UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

MASTER'S THESIS IN INDUSTRIAL ENGINEERING

VARIANCE-BASED SENSITIVITY ANALYSIS IN VEHICLE DYNAMICS SIMULATION: DEVELOPMENT AND APPLICATION OF A POLYNOMIAL CHAOS EXPANSION METHOD

AUTHOR: ALEJANDRO VALERO ANDREU

SUPERVISOR: FRANCISCO JOSÉ VALERO CHULIÁ

Academic year: 2015-16

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

RESUMEN

Durante las primeras fases del proceso de desarrollo de nuevos productos, muchos de los parámetros del producto desarrollado pueden cambiar y están por tanto sujetos a algún tipo de incertidumbre. En particular, en la industria automotriz, estas incertidumbres en los parámetros afectarán por último al comportamiento del vehículo que está siendo desarrollado. Si se tiene como objetivo un comportamiento dinámico concreto, resulta de suma importancia conocer que parámetro específico del vehículo tiene mayor importancia en determinar la incertidumbre de la salida de interés.

Los modelos para la simulación del comportamiento dinámico de vehículos son conocidos desde hace ya mucho tiempo y han sido utilizados con éxito para simular el comportamiento de un vehículo bajo ciertas maniobras, cuando los parámetros del mismo son conocidos. Sin embargo, incluso cuando algunos parámetros son inciertos, normalmente se pueden aceptar ciertos supuestos razonables con respecto a su varianza, basados en la experiencia o en productos anteriores similares, por ejemplo. Consecuentemente, la propagación de la incertidumbre de los parámetros a las variables de salida y con ello el comportamiento estocástico del modelo quedan, en principio, determinados.

En este trabajo se investigarán medidas de sensibilidad basadas en la varianza y métodos para el análisis global de sensibilidad. En particular, se aplicará a la dinámica vehicular el método de expansión en caos polinomial (con muchas aplicaciones en otros campos de la ciencia y la ingeniería) y el cálculo de los índices de sensibilidad de Sobol basados en dicha expansión.

Palabras Clave: Dinámica de Vehículos, Simulación, Métodos Númericos, Modelos Estocásticos, Cuantificación de la Incertidumbre, Caos Polinomial, Análisis de Sensibilidad, Medidas de Sensibilidad basadas en la Varianza, Índices de Sobol

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

ABSTRACT

During the early stages of any new product development, many parameters of the product being developed will change and are therefore subject to uncertainty. In particular, in the automotive industry, these uncertainties in the parameters will ultimately affect the behavior of the vehicle being developed. If particular vehicle dynamics are targeted, knowing which specific input parameter of the vehicle is more important in determining the uncertainty in the output of interest is of utmost importance.

Vehicle dynamics models have been known and used for a long time and they can be very useful for simulating the behavior of the vehicle under certain maneuvers if its parameters are known. Nonetheless, even when some parameters are uncertain, reasonable assumptions can usually be accepted regarding their variance, based on experience or previous similar products, for example. Consequently, the propagation of uncertainty from the parameters to the output variables and thus the stochastic behavior of the model can be determined in principle.

In this project, variance-based sensitivity measures and methods for global Sensitivity Analysis will be investigated. In particular, the Polynomial Chaos Expansion method (with many applications in other fields of science and engineering) and the computation of Sobol' sensitivity indices based on polynomial chaos expansions will be applied to the simulation of vehicle dynamics.

Keywords: Vehicle Dynamics, Simulation, Numerical Methods, Stochastic Models, Uncertainty Quantification, Polynomial Chaos, Sensitivity Analysis, Variance-based Sensitivity Measures, Sobol' Indices

Contents

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Master's Thesis in Industrial Engineering

Alejandro Valero Andreu

Variance-based sensitivity analysis in vehicle dynamics simulation: development and application of a polynomial chaos expansion method

REPORT

Contents of the report

1 INTRODUCTION

The design of vehicle dynamics depends on several factors which will change over the course of the development process of the vehicle, until the final product is ready for production. These changes ultimately affect the behavior of the vehicle and thus it is of utmost importance to be able to quantify their effects on the output signals of interest. In this regard, simulation models offer a valuable solution since they provide a means of replicating the behavior of a vehicle under certain maneuvers in a reproducible fashion. Consequently, different sets of uncertain parameters can be applied to simulation models in order to investigate the resulting uncertainty in any of the relevant output signals.

Uncertainties may arise from different sources. For example, the mathematical model may not be able to describe the underlying physics accurately, this is known as structural uncertainty or model inadequacy. Furthermore, the model parameters may not be exactly inferable or determinable, also known as parameter uncertainty. In addition, if the input parameters are determined from measurements, there is always an associated experimental uncertainty. All these sources of uncertainty have an impact on the results of the simulations and consequently the output signals are no longer single values but random variables with their own probability distributions. Therefore uncertainties propagate from the input parameters to the output variables and quantifying this propagation may be further investigated by means of a global Sensitivity Analysis (SA).

Nowadays there are several approaches to deal with these problems. One of the first and best known techniques is the Monte Carlo Simulation (MCS) Method (Metropolis & Ulam, 1949), whose use is indeed fairly restricted due to its slow rate of convergence, which causes long computing times. Nevertheless, alternatives to this method have been proposed and developed, for example using Fuzzy-Set Theory (Hanss, 2002), the Method of Dimensionality Reduction (Rahman & Xu, 2004; Xu & Rahman, 2004) or the modified Monte Carlo Method with Stratified Sampling (Shields, Teferra, Hapij, & Daddazio, 2013). In this work a method for the quantification of output uncertainties caused by uncertainties in the input parameters will be presented and used – namely the Polynomial Chaos Expansion (PCE) – and variance-based measures for SA will be computed by means of the so-called Sobol' indices.

Based on the theory of Homogeneous Chaos formulated by Wiener (1938) and subsequently enhanced by Cameron and Martin (1947), Ghanem and Spanos (1991) along with Xiu and Karniadakis (2002), PCE is by now a widely spread technique according to Xiu (2010, p. 4), with many applications in different fields of science, such as Fluid Dynamics (Le Maitre & Knio, 2010), Finite Element Methods (Ghanem & Spanos, 1991) or Failure Detection (Paffrath & Wever, 2007), among others. The fundamental idea of PCE is the approximation of a model's output by an expansion in an orthogonal polynomial basis, whose appropriate selection results in exponential convergence of the approximation.

Regarding global SA, a number of methods are available depending on the different constraints of the problem being investigated, such as linearity of the model output with respect to its inputs, the number of outputs, the presence of correlated inputs and/or interactions

between them. In particular, variance-based methods are suitable whenever non-linearity, non-monotonic behavior or both are present in the model response (see Andrea Saltelli, Tarantola, Campolongo, & Ratto, 2004, p. 188).

The main goal of this work is to apply the PCE Method to the simulation of vehicle dynamics and to derive Sobol' indices for their use in global SA from the coefficients of the PCE. Since, as mentioned before, most of the parameters affecting vehicle dynamics change throughout the development process, it is of maximum importance to be able to quantify the changes in the behavior of the system for a certain bandwidth of the input parameters and thus being able to answer questions such as: "*which of the uncertain input factors is more important in determining the uncertainty in the output of interest?*" or "*if we could eliminate the uncertainty in one of the input factors, which factor should we choose to reduce the most the variance of the output?*" (see Andrea Saltelli et al., 2004, p. ix), which may be relevant for the development process of the vehicle.

To clarify the methods and techniques used in this work, they will be first applied to some test functions that are commonly used in the field of Uncertainty Quantification (UQ) and next to a simple Single-Track Model (STM) before they are finally put to use in a more complex environment with a non-linear Two-Track Model (TTM). The present work is therefore structured following this arrangement. In Chapter 2 the theoretical framework for the PCE and the global variance-based SA will be established. Chapter 3 will present a comparison of different non-intrusive methods that can be used for the computation of the PCE coefficients, paying special attention to differences in their performance. Additionally, the method for computing Sobol' indices from the PCE coefficients will also be presented in this chapter. The results of applying the aforementioned methods to appropriate test functions, the STM and the TTM will be presented in Chapter 4. Finally, Chapter 5 will encompass a brief summary of this work, the conclusions and some recommendations for future work.

2 THEORETICAL FRAMEWORK

This chapter presents a summary of the basic theoretical knowledge required in order to understand the methods and problems described within the present work, starting with the foundations of Polynomial Chaos (PC) and following with a brief introduction to Sensitivity Analysis (SA).

2.1 Polynomial Chaos

The foundations for PC were first established by Norbert Wiener in 1938 with his work in Homogeneous Chaos (Wiener, 1938), where he showed that any normally distributed random variable could be expressed as an infinite sum of Hermite polynomials of different orders weighted by some coefficients, thus enabling stochastic processes to be modeled accordingly. The application of Homogeneous Chaos to arbitrary L^2-functions dates back to 1947 and is attributed to Cameron and Martin (Cameron & Martin, 1947), while the extension of PC to common distributions other than Gaussian was carried out many years later by Xiu and Karniadakis (Xiu & Karniadakis, 2002). In this section, the foundations of PC will be explained beginning with the most fundamental ideas, i.e. orthogonal polynomials.

2.1.1 *Orthogonal polynomials*

Orthogonal polynomials are of utmost importance for the theory of PC, since they are used as the basis for the expansion of the random variable of interest.

In particular, a system of polynomials $\{\Psi_n(x), n \in \mathbb{N}_0\}$ of n-th degree with:

$$\Psi_n(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_1 x + b_0, \qquad b_n \neq 0, \qquad x \in \mathbb{R} \tag{2.1}$$

is orthogonal, as stated in Schoutens (2000, p. 3), in terms of a positive measure $\alpha(x) \in \mathbb{R}_0^+$ with respect to the sample space Ω, if the property of orthogonality in terms of the inner product:

$$\langle \Psi_n(x), \Psi_m(x) \rangle = \mathbb{E}[\Psi_n(x)\Psi_m(x)] = \int_\Omega \Psi_n(x)\Psi_m(x)\mathrm{d}\alpha(x) = \gamma_n \delta_{nm}, \qquad n, m \in \mathbb{N}_0 \tag{2.2}$$

is fulfilled. $\alpha(x)$ usually represents a probability density function $\rho(x)$, γ_n is a non-zero value, also known as normalizing constant and δ_{nm} is Kronecker's delta, defined as:

$$\delta_{nm} = \begin{cases} 1, & \text{if } n = m \\ 0, & \text{if } n \neq m \end{cases}$$

That is, if $n \neq m$ then:

$$\langle \Psi_n, \Psi_m \rangle = 0$$

Consequently, for continuous probability distributions one can use the following expression:

$$\int_{\Omega} \Psi_n(x)\Psi_m(x)\rho(x)\mathrm{d}x = \gamma_n\delta_{nm} \tag{2.3}$$

On the other hand, if the probability distribution is discrete, Schoutens (2000, p. 3) defines equation (2.3) for nodes x_j with weighting function $\rho(x_j)$ as:

$$\sum_j \Psi_n(x_j)\Psi_m(x_j)\rho(x_j) = \gamma_n\delta_{nm}, \qquad j \in \mathbb{N}_0 \tag{2.4}$$

In addition, if $\gamma_n = 1$ the system is said to be orthonormal.

For further details on orthogonal polynomials the reader is referred to specific literature on the subject, for example Szegö (1939); Chihara (1978) or Xiu (2010).

Following the theory of discrete chaos as exposed by Wiener in his article (Wiener, 1938), stochastic processes with Gaussian random variables may be modeled by an expansion in terms of Hermite polynomials and, in fact, the PCE derived thereof converges exponentially. Xiu and Karniadakis (2002) generalized the results of the Cameron-Martin theorem (Cameron & Martin, 1947) to other continuous and discrete distributions by using appropriate orthogonal polynomials from the so-called Askey-Scheme (Askey & Wilson, 1985), thus giving rise to generalized Polynomial Chaos (gPC), which will be presented in what follows.

2.1.2 *Generalized polynomial chaos expansion*

Next, the basic theory of gPC will be clarified. gPC is based on the expansion of stochastic processes or random variables by means of orthogonal polynomial bases.

The foundation of a stochastic mathematical model is the existence of a probability space or triple $(\Omega, \mathcal{F}, \mathcal{P})$ where Ω represents the sample space, $\mathcal{F} \subset 2^{\Omega}$ is a set of events (a σ-algebra on Ω) and $\mathcal{P}$ is a probability measure. Moreover, convergence is satisfied in a mean-squared sense for a stochastic process of finite second-order moments by virtue of the Cameron-Martin theorem (Cameron & Martin, 1947), implying L^2 convergence in the Hilbert space defined as:

$$L^2 = L^2(\Omega, \mathcal{F}, \mathcal{P}) = \{X : \mathbb{E}[X^2] < \infty\} \tag{2.5}$$

with the norm $\|X\| = \sqrt{\langle X, X\rangle}$ induced by the inner product $\langle X, Y\rangle = \mathbb{E}[XY]$ (see Xiu, 2010, p. 23), with the (weighted) inner product being defined by the integrals in equation (2.2).

If the weighting function of the orthogonal polynomials being used has the same form (except for a constant factor) as the weighting function of the distribution of the random variable the convergence rate for the corresponding PCE is optimal (see Xiu & Karniadakis, 2002). Table 2.1 shows some common continuous and discrete probability distributions and their corresponding gPC basis polynomials from the Askey-Scheme as first investigated by Xiu and Karniadakis (2002) and as presented in Xiu (2010, p. 59).

Following, gPC will be further detailed by means of the more general multivariate case, for a description of the univariate case the reader is referred to the bibliography on the subject, for example Xiu (2010, pp. 57–63) or Ghanem and Spanos (1991, p. 47 ff.)

	Distribution of the random variable z_i	gPC basis polynomial Ψ_i	Support
Continuous	Gaussian	Hermite	$(-\infty, \infty)$
	Gamma	Laguerre	$[0, \infty)$
	Beta	Jacobi	$[a, b]$
	Uniform	Legendre	$[a, b]$
Discrete	Poisson	Charlier	{0, 1, 2, ...}
	Binomial	Krawtchouk	{0, 1, ..., N}
	Negative binomial	Meixner	{0, 1, 2, ...}
	Hypergeometric	Hahn	{0, 1, ..., N}

Table 2.1: Correspondence between the gPC polynomial basis and their underlying random variables, Askey-Scheme (Xiu, 2010, p. 59)

Multivariate gPC

When more than one independent random variables are present in the stochastic models being used, multivariate gPC must be used.

As detailed in Xiu (2010, p. 64 ff.), let $Z = (Z_1, \dots, Z_d)$ be a random vector with dimension d, whose components are mutually independent with joint Cumulative Distribution Function (CDF) $F_Z(z_1, \dots, z_d) = P(Z_1 \leq z_1, \dots,)$ and marginal CDF $F_{Z_i}(z_i) = P(Z_i \leq z_i)$ for Z_i with $i = 1 \dots d$ and support I_{Z_i}. Mutual independence of every Z_i means that $F_Z(z) = \prod_{i=1}^{d} F_{Z_i}(z_i)$ and $I_z = I_{Z_1} \times \cdots \times I_{Z_d}$. Moreover, let $\{\Psi_k(Z_i)\}_{k=0}^{N} \in \mathbb{P}_N(Z_i)$ be the univariate gPC basis functions in Z_i of degree up to N, so that:

$$\mathbb{E}[\Psi_n(Z_i)\Psi_m(Z_i)] = \int \Psi_n(z)\Psi_m(z)\mathrm{d}F_{Z_i}(z) = \gamma_n \delta_{nm}, \qquad 0 \leq n, m \leq N \tag{2.6}$$

Then, in order to define the d-variate Nth-degree gPC basis functions a multi-index is needed, which is defined as follows:

$$\boldsymbol{i} = (i_1, \dots, i_d) \in \mathbb{N}_0^d, \qquad \text{with } |\boldsymbol{i}| = i_1 + \cdots + i_d \tag{2.7}$$

Thus, the d-variate basis functions can now be defined in terms of products of univariate gPC polynomials of total degree less than or equal to N:

$$\boldsymbol{\Psi_i} = \Psi_{i_1}(Z_1) \dots \Psi_{i_d}(Z_d) = \prod_{k=1}^{d} \Psi_{i_k}(Z_k), \qquad 0 \leq |\boldsymbol{i}| \leq N \tag{2.8}$$

From 2.6 it follows that:

$$\mathbb{E}[\boldsymbol{\Psi_i}(Z)\boldsymbol{\Psi_j}(Z)] = \int \boldsymbol{\Psi_i}(z)\boldsymbol{\Psi_i}(z)\mathrm{d}F_Z(z) = \gamma_i \delta_{ij} \tag{2.9}$$

where $\gamma_i = \mathbb{E}[\boldsymbol{\Psi_i}^2] = \gamma_{i_1} \dots \gamma_{i_d}$ are the normalizing constants and $\delta_{ij} = \delta_{i_1 j_1} \dots \delta_{i_d j_d}$ is the d-variate Kronecker delta function.

According to Sudret (2008) and Xiu (2010) if a stochastic model is represented by $Y = f(Z) \in L^2_{\mathrm{d}F_Z}(I_Z)$, with $L^2_{\mathrm{d}F_Z}(I_Z)$ being the space of all mean-square integrable functions

with respect to the measure $\mathrm{d}F_Z$, then the stochastic model may be approximated by its Nth-degree gPC as:

$$Y = f(Z) \approx \sum_{|\boldsymbol{i}| \leq N} y_{\boldsymbol{i}} \boldsymbol{\Psi}_{\boldsymbol{i}}(Z), \qquad Z = (Z_1, \ldots, Z_d) \tag{2.10}$$

Note that although convenient for the formulation, the multi-index notation can be cumbersome to manipulate and thus a single index (k) is used in practice. A common choice for the computation of this single index being the graded lexicographic order, where $\boldsymbol{i} > \boldsymbol{j}$ if and only if $|i| \geq |j|$ and the left-most non-zero entry of the difference, $\boldsymbol{i} - \boldsymbol{j}$, is positive. The multi-index can now be ordered in an ascending order following a single index. An example for a 4-dimensional case is shown in table 2.2 (as presented in Xiu (2010, p. 66) and corrected according to the definition of graded lexicographic ordering).

$\lvert\boldsymbol{i}\rvert$	Multi-index $\boldsymbol{i}$	Single index k
0	(0 0 0 0)	0
1	(0 0 0 1)	1
	(0 0 1 0)	2
	(0 1 0 0)	3
	(1 0 0 0)	4
2	(0 0 0 2)	5
	(0 0 1 1)	6
	(0 1 0 1)	7
	(1 0 0 1)	8
	(0 0 2 0)	9
	(0 1 1 0)	10
	(1 0 1 0)	11
	(0 2 0 0)	12
	(1 1 0 0)	13
	(2 0 0 0)	14
3	(0 0 0 3)	15
	(0 0 1 2)	16
	(0 1 0 2)	17
	...	...

Table 2.2: Example of graded lexicographic ordering of a multi-index $\boldsymbol{i}$ in 4 dimensions, adapted from Xiu (2010, p. 66)

Therefore, the gPC approximation of the stochastic model Y can be now expressed in terms of the single index k as follows:

$$Y = f(Z) \approx \sum_{k=0}^{P-1} y_k \Psi_k(Z) \tag{2.11}$$

with the number of terms in the sum (also the number of unknown (vector) coefficients y_k) being:

$$P = \begin{pmatrix} N+d \\ N \end{pmatrix} = \frac{(N+d)!}{N!d!} \tag{2.12}$$

The coefficients $y_k = y_k(t)$ are unknown but deterministic, and they are in general time-dependent (Xiu, 2010, p. 67). They are usually referred to as the spectral coefficients of the gPC expansion, unknown coefficients, response coefficients or simply expansion coefficients. Different methods for the computation of these spectral coefficients are available and will be presented later in this work, in Chapter 3.1.

Additionally, note that the number of terms of the PCE to be computed depends on the dimension d and the order N being used (as given by equation (2.12)).

Given the PCE of a random variable, important properties and statistics thereof can be easily computed, for instance statistical moments, as presented in the following section.

2.1.3 *Computation of statistical moments*

In general, the k-th moment of a random variable $Z \in L^n(\mathcal{P})$ with $n \in \mathbb{N}$ can be defined as:

$$m_k = \mathbb{E}[Z^k] = \int_\Omega Z^k \mathrm{d}\mathcal{P} \tag{2.13}$$

(See Klenke, 2014, p. 101).

In particular, given the gPC approximation, the expectation and the variance of the output Y are easily computed by using the orthogonality property (equation (2.2)) and the fact that $\Psi_0 = 1$, as presented in Augustin (2012, p. 11 ff.); Berveiller (2005, p. 57) and Sudret (2008, p. 971):

$$\begin{aligned} \mathbb{E}[Y] &= m_1 \\ &= \int_\Omega \left(\sum_{k=0}^{P-1} y_k \Psi_k \right) \Psi_0 \mathrm{d}\mathcal{P} \\ &= \langle y_0 \Psi_0, \Psi_0 \rangle \\ &= y_0 \end{aligned} \tag{2.14}$$

That is, the expectation of Y can be computed as the first term of its PCE.

$$\begin{aligned} \mathrm{Var}[S] &= m_2 - m_1^2 \\ &= \int_\Omega \left(\sum_{k=0}^{P-1} y_k \Psi_k \right) \left(\sum_{j=0}^{P-1} y_j \Psi_j \right) \mathrm{d}\mathcal{P} - y_0^2 \\ &= \sum_{k=1}^{P-1} y_k^2 \mathbb{E}[\Psi_k^2] \end{aligned} \tag{2.15}$$

Moments of higher order can be computed analogously, although the resulting expressions are cumbersome and the reader is referred to the aforementioned literature (Augustin (2012) and Berveiller (2005)) for further details.

From these expressions one may observe that in order to compute the statistics of the model $Y = f(Z)$ it is not necessary to compute its full response but only the coefficients of its gPC expansion are required. Furthermore, one can notice that it is also possible to approximate

the statistics of the response Y by randomly sampling its gPC expansion and estimating the desired statistics from the resulting histogram.

2.1.4 *Probability density function estimation*

Besides statistical moments, it is also useful to have some information about the Probability Density Functions (PDFs) of the output variables of interest. Nonetheless, obtaining the PDF from the roots of the polynomials is not a simple task and an analytical solution requires considerable effort, specially with increasing stochastic dimension and, in fact, a general expression does not exist, according to B. J. Debusschere et al. (2004). Therefore approximations to the actual PDF are usually computed instead, for instance the Kernel Density Estimation (KDE) (Parzen, 1962).

As defined in Härdle and Simar (2007, p. 13) the general Kernel Density Estimator has the following form:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{2.16}$$

where n is the number of samples used, h is the so-called bandwidth and $K(u)$ is the kernel. The bandwidth h determines the degree of smoothness of the estimate $\hat{f}_h$ (Härdle & Simar, 2007, p. 16) and can be optimized (for example, by cross-validation), while different kernels generate different shapes of the estimated PDF. Table 2.3 lists some commonly used kernels as presented in Härdle and Simar (2007, p. 14), where $\boldsymbol{I}(M)$ is the indicator function for a set M, which is 1 on M and 0 otherwise.

Name	Kernel $K(u)$
Uniform	$\frac{1}{2}\boldsymbol{I}(\lvert u\rvert \leq 1)$
Triangle	$(1 - \lvert u\rvert)\boldsymbol{I}(\lvert u\rvert \leq 1)$
Epanechnikov	$\frac{3}{4}(1 - u^2)\boldsymbol{I}(\lvert u\rvert \leq 1)$
Quartic (Biweight)	$\frac{15}{16}(1 - u^2)^2\boldsymbol{I}(\lvert u\rvert \leq 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$

Table 2.3: A list of commonly used kernels, (Härdle & Simar, 2007, p. 14)

The basic principle behind KDE is to apply the kernel centered on each sample x_i, the density estimate is then the average over all kernels, as indicated by equation (2.16).

The MATLAB function `ksdensity` serves the purpose of KDE and will be used in this work.

2.1.5 *Sobol' decomposition of the PCE*

In order to compute the sensitivity indices as will be explained later in section 2.2.3, it is convenient to express the gPC expansion of the model $f_{PC}(Z) \approx f(Z) = Y$ by means of the so-called Sobol' decomposition, which will be detailed in what follows.

Recall that the gPC expansion of Y is given by the following expression: $f_{PC}(Z) = \sum_{k=0}^{P-1} y_k \Psi_k(Z)$. Now, represent the multivariate orthogonal polynomial being used by means of the multi-

index $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, that is:

$$\boldsymbol{\Psi}_{\boldsymbol{\alpha}} = \prod_{i=1}^{d} \Psi_{\alpha_i}(Z_i) \tag{2.17}$$

Then define $\mathcal{I}_{i_1,\ldots,i_s}$, the set of $\boldsymbol{\alpha}$ tuples such that only the indices $(i_1, \ldots, i_s)$ are non-zero:

$$\mathcal{I}_{i_1,\ldots,i_s} = \left\{ \boldsymbol{\alpha} : \begin{array}{lll} \alpha_k > 0 & \forall k = 1, \ldots, d, & k \in (i_1, \ldots, i_s) \\ \alpha_k = 0 & \forall k = 1, \ldots, d, & k \notin (i_1, \ldots, i_s) \end{array} \right\} \tag{2.18}$$

Since $\mathcal{I}_i$ corresponds to the polynomials depending only on parameter Z_i, the terms in the gPC expansion in 2.11 may now be grouped according to the parameters they depend on, as described by Sudret (2008, p. 971):

$$\begin{aligned} f_{PC}(Z) = & f_0 + \sum_{1 \le i \le d} \left(\sum_{\boldsymbol{\alpha} \in \mathcal{I}_i} y_{\boldsymbol{\alpha}} \boldsymbol{\Psi}_{\boldsymbol{\alpha}}(Z_i) \right) \\ & + \sum_{1 \le i_1 < i_2 \le d} \left(\sum_{\boldsymbol{\alpha} \in \mathcal{I}_{i_1,i_2}} y_{\boldsymbol{\alpha}} \boldsymbol{\Psi}_{\boldsymbol{\alpha}}(Z_{i_1}, Z_{i_2}) \right) + \ldots \\ & + \sum_{1 \le i_1 < \cdots < i_s \le d} \left(\sum_{\boldsymbol{\alpha} \in \mathcal{I}_{i_1,\ldots,i_s}} y_{\boldsymbol{\alpha}} \boldsymbol{\Psi}_{\boldsymbol{\alpha}}(Z_{i_1}, \ldots, Z_{i_s}) \right) \\ & + \cdots + \sum_{\boldsymbol{\alpha} \in \mathcal{I}_{1,2,\ldots,d}} y_{\boldsymbol{\alpha}} \boldsymbol{\Psi}_{\boldsymbol{\alpha}}(Z_1, \ldots, Z_d) \end{aligned} \tag{2.19}$$

Due to the uniqueness of the Sobol' decomposition, the expansion in equation 2.19 is unique see, Sudret (2008, p. 971) and thus it is *the* Sobol' decomposition of $f_{PC}(Z)$.

Next, an introduction to Sensitivity Analysis (SA) is presented together with a description of variance-based measures and the definition of the Sobol' indices.

2.2 Sensitivity Analysis

A possible definition of Sensitivity Analysis (SA) according to Andrea Saltelli et al. (2004, p. 45) is the following: "*The study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input*". However, in order to answer problem-specific questions, the choice of an appropriate SA method and an adequate sensitivity measure must be carried out carefully, taking into account a number of "desired properties" for the analysis and "possible settings" of constraints of the model being used (Andrea Saltelli et al., 2004, p. 45).

Independently of the method being used, the general methodology in SA (as described by Andrea Saltelli et al., 2004, pp. 45–47) is as follows:

1. Establish the goal of the analysis and the output function that answers the question(s) of interest.
2. Decide which input factors should be investigated.
3. Choose a distribution function for each of the inputs, from the literature, derived from data, based on an expert's opinions, etc.

4. Choose a SA method based on:

   The questions being addressed.

   The number of model evaluations that can be afforded.

   The presence of correlations between input factors.

5. Generate the input sample, according to the method used for the SA.
6. Evaluate the model on the generated sample and produce the output.
7. Analyze the model outputs and draw conclusions from them. It is also possible that a new iteration of the whole process is needed.

Consequently, in order to select an appropriate SA method it is important to define which are the “desired properties", which will be briefly presented here (see Andrea Saltelli et al., 2004, pp. 47–49):

Scale and shape Ability to handle different shapes and forms of PDFs.

Multidimensional averaging Evaluate the effect of a factor while all others are also varying.

Model independence The method should work independently of the additivity or linearity of the model. It should as well be able to capture interaction effects.

Grouping of factors Being able to treat grouped factors as if they were single factors.

For further details on the subject, the reader is referred to Andrea Saltelli et al. (2004, Chapter 2).

2.2.1 *Local vs. global*

In addition, SA methods can be categorized into two groups, local methods and global ones.

On the one hand local methods are usually based on partial derivatives, that is, the influence of the variation of an input factor X_j *while all others are kept constant* on the model output Y. This influence can be computed by means of a partial derivative:

$$S_j = \frac{\partial Y}{\partial X_j} \tag{2.20}$$

While useful in some situations, this method lack some important desired properties for SA according to Andrea Saltelli et al. (2004, p. 48, Table 2.1).

On the other hand global methods explore the entire interval of definition or the full spectrum of each factor, as opposed to local methods, where only *one point* of the factors' space is explored. Therefore the influence of the variation of an input factor on the model output is in fact an average over all possible values of the other factors, for that reason global methods possess multidimensional scaling (Andrea Saltelli et al., 2004, p. 44 ff.).

There are several global SA methods. In what follows, variance-based sensitivity will be presented. For further information on different global sensitivity measures the reader is again referred to the literature on the subject, such as Andrea Saltelli et al. (2004) or A. Saltelli et al. (2008).

2.2.2 *Variance-based measures*

As shown in Andrea Saltelli et al. (2004, p. 48, Table 2.1) variance-based methods possess all of the aforementioned desirable properties for a sensitivity measure and are thus very useful in a wide number of scenarios.

Variance-based measures are based on the so-called Analysis of Variance (ANOVA) or Sobol' functional decomposition, as presented in Sobol (2001, p. 272); Crestaux, Le Maitre, and Martinez (2009, p. 1165) or Sudret (2008, p. 966) and shown in equation (2.21).

Consider, for simplicity, the d-dimensional random vector of input variables $\boldsymbol{\xi}$, whose components are independent identically distributed random variables ξ_i, with $\Omega^d = \underbrace{\Omega \times \cdots \times \Omega}_{d\text{-times}}$ as range and $p(\boldsymbol{\xi})$ as joint PDF. Since the random variables are independent the joint PDF can be expressed as the product of the marginal PDFs: $p(\boldsymbol{\xi}) = \prod_i p(\xi_i)$. Finally, the model output $Y = f(\xi)$ assuming that $\boldsymbol{\xi} \in \Omega^d \mapsto f(\boldsymbol{\xi}) \in L^2(\Omega^d, p(\xi))$ can be expressed by its Sobol' functional decomposition:

$$f(\boldsymbol{\xi}) = f_0 + \sum_{s=1}^{d} \sum_{i_1<\cdots<i_s}^{d} f_{i_1\ldots i_d}(\xi_{i_1}, \ldots, \xi_{i_s}) = f_0 + \sum_i f_i(\xi_i) + \sum_{i<j} f_{ij}(\xi_i, \xi_j) + \cdots + f_{12\ldots d}(\xi_1, \ldots, \xi_d) \tag{2.21}$$

Whose total number of summands is 2^d. As detailed in Crestaux et al. (2009, p. 1165), it can be shown that the integral of each summand $f_{i_1\ldots i_s}(\xi_{i_1}, \ldots, \xi_{i_s})$ over any of its independent variables ξ_{i_k} is zero:

$$\int_\Omega f_{i_1\ldots i_s}(\xi_{i_1}, \ldots, \xi_{i_s}) p(\xi_{i_k}) \mathrm{d}\xi_{i_k} = 0, \qquad \forall i_k \in (i_1, \ldots, i_s) \tag{2.22}$$

This implies the orthogonality of the summands in equation (2.21), let $\boldsymbol{u} = (u_1, \ldots, u_s)$ be a multi-index, with $|\boldsymbol{u}| = \mathrm{card}(\boldsymbol{u}) = s$, and let $\boldsymbol{v}$ be another multi-index defined analogously, then:

$$\int_{\Omega^d} f_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}}) f_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = 0, \qquad \forall \boldsymbol{u} \neq \boldsymbol{v} \tag{2.23}$$

And thus they can be computed as integrals of the model output $f(\boldsymbol{\xi})$, indeed according to Crestaux et al. (2009, p. 1165):

$$f_{\boldsymbol{u}}(\boldsymbol{\xi}_{\boldsymbol{u}}) = \int_{\Omega^{d-|\boldsymbol{u}|}} f(\boldsymbol{\xi}) p(\boldsymbol{\xi}_{\sim \boldsymbol{u}}) \mathrm{d}\boldsymbol{\xi}_{\sim \boldsymbol{u}} - \sum_{\substack{\boldsymbol{v} \subset \boldsymbol{u} \\ \boldsymbol{v} \neq \boldsymbol{u}}} f_{\boldsymbol{v}}(\boldsymbol{\xi}_{\boldsymbol{v}}) \tag{2.24}$$

Where $\boldsymbol{\xi}_{\sim \boldsymbol{u}}$ is the vector $\boldsymbol{\xi}$ without the elements of the multi-index $\boldsymbol{u}$. For example:

$$\boldsymbol{\xi}_{\sim i} = (\xi_1, \ldots, \xi_{i-1}, \xi_{i+1}, \ldots, \xi_d) \tag{2.25}$$

With this functional decomposition and defining the total variance D of the output $Y = f(\boldsymbol{\xi})$ as described in Crestaux et al. (2009, p. 1165):

$$D = \int_{\Omega^d} f^2(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} - f_0^2 \tag{2.26}$$

The conditional variances $D_{\boldsymbol{u}}$ are easily defined by:

$$D_{\boldsymbol{u}} = \int_{\Omega^{|\boldsymbol{u}|}} f_{\boldsymbol{u}}^2(\boldsymbol{\xi}_{\boldsymbol{u}}) p(\boldsymbol{\xi}_{\boldsymbol{u}}) \mathrm{d}\boldsymbol{\xi}_{\boldsymbol{u}} \tag{2.27}$$

Furthermore, according to Crestaux et al. (2009) $D_{\boldsymbol{u}}$ can also be expressed as a combination of conditional variances:

$$D_{\boldsymbol{u}} = \mathrm{Var}(\mathbb{E}[f(\boldsymbol{\xi})|\boldsymbol{\xi}_{\boldsymbol{u}}]) - \sum_{\substack{\boldsymbol{v} \subset \boldsymbol{u} \\ \boldsymbol{v} \neq \boldsymbol{u} \\ \boldsymbol{v} \neq \emptyset}} D_{\boldsymbol{v}} \tag{2.28}$$

Finally, due to the orthogonality of the decomposition, the sum of the conditional variances $D_{\boldsymbol{u}}$ is the total variance of the output D:

$$D = \sum_{\substack{\boldsymbol{u} \subseteq \{1,2,\ldots,d\} \\ \boldsymbol{u} \neq \emptyset}} D_{\boldsymbol{u}} \tag{2.29}$$

(See Crestaux et al., 2009).

2.2.3 *Sobol' indices*

From the above definitions one can then define the (global) Sobol' sensitivity indices of order $s = |\boldsymbol{u}|$ as:

$$S_{\boldsymbol{u}} = S_{u_1,\ldots,u_s} = \frac{D_{u_1,\ldots,u_s}}{D} = \frac{D_{\boldsymbol{u}}}{D} \tag{2.30}$$

(See Crestaux et al., 2009).

And from equation (2.28) one has that:

$$\sum_{\substack{\boldsymbol{u} \subseteq \{1,2,\ldots,d\} \\ \boldsymbol{u} \neq \emptyset}} S_{\boldsymbol{u}} = 1 \tag{2.31}$$

As described in Crestaux et al. (2009, p. 1166), each Sobol' index $S_{\boldsymbol{u}}$ is a measure of the contribution of the variance of the variables in $\boldsymbol{\xi}_{\boldsymbol{u}}$ to the variance in the output $Y = f(\boldsymbol{\xi})$, without taking into account the effect of the variables in $\boldsymbol{\xi}_{\boldsymbol{v}}$ for $\boldsymbol{v} \subset \boldsymbol{u}$ and $\boldsymbol{v} \neq \boldsymbol{u}$. That is, for example, the second-order sensitivity index S_{ij}, measures the sensitivity of the variance of the output $Y = f(\boldsymbol{\xi})$ with respect to ξ_i and ξ_j, without taking into account the effects of each variable separately (described by S_i and S_j).

Furthermore, the number of Sobol' sensitivity indices increases exponentially with the dimension d as $2^d - 1$. Consequently, specially for problems with many uncertain parameters, total Sobol' indices were introduced first by Homma and Andrea Saltelli (1996).

Total Sobol' sensitivity indices denoted by S_{T_i} measure the total sensitivity of the variance of $Y = f(\boldsymbol{\xi})$ caused directly by variable ξ_i and all of its interactions with the rest of the variables in $\boldsymbol{\xi}$. According to Crestaux et al. (2009, p. 1166), the total Sobol' sensitivity indices can be defined as:

$$S_{T_i} = \sum_{\boldsymbol{u} \ni i} S_{\boldsymbol{u}} \tag{2.32}$$

Or, alternatively, as defined by Sudret (2008, p. 967), as the sum of the partial sensitivity indices $S_{i_1,\ldots,i_s}$ involving index i:

$$S_{T_i} = \sum_{\mathcal{I}_i} S_{i_1,\ldots,i_s}, \qquad \text{with} \quad \mathcal{I}_i = \{(i_1,\ldots,i_s) : \exists k,\ 1 \leq k \leq s,\ i_k = i\} \tag{2.33}$$

For example for the 3-dimensional case ($d = 3$) S_{T_1} can be computed as:

$$S_{T_1} = S_1 + S_{12} + S_{13} + S_{123}$$

An important property, as described in Crestaux et al. (2009, p. 1166), is that:

$$\underbrace{\frac{\mathbb{E}[\mathrm{Var}(Y|\boldsymbol{\xi}_{\sim i})]}{\mathrm{Var}(Y)}}_{S_{T_i}} + \frac{\mathrm{Var}(\mathbb{E}[Y|\boldsymbol{\xi}_{\sim i}])}{\mathrm{Var}(Y)} = 1 \tag{2.34}$$

And consequently, S_{T_i} can be also expressed, according to Sudret (2008, p. 967), as:

$$S_{T_i} = 1 - S_{\sim i} \tag{2.35}$$

Where $S_{\sim i}$ is the sum of all partial sensitivity indices $S_{i_1,\ldots,i_s}$ that do not contain index i.

As it will be shown later in section 3.2, Sobol' indices can be directly and easily computed from the spectral coefficients of the PCE, but they can also be computed with classical methods such as Monte Carlo (MC) or even Quasi-Monte Carlo (QMC) simulation. This last alternative will come in useful in order to compare and validate the results of the PC-based method.

3 NUMERICAL METHODS

In this chapter, the numerical methods used in this work will be presented. Starting with some of the most common methods for the computation of the PCE coefficients, whilst describing first the existing differences between the intrusive and non-intrusive approaches. Following, the PC-based technique used to compute the Sobol' sensitivity indices will be presented and compared with the sampling-based methods.

3.1 Computation of the PCE Coefficients

As presented in section 2.1 and in particular in equation (2.11), the use of gPC expansions requires the computation of a number of deterministic (in general, time-dependent) coefficients or unknowns, the so-called spectral coefficients ($y_k, \quad k \in 0, 1, \ldots, P$). The determination of these coefficients is the most expensive step in terms of computational cost, since the post-processing, i.e. computing statistical moments or the Sobol' indices is relatively fast once the coefficients are known. For the purpose of computing the spectral coefficients, a number of methods are available, some of which will be presented in the following section.

3.1.1 *Intrusive vs. non-intrusive methods*

First of all, a distinction must be made, between intrusive and non-intrusive methods. The intrusive non-deterministic method, is based on a Galerkin projection. According to Xiu and Karniadakis (2002, p. 9 ff.) if the following Stochastic Differential Equation (SDE) is considered:

$$\begin{cases} \mathcal{L}(x, t, \boldsymbol{\xi}; Y) = f(x, t; \boldsymbol{\xi}), & (0, T] \wedge \mathbb{R}^d, \quad (d \in \mathbb{N}_0) \\ Y = Y_0, & t_0 \wedge \mathbb{R}^d \end{cases} \tag{3.1}$$

where $Y = Y(x, t; \boldsymbol{\xi})$ is the solution of the SDE and $f(x, t; \xi)$ is the right-hand side, containing no differential operators. Furthermore, the differential operator $\mathcal{L}$, be it in space and/or time, can be linear or non-linear. Moreover, the initial value Y_0 must be considered for the existence of a solution to the initial value problem. The vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N) \in \mathbb{R}^N$ represents the random input parameters, which can indeed refer to initial, boundary or system parameters.

The Galerkin projection method is firstly based on the approximation of the solution by a gPC expansion. Next, the SDE is projected onto the subspace of orthogonal polynomials Ψ_j being used and the ensemble average or inner product is further applied. As presented by Xiu and Karniadakis (2002), the resulting coupled deterministic system of equations is:

$$\left\langle \mathcal{L}\left(x, t, \boldsymbol{\xi}; \sum_{k=0}^{P-1} y_k \Psi_k\right), \Psi_i \right\rangle = \langle f(x, t; \boldsymbol{\xi}), \Psi_i \rangle \tag{3.2}$$

Afterwards the orthogonality property of the polynomial basis can be used to simplify the approximate solution. Nonetheless, a general solution to equation (3.2) does not exist and a problem-specific numerical method must be explored. For that reason this method is sometimes qualified as *intrusive*, since according to Sudret (2008, p. 968), it requires heavy *ad hoc* implementation for the specific problem of interest.

As an alternative, non-intrusive methods have been recently developed, for example the Non-intrusive Spectral Projection (NISP) method and the Linear Regression Method (LRM). In the following, both methods will be briefly explained and two different strategies for the numerical integration of the expressions appearing in the NISP will be presented, namely the tensor product rule and Smolyak's Sparse Grid Integration (SGI).

3.1.2 *Linear regression method*

The linear regression approach is based on the regression of the exact solution Y with respect to the gPC basis: $\{\Psi_j(\boldsymbol{\xi}), \quad j = 0, \ldots, P-1\}$. As described in Sudret (2008, p. 969), assuming that the scalar response quantity Y can be expressed as:

$$Y = f(\boldsymbol{Z}) = \tilde{Y}(\boldsymbol{\xi}) + \varepsilon \tag{3.3}$$

$$\tilde{Y}(\boldsymbol{\xi}) = \sum_{k=0}^{P-1} y_k \Psi_k(\boldsymbol{\xi}) \tag{3.4}$$

where the residual ε is supposed to be a zero-mean variable and $\mathcal{Y} = \{y_k,\ k = 0, \ldots, P-1\}$ is the set of unknown coefficients. Minimizing the Mean Squared Error (MSE) of the residual with respect to the unknown coefficients leads to:

$$\mathcal{Y} = \text{Argmin}\left\{\mathbb{E}[(f(\boldsymbol{Z}(\boldsymbol{\xi})) - \tilde{Y}(\boldsymbol{\xi}))^2]\right\} \tag{3.5}$$

Equation (3.5) can be solved by choosing a set of N regression points $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^N\}$. Then, according to Sudret (2008) the MSE minimization leads to:

$$\mathcal{Y} = \text{Argmin}\left\{\frac{1}{N}\sum_{j=1}^{N}\left[f(\boldsymbol{z}^j) - \sum_{k=0}^{P-1} y_k \Psi_k(\boldsymbol{\xi}^j)\right]^2\right\} \tag{3.6}$$

The solution to equation (3.6) can then be obtained as:

$$\mathcal{Y} = (\boldsymbol{\Psi}^T\boldsymbol{\Psi})^{-1}\,\boldsymbol{\Psi}^T\,\mathcal{Y}_{\text{ex}} \tag{3.7}$$

Where $\boldsymbol{\Psi}$ is a matrix whose entries are $\boldsymbol{\Psi}_{ij} = \Psi_j(\boldsymbol{\xi}^j), \quad \forall i = 1, \ldots, N,\ \forall j = 0, \ldots, P-1$ and $\mathcal{Y}_{\text{ex}} = \{f(\boldsymbol{z}^i),\ i = 1, \ldots, N\}$ is the vector containing the exact response of the model at the regression points, the set $\{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^N\}$ also known as the *experimental design*. The matrix $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ is the so-called *information matrix*, which might be, unfortunately, ill-conditioned. Thus a specific solver such as the singular value decomposition method should be used, as suggested by Sudret (2008, p. 969).

The information matrix $\boldsymbol{A} = \boldsymbol{\Psi}^T\boldsymbol{\Psi}$ can be expressed as:

$$A_{ij} = \sum_{k=1}^{N} \Psi_i(\boldsymbol{\xi}^k)\Psi_j(\boldsymbol{\xi}^k) \tag{3.8}$$

And therefore $\boldsymbol{A}$ can be seen as the sum of N basic matrices $a_{ij}^k = \Psi_i(\boldsymbol{\xi}^k)\Psi_j(\boldsymbol{\xi}^k)$. In order for the information matrix $\boldsymbol{A}$ to be invertible, it is obvious that $\text{rank}(\boldsymbol{A}) \geq P$. According to Sudret (2008), since the basic matrices $\boldsymbol{a}^k$ are square matrices of size P and they have unit rank, at least P such matrices should be added.

The optimal design of experiment in terms of the D-optimality criterion is given by the tensor product of the roots of the orthogonal polynomial being used ($\{r_1, \ldots, r_P\}$) in the gPC expansion, that is, the set of M-tuples built using all possible combinations of the unidimensional roots, as presented by Sudret in Sudret (2008, p. 969):

$$\boldsymbol{r}^k = (r_{i_1}, \ldots, r_{i_M}), \qquad 1 \leq i_1 \leq \cdots \leq i_M \leq P, \quad k = 1, \ldots, P^M \tag{3.9}$$

Nevertheless, since only the multivariate polynomials of total degree less than $P-1$ are considered (and not the full tensor of unidimensional polynomials of degree less than $P-1$), the following iterative algorithm can be used, in order to obtain an improved design of experiment which requires fewer evaluations of the model, according to Sudret (2008, p. 970 ff.):

- Compute the M-tuples based on the roots of the orthogonal polynomial being used in the gPC expansion.
- Order the M-tuples by increasing norm.
- Iteratively assemble the information matrix $\boldsymbol{A}$ by adding the corresponding basic matrices $\boldsymbol{a}^k$ for the kth M-tuple, until it becomes invertible.

Note that the number of model evaluations N depends on the number of random input variables M and the degree of the gPC expansion $P-1$, and that it cannot be determined *a priori* and is also a result of the iterative assembly process. Furthermore, although the information matrix $\boldsymbol{A}$ built in this way is invertible, it might still be ill-conditioned, as mentioned previously, so that a singular value decomposition algorithm should still be used for the computation of the rank at each step of the assembly process. In this work MATLAB's `rank` function is used, which applies such a singular value decomposition algorithm.

3.1.3 *Non-intrusive spectral projection*

Another approach for the computation of the spectral coefficients of the gPC expansion is the NISP method. This technique takes advantage of the orthogonality of the PC basis. Computing the inner product of the gPC expansion of the model output $Y = f(\boldsymbol{\xi})$ with Ψ_j:

$$\left\langle f(\boldsymbol{\xi}), \Psi_j \right\rangle = \left\langle \sum_{k=0}^{P-1} y_k \Psi_k(\boldsymbol{\xi}), \Psi_j \right\rangle \tag{3.10}$$

Using the orthogonality property ($\left\langle \Psi_k, \Psi_j \right\rangle = 0, \quad \forall k \neq j$ and $\langle \Psi_k, \Psi_k \rangle = \gamma_k$) one obtains, according to Crestaux et al. (2009, p. 1164):

$$y_j = \frac{\left\langle f(\boldsymbol{\xi}), \Psi_j(\boldsymbol{\xi}) \right\rangle}{\left\langle \Psi_j, \Psi_j \right\rangle}, \qquad \forall j = 0, \ldots, P-1 \tag{3.11}$$

In this expression the denominator $\left\langle \Psi_j^2 \right\rangle$ is known analytically for the corresponding polynomial basis being used in the gPC expansion and the numerator may be computed from the

definition of the weighted inner product as the following integral:

$$\left\langle f(\boldsymbol{\xi}), \Psi_j(\boldsymbol{\xi}) \right\rangle = \int_{\Omega^d} f(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \tag{3.12}$$

Consequently, the problem now reduces to finding an appropriate multi-dimensional numerical integration method. In this regard, several schemes are available with different properties that have to be considered. Some of the best known methods and their properties are summarized below, as according to Crestaux et al. (2009, p. 1164):

Monte Carlo Integration robust with respect to the smoothness of the integrand, with convergence guaranteed for any L^2-function and a convergence rate independent of the dimension d. However, it has a poor convergence rate of order $\mathcal{O}(1/\sqrt{n})$.

Quasi Monte Carlo Integration less robust than classical MC, but with a convergence rate of order $\mathcal{O}(\log(n)^d/n)$, if some assumptions regarding the smoothness of the integrand are fulfilled.

Full Tensorization / Tensor Product Rule quadrature with fast convergence rate for smooth integrands. Nevertheless, its computational cost increases exponentially with the stochastic dimension d as: $n = (n_1)^d$, with n_1 the number of quadrature points in the one-dimensional formula.

Composite Methods based on adaptive partitions of the integration domain, they are also robust, but their computational cost also increases with the stochastic dimension d exponentially.

Smolyak's Cubature if the integrand is relatively smooth its convergence rate is fast and the computation cost increases much slower than for the fully tensored formula.

Previous works on the usage of PCE for uncertainty quantification in vehicle dynamics simulation, such as the one by Cristoph Stadler (see Stadler, 2014–2015), were based on the tensor product rule. In the present work, however, sparse grid integration was additionally implemented and investigated in order to reduce the number of model evaluations and the computation cost of the gPC algorithm.

Therefore, in what follows, the tensor product rule and sparse grid integration (Smolyak's cubature) will be detailed and the results obtained with both methods will be compared later in chapter 4.

3.1.3.1 Tensor product rule

The tensor product rule is based on a full tensorization of the unidimensional quadrature rule, which is built upon the roots of an appropriate polynomial basis chosen according to the Askey-scheme presented in table 2.1. Let $Q_M[f]$ be the Mth-order unidimensional quadrature operator, such that:

$$I_1[f] = \int_{\Omega} f(\xi) p(\xi) \mathrm{d}\xi \approx Q_M[f] = \sum_{i=1}^{M} w^i f(\xi^i) \tag{3.13}$$

Such an unidimensional quadrature rule of order M can exactly integrate polynomials of degree less than or equal to $2M-1$ (see Florian Heiss & Viktor Winschel, 2006, p. 3).

Consider now the multivariate case where the following integral is to be evaluated:

$$I_d[f] = \int_{\Omega^d} f(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \int_{\Omega_1} \dots \int_{\Omega_d} f(\xi_1, \dots, \xi_d) p(\xi_1, \dots, \xi_d) \mathrm{d}\xi_d \dots \mathrm{d}\xi_1 \tag{3.14}$$

Univariate quadrature rules can be extended rather easily to the multivariate case by the product rule, that is, by defining the tensor product over univariate quadrature rules. Considering that the input random variables are independent, i.e. $p(\boldsymbol{\xi}) = \prod_i p(\xi_i)$, and the set of nodes $\Xi_{M_i} \subset \mathbb{R}$ and weights, w_{M_i} obtained from the corresponding univariate quadrature rules Q_{M_i}, $\forall i = 1, \dots, d$, the tensor product rule can be expressed as:

$$(Q_{M_1} \otimes \dots \otimes Q_{M_d})[f] = \sum_{\xi_1 \in \Xi_{M_1}} \dots \sum_{\xi_d \in \Xi_{M_d}} f(\xi_1, \dots, \xi_d) \prod_{j=1}^{d} w_{M_j}(\xi_j) \tag{3.15}$$

As clarified by Florian Heiss and Viktor Winschel (2006, p. 6), if the accuracy level M is the same for each dimension, the product rule is then $(Q_M \otimes \dots \otimes Q_M)[f]$ and thus the model $f(\boldsymbol{\xi})$ is evaluated at the full grid of points $\Xi_M \otimes \dots \otimes \dots \Xi_M$, consequently requiring M^d evaluations. This exponential growth of the computational cost is known as *curse of dimensionality* and restricts the usage of the tensor product rule for higher dimensions. Therefore a tailored product rule which restricts the total order (the sum of exponents) of the polynomials and not the maximum exponent of each monomial would reduce the required amount of model evaluations. A feasible solution to the curse of dimensionality is detailed in the following section, i.e. sparse grid integration, particularly, Smolyak's construction of cubature formulas.

3.1.3.2 Smolyak's Cubature

The method devised by Smolyak (1963) is based on a partial tensorization (sparse grid) of univariate quadrature formulas.

According to Crestaux et al. (2009, p. 1165), if we consider the sequence of unidimensional integration formulas with increasing accuracy level given by $L = 0, 1, \dots$ and corresponding with a number of nodes n_L. Each univariate quadrature formula of level L can be expressed similarly to equation (3.13)

$$\int_{\Omega} f(\xi) p(\xi) \mathrm{d}\xi \approx Q_L^1[f] = \sum_{i=1}^{n_L} f(\xi^{i,L}) w^{i,L} \tag{3.16}$$

Now, according to Crestaux et al. (2009, p. 1165), by setting $Q_0^1[f] = 0$, the difference quadrature formula is defined as:

$$\Delta_{k \geq 1}^1[f] = (Q_k^1 - Q_{k-1}^1)[f] \tag{3.17}$$

And using the multi-index $\boldsymbol{k} = (k_1, \dots, k_d)$, the d-dimensional difference formula is computed as the tensor product of the unidimensional difference quadratures:

$$\Delta_{\boldsymbol{k}}[f] = (\Delta_{k_1}^1 \otimes \dots \otimes \Delta_{k_d}^1)[f] \tag{3.18}$$

Therefore, the d-dimensional Smolyak's cubature formula of level L is constructed by the sum of the tensor products of the difference quadratures, over the set of multi-indices such

that: $|\boldsymbol{k}| \leq L + d - 1$. That is:

$$\int_{\Omega^d} f(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \approx Q_L^d[f] = \sum_{|\boldsymbol{k}| \leq L+d-1} \Delta_{\boldsymbol{k}}[f], \qquad L \in \mathbb{N}_0,\ \boldsymbol{k} \in \mathbb{N}_0^d \tag{3.19}$$

In this work, a slightly modified version of the MATLAB code offered by Florian Heiss and Viktor Winschel (2006) (freely available at $\mathrm{http://sparse\text{-}grids.de}$) shall be used, which computes the set of nodes and their corresponding weights for the multi-dimensional integration. In this MATLAB function (`nwspgr.m`), four different quadrature rules are implemented: two corresponding to the uniform distribution (unweighted integration in $\Omega = [0, 1]^d,\ w = 1$), an univariate Gaussian quadrature rule and a nested univariate Kronrod-Patterson rule; and two corresponding to a Gaussian distribution (integration in $\Omega = \mathbb{R}^d$ with weighting function $w(x) = \frac{1}{2\pi} e^{-\frac{x^2}{2}}$), again an univariate Gaussian quadrature rule and a nested univariate Kronrod-Patterson rule. Note again that each rule with accuracy level L exactly integrates complete polynomials of total order up to $2L - 1$.

Using sparse grid integration, the number of model evaluations is drastically reduced. However, it must be noted that although Smolyak's method has a fast convergence rate for smooth integrands, the smoothness of the model output $f(\boldsymbol{\xi})$ is not known *a priori*, so that the integration scheme must be carefully chosen, since as noted by Crestaux et al. (2009, p. 1165) there exists an optimal order for the PCE for which the projection error is minimized and if the order is either too high or too low the projection error increases. Consequently, the need for a validation of the results cannot be understated. The validation can be performed by comparing the results of the PC approximation with the results obtained by sampling techniques, such as MCS or QMC simulation, or by means of confidence intervals for some given statistics. Therefore, following an explanation of the computation of PC-based Sobol' indices, both sampling techniques and the computation of confidence intervals using bootstrapping will be presented.

3.2 Computation of the Sobol' Indices

3.2.1 *PC-based Sobol' indices*

Once the coefficients of the gPC have been computed as shown in section 3.1, and the gPC expansion is thus fully determined, the PC-based Sobol' indices of sth-order can be easily computed as follows (see Sudret, 2008, p. 971 ff. or Crestaux et al., 2009, p. 1167):

$$S_{i_1,\ldots,i_s} \approx S^{PC}_{i_1,\ldots,i_s} = \sum_{\boldsymbol{\alpha} \in \mathcal{I}_{i_1,\ldots,i_s}} \frac{y_{\boldsymbol{\alpha}}^2 \left\langle \Psi_{\boldsymbol{\alpha}}^2 \right\rangle}{D_{\mathrm{PC}}} \tag{3.20}$$

Where $y_{\boldsymbol{\alpha}}$ are the spectral coefficients as defined in equation (2.19), and $D_{\mathrm{PC}} = \sum_{k=0}^{P-1} y_k^2 \left\langle \Psi_k^2 \right\rangle$ the PC approximation of the total variance of model output Y. Note that the expansion coefficients are grouped according to their dependency on each basis polynomial, square-summed and normalized (Sudret, 2008, p. 971).

Finally, the total Sobol' indices can also be computed by virtue of equation (2.32) or (2.35).

3.2.2 *Sampling-based Sobol' indices*

Although Sobol' indices can be easily approximated by means of a gPC expansion of the model output Y, they can also be computed using a sampling approach. Thus, in order to validate the approximation obtained with the PC-based approach, a sampling-based method will also be used. First of all, a MCS approach will be introduced, before low-discrepancy sampling methods are briefly presented.

Consider the expression for the total variance of the model output D given by equation (2.26). Using the MC estimator with a sample set of N realizations of the random input variables $\{\boldsymbol{\xi}^{(i)}\}_{i=1}^{N}$, the expression can be rewritten as follows (see Crestaux et al., 2009, p. 1166):

$$D \approx \hat{D} = \frac{1}{N}\sum_{i=1}^{N} f^2(\boldsymbol{\xi}^{(i)}) - \hat{f}_0^2 \tag{3.21}$$

With $\hat{f}_0$ the sample estimate of the average of the model output:

$$\hat{f}_0 = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{\xi}^{(i)}) \tag{3.22}$$

Then in order to compute the conditional variances $D_{\boldsymbol{u}}$ appearing in equation (2.28) one can use two independent samples $\{\boldsymbol{\xi}^{(i)}\}_{i=1}^{N}$ and $\{\boldsymbol{\eta}^{(i)}\}_{i=1}^{N}$ each of N points, as described in Crestaux et al. (2009, p. 1166). Using the two sample sets the conditional variance $\mathrm{Var}(\mathbb{E}[Y|\boldsymbol{\xi}_{\boldsymbol{u}}]) = \mathbb{E}[\mathbb{E}[Y|\boldsymbol{\xi}_{\boldsymbol{u}}]^2] - \mathbb{E}[Y]^2$ can be rewritten as:

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[Y|\boldsymbol{\xi}_{\boldsymbol{u}}]^2] &= \mathbb{E}[\mathbb{E}[Y|\boldsymbol{\xi}_{\boldsymbol{u}}]\,\mathbb{E}[Y|\boldsymbol{\xi}_{\boldsymbol{u}}]] = \\
&= \int \left(\int f(\boldsymbol{\xi}_{\sim\boldsymbol{u}}, \boldsymbol{\xi}_{\boldsymbol{u}}) p(\boldsymbol{\xi}_{\sim\boldsymbol{u}}) \mathrm{d}\boldsymbol{\xi}_{\sim\boldsymbol{u}}\right) \times \\
&\times \left(\int f(\boldsymbol{\xi}_{\sim\boldsymbol{u}}, \boldsymbol{\xi}_{\boldsymbol{u}}) p(\boldsymbol{\xi}_{\sim\boldsymbol{u}}) \mathrm{d}\boldsymbol{\xi}_{\sim\boldsymbol{u}}\right) p(\boldsymbol{\xi}_{\boldsymbol{u}}) \mathrm{d}\boldsymbol{\xi}_{\sim\boldsymbol{u}} = \\
&= \int\int\int f(\boldsymbol{\xi}) f(\boldsymbol{\eta}_{\sim\boldsymbol{u}}, \boldsymbol{\xi}_{\boldsymbol{u}}) p(\boldsymbol{\xi}) p(\boldsymbol{\eta}_{\sim\boldsymbol{u}}) \mathrm{d}\boldsymbol{\xi} \mathrm{d}\boldsymbol{\eta}_{\sim\boldsymbol{u}}
\end{aligned} \tag{3.23}$$

According to Crestaux et al. (2009), after some manipulations the conditional variance $D_{\boldsymbol{u}}$ can be approximated by its sample estimate as:

$$D_{\boldsymbol{u}} \approx \hat{D}_{\boldsymbol{u}} = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{\xi}^{(i)}) f(\boldsymbol{\zeta_u}^{(i)}) - \sum_{\substack{\boldsymbol{v}\subset\boldsymbol{u} \\ \boldsymbol{v}\neq\boldsymbol{u}}} \hat{D}_{\boldsymbol{v}} \tag{3.24}$$

Where the sample set $\boldsymbol{\zeta_u}^{(i)}$ is defined by:

$$(\zeta_j)_{\boldsymbol{u}}^{(i)} = \begin{cases} \xi_j^{(i)} & \text{if } j \in \boldsymbol{u}, \\ \eta_j^{(i)} & \text{otherwise} \end{cases} \tag{3.25}$$

Finally, the sampling-based Sobol' indices of order s are computed from equation (2.30):

$$\hat{S}_{u_1,\ldots,u_s} = \hat{S}_{\boldsymbol{u}} = \frac{\hat{D}_{\boldsymbol{u}}}{\hat{D}} \tag{3.26}$$

Analogously the sampling-based method for the computation of the total Sobol' indices can be derived from equation (2.34):

$$\hat{S}_{T_i} = 1 - \frac{1}{\hat{D}} \left(\frac{1}{N} \sum_{k=1}^{N} f(\boldsymbol{\xi}^{(k)}) f(\zeta_i^{(k)}) - \hat{f}_0^2 \right) \tag{3.27}$$

Where $\hat{f}_0$ is given by equation (3.22). In practice, instead of generating two different independent sets of samples, a quasi-random low-discrepancy sequence is used (in this work, the Sobol sequence), which improves the convergence rate of the sensitivity index estimate, as noted by Crestaux et al. (2009, p. 1167). However, since the sample sets must be independent, in practice both sets are generated from a unique quasi-random sequence of size N and $2d$ dimensions, which are then split into two sets of size N and d dimensions each.

In this work, the MATLAB function `sobolset` is used to generate such a low-discrepancy Sobol sequence and before it used the nodes are scrambled by using the `scramble` function with the *'MatousekAffineOwen'* scramble type (see Matousek, 1998).

For further details on sampling-based methods for the computation of variance-based sensitivity measures, the reader is referred to Andrea Saltelli et al. (2004, p. 124 ff.). For more information on low-discrepancy sequences and QMC methods the reader is referred to the literature on the subject, such as Niederreiter (1988); Niederreiter (1992); Sobol (2001) or Andrea Saltelli et al. (2004, p. 197 ff.).

3.3 Confidence Intervals Using Bootstrapping

When an unknown statistic or parameter $\theta \in \Theta$ of a population is to be estimated, so-called *interval estimates* can be used (in contrast to *point estimates*). A $\gamma \cdot 100\%$ *confidence interval* for θ is such an interval estimate defined by two statistics $T_l = h_l(\{X\}_{i=0}^N)$ and $T_u = h_u(\{X\}_{i=0}^N)$ based on a random sample $\{X\}_{i=0}^N$, which fulfills, according to Held and Bové (2014, p. 55 ff.):

$$\Pr\left(T_l \leq \theta \leq T_u\right) = \gamma, \qquad \forall \theta \in \Theta \tag{3.28}$$

The statistics T_l and T_u are called the limits of the confidence interval, and it is assumed that $T_l \leq T_u$. Note that the limits are functions of the random sample $\{X\}_{i=0}^N$ and are therefore also random variables, whilst the unknown parameter θ is fixed.

According to Held and Bové (2014, p. 55 ff.), if many identical repetitions of the sampling experiment are carried out, then a $\gamma \cdot 100\%$ confidence interval will cover the unknown parameter θ in $\gamma \cdot 100\%$ of the cases.

Confidence intervals will be computed later on in section 4.4.2.3 for the validation of the moments m_k of the model outputs of interest for the corresponding gPC expansion. For the computation of the confidence intervals the bootstrapping method, first introduced by Efron (1979) and Efron and R. Tibshirani (1986)), will be used. Therefore the bootstrapping method will be presented in what follows.

Bootstrapping belongs to the class of so-called re-sampling methods. As mentioned before, consider the sought parameter of the population to be $\theta = \theta(F)$ with unknown distribution

F and consider that n samples $x_1, \ldots, x_n$, realizations from identically distributed random variables $X_1, \ldots, X_n$ (with distribution $X_i \sim F, \quad \forall i \in \{1, \ldots, n\}$), are available. Based on the observed samples, the parameter θ can be estimated by $\hat{\theta} = \hat{\theta}(x_1, \ldots, x_n)$. The distribution of the sought parameter which is a function of the random variables $X_1, \ldots, X_n$: $\hat{\theta} = \hat{\theta}(X_1, \ldots, X_n|F)$ is also unknown, since F is unknown.

The bootstrapping method comprises two basic steps. In the first step, the real unknown distribution F is replaced by the empirical distribution $\hat{F}_n$, which can be defined according to Engel and Grübel (2008, p. 8) as:

$$\hat{F}_n(x) = \frac{1}{n}\#\{x_i|x_i \leq x\} \tag{3.29}$$

That is, the real unknown Cumulative Distribution Function (CDF) is replaced by the cumulative frequency (where $\#\{U\}$ represents the cardinal number of the set U), the percentage of observations that lie below x. Consequently, the distribution of $\hat{\theta}(X_1, \ldots, X_n|F)$ can be estimated by $\hat{\theta}(X_1, \ldots, X_n|\hat{F}_n)$, known as the *bootstrap-distribution*.

The second step is a MCS, a sample of size n is drawn *with replacement* according to $\hat{F}_n$: $x_1^*, \ldots, x_n^*$, the *bootstrap-sample*, which is actually a re-sample of the original sample. The unknown parameter is then computed from it as $\hat{\theta}(x_1^*, \ldots, x_n^*)$. If this process is repeated several times, for example B times, an empirical approximation to the bootstrap-distribution $\hat{\theta}(X_1, \ldots, X_n|\hat{F}_n)$ can be obtained.

For the computation of confidence intervals by bootstrapping, the MATLAB function `bootci` will be used, and in particular, the *bias-corrected percentile method* will be used for its accuracy and relatively low computation cost. For further information the reader is referred to Efron and R. J. Tibshirani (1993) and other literature on the subject.

4 NUMERICAL RESULTS

In this chapter, the numerical results obtained using the aforementioned methods will be presented and described in detail. First, the selection of an appropriate UQ toolkit is discussed, followed by a brief description of the modifications, additions and improvements implemented within the chosen toolkit. Next, the results obtained by using common test functions are presented and discussed. The different non-intrusive methods for the computation of the gPC expansion coefficients and the resulting first order and total Sobol' indices are compared in terms of accuracy and computational cost. Convergence of important statistics such as mean and standard deviation and the Sobol' indices of the output variables of interest is validated against sampling-based alternatives, such as MC or QMC simulation. The same structure is used to analyze the results gained by the Single-Track Model (STM). Finally, the Python-MATLAB-DYNA4 framework used for the Two-Track Model (TTM), derived from previous works on related topics and adapted to fit the purposes of SA, will be presented along with the obtained results.

4.1 Selection of an Uncertainty Quantification Toolkit

A number of toolkits for UQ are freely available implemented in different programming languages. For that reason, a completely new package is not developed in this work, but an existing toolkit is chosen instead and modified as required.

For convenience and compatibility reasons with DYNA4, a MATLAB toolkit is chosen in this work. In particular, the MATLAB version of Uncertainty Quantification Toolkit (UQTk) developed by Sandia National Laboratories (B. Debusschere, Sargsyan, & Safta, 2014) will be used. Although other available packages such as the Design Analysis Kit for Optimization and Terascale Applications (DAKOTA) toolkit (written in C++) might be better in terms of performance (due to the compiled nature of C++, as opposed to interpreted programming languages such as MATLAB), UQTk's MATLAB version can be more easily modified and adapted for the purposes of this work and thus greater emphasis may be laid upon its application.

Having presented the reasons for the choice of the toolkit that will be used within the present work, the modifications carried out on UQTk will be detailed in what follows.

4.2 Modifications to the Chosen Toolkit

The UQTk provides non-intrusive algorithms for gPC and some other useful algorithms for UQ, all of which are listed in table A.1 in appendix A. The modifications performed on UQTk (all of which shall be found in table A.2) are basically focused on improving the NISP methods already available in the toolkit.

Firstly, the LRM was implemented in `regressionPCE.m`, this function computes the necessary matrices, i.e.: $\Psi_{ij} = \Psi_j(X^i)$ and the information matrix $A_{ij} = \boldsymbol{\Psi}^T \boldsymbol{\Psi} = \Psi_{ji}\Psi_{ij}$, as well as the design of experiment X_{ij} (the sample nodes necessary to compute the linear regression) and its size N, which is unknown before the information matrix is assembled.

Secondly, the original code by F. Heiss and V. Winschel (2015) was adapted for its use with UQTk in order to obtain the necessary *quadrature* structure containing the nodes and weights for the SGI method. This allows for the use of Gaussian quadrature rules and of Kronrod-Patterson nested quadrature rules (as already shown in section 3.1.3.2).

Additionally a generalization of UQTk's `uq_getSensitivity.m` was implemented in `sobolIndex.m`, which computes the sth-order Sobol' sensitivity indices for an appropriate multi-index (a vector of single indices) of dimension s.

Further methods for the sampling of the PCE at a given instant and for the estimation and visualization of the PDF were implemented in `samplePCE.m` and `plotPDF.m`, respectively.

Finally, the test functions and the STM that will be presented in the next sections, as well as some useful functions used within the Python-MATLAB-DYNA4 framework were also implemented in MATLAB. They will only be briefly described here though, in the interest of conciseness and clarity.

For each of the test functions, the analytical results for the first order and total Sobol' sensitivity indices are either computed with a MATLAB function or hard-coded into a vector such that they can be compared later on by means of two benchmark functions, one that computes the aforementioned indices from a PCE, whose coefficients are computed using the LRM and another one that computes the Sobol' indices by using SGI for the computation of the PCE coefficients. Both methods are compared using a fixed order for the PCE in this case. Additionally, for both the LRM and SGI separately, the results are also compared against different orders of the PCE. In the case of the Oakley & O'Hagan function, the model presents 15 stochastic dimensions and therefore the comparison for different orders of the PCE is not feasible, since the LRM must first compute the full tensor of nodes and weights for the integration (despite the fact that not all of them are used). That means that even for a 2nd-order PCE ($N = 2$) the number of nodes is already $(N+1)^d = 3^{15} = 14348907$. In the next sections many other scripts and functions were implemented for the sole purpose of results visualization and thus these will not be detailed here.

The following sections present the actual results for the applications described so far, starting with the test functions, followed by the STM and concluding with the TTM.

4.3 Test Functions

In the field of UQ there are some typical test functions which possess known analytical solutions for different statistics and are thus useful for the validation of numerical methods and approximations. In the present work, the G-function and the Oakley & O'Hagan function will be used for that purpose, and these are consequently detailed in what follows.

4.3.1 *G-function*

The G-function, commonly used in UQ, is defined according to A. Saltelli et al. (2010, p. 264) as:

$$f(\boldsymbol{x}) = \prod_{i=1}^{d} \frac{|4x_i - 2| + a_i}{1 + a_i} \tag{4.1}$$

For Uncertainty Quantification (UQ), the input random variables are independent and identically distributed, following a uniform distribution: $x_i \sim \mathcal{U}(0,1), \quad i = 1, \ldots, d$. A lower value of a_i means a higher importance of the input random variable x_i.

In this work, the same procedure followed by Sudret (2008, p. 973 ff.) is used, thus the vector of a_i's is described by: $\boldsymbol{a} = [1, 2, 5, 10, 20, 50, 100, 500]$, so that initially the number of dimensions is 8, although according to the results shown by Sudret, only the first 3 or 4 parameters have a significant influence on the output variance and thus an approximate model using the first 4 parameters is actually used.

Figure 4.1 presents a comparison of the different methods for the computation of the gPC coefficients for a PCE of 7th-order in terms of the resulting first order and total Sobol' sensitivity indices.

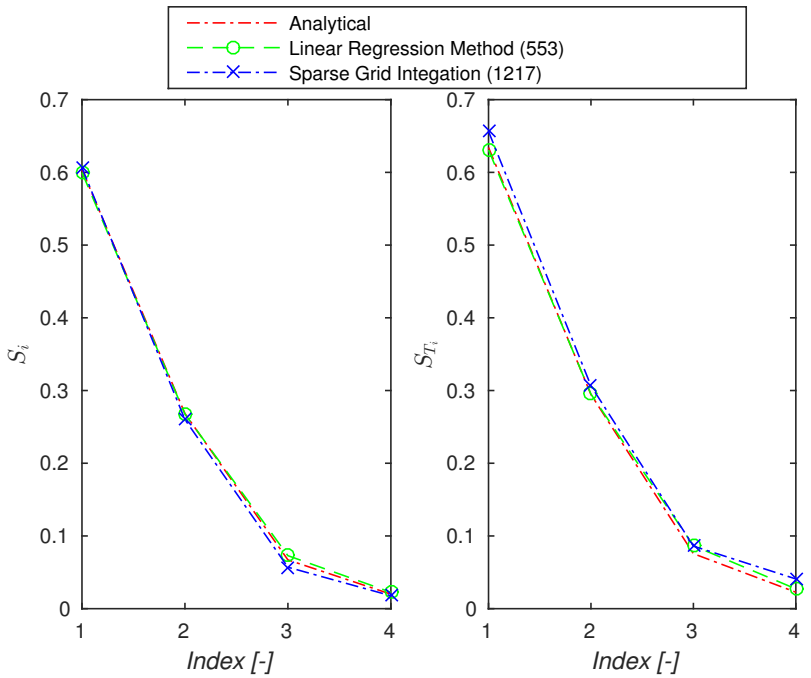


Figure 4.1: First order (left) and total (right) Sobol' sensitivity indices for the G-function with a PCE of 7th-order using different methods for the computation of the coefficients, no. of model evaluations in parentheses

As can be observed the differences in the first order and total Sobol' indices computed by means of a linear regression method and sparse grid integration are negligible in practice and so is the error of both methods compared to the analytical solutions. The analytical solution is obtained according to A. Saltelli et al. (2010, Appendix A):

$$V_i = \frac{1/3}{(1+a_i)^2}, \quad i \in \{1, 2, \ldots, d\} \tag{4.2}$$

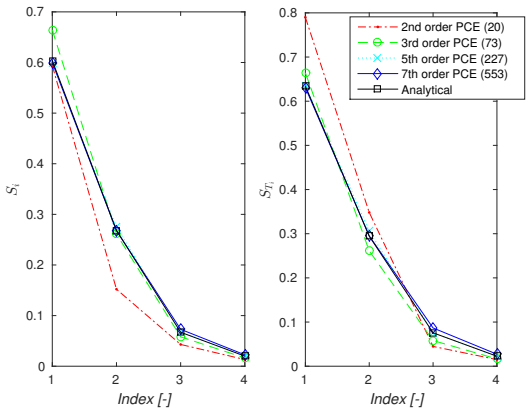
$$V_{T_i} = V_i \prod_{j \neq i} (1 + V_j), \quad i \in \{1, 2, \ldots, d\} \tag{4.3}$$

$$V = \prod_{i=1}^{d} (1 + V_i) - 1 \tag{4.4}$$

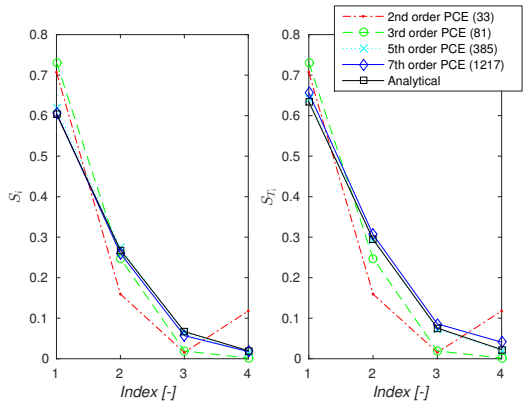
$$S_i = \frac{V_i}{V}, \quad i \in \{1, 2, \ldots, d\} \tag{4.5}$$

$$S_{T_i} = \frac{V_{T_i}}{V}, \quad i \in \{1, 2, \ldots, d\} \tag{4.6}$$

Figure 4.2a displays a benchmark comparing the LRM for the computation of the gPC coefficients and the resulting first order and total Sobol' indices for the G-function for different orders of the PCE to their analytical counterpart.



(a) LRM used for the computation of the PCE coefficients



(b) SGI used for the computation of the PCE coefficients

Figure 4.2: First order (left images) and total (right images) Sobol' sensitivity indices for the G-Function using different PCEs of increasing order and different methods for the computation of the PCE coefficients, no. of model evaluations in parentheses

In addition, figure 4.2b shows a benchmark comparing the SGI method for the computation of the gPC coefficients and the resulting first order and total Sobol' indices for the G-function for different orders of the PCE to their analytical counterparts.

As can be seen, increasing the order of the PCE achieves a higher accuracy in the approximation of the Sobol' indices, converging to the analytical solution, at the expense of an increased computational cost.

4.3.2 *Oakley & O'Hagan function*

The Oakley & O'Hagan function, as first introduced in Oakley and O'Hagan (2004), is given by:

$$f(\boldsymbol{x}) = \boldsymbol{a_1}^T \boldsymbol{x} + \boldsymbol{a_2}^T \sin(\boldsymbol{x}) + \boldsymbol{a_3}^T \cos(\boldsymbol{x}) + \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} \tag{4.7}$$

Where the input vector $\boldsymbol{x} = (x_1, \ldots, x_{15})$ has 15 dimensions and the $\boldsymbol{a}$-coefficients are chosen such that 5 of the input variables contribute significantly to the output variance, 5 have a much smaller effect and the remaining 5 have almost no effect on the output variance. The input random variables are independent and identically distributed and they follow a normal distribution with parameters $x_i \sim \mathcal{N}(\mu = 0, \sigma^2 = 1), \qquad \forall i = 1, \ldots, 15$. For the values of the coefficient vectors $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$, and the matrix $\boldsymbol{M}$ used in this work see Oakley (2015).

Figure 4.3 shows a benchmark comparing different methods for the computation of the gPC coefficients and the resulting total Sobol' indices for the Oakley & O'Hagan function. The analytical values are obtained from Campolongo, Cariboni, Saltelli, and Schoutens (2005, p. 373).

As mentioned before, only a 2nd-order PCE was used in this case due to the high number of stochastic dimensions of this model (15), which limits the use of higher order PCEs, specially for the linear regression method, since the full tensor must be computed in the first place, thus requiring the computation of $(N+1)^d = 3^{15} = 14348907$ nodes and their corresponding weights, while a 3rd-order PCE would already require the computation of $4^{15} = 1073741824$ nodes (and again their corresponding weights), thus making the comparison unfeasible for an order $N > 2$.

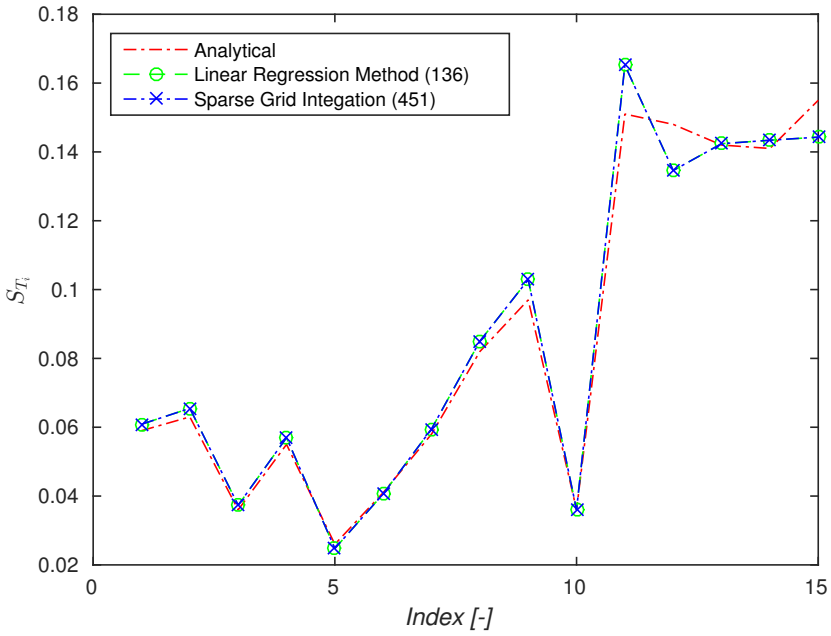


Figure 4.3: Comparison of Total Sobol' sensitivity indices for the Oakley & O'Hagan function computed analytically and by using a PCE with LRM and SGI, no. of model evaluations in parentheses

As can be observed, the results are considerably accurate, with the maximum relative error being approximately 8% (and the relative error of most of the indices being under 5%). Recall that a 2nd-order PCE was used in both cases and thus the accuracy could theoretically be

improved by increasing the order of the PCE. The results obtained by using the linear regression method and those achieved by using sparse grid integration are identical and thus there is no practical difference in using one method or the other in this case.

Finally, note that although the number of model of evaluations is always lower for the LRM (compared to SGI) for all the investigated test functions, its computation time is actually higher (as will be shown in section 4.4.2.1) since the LRM requires the prior computation of the tensor product and the sorting of all the nodes by increasing norm (as according to the assembly algorithm presented in Sudret (2008)).

4.4 Single-track Model

After testing and validating the methods implemented in this work with the aforementioned test functions, they will be applied to a more complex model in order to test their robustness and reliability and to exemplify their usage in a practical case. A vehicle dynamics model will be used, in particular, a transient linear Single-Track Model (STM) implemented in MATLAB. Next, the mathematical model will be presented for completeness and later on the results obtained thereof will be detailed.

4.4.1 *Mathematical model*

The transient linear STM that will be used in this work, which was originally implemented in MATLAB for the lecture *Dynamik der Straßenfahrzeuge* (see Lienkamp, 2015), is based on the work by Riekert and Schunk (1940), who developed a simplified vehicle dynamics model consisting of a front- and a rear axle with only one wheel each, whereby more complex dynamic effects such as rolling and pitching (and thus any load transfer between front- and rear axle) are neglected, consequently reducing the physical phenomenon to three degrees of freedom: two translations and one rotation.

As discussed in Stadler (2014–2015, p. 36 ff.), Isermann (2006, p. 57 ff.) devised the following state space model for the transient linear STM:

$$\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \boldsymbol{y}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{C} & \boldsymbol{d} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ u(t) \end{bmatrix} \tag{4.8}$$

$$\underbrace{\begin{bmatrix} \dot{\beta} \\ \ddot{\psi} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -\frac{c_{\alpha_v}+c_{\alpha_h}+m\dot{v}}{mv} & \frac{c_{\alpha_h}l_h-c_{\alpha_v}l_v}{mv^2}-1 \\ \frac{c_{\alpha_h}l_h-c_{\alpha_v}l_v}{J_{zz}} & -\frac{c_{\alpha_h}l_h^2+c_{\alpha_v}l_v^2}{J_{zz}v} \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \frac{c_{\alpha_v}}{mv} \\ \frac{c_{\alpha_v}l_v}{J_{zz}} \end{bmatrix}}_{\boldsymbol{b}} \underbrace{\delta}_{u} \tag{4.9}$$

$$\underbrace{\begin{bmatrix} a_y \\ \beta \\ \dot{\psi} \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} -\frac{c_{\alpha_v}+c_{\alpha_h}+m\dot{v}}{m} & \frac{c_{\alpha_h}l_h-c_{\alpha_v}l_v}{mv} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{C}} \underbrace{\begin{bmatrix} \beta \\ \dot{\psi} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \frac{c_{\alpha_v}}{m} \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{d}} \underbrace{\delta}_{u} \tag{4.10}$$

This coupled system of differential equations 4.9 represents the relations between sideslip angle β, yaw rate $\dot{\psi}$ and the front wheel steering angle δ and allows to additionally compute

the lateral acceleration by means of equation (4.10). Thanks to the state space representation the transient linear STM can be solved over time using MATLAB's `lsim` function.

In the following, the results obtained with the discussed model will be presented.

4.4.2 *Step steering input*

The STM and the resulting statistics and Sobol' indices of the model outputs are tested by using 7 uncertain parameters. All parameters are assumed independent and follow normal distributions ($Z_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \quad i = 1, 2, \ldots, 7$), whose corresponding means and standard deviations are summarized in table 4.1.

Parameter description	Symbol	Units	Mean (μ)	Std. dev. (σ)
Distance front axle-center of gravity	l_v	[m]	1.45	0.0725
Distance rear axle-center of gravity	l_h	[m]	1.45	0.0725
Total mass	m	[kg]	2250	112.5
Wheel cornering stiffness front axle	c_v	[N/rad]	60000	3000
Wheel cornering stiffness rear axle	c_h	[N/rad]	75000	3750
Yaw moment of inertia	J_{zz}	[kg m^2]	3000	150
Vehicle speed in x-direction	v_x	[m/s] ([km/h])	27.78 (100)	1.389 (5)

Table 4.1: Uncertain parameters and their distribution for the STM study

The parameters are chosen as discussed in Stadler (2014–2015), so that the behavior of the vehicle remains stable, which enables a better visualization of the results and an easier understanding of the underlying phenomena. Note that the parameters can be freely chosen (more or less), depending on the specifications of the automotive manufacturer. Here a rough specification is represented, by choosing the standard deviation of each parameter to be 5% of their corresponding mean. The input parameters actually correspond with a small Sports or Suburban Utility Vehicle (SUV), as according to Stadler (2014–2015).

As shown in the previous sections, differences in the accuracy of the different methods used to compute the spectral coefficients are, in general, negligible, if the model output is smooth enough. Consequently, first an analysis of the computation time required by each of these methods is carried out in order to choose the best one in terms of performance. Later on, convergence plots for the Sobol' indices of the different model outputs (lateral acceleration a_y, side slip angle β and yaw-rate $\dot{\psi}$) and their statistics will be analyzed.

But first of all, the time evolution of the STM outputs is shown in figure 4.4. The mean (μ) of each variable over time is represented with a solid line, whilst the mean plus-minus two standard deviations ($\mu \pm 2 \cdot \sigma$) are plotted with dashed lines.

As it can be observed, the response of the STM is computed for a step steering input maneuver of $\delta = 5°$. The step steering input takes place at $t = 1\ s$ and is held constant thereafter.

4.4.2.1 Performance analysis

In figure 4.5 the computation time is compared for the different approaches to computing the spectral coefficients, i.e. the tensor product rule, the linear regression method and sparse grid integration. It then becomes clear that sparse grid integration outperforms any of the

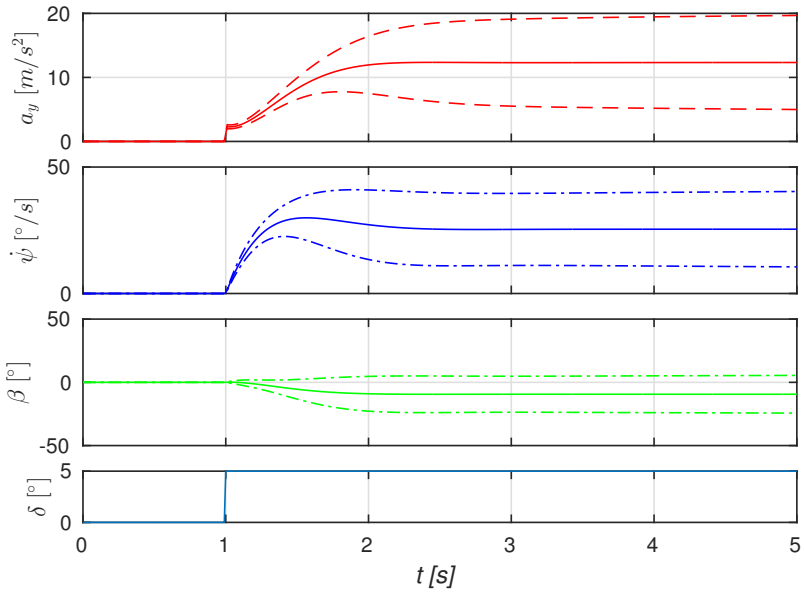


Figure 4.4: Time evolution of the STM outputs, computed with a 6th-order PCE

other two methods, specially as the order of the PCE expansion increases. The tensor product rule becomes unfeasible as soon the order increases too much, due to the curse of dimensionality. As can be seen in the figure, for the 4th-order PCE the tensor product rule is almost 100 times slower than sparse grid integration. The linear regression method also performs good, or at least better than the tensor product rule, but it is still slower than sparse grid integration as soon as the order of the PCE increases.

Therefore it is important to note that, from now on, all of the computations in the present work will be based on sparse grid integration, being cautious with the results, which shall be always validated.

Figure 4.6 shows the computation time required for the determination of the spectral coefficients (by means of SGI) versus the number of model evaluations, which increases with the order of the PCE being used (in the figure 1st to 6th-order PCE are represented). As can be observed, the computation time is linear with the number of model evaluations performed.

Nonetheless, the number of model evaluations increases exponentially with the order of the PCE and thus the computation time increases logarithmically with the order of the PCE, as can be seen in figure 4.7 (note the logarithmic scale of the y-axis).

An additional image of the computation time depending on the number of PCE terms to be computed is shown in appendix B. It is noted that the computation time increases with the number of PCE terms to be computed according to $P = \binom{N+7}{N}$ for $N = 1, \ldots, 6$ (the order of the PCE), since the stochastic dimension of the PCE (the number of uncertain model input parameters) is $d = 7$.

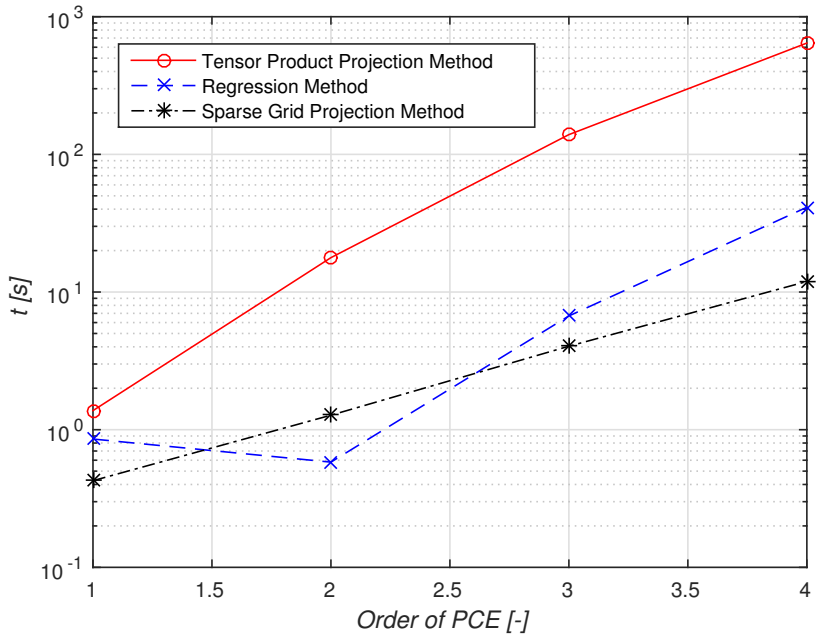


Figure 4.5: Computation time vs. Order of the PCE for the STM, with different methods for the computation of the spectral coefficients

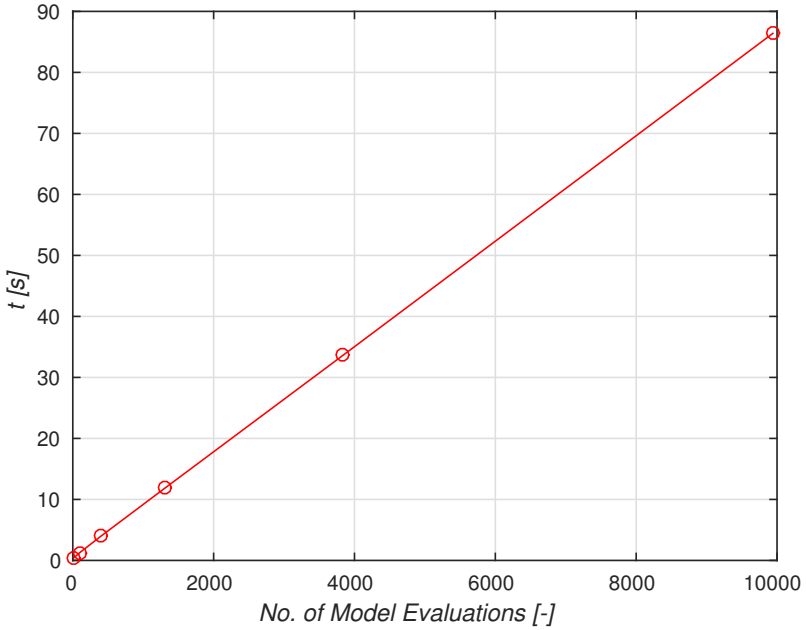


Figure 4.6: Computation time vs. Number of model evaluations for the STM using SGI (Kronrod-Patterson rule) for the computation of the PCE coefficients

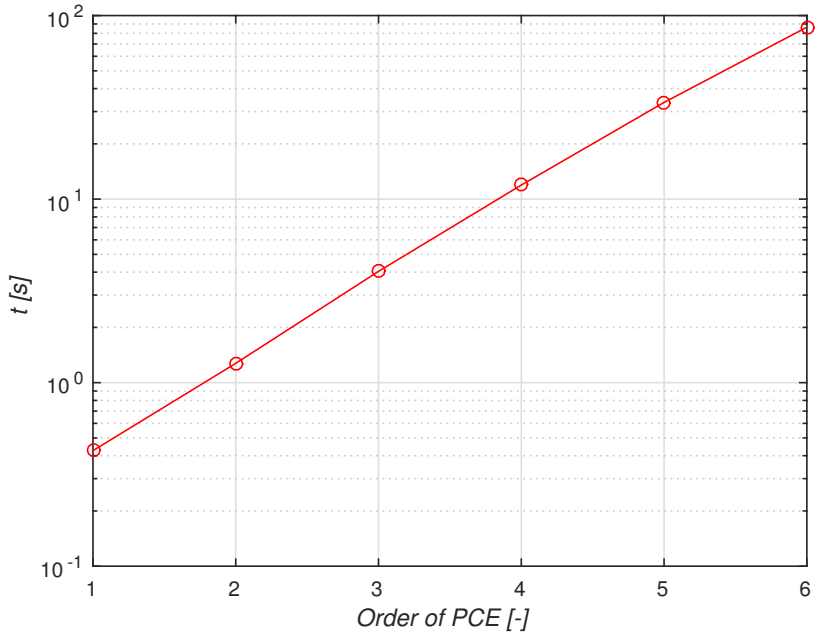


Figure 4.7: Computation time vs. Order of the PCE for the STM using SGI (Kronrod-Patterson rule) for the computation of the PCE coefficients

4.4.2.2 Convergence analysis

Next, convergence plots for the first order and total Sobol' indices of each STM output at $t = 4\ s$, as well as for their sums will be presented in figures 4.8, 4.9 and 4.10. It can be observed, that all of the aforementioned indices converge as the order of the PCE increases. In particular, for most of them, a PCE of 3th-order is enough to achieve a low error and increasing the order of the PCE does not improve the accuracy much more, so that the increase in computation time is not worth the small improvement in accuracy, in this case.

Furthermore, convergