model of the output response, building a surrogate model of the likelihood function that is at the core of Bayesian calibration. The novelty of this work also lies in the consideration of the uncertainty introduced by the surrogate model.

We start this paper by recalling in Section 2 the main features of MCMC-based Bayesian calibration, followed by two examples of calibration procedures. The approach we propose is then detailed in Section 3, with a particular focus put on including the surrogate model uncertainty in the calibration procedure. The method is then applied on our industrial railway case in Section 4.

2. Setting the calibration problem in a Bayesian framework

This section recalls the main principles of Bayesian calibration and introduces the formalism that is used in the rest of the paper. Two calibration methods for cases approaching the one treated in this paper are also presented.

2.1. Bayesian calibration principle

We consider a system described by the model $\mathbf{Y} = \mathbf{H}(\mathbf{W})$. Variable $\mathbf{W}$ gathers the various input parameters of the model. In general, $\mathbf{W}$ can be defined as a vector belonging to an admissible set $\mathcal{C}_{\mathbf{W}}$. In the context of Bayesian calibration, the value of these parameters is uncertain, hence vector $\mathbf{W}$ is random. The initial information about the parameters is given by the prior probability density function (PDF) $p_{\mathbf{W}}^{\text{prior}}$ of $\mathbf{W}$. Variable $\mathbf{Y}$ is an observable output quantity of the system. The dependence of $\mathbf{Y}$ to parameters $\mathbf{W}$ is modeled by the function $\mathbf{H}$. Function $\mathbf{H}$ is considered random (meaning that $\mathbf{H}(\mathbf{w}_0)$ is a random value even if $\mathbf{w}_0$ is deterministic), which makes $\mathbf{Y}$ random as well. Various sources of uncertainty may account for $\mathbf{H}$ randomness: model error, approximation errors... A measurement $\mathbf{y}^{\text{mes}}$ is provided as experimental data that can be affected by measurement noise. It is considered as a realization of random quantity $\mathbf{Y}$.

The goal of the Bayesian calibration procedure is to determine the posterior PDF $p_{\mathbf{W}}^{\text{post}}$ of parameters $\mathbf{W}$. The latter represents the updated knowledge about the parameters, according to the new information about the system brought by measurement $\mathbf{y}^{\text{mes}}$. Mathematically, $p_{\mathbf{W}}^{\text{post}}$ corresponds to the conditional PDF of $\mathbf{W}$ knowing $\mathbf{Y}$

$$p_{\mathbf{W}}^{\text{post}}(\mathbf{w}) = p_{\mathbf{W}\,|\,\mathbf{Y}}(\mathbf{w}\,|\,\mathbf{y}^{\text{mes}}),\ \mathbf{w} \in \mathcal{C}_{\mathbf{W}}. \tag{1}$$

According to the Bayes formula, the conditional probability can then be decomposed in the following way:

$$p_{\mathbf{W}}^{\text{post}}(\mathbf{w}) = \frac{p_{\mathbf{Y}\,|\,\mathbf{W}}(\mathbf{y}^{\text{mes}}\,|\,\mathbf{w})\, p_{\mathbf{W}}^{\text{prior}}(\mathbf{w})}{p_{\mathbf{Y}}(\mathbf{y}^{\text{mes}})} \tag{2}$$

$$\propto \mathcal{L}(\mathbf{w})\, p_{\mathbf{W}}^{\text{prior}}(\mathbf{w}) \tag{3}$$

where, for a fixed $\mathbf{y}^{\text{mes}}$, the function $\mathcal{L} : \mathbf{w} \mapsto p_{\mathbf{Y}\,|\,\mathbf{W}}(\mathbf{y}^{\text{mes}}\,|\,\mathbf{w})$ is called the likelihood function. Its computation depends on the relationship $\mathbf{Y} = \mathbf{H}(\mathbf{W})$. Because

of dimensionality problems, the log-likelihood $L : \mathbf{w} \mapsto \log(\mathcal{L}(\mathbf{w}))$ is usually used instead.

From Eq. (3), Markov Chain Monte Carlo (MCMC) is a classical way to construct independent realization of posterior PDF $p_{\mathbf{W}}^{\mathrm{post}}$. This class of algorithms is briefly presented in the following section. It should be noted that calibration methods that do not rely on MCMC have been developed, in order to avoid the numerous difficulties associated with the implementation of MCMC algorithms. Other approaches for constructing surrogate models consist in using polynomial chaos expansions [1, 2, 3, 4].

2.2. Estimation of the posterior density with MCMC

The purpose of Markov Chain Monte Carlo (MCMC, see [5]) is to draw a sample according to a target PDF. It requires the ability to evaluate this PDF or a function proportional to it anywhere in its definition set. MCMC algorithms perform this draw by building a Markov Chain for which the invariant probability distribution is represented by the target PDF. For Bayesian calibration, the target PDF is posterior PDF $p_{\mathbf{W}}^{\mathrm{post}}$; the function injected in the algorithm is $\mathbf{w} \mapsto \mathcal{L}(\mathbf{w})\, p_{\mathbf{W}}^{\mathrm{prior}}(\mathbf{w})$.

The drawn sample usually needs to be large in order to be distributed as the target PDF. At least one call to the likelihood function is required for each point of the sample. Once the sample is drawn, the target PDF can be studied by estimating its moments and quantiles for instance. The marginal PDFs can also be plotted using histograms or kernel methods.

2.3. Calibration of an expensive computer code with scalar output

Let us first consider the case of a simulation-based model with scalar output, represented for example by the relationship

$$Y = h(\mathbf{W}) + \varepsilon \tag{4}$$

where the simulation is represented by the deterministic function h and ε that is an additive noise representing the measurement noise and the model error, modeled by a Gaussian centered random variable of variance σ_ε^2. For a single measurement y^{mes}, the corresponding likelihood function is

$$\mathcal{L}(\mathbf{w}) = p_{\mathcal{N}}(y^{\mathrm{mes}}; h(\mathbf{w}), \sigma_\varepsilon^2), \tag{5}$$

where $p_{\mathcal{N}}(.;\mu,\sigma^2)$ stands for the normal density of mean μ and variance σ^2.

Using MCMC requires numerous evaluations of the likelihood function $\mathcal{L}$ and consequently numerous runs of the simulation represented by function h. An expensive simulation makes the procedure unaffordable. A classical way of addressing such numerical issue is to rely on surrogate models. In particular, Gaussian process (GP) models is a commonly used class of surrogate models, as it offers closed-form expressions and an estimation of the approximation error. The principle of GP surrogate modeling is to represent a deterministic function by a conditioned Gaussian process. The mean function of the process (also called the Kriging predictor) constitutes the best approximation of the function anywhere in its definition set in a L^2-sense, while its variance represents the approximation error. More details, formulas and their mathematical justifications are provided in Appendix A.

For the present calibration case, the simulation is replaced by a GP surrogate model: for any $\mathbf{w} \in \mathcal{C}_{\mathbf{w}}$, the simulation output $h(\mathbf{w})$ is replaced by the random value $\hat{h}(\mathbf{w}) + Z(\mathbf{w})$ where $\hat{h}$ is the Kriging predictor and Z a centered GP of variance σ_Z^2 independent from noise ε. The model then becomes

$$Y = \hat{h}(\mathbf{W}) + Z(\mathbf{W}) + \varepsilon \tag{6}$$

and the associated likelihood function

$$\mathcal{L}(\mathbf{w}) = p_{\mathcal{N}}(y^{\mathrm{mes}}; \hat{h}(\mathbf{w}), \sigma_Z^2(\mathbf{w}) + \sigma_\varepsilon^2), \tag{7}$$

As shown by this last equation, the interest of building a GP surrogate model of the simulation output is that the surrogate model uncertainty can be readily introduced in the likelihood function. The calibration procedure is not modified. The Bayesian formalism is kept, which allows for a correct evaluation of the posterior uncertainty on the parameters $\mathbf{W}$.

2.4. Calibration of an expensive computer code with functional output

We now consider a simulation-based model with an output which is no longer scalar but functional. The method presented in the previous section can no longer be applied

as such. Indeed, the surrogate modeling of functional quantities is a complex task and remains a subject of current research (see for example [6]).

For the calibration of expensive computer codes, the issue of functional outputs is addressed in [7] and [8]. In both papers, the authors perform a scalarization of the problem. They define a distance between the experimental and modeled output, based on a L^2-norm for [7] or on likelihood ratios for [8]. The distance being scalar, it can be represented by a GP surrogate model. The calibration procedure then consists in an optimization problem. The GP model is used to minimize the defined distance. The goal is to obtain the optimal parameters values that provide the best fit between the experimental data and the model, according to the chosen distance.

Although both methods achieve interesting results, they cannot be considered as Bayesian approaches. They do not take into account the uncertainties of the model to determine an uncertainty on the calibrated parameters. In [8], the authors propose to estimate the distribution of the minimum of the GP model. This analysis allows for evaluating the uncertainty on the parameters stemming from the approximation by a surrogate model, but still ignores the other sources of uncertainty, such as model error or measurement noise.

3. Calibration with GP surrogate model of the likelihood function

The goal of this paper is the development of a Bayesian calibration method for expensive computer codes with functional output. As shown in [7] and [8], scalarization combined with GP surrogate modeling is an efficient way of addressing high computational costs caused by a simulation-based model. In the Bayesian formalism, a natural scalarization is provided by the likelihood function. Hence we propose to perform the calibration using a GP surrogate model of the likelihood function. The objective is to remain in the Bayesian framework while taking advantage of GP surrogate modeling. This approach raises various questions that will be addressed in this paper:

- how should a GP model of the likelihood function be built ?
- how to perform MCMC with a random likelihood function ?

- how to take into account the uncertainty associated with the approximation error in the estimation of the parameters posterior probability distribution ?

3.1. GP surrogate modeling of the likelihood function

We choose to work with the log-likelihood function L instead of the likelihood function $\mathcal{L}$. The first reason is the fact that the likelihood function has to respect positivity, while the log-likelihood function does not. Moreover, in some cases, the regularity of the log-likelihood function tends to be better, when the likelihood function corresponds to a very peaked density for instance.

As explained in Appendix A, the computation of the real likelihood value on an initial training set is necessary to build the surrogate problem. This computation depends on the problem and the function $\mathbf{H}$ chosen to model the output $\mathbf{Y}$. It consists in a probability calculation for a stochastic process that may become complex if $\mathbf{H}$ defines a complex distribution.

The GP surrogate model of L is denoted as $L(.;\Theta)$. Random variable Θ explicitly accounts for the randomness of the surrogate model. At any point $\mathbf{w} \in \mathcal{C}_{\mathbf{W}}$, the Kriging predictor is denoted as the expectation $E_\Theta\{L(\mathbf{w};\Theta)\}$ and constitutes the best approximation of $L(\mathbf{w})$ in a L^2-sense. The variance $\mathrm{Var}_\Theta\{L(\mathbf{w};\Theta)\}$ quantifies the approximation error of the Kriging predictor. If θ is a realization of Θ, then $L(.;\theta)$ represents a deterministic trajectory of the GP model.

MCMC must be applied on a deterministic likelihood function. The GP surrogate model being random, it cannot be used directly as such. To perform the posterior PDF estimation with MCMC, the most straightforward solution is to use the Kriging predictor $E_\Theta\{L(.;\Theta)\}$ in place of L. In Section 4.5, we refer to this approach as the KP (for "Kriging predictor") method. Once the GP model is built, this solution is easy to implement. It can provide useful results, especially if only the most probable value or the mean value of the system parameters are needed. However, this solution does completely ignore the uncertainty introduced by the use of a surrogate model, which represents the approximation error intrinsic in such modeling. The uncertainty on the calibrated parameters would only stem from the uncertainties considered in the model $\mathbf{H}$.

