POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Gaussian process emulators to accelerate sensitivity analysis and Bayesian parameter estimation: application to cardiovascular modeling

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Author: **Alessandro Pirozzi**

Student ID: 946326
Advisor: Prof. Alfio Maria Quarteroni
Co-advisors: Stefano Pagani, Francesco Regazzoni
Academic Year: 2021-22

Abstract

Numerical models have become popular in many engineering fields, even if they present several limitations in terms of computational cost, making their usage often complicated in practical circumstances.
The focus of this work is on overcoming these limitations by employing a non-parametric regression method, namely the Gaussian process (GP) regression, with the aim of building fast and reliable surrogate models. We test this method on some benchmark test cases in order to investigate its performances in view of our application, i.e. a lumped parameter closed-loop model for the whole circulatory network. This model is characterized by a high number of parameters and outputs, making the usage of an anisotropic GP promising.
We show the effectiveness of the proposed method on two relevant contexts in uncertainty quantification for cardiac modeling. First, we employ a GP based method to perform a global sensitivity analysis on a lumped parameter closed-loop model for the hemodynamics of the circulatory system. Second, we perform a Bayesian estimation of parameters starting from noisy measurements of some scalar outputs by means of the Markov chain Monte Carlo (MCMC) method. In both cases, the idea is to replace the high-fidelity circulation model with the GP based surrogate model to perform all the numerical simulations that would be otherwise computationally intense.
In this way, we will prove that our GP emulator is able to speed up the numerical simulations reducing the computational time.

Keywords: Machine learning, Gaussian process, Circulation model, Global sensitivity analysis, Bayesian parameter estimation, Markov chain Monte Carlo method

Abstract in lingua italiana

I modelli numerici sono diventati particolarmente popolari in molti campi dell'ingegneria, anche se presentano diverse limitazioni in termini di costo computazionale, rendendo il loro utilizzo spesso complicato in circostanze pratiche.
L'obiettivo di questa tesi è quello di risolvere tali limiti utilizzando un metodo di regressione non parametrica, cioè la regressione del processo Gaussiano (GP), con l'obiettivo di costruire modelli surrogati veloci ed affidabili. Testiamo questo metodo su alcuni casi test per indagare le sue prestazioni in vista della nostra applicazione, cioè un modello lumped parameter closed-loop per l'intera rete circolatoria. Questo modello è caratterizzato da un elevato numero di parametri e output, rendendo incoraggiante l'utilizzo di un GP anisotropo.
Dimostriamo l'efficacia del metodo proposto in due contesti molto importanti nella quantificazione dell'incertezza per la modellazione cardiaca. Innanzitutto, utilizziamo un metodo basato sul processo Gaussiano per eseguire un'analisi di sensibilità globale su un modello lumped parameter closed-loop per l'emodinamica del sistema circolatorio. In secondo luogo, eseguiamo una stima Bayesiana dei parametri partendo da misure rumorose di alcuni output scalari mediante il metodo Markov chain Monte Carlo (MCMC). In entrambi i casi, l'idea è quella di sostituire il modello matematico di circolazione con il modello surrogato basato sul processo Gaussiano per eseguire tutte le simulazioni numeriche che altrimenti risulterebbero computazionalmente costose.
In questo modo, proveremo che il nostro emulatore GP è in grado di velocizzare le simulazioni numeriche riducendo il tempo computazionale.

Parole chiave: Machine learning, Processo Gaussiano, Modello di circolazione, Analisi di sensibilità globale, Stima Bayesiana dei parametri, Metodo Markov chain Monte Carlo

Contents

Introduction

Numerical models have been successfully applied to many complex engineering fields with the aim of studying the behaviour of systems whose mathematical models are too complicated to provide analytical solutions. However, they are often limited in several contexts because of the high computational costs associated with high-fidelity models based on partial differential equations.
Some computational challenges may arise when considering the impact of uncertainty on input and output variables of numerical models, such as in sensitivity analysis or in inverse problems resolution. Indeed, they are contexts in which it is possible to quantitatively study how the model response is affected by uncertainty or to better understand the underlying phenomenon by highlighting the interactions between variables. Moreover, a large number of simulations is usually required to perform reliable analyses.
One way to significantly reduce the computational cost of numerical models is to employ computationally cheap surrogate models instead of computationally expensive high-fidelity models. The computational cost can be greatly reduced by exploiting surrogate models, since the efforts required in building such a models are much lower than in the usual approach in which the high-fidelity model is used [31].
In this framework, non-parametric regression methods have aroused significant interest, since they are completely data-driven and do not need any explicit knowledge about the functional relationship between variables. Gaussian process regression is one of such a techniques; indeed, it is completely data-driven and provides a suitable Bayesian framework for modeling [19]. Other Bayesian methods can also provide probabilistic predictions, such as Bayesian artificial neural networks (ANN) [17], Bayesian linear regression [14] and Bayesian least-squares support vector machines (LS-SVM) [27]. However, Gaussian processes result particularly interesting for several reasons:

- they are able to specify prior distributions over functions;
- their implementation is flexible, since it only accounts for the optimization of covariance function hyperparameters;
- they provide a reasonable trade-off between computational costs and accuracy of

predictions.

In this work, we aim to investigate these features of Gaussian processes in view of our application, i.e. a lumped parameter closed-loop model for the simulation of the circulatory system. This high-fidelity model was implemented by describing the hemodynamics as an electric circuit, where the governing ordinary differential equations are solved by means of a Runge-Kutta algorithm of 4^{th} order [12]. However, even if this model is able to provide reliable results, it appears inefficient when analyzing the impact of uncertainty on input and output variables because of the high computational burden required. For this reason, our goal is to prove that it is possible to perform reliable analyses and significantly reduce the computational costs by employing Gaussian processes in place of this high-fidelity circulation model.

Outline

This work is structured as follows:

- in Chapter 1 we provide an overview of the circulatory system and the definition of the high-fidelity model used in hemodynamics to reproduce the whole circulatory network for the numerical experiments;
- in Chapter 2 we highlight all the main characteristics of Gaussian processes by addressing simple test cases with the aim of better understanding how to manage them in more complex prediction problems, such as the circulation model mentioned above;
- in Chapter 3 we employ Gaussian process predictions to solve problems related to the context of uncertainty quantification, such as sensitivity analysis, which allows us to rank the most influential input parameters;
- in Chapter 4 we exploit Gaussian process predictions to reduce the computational effort required by the Markov chain Monte Carlo method, which is used to solve the inverse problem for parameters estimation;
- in Chapter 5 we sum up the conclusions of this work, addressing its limitations and possible future developments.

1 | Modeling of the cardiovascular system

In this chapter, we highlight the main characteristics of the high-fidelity circulation model considered in this work. First, we provide a brief introduction to the anatomy and physiology of the cardiovascular system (for more information, we refer to [26]); second, we detail the model used for the hemodynamics of the whole circulatory network.

1.1. Circulatory system

The circulatory system is divided into the systemic circulation, in which the arteries carry the oxygenated blood ejected by the left heart and the veins allow the non-oxygenated blood to come back to the right heart, and the pulmonary circulation, in which the non-oxygenated blood ejected by the right heart flows in the pulmonary arteries towards the lungs and goes back (oxygenated) to the left heart through the pulmonary veins. All this process is sustained by the heart which pumps the blood into arteries and veins, as shown in Figure 1.1.

As reported in [7], both pulmonary and systemic circulations are composed by arteries, veins and microvasculature. Arteries are responsible to carry the blood from the heart to all the organs, whereas veins are the vessels that bring the blood from organs to the heart. For what concerns the exchange of nutrients, oxygen and waste between circulation and tissues, it is performed by microvasculature and capillaries.

The presence of muscular fibers in the heart tissue permits the contraction process which sustains the circulation of the blood. In particular, the contraction creates a pressure wave that propagates along the whole circulatory system by exploiting the elasticity properties of largest arteries. This ability is named vessels compliance and is associated to the smoothing of the blood flow guaranteeing an almost continuous exchange of oxygen with the tissues (windkessel effect).

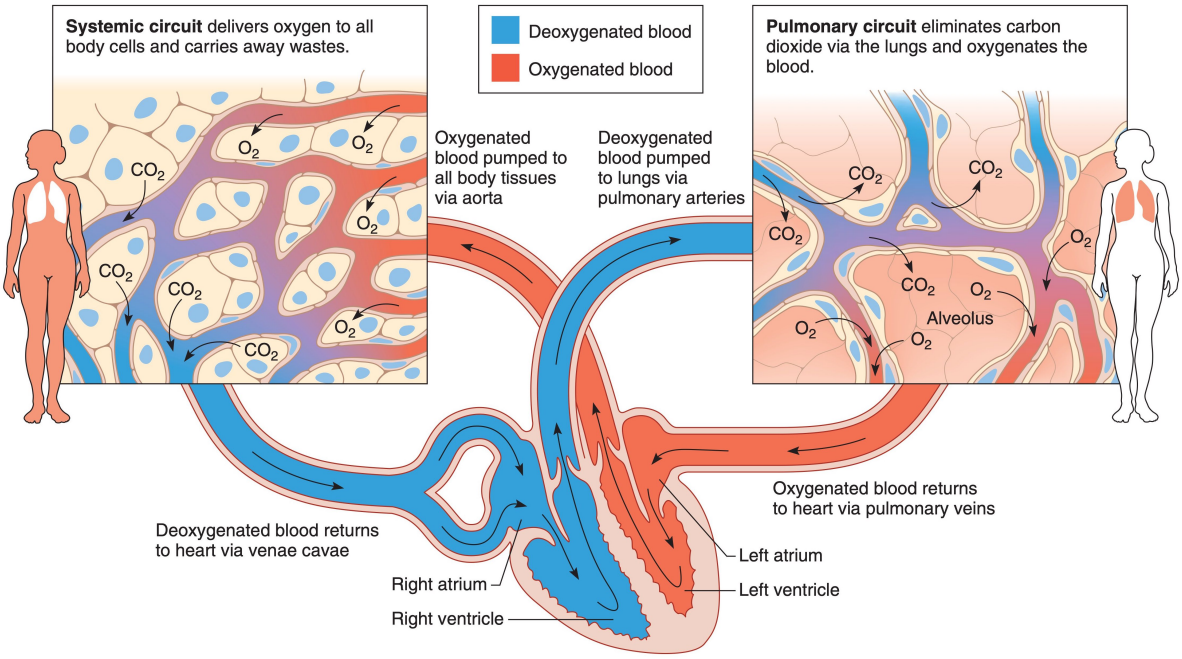


Figure 1.1: Schematic representation of the cardiovascular system (image from [26]).

1.1.1. Anatomy and physiology of the heart

As pointed out in [7], the heart is internally divided into four chambers, two on the left and two on the right, as shown in Figure 1.2. The upper chambers are called atria (where the blood enters from the veins), whereas the lower chambers are called ventricles (where we have the ejection to the arteries).
The cardiac valves separate atria by ventricles and ventricles by pulmunary trunk and aorta. Valves do not allow blood backflow in ventricular contraction. In particular, we have:

- the mitral valve in the left side of the heart;
- the tricuspid valve in the right side of the heart;
- the aortic valve at the exit of the left ventricle;
- the pulmonary valve at the exit of the right ventricle.

The conduction system plays a key role for the muscular contraction of the heart. Indeed, it allows the propagation of the potential which is responsible of the heart contraction by means of its main components, namely the sinoatrial node, often called the pacemaker, the atrioventricular node and the Purkinje fibers network.

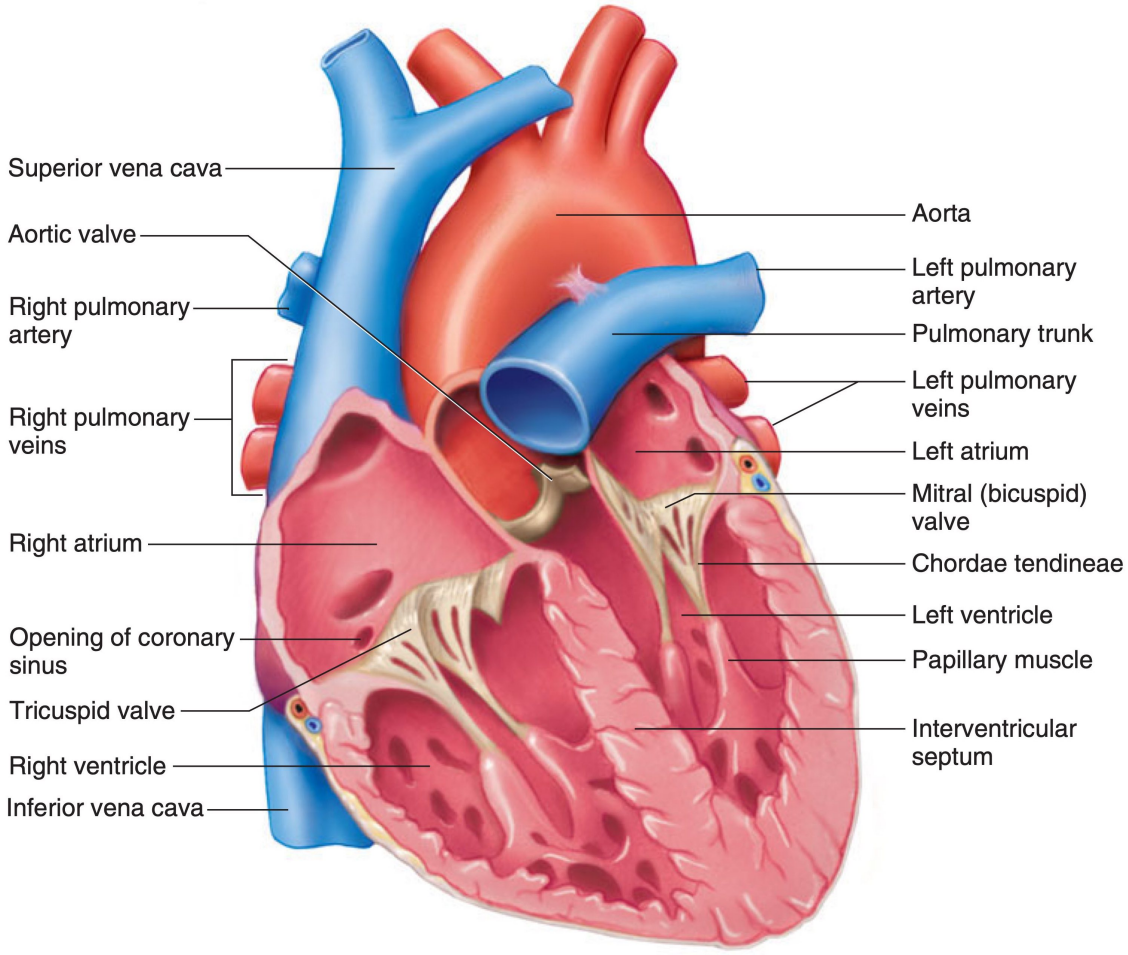


Figure 1.2: Frontal section of the heart (image from [26]).

1.1.2. Cardiac cycle

As reported in [7], the cardiac cycle begins with the contraction of the atria and ends with the ventricular relaxation, as shown in Figure 1.3. Two different moments are responsible of the process total time: the systole, in which we have a contraction and thus a blood ejection, and the diastole, in which we have a relaxation and thus a chamber filling.
During the cardiac cycle, we have that the pressure in heart chambers continuously rises and falls. In particular, the difference of pressure between atria and ventricles makes the atrioventricular valves open by pushing down the leaflets. In this way, since the blood flows from the veins into the atria, it is possible to observe the atrial systole, which is a contraction that ejects the blood to the ventricle. At this point, the atrioventricular valves close and the atria start filling of blood, while ventricles begin their contraction and, consequently, the ejection of blood into the circulation.

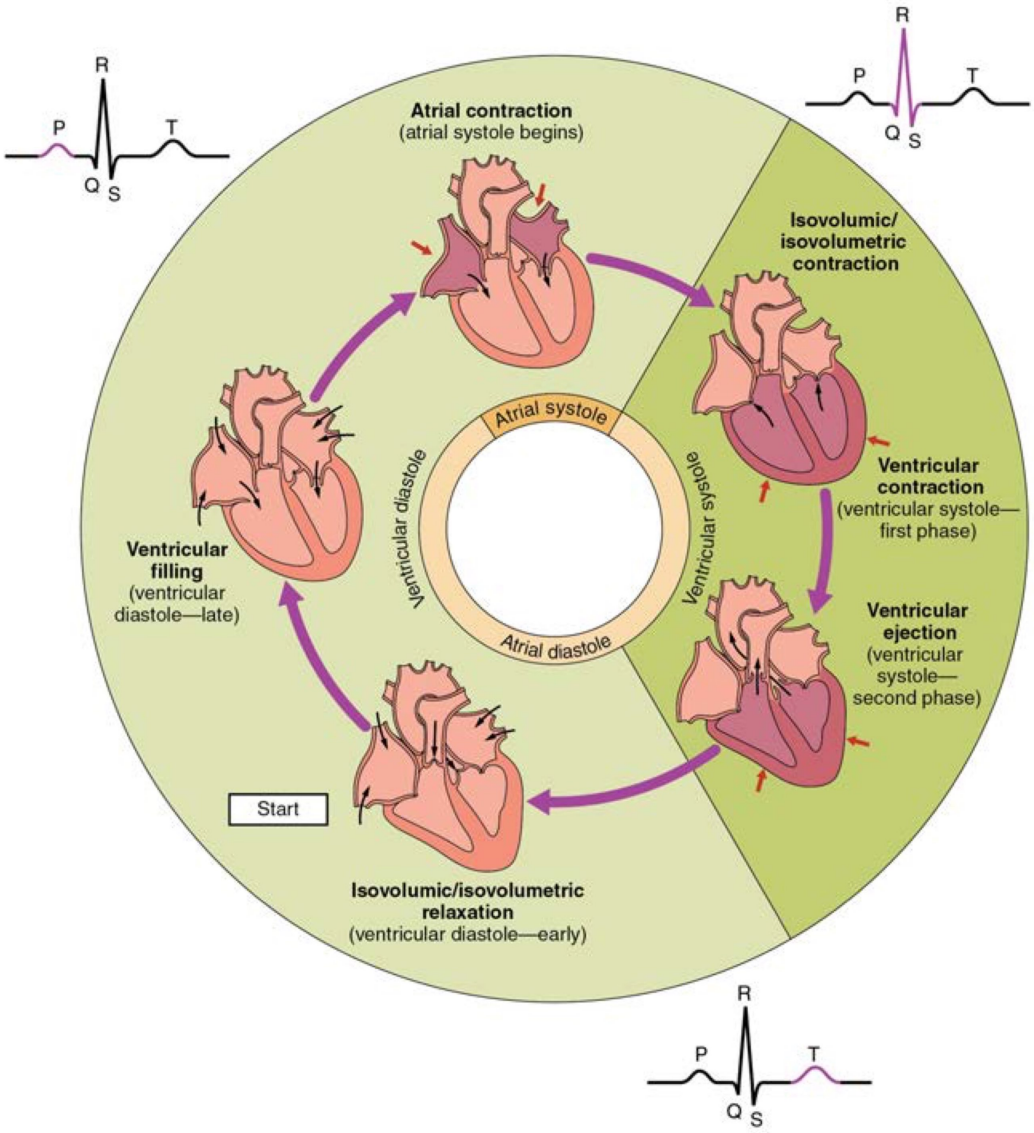


Figure 1.3: Events occurring in the cardiac cycle (image from [3]).

As anticipated before, the contraction process is due to the electrical activity of the heart. The sinoatrial node starts the excitation and the propagation of the signal through the atria muscle fibers (atrial depolarization), inducing the contraction of the atria. Then we have the propagation of the signal through the Purkinje fibers by means of the atrioventricular node excitation. Here the conduction along the myocardial cells is significantly slower, so that a homogeneous propagation of the electric potential along the wall of the ventricle (ventricular depolarization) is guaranteed. In conclusion, the ventricular repolarization concludes the systolic phase, leading to the relaxation of the ventricle (ventricular diastole) in which the cardiac cycle finishes.

1.2. 0D models in hemodynamics

As explained in [7], numerical modeling of the cardiovascular system is a computationally expensive problem to be solved because of its highly complicated structure. In particular, a complete simulation of the system, considering all the arteries, veins and capillaries, would be unfeasible. For this reason, some high-fidelity reduced models can be introduced in order to minimize the computational burden, such as 0D models or lumped parameter models. These approaches, based on geometrical reduction, highly simplify the description of the behaviour of spatially distributed physical systems, such as the circulation system, by introducing discrete elements that approximate the behaviour of the whole system under certain assumptions. Mathematically speaking, this simplification allows us to turn the PDEs of the time and space dependent model of the physical system into ODEs which depend only on time.
This kind of models is particularly popular to describe electric circuits; for this reason, it is possible to establish an analogy between the circulatory network and an electric circuit, as shown in Figures 1.4–1.5.

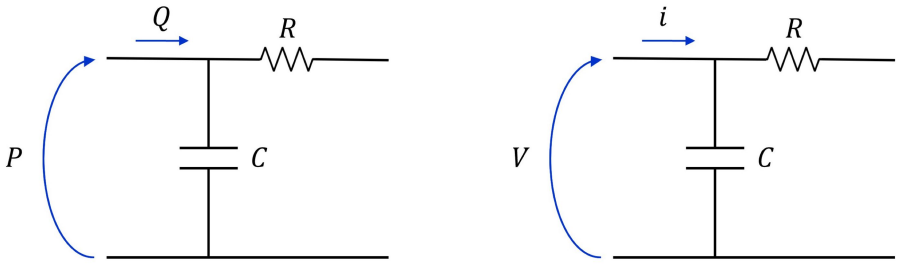


Figure 1.4: Electric analogy of the circulation system of ODEs (image from [7]).

The variables of this problem are:

- the blood flow rate Q, which plays the role of a current;
- the pressure P, which can be associated to an electric potential.

We also need to provide a physical meaning for the electric elements we typically find in a circuit. Indeed, we have:

- the resistance R, which models the dissipation due to the fluid viscosity;
- the capacitance C, which models the vessel compliance due to the elasticity of the wall;
- the inductance L, which models the inertial properties of the fluid;

- the diode R_V, which models the time dependent obstructions in the system (e.g., cardiac valves).

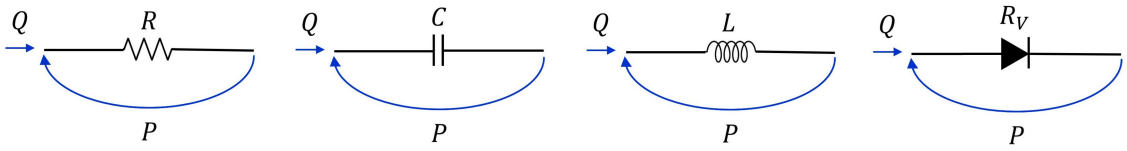


Figure 1.5: Elements commonly used in the circuital representation of the 0D model (image from [7]).

1.2.1. Lumped parameter closed-loop model

In order to model the hemodynamics of the whole circulatory network, we consider a lumped parameter closed-loop model proposed in [21]. In particular,

- the systemic and pulmonary circulations are modeled with resistance-inductance-capacitance (RLC) circuits, one for the arterial part and the other one for the venous part;
- the four chambers are modeled by time-varying elastance elements;
- the four valves are represented as non-ideal diodes.

The variables of the system related to the heart chambers are listed below:

- pressure $p_{\mathrm{RA}}(t)$, volume $V_{\mathrm{RA}}(t)$ and elastance $E_{\mathrm{RA}}(t)$ for the right atrium;
- pressure $p_{\mathrm{LA}}(t)$, volume $V_{\mathrm{LA}}(t)$ and elastance $E_{\mathrm{LA}}(t)$ for the left atrium;
- pressure $p_{\mathrm{RV}}(t)$, volume $V_{\mathrm{RV}}(t)$ and elastance $E_{\mathrm{RV}}(t)$ for the right ventricle;
- pressure $p_{\mathrm{LV}}(t)$, volume $V_{\mathrm{LV}}(t)$ and elastance $E_{\mathrm{LV}}(t)$ for the left ventricle.

The elastaces vary in a limited range of values:

$$E_j(t) \in (E_j^{\mathrm{pass}}, E_j^{\mathrm{pass}} + E_j^{\mathrm{act,max}}), \quad j \in \{\mathrm{RA}, \mathrm{LA}, \mathrm{RV}, \mathrm{LV}\}, \tag{1.1}$$

where the minimum corresponds to the moment in which the chambers are at rest and the maximum corresponds to the fully contracted case.

Then we have the fluxes which are related to the valves:

- the mitral valve flux $Q_{\mathrm{MV}}(t)$;
- the aortic valve flux $Q_{\mathrm{AV}}(t)$;
- the tricuspid valve flux $Q_{\mathrm{TV}}(t)$;
- the pulmonary valve flux $Q_{\mathrm{PV}}(t)$.

For what concerns the systemic and pulmonary circulations, we can find four couples of pressure and flux:

- pressure $p_{\mathrm{VEN}}^{\mathrm{PUL}}(t)$ and flux $Q_{\mathrm{VEN}}^{\mathrm{PUL}}(t)$ for the pulmonary veins;
- pressure $p_{\mathrm{AR}}^{\mathrm{PUL}}(t)$ and flux $Q_{\mathrm{AR}}^{\mathrm{PUL}}(t)$ for the pulmonary arteries;
- pressure $p_{\mathrm{VEN}}^{\mathrm{SYS}}(t)$ and flux $Q_{\mathrm{VEN}}^{\mathrm{SYS}}(t)$ for the systemic veins;
- pressure $p_{\mathrm{AR}}^{\mathrm{SYS}}(t)$ and flux $Q_{\mathrm{AR}}^{\mathrm{SYS}}(t)$ for the systemic arteries.

Notice that each variable is only time-dependent because of the geometric reduction of the problem.
Finally, the heart valves are modeled with the use of non-ideal diodes:

$$R_j(p_1, p_2) = \begin{cases} R_{\min}, & p_1 < p_2 \\ R_{\max}, & p_1 \geq p_2 \end{cases} \qquad j \in \{\mathrm{MV}, \mathrm{AV}, \mathrm{TV}, \mathrm{PV}\}, \tag{1.2}$$

where p_1 and p_2 denote the pressures ahead and behind the valve leaflets with respect to the flow direction, whereas $R_{\min}$ and $R_{\max}$ are the minimum and maximum resistance of the valves. In particular, one would have $R_{\min} = 0$ and $R_{\max} = +\infty$ for an idealized valve. By setting $R_{\min} > 0$, one has dissipation of mechanical energy taking place when the blood flows through the opened valve. On the other hand, it is set $R_{\max} < +\infty$ sufficiently large so that blood leakage is negligible when the valve is closed (for more information about the lumped parameter closed-loop model, we refer to [4, 10]).

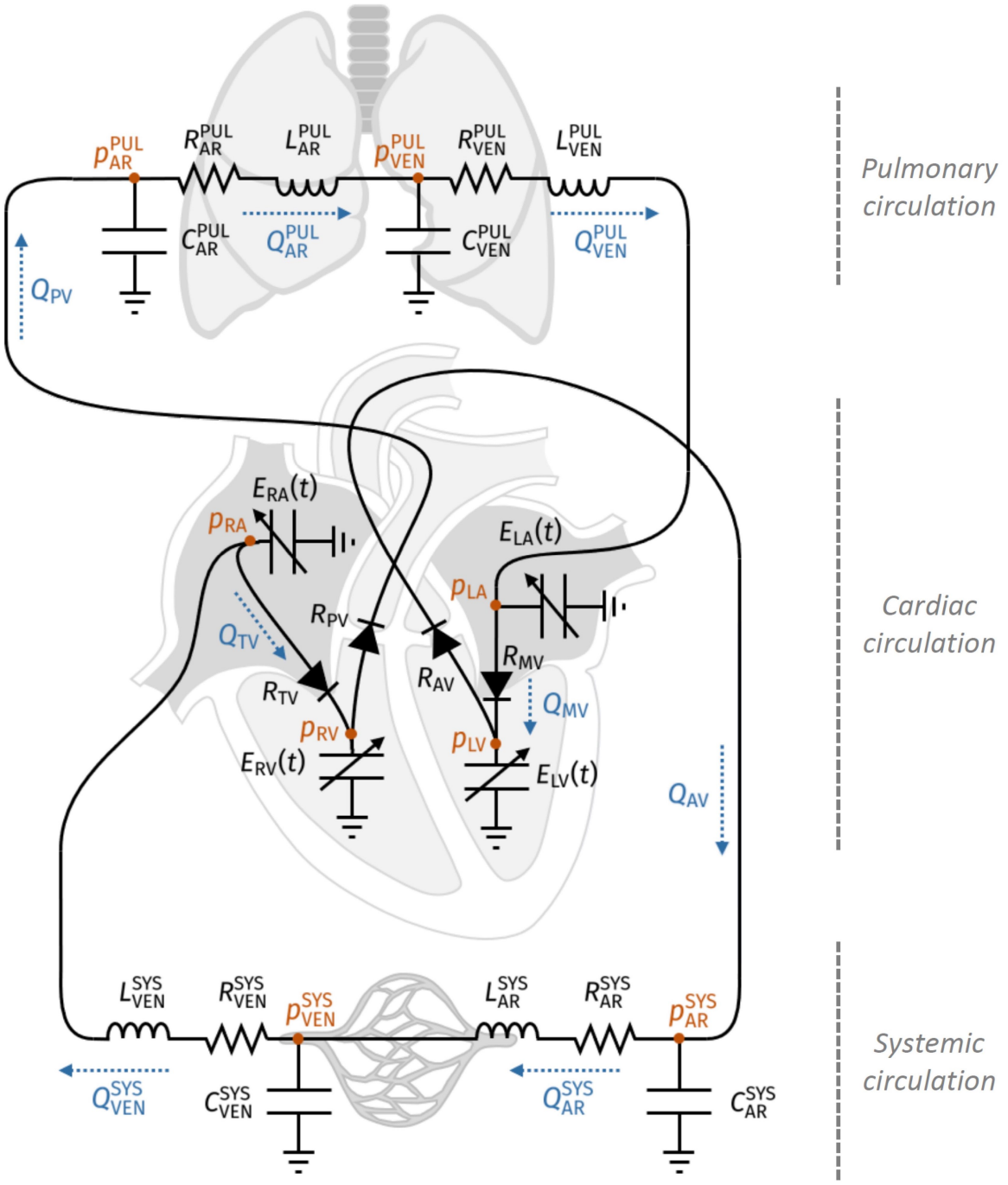


Figure 1.6: 0D lumped parameter closed-loop model (image from [23]).

Given these definitions, the circulation model is written as follows:

$$
\begin{cases}
\dfrac{dV_{\mathrm{LA}}(t)}{dt} = Q_{\mathrm{VEN}}^{\mathrm{PUL}}(t) - Q_{\mathrm{MV}}(t) \\
\dfrac{dV_{\mathrm{LV}}(t)}{dt} = Q_{\mathrm{MV}}(t) - Q_{\mathrm{AV}}(t) \\
\dfrac{dV_{\mathrm{RA}}(t)}{dt} = Q_{\mathrm{VEN}}^{\mathrm{SYS}}(t) - Q_{\mathrm{TV}}(t) \\
\dfrac{dV_{\mathrm{RV}}(t)}{dt} = Q_{\mathrm{TV}}(t) - Q_{\mathrm{PV}}(t) \\
C_{\mathrm{AR}}^{\mathrm{SYS}} \dfrac{dp_{\mathrm{AR}}^{\mathrm{SYS}}(t)}{dt} = Q_{\mathrm{AV}}(t) - Q_{\mathrm{AR}}^{\mathrm{SYS}}(t) \\
C_{\mathrm{AR}}^{\mathrm{PUL}} \dfrac{dp_{\mathrm{AR}}^{\mathrm{PUL}}(t)}{dt} = Q_{\mathrm{PV}}(t) - Q_{\mathrm{AR}}^{\mathrm{PUL}}(t) \\
C_{\mathrm{VEN}}^{\mathrm{SYS}} \dfrac{dp_{\mathrm{VEN}}^{\mathrm{SYS}}(t)}{dt} = Q_{\mathrm{AR}}^{\mathrm{SYS}}(t) - Q_{\mathrm{VEN}}^{\mathrm{SYS}}(t) \\
C_{\mathrm{VEN}}^{\mathrm{PUL}} \dfrac{dp_{\mathrm{VEN}}^{\mathrm{PUL}}(t)}{dt} = Q_{\mathrm{AR}}^{\mathrm{PUL}}(t) - Q_{\mathrm{VEN}}^{\mathrm{PUL}}(t) \\
\dfrac{L_{\mathrm{AR}}^{\mathrm{SYS}}}{R_{\mathrm{AR}}^{\mathrm{SYS}}} \dfrac{dQ_{\mathrm{AR}}^{\mathrm{SYS}}(t)}{dt} = -Q_{\mathrm{AR}}^{\mathrm{SYS}}(t) - \dfrac{p_{\mathrm{VEN}}^{\mathrm{SYS}}(t) - p_{\mathrm{AR}}^{\mathrm{SYS}}(t)}{R_{\mathrm{AR}}^{\mathrm{SYS}}} \\
\dfrac{L_{\mathrm{VEN}}^{\mathrm{SYS}}}{R_{\mathrm{VEN}}^{\mathrm{SYS}}} \dfrac{dQ_{\mathrm{VEN}}^{\mathrm{SYS}}(t)}{dt} = -Q_{\mathrm{VEN}}^{\mathrm{SYS}}(t) - \dfrac{p_{\mathrm{RA}}(t) - p_{\mathrm{VEN}}^{\mathrm{SYS}}(t)}{R_{\mathrm{VEN}}^{\mathrm{SYS}}} \\
\dfrac{L_{\mathrm{AR}}^{\mathrm{PUL}}}{R_{\mathrm{AR}}^{\mathrm{PUL}}} \dfrac{dQ_{\mathrm{AR}}^{\mathrm{PUL}}(t)}{dt} = -Q_{\mathrm{AR}}^{\mathrm{PUL}}(t) - \dfrac{p_{\mathrm{VEN}}^{\mathrm{PUL}}(t) - p_{\mathrm{AR}}^{\mathrm{PUL}}(t)}{R_{\mathrm{AR}}^{\mathrm{PUL}}} \\
\dfrac{L_{\mathrm{VEN}}^{\mathrm{PUL}}}{R_{\mathrm{VEN}}^{\mathrm{PUL}}} \dfrac{dQ_{\mathrm{VEN}}^{\mathrm{PUL}}(t)}{dt} = -Q_{\mathrm{VEN}}^{\mathrm{PUL}}(t) - \dfrac{p_{\mathrm{LA}}(t) - p_{\mathrm{VEN}}^{\mathrm{PUL}}(t)}{R_{\mathrm{VEN}}^{\mathrm{PUL}}}
\end{cases}
\tag{1.3}
$$

with $t \in (0, T)$. In order to close the system, some algebraic relations which use the external pressure p_{EX} are needed. The external pressure models the pressure exerted by the other organs on the heart. For this reason, we use this value to obtain all the chamber pressures. Moreover, we consider for each chamber j a volume $V_{0,j}$ such that the pressure is only equal to the external contribution.

In this way, the following system is obtained:

$$\begin{cases} p_{\mathrm{LV}}(t) = p_{\mathrm{EX}}(t) + E_{\mathrm{LV}}(t)(V_{\mathrm{LV}}(t) - V_{0,\mathrm{LV}}) \\ p_{\mathrm{LA}}(t) = p_{\mathrm{EX}}(t) + E_{\mathrm{LA}}(t)(V_{\mathrm{LA}}(t) - V_{0,\mathrm{LA}}) \\ p_{\mathrm{RV}}(t) = p_{\mathrm{EX}}(t) + E_{\mathrm{RV}}(t)(V_{\mathrm{RV}}(t) - V_{0,\mathrm{RV}}) \\ p_{\mathrm{RA}}(t) = p_{\mathrm{EX}}(t) + E_{\mathrm{RA}}(t)(V_{\mathrm{RA}}(t) - V_{0,\mathrm{RA}}) \\ Q_{\mathrm{PV}}(t) = \dfrac{p_{\mathrm{RV}}(t) - p_{\mathrm{AR}}^{\mathrm{PUL}}(t)}{R_{\mathrm{PV}}(p_{\mathrm{RV}}(t), p_{\mathrm{AR}}^{\mathrm{PUL}}(t))} \\ Q_{\mathrm{AV}}(t) = \dfrac{p_{\mathrm{LV}}(t) - p_{\mathrm{AR}}^{\mathrm{SYS}}(t)}{R_{\mathrm{AV}}(p_{\mathrm{LV}}(t), p_{\mathrm{AR}}^{\mathrm{SYS}}(t))} \\ Q_{\mathrm{TV}}(t) = \dfrac{p_{\mathrm{RA}}(t) - p_{\mathrm{RV}}(t)}{R_{\mathrm{TV}}(p_{\mathrm{RA}}(t), p_{\mathrm{RV}}(t))} \\ Q_{\mathrm{MV}}(t) = \dfrac{p_{\mathrm{LA}}(t) - p_{\mathrm{LV}}(t)}{R_{\mathrm{MV}}(p_{\mathrm{LA}}(t), p_{\mathrm{LV}}(t))} \end{cases} \tag{1.4}$$

For the discretization of this problem a Runge-Kutta algorithm of 4^{th} order is used. However, we do not enter in the specific of this finite difference scheme (for more information, we refer to [22]).

2 | Gaussian processes

Inferring a continuous function from a set of individual (observed or computed) data points is a common task in scientific research [8]. However, not all the processes can be well modeled by simple expressions. For this reason, instead of trying to understand the dependence between inputs and outputs analytically, one may describe the relation between variables by exploiting non-parametric regression methods which are completely data-driven. Indeed, interpolation and regression techniques provide tools to "fill in the space" between data points, resulting in a continuous function representation which, once established, can be efficiently used for each new data point.
In this context, we investigate the performances of Gaussian process models, which have become particularly popular for solving non-linear regression and classification problems [30]. In particular, Gaussian processes are easily interpretable, avoid over-fitting by using prior distributions and have good predictive performances in many empirical cases.

2.1. Gaussian process regression

Let f denote an (unknown) function which maps the input $\mathbf{x} \in \mathcal{X}$ to the output $y \in \mathcal{Y}$, i.e. $f : \mathcal{X} \to \mathcal{Y}$. As reported in [20], a Gaussian process is a collection of random variables, any finite number of which has a joint Gaussian distribution. One of the main characteristics of Gaussian processes is that they are completely identified by a mean function and a covariance function. In particular, the mean function $\mu(\mathbf{x})$ provides the average of all the functions at any point $\mathbf{x}$ of the input space, whereas the covariance function $k(\mathbf{x}, \mathbf{x}')$ sets the correlation between the function values at different input points $\mathbf{x}$ and $\mathbf{x}'$.
Given $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, the mean function $\mu(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ are defined as

$$\begin{aligned} \mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))], \end{aligned} \tag{2.1}$$

so that the Gaussian process can be written as

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{2.2}$$

As noticed in [20], the marginalization property is automatically fulfilled for a Gaussian process, since, by definition, any of its finite collections of random variables are jointly Gaussian. This means that, if the Gaussian process specifies, for instance, $(y_1, y_2) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, then also $y_1 \sim \mathcal{N}(\mu_1, \Sigma_{11})$ is specified, where

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}. \tag{2.3}$$

In order to better understand how Gaussian processes work, [20] provides a very simple example based on the Bayesian linear regression model $f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}$ with prior $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. In this case the mean and covariance become

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})] &= \phi(\mathbf{x})^T \mathbb{E}[\mathbf{w}] = 0, \\ \mathbb{E}[f(\mathbf{x}) f(\mathbf{x}')] &= \phi(\mathbf{x})^T \mathbb{E}[\mathbf{w}\mathbf{w}^T] \phi(\mathbf{x}') = \phi(\mathbf{x})^T \Sigma \phi(\mathbf{x}'), \end{aligned} \tag{2.4}$$

so that $f(\mathbf{x})$ and $f(\mathbf{x}')$ are jointly Gaussian with zero mean and covariance function given by $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \Sigma \phi(\mathbf{x}')$.

2.2. Covariance functions

In this section, we want to discuss different choices of commonly used covariance functions and examine their properties. Covariance functions are crucial ingredients in Gaussian processes, since they encode our assumptions about the function we aim to model. Indeed, they determine the shape, smoothness and other properties of the function of interest. For instance, as reported in [20], covariance functions define the key concept of similarity which states that points with inputs $\mathbf{x}$ close to each other are likely to have a similar target value y. As a consequence, the prediction at a specific test point should be affected by the closeness of training points at that point.

2.2.1. Properties and preliminary notions

As explained in [20], an arbitrary function with input pairs $\mathbf{x}$ and $\mathbf{x}'$ will not be, in general, a valid covariance function, since it must be symmetric and positive semidefinite.

In particular, a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be symmetric if

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x}). \tag{2.5}$$

On the other hand, it is positive semidefinite if

$$\int_{\mathcal{X}\times\mathcal{X}} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) f(\mathbf{x}') d\mu(\mathbf{x}) d\mu(\mathbf{x}') \geq 0, \tag{2.6}$$

for all $f \in L^2(\mathcal{X}, \mu)$, with μ that denotes a measure.
Notice that this kind of functions of two arguments mapping a pair of inputs $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ into $\mathbb{R}$ is called kernel; for this reason, we can also refer to covariance functions as kernel functions.
In [20] some basic notions that could be useful to better understand the main properties of the most commonly used covariance functions are provided. In particular,

- a stationary covariance function is a function of $\mathbf{x} - \mathbf{x}'$, thus it is invariant to translations in the input space;
- an isotropic covariance function is a function only of $|\mathbf{x} - \mathbf{x}'|$, thus it is invariant to all rigid motions;
- a dot product covariance function is a function of $\mathbf{x} \cdot \mathbf{x}'$, thus it is invariant to a rotation of the coordinates around the origin, but not translations in general.

2.2.2. Examples of covariance functions

Ideally, a user would specify a kernel that exactly encodes all the prior beliefs about the properties of the function of interest. However, this is not always the case because the strategy consists of selecting a kernel from a relatively small set of commonly used families of covariance functions depending on the function that we want to model.
In this subsection, some examples of the most widely used covariance functions are given, so that a generic selection criterion is provided.

Linear covariance function

The linear covariance function is one of the simplest covariance functions and is defined as

$$k(\mathbf{x}, \mathbf{x}'; \sigma_0) = \sigma_0^2 + \mathbf{x} \cdot \mathbf{x}'. \tag{2.7}$$

It can be simply obtained from linear regression by setting $\mathcal{N}(0,1)$ priors on the coefficients of $\mathbf{x}$ and a prior of $\mathcal{N}(0,\sigma_0^2)$ on the bias.

Squared exponential covariance function

The most popular and used covariance function is the squared exponential covariance function, also known as the exponentiated quadratic covariance function (or radial basis function). It has the form

$$k(\mathbf{x},\mathbf{x}';\sigma,\lambda) = \sigma^2 \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\lambda^2}\right), \tag{2.8}$$

where σ^2 is called amplitude, that determines the average distance of our function away from its mean, and λ is called length scale, which regulates the speed of decay of the correlation among the points. For this specific covariance function, we can observe that it tends to the amplitude value when inputs are very close, whereas it decreases as their distance in the input space increases.

Rational quadratic covariance function

The rational quadratic covariance function is extremely similar to the squared exponential one. Indeed, it is defined as

$$k(\mathbf{x},\mathbf{x}';\sigma,\lambda,\alpha) = \sigma^2 \left(1+\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\alpha\lambda^2}\right)^{-\alpha}, \tag{2.9}$$

where $\alpha > 0$ is the scale mixture. The rational quadratic covariance function can be interpreted as an infinite sum of different exponentiated quadratic covariance functions with different length scales, where α has the role of determining the weighting between different length scales. Moreover, when $\alpha \to \infty$, the rational quadratic covariance function converges to the squared exponential covariance function.

Periodic covariance function

The periodic covariance function is mainly used to model periodic functions and is defined as

$$k(\mathbf{x},\mathbf{x}';\sigma,\lambda,p) = \sigma^2 \exp\left(-\frac{2}{\lambda^2}\sin^2\left(\pi\frac{\|\mathbf{x}-\mathbf{x}'\|}{p}\right)\right), \tag{2.10}$$

where p represents the period.

Matérn class of covariance functions

The Matérn class of covariance functions is another popular choice. It is defined as

$$k(\mathbf{x}, \mathbf{x}'; \nu, \lambda) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\lambda} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\lambda} \right), \tag{2.11}$$

where $\nu > 0$ is the normalization constant and K_{ν} is a modified Bessel function. The Matérn class of covariance functions has two important properties. The first one is that, when $\nu \to \infty$, it converges to the squared exponential covariance function. The second one is that its expression significantly simplifies for $\nu = 3/2$ and $\nu = 5/2$:

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}'; \lambda)_{\nu=3/2} &= \left(1 + \frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|}{\lambda} \right) \exp\left(-\frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|}{\lambda} \right), \\ k(\mathbf{x}, \mathbf{x}'; \lambda)_{\nu=5/2} &= \left(1 + \frac{\sqrt{5}\|\mathbf{x} - \mathbf{x}'\|}{\lambda} + \frac{5\|\mathbf{x} - \mathbf{x}'\|^2}{3\lambda^2} \right) \exp\left(-\frac{\sqrt{5}\|\mathbf{x} - \mathbf{x}'\|}{\lambda} \right). \end{aligned} \tag{2.12}$$

As explained in [20], it is also possible to combine or modify existing covariance functions to make new ones. In particular, the sum of two kernels is a kernel as well. This construction can be used, for instance, to add together kernels with different length scales. Moreover, we also have that the product of two kernels is a kernel (for more information about the covariance functions, we refer to [9, 13]).
As reported in [20], all of these families of covariance functions are characterized by a number of free hyperparameters whose values need to be determined, even if it may not be easy to specify all these aspects with confidence in practical applications. Therefore, the choice of a suitable covariance function for a specific application is not simple, since it comprises both the setting of the values of the hyperparameters and the selection of the most appropriate one across different families depending on the process that we aim to predict.

2.3. Predictive distribution

As explained in [20], the choice of the covariance function implies a distribution over functions. This can also be observed in Figure 2.1, where, given the input values X_*, three functions drawn from the Gaussian process prior distribution

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, K(X_*, X_*)) \tag{2.13}$$

are plotted. Notice that in (2.13) the covariance matrix $K(X_*, X_*)$ is defined by using the squared exponential covariance function (2.8) elementwise.

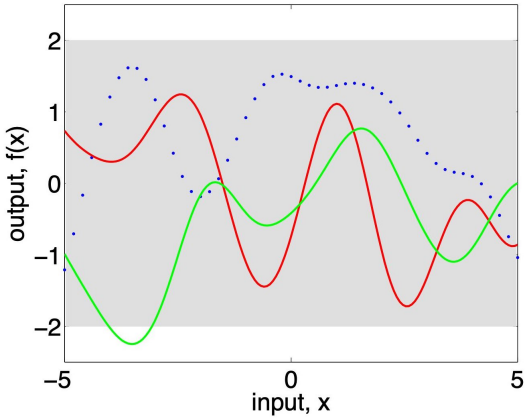


Figure 2.1: Three functions drawn at random from a GP prior, where the dots indicate values of the output actually generated, whereas the two other functions have (less correctly) been drawn as lines by joining a large number of evaluated points (image from [20]).

2.3.1. Prediction with noise free observations

As argued in [20], since we are not interested in drawing functions from the prior, the information given by the training data are exploited. First, the simple case in which the observations are noise free is considered. Given the training inputs and outputs $\{x_i, f_i | i = 1, \ldots, n\}$ and the test inputs and outputs $\{x_{*,j}, f_{*,j} | j = 1, \ldots, n_*\}$, the joint distribution of the training outputs $\mathbf{f}$ and the test outputs $\mathbf{f}_*$ according to the prior is defined as

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \tag{2.14}$$

In order to obtain a suitable posterior distribution, a way to discard those functions whose values do not correspond with the observations is needed. From a probabilistic point of view, this is translated by conditioning the joint Gaussian prior distribution on the training outputs to give

$$\mathbf{f}_* \,|X_*, X, \mathbf{f} \sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)), \tag{2.15}$$

where

$$\bar{\mathbf{f}}_* = \mathbb{E}[\mathbf{f}_* \,|X_*, X, \mathbf{f}] = K(X_*, X)K(X, X)^{-1}\mathbf{f}, \tag{2.16}$$

$$\mathrm{cov}(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*). \tag{2.17}$$

Figure 2.2 shows the results of these computations given five data points marked with a "+" sign, where function values $\mathbf{f}_*$ (corresponding to test inputs X_*) can be sampled by evaluating the mean vector and the covariance matrix from (2.15).

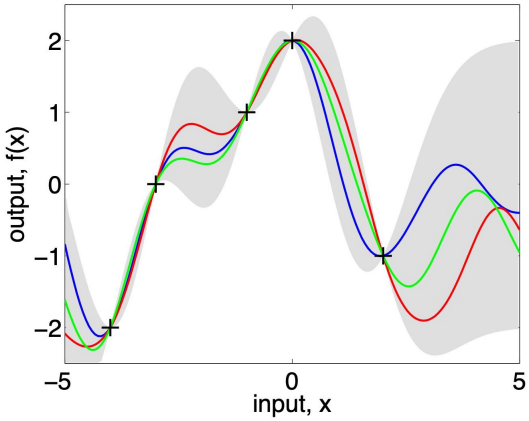


Figure 2.2: Three random functions drawn from the posterior, i.e. the prior conditioned on the five noise free observations indicated. The shaded area corresponds to the 95% confidence region (image from [20]).

2.3.2. Prediction using noisy observations

As explained in [20], in real-life situations we only have access to noisy function values $y = f(\mathbf{x}) + \epsilon$, where ϵ is an independent identically distributed Gaussian noise. As a consequence, the prior on noisy observations becomes

$$\text{cov}(\mathbf{y}) = K(X, X) + \sigma_n^2 I, \tag{2.18}$$

where σ_n^2 represents the noise variance. The main difference with respect to the noise free case is that a diagonal matrix is added because of the effect of the noise. Following the same procedure adopted in (2.14), but accounting for the noise term, the joint distribution of the observed values and the function values at the test points under the prior can be written as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \tag{2.19}$$

The predictive equations for Gaussian process regression in the noisy case are given by

$$\mathbf{f}_* \,|X_*, X, \mathbf{y} \sim \mathcal{N}(\bar{\mathbf{f}}_*, \text{cov}(\mathbf{f}_*)), \tag{2.20}$$

where

$$\bar{\mathbf{f}}_* = \mathbb{E}[\mathbf{f}_* \,|X_*, X, \mathbf{y}] = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}\mathbf{y}, \tag{2.21}$$

$$\text{cov}(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X, X_*). \tag{2.22}$$

For notation purposes, [20] proposes to introduce a compact form of notations $K = K(X, X)$ and $K_* = K(X, X_*)$. Similarly, it is possible to write $\mathbf{k}(\mathbf{x}_*) = \mathbf{k}_*$ to denote the vector containing the covariances between one test point $\mathbf{x}_*$ and the n training points. Therefore, for only one test point (2.20), (2.21) and (2.22) become

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}(\bar{f}_*, \mathbb{V}(f_*)), \tag{2.23}$$

where

$$\bar{f}_* = \mathbf{k}_*^T[K + \sigma_n^2 I]^{-1}\mathbf{y}, \tag{2.24}$$

$$\mathbb{V}(f_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T(K + \sigma_n^2 I)^{-1}\mathbf{k}_*. \tag{2.25}$$

As reported in [20], the mean prediction (2.24) is a linear combination of the observations $\mathbf{y}$. Moreover, it is possible to observe that (2.22), which is defined as the difference between the prior covariance and a positive term representing the information about the function given by the observations, does not depend on the outputs. The posterior covariance with respect to the posterior distribution seen in Figure 2.2, is illustrated in Figure 2.3.

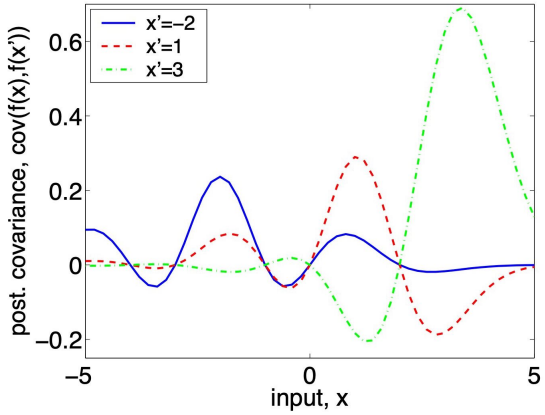


Figure 2.3: Posterior covariance between $f(\mathbf{x})$ and $f(\mathbf{x}')$ for the same data for three different values of $\mathbf{x}'$ (image from [20]).

2.4. Optimization of the hyperparameters

As we anticipated in Section 2.2.2, the choice of a suitable covariance function is strictly related to the setting of the hyperparameters. This problem is treated by the training of the Gaussian process, which consists of the maximization of the log marginal likelihood

with respect to the hyperparameters [25]. In particular, the log marginal likelihood is defined as

$$\log p(\mathbf{y}|X,\boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T K_y^{-1}\mathbf{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi, \tag{2.26}$$

where X represents the training inputs, $\mathbf{y}$ defines the noisy targets, $\boldsymbol{\theta}$ is the vector containing all the hyperparameters and $K_y = K_y(\boldsymbol{\theta})$ is the covariance matrix with respect to the noisy outputs $\mathbf{y}$. As explained in [20], the three terms of the log marginal likelihood have specific roles. In particular, the first term measures the data fit, that is how well the current kernel parametrization explains the dependent variable, while the second one and the last one represent a complexity penalization term and a normalization constant respectively.
In order to set the hyperparameters by maximizing the log marginal likelihood, its partial derivatives with respect to the hyperparameters can be computed:

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\log p(\mathbf{y}|X,\boldsymbol{\theta}) &= \frac{1}{2}\mathbf{y}^T K_y^{-1}\frac{\partial K_y}{\partial\theta_j}K_y^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(K_y^{-1}\frac{\partial K_y}{\partial\theta_j}\right)\\ &= \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - K_y^{-1}\right)\frac{\partial K_y}{\partial\theta_j}\right),\end{aligned} \tag{2.27}$$

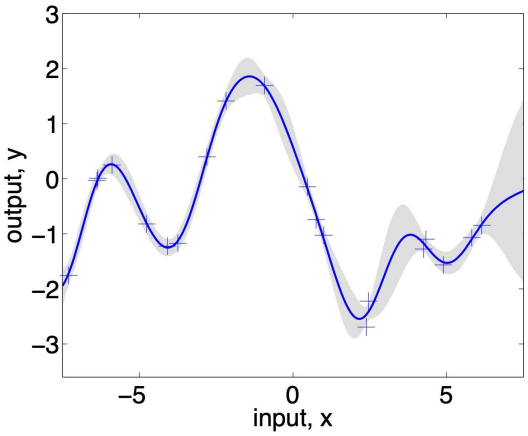
where $\boldsymbol{\alpha} = K_y^{-1}\mathbf{y}$. As noticed in [20], the computational burden required in computing the log marginal likelihood (2.26) is predominantly due to the need to invert the K_y matrix. However, standard methods for inversion of positive definite matrices can be used with a required time $\mathcal{O}(n^3)$ for a $n \times n$ matrix. Once K_y^{-1} is known, only the computation of the derivatives in (2.27) is needed, which requires time $\mathcal{O}(n^2)$ per hyperparameter, since in the first term the matrix-vector multiplications are computed first and in the trace term only the diagonal terms of the product are needed. Therefore, the computational cost of computing derivatives is relatively small and a gradient based optimizer can be used.

2.4.1. Variation of the hyperparameters

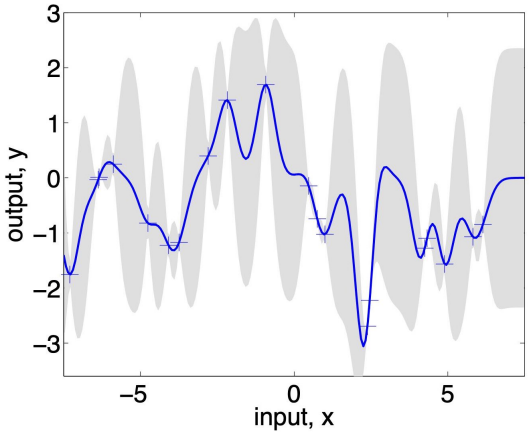
In this section, we report an example in [20] (Figures 2.4–2.5) with the corresponding comments which is useful in order to explain the impact of varying the hyperparameters on Gaussian process predictions and on the log marginal likelihood.
In this example the squared exponential covariance function in one dimension is considered:

$$k_y(x_i,x_j;\boldsymbol{\theta}) = \sigma^2\exp\left(-\frac{(x_i-x_j)^2}{2\lambda^2}\right) + \sigma_n^2\delta_{ij}, \tag{2.28}$$

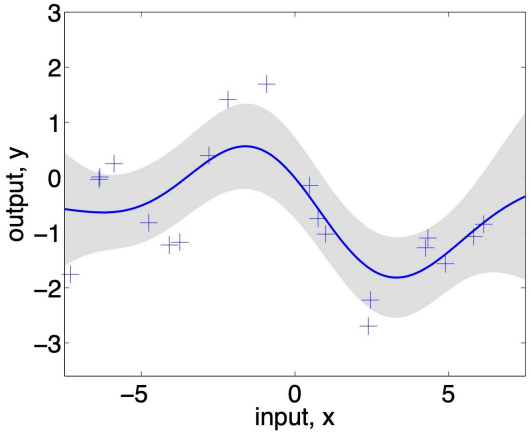
where $\boldsymbol{\theta} = (\lambda,\sigma^2,\sigma_n^2)^T$ is the vector containing all the hyperparameters.



(a) $\lambda = 1$



(b) $\lambda = 0.3$

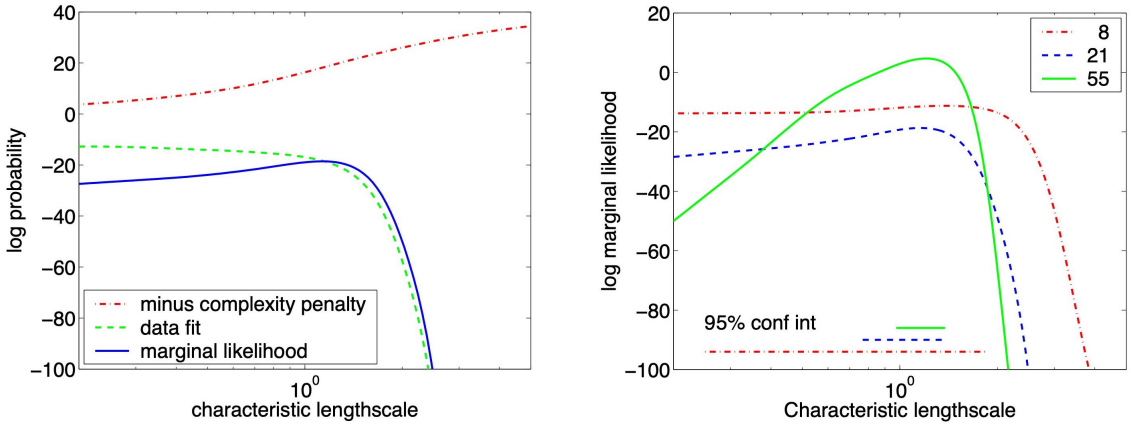


(c) $\lambda = 3$

Figure 2.4: Panel (a) shows data generated from a GP with hyperparameters $(\lambda, \sigma^2, \sigma_n^2) = (1, 1, 0.1)$. Panels (b) and (c) show the predictions with hyperparameters $(\lambda, \sigma^2, \sigma_n^2) = (0.3, 1.08, 0.00005)$ and $(\lambda, \sigma^2, \sigma_n^2) = (3, 1.16, 0.89)$ respectively. In all the plots, the shaded area corresponds to the 95% confidence region (image from [20]).

Results for three different values of the length scale are reported in order to understand the concepts of over-fitting and under-fitting. First, it is possible to observe in Figure 2.4a the data generated from a Gaussian process with hyperparameters $(\lambda, \sigma^2, \sigma_n^2) = (1, 1, 0.1)$, where the shaded area represents the 95% confidence region and gets larger for test inputs which are far from the training values.
Second, Figure 2.4b shows that over-fitting occurs if predictions with a shorter length scale (e.g., $\lambda = 0.3$) are made on the data generated by the process with $\lambda = 1$. Concerning the other two parameters, they are set in such a way that the marginal likelihood is optimized. For this reason, the noise parameter becomes $\sigma_n^2 = 0.00005$. For instance, if the two data points near $x = 2.5$ in the plots are considered, it is possible to observe that they provide

completely different results depending on the value of the length scale. Indeed, for $\lambda = 1$ a similar function value with different noise is obtained, whereas for $\lambda = 0.3$ the noise level is very low and a significant variation in the value of the function is detectable.
In conclusion, Figure 2.4c shows the occurrence of under-fitting for a greater value of the length scale (e.g., $\lambda = 3$); indeed, the function is not able to take over the variations under these assumptions and the result is a slowly varying behaviour affected by a lot of noise.
For what concerns the log marginal likelihood, Figure 2.5a illustrates how the data fit decreases as the value of the length scale increases, since this results in a model which presents difficulties in detecting the variations of the function. On the other hand, the negative complexity penalty increases because the model gets less complex as the length scale grows.
Figure 2.5b, instead, shows the dependence of the log marginal likelihood on the length scale for different numbers of training values. In general, the log marginal likelihood tends to be more peaked as the number of training points increases, since this implies a more severe complexity term.



(a) (b)

Figure 2.5: Panel (a) shows a decomposition of the log marginal likelihood into its constituents (data fit and complexity penalty) as a function of the length scale. Panel (b) shows the log marginal likelihood as a function of the length scale for different sizes of the training sets (image from [20]).

2.5. Diagnostics for Gaussian processes

Emulators (like Gaussian processes) are usually used as stochastic approximations of computationally expensive simulators. However, their adoption requires a robust validation. In particular, we consider throughout this work diagnostic methods, which represent a

set of techniques to prove the quality of the statistical model, as a validation process. For instance, they can be useful tools (with graphical or quantitative results) for assessing the assumptions of the underlying model and can provide suggestions to improve its validity. Therefore, by using diagnostic methods, we aim to investigate the best probability model for describing the simulator [1].
As reported in [2], there are several metrics that can be adopted to test Gaussian process emulators. One of the most commonly used is the root mean squared error (RMSE):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \mu_i)^2}{n}}, \tag{2.29}$$

where μ_i represents the prediction of the Gaussian process and y_i stands for the observed output. Since it evaluates the mean squared difference between the emulator predictions and the simulator outputs, which can be roughly seen as a measure of the "distance" between observed outputs and predictions, then it is better to obtain a low value from this method.
A different widely used diagnostic method which takes into account the uncertainty is based on the individual prediction errors. The idea is always to somehow analyze the differences between the observed simulator outputs and the Gaussian process predictions at the same inputs. In particular, each standardized prediction error is considered as a diagnostics:

$$D_i = \frac{y_i - \mu_i}{\sigma_i}, \tag{2.30}$$

where σ_i represents the predictive standard deviation of the Gaussian process. As explained in [2], in general, the individual standardized errors will have a standard Student-t distribution if the uncertainty about the simulator is properly represented by the Gaussian process. However, it tends to a standard normal distribution as the number of degrees of freedom increases.
Some problems my arise if large individual standardized errors are found with an absolute value greater than 3. Isolated extreme individual standardized errors may suggest a local problem only for those validation points. If large individual standardized errors tend to correspond to validation points that are close to training points, then there are some issues of overestimation of the length parameters of the Gaussian process. This means that the closeness to the training data points may influence the predictions too strongly. On the other hand, the underestimation of the length parameters may be suggested if there are small individual standardized errors.

2.6. Numerical experiments

In this section, we test the efficiency of Gaussian processes in making predictions. We start by analyzing their accuracy for a benchmark problem with a varying number of input parameters. Once this efficiency is verified, we can proceed with the analysis of Gaussian process capabilities in predicting the behaviour of the lumped parameter closed-loop model for the circulation system.

All the numerical simulations considered in this work are performed in Python. In particular, Gaussian process is implemented by means of the open source Python library `TensorFlow` with a squared exponential covariance function (for more information about the GP implementation, see Algorithm 2.1). Once the kernel function is given, a preliminary crucial step in Gaussian process regression is the choice of the initial values for the hyperparameters. We chose the values proposed in `TensorFlow` documentation for the benchmark problem, whereas the initial values of the hyperparameters in [28] are used for the circulation system (see Tables 2.1–2.2 respectively).

For the high-fidelity circulation model we exploit the Python class `circulation_closed_loop`, which is available in the GitHub repository[1]. In Tables 2.3–2.4 we report parameters and outputs of the lumped parameter closed-loop model for the circulation system.

Hyperparameter	Initial value	Description
σ^2	1.0	Amplitude
λ	1.0	Length scale
σ_n^2	1.0	Noise variance

Table 2.1: Initial values of the squared exponential covariance function hyperparameters for the benchmark problem.

Hyperparameter	Initial value	Description
σ^2	0.1209	Amplitude
λ	0.0816	Length scale
σ_n^2	0.0146	Noise variance

Table 2.2: Initial values of the squared exponential covariance function hyperparameters for the circulation system.

[1] https://github.com/AlessandroPirozzi/Cardio_Circulation_UQ

Algorithm 2.1 Gaussian process

1: **Input**: training inputs X, training outputs $\mathbf{y}$ and test inputs X_*
2: **Output**: hyperparameters $\boldsymbol{\theta} = (\lambda, \sigma^2, \sigma_n^2)^T$, Gaussian process mean function $\boldsymbol{\mu}$ and predictive standard deviations $\boldsymbol{\sigma}$
3: Initialize the hyperparameters $\boldsymbol{\theta}$
4: **for** $0 \leq i \leq N$ **do**
5: Compute the loss function (2.26)
6: Update the hyperparameters $\boldsymbol{\theta}$ using the Adam algorithm for first order gradient based optimization with respect to the gradients in (2.27)
7: **end for**
8: Define the squared exponential covariance function with respect to the optimized hyperparameters $\hat{\boldsymbol{\theta}}$
9: Compute the mean function $\boldsymbol{\mu}$ and the corresponding predictive standard deviations $\boldsymbol{\sigma}$ by using (2.21)–(2.22)

2.6.1. One-dimensional benchmark problem

We start from the analysis of Gaussian process prediction performances in a one-dimensional benchmark problem. In particular, we test Gaussian processes efficiency in different situations; in this way, we aim to highlight some important characteristics, such as the accuracy improvement in predictions as the number of training values increases and the effects of the noise on the observations. For this reason, we will consider 25, 50, 100 training points and 10^{-2}, 10^{-3}, 10^{-4} for the observations noise variance, whereas the number of test values is always equal to 10000. The model output is given by the function

$$f(x) = \sin(3\pi x), \tag{2.31}$$

where x represents the input parameter.
In Figures 2.6–2.8 the dots represent the training values, while the solid blue line and the red lines are the real function and the posterior predictive distribution obtained from the Gaussian process regression respectively.

Parameter	Baseline	Unit	Description
$E_{\mathrm{LA}}^{\mathrm{act}}$, $E_{\mathrm{LV}}^{\mathrm{act}}$, $E_{\mathrm{RA}}^{\mathrm{act}}$, $E_{\mathrm{RV}}^{\mathrm{act}}$	0.07, 3.35, 0.06, 0.55	$\mathrm{mmHg\,mL^{-1}}$	LA/LV/RA/RV active elastance
$E_{\mathrm{LA}}^{\mathrm{pass}}$, $E_{\mathrm{LV}}^{\mathrm{pass}}$, $E_{\mathrm{RA}}^{\mathrm{pass}}$, $E_{\mathrm{RV}}^{\mathrm{pass}}$	0.18, 0.2, 0.07, 0.05	$\mathrm{mmHg\,mL^{-1}}$	LA/LV/RA/RV passive elastance
$T_{\mathrm{LA}}^{\mathrm{contr}}$, $T_{\mathrm{LV}}^{\mathrm{contr}}$, $T_{\mathrm{RA}}^{\mathrm{contr}}$, $T_{\mathrm{RV}}^{\mathrm{contr}}$	0.17, 0.25, 0.17, 0.25	s	LA/LV/RA/RV contraction time
$T_{\mathrm{LA}}^{\mathrm{rel}}$, $T_{\mathrm{LV}}^{\mathrm{rel}}$, $T_{\mathrm{RA}}^{\mathrm{rel}}$, $T_{\mathrm{RV}}^{\mathrm{rel}}$	0.17, 0.4, 0.17, 0.4	s	LA/LV/RA/RV relaxation time
$V_{0,\mathrm{LA}}$, $V_{0,\mathrm{LV}}$, $V_{0,\mathrm{RA}}$, $V_{0,\mathrm{RV}}$	4.0, 42.0, 4.0, 16.0	mL	LA/LV/RA/RV reference volume
$t_{\mathrm{LA}}^{\mathrm{del}}$, $t_{\mathrm{LV}}^{\mathrm{del}}$, $t_{\mathrm{RA}}^{\mathrm{del}}$, $t_{\mathrm{RV}}^{\mathrm{del}}$	0.9, 0.1, 0.9, 0.1	s	LA/LV/RA/RV time delay
R_{min}, R_{max}	0.0075, 75006.2	$\mathrm{mmHg\,s\,mL^{-1}}$	Valve minimum/maximum resistance
$R_{\mathrm{AR}}^{\mathrm{SYS}}$, $R_{\mathrm{VEN}}^{\mathrm{SYS}}$	0.64, 0.32	$\mathrm{mmHg\,s\,mL^{-1}}$	Systemic arterial/venous resistance
$R_{\mathrm{AR}}^{\mathrm{PUL}}$, $R_{\mathrm{VEN}}^{\mathrm{PUL}}$	0.032116, 0.035684	$\mathrm{mmHg\,s\,mL^{-1}}$	Pulmonary arterial/venous resistance
$C_{\mathrm{AR}}^{\mathrm{SYS}}$, $C_{\mathrm{VEN}}^{\mathrm{SYS}}$	1.2, 60.0	$\mathrm{mL\,mmHg^{-1}}$	Systemic arterial/venous capacitance
$C_{\mathrm{AR}}^{\mathrm{PUL}}$, $C_{\mathrm{VEN}}^{\mathrm{PUL}}$	10.0, 16.0	$\mathrm{mL\,mmHg^{-1}}$	Pulmonary arterial/venous capacitance
$L_{\mathrm{AR}}^{\mathrm{SYS}}$, $L_{\mathrm{VEN}}^{\mathrm{SYS}}$	0.005, 0.0005	$\mathrm{mmHg\,s^2\,mL^{-1}}$	Systemic arterial/venous inductance
$L_{\mathrm{AR}}^{\mathrm{PUL}}$, $L_{\mathrm{VEN}}^{\mathrm{PUL}}$	0.0005, 0.0005	$\mathrm{mmHg\,s^2\,mL^{-1}}$	Pulmonary arterial/venous inductance

Table 2.3: Parameters of the circulation model and associated baseline values.

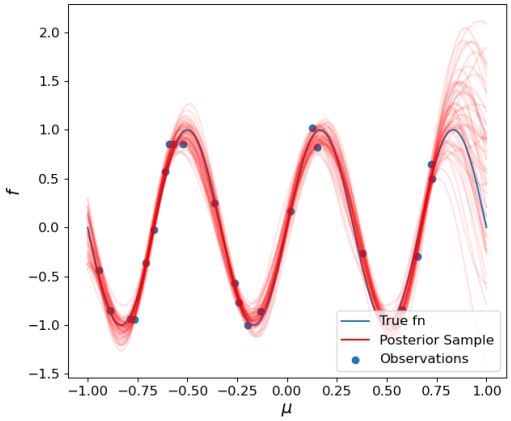
Parameter	Unit	Description
$V_{\mathrm{LA}}^{\mathrm{min}}$, $V_{\mathrm{LA}}^{\mathrm{max}}$	mL	End systolic/end diastolic volume of left atrium
$p_{\mathrm{LA}}^{\mathrm{min}}$, $p_{\mathrm{LA}}^{\mathrm{max}}$	mmHg	Minimum/maximum pressure in left atrium
$V_{\mathrm{LV}}^{\mathrm{min}}$, $V_{\mathrm{LV}}^{\mathrm{max}}$	mL	End systolic/end diastolic volume of left ventricle
$p_{\mathrm{LV}}^{\mathrm{min}}$, $p_{\mathrm{LV}}^{\mathrm{max}}$	mmHg	Minimum/maximum pressure in left ventricle
$V_{\mathrm{RA}}^{\mathrm{min}}$, $V_{\mathrm{RA}}^{\mathrm{max}}$	mL	End systolic/end diastolic volume of right atrium
$p_{\mathrm{RA}}^{\mathrm{min}}$, $p_{\mathrm{RA}}^{\mathrm{max}}$	mmHg	Minimum/maximum pressure in right atrium
$V_{\mathrm{RV}}^{\mathrm{min}}$, $V_{\mathrm{RV}}^{\mathrm{max}}$	mL	End systolic/end diastolic volume of right ventricle
$p_{\mathrm{RV}}^{\mathrm{min}}$, $p_{\mathrm{RV}}^{\mathrm{max}}$	mmHg	Minimum/maximum pressure in right ventricle
$p_{\mathrm{AR,min}}^{\mathrm{SYS}}$, $p_{\mathrm{AR,max}}^{\mathrm{SYS}}$	mmHg	Minimum/maximum pressure in systemic arterial circulation

Table 2.4: List of outputs of the circulation model.

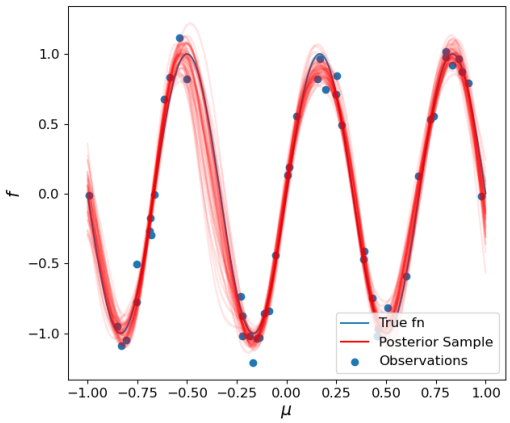
We can observe how increasing the number of training values or reducing the noise of the observations results in improved predictions. In particular, this improvement is significantly relevant when the number of training values varies, whereas, since the value 10^{-2} for the observations noise variance is already low enough, no sharp improvements are detected when reducing it. In any case, we can conclude that Gaussian processes are characterized by a significant prediction ability, especially if combined with its irrelevant required computational time, as shown in Table 2.5.

Training set	Hyperparameters	Predictions
$\sim$ 0.01 s	$\sim$ 9–14 s	$\sim$ 67–105 s

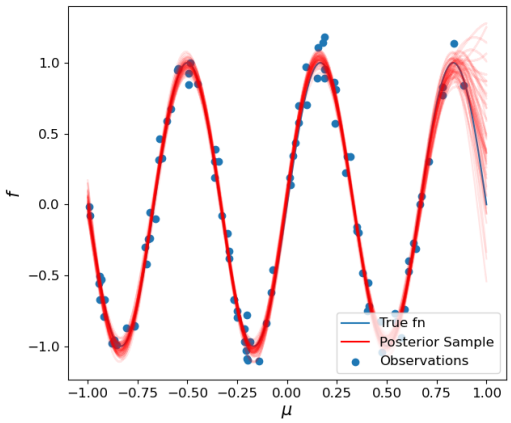
Table 2.5: Computational time for each section of the process: generation of the training set, optimization of the hyperparameters and predictions.



(a) 25 training values

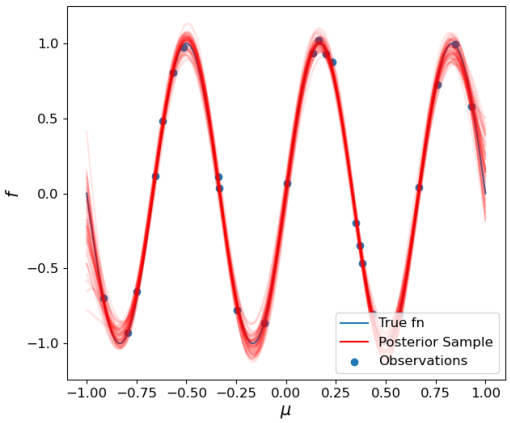


(b) 50 training values

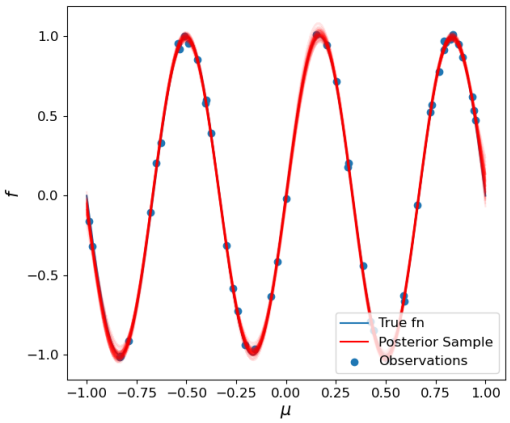


(c) 100 training values

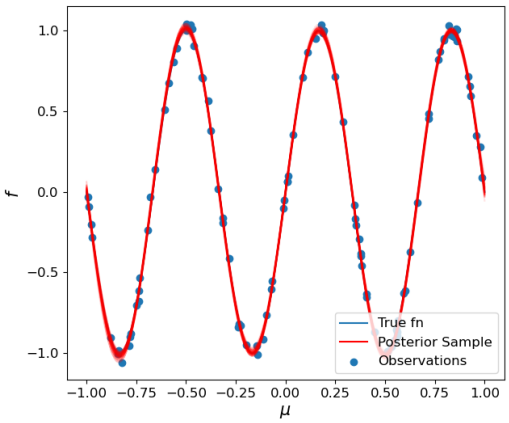
Figure 2.6: Real function (2.31) and its posterior predictive distribution given by the Gaussian process with 25, 50, 100 training values and 10^{-2} observations noise variance.



(a) 25 training values

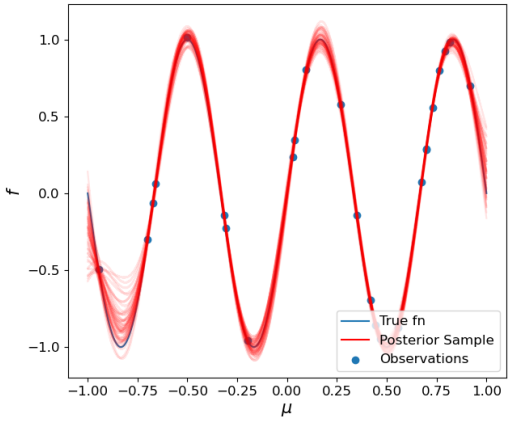


(b) 50 training values

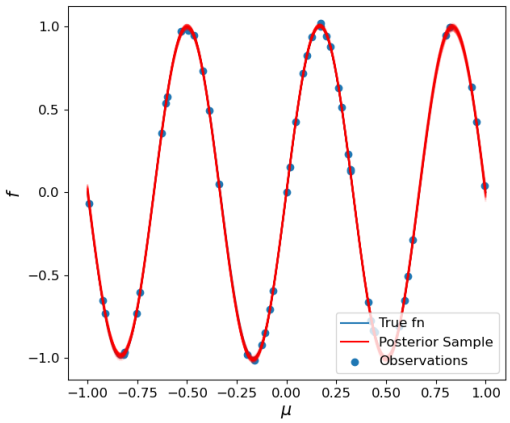


(c) 100 training values

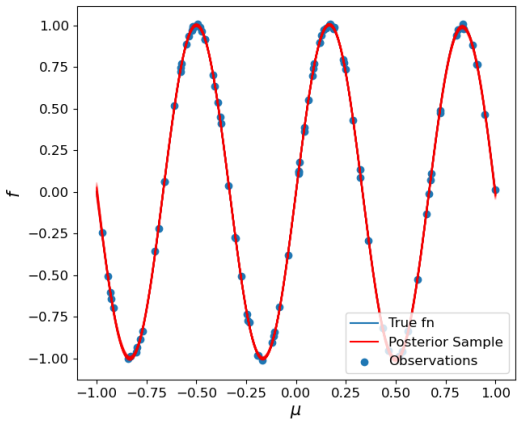
Figure 2.7: Real function (2.31) and its posterior predictive distribution given by the Gaussian process with 25, 50, 100 training values and 10^{-3} observations noise variance.



(a) 25 training values



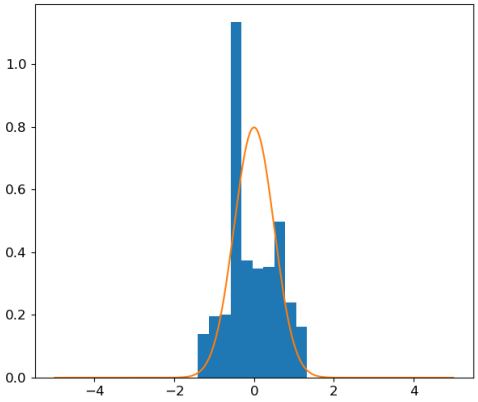
(b) 50 training values



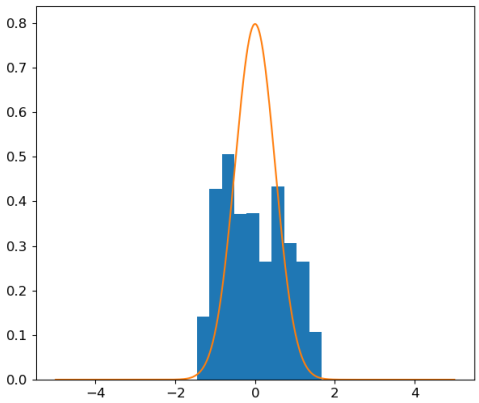
(c) 100 training values

Figure 2.8: Real function (2.31) and its posterior predictive distribution given by the Gaussian process with 25, 50, 100 training values and 10^{-4} observations noise variance.

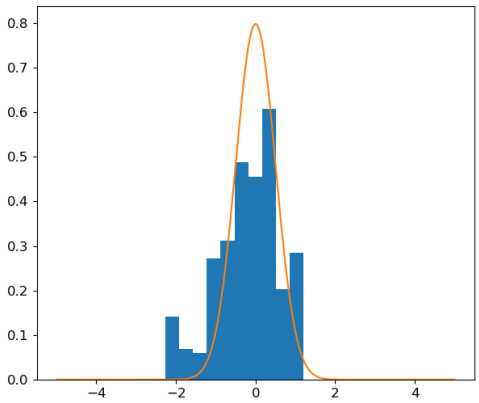
For what concerns the diagnostics, we can refer to the individual standardized errors. Indeed, Figure 2.9 shows that the distribution of the individual standardized errors tends to be a standard normal distribution as the number of training values increases. This is coherent with what is explained in Section 2.5, namely that the uncertainty about the simulator is properly represented by the Gaussian process.



(a) 25 training values



(b) 50 training values



(c) 100 training values

Figure 2.9: Distribution of the individual standardized errors with 25, 50, 100 training values and 10^{-2} observations noise variance.

2.6.2. Anisotropic covariance functions

In this section, we motivate the need of considering an anisotropic nature for covariance functions. Indeed, in all the numerical experiments we used the squared exponential covariance function, which is by definition an isotropic covariance function (see Section 2.2.1). This is a suitable choice if we have to deal with the variation of only one parameter, as in Section 2.6.1, or if the output varies uniformly in all the directions. However, if we consider more than one varying input parameter, then an anisotropic covariance function is required for better results (for more information, we refer to [29]).
In order to obtain an anisotropic version of the squared exponential covariance function,

[20] proposes to rewrite (2.28) as

$$k_y(x_i, x_j) = \sigma^2 \exp\Big(-\frac{1}{2}(x_i - x_j)^T M (x_i - x_j)\Big) + \sigma_n^2 \delta_{ij}, \tag{2.32}$$

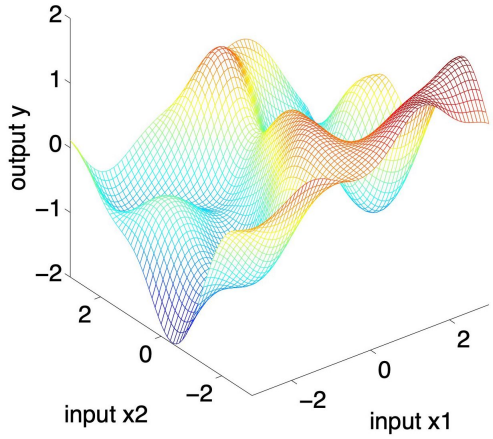
where $\boldsymbol{\theta} = (\{M\}, \sigma^2, \sigma_n^2)^T$ is a vector containing all the hyperparameters and $\{M\}$ denotes the parameters in the symmetric matrix M. Possible choices for the matrix M include

$$M_1 = \lambda^{-2} I, \quad M_2 = \text{diag}(\boldsymbol{\lambda})^{-2}, \quad M_3 = \Lambda\Lambda^T + \text{diag}(\boldsymbol{\lambda})^{-2}, \tag{2.33}$$

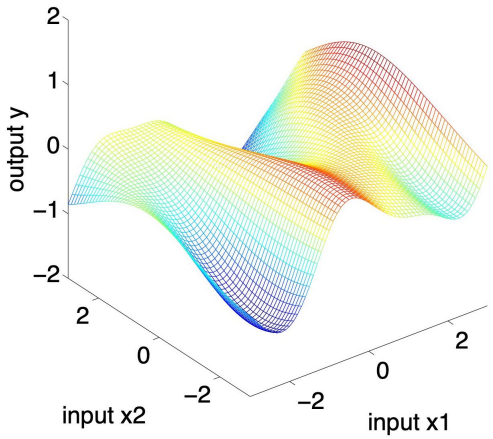
where $\boldsymbol{\lambda}$ is a vector of positive values and Λ is a rectangular matrix $D \times k$, $k < D$.
As we widely said in Section 2.2, the properties of the function that we want to model inferred from the covariance function are strictly related to the values of the hyperparameters. The interpretation of these hyperparameters is of great importance to understand the obtained results. As reported in [20], the possible choices of M in (2.33) denote the length scale for the squared exponential covariance function (2.32). Depending on the used matrix, one can obtain different behaviours of the posterior predictive distribution. In particular, M_1 is the one used for the numerical experiments in the previous section, whereas M_2 allows us to consider an anisotropic nature of covariance functions, since $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_D)^T$ represents the vector containing the length scales with respect to each parameter. In conclusion, matrix M_3 defines the factor analysis distance which, for high dimensional data sets, allows to identify the k columns of the Λ matrix that represent the directions in the input space with a special high relevance.
In Figure 2.10 functions drawn at random from Gaussian processes with a squared exponential covariance function for different choices of M are shown. In particular, Figure 2.10a shows the isotropic behaviour, whereas in Figure 2.10b the length scale is different along the two input axes. Indeed, the function varies more rapidly with respect to x_1 than the other input parameter x_2. In conclusion, Figure 2.10c shows that the direction of most rapid variation is perpendicular to the direction $(1, 1)$.
Since the choice of an anisotropic covariance function aims to obtain better results, we want to prove that this actually happens for a benchmark problem. We consider the two-dimensional case with the model output which is given by the function

$$\begin{aligned} f(\mathbf{x}) = &\sin(6\mu_2) + 0.1\cos(\mu_1) + \sin(\mu_1)\cos(3\mu_2) + \\ &0.25\sin(\mu_3) + 0.6\cos(2\mu_4) + 0.5\cos(\mu_3)\sin(\mu_4) + \\ &\sin(\mu_4)\cos(\mu_8) + 0.3\cos(\mu_{10}) + 0.7\sin(\mu_{11}) + \\ &0.1\cos(\mu_5) + \sin(\mu_5)\cos(\mu_{12}) + 0.25\sin(\mu_9) + \\ &0.1\sin(\mu_6)\cos(\mu_7), \end{aligned} \tag{2.34}$$

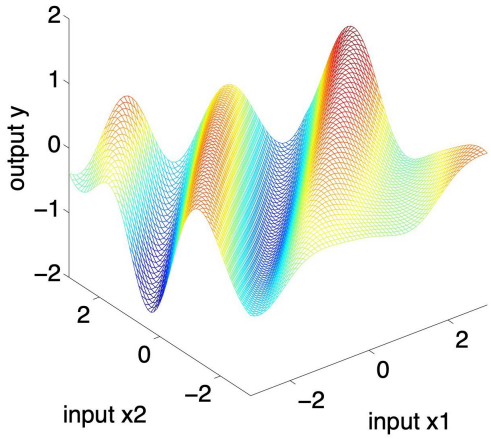
where $\mathbf{x} = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7, \mu_8, \mu_9, \mu_{10}, \mu_{11}, \mu_{12})^T$ is the vector containing all the input parameters, but only μ_1 and μ_2 vary (the others are fixed at zero).



(a) $\lambda = 1$



(b) $\boldsymbol{\lambda} = (1, 3)^T$

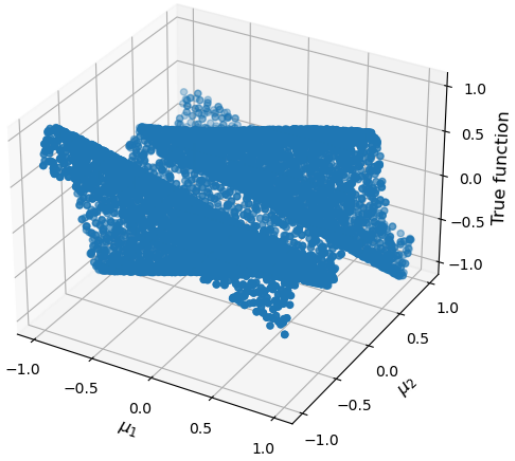


(c) $\boldsymbol{\lambda} = (6, 6)^T$, $\Lambda = (1, -1)^T$

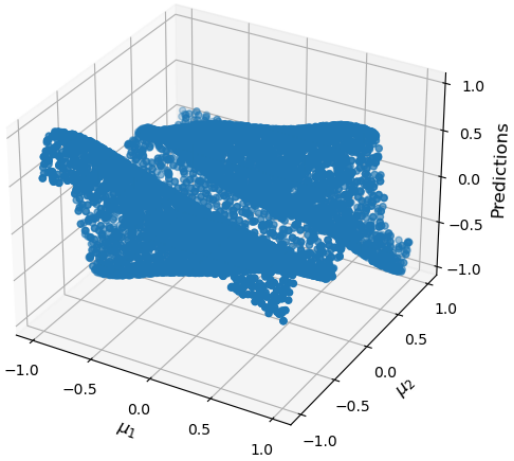
Figure 2.10: Functions with two dimensional input drawn at random from noise free squared exponential covariance function Gaussian processes, corresponding to the three different distance measures in (2.33) respectively. In panel (a) the two inputs are equally important, while in panel (b) the function varies less rapidly with respect to the second input x_2. In panel (c) the Λ column gives the direction of most rapid variation (image from [20]).

We adopt the root mean squared error and the L^∞ error as diagnostics in order to show the improvements given by the anisotropic approach. Figure 2.11 shows how the posterior predictive distribution changes as we modify the approach. Moreover, Table 2.6 explicitly shows from a quantitative point of view that a significant decrease of the errors is obtained passing from an isotropic covariance function, where the errors are extremely larger, to

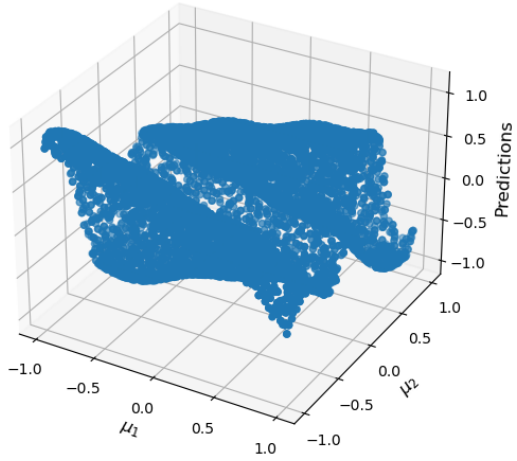
an anisotropic one, where acceptable values of the errors are registered.



(a) True function



(b) Anisotropic



(c) Isotropic

Figure 2.11: Comparison between the true function (2.34) in panel (a) and its posterior predictive distribution with 100 training values and 9000 test values by using the anisotropic squared covariance function in panel (b) and the isotropic squared covariance function in panel (c).

Error	**Isotropic**	**Anisotropic**
RMSE	0.1077	0.0671
L^∞ error	0.4339	0.2220

Table 2.6: Root mean squared error and L^∞ error in the isotropic and anisotropic cases.

2.6.3. Multi-dimensional benchmark problem

Once we proved the efficacy of the anisotropic covariance function, we decide to use the squared exponential covariance function with M_2 from (2.33) in the multi-dimensional problem, since we want the length scales to be different for all the parameters.
The model output is given by the function

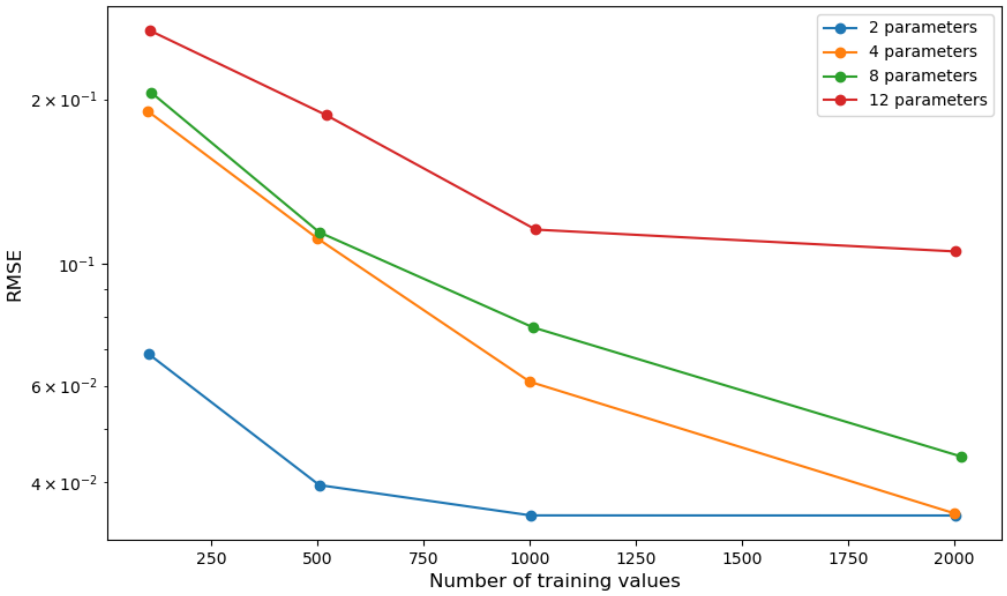
$$\begin{aligned} f(\mathbf{x}) = &\sin(3\mu_2) + 0.1\cos(\mu_1) + \sin(\mu_1)\cos(\mu_2) + \\ &0.25\sin(\mu_3) + 0.6\cos(2\mu_4) + 0.5\cos(\mu_3)\sin(\mu_4) + \\ &\sin(\mu_4)\cos(\mu_8) + 0.3\cos(\mu_{10}) + 0.7\sin(\mu_{11}) + \\ &0.1\cos(\mu_5) + \sin(\mu_5)\cos(\mu_{12}) + 0.25\sin(\mu_9) + \\ &0.1\sin(\mu_6)\cos(\mu_7), \end{aligned} \tag{2.35}$$

where $\mathbf{x} = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_7, \mu_8, \mu_9, \mu_{10}, \mu_{11}, \mu_{12})^T$ is the vector containing all the input parameters. Notice that, for simplicity, the output is generated by a superposition of oscillating functions, so that all the input parameters have baseline value equal to zero and variate in $[-1, 1]$.
We test the prediction efficiency of Gaussian process regression in four different cases with 9000 test values:

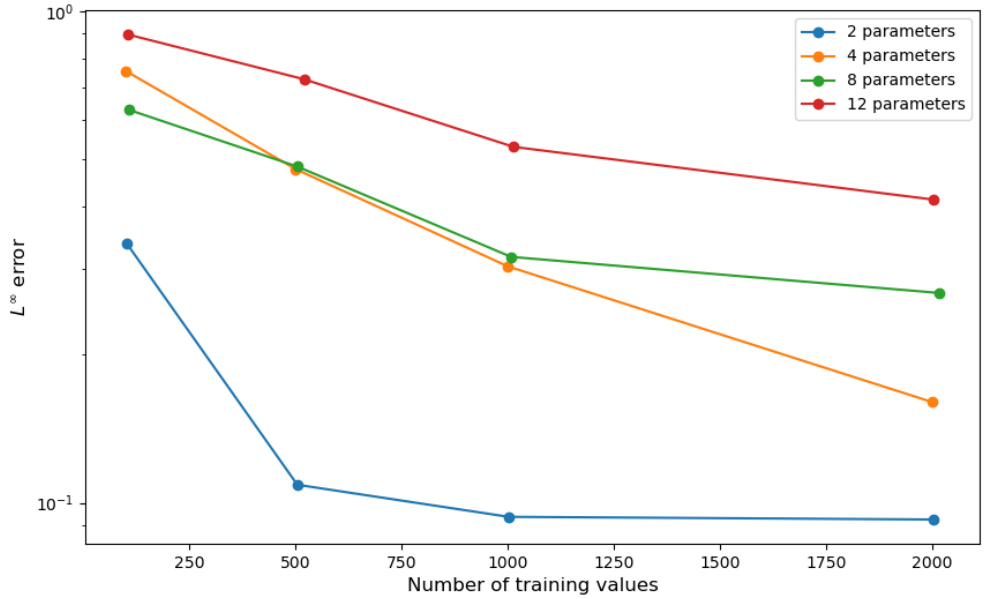
- Case 1: 100, 500, 1000, 2000 training values with 2 varying parameters;
- Case 2: 100, 500, 1000, 2000 training values with 4 varying parameters;
- Case 3: 100, 500, 1000, 2000 training values with 8 varying parameters;
- Case 4: 100, 500, 1000, 2000 training values with 12 varying parameters.

For each numerical experiment we fix the non-varying input parameters at their baseline value and test the reliability of the Gaussian process posterior predictive distribution by means of the diagnostic methods based on the computation of the root mean squared error and the L^∞ error.
We can observe both from a graphical and quantitative point of view the errors behaviour in Figure 2.12 and in Table 2.7 respectively. As we expected, the errors increase as the number of parameters gets larger (e.g., the RMSE increases from 0.0685 with 100 training values and 2 parameters to 0.2672 with 100 training values and 12 parameters), since more varying input parameters implies a more complex behaviour to be predicted. On the other hand, we can observe that errors tend to decrease when the number of training values increases (e.g., the L^∞ error decreases from 0.7559 with 100 training values and 4 parameters to 0.1604 with 2000 training values and 4 parameters). However, this forces

us to deal with an increase of the required computational time, which nevertheless is not too penalizing (see Table 2.8).



(a) RMSE



(b) L^∞ error

Figure 2.12: Behaviour of the root mean squared error and L^∞ error for the benchmark problem as the size of the training sample varies.

		N° of parameters			
N° of training values	**Error**	**2**	**4**	**8**	**12**
100	RMSE	0.0685	0.1902	0.2058	0.2672
	L^∞ error	0.3370	0.7559	0.6299	0.8973
500	RMSE	0.0395	0.1114	0.1144	0.1872
	L^∞ error	0.1089	0.4775	0.4832	0.7273
1000	RMSE	0.0348	0.0610	0.0767	0.1157
	L^∞ error	0.0937	0.3028	0.3165	0.5298
2000	RMSE	0.0348	0.0351	0.0446	0.1055
	L^∞ error	0.0925	0.1604	0.2673	0.4141

Table 2.7: Root mean squared errors and L^∞ errors for the benchmark problem as the size of the training sample varies.

Training set	**Hyperparameters**	**Predictions**
$\sim$ 0.03–0.09 s	$\sim$ 13–7835 s	$\sim$ 21–58 s

Table 2.8: Computational time for each section of the process: generation of the training set, optimization of the hyperparameters and predictions.

We can conclude that a similar trend is recognizable for all the cases, but the higher the variability of the output, the more demanding the training of the Gaussian process.

2.6.4. Multi-dimensional circulation model

We finally perform the same analysis presented in the previous section on the high-fidelity circulation model (1.3)–(1.4). We expect a general trend of the results similar to that of the benchmark problem.
Also in this case we test the prediction efficiency of Gaussian process regression in four different cases (see Table 2.9) with 9000 test values, whereas the considered output is in Table 2.10.

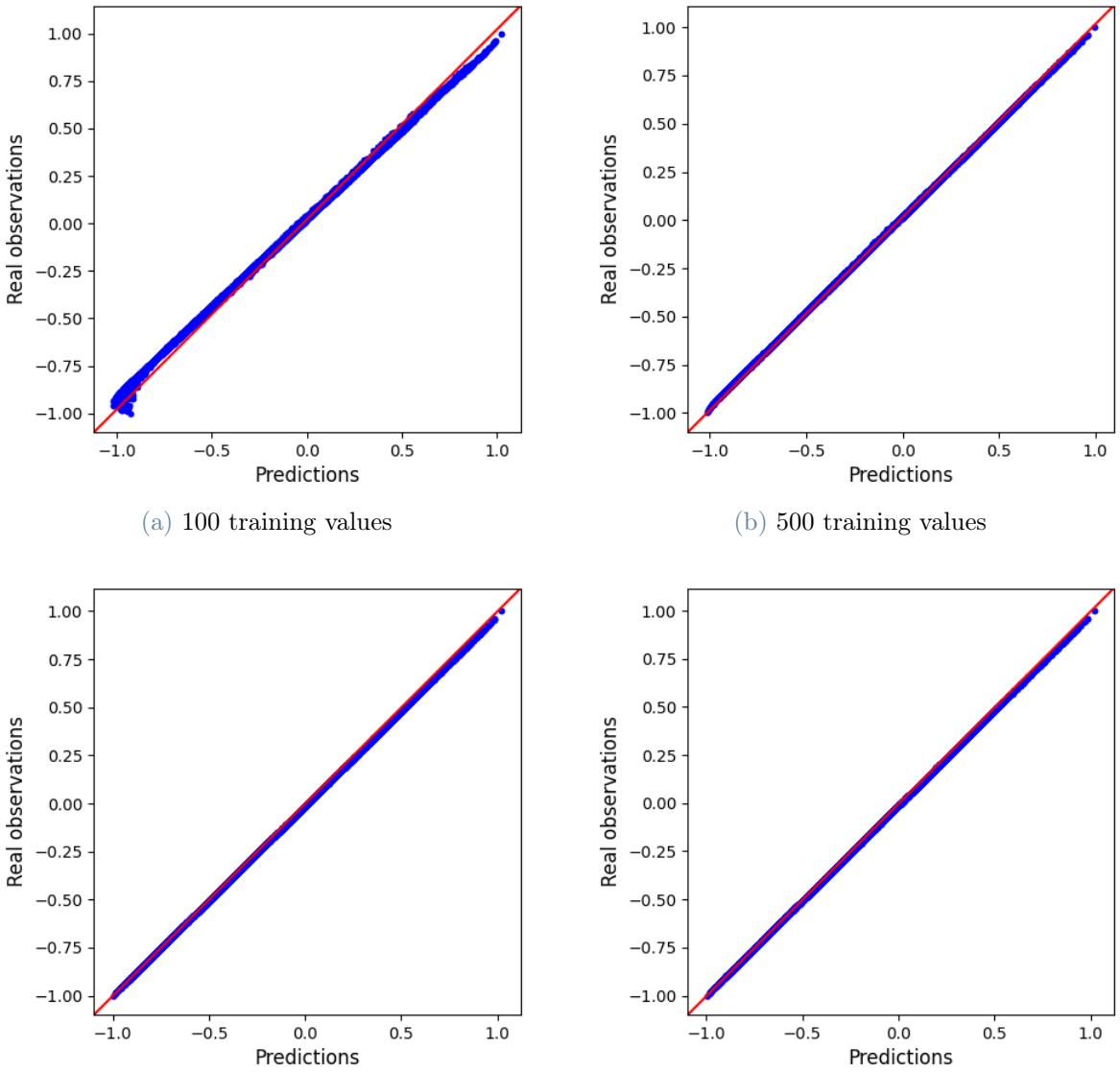
Case	Parameters
Case 1	$R_{\text{AR}}^{\text{SYS}}$, $C_{\text{AR}}^{\text{SYS}}$
Case 2	$R_{\text{AR}}^{\text{SYS}}$, $C_{\text{AR}}^{\text{SYS}}$, $E_{\text{LV}}^{\text{act}}$, $E_{\text{LV}}^{\text{pass}}$
Case 3	$R_{\text{AR}}^{\text{SYS}}$, $C_{\text{AR}}^{\text{SYS}}$, $E_{\text{LV}}^{\text{act}}$, $E_{\text{LV}}^{\text{pass}}$, $T_{\text{LV}}^{\text{contr}}$, $T_{\text{LV}}^{\text{rel}}$, $t_{\text{LA}}^{\text{del}}$, $t_{\text{RA}}^{\text{del}}$
Case 4	$R_{\text{AR}}^{\text{SYS}}$, $C_{\text{AR}}^{\text{SYS}}$, $E_{\text{LV}}^{\text{act}}$, $E_{\text{LV}}^{\text{pass}}$, $T_{\text{LV}}^{\text{contr}}$, $T_{\text{LV}}^{\text{rel}}$, $t_{\text{LA}}^{\text{del}}$, $t_{\text{RA}}^{\text{del}}$, $R_{\text{VEN}}^{\text{SYS}}$, $R_{\text{VEN}}^{\text{PUL}}$, $C_{\text{VEN}}^{\text{SYS}}$, $C_{\text{VEN}}^{\text{PUL}}$

Table 2.9: Varying input parameters of the circulation model considered in the four cases.

Parameter	Unit	Description
$p_{\text{LV}}^{\max}$	mmHg	Maximum pressure in left ventricle

Table 2.10: Output of the circulation model used for diagnostics of the Gaussian process.

Figures 2.13–2.16 show that the prediction efficiency of the Gaussian process posterior predictive distribution improves as the number of training values increases (e.g., the RMSE decreases from 0.2285 with 100 training values and 8 parameters to 0.0698 with 2000 training values and 8 parameters). On the other hand, it is possible to detect a worsening of the predictions as the number of varying parameters gets larger (e.g., the L^∞ error increases from 0.0946 with 100 training values and 2 parameters to 0.8019 with 100 training values and 12 parameters). However, even if a more complicated model implies a worse prediction ability of the Gaussian process, an improvement is always noticeable increasing the number of training values. These results are coherent with those of the benchmark problem and can also be observed in Figure 2.17 and Table 2.11, which report both the graphical and quantitative trend of the errors respectively. The only difference concerns the computational time which significantly increases, above all for the generation of the training set and the optimization of the hyperparameters (see Table 2.12).



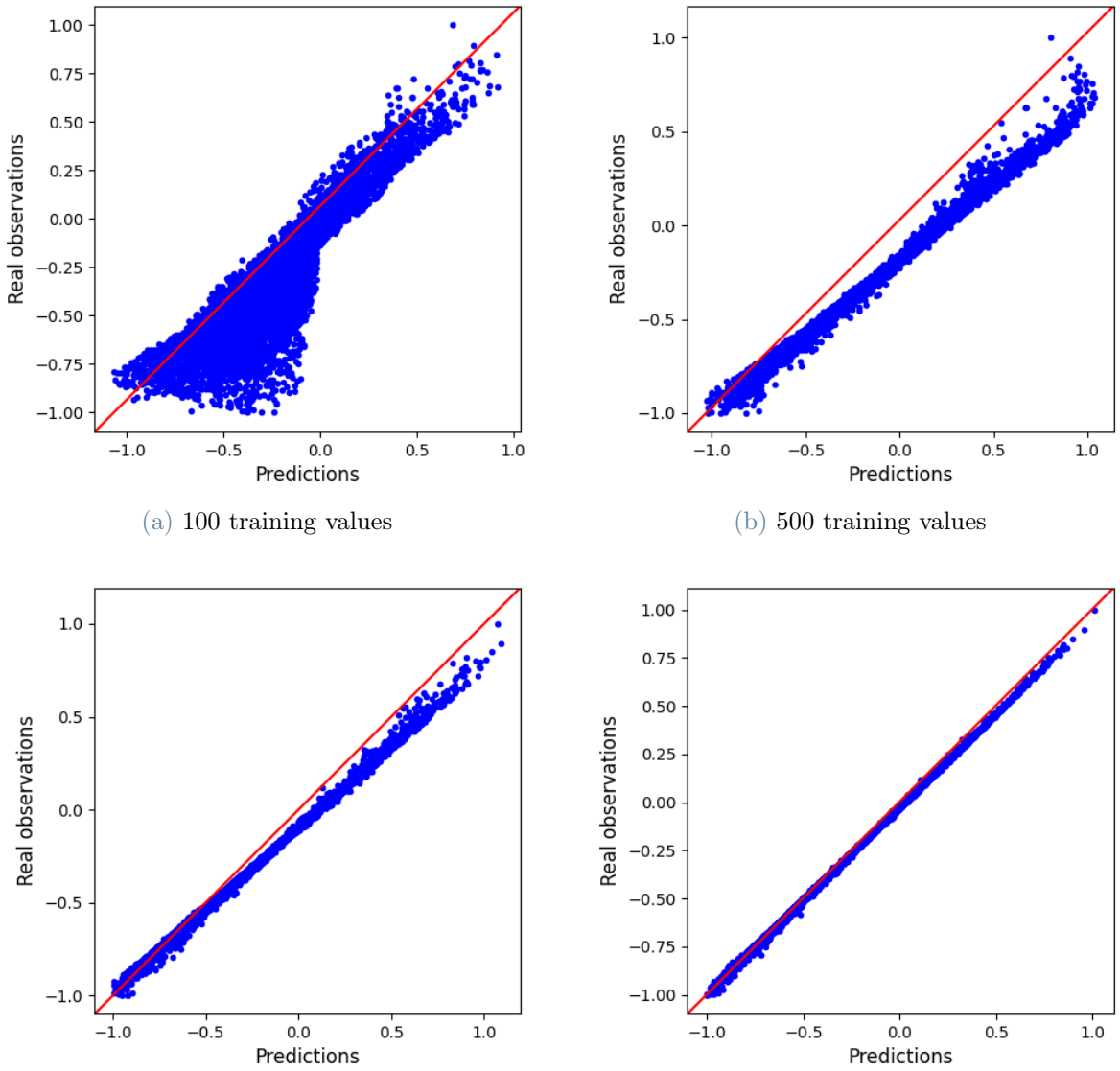
(a) 100 training values

(b) 500 training values

(c) 1000 training values

(d) 2000 training values

Figure 2.13: Behaviour of Gaussian process predictions with respect to the high-fidelity model observations with 100, 500, 1000 and 2000 training values as 2 parameters variate.



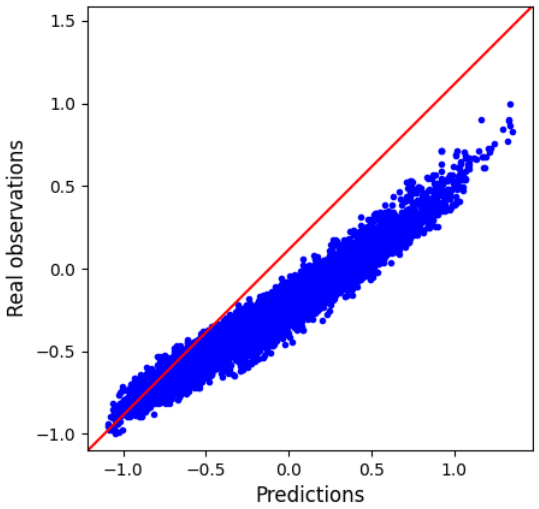
(a) 100 training values

(b) 500 training values

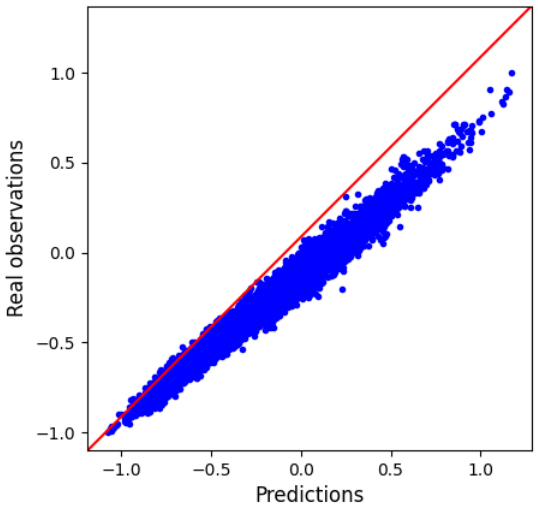
(c) 1000 training values

(d) 2000 training values

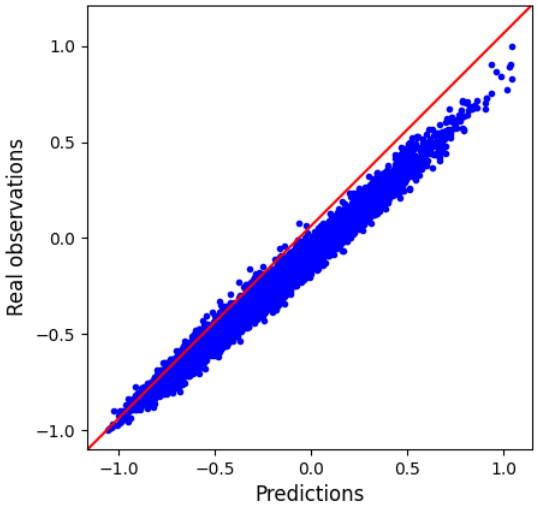
Figure 2.14: Behaviour of Gaussian process predictions with respect to the high-fidelity model observations with 100, 500, 1000 and 2000 training values as 4 parameters variate.



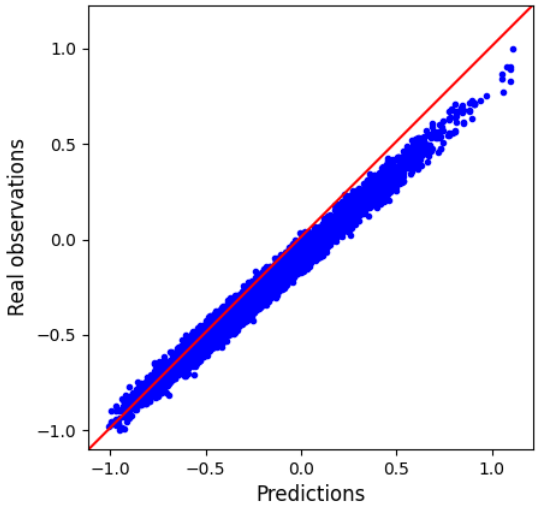
(a) 100 training values



(b) 500 training values

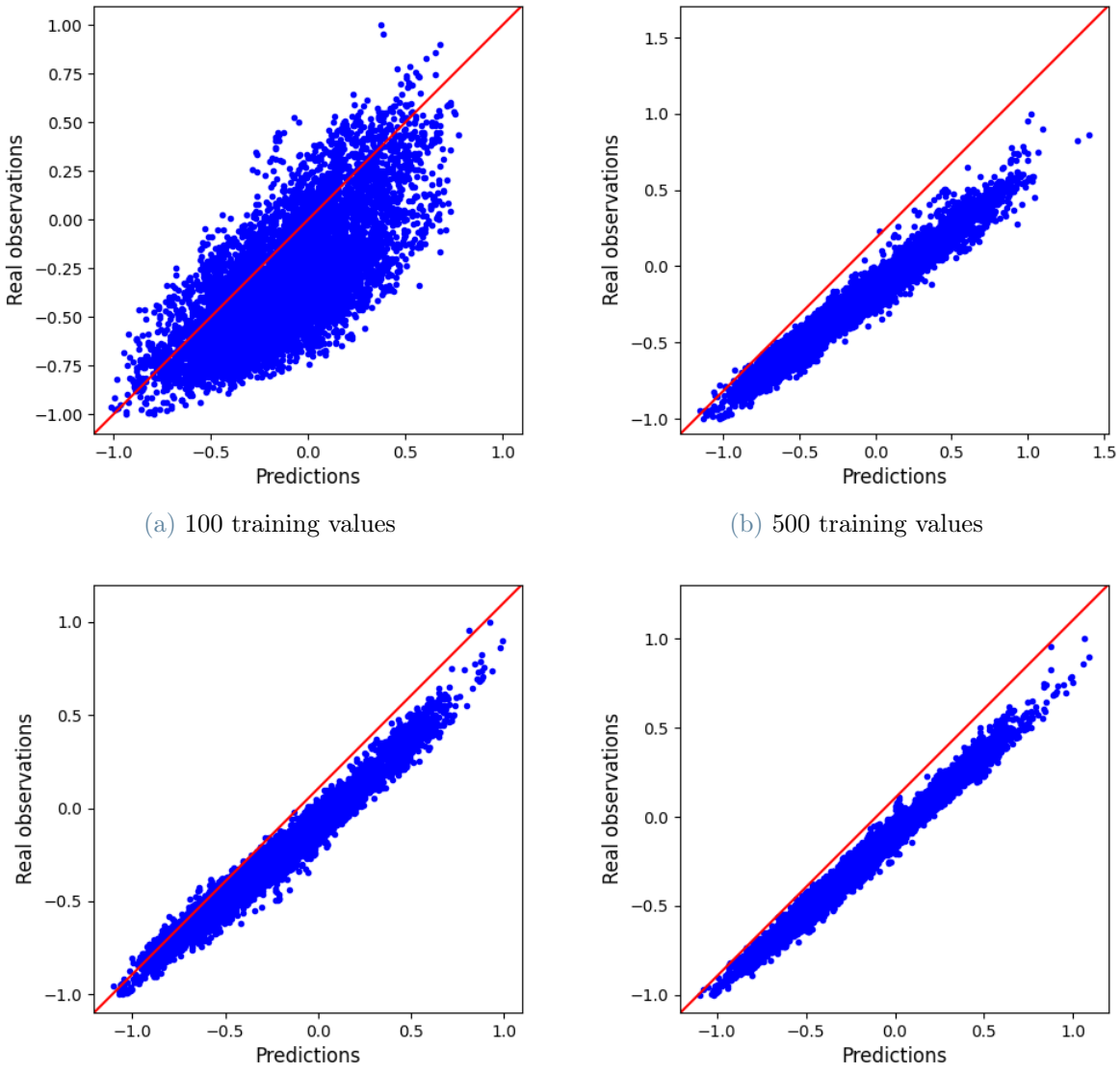


(c) 1000 training values



(d) 2000 training values

Figure 2.15: Behaviour of Gaussian process predictions with respect to the high-fidelity model observations with 100, 500, 1000 and 2000 training values as 8 parameters variate.



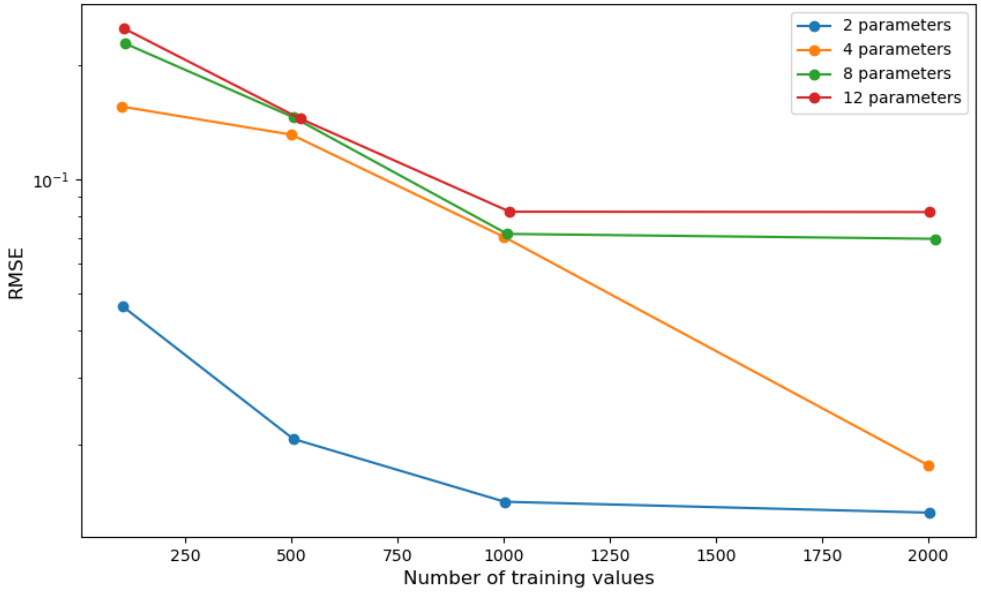
(a) 100 training values

(b) 500 training values

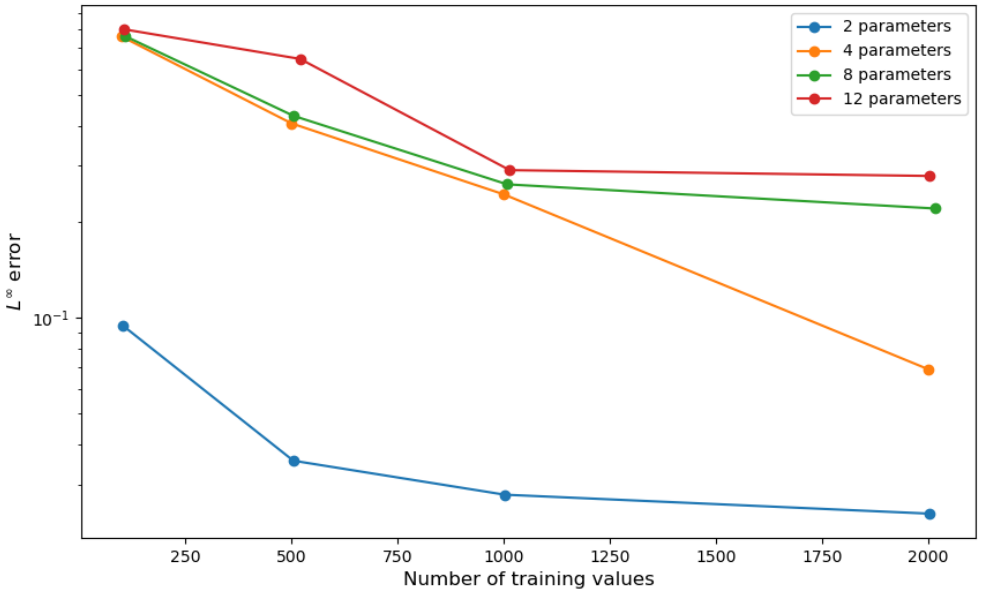
(c) 1000 training values

(d) 2000 training values

Figure 2.16: Behaviour of Gaussian process predictions with respect to the high-fidelity model observations with 100, 500, 1000 and 2000 training values as 12 parameters variate.



(a) RMSE



(b) L^∞ error

Figure 2.17: Behaviour of the root mean squared error and L^∞ error for the circulation model as the size of the training sample varies.

		N° of parameters			
N° of training values	**Error**	**2**	**4**	**8**	**12**
100	RMSE	0.0463	0.1558	0.2285	0.2508
	L^∞ error	0.0946	0.7601	0.7625	0.8019
500	RMSE	0.0207	0.1313	0.1460	0.1448
	L^∞ error	0.0357	0.4065	0.4296	0.6474
1000	RMSE	0.0141	0.0706	0.0719	0.0823
	L^∞ error	0.0279	0.2436	0.2619	0.2902
2000	RMSE	0.0132	0.0176	0.0698	0.0821
	L^∞ error	0.0243	0.0690	0.2200	0.2783

Table 2.11: Root mean squared errors and L^∞ errors for the circulation model as the size of the training sample varies.

Training set	**Hyperparameters**	**Predictions**
$\sim$ 101–1923 s	$\sim$ 31–13561 s	$\sim$ 14–25 s

Table 2.12: Computational time for each section of the process: generation of the training set, optimization of the hyperparameters and predictions.

3 | Forward uncertainty quantification

Mathematical models can be highly complex, making the relationships between inputs and outputs difficult to understand. In this context, model parameters are either directly measured or indirectly calibrated using experimental data. However, every experimental measurement is subject to variability. In particular, we can distinguish between intrinsic variability, which is inferred from inherent randomness, and extrinsic variability, which is related to the natural differences between individual samples. These variabilities are responsible of the uncertainty for a certain measured quantity, which results in some limitations on our confidence in the response of the model.

As reported in [15], mathematical models either treat inputs as quantities with a deterministic value, generating outputs that are deterministic as well, or are subject to an uncertainty in the inputs, which results in uncertain model outputs. This concept is illustrated in Figure 3.1, where two inputs are combined to produce an output by considering both the deterministic and the uncertainty affected approaches. The first one is illustrated with grey lines for I1, I2, O, which show that deterministic inputs result in a fixed value of the output. On the other hand, the uncertainty on the inputs is expressed by assigning a distribution to each input, as shown by the black lines, and results in uncertainty in the output as well. As explained in [15], it is also possible to observe that different effects on the model output distribution are highlightable depending on the chosen input distributions. Indeed, it is illustrated that doubling the width of the first input distribution (red line) or producing a similar change for the second input distribution (blue line) does not affect the output distribution in the same way.

This process for considering the impact of input uncertainties on outputs is called forward uncertainty quantification (UQ) (for more information about the uncertainty quantification, we refer to [18]).

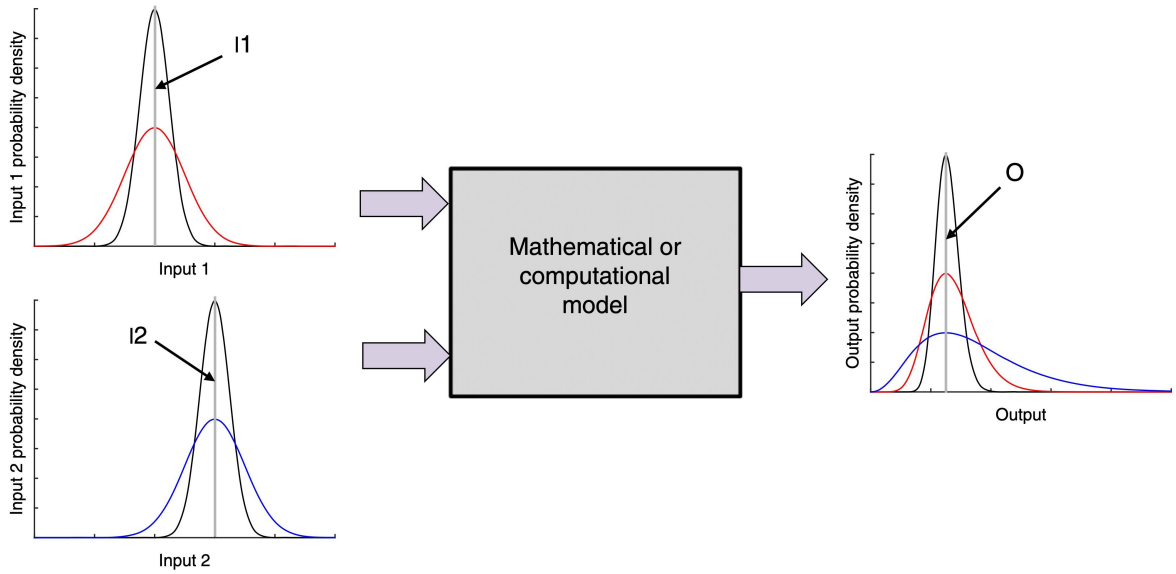


Figure 3.1: Illustrative example which shows how model inputs and outputs can be characterized as probability distributions rather than fixed values (image from [15]).

3.1. Sensitivity analysis

A related concept is the sensitivity analysis, which quantifies how sensitive model outputs are with respect to changes in model inputs. It can be used to determine a criterion to rank the most influential input parameters and, conversely, to identify which of them do not have a strong effect on a specific output.
As reported in [18], sensitivity analysis approaches can be divided in two groups: local and global methods. Basically, local methods are employed to analyze the impact of a single input parameter and are called local because the partial derivatives of the output with respect to input parameters are performed at a specific point in the parameter space. On the other hand, global methods examine the sensitivity with regard to the entire parameter distribution. They are able to provide more information about the model than local ones, since the input parameters are varied simultaneously [11].

3.1.1. Sobol indices

In this work, we consider a variance-based global sensitivity method which describes the amount of output variance generated from the variation of each parameter [24]. We will focus on the concept of Sobol indices, which relies on the following functional decomposition scheme reported in [16]. Suppose that the output of a deterministic process is governed by an unknown function $f : \mathbb{R}^d \to \mathbb{R}$ with inputs $\mathbf{x} = (x_1, \ldots, x_d)^T$. Since the

exact value of the input parameters is unknown, each parameter is considered as a random variable and the associated uncertainty is described in terms of probability distributions. This makes the model output Y a random variable as well, even if f is deterministic. Therefore,

$$Y = f(\mathbf{X}), \tag{3.1}$$

where now the input $\mathbf{X} = (X_1, \dots, X_d)^T$ consists of d statistically independent random variables with known distributions. The functional decomposition scheme is reported as follows:

$$Y = f(\mathbf{X}) = f_0 + \sum_{i=1}^{d} f_i(X_i) + \sum_{i} \sum_{j>i} f_{ij}(X_i, X_j) + \cdots + f_{1,\dots,d}(\mathbf{X}), \tag{3.2}$$

where f_0 is a constant. The remaining $2^d - 1$ elementary functions have zero mean and are orthogonal (mutually uncorrelated) with each other:

$$\mathbb{E}[f_i(X_i)] = 0, \tag{3.3}$$

$$\mathbb{E}[f_i(X_i) f_j(X_j)] = 0, \quad \forall i \neq j, \tag{3.4}$$

with $i, j \subseteq \mathcal{D} = \{1, \dots, d\}$. Applying the variance operator on both sides of (3.2), it is possible to obtain

$$\mathbb{V}(Y) = \sum_{i=1}^{d} \mathbb{V}_i + \sum_{i} \sum_{j>i} \mathbb{V}_{ij} + \cdots + \mathbb{V}_{1,\dots,d}, \tag{3.5}$$

where

$$\mathbb{V}_i = \mathbb{V}(f_i(X_i)) = \mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}[Y|X_i]), \tag{3.6}$$

$$\mathbb{V}_{ij} = \mathbb{V}(f_{ij}(X_i, X_j)) = \mathbb{V}_{X_i, X_j}(\mathbb{E}_{\mathbf{X}_{\sim i,j}}[Y|X_i, X_j]) - \mathbb{V}_i - \mathbb{V}_j \tag{3.7}$$

and the other terms are similarly defined. The notation $\mathbf{X}_{\sim i}$ ($\mathbf{X}_{\sim i,j}$) is used to indicate the set of all the input factors excluding X_i (X_i and X_j).
The Sobol indices are defined as

$$S_i = \frac{\mathbb{V}_i}{\mathbb{V}(Y)}, \tag{3.8}$$

$$S_{ij} = \frac{\mathbb{V}_{ij}}{\mathbb{V}(Y)}, \tag{3.9}$$

$$\vdots$$

where S_i is the first order (or main) effect of X_i, S_{ij} is the second order effect of (X_i, X_j) (which represents the contribution of interaction between X_i and X_j on the model output uncertainty without their individual effects), and so on. By definition of Sobol indices,

the following condition is obtained:

$$\sum_{i=1}^{d} S_i + \sum_{i} \sum_{j>i} S_{ij} + \cdots + S_{1,\ldots,d} = 1. \tag{3.10}$$

The first order Sobol index S_i is of great significance; indeed, it reflects the direct contribution of X_i on the total variance $\mathbb{V}(Y)$ and is used as a measure of importance of X_i. As reported in [16], the first order Sobol index can be also interpreted as the expected reduction in the total variance $\mathbb{V}(Y)$ when X_i is fixed to a constant. This can be shown by using the law of total variance (for more information, we refer to Appendix A):

$$S_i = \frac{\mathbb{V}_{X_i}(\mathbb{E}_{\mathbf{X}_{\sim i}}[Y|X_i])}{\mathbb{V}(Y)} = \frac{\mathbb{V}(Y) - \mathbb{E}_{X_i}[\mathbb{V}_{\mathbf{X}_{\sim i}}(Y|X_i)]}{\mathbb{V}(Y)}. \tag{3.11}$$

However, the first order Sobol index does not account for the interactions among parameters. For this reason, one may verify the importance of a parameter in determining a certain quantity of interest also accounting for the interactions among parameters, by resorting the total order Sobol index, defined as

$$S_i^T = \frac{\mathbb{V}(Y) - \mathbb{V}_{\mathbf{X}_{\sim i}}(\mathbb{E}_{X_i}[Y|\mathbf{X}_{\sim i}])}{\mathbb{V}(Y)} = \frac{\mathbb{E}_{\mathbf{X}_{\sim i}}[\mathbb{V}_{X_i}(Y|\mathbf{X}_{\sim i})]}{\mathbb{V}(Y)}, \tag{3.12}$$

where $\mathbb{V}_{\mathbf{X}_{\sim i}}(\mathbb{E}_{X_i}[Y|\mathbf{X}_{\sim i}])$ stands for the first order effect of $\mathbf{X}_{\sim i}$, i.e. all the factors but X_i.

As explained in [16], only the first and total order indices are generally considered for sensitivity studies in practical circumstances. The main effects are typically used in order to provide the ranking of the input parameters according to their contribution to the total variance of the output. On the other hand, the total effects are useful where insignificant inputs are set to a given value over their range of uncertainty. In particular,

- the condition $S_i^T = 0$ means that X_i is a non-influential parameter;
- if S_i^T is almost zero, then X_i can be fixed to any value in its range without changing the value of the output variance.

Finally, the following relation holds between the first and total effect indices given that the input parameters are independent:

$$\sum_{i=1}^{d} S_i \leq 1 \leq \sum_{i=1}^{d} S_i^T. \tag{3.13}$$

3.2. Numerical tests

In this section, we combine together some important tools of the variance-based global sensitivity analysis, such as the first order Sobol indices, and the Gaussian process regression discussed in Chapter 2. We aim to provide a sufficiently accurate posterior predictive distribution from GP regression to compute the first order Sobol indices by means of the open source Python library `SALib`.
We perform this analysis for the high-fidelity circulation model in the four cases reported in Table 2.9, where the output is the one in Table 2.10. The idea is to prove that it is possible to obtain a reliable ranking of the input parameters with respect to the predictions by comparing these results with those obtained in the high-fidelity model.
In Tables 3.1–3.2 we can observe that the first order Sobol indices are approximately equivalent to the ones obtained with the high-fidelity model when the number of input parameters is sufficiently small. This result is in line with what we observed in Section 2.6.4, where the Gaussian process posterior predictive distribution proved to be particularly efficient whatever it was the size of the training sample.

	N° of training values				
Parameter	**100**	**500**	**1000**	**2000**	**High-fidelity model**
$R_{\text{AR}}^{\text{SYS}}$	0.7173	0.7105	0.7117	0.7118	0.7113
$C_{\text{AR}}^{\text{SYS}}$	0.2833	0.2908	0.2898	0.2896	0.2901

Table 3.1: First order Sobol indices with 2 parameters as the number of training values varies with respect to the ones obtained with the high-fidelity model.

	N° of training values				
Parameter	**100**	**500**	**1000**	**2000**	**High-fidelity model**
$R_{\text{AR}}^{\text{SYS}}$	0.2846	0.3759	0.3657	0.3674	0.3644
$C_{\text{AR}}^{\text{SYS}}$	0.0694	0.1403	0.1448	0.1418	0.1437
$E_{\text{LV}}^{\text{act}}$	0.0720	0.0639	0.0628	0.0629	0.0618
$E_{\text{LV}}^{\text{pass}}$	0.3872	0.3944	0.4071	0.4079	0.4057

Table 3.2: First order Sobol indices with 4 parameters as the number of training values varies with respect to the ones obtained with the high-fidelity model.

On the other hand, we know that the situation becomes complicated when many varying input parameters are involved. For this reason, we expect that the ranking of the input parameters with respect to Gaussian process predictions tends to approach the real one in a sharper way as the number of training values increases.
Tables 3.3–3.4 show that the values of the first order Sobol indices significantly change for some parameters from the lowest number of training values to the greatest one (e.g., $E_{\mathrm{LV}}^{\mathrm{pass}}$ passes from 0.3854 with 100 training values to 0.4298 with 2000 training values for the case with 12 parameters). This can also be observed in Figures 3.2–3.5, where pie and Pareto charts are provided in order to highlight the behaviour of parameters ranking as the number of training values varies.
We can observe that the reliability of the sensitivity analysis performed on Gaussian process predictions is rather low for a small number of training values. For instance, Figures 3.4–3.5 show that the order of parameters with a lower impact on the output considerably changes from the case with minimum number of training values to the case with maximum number of them. However, these charts suggest that it is always possible to obtain a significant improvement by increasing the size of the training set.

	N° of training values				
Parameter	**100**	**500**	**1000**	**2000**	**High-fidelity model**
$R_{\mathrm{AR}}^{\mathrm{SYS}}$	0.3277	0.2871	0.2885	0.2900	0.2861
$C_{\mathrm{AR}}^{\mathrm{SYS}}$	0.1307	0.1222	0.1278	0.1198	0.1219
$E_{\mathrm{LV}}^{\mathrm{act}}$	0.0564	0.0664	0.0590	0.0602	0.0595
$E_{\mathrm{LV}}^{\mathrm{pass}}$	0.3843	0.3898	0.4092	0.4106	0.3987
$T_{\mathrm{LV}}^{\mathrm{contr}}$	0.0496	0.0418	0.0443	0.0491	0.0487
$T_{\mathrm{LV}}^{\mathrm{rel}}$	0.0273	0.0321	0.0317	0.0298	0.0301
$t_{\mathrm{LA}}^{\mathrm{del}}$	0.0010	0.0015	0.0209	0.0118	0.0223
$t_{\mathrm{RA}}^{\mathrm{del}}$	0.0010	0.0086	0.0075	0.0015	0.0066

Table 3.3: First order Sobol indices with 8 parameters as the number of training values varies with respect to the ones obtained with the high-fidelity model.

	N° of training values				
Parameter	**100**	**500**	**1000**	**2000**	**High-fidelity model**
$R_{\text{AR}}^{\text{SYS}}$	0.3137	0.3025	0.3031	0.2966	0.2976
$C_{\text{AR}}^{\text{SYS}}$	0.1021	0.1023	0.0979	0.0962	0.1032
$E_{\text{LV}}^{\text{act}}$	0.0615	0.0758	0.0659	0.0710	0.0697
$E_{\text{LV}}^{\text{pass}}$	0.3854	0.4236	0.4322	0.4298	0.4244
$T_{\text{LV}}^{\text{contr}}$	0.0556	0.0218	0.0337	0.0169	0.0291
$T_{\text{LV}}^{\text{rel}}$	0.0218	0.0182	0.0217	0.0211	0.0225
$t_{\text{LA}}^{\text{del}}$	0.0076	0.0084	0.0032	0.0080	0.0206
$t_{\text{RA}}^{\text{del}}$	0.0031	0.0034	0.0000	0.0049	0.0041
$R_{\text{VEN}}^{\text{SYS}}$	0.0246	0.0222	0.0259	0.0173	0.0182
$C_{\text{VEN}}^{\text{SYS}}$	0.0010	0.0006	0.0010	0.0023	0.0001
$R_{\text{VEN}}^{\text{PUL}}$	0.0144	0.0160	0.0182	0.0171	0.0160
$C_{\text{VEN}}^{\text{PUL}}$	0.0008	0.0028	0.0009	0.0002	0.0035

Table 3.4: First order Sobol indices with 12 parameters as the number of training values varies with respect to the ones obtained with the high-fidelity model.

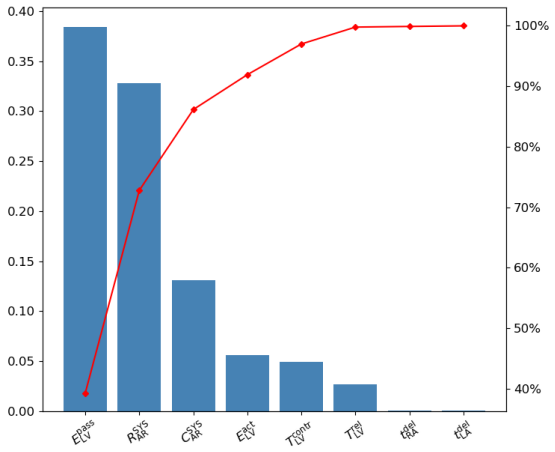
This situation is a consequence of what we observed in Figures 2.15–2.16; indeed, if the number of training values is not sufficient to guarantee a good reliability of Gaussian process predictions, then the root mean squared error and the L^∞ error are not negligible and a less accurate estimation of the indices is obtained. However, the situation can be extremely improved by increasing the number of training values.
We can conclude that the improvement of the accuracy of the Gaussian process predictions and the estimation of the first order Sobol indices depend on the size of the training set, in the sense that better results are observable by increasing the number of training samples, but with a trade-off in terms of computational cost (see Table 2.12).

(a) 100 training values

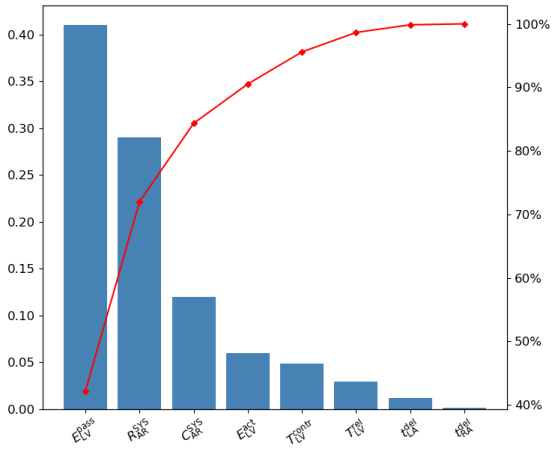
(b) 2000 training values

(c) High-fidelity model

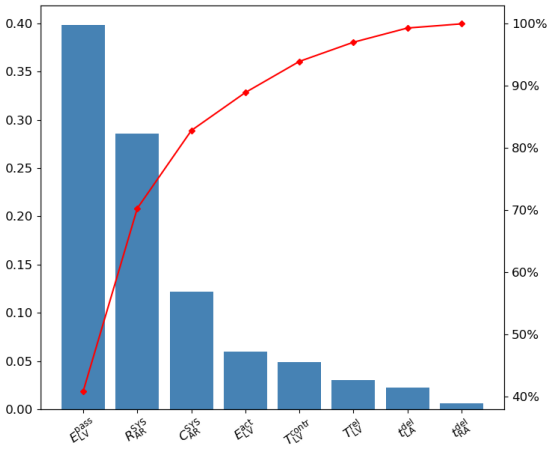
Figure 3.2: Comparison between pie charts with respect to Gaussian process predictions for the minimum/maximum number of training values and the one obtained from the high-fidelity circulation model in the case with 8 parameters.



(a) 100 training values

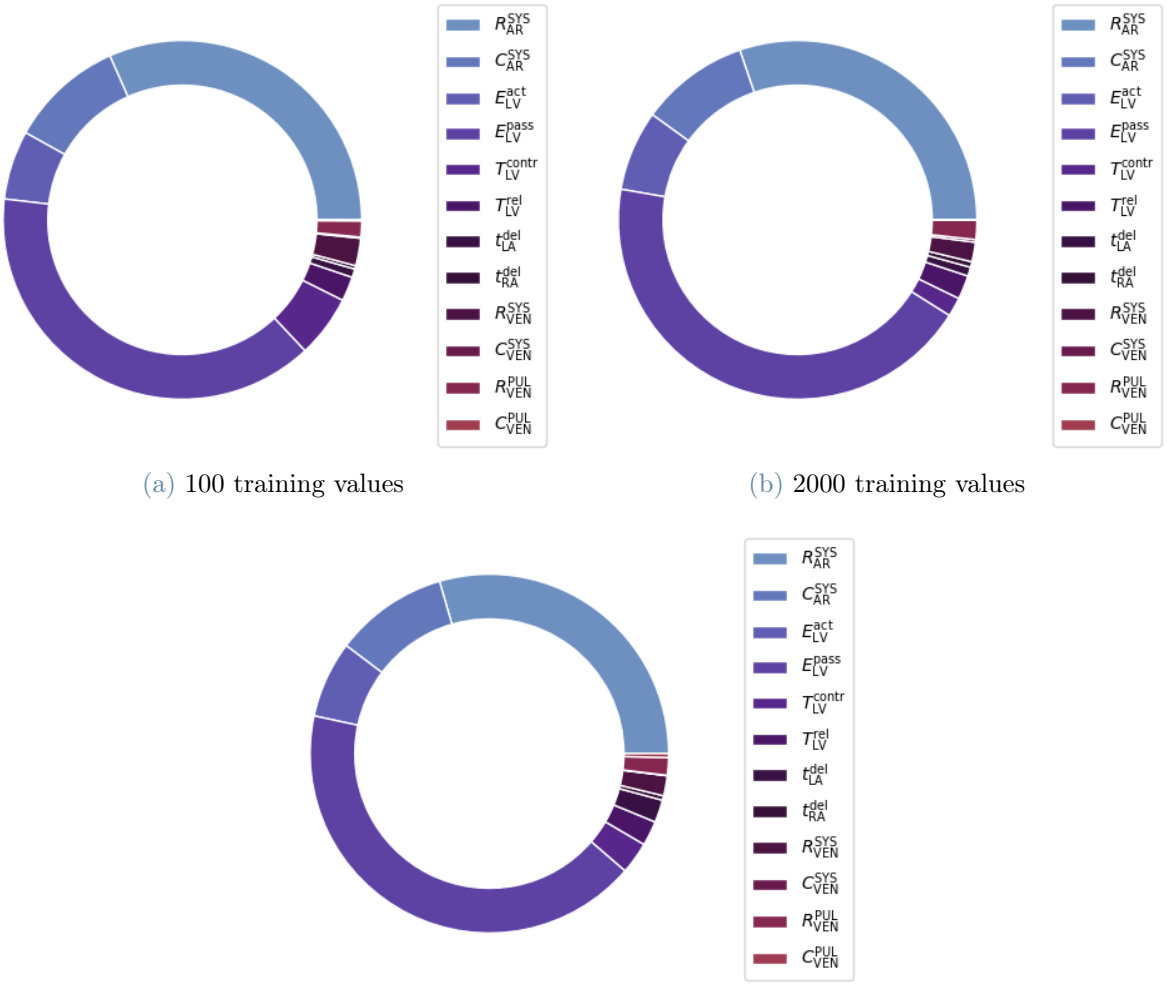


(b) 2000 training values



(c) High-fidelity model

Figure 3.3: Comparison between Pareto charts with respect to Gaussian process predictions for the minimum/maximum number of training values and the one obtained from the high-fidelity circulation model in the case with 8 parameters.

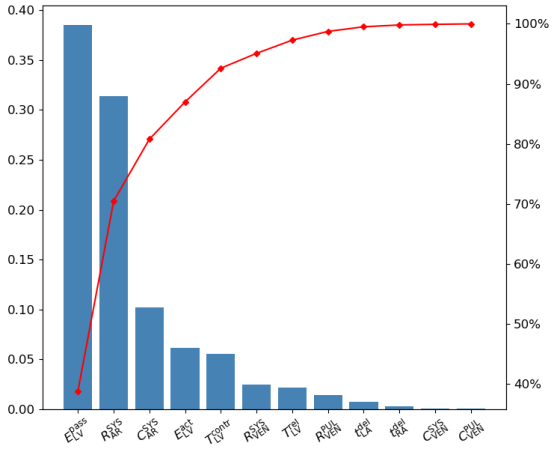


(a) 100 training values

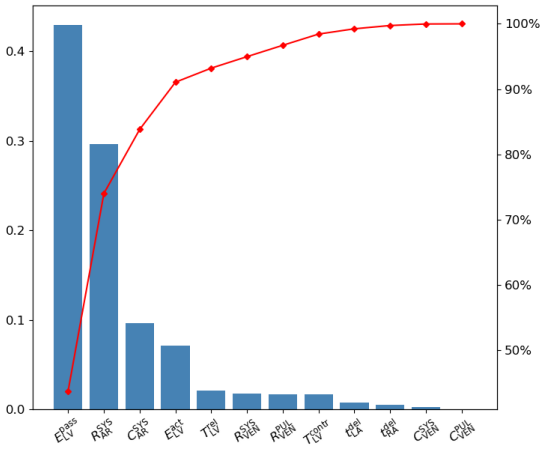
(b) 2000 training values

(c) High-fidelity model

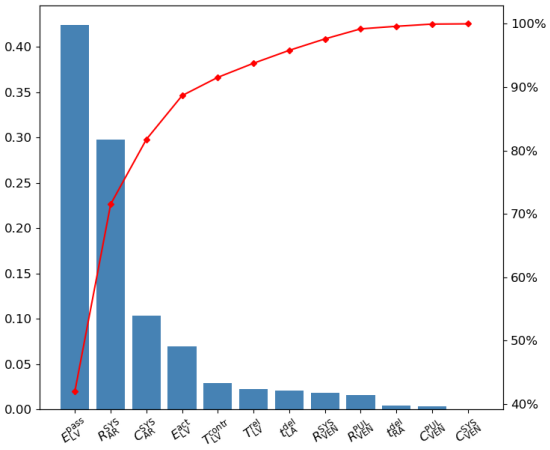
Figure 3.4: Comparison between pie charts with respect to Gaussian process predictions for the minimum/maximum number of training values and the one obtained from the high-fidelity circulation model in the case with 12 parameters.



(a) 100 training values



(b) 2000 training values



(c) High-fidelity model

Figure 3.5: Comparison between Pareto charts with respect to Gaussian process predictions for the minimum/maximum number of training values and the one obtained from the high-fidelity circulation model in the case with 12 parameters.

4 | Parameter estimation under uncertainty

In this chapter, we want to replicate a procedure for the Bayesian parameter estimation reported in [24], with the difference that here we replace the ANN reduced-order model by the Gaussian process. The goal is to demonstrate that Gaussian processes are able to provide a reliable estimate of the unknown parameters taking into account the associated uncertainty.

4.1. Bayesian parameter estimation

As explained in [24], the patient-specific personalization of the circulation model requires the estimation of several parameters starting from clinical measurements. However, not all the required scalar quantities are usually available for this purpose and, when present, they are affected by noise.

For this reason, the Markov chain Monte Carlo method is exploited, which is a Bayesian method that allows us to solve the inverse problem (i.e. estimating the parameters $\mathbf{x} \in \mathcal{X}$ from the outputs $\mathbf{y} \in \mathcal{Y}$) estimating the impact of the noise that affects the measurement of the quantities of interest and that reflects in uncertainty on parameters (for more information, we refer to [5]).

As reported in [24], this method is able to provide the probability distribution of the parameters given the observations (denoted by $\mathbf{y}$); in particular, it is based on the computation of the credibility for each parameters combination, which is strictly related to the uncertainty on measurements. This uncertainty is encoded by the covariance matrix Σ, which is associated to the noise on the observations, and the prior distribution on parameters $\pi_{\text{prior}}(\mathbf{x})$, which provides important prior information about parameters.

Given these assumptions, we can refer to the observations as

$$\mathbf{y} = f(\mathbf{x}) + \epsilon_{\text{exp}}, \tag{4.1}$$

where $f : \mathcal{X} \to \mathcal{Y}$ represents the map between parameters and outputs and $\epsilon_{\text{exp}} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ denotes the experimental measurement error. Based on Bayes' theorem, the posterior distribution of parameters is given by

$$\pi_{\text{post}}(\mathbf{x}) = \frac{1}{Z}\mathcal{N}(\mathbf{y}|f(\mathbf{x}), \Sigma)\pi_{\text{prior}}(\mathbf{x}), \tag{4.2}$$

where the normalization constant Z is defined as

$$Z = \int_{\mathcal{X}} \mathcal{N}(\mathbf{y}|f(\mathbf{x}'), \Sigma) d\pi_{\text{prior}}(\mathbf{x}'). \tag{4.3}$$

In practice, an approximation of the posterior distribution (4.2) can be provided by exploiting the MCMC method, which results significantly useful to reduce the computational burden. Indeed, it generates a non-intrusive method which analyzes a large number of model evaluations for different parameter values with the aim of providing a suitable approximation of the posterior distribution.
As proposed in [24], since the MCMC method only needs the evaluations of the map $f : \mathcal{X} \to \mathcal{Y}$, it is possible to use Gaussian processes instead of the high-fidelity circulation model in order to reduce the computational time. However, this implies that a further error must be taken into account. Indeed, if we denote by $\tilde{f}$ the approximation of the high-fidelity model by the Gaussian process, then

$$f(\mathbf{x}) = \tilde{f}(\mathbf{x}) + \epsilon_{\text{GP}}, \tag{4.4}$$

where $\epsilon_{\text{GP}} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\text{GP}})$ is the approximation error obtained by considering the Gaussian process evaluations instead of the high-fidelity circulation model outputs. It follows that

$$\mathbf{y} = \tilde{f}(\mathbf{x}) + \epsilon_{\text{GP}} + \epsilon_{\text{exp}}, \tag{4.5}$$

where the two sources of error can be assumed independent. The independence assumption is crucial, since it allows us to take into account the error introduced by the Gaussian process in the parameter estimation procedure by simply adding it to the measurement error. Indeed, it is possible to write $\Sigma = \Sigma_{\text{GP}} + \Sigma_{\text{exp}}$, where Σ_{GP} represents the Gaussian process covariance matrix (it encodes the error introduced by the usage of Gaussian process predictions) and Σ_{exp} is the experimental error covariance matrix mentioned above.

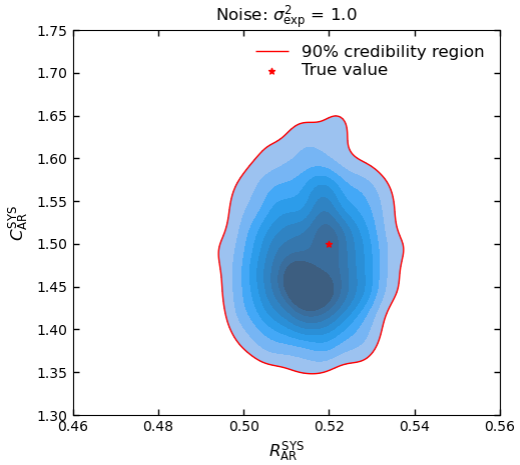
4.2. Numerical tests

Numerical tests are performed by means of the open source Python library `UQpy`. The Bayesian inverse problem is solved by exploiting the MCMC method. We consider 500 samples per chain, a jump of 5 samples and a burn-in of 500 samples, with a number of chains equal to 5.
As pointed out in [24], the procedure used to prove the capability of Gaussian processes to accelerate the estimation of parameters can be summarized as follows:

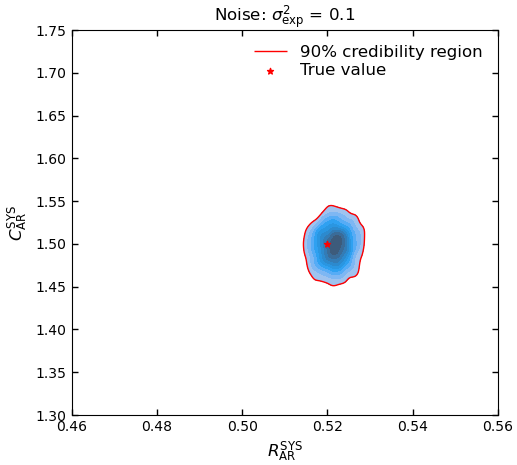
- we perform a simulation with the high-fidelity circulation model from which we derive a set of observations $\mathbf{y}$ to which we add a synthetic measurement noise;
- we employ the Gaussian process instead of the high-fidelity circulation model in the Bayesian estimation of parameters performed by means of the MCMC method;
- we validate the obtained results with respect to the values used to generate the observations.

We consider a pair of parameters, namely $(R_{\text{AR}}^{\text{SYS}}, C_{\text{AR}}^{\text{SYS}})$, which are the systemic arterial resistance and the systemic arterial capacitance respectively, whereas the remaining ones are fixed at their baseline value. For parameters under investigation we prescribe a value that is slightly different from the baseline one, such as $R_{\text{AR}}^{\text{SYS}} = 0.52$ mmHg s mL^{-1} and $C_{\text{AR}}^{\text{SYS}} = 1.5$ mL mmHg^{-1}. We run a simulation with respect to these values of parameters using the high-fidelity circulation model and select some quantities of interest, which consist of the minimum and maximum arterial pressures. We voluntarily choose these outputs in order to make this test as close as possible to real-life problems. Indeed, the goal is to provide a reliable estimate of the mentioned parameters, which cannot be easily measured, starting from outputs that, conversely, can be measured in a non-invasively way, as in this case.
The goal is to demonstrate that Gaussian process emulators are able to estimate the parameters and quantify the uncertainty associated with this estimate, reducing the computational effort. As we anticipated before, this can be done by exploiting the MCMC method for the Bayesian parameters estimation starting from noisy measurements of the minimum and maximum arterial pressures. The measurement errors are added as artificial noises with zero mean and variance $\sigma_{\text{exp}}^2 = \{1.0, 0.1, 0.01\}$ mmHg^2 over the output values given by the high-fidelity model. For the unknown parameters a uniform prior is given with respect to the ranges used to train the Gaussian process, namely $R_{\text{AR}}^{\text{SYS}} \in [0.32, 0.96]$ mmHg s mL^{-1} and $C_{\text{AR}}^{\text{SYS}} \in [0.6, 1.8]$ mL mmHg^{-1}. Then we set $\Sigma = \Sigma_{\text{GP}} + \Sigma_{\text{exp}}$, where the experimental error covariance matrix is given by $\Sigma_{exp} = \sigma_{\text{exp}}^2 \mathbb{I}$, whereas the Gaussian

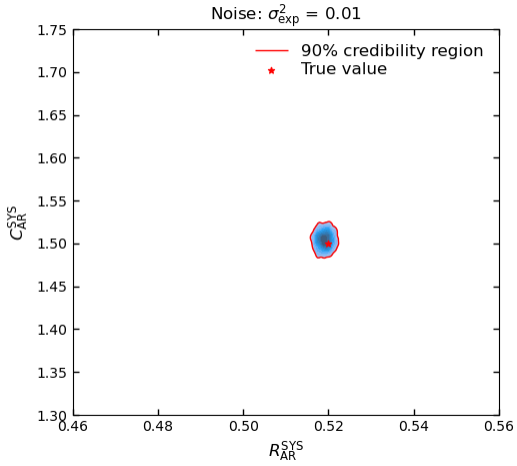
process covariance matrix is diagonal with entries on the main diagonal given by the observations noise variances obtained from the optimization of the GP hyperparameters. Figure 4.1 shows the obtained results for the posterior distribution π_{post} on the parameters pair $(R_{\text{AR}}^{\text{SYS}}, C_{\text{AR}}^{\text{SYS}})$ as the noise level varies. The red line defines the 90% credibility region, whereas the exact value of the parameters, is marked with a "star" sign.



(a) $\sigma^2_{\text{exp}} = 1.0$ mmHg^2



(b) $\sigma^2_{\text{exp}} = 0.1$ mmHg^2



(c) $\sigma^2_{\text{exp}} = 0.01$ mmHg^2

Figure 4.1: Posterior distribution on the parameters pair $(R_{\text{AR}}^{\text{SYS}}, C_{\text{AR}}^{\text{SYS}})$ (prescribed values $R_{\text{AR}}^{\text{SYS}} = 0.52$ mmHg s mL^{-1} and $C_{\text{AR}}^{\text{SYS}} = 1.5$ mL mmHg^{-1}) computed by means of the MCMC method with the Gaussian process for $\sigma^2_{\text{exp}} = \{1.0, 0.1, 0.01\}$ mmHg^2.

We notice that for each value of the measurement noise, the credibility region contains the exact value of the parameters and, as expected, for larger values of the noise, the

size of the credibility region increases, since the estimate is more uncertain. Moreover, it is possible to detect from the output of Bayesian parameter estimation if there is a correlation among the estimated parameters. Indeed, a straight shape of the credibility region is showed in Figure 4.1, which implies a no strong correlation between $R_{\mathrm{AR}}^{\mathrm{SYS}}$ and $C_{\mathrm{AR}}^{\mathrm{SYS}}$. However, if we consider a different pair of parameters, namely $(R_{\mathrm{AR}}^{\mathrm{SYS}}, E_{\mathrm{LV}}^{\mathrm{act}})$, where $E_{\mathrm{LV}}^{\mathrm{act}} \in [1.675, 5.025]$ mmHg mL^{-1} represents the active elastance of the left ventricle, then a different behaviour is observable. Indeed, Figure 4.2 shows an oblique shape of the credibility region which may intend a higher correlation between parameters.

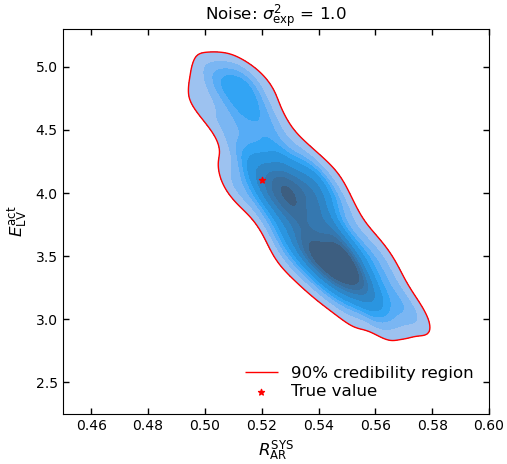


Figure 4.2: Posterior distribution on the parameters pair $(R_{\mathrm{AR}}^{\mathrm{SYS}}, E_{\mathrm{LV}}^{\mathrm{act}})$ (prescribed values $R_{\mathrm{AR}}^{\mathrm{SYS}} = 0.52$ mmHg s mL^{-1} and $E_{\mathrm{LV}}^{\mathrm{act}} = 4.1$ mmHg mL^{-1}) computed by means of the MCMC method with the Gaussian process for $\sigma^2_{\mathrm{exp}} = 1.0$ mmHg^2.

A more quantitative analysis of how well the posterior distribution is able to estimate the parameter values by solving the inverse problem with a Gaussian process based approach can be performed through the mean value and the maximum a posteriori (MAP) reported in Table 4.1. Indeed, it is possible to observe that both these indicators tend to the prescribed value of the parameters pair $(R_{\mathrm{AR}}^{\mathrm{SYS}}, C_{\mathrm{AR}}^{\mathrm{SYS}})$ as the measurement noise decreases, proving the efficiency of the obtained posterior distribution.
To prove the efficacy of Gaussian processes in reducing the computational burden required by the Bayesian inverse problem, we can compare the obtained results with the posterior distribution on parameters produced by the same approach with the high-fidelity circulation model.

Parameter	Indicator	$\sigma^2_{\text{exp}} = 1.0$	$\sigma^2_{\text{exp}} = 0.1$	$\sigma^2_{\text{exp}} = 0.01$
$R^{\text{SYS}}_{\text{AR}}$	Mean	0.5160	0.5216	0.5191
	MAP	0.5171	0.5224	0.5197
$C^{\text{SYS}}_{\text{AR}}$	Mean	1.4851	1.4977	1.5042
	MAP	1.4515	1.4995	1.5059

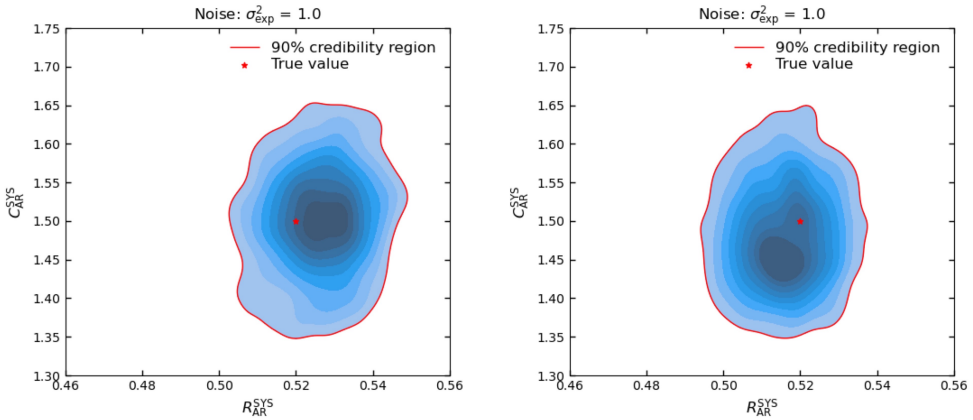
Table 4.1: Mean and MAP indicators of the posterior distribution on the parameters pair $(R^{\text{SYS}}_{\text{AR}}, C^{\text{SYS}}_{\text{AR}})$ (prescribed values $R^{\text{SYS}}_{\text{AR}} = 0.52$ mmHg s mL^{-1} and $C^{\text{SYS}}_{\text{AR}} = 1.5$ mL mmHg^{-1}) computed by means of the MCMC method with the Gaussian process for $\sigma^2_{\text{exp}} = \{1.0, 0.1, 0.01\}$ mmHg^2.

Figure 4.3 shows this comparison between the output of Bayesian parameter estimation with respect to the parameters pair $(R^{\text{SYS}}_{\text{AR}}, C^{\text{SYS}}_{\text{AR}})$ achieved by exploiting the high-fidelity circulation model and the one obtained with the GP based approach. It is possible to observe that the size of the credibility region in the high-fidelity case is slightly smaller for each value of the measurement noise. However, this does not represent an advantageous trade-off if compared with the GP based approach, since it results in a computationally expensive method (see Table 4.2) with no substantial improvements, especially as the measurement noise gets larger.

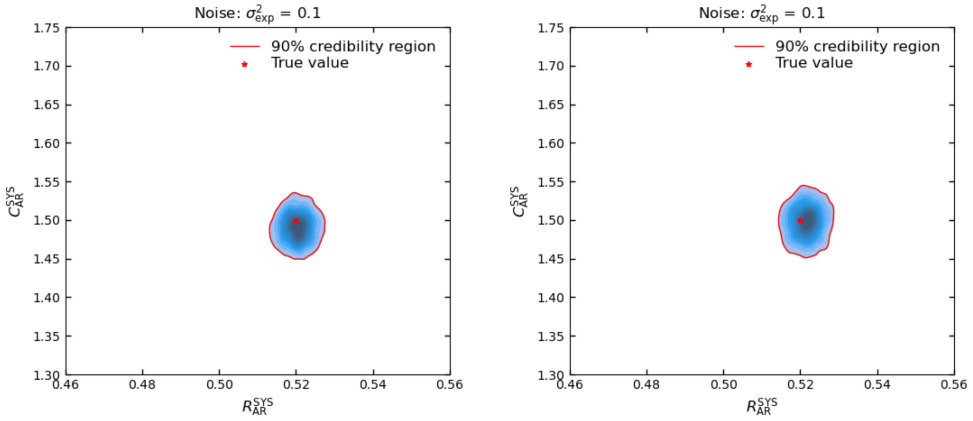
Gaussian process based approach	High-fidelity circulation model
$\sim$ 1153 s	$\sim$ 13728 s

Table 4.2: Computational time required for Bayesian parameter estimation by exploiting Gaussian processes and the high-fidelity circulation model.

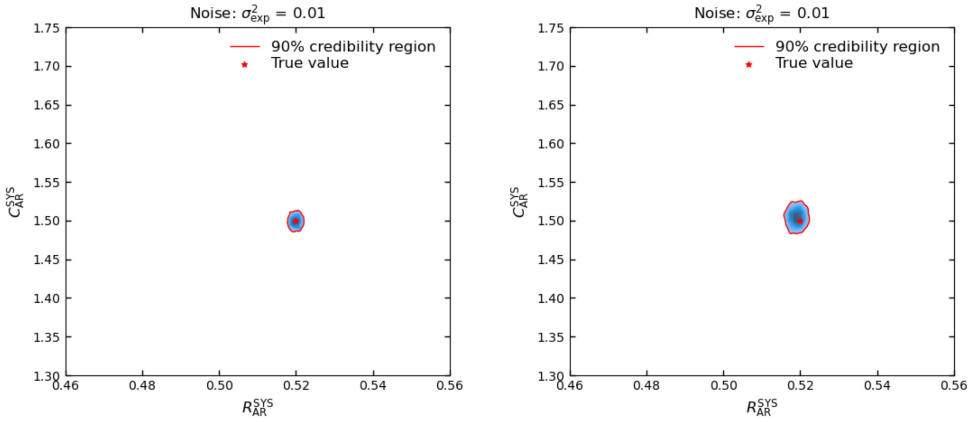
The computational time required using the high-fidelity circulation model in the MCMC method is actually ten times that of the Gaussian process based approach. We pass from almost 4 hours to about 20 minutes of simulation with Gaussian processes, obtaining results which are overall comparable with those achieved with the high-fidelity circulation model. Therefore, we can conclude that the computational effort required to perform the Bayesian parameter estimation significantly reduces if Gaussian process predictions are used, also obtaining reliable results.



(a) $\sigma^2_{\text{exp}} = 1.0$ mmHg^2



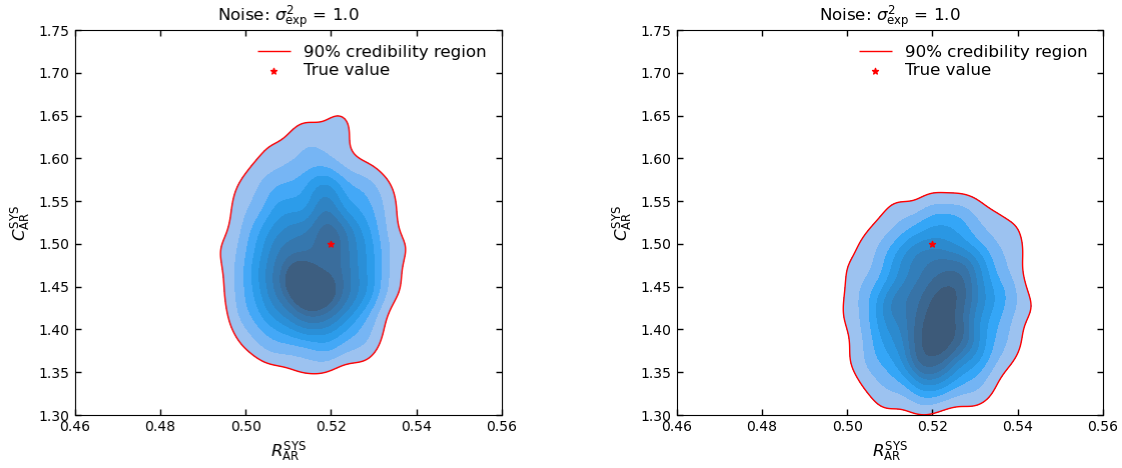
(b) $\sigma^2_{\text{exp}} = 0.1$ mmHg^2



(c) $\sigma^2_{\text{exp}} = 0.01$ mmHg^2

Figure 4.3: Posterior distributions on the parameters pair $(R^{\text{SYS}}_{\text{AR}}, C^{\text{SYS}}_{\text{AR}})$ (prescribed values $R^{\text{SYS}}_{\text{AR}} = 0.52$ mmHg s mL^{-1} and $C^{\text{SYS}}_{\text{AR}} = 1.5$ mL mmHg^{-1}) computed by means of the MCMC method with the high-fidelity model (on the left) and the Gaussian process for $\sigma^2_{\text{exp}} = \{1.0, 0.1, 0.01\}$ mmHg^2.

Several analyses can be performed to understand how effective Gaussian processes are in reducing the overall computational burden preserving a significant reliability of predictions. As we mentioned in Chapter 2, one of the main characteristics of Gaussian processes is that, once suitably trained, they are able to provide the mean and the corresponding variance of the modeled function for each data point. This means that a different error estimate associated to the usage of Gaussian process predictions can be encoded by this variance for each value of the input parameters.
Instead of considering a fixed Gaussian process covariance matrix as in Figure 4.1, we propose to employ a varying diagonal matrix with entries on the main diagonal given by the predictive variances of the Gaussian process for each set of input parameters. The goal is to verify if it is possible to improve the Gaussian process based approach for the Bayesian parameter estimation by taking full advantage of Gaussian process properties. Figure 4.4 shows the posterior distributions with respect to the parameters pair $(R_{\mathrm{AR}}^{\mathrm{SYS}}, C_{\mathrm{AR}}^{\mathrm{SYS}})$ for the cases mentioned above.



(a) Fixed GP error

(b) Varying GP error

Figure 4.4: Posterior distributions on the parameters pair $(R_{\mathrm{AR}}^{\mathrm{SYS}}, C_{\mathrm{AR}}^{\mathrm{SYS}})$ (prescribed values $R_{\mathrm{AR}}^{\mathrm{SYS}} = 0.52\ \mathrm{mmHg\,s\,mL^{-1}}$ and $C_{\mathrm{AR}}^{\mathrm{SYS}} = 1.5\ \mathrm{mL\,mmHg^{-1}}$) computed by means of the MCMC method with the Gaussian process considering a fixed GP error in panel (a) and a varying GP error in panel (b) for $\sigma^2_{\mathrm{exp}} = 1.0\ \mathrm{mmHg}^2$.

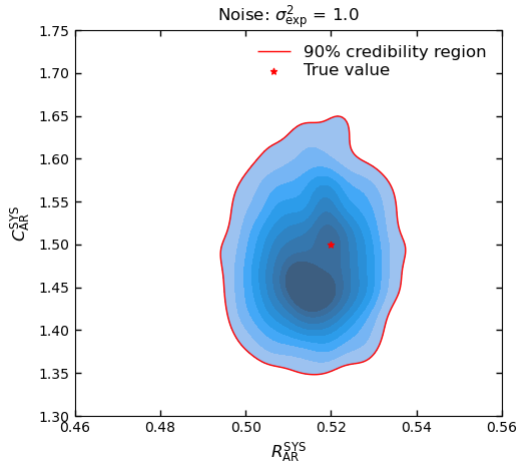
Figure 4.4a is equivalent to Figure 4.1a, whereas Figure 4.4b shows the output of the Bayesian parameter estimation with a measurement noise $\sigma^2_{\mathrm{exp}} = 1.0\ \mathrm{mmHg}^2$ considering that GP error varies for each value of the input parameters. Notice that, even if this procedure allows us to exploit the predictive variances given by the Gaussian process, there are not considerable improvements. Moreover, this approach results even more

penalizing than the one with fixed GP error computationally speaking. Indeed, as shown in Table 4.3, the computational time required using a varying GP error is twice what we would obtain considering a fixed GP error.

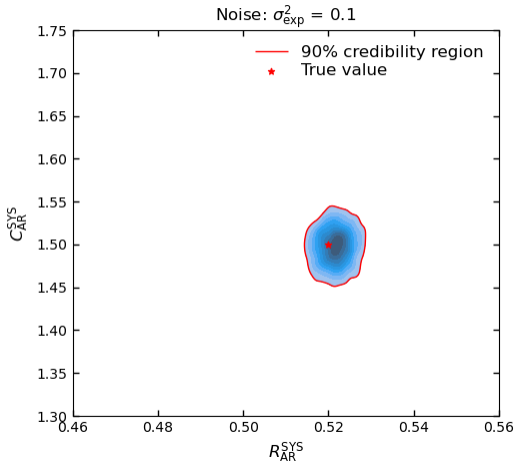
Fixed GP error	**Varying GP error**
~ 1153 s	~ 2312 s

Table 4.3: Computational time required for Bayesian parameter estimation by exploiting Gaussian process predictions with a fixed GP error and a varying GP error.

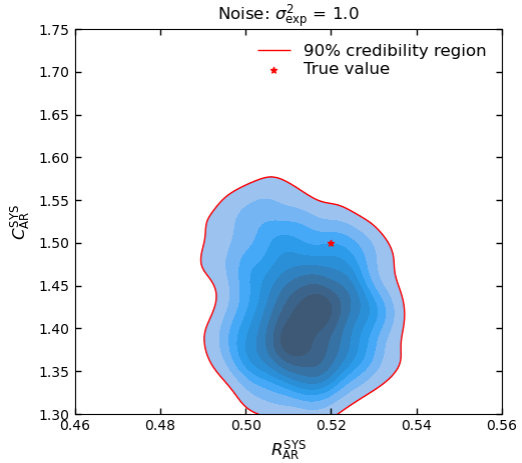
All the numerical tests seen so far assume that the Gaussian process is already trained. The training process is performed taking into account what we observed in Section 2.6.4; indeed, Figure 2.13 showed that reliable predictions are obtained whatever the size of the training set was. However, we decided to perform a training procedure with 500 training samples in all the numerical tests to obtain an optimal balance between prediction performance and computational effort.
Now we compare the results obtained from the Bayesian parameter estimation with a Gaussian process based approach considering different sizes of the training set. In particular, we test if it is possible to obtain a suitable posterior distribution on the parameters pair $(R_{\text{AR}}^{\text{SYS}}, C_{\text{AR}}^{\text{SYS}})$ even if a Gaussian process trained in a low data regime is considered, namely with 60 training samples.
Figure 4.5 shows these results with respect to those in Figure 4.1 with measurement noise $\sigma_{\text{exp}}^2 = \{1.0, 0.1\}$ mmHg^2. Notice that, for $\sigma_{\text{exp}}^2 = 1.0$ mmHg^2, the GP trained in a low data regime is able to provide an output which is comparable with that of the GP obtained with 500 training samples. Some issues arise when the measurement noise gets smaller, as shown in Figure 4.5d. Indeed, we get that the credibility region does not contain the exact value of parameters. This is due to the fact that a GP trained in a low data regime is much less sensitive to these changes and, as a consequence, an inaccurate result is obtained, as shown in Table 4.4.
However, we have a further decrease of the computational time required for the Bayesian parameter estimation when such a GPs trained in a low data regime are used, as shown in Table 4.5. This allows us to conclude that a GP obtained with a small number of training samples can be efficiently used when high measurement errors are assumed, but, conversely, it provides a less accurate Bayesian parameter estimation for lower values of σ_{exp}^2.



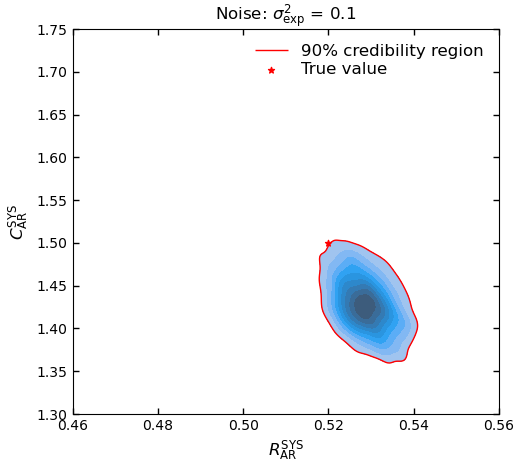
(a) GP with 500 training samples for $\sigma^2_{\text{exp}} = 1.0$ mmHg^2



(b) GP with 500 training samples for $\sigma^2_{\text{exp}} = 0.1$ mmHg^2



(c) GP with 60 training samples for $\sigma^2_{\text{exp}} = 1.0$ mmHg^2



(d) GP with 60 training samples for $\sigma^2_{\text{exp}} = 0.1$ mmHg^2

Figure 4.5: Posterior distributions on the parameters pair $(R^{\text{SYS}}_{\text{AR}}, C^{\text{SYS}}_{\text{AR}})$ (prescribed values $R^{\text{SYS}}_{\text{AR}} = 0.52$ mmHg s mL^{-1} and $C^{\text{SYS}}_{\text{AR}} = 1.5$ mL mmHg^{-1}) computed by means of the MCMC method with a Gaussian process trained with 500 training samples (upper panels) and considering a Gaussian process trained in a low data regime (lower panels) for $\sigma^2_{\text{exp}} = \{1.0, 0.1\}$ mmHg^2.

		GP with 500 training samples		GP with 60 training samples	
Parameter	**Indicator**	$\boldsymbol{\sigma^2_{\text{exp}} = 1.0}$	$\boldsymbol{\sigma^2_{\text{exp}} = 0.1}$	$\boldsymbol{\sigma^2_{\text{exp}} = 1.0}$	$\boldsymbol{\sigma^2_{\text{exp}} = 0.1}$
$R^{\text{SYS}}_{\text{AR}}$	Mean	0.5160	0.5216	0.5132	0.5285
	MAP	0.5171	0.5224	0.5144	0.5277
$C^{\text{SYS}}_{\text{AR}}$	Mean	1.4851	1.4977	1.4263	1.4302
	MAP	1.4515	1.4995	1.4130	1.4258

Table 4.4: Mean and MAP indicators of the posterior distribution on the parameters pair $(R^{\text{SYS}}_{\text{AR}}, C^{\text{SYS}}_{\text{AR}})$ (prescribed values $R^{\text{SYS}}_{\text{AR}} = 0.52$ mmHg s mL^{-1} and $C^{\text{SYS}}_{\text{AR}} = 1.5$ mL mmHg^{-1}) computed by means of the MCMC method with a Gaussian process trained with 500 training samples and considering a Gaussian process trained in a low data regime for $\sigma^2_{\text{exp}} = \{1.0, 0.1\}$ mmHg^2.

GP with 500 training samples	**GP with 60 training samples**
$\sim$ 1153 s	$\sim$ 412 s

Table 4.5: Computational time required for Bayesian parameter estimation by exploiting Gaussian process predictions with a Gaussian process trained with 500 training samples and a Gaussian process trained in a low data regime.

5 | Conclusions and future developments

In this work, we aimed to point out the properties and performances of Gaussian processes, highlighting the role of covariance functions.
In Chapter 2 we tested the accuracy/efficiency balance of Gaussian processes and proposed some diagnostic methods to prove their reliability in simple benchmark problems and for a more complex circulation model. We observed that Gaussian processes are a powerful tool in making predictions also thanks to the fact that they are easily interpretable and not particularly computationally demanding.
This last feature makes them computationally cheap surrogate models for problems in the context of uncertainty quantification (see Chapter 3). Indeed, we successfully employed Gaussian processes in sensitivity analysis to rank the most influential input parameters without accounting for the high-fidelity circulation model.
In Chapter 4 we also proved that Gaussian processes are able to provide reliable results when used in place of the high-fidelity circulation model in solving the inverse problem for parameters estimation by means of the Markov chain Monte Carlo method.
In general, we can conclude that Gaussian processes provide accurate estimates in sensitivity analysis and in Bayesian parameter estimation reducing the computational burden that would be otherwise too high if the same analyses were carried out by means of the high-fidelity circulation model.
Some issues may arise when the number of varying input parameters is large; indeed, we proved that the complexity of the problem increases as the number of parameters grows and a worse Gaussian process posterior predictive distribution is obtained if not suitably trained. This reflects in the computation of the first order Sobol indices, as shown in Section 3.2. However, this problem is successfully overcome by increasing the number of training values, even if this leads to pay something in terms of computational cost. Moreover, this is not always possible in real-life problems; indeed, it may happen that a sufficiently large amount of data is not available for practical difficulties.
For what concerns possible future developments, we propose a list of interesting sugges-

tions:

- analysis of how Gaussian processes behave for different numerical models, testing their reliability as the complexity of the model increases;
- modification of the Sobol indices computation by taking into account the uncertainty of Gaussian process predictions;
- parallelization of the procedures for the sensitivity analysis and the Markov chain Monte Carlo method;
- employment of Gaussian process based methods in multifidelity approaches for combining information between different fidelity models.

Bibliography

[1] Y. Al-Taweel. Diagnostics and Simulation-Based Methods for Validating Gaussian Process Emulators. Master's thesis, School of Mathematics and Statistics, 2018.

[2] L. S. Bastos and A. O'Hagan. Diagnostics for Gaussian Process Emulators. *Technometrics*, pages 1–24, 2009.

[3] J. G. Betts. *Anatomy and Physiology*. OpenStax College, 1 edition, 2013.

[4] P. J. Blanco and R. A. Feijòo. A 3D-1D-0D Computational Model for the Entire Cardiovascular System. *Computational Mechanics 24*, pages 5887–5911, 2010.

[5] S. P. Brooks. Markov chain Monte Carlo method and its application. *The Statistician*, pages 69–100, 1998.

[6] G. Casella and R. L. Berger. *Statistical Inference*. McGrawHill, 2 edition, 2002.

[7] M. Corti. Effects of Atrial Fibrillation on Left Atrium Haemodynamics: A Patient Specific Computational Fluid Dynamics Study. Master's thesis, School of Industrial and Information Engineering, 2021.

[8] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi. Gaussian Process Regression for Materials and Molecules. *Chemical Reviews*, pages 10073–10141, 2021.

[9] D. Duvenaud. The Kernel Cookbook: Advice on Covariance functions, 2014. URL `https://www.cs.toronto.edu/~duvenaud/cookbook/`.

[10] M. Hirschvogel, M. Bassilious, L. Jagschies, S. M. Wildhirt, and M. W. Gee. A monolithic 3D-0D coupled closed-loop model of the heart and the vascular system: Experiment-based parameter estimation for patient-specific cardiac mechanics. *International Journal for Numerical Methods in Biomedical Engineering*, pages 1–22, 2017.

[11] T. Homma and A. Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering and System Safety*, pages 1–17, 1996.

[12] F. Liang and H. Liu. A Closed-Loop Lumped Parameter Computational Model for Human Cardiovascular System. *JSME International Journal*, pages 484–493, 2005.

[13] E. F. Mameo. Applications of Gaussian Processes in quantitative finance. Master's thesis, School of Industrial and Information Engineering, 2021.

[14] T. Minka. Bayesian Linear Regression, 2000. URL `https://www.microsoft.com/en-us/research/publication/bayesian-linear-regression/`.

[15] G. R. Mirams, P. Pathmanathan, R. A. Gray, P. Challenor, and R. H. Clayton. Uncertainty and variability in computational and mathematical models of cardiac physiology. *The Journal of Physiology*, pages 6833–6847, 2016.

[16] H. Mohammadi, P. Challenor, and C. Prieur. Variance-based global sensitivity analysis of numerical models using R, 2022. URL `https://arxiv.org/pdf/2206.11348.pdf`.

[17] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer New York, 1 edition, 1996.

[18] S. Pagani. *Reduced-Order Models for Inverse Problems and Uncertainty Quantification in Cardiac Electrophysiology*. PhD thesis, School of Industrial and Information Engineering, 2017.

[19] R. Ranjan, B. Huang, and A. Fatehi. Robust Gaussian process modeling using EM algorithm. *Journal of Process Control*, pages 125–136, 2016.

[20] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[21] F. Regazzoni, M. Salvador, P. C. Africa, M. Fedele, L. Dedè, and A. Quarteroni. A cardiac electromechanical model coupled with a lumped-parameter model for closed-loop blood circulation. Part I: model derivation. *Mox Reports*, pages 1–23, 2020.

[22] F. Regazzoni, M. Salvador, P. C. Africa, M. Fedele, L. Dedè, and A. Quarteroni. A cardiac electromechanical model coupled with a lumped-parameter model for closed-loop blood circulation. Part II: numerical approximation. *Mox Reports*, pages 1–35, 2020.

[23] F. Regazzoni, C. Vergara, L. Dedè, P. Zunino, M. Guglielmo, R. Scrofani, L. Fusini, C. Cogliati, G. Pontone, and A. Quarteroni. Modeling the effect of COVID-19 disease on the cardiac function: a computational study. *MOX Reports*, pages 1–17, 2020.

[24] F. Regazzoni, M. Salvador, L. Dedè, and A. Quarteroni. A machine learning method

for real-time numerical simulations of cardiac electromechanics. *Computer methods in applied mechanics and engineering*, pages 1–26, 2022.

[25] E. Schulz, M. Speekenbrink, and A. Krause. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, pages 1–16, 2018.

[26] D. Shier, J. Butler, and R. Lewis. *Hole's Human Anatomy and Physiology.* McGrawHill, 12 edition, 2010.

[27] P. Sollich. Bayesian methods for support vector machines: evidence and predictive class probabilities. *Machine Learning 46*, pages 21–52, 2002.

[28] N. Ulapane, K. Thiyagarajan, and S. Kodagoda. Hyper-Parameter Initialization for Squared Exponential Kernel-based Gaussian Process Regression, 2020. URL `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9248120`.

[29] F. Vivarelli and C. K. I. Williams. Discovering hidden features with Gaussian processes regression. *Advances in Neural Information Processing Systems*, pages 613–619, 1999.

[30] A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian Process Regression Networks, 2011. URL `https://arxiv.org/pdf/1110.4411.pdf`.

[31] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum. Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 66–76, 2007.

A | Appendix A

In this appendix we report the proof of the law of total variance used in (3.11) and a preliminary result useful for its derivation, that is the law of total expectation [6].

Proposition A.1 (Law of total expectation)**.** *If X is a random variable whose expected value $\mathbb{E}[X]$ is defined and Y is any random variable on the same probability space, then*

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|Y]], \tag{A.1}$$

i.e. the expected value of the conditional expected value of X given Y is the same as the expected value of X.

Proof. Let $(\Omega, \mathcal{F}, P)$ be a probability space on which two sub σ-algebras $\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \mathcal{F}$ are defined. For a random variable X on such a space, the smoothing law states that if $\mathbb{E}[X]$ is defined, then

$$\mathbb{E}[\mathbb{E}[X|\mathcal{G}_2]|\mathcal{G}_1] = \mathbb{E}[X|\mathcal{G}_1]. \tag{A.2}$$

Indeed, since a conditional expectation is a Radon-Nikodym derivative, verifying the following two properties establishes the smoothing law:

1. $\mathbb{E}[\mathbb{E}[X|\mathcal{G}_2]|\mathcal{G}_1]$ is $\mathcal{G}_1$-measurable;
2. $\int_{G_1} \mathbb{E}[\mathbb{E}[X|\mathcal{G}_2]|\mathcal{G}_1]dP = \int_{G_1} X dP$, for all $G_1 \in \mathcal{G}_1$.

The first of these properties holds by definition of the conditional expectation. In order to prove the second one, we can write

$$\begin{aligned} \min\left(\int_{G_1} X_+ dP, \int_{G_1} X_- dP\right) &\leq \min\left(\int_{\Omega} X_+ dP, \int_{\Omega} X_- dP\right) \\ &= \min(\mathbb{E}[X_+], \mathbb{E}[X_-]) < \infty \end{aligned} \tag{A.3}$$

because $\mathbb{E}[X]$ is defined by assumption. Therefore, $\int_{G_1} X dP$ is defined and the second

property holds since $G_1 \in \mathcal{G}_1 \subseteq \mathcal{G}_2$ implies

$$\int_{G_1} \mathbb{E}[\mathbb{E}[X|\mathcal{G}_2]|\mathcal{G}_1]dP = \int_{G_1} \mathbb{E}[X|\mathcal{G}_2]dP = \int_{G_1} X dP. \tag{A.4}$$

In the special case when $\mathcal{G}_1 = \{\emptyset, \Omega\}$ and $\mathcal{G}_2 = \sigma(Y)$, the smoothing law reduces to

$$\mathbb{E}[\mathbb{E}[X|Y]] = \mathbb{E}[X]. \tag{A.5}$$

□

Proposition A.2 (Law of total variance)**.** *If X and Y are random variables on the same probability space and the variance of Y is finite, then*

$$\mathbb{V}(Y) = \mathbb{E}[\mathbb{V}(Y|X)] + \mathbb{V}(\mathbb{E}[Y|X]). \tag{A.6}$$

Proof. The law of total variance can be proved by using the law of total expectation. First of all, we recall that

$$\mathbb{V}(Y) = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 \tag{A.7}$$

from the definition of variance. Again, from the definition of variance and applying the law of total expectation, we have

$$\mathbb{E}[Y^2] = \mathbb{E}[\mathbb{E}[Y^2|X]] = \mathbb{E}[\mathbb{V}(Y|X) + \mathbb{E}[Y|X]^2]. \tag{A.8}$$

Now we rewrite the conditional second moment of Y in terms of its variance and first moment and apply the law of total expectation on the right hand side:

$$\begin{aligned} \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 &= \mathbb{E}[\mathbb{V}(Y|X) + \mathbb{E}[Y|X]^2] - \mathbb{E}[\mathbb{E}[Y|X]]^2 \\ &= \mathbb{E}[\mathbb{V}(Y|X)] + (\mathbb{E}[\mathbb{E}[Y|X]^2] - \mathbb{E}[\mathbb{E}[Y|X]]^2). \end{aligned} \tag{A.9}$$

Finally, we can recognize the terms in the parentheses as the variance of the conditional expectation $\mathbb{E}[Y|X]$:

$$\mathbb{V}(Y) = \mathbb{E}[\mathbb{V}(Y|X)] + \mathbb{V}(\mathbb{E}[Y|X]). \tag{A.10}$$

□

List of Figures

List of Tables