In the following sections, we propose a method that takes into account the uncertainty of the GP surrogate model in order to estimate more correctly the calibration error. In Section 4.5, we refer to this new approach as the MCT (for "Monte Carlo on the trajectories") method.

3.2. Monte Carlo on the GP model trajectories

If the deterministic log-likelihood function L is replaced by the random surrogate model $L(.;\Theta)$, the corresponding posterior PDF becomes random as well. Random variable Θ accounts for the randomness of the surrogate model; it is independent of the other "physical" random quantities involved in the calibration problem, such as the system parameters $\mathbf{W}$ or the system response $\mathbf{Y}$. The random posterior PDF corresponding to $L(.;\Theta)$ can be expressed using the conditioning by Θ as

$$\mathbf{w} \mapsto p_{\mathbf{W}\,|\,\mathbf{Y},\Theta}(\mathbf{w}\,|\,\mathbf{y}^{\mathrm{mes}},\Theta)\,. \tag{8}$$

Yet, the quantity we are looking for remains unchanged: the conditional PDF of parameters $\mathbf{W}$ knowing $\mathbf{Y}$. The rule of conditional probabilities states that it is equal to the expectation with respect to Θ of the previous PDF:

$$\begin{aligned} p^{\mathrm{post}}_{\mathbf{W}}(\mathbf{w}) &= p_{\mathbf{W}\,|\,\mathbf{Y}}(\mathbf{w}\,|\,\mathbf{y}^{\mathrm{mes}}) && (9)\\ &= E_{\Theta}\left\{p_{\mathbf{W}\,|\,\mathbf{Y},\Theta}(\mathbf{w}\,|\,\mathbf{y}^{\mathrm{mes}},\Theta)\right\} && (10)\\ &\approx \frac{1}{N}\sum_{i=1}^{N} p_{\mathbf{W}\,|\,\mathbf{Y},\Theta}(\mathbf{w}\,|\,\mathbf{y}^{\mathrm{mes}},\theta_i). && (11) \end{aligned}$$

Eq. (11) corresponds to the empirical estimate of the expectation using N realizations $\{\theta_i\}_{1\leq i\leq N}$ of random variable Θ. This Monte Carlo approach requires the estimation of N PDF $p_{\mathbf{W}\,|\,\mathbf{Y},\Theta}(\mathbf{w}\,|\,\mathbf{y}^{\mathrm{mes}},\theta_i)$ by MCMC using N deterministic trajectories $L(.;\theta_i)$ of surrogate model $L(.;\Theta)$.

In practice, for each i, the MCMC algorithm provides a sample, subset of $\mathcal{C}_{\mathbf{W}}$, distributed as $p_{\mathbf{W}\,|\,\mathbf{Y},\Theta}(\mathbf{w}\,|\,\mathbf{y}^{\mathrm{mes}},\theta_i)$. If, for all i, these samples have the same number of points, their simple concatenation corresponds to a sample approximately distributed as $p^{\mathrm{post}}_{\mathbf{W}}$.

3.3. Approximation of the GP model trajectories

The Monte Carlo approach proposed in the previous section implies the ability to draw a trajectory $L(.;\theta)$ of the Gaussian surrogate model and to evaluate its value at any point $\mathbf{w}$ in admissible set $\mathcal{C}_{\mathbf{W}}$. A classic way to manipulate a Gaussian process trajectory is to compute its value on a fine grid spanning the entire admissible set. However, this solution is not adapted to the calibration of more than two or three parameters, because the number of points on the grid increases exponentially with the dimension of the admissible set, equal to the number of parameters to calibrate. Moreover, since the MCMC algorithm randomly draws the points at which the likelihood needs to be evaluated, computing the values of a trajectory in advance is not relevant. Another approach could consist in iteratively conditioning the Gaussian process by the previously drawn points for each step of the MCMC. However, it implies the manipulation of a full covariance matrix whose size would increase at each step.

Instead of exactly computing the trajectory, we propose to approximate it in the following way: first, the value of trajectory $L(.;\theta)$ is computed at the points of a conditioning set $\mathcal{W}^{\mathrm{c}} = \{\mathbf{w}_j^{\mathrm{c}}\}_{1\leq j\leq N_{\mathrm{c}}}$. It should contain only a limited number N_{c} of points. The approximation $\widetilde{L}(.;\theta)$ of $L(.;\theta)$ then consists of the expectation of the surrogate model conditioned by the value of the trajectory at the points of $\mathcal{W}^{\mathrm{c}}$:

$$\widetilde{L}(.;\theta) : \mathbf{w} \mapsto E_{\Theta}\left\{L(\mathbf{w};\Theta) \,|\, L(\mathbf{w}_j^{\mathrm{c}};\Theta) = L(\mathbf{w}_j^{\mathrm{c}};\theta), 1 \leq j \leq N_{\mathrm{c}}\right\} . \qquad (12)$$

The deterministic function $\widetilde{L}(.;\theta)$ can then be injected in the MCMC algorithm to estimate PDF $p_{\mathbf{W}\,|\,\mathbf{Y},\Theta}(.\,|\,\mathbf{y}^{\mathrm{mes}},\theta)$.

The crucial step of the method lies in the choice of conditioning set $\mathcal{W}^{\mathrm{c}}$. The aim of this conditioning approach is to decrease as much as possible the variance of the surrogate model. Indeed, the lower the variance of a stochastic process, the closer any trajectory is to the mean function. We chose to focus on the region of interest of $\mathcal{C}_{\mathbf{W}}$, where the relative value of the likelihood function is high. We define this region as the set $\mathcal{P}$ where the surrogate model has a probability higher than a tolerance $\rho \in]0,1[$ to be greater than at the point where its mean function is maximum:

$$\mathcal{P} = \{\mathbf{w} \in \mathcal{C}_{\mathbf{W}} \,|\, P\left(L(\mathbf{w};\Theta) > L(\mathbf{w}^{\max};\Theta)\right) \geq \rho\} \qquad (13)$$

with $\mathbf{w}^{\max} = \underset{\mathbf{w}\in\mathcal{C}_{\mathbf{W}}}{\arg\max}\, E_{\Theta}\{L(\mathbf{w};\Theta)\}$.

Conditioning set $\mathcal{W}^{\mathrm{c}}$ is then built as a discrete set of N_{c} points space-filling in $\mathcal{P}$. First a large sample uniformly distributed in $\mathcal{P}$ is drawn using MCMC with the indicator function of $\mathcal{P}$. Then several subsets of size N_{c} are randomly drawn in this sample. The "most space-filling" subset according to the maximin criterion defined in Appendix A.6 is kept as $\mathcal{W}^{\mathrm{c}}$.

4. Application: State health monitoring of high-speed train suspensions

4.1. Industrial context and objectives

High-speed trains dynamic behavior strongly relies on suspensions that ensure the train stability and thus the ride safety. The suspensions also filter most of the vibrations for passengers comfort. Because the suspensions undergo damage throughout their lifetime, regular maintenance is required. Maintenance is however performed without having access to the real state of the suspensions, especially its (potentially degraded) mechanical properties. Presently, it mostly relies on visual inspection and age or mileage criteria. A monitoring solution providing the actual suspensions health state would allow for implementing maintenance rules closer to the real needs. The work presented here is part of a project developing a remote diagnosis method for high-speed train suspensions. The method relies on on-track measurements of the train dynamic response performed thanks to embedded accelerometers. The objective of this work is the development of the mathematical method required to determine the suspensions health state from such measurements.

However, relying solely on measurements of the train dynamic response is not sufficient, because of its strong dependence on the track geometric irregularities. The latter constitute the main excitation source of a rolling train and, consequently, have a major influence on its dynamic behavior (see [9, 10, 11]). Track geometry is also subject to damage caused by railway traffic (see [12]): it is gradually degraded and regularly maintained. Consequently, the track geometric irregularities evolve through time. Because of the high sensitivity of the system to these irregularities, their evolution has to be taken into account to perform a correct monitoring of the suspensions

state. Train dynamics simulation is thus necessary to include the excitation source in the analysis. More precisely, we propose to build a simulation-based model of the train dynamic response and to calibrate its parameters describing the mechanical properties of the suspensions of interest. The experimental data used for this calibration consist of joint measurements of the track geometric irregularities and of the train dynamic response.

4.2. Probabilistic model of the high-speed train dynamic response

This section presents how the train response model is built, based on simulation. This construction is sketched in figure 1.

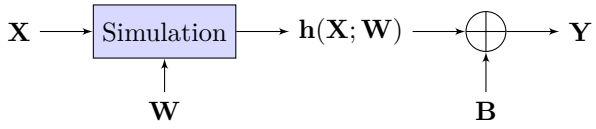

Figure 1: Diagram of the system quantities of interest

The railway dynamics simulation takes as input the track geometric irregularities $\mathbf{X}$. The model is parameterized by the q uncertain parameters $\mathbf{W}$ describing the mechanical properties of the monitored suspensions. The railway dynamics simulations are performed with the commercial multibody software *Vampire*, used as a black-box and represented by the deterministic mapping $\mathbf{h}$. To take into account the model errors and the measurement noise, a random error $\mathbf{B}$ is added to the simulation output $\mathbf{h}(\mathbf{X}; \mathbf{W})$ to obtain the train dynamic response model $\mathbf{Y}$. The following sections detail the characteristics of these various quantities.

4.2.1. Description of the system input: the track geometric irregularities

The track geometric irregularities consist of the small scale displacements of the rails with respect to the track design. For a track stretch of length S, they are denoted as $\{\mathbf{x}(s) \in \mathbb{R}^4, s \in [0, S]\}$. An illustration of the four irregularities is provided in figure 2. Combined with the track design and the train speed, they are used to compute the time-varying displacement condition imposed at the wheel-rail interface for each wheelset.

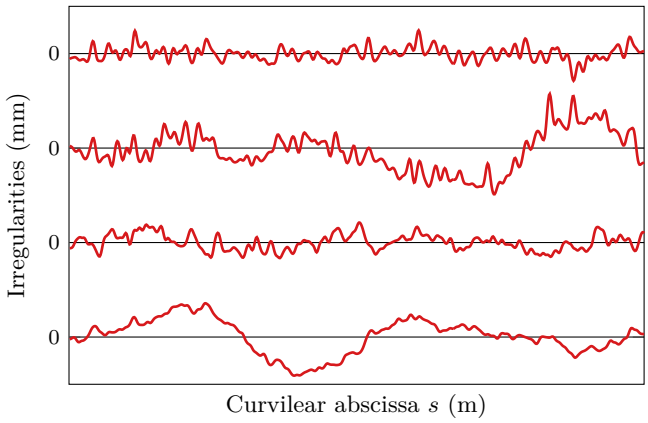

Figure 2: Track irregularities example. From bottom to top: lateral, gauge, vertical, cross-level irregularities.

It has been shown in [13] that the track geometric irregularities can be modeled as a nonstationary $\mathbb{R}^4$-valued random field $\{\mathbf{X}(s),\ s \in [0\,,S]\}$ indexed by curvilinear abscissa $[0\,,S]$. Though the irregularities are represented by random field $\mathbf{X}$, the calibration procedure presented here does not make use of the model developed in [13]. Instead, the available irregularities measurements performed on various track stretches are considered as realizations of random field $\mathbf{X}$.

The measurement method for the track geometric irregularities has shown a very good reproducibility. Consequently, the measurement noise affecting these irregularities is considered negligible compared to the other sources of uncertainty.

4.2.2. Description of the high-speed train model

The train multibody model consists of rigid bodies linked together by mechanical joints (stiffnesses and dampers) with nonlinear behavior. The wheel-rail contact law is also nonlinear. The flexible modes of the different bodies are not taken into account. More details about the definition of the multibody train model can be found in [14].

Such multibody model contains numerous parameters : body masses and inertiæ, mechanical properties of the joints and relative positions of the bodies and the joints. For this application, only $q = 7$ mechanical parameters are involved in the calibration process. These parameters of interest have been chosen based on the criticity of the corresponding suspension element and on their influence on the train response. Moreover,

for each bogie, all the suspensions of the same type depend on one single parameter. Indeed, we observed that we were unable to distinguish between the contributions of the different elements of the same type with the available sensors. It also allows for significantly decreasing the dimension of the problem.

The q parameters are gathered in the random vector $\mathbf{W}$, belonging to the admissible set $\mathcal{C}_{\mathbf{W}} \subset \mathbb{R}^q$. The randomness accounts for the fact that the actual values of these parameters are unknown. From the technical specifications of each type of suspension, admissible intervals centered around the nominal value are defined for each mechanical parameter. The admissible set $\mathcal{C}_{\mathbf{W}}$ simply consists of the product of these admissible intervals. The initial knowledge we have about them is represented by the prior probability density function $p_{\mathbf{W}}^{\text{prior}}$ of random vector $\mathbf{W}$. For this application, the prior PDF is set uniform on the admissible set. All the other parameters of the model are set to their nominal values.

4.2.3. Description of the system output: the high-speed train dynamic response

The train dynamic response consists of n acceleration signals of specific points in the train, along the vertical and lateral axes (in the axis system attached to the train). These points correspond to the location of the embedded accelerometers, on carbodies and bogies near the carbodies junctions. These different accelerations constitute the different components of the vector output signal.

The simulation is performed step by step in the time domain. The output signals, also given in the time domain, are transformed in the frequency domain. More precisely, in order to avoid systematic phase-shift between the measured and the simulated signals, the studied quantity is the amplitude of the Fourier transform of the acceleration time signals. This amplitude is taken in dB to characterize the system resonances as well as antiresonances.

For $1 \leq k \leq n$, $\{a_k(t)\,, t \in [0\,, T]\}$ denotes the k^{th}-acceleration signal in the time domain for a circulation of duration T; the corresponding response in the frequency domain $\{\hat{a}_k(\omega)\,, \omega \in \Omega\}$ is computed as:

$$\hat{a}_k(\omega) = 10\,\log_{10}\left|\int_0^T \frac{1}{\sqrt{T}} a_k(t) \mathrm{e}^{-i\omega t}\mathrm{d}t\right|. \tag{14}$$

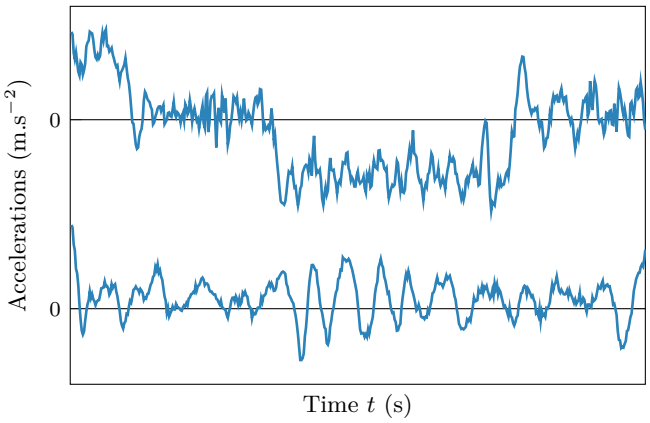

Figure 3: Vertical (top curve) and lateral (bottom curve) accelerations in a carbody, in the time domain.

The set Ω is the frequency band of interest.

Figures 3 and 4 present an illustration of the four accelerations signal in the time domain. Figure 5 presents an illustration of one component of the measured train dynamic response, in the frequency domain. The mean function and quantiles have been obtained from measurements performed on multiple track stretches, each one considered as an independent realization of the stochastic train dynamic response. No axes scales are indicated for confidentiality reasons.

The black-box simulation is represented by the deterministic mapping

$$\mathbf{h} : (\{\mathbf{x}(s)\,, s \in [0\,, S]\}; \mathbf{w}) \mapsto \{\hat{\mathbf{a}}(\omega)\,, \omega \in \Omega\}\,. \tag{15}$$

It associates the response $\hat{\mathbf{a}} = \mathbf{h}(\mathbf{x}; \mathbf{w})$ in the frequency domain with the irregularity signal $\mathbf{x}$ and vector $\mathbf{w}$ of parameters. If the quantities $\mathbf{x}$ and $\mathbf{w}$ are replaced by their stochastic counterpart $\mathbf{X}$ and $\mathbf{W}$, the simulation output $\mathbf{h}(\mathbf{X}; \mathbf{W})$ becomes stochastic as well.

The multibody modeling and simulation necessarily contain inaccuracies and simplifications compared to the real system. Numerical solving is also a source of errors. To perform a robust calibration, a train model error has to be introduced, in order to take into account the fact that the model cannot exactly represent the reality. Moreover, the measurements of the train response performed by embedded accelerometers may contain a certain level of measurement noise. These two types of uncertainties (model

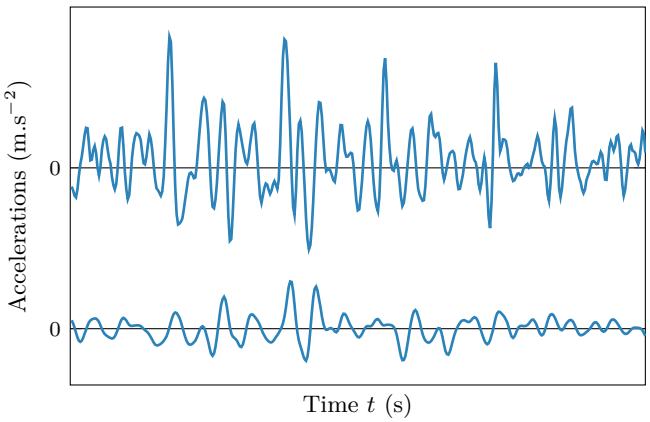

Figure 4: Vertical (top curve) and lateral (bottom curve) accelerations in a bogie, in the time domain.

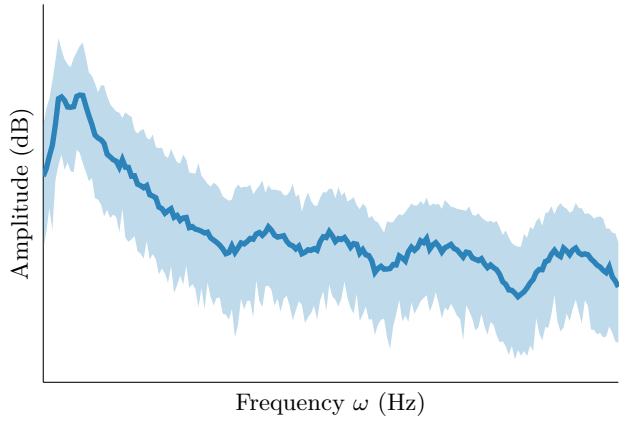

Figure 5: Lateral acceleration in a carbody, represented in the frequency domain. The solid line represents the mean function, while the filled area represents the 5% and 95% quantiles.

errors and measurements noise) create a distance between the measured and simulated train dynamic response. As explained at the beginning of Section 4.2, this distance is taken into account and represented by a random output predictive error $\mathbf{B}$ that is added to the output of the computational model in order to obtain the final model of the train dynamic response including the model and measurement errors.

We choose to define the error $\mathbf{B}$ as a $\mathbb{R}^n$-valued Gaussian process $\{\mathbf{B}(\omega)\,, \omega \in \Omega\}$ indexed by frequency band Ω. This choice is made to simplify both the identification of $\mathbf{B}$ and the computation the likelihood function values (see Section 4.3.2). Moreover, since the amplitude of the train dynamic response is studied on a log-scale, it is not necessary to respect positivity. Its dimension and definition set are identical to those of the response $\mathbf{Y}$. The identification of this output predictive error is performed from a reference set of measurements for which the model parameters $\mathbf{W}$ are known. This approach relies on the strong hypothesis that the error is independent from the train parameters and does not evolve with time. In the present case, this reference set corresponds to measurements performed after the renewal of the suspension elements. Their mechanical characteristics are assumed to be nominal. The details of the identification of process $\mathbf{B}$ from the reference set are given in Appendix B.

The train response model, denoted as $\{\mathbf{Y}(\omega),\, \omega \in \Omega\}$ is a $\mathbb{R}^n$-valued stochastic process indexed by frequency band Ω. Each component corresponds to the acceleration of a specific point in a specific direction. Its relationship to the model input and parameters is summed up by the following expression:

$$\mathbf{Y} = \mathbf{h}(\mathbf{X}; \mathbf{W}) + \mathbf{B}\,. \tag{16}$$

4.3. Specificities of the calibration procedure for the applicative railway case

4.3.1. Experimental data

The calibration is performed to determine the suspensions health state at a given date. We are focusing on a single train, for which a model error has been identified. At a given date, the experimental data consist of a set $\mathcal{U}$ of independent joint measurements of the track geometric irregularities and of the corresponding train dynamic response,

denoted respectively $\mathbf{x}^{\mathrm{mes},i}$ and $\mathbf{y}^{\mathrm{mes},i}$, $1 \leq i \leq \nu$ on ν different track stretches:

$$\mathcal{U} = \{(\mathbf{x}^{\mathrm{mes},i}, \mathbf{y}^{\mathrm{mes},i})\}_{1 \leq i \leq \nu} \,. \tag{17}$$

4.3.2. Calculation of the likelihood function

The calculation of the likelihood function presented in Section 2 is adapted to the present railway case. We need to take into account measurements of the track geometric irregularities. Consequently, the model parameters are not conditioned solely by the output $\mathbf{Y}$ but also by the input-output couple $(\mathbf{X}, \mathbf{Y})$. It makes a difference to consider the probability of obtaining a certain dynamic response whatever the irregularities, and to obtain the same response knowing the irregularities that have triggered it.

Moreover, since several independent measurements are used to perform the calibration, it is necessary to introduce a set $\{(\mathbf{X}^i, \mathbf{Y}^i)\}_{1 \leq i \leq \nu}$ of ν independent and identically distributed copies of couple $(\mathbf{X}, \mathbf{Y})$. Eq. (1) giving the expression of the posterior PDF of parameters $\mathbf{W}$ then becomes (the PDF arguments are omitted for simplicity):

$$\begin{aligned} p_{\mathbf{W}}^{\mathrm{post}} &= p_{\mathbf{W} \,|\, \mathbf{X}^1, \mathbf{Y}^1 \ldots \mathbf{X}^\nu, \mathbf{Y}^\nu} && (18) \\ &\propto p_{\mathbf{Y}^1 \ldots \mathbf{Y}^\nu \,|\, \mathbf{X}^1 \ldots \mathbf{X}^\nu, \mathbf{W}} \, p_{\mathbf{W} \,|\, \mathbf{X}^1 \ldots \mathbf{X}^\nu} && (19) \\ &\propto \prod_{i=1}^{\nu} p_{\mathbf{Y}^i | \mathbf{X}^i, \mathbf{W}} \, p_{\mathbf{W}}^{\mathrm{prior}} \,. && (20) \end{aligned}$$

Eq. (20) is obtained by considering that:

- the track geometric irregularities $\mathbf{X}$ and the train parameters $\mathbf{W}$ are independent, so $p_{\mathbf{W} \,|\, \mathbf{X}} = p_{\mathbf{W}}^{\mathrm{prior}}$;
- for any $(i, j) \in \{1, \ldots \nu\}^2$, $\mathbf{X}^i$ and $\mathbf{Y}^i$ are supposed to be independent from $\mathbf{X}^j$ and $\mathbf{Y}^j$ if $i \neq j$. Indeed, the track has been divided into numerous stretches so that the realizations of the track geometric irregularities and of the train dynamic response can be considered independent on two different stretches.

The likelihood function is then given by

$$\mathcal{L} : \mathbf{w} \mapsto \prod_{i=1}^{\nu} p_{\mathbf{Y}^i | \mathbf{X}^i, \mathbf{W}}(\mathbf{y}^{\mathrm{mes},i} | \mathbf{x}^{\mathrm{mes},i}, \mathbf{w}) \,. \tag{21}$$

Using Eq. (16) the log-likelihood function can be written

$$L : \mathbf{w} \mapsto \sum_{i=1}^{\nu} \log \left(p_{\mathbf{B}}(\mathbf{y}^{\mathrm{mes},i} - \mathbf{h}(\mathbf{x}^{\mathrm{mes},i}; \mathbf{w})) \right) . \tag{22}$$

From the previous expression, one can observe that ν simulation runs are required for each evaluation of log-likelihood L at each points $\mathbf{w} \in \mathcal{C}_{\mathbf{W}}$. The number ν of available measurements can typically be several hundreds in the present case, which explains the high computational cost for each call to the likelihood function.

4.3.3. Parameters of the GP surrogate modeling

For this application, the number of suspension parameters to identify is $q = 7$ (see Section 4.2.2). The number of parameters gives the dimension of random vector $\mathbf{W}$ and the dimension of the definition set of likelihood function $\mathcal{L}$.

The size of training set is 500. The chosen form for the covariance function is Matérn-$\frac{5}{2}$ [15]. Moreover, we chose to build the GP surrogate model considering that the observations of the log-likelihood function were noisy (see Appendix A.5). The variance of this noise was optimized to fit the available data along with the other correlation parameters. This choice was made in order to improve the quality of the surrogate model. Introducing a small noise improved its regularity. Indeed, it offers the surrogate a certain margin of freedom around the observation. On the contrary, forcing the surrogate model to be strictly interpolating by considering that the observations are exact sometimes resulted in unexpected oscillations of the surrogate model.

We tried to refine the Gaussian surrogate around the maximum of the likelihood function following the KGCP policy proposed in [16]. However, no significant improvement of the calibration results were observed with the refined surrogate model. It suggests that the initial training set is large enough to correctly know the behavior of the likelihood function. The observations being considered noisy, adding new points to the training set does not significantly increase the accuracy of the maximum location.

4.4. Choice of the MCMC algorithm

In our preliminary tests, the posterior PDF appeared to be very peaked, which prevents the classical Metropolis-Hastings algorithm [17] from correctly sampling from

it. We also tried to run several chains in parallel, but the final results were too dependent on the initial choice of the chains starting points. Consequently, we chose to use a different algorithm called Transitional MCMC [18, 19], more adapted to peaked target PDF. Details about the algorithm are given in Appendix C.

4.5. Calibration results

This section presents the results obtained with the Bayesian calibration method for different cases:

- numerical experiments, that is to say using simulated data;
- with measurements performed at the reference date, which were used to identify the model error;
- with measurements performed six month after the reference date.

The goals of this section are to:

- validate the proposed method, using the numerical experiments and the measurements at the reference date;
- show the impact of considering the uncertainty introduced by the use of a surrogate model of the likelihood function on the calibration results.

It should be noted that the speed up provided by the use of a surrogate model compared to an estimation of the parameters posterior distribution that would rely on the computation of the exact likelihood is estimated of order 10^6.

4.5.1. Numerical experiments

In the present case, a numerical experiment consists of simulated train responses that are used as if they were experimental data. They are generated using actual measurements of the track geometric irregularities on several track stretches and known degraded suspension parameters. Moreover, an independent realization of the error $\mathbf{B}$ is added to the response signal on each track stretch in order to generate a quantity as close as possible to an actual measurement. The numerical experiments allows for

validating the calibration procedure: the procedure is applied on the artificial train response, the calibration results can then be compared to the reference parameters used to generate the response.

At a given date, we suppose that a set of ν_1 track irregularities measurements $\{\mathbf{x}^{\mathrm{mes},i}\}_{1\leq i\leq \nu_1}$ is available. The validation procedure from a numerical experiment can then be summed up as follows.

1. Choose artificial parameters $\mathbf{w}_1$;
2. Run the simulation on the ν_1 track stretches with parameters $\mathbf{w}_1$;
3. Generate ν_1 independent realizations $\{\mathbf{b}^i\}_{1\leq i\leq \nu_1}$ of output predictive error $\mathbf{B}$. Because $\mathbf{B}$ is Gaussian, this generation can be performed very easily from normal independent samples using Eq. (B.6) ;
4. Add these realizations to the simulated response to obtain artificial realizations of the train dynamic response

$$\mathbf{y}^{\mathrm{num},i} = \mathbf{h}(\mathbf{x}^{\mathrm{mes},i};\mathbf{w}_1) + \mathbf{b}^i,\ 1\leq i\leq \nu_1\,; \tag{23}$$

5. Perform the calibration using input data

$$\mathcal{U}_1 = \{(\mathbf{x}^{\mathrm{mes},i},\mathbf{y}^{\mathrm{num},i})\}_{1\leq i\leq \nu_1} \tag{24}$$

   to obtain the calibrated (random) parameters $\mathbf{W}_1^{\mathrm{opt}}$;
6. Compare $\mathbf{W}_1^{\mathrm{opt}}$ to $\mathbf{w}_1$.

The validation procedure has been performed with different values of the artificial parameters $\mathbf{w}_1$. Figure 6 presents calibration results with $\mathbf{w}_1$ set to the nominal values. On this graph are displayed the marginal densities of the posterior PDF of $\mathbf{W}_1^{\mathrm{opt}}$. It also compares the results obtained with the KP method that solely uses the Kriging predictor provided by the surrogate model of the likelihood function and with the MCT method that also includes the uncertainty related to the surrogate model.

One can first observe that the distributions are close to the nominal value (corresponding to 0.5 on the graph). Except parameter 2, the difference between the maximum of the marginal PDFs and the nominal value is always lower than 5% of the size of the admissible interval. The dispersion varies from one parameter to another. This

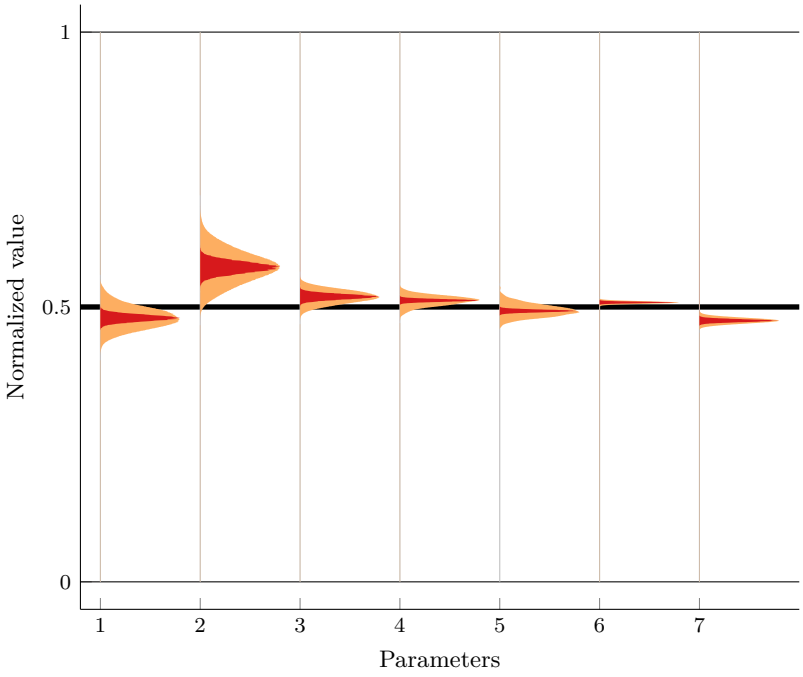

Figure 6: Results of the numerical experiment with nominal parameters. The parameter values are also normalized so that every parameter varies between 0 and 1; 0.5 then corresponds to the nominal value. The curves represent the normalized posterior PDF marginals for each parameter, obtained using the KP (red or dark gray) and MCT (orange or light gray) methods.

can be explained by the initial choice of the admissible intervals, since the results are rescaled according to their size, but more importantly by the sensitivity of the train dynamic response to the different mechanical parameters. The results for parameter 2 are a good illustration of this sensitivity question. Contrary to the other parameter, it corresponds to a suspension that is not located on the same bogie as the sensors, but at the other end of the carbody. Consequently, we expect this parameter to have less influence on the train dynamic response measured by the sensors. This is coherent with the results: the distribution is further away from the nominal value, and its dispersion is greater than for the other parameters. These considerations also highlight the interest of Bayesian calibration: because we measure the uncertainty on the calibrated parameter, we have a way to assess the accuracy of the calibration and the confidence we can put on its results. The fact that the marginal PDFs are rather peaked comes from the large database that is available for the calibration. The comparison of the results obtained using the KP and MCT methods shows that the marginals PDF are more spread with the MCT methods. With the KP method, a source of uncertainty, the error introduced by the approximation of the likelihood function by a surrogate model, is not taken into account. As a consequence, the uncertainty on the calibrated parameters is reduced. Using the KP method thus leads to an overestimation of the calibration accuracy. Nevertheless, the marginals maxima appear to be located at very similar parameter values with the two methods. The KP method seems satisfying to determine only the most probable parameter values. It also allows for estimating the proportion of the calibrated parameters variance that stems from the system uncertainties.

Figure 7 presents calibration for arbitrary degraded values of the artificial parameters $\mathbf{w}_1$. Two different locations of the bogie are considered: at the head and the rear of the train. The vertical and lateral accelerations the train undergoes at these two locations are indeed very different. One can also note on the bottom graph that we studied the case of a parameter set at the boundary of the admissible interval. The calibrated parameters $\mathbf{W}_1^{\mathrm{opt}}$ show a good correspondence to the input parameters $\mathbf{w}_1$. The numerical experiments results are conclusive for the validation of both the KP and the MCT methods, with the limits previously noted for the KP method.

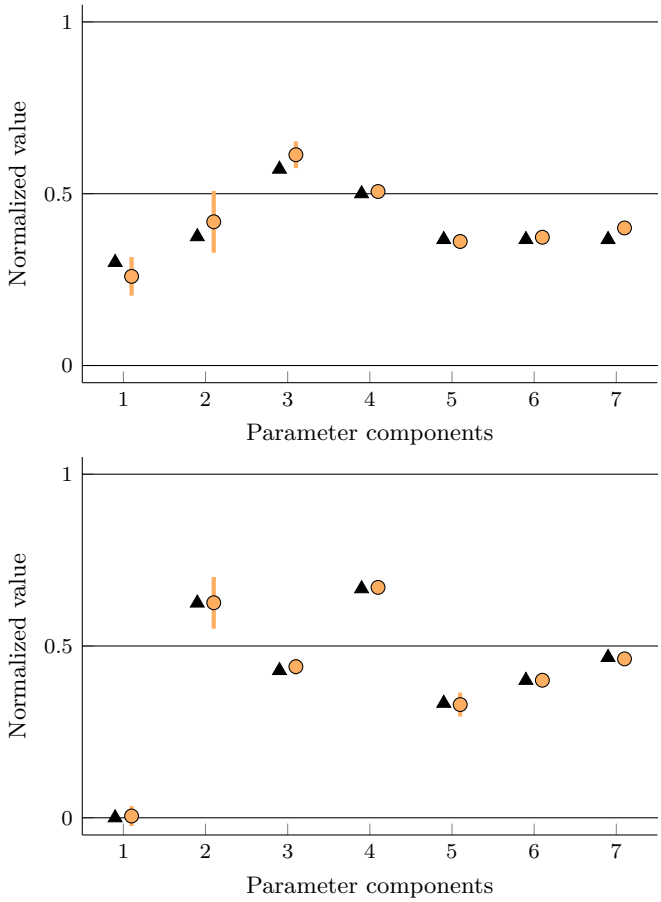

Figure 7: Results of the numerical experiments with arbitrary parameters, using the MCT method, for a bogie located at the head (top graph) and rear (bottom graph) of the train. The input parameters values $\mathbf{w}_1$ (black triangles) are compared to the mean of the posterior PDF marginals (orange dots). The orange lines represents the 98% confidence intervals around these calibrated values. The parameter scale normalization is identical to figure 6.

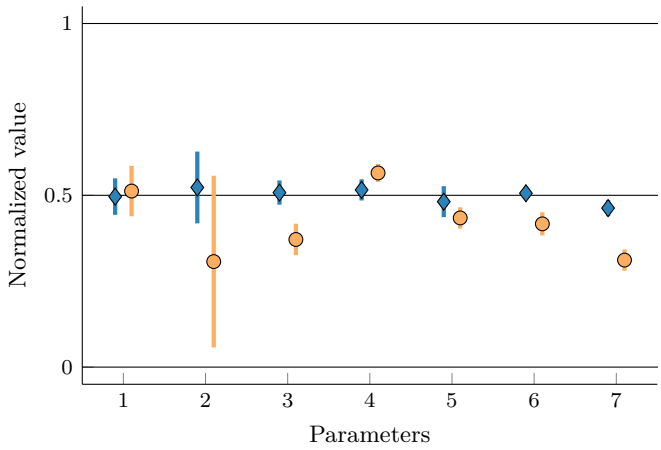

Figure 8: Calibration results using the MCT method at time T_0 (blue diamonds) and T_1 (orange circles). The graph layout is the identical to figure 7.

4.5.2. Results using actual measurements of the high-speed train dynamic response

This section presents calibration results obtained with actual measurements. The first set of measurements is the one collected at the reference date, which is used to identify the output predictive error $\mathbf{B}$, denoted as T_0. The second set gathers measurements performed at a date T_1, six months after the reference date T_0.

Figure 8 presents the calibration results obtained for these two dates using the MCT method. As expected, one can observe that the parameter mean values for T_0 are close to the nominal value (0.5 on the normalized scale). Indeed, the output predictive error has been identified considering the parameters value is nominal at the reference date T_0. This good correspondence is a second way of validating of the method. The results for the date T_1 show a significant evolution compared to the nominal values. Except for the second parameter, whose case has been treated in the previous section, we can observe a high confidence in the calibrated results. To obtain a complete validation of the identification method, experimental tests need to be performed on the isolated suspension elements in order to measured their mechanical characteristics. Such tests have not been conducted yet.

Figure 9 compares the calibration results using the KP and the MCT method for date T_1. The observations concerning the comparison of the two methods are similar

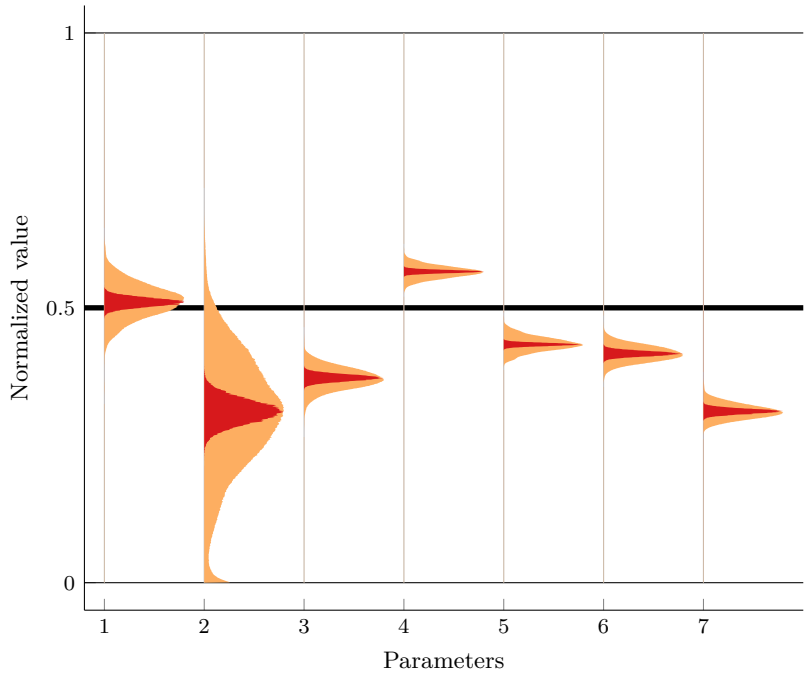

Figure 9: Comparison of calibration results obtained at time T_1 using the KP (red or dark gray curves) and the MCT (orange or light gray curves) methods. The graph layout is identical to figure 6.

to the ones made for figure 6. A difference that can be noted between the numerical experiments and the calibration on the real measurements is the fact that the posterior PDF is more dispersed for the latter. This higher dispersion highlights the limitation of the chosen model for the output predictive error (a Gaussian additive error being independent of the parameter space). For the numerical experiments, by construction, this model exactly matches the error artificially applied on the generated data, which appears to be less true for the measured data. The use of a more complex, non-Gaussian model for the output predictive error could be considered in future works.

5. Conclusion and perspectives

We have proposed a method for the Bayesian calibration of expensive computer codes. The main particularity consists in the surrogate modeling of the likelihood function to address the computational costs. We then focused on evaluating the consequences of this use of an approximation on the calibration results. The uncertainty associated with the surrogate modeling is taken into account by estimating the parameters posterior PDF from trajectories of the random surrogate model. We thus make full use of the variance of the GP model that evaluates the accuracy of the approximation. The need to draw GP trajectories also made us develop an approximation method for the latter, relying on a further conditioning of the GP model.

The procedure has been tested on an industrial case, the monitoring of high-speed train suspensions using acceleration measurements. In this application, the complete MCT method appears to have little effect on the mean value of the calibrated parameters compared to the simple use of the KP method. It however significantly increases the variance of the calibrated parameters. This emphasizes the interest of the method if importance is put on correctly estimating the confidence of the calibration.

Acknowledgments

This research has been supported by *SNCF*, the French National Railway Company.

Appendix A. Gaussian process surrogate model

Gaussian process surrogate modeling consists in representing a deterministic scalar target function by a conditioned Gaussian process (GP). Information about the target function is provided by a set of observations of the function value at a few points in the definition set. These points constitute the initial training set. Randomness is introduced as a way to quantify the approximation stemming from the use of a surrogate model. The objectives of GP surrogate modeling are for example to build an easy-to-compute approximation of an expensive computer code or to model a phenomenon for which all observations are affected by a random noise.

The evaluation of the target function at the points of the training set is generally the computationally expensive step of GP surrogate modeling. Consequently, this set should contain only a limited number of points while maximizing the information provided. Therefore a common choice is to define a space-filling training set to get as much information as possible on the global behavior of the target function on its whole definition set. Appendix A.6 provides an example on how to build a space-filling training set.

After this initial training phase, a second refining step (that will not be detailed in this paper) can be performed in order to explore particular features of the target function. For instance, [20] proposes the EGO algorithm for the optimization of the target function using an expected improvement criterion. In [21], an adaptive design of experiment is proposed for a target function that needs to be accurately approximated around a certain level.

In this appendix, we present the principle of GP surrogate modeling based on the Bayesian approach presented in [22]. Another approach giving equivalent results, based on the minimization of the mean square error, can be found in [23].

Appendix A.1. Problem statement

The goal is to build a surrogate model of a given deterministic real-valued function y defined on a subset $\mathcal{X}$ of $\mathbb{R}^q$, whose value is known only at the n training points $(\mathbf{x}_i)_{1\leq i\leq n}$ in $\mathcal{X}$. An illustration is provided on figure A.10.

Function y is supposed to be a particular sample path of an underlying Gaussian second-order real-valued stochastic process $\{Y(\mathbf{x}),\ \mathbf{x} \in \mathcal{X}\}$, indexed by $\mathcal{X}$. The surrogate model consists of this GP conditioned by the observations of y at the points of the training set. The underlying process is defined according to a parametric formulation, and denoted

$$Y \mid \{\boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\} \sim \mathcal{GP}\left(\mathbf{f}(.)^t\boldsymbol{\beta}\,;\, \sigma^2 R(.,. \mid \boldsymbol{\psi})\right) \tag{A.1}$$

where:

- $\mathcal{GP}(m; \Sigma)$ denotes the Gaussian process whose mean function is $\mathbf{x} \mapsto m(\mathbf{x}) = E\{Y(\mathbf{x})\}$ from $\mathcal{X}$ into $\mathbb{R}$ and covariance function is $(\mathbf{x}, \mathbf{x}') \mapsto \Sigma(\mathbf{x}, \mathbf{x}') = E\{(Y(\mathbf{x}) - m(\mathbf{x}))(Y(\mathbf{x}') - m(\mathbf{x}'))\}$ from $\mathcal{X}^2$ into $\mathbb{R}$;
- $\mathbf{f} : \mathcal{X} \to \mathbb{R}^p$ gathers p deterministic regression functions on $\mathcal{X}$;
- $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of regression coefficients parameterizing the GP mean function;
- $R(\mathbf{x}, \mathbf{x}' \mid \boldsymbol{\psi})$ gives the shape of the covariance function. It depends only on $\mathbf{x} - \mathbf{x}'$ and on a vector $\boldsymbol{\psi}$ of parameters (for example, correlation lengths). It is assumed that $R(\mathbf{x}, \mathbf{x} \mid \boldsymbol{\psi}) = 1$. Consequently, σ^2 is the variance of random variable $Y(\mathbf{x})$ for $\mathbf{x}$ fixed in $\mathcal{X}$ and is thus independent of $\mathbf{x}$.

The mean function of the process is decomposed as a linear combination of regression functions chosen by the user, polynomials for example. The first parameter is thus the vector of regression coefficients $\boldsymbol{\beta}$. The second parameter is the GP variance. The shape of the correlation function $R(.,.)$ (exponential or Matérn for instance) is also chosen by the user according to the expected regularity of y. The third parameter is the vector of correlation parameters $\boldsymbol{\psi}$. They depend on the shape of the correlation function. Parameters $(\boldsymbol{\beta}, \sigma, \boldsymbol{\psi})$ are *a priori* unknown. They are not set by the user but have to be determined using the information about y provided by the observations on the training set.

Appendix A.2. Conditioning by the observations

Let $\mathbf{y}^n$ and $\mathbf{Y}^n$ be respectively the values of function y and of process Y at points $(\mathbf{x}_i)_{1\leq i\leq n}$:

$$\mathbf{y}^n = \begin{bmatrix} y(x_1) \\ \vdots \\ y(x_n) \end{bmatrix}, \qquad \mathbf{Y}^n = \begin{bmatrix} Y(x_1) \\ \vdots \\ Y(x_n) \end{bmatrix}. \tag{A.2}$$

$\mathbf{Y}^n \,|\, \{\boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\}$ is a marginal of process $Y \,|\, \{\boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\}$. As a consequence, for any $\mathbf{x} \in \mathcal{X}$, the vector concatenating $Y(\mathbf{x}) \,|\, \{\boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\}$ and $\mathbf{Y}^n \,|\, \boldsymbol{\beta}, \sigma, \boldsymbol{\psi}$ is Gaussian:

$$\begin{bmatrix} Y(\mathbf{x}) \\ \mathbf{Y}^n \end{bmatrix} \,|\, \{\boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{f}(\mathbf{x})^t \\ [F] \end{bmatrix} \boldsymbol{\beta} \,;\, \sigma^2 \begin{bmatrix} 1 & \mathbf{r}(\mathbf{x})^t \\ \mathbf{r}(\mathbf{x}) & [R] \end{bmatrix} \right) \tag{A.3}$$

with

$$[F] = \begin{bmatrix} \mathbf{f}(\mathbf{x}_1)^t \\ \vdots \\ \mathbf{f}(\mathbf{x}_n)^t \end{bmatrix}, \qquad \mathbf{r}(\mathbf{x}) = \begin{bmatrix} R(\mathbf{x}, \mathbf{x}_1 \,|\, \boldsymbol{\psi}) \\ \vdots \\ R(\mathbf{x}, \mathbf{x}_n \,|\, \boldsymbol{\psi}) \end{bmatrix} \tag{A.4}$$

and $[R]_{ij} = R(\mathbf{x}_i, \mathbf{x}_j \,|\, \boldsymbol{\psi})$, $1 \leq i, j \leq n$. Matrix $[R]$ is assumed to be invertible.

The surrogate model is then obtained by conditioning process Y by $\mathbf{Y}^n = \mathbf{y}^n$. When doing so, $Y \,|\, \{\mathbf{Y}^n, \boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\}$ remains Gaussian. Its mean function and variance are immediately deduced from the formula of the conditioned Gaussian random variables:

$$\begin{aligned} Y \,|\, \{\mathbf{Y}^n = \mathbf{y}^n, \boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\} \sim \mathcal{GP}\big(&\mathbf{x} \mapsto \mathbf{f}(\mathbf{x})^t\boldsymbol{\beta} + \mathbf{r}(\mathbf{x})^t[R]^{-1}(\mathbf{y}^n - F\boldsymbol{\beta}) \;; \\ &(\mathbf{x}, \mathbf{x}') \mapsto \sigma^2(R(\mathbf{x}, \mathbf{x}' \,|\, \boldsymbol{\psi}) - \mathbf{r}(\mathbf{x})^t[R]^{-1}\mathbf{r}(\mathbf{x}'))\big) \end{aligned} \tag{A.5}$$

The following sections detail how to deal with the fact that parameters $(\boldsymbol{\beta}, \sigma, \boldsymbol{\psi})$ are actually unknown by exploiting the information provided by the training set.

Appendix A.3. Mean function parameter $\boldsymbol{\beta}$

In this section, σ and $\boldsymbol{\psi}$ are supposed to be known. Only the regression coefficients $\boldsymbol{\beta}$ are supposed to be unknown. The fact that no information is *a priori* available about $\boldsymbol{\beta}$ is taken into account by following a hierarchical approach. Parameter $\boldsymbol{\beta}$ is represented by a random vector with non-informative prior: $p_{\boldsymbol{\beta}} \propto 1$. The training

set is then used to learn about the distribution of $\boldsymbol{\beta}$. The principle is to determine the distribution of $\boldsymbol{\beta} \,|\, \mathbf{Y}^n = \mathbf{y}^n$ for a fixed value of σ and $\boldsymbol{\psi}$. In this section, the conditioning on σ, $\boldsymbol{\psi}$ is not systematically repeated for simplicity. Using the Bayes formula, one can write:

$$\begin{aligned} p_{\boldsymbol{\beta}}(\mathbf{b} \,|\, \mathbf{Y}^n = \mathbf{y}^n) &\propto p_{\mathbf{Y}^n}(\mathbf{y}^n \,|\, \boldsymbol{\beta} = \mathbf{b})\, p_{\boldsymbol{\beta}}(\mathbf{b}) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}^n - [F]\mathbf{b})^t [R]^{-1}(\mathbf{y}^n - [F]\mathbf{b})\right) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}(\mathbf{b} - \widehat{\boldsymbol{\beta}})^t [Q](\mathbf{b} - \widehat{\boldsymbol{\beta}})\right) \end{aligned} \tag{A.6}$$

with $[Q] = [F]^t[R]^{-1}[F]$ and $\widehat{\boldsymbol{\beta}} = [Q]^{-1}[F]^t[R]^{-1}\mathbf{y}^n$. Therefore

$$\boldsymbol{\beta} \,|\, \{\mathbf{Y}^n = \mathbf{y}^n, \sigma, \boldsymbol{\psi}\} \sim \mathcal{N}\left(\widehat{\boldsymbol{\beta}}; \sigma^2 [Q]^{-1}\right). \tag{A.7}$$

Knowing the probability distribution of $\boldsymbol{\beta} \,|\, \mathbf{Y}^n$ for a fixed value of σ and $\boldsymbol{\psi}$, process $Y \,|\, \{\mathbf{Y}^n, \boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\}$ can be statistically averaged with respect to random vector $\boldsymbol{\beta}$. It means that $Y \,|\, \{\mathbf{Y}^n, \sigma, \boldsymbol{\psi}\}$ can be used instead of $Y \,|\, \{\mathbf{Y}^n, \boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\}$. This is achieved by relying on the rule of conditional expectation. For $\mathbf{x} \in \mathcal{X}$, the mean value and variance of $Y(\mathbf{x}) \,|\, \{\mathbf{Y}^n = \mathbf{y}^n, \sigma, \boldsymbol{\psi}\}$ are:

$$\begin{aligned} E\{Y(\mathbf{x}) \,|\, \mathbf{Y}^n\} &= E_{\boldsymbol{\beta}}\{E\{Y(\mathbf{x}) \,|\, \mathbf{Y}^n, \boldsymbol{\beta}\} \,|\, \mathbf{Y}^n\} \\ &= E_{\boldsymbol{\beta}}\{\mathbf{f}(\mathbf{x})^t\boldsymbol{\beta} + \mathbf{r}(\mathbf{x})^t[R]^{-1}(\mathbf{y}^n - [F]\boldsymbol{\beta}) \,|\, \mathbf{Y}^n\} \\ &= \mathbf{f}(\mathbf{x})^t\widehat{\boldsymbol{\beta}} + \mathbf{r}(\mathbf{x})^t[R]^{-1}(\mathbf{y}^n - [F]\widehat{\boldsymbol{\beta}})\,; \end{aligned} \tag{A.8}$$

$$\begin{aligned} \mathrm{Var}\{Y(\mathbf{x}) \,|\, \mathbf{Y}^n\} &= E_{\boldsymbol{\beta}}\{\mathrm{Var}\{Y(\mathbf{x}) \,|\, \mathbf{Y}^n, \boldsymbol{\beta}\} \,|\, \mathbf{Y}^n\} \\ &\quad + \mathrm{Var}_{\boldsymbol{\beta}}\{E\{Y(\mathbf{x}) \,|\, \mathbf{Y}^n, \boldsymbol{\beta}\} \,|\, \mathbf{Y}^n\} \\ &= E_{\boldsymbol{\beta}}\{\sigma^2(1 - \mathbf{r}(\mathbf{x})^t[R]^{-1}\mathbf{r}(\mathbf{x})) \,|\, \mathbf{Y}^n\} \\ &\quad + \mathrm{Var}_{\boldsymbol{\beta}}\{\mathbf{f}(\mathbf{x})^t\boldsymbol{\beta} + \mathbf{r}(\mathbf{x})^t[R]^{-1}(\mathbf{y}^n - [F]\boldsymbol{\beta}) \,|\, \mathbf{Y}^n\} \\ &= \sigma^2(1 - \mathbf{r}(\mathbf{x})^t[R]^{-1}\mathbf{r}(\mathbf{x}) + \mathbf{u}(\mathbf{x})^t[Q]^{-1}\mathbf{u}(\mathbf{x})) \end{aligned} \tag{A.9}$$

with $\mathbf{u}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - [F]^t[R]^{-1}\mathbf{r}(\mathbf{x})$. This can be written as

$$Y \mid \{\mathbf{Y}^n, \sigma, \boldsymbol{\psi}\} \sim \mathcal{GP}\left(\mathbf{x} \mapsto \mathbf{f}(\mathbf{x})^t\widehat{\boldsymbol{\beta}} + \mathbf{r}(\mathbf{x})^t[R]^{-1}(\mathbf{y}^n - F\widehat{\boldsymbol{\beta}}) \ ; \right.$$
$$\left. (\mathbf{x}, \mathbf{x}') \mapsto \sigma^2(R(\mathbf{x}, \mathbf{x}' \mid \boldsymbol{\psi}) - \mathbf{r}(\mathbf{x})^t[R]^{-1}\mathbf{r}(\mathbf{x}') + \mathbf{u}(\mathbf{x})^t[Q]^{-1}\mathbf{u}(\mathbf{x}'))\right). \quad \text{(A.10)}$$

Appendix A.4. Variance and correlation parameters

Parameters σ and $\boldsymbol{\psi}$ could be estimated using the same hierarchical approach, by putting prior distributions on these variables. However, in this case, no closed form can be determined in general for Y. Instead, parameters σ and $\boldsymbol{\psi}$ are determined according to a criterion assessing how well the Gaussian process is fitting the data provided by the training set. The criterion usually used is the Maximum Likelihood Estimation (MLE). The principle of MLE criterion is to maximize the density

$$p_{\mathbf{Y}^n \mid \boldsymbol{\beta}, \sigma, \boldsymbol{\psi}}(\mathbf{y}^n \mid \widehat{\boldsymbol{\beta}}, \sigma, \boldsymbol{\psi}) = \left((2\pi)^n \det(\sigma^2[R])\right)^{-\frac{1}{2}}$$
$$\times \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}^n - [F]\widehat{\boldsymbol{\beta}})^t[R]^{-1}(\mathbf{y}^n - [F]\widehat{\boldsymbol{\beta}})\right) \quad \text{(A.11)}$$

The value of σ maximizing this density can be determined explicitly:

$$\sigma^2 = \frac{1}{n}(\mathbf{y}^n - [F]\widehat{\boldsymbol{\beta}})^t[R]^{-1}(\mathbf{y}^n - [F]\widehat{\boldsymbol{\beta}}) \quad \text{(A.12)}$$

With this value of σ, the previous maximization is equivalent to minimizing the quantity $\sigma^2 \det([R])^{\frac{1}{n}}$ in order to determine the optimal value of $\boldsymbol{\psi}$. In general, no closed form exists for the covariance parameter; this step has to be performed numerically.

Appendix A.5. Case with noisy observations

Building a surrogate model based on a conditioned Gaussian process is also possible when the values of y on the training set are not computed exactly but affected by a random noise. An example is shown on figure A.11. The noise is considered Gaussian, of zero mean, of variance σ_ε^2 and uniform on the definition set $\mathcal{X}$.

In such a case, the available data are not $\mathbf{y}^n$ as previously defined, but rather $\widetilde{\mathbf{y}}^n$ that gathers the observations of y at each point $\mathbf{x}_i$ of the training set plus an unknown

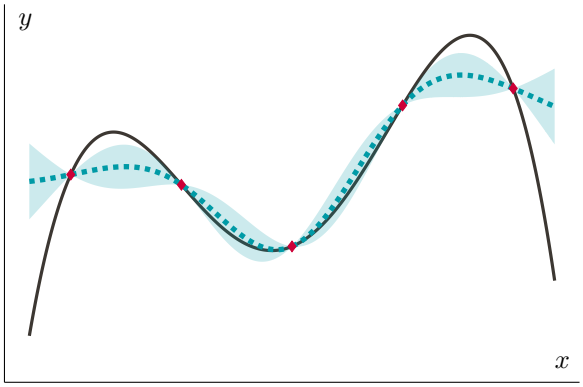

Figure A.10: 1-D example of GP surrogate model with exact observations: the solid black line represents target function y, the red diamonds the points of the training set, the green dashed line the mean function of the GP model and the green area its standard deviation.

realization ε_i of the random noise:

$$\widetilde{\mathbf{y}}^n = \begin{bmatrix} y(x_1) + \varepsilon_1 \\ \vdots \\ y(x_n) + \varepsilon_n \end{bmatrix}. \tag{A.13}$$

The corresponding model is the random vector $\widetilde{\mathbf{Y}}^n = \mathbf{Y}^n + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon}$ is a centered Gaussian vector with covariance matrix $\sigma_\varepsilon^2[I_n]$, with $[I_n]$ the identity matrix of dimension n. Random vector $\boldsymbol{\varepsilon}$ is independent of $\mathbf{Y}^n$.

Y and $\boldsymbol{\beta}$ must now be conditioned by $\widetilde{\mathbf{Y}}^n = \widetilde{\mathbf{y}}^n$ and not by $\mathbf{Y}^n = \mathbf{y}^n$. For $\mathbf{x} \in \mathcal{X}$, the joint probability distribution of $Y(\mathbf{x})$ and $\widetilde{\mathbf{Y}}^n$ can be expressed as follows:

$$\begin{bmatrix} Y(\mathbf{x}) \\ \widetilde{\mathbf{Y}}^n \end{bmatrix} \mid \{\boldsymbol{\beta}, \sigma, \boldsymbol{\psi}\} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{f}(\mathbf{x})^t \\ [F] \end{bmatrix} \boldsymbol{\beta}\,;\, \sigma^2 \begin{bmatrix} 1 & \mathbf{r}(\mathbf{x})^t \\ \mathbf{r}(\mathbf{x}) & [R] + \sigma_\varepsilon^2[I_n] \end{bmatrix} \right) \tag{A.14}$$

The results of Appendix A.2 and Appendix A.3 hold, with covariance matrix $[R]$ replaced by $[\widetilde{R}] = [R] + \sigma_\varepsilon^2[I_n]$. No closed form can be found anymore for the optimal value of σ using the MLE criterion. Consequently, it has to be optimized numerically along with parameter $\boldsymbol{\psi}$. The variance σ_ε^2 of the noise can be set by the user or optimized along with parameters σ and $\boldsymbol{\psi}$.

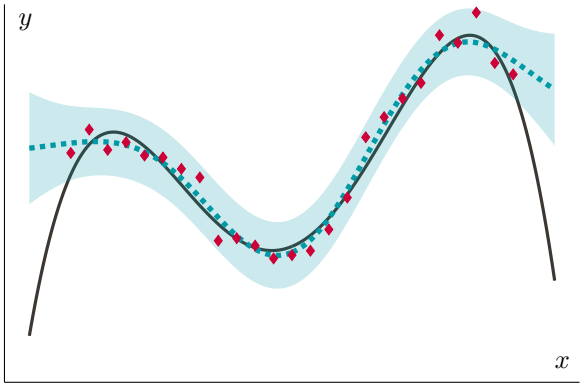

Figure A.11: 1-D example of GP surrogate model with noisy observations (same legend as figure A.10).

Appendix A.6. Space-filling training set

This appendix presents one method for the generation of a space filling training set in a set $\mathcal{X}$ consisting of the set product of bounded intervals of $\mathbb{R}$. The chosen method is the optimization of a Latin Hypercube Sample (LHS) according to a *maximin* distance (see [24]).

To build a N-points LHS, admissible set $\mathcal{X}$ must be divided into N cells along each dimension. The points are then scattered in $\mathcal{X}$ so that in every dimension, each cell only contains one point of the sample. The interest of a LHS is that if you consider a particular dimension, the sample points are regularly spaced, with no redundancy.

However, a LHS is not necessarily space-filling. We choose to measure the space-filling property with the distance δ that consists of the smaller Euclidian distance between two points of the training set: for a finite discrete subset $\mathcal{W}$ of $\mathcal{X}$, distance δ is defined as

$$\delta(\mathcal{W}) = \min_{\substack{(\mathbf{w}_1,\mathbf{w}_2)\in\mathcal{W}^2 \\ \mathbf{w}_1\neq\mathbf{w}_2}} \|\mathbf{w}_1 - \mathbf{w}_2\|_2 \tag{A.15}$$

The greater $\delta(\mathcal{W})$, the most space-filling $\mathcal{W}$ is. In practice, numerous LHS candidates are drawn. The one for which the value of criterion δ is the greatest is kept as the best training set in $\mathcal{X}$.

Appendix B. Identification of the output predictive error

In this section, we consider that all the functional quantities are discretized. The μ discretization points $(\omega_i)_{1\leq i\leq \mu}$ span the frequency band Ω. For the processes containing several components, the latter are concatenated to obtain a single column vector. For instance, the discretized version of process $\{\mathbf{Y}(\omega), \omega \in \Omega\}$ is the vector of size $n\mu$

$$[Y_1(\omega_1)\dots Y_1(\omega_\mu)\dots Y_n(\omega_1)\dots Y_n(\omega_\mu)]^t. \tag{B.1}$$

Reference set $\mathcal{U}_0$ contains independent joint measurements of the track geometric irregularities and of the train response, denoted respectively $\mathbf{x}^{\mathrm{ref},i}$ and $\mathbf{y}^{\mathrm{ref},i}$, $1 \leq i \leq \nu_0$, performed on ν_0 track stretches:

$$\mathcal{U}_0 = \{(\mathbf{x}^{\mathrm{ref},i}, \mathbf{y}^{\mathrm{ref},i})\}_{1\leq i\leq \nu_0}\,. \tag{B.2}$$

For $1 \leq i \leq \nu_0$, let then $\mathbf{b}^{\mathrm{ref},i}$ be the difference between the measured train response and the simulated one (with nominal parameters) on the i^{th} track stretch:

$$\mathbf{b}^{\mathrm{ref},i} = \mathbf{y}^{\mathrm{ref},i} - \mathbf{h}(\mathbf{x}^{\mathrm{ref},i}, \mathbf{w}_0) \tag{B.3}$$

with $\mathbf{w}_0$ the nominal values of the train parameters $\mathbf{W}$. The set $\{\mathbf{b}^{\mathrm{ref},i}\}_{1\leq i\leq \nu_0}$ is considered as a set of realizations of error $\mathbf{B}$. As shown in Section 4.3.2, in order to evaluate the likelihood function value, we need the ability to estimate the PDF of $\mathbf{B}$. Since random vector $\mathbf{B}$ is Gaussian, it is completely defined by its mean vector and its covariance matrix.

From the realizations $\{\mathbf{b}^{\mathrm{ref},i}\}_{1\leq i\leq \nu_0}$ the empirical estimates of the mean function $\widehat{\mathbf{m}}_{\mathrm{B}}$ and covariance function $[\widehat{C}_{\mathrm{B}}]$ are computed:

$$\widehat{\mathbf{m}}_{\mathbf{B}} = \frac{1}{\nu_0}\sum_{i=1}^{\nu_0} \mathbf{b}^{\mathrm{ref},i} \tag{B.4}$$

$$[\widehat{C}_{\mathbf{B}}] = \frac{1}{\nu_0 - 1}[\bar{B}][\bar{B}]^t \tag{B.5}$$

where matrix $[\bar{B}] = [\mathbf{b}^{\mathrm{ref},1} - \widehat{\mathbf{m}}_{\mathrm{B}} \dots \mathbf{b}^{\mathrm{ref},\nu_0} - \widehat{\mathbf{m}}_{\mathrm{B}}]$ gathers the centered realizations. However, since $\nu_0 < n\mu$, the estimate $[\widehat{C}_{\mathbf{B}}]$ of the covariance matrix $C_{\mathbf{B}}$ of $\mathbf{B}$ is not invertible.

Since we need the probability density function of $\mathbf{B}$ for the calculation of the likelihood function, we introduce the following representation of $\mathbf{B}$

$$\mathbf{B} = \mathbf{m_B} + [A]\boldsymbol{\xi} \tag{B.6}$$

where

- $\mathbf{m_B} = E\{\mathbf{B}\}$ is the mean vector of $\mathbf{B}$ whose estimate is $\widehat{\mathbf{m}}_\mathbf{B}$;
- $\boldsymbol{\xi}$ is a centered Gaussian vector of dimension ℓ, with $\ell \leq n\mu$ and the identity matrix as covariance matrix;
- $[A]$ is a full-rank rectangular matrix of dimension $n\mu \times \ell$.

Keeping the same notation for $\boldsymbol{\xi}$ and its projection $\boldsymbol{\xi} = [A]_{\rm pi}^{-1}(\mathbf{B} - \widehat{m}_\mathbf{B})$ in which $[A]_{\rm pi}^{-1} = ([A]^t[A])^{-1}[A]^t$ is the left pseudo-inverse of $[A]$, the pseudo-PDF of $\mathbf{B}$ can then be written as

$$p_\mathbf{B}(\mathbf{y}) = p_{\boldsymbol{\xi}}([A]_{\rm pi}^{-1}(\mathbf{y} - \mathbf{m}_{\rm B})),\ \mathbf{y} \in \mathbb{R}^{n\mu} \tag{B.7}$$

with $p_{\boldsymbol{\xi}}$ the canonical Gaussian multivariate PDF:

$$p_{\boldsymbol{\xi}} : \mathbf{z} \mapsto (2\pi)^{-\frac{\ell}{2}} \exp\left(-\frac{1}{2}\mathbf{z}^t\mathbf{z}\right). \tag{B.8}$$

Below, $[A]_{\rm pi}^{-1}$ is calculated as a function of $[\widehat{C}_\mathbf{B}]$.

In general, the identification of $[A]$ is obtained from the spectral decomposition of the covariance estimate $[\widehat{C}_\mathbf{B}]$, which is equivalent to performing the Principal Component Analysis (PCA) of random vector $\mathbf{B}$. However, in the present case, the number ν_0 of available realization is small compared to the dimension $n\mu$ of $\mathbf{B}$, which may cause a problem of overlearning. To limit it, a solution proposed in [25] is to perform the spectral decomposition of a matrix $[\widetilde{C}_{\rm B}]$ obtained by partly stationarizing $[\widehat{C}_\mathbf{B}]$. Matrix $[\widetilde{C}_{\rm B}]$ is defined as the linear combination

$$[\widetilde{C}_{\rm B}] = \alpha[\widehat{C}_{\rm B}] + (1-\alpha)[C_{\rm B}^{\rm stat}] \tag{B.9}$$

where $\alpha \in [0,1]$ is a weighting coefficient and $[C_{\rm B}^{\rm stat}]$ the stationarization of $[\widehat{C}_\mathbf{B}]$. Stationarizing consists in averaging on the diagonals of the matrix in order to obtain the

covariance matrix of a stationary process. Process $\mathbf{B}$ has n components, concatenated in the discretized version of the process as shown in Eq. (B.1). One should be careful to apply the stationarization independently on each submatrix of $[\widehat{C}_{\mathbf{B}}]$ corresponding to each component or correlation between two components, in the following way:

$$[M^{\mathrm{stat}}]_{ij} = \begin{cases} \dfrac{1}{\mu} \displaystyle\sum_{k=1}^{\mu-j+i} [M]_{k,k+j-i} & \text{if } i \leq j\,, \\ \dfrac{1}{\mu} \displaystyle\sum_{k=1}^{\mu-i+j} [M]_{k+i-j,k} & \text{if } i \geq j\,, \end{cases} \tag{B.10}$$

where $[M]$ stands for any square matrix of size $\mu \times \mu$ and $[M^{\mathrm{stat}}]$ its stationarized version.

After the spectral decomposition of $[\widetilde{C}_{\mathrm{B}}]$, only the m eigenvectors of highest eigenvalues are kept. They are gathered in columns in matrix $[V]$; the corresponding eigenvalues are gathered on the diagonal of diagonal matrix $[D]$. The basis is truncated in order to consider only the most statistically significant eigenvectors. Indeed, we observed that as the value of the eigenvalues decreases, the eigenvectors display characteristics similar to white noise. We choose $m \leq \nu_0$, which has the advantage to make the estimated covariance matrix of the projection coefficients invertible (see Eq. (B.12)).

Coefficient α is determined by minimizing the Leave-One-Out error of projection (see [25]) of the realizations $\{\mathbf{b}^{\mathrm{ref},i}\}_i$ on the basis $[V]$ obtained by the spectral decomposition of $[\widetilde{C}_{\mathrm{B}}]$, which depends on α.

Random vector $\mathbf{B}$ can then be represented as

$$\mathbf{B} = \widehat{\mathbf{m}}_{\mathrm{B}} + [V][D]^{\frac{1}{2}}\boldsymbol{\gamma} \tag{B.11}$$

where $\boldsymbol{\gamma}$ is the centered random vector gathering the projection coefficients of $\mathbf{B}$. For standard PCA, the components of $\boldsymbol{\gamma}$ are uncorrelated, but it is not the case here because of the stationarization. Thus it becomes necessary to estimate the covariance matrix of $\boldsymbol{\gamma}$

$$[\widehat{C}_{\gamma}] = \frac{1}{\nu_0 - 1}[\gamma][\gamma]^t = [L][L]^t \tag{B.12}$$

where $[\gamma] = [D]^{-\frac{1}{2}}[V]^t[\bar{B}]$ gathers the realizations of $\boldsymbol{\gamma}$ and $[L][L]^t$ corresponds to the Cholesky factorization of $[\widehat{C}_{\gamma}]$.

Finally, we get $[A] = [V][D]^{\frac{1}{2}}[L]$ and $[A]^{-1}_{\mathrm{pi}} = [L]^{-1}[D]^{-\frac{1}{2}}[V]^t$.

Appendix C. Transitional Markov Chain Monte Carlo

The TMCMC algorithm [18, 19] is a MCMC algorithm designed to sample from the posterior PDF knowing the prior PDF and the likelihood function in a Bayesian framework. It is based on the following equation giving the relationship between these quantities:

$$p^{\mathrm{post}}(\mathbf{x}) \propto \mathcal{L}(\mathbf{x})\, p^{\mathrm{prior}}(\mathbf{x}) \tag{C.1}$$

The particularity of TMCMC is to start from a sample distributed according the prior PDF, and making it gradually evolve toward a sample distributed according to the posterior PDF. The algorithms work in m steps by sampling successively from the distributions p_j defined for $0 \leq j \leq m$ and $\mathbf{x} \in \mathcal{X}$ by

$$p_j(\mathbf{x}) \propto \mathcal{L}(\mathbf{x})^{q_j}\, p^{\mathrm{prior}}(\mathbf{x}) \tag{C.2}$$

where $0 = q_0 < q_1 < \cdots < q_{m-1} < q_m = 1$. One can immediately notice that $p_0 = p^{\mathrm{prior}}$ and $p_m = p^{\mathrm{post}}$.

At the beginning of each step j, $1 \leq j \leq m$, the sample of size N_{s} is distributed as p_{j-1}. Each point $\mathbf{x}_{j-1,k}$ of the sample is then affected a weighting coefficient $w_{j,k} = \mathcal{L}(\mathbf{x}_{j-1,k})^{q_j - q_{j-1}}$, $1 \leq k \leq N_{\mathrm{s}}$. A new sample of size N_{s} is drawn from $\{\mathbf{x}_{j-1,k}\}_k$ according to these weighting coefficients. This new sample is distributed as p_j. In order to avoid the repetition of identical elements in the new sample, MCMC steps are applied to disturb the sample while keeping the same distribution. The Metropolis-Hastings algorithm is used to draw the proposals for the MCMC steps: a Gaussian distribution around the previous point of the Markov Chain. Its covariance matrix is estimated from the sample $\{\mathbf{x}_{j-1,k}\}_k$. A factor β is introduced to control the step size. The algorithm can be summed up as follows.

Initialization: set $j = 1$, draw a sample $\{\mathbf{x}_{0,k}\}_{1 \leq k \leq N_{\mathrm{s}}}$ of N_{s} distributed as p^{prior}.

Iterations: while $q_{j-1} < 1$,

1. Determine the optimal value for q_j by solving

$$q_j = \underset{q \in]q_{j-1}, 1]}{\arg\min} \left| \mathrm{CV}_{j-1}(q) - 1 \right| \tag{C.3}$$

where $\mathrm{CV}_j(q)$ is the coefficient of variation (equal to the standard deviation divided by the mean) of set $\{\mathcal{L}(\mathbf{x}_{j,k})^{q-q_j}\}_{1\leq k\leq N_\mathrm{s}}$. If $q_j > 1$, set it to 1.

2. For $1 \leq k \leq N_\mathrm{s}$, compute the weighting coefficient

$$w_{j,k} = \mathcal{L}(\mathbf{x}_{j-1,k})^{q_j - q_{j-1}} \tag{C.4}$$

and normalize it

$$\bar{w}_{j,k} = \frac{w_{j,k}}{\sum_{\ell=1}^{N_\mathrm{s}} w_{j,\ell}} \tag{C.5}$$

3. Compute the covariance matrix for the proposal distribution

$$[\Sigma_j] = \beta^2 \sum_{k=1}^{N_\mathrm{s}} \bar{w}_{j,k}(\mathbf{x}_{j-1,k} - \mathbf{m}_j)(\mathbf{x}_{j-1,k} - \mathbf{m}_j)^t \tag{C.6}$$

with $\mathbf{m}_j = \sum_{k=1}^{N_\mathrm{s}} \bar{w}_{j,k}\mathbf{x}_{j-1,k}$.

4. Initialize $\{\mathbf{x}_k^\mathrm{c}\}_{1\leq k\leq N_\mathrm{s}}$ such that $\mathbf{x}_k^\mathrm{c} = \mathbf{x}_{j-1,k}$ for any k.

5. Build the sample $\{\mathbf{x}_{j,k}\}_{1\leq k\leq N_\mathrm{s}}$ by iterating as following : for k going from 1 to N_s,

   (a) Select index ℓ between 1 and N_s, with a probability given by the weight $\bar{w}_{j,\ell}$;

   (b) Draw a proposal $\mathbf{x}^*$ from a normal distribution $\mathcal{N}(\mathbf{x}_\ell^\mathrm{c}; [\Sigma_j])$;

   (c) Draw r from a uniform distribution on $[0, 1]$;

   (d) If $r \leq \frac{p_j(\mathbf{x}^*)}{p_j(\mathbf{x}_\ell^\mathrm{c})}$ then set $\mathbf{x}_\ell^\mathrm{c} = \mathbf{x}^*$, otherwise do nothing;

   (e) Set $\mathbf{x}_{j,k} = \mathbf{x}_\ell^\mathrm{c}$.

6. Increase j.

References

[1] Y. Marzouk, D. Xiu, A stochastic collocation approach to bayesian inference in inverse problems, Communications in Computational Physics 6 (4) (2009) 826–847.

[2] M. Arnst, R. Ghanem, C. Soize, Identification of bayesian posteriors for coefficients of chaos expansions, Journal of Computational Physics 229 (2010) 3134–3154.

[3] M. Frangos, Y. Marzouk, K. Willcox, B. van Bloemen Waanders, Surrogate and reduced-order modeling: A comparison of approaches for large-scale statistical inverse problems, Large-Scale Inverse Problems and Quantification of Uncertainty (2010) 123–149.

[4] J. B. Nagel, B. Sudret, Spectral likelihood expansions for Bayesian inference, Journal of Computational Physics 309 (2016) 267–294.

[5] G. H. Givens, J. A. Hoeting, Computational statistics, 2nd edition, John Wiley & Sons, 2013.

[6] G. Perrin, Adaptive calibration of a computer code with time-series output, Journal of Statistical Planning and Inference.

[7] P. Ranjan, M. Thomas, H. Teismann, S. Mukhoti, Inverse Problem for a Time-Series Valued Computer Simulator via Scalarization, Open Journal of Statistics (June) (2016) 528–544.

[8] M. T. Pratola, S. R. Sain, D. Bingham, M. Wiltberger, E. J. Rigler, Fast sequential computer model calibration of large nonstationary spatial-temporal processes, Technometrics 55 (2) (2013) 232–242.

[9] G. Perrin, D. Duhamel, C. Soize, C. Funfschilling, Quantification of the influence of the track geometry variability on the train dynamics, Mechanical Systems and Signal Processing 60 (2015) 945–957.

[10] N. Lestoille, C. Soize, C. Funfschilling, Sensitivity of train stochastic dynamics to long-time evolution of track irregularities, Vehicle System Dynamics 54 (5) (2016) 545–567.

[11] N. Lestoille, C. Soize, C. Funfschilling, Stochastic prediction of high-speed train dynamics to long-term evolution of track irregularities, Mechanics Research Communications 75 (2016) 29–39.

[12] A. J. Bing, A. Gross, Development of railroad track degradation models, Transportation research record 939 (1983) 27–31.

[13] G. Perrin, C. Soize, D. Duhamel, C. Funfschilling, Track irregularities stochastic modeling, Probabilistic Engineering Mechanics 34 (2013) 123–130.

[14] S. Kraft, Parameter identification for a TGV model, Ph.D. thesis, Ecole Centrale Paris (2012).

[15] B. Matérn, Spatial variation, 2nd edition, Springer-Verlag, 1986.

[16] W. Scott, P. I. Frazier, W. B. Powel, The Correlated Knowledge Gradient for Simulation Optimization of Continuous Parameters using Gaussian Process Regression, SIAM Journal on Optimization 21 (3) (2011) 996–1026.

[17] W. . K. Hastings, Monte Carlo Sampling Methods Using Markov Chains and Their Applications, Biometrika 57 (1) (1970) 97–109.

[18] J. Ching, Y.-C. Chen, Transitional Markov Chain Monte Carlo Method for Bayesian Model Updating, Model Class Selection, and Model Averaging, Journal of Engineering Mechanics 133 (7) (2007) 816–832.

[19] W. Betz, I. Papaioannou, D. Straub, Transitional Markov Chain Monte Carlo: Observations and Improvements, Journal of Engineering Mechanics 142 (5) (2016) 04016016.

[20] D. R. Jones, M. Schonlau, W. J. Welch, Efficient Global Optimization of Expensive Black-Box Functions, Journal of Global Optimization 13 (1998) 455–492.

[21] V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, N.-H. Kim, Adaptive Designs of Experiments for Accurate Approximation of a Target Region, Journal of Mechanical Design 132 (7) (2010) 071008.

[22] T. Santner, B. Williams, W. Notz, The Design and Analysis of Computer Experiments, Springer-Verlag, Berlin, New York, 2003.

[23] J. Sacks, W. J. Welch, J. S. B. Mitchell, P. W. Henry, Design and Experiments of Computer Experiments, Statistical Science 4 (4) (1989) 409–423.

[24] V. C. P. Chen, K. L. Tsui, R. R. Barton, J. K. Allen, A review of design and modeling in computer experiments, Statistics in industry 22 (2003) (2003) 231–261.

[25] G. Perrin, C. Soize, D. Duhamel, C. Funfschilling, A Posteriori error and optimal reduced basis for stochastic processes defined by a finite set of realizations, SIAM/ASA Journal on Uncertainty Quantification 2 (2014) 745–762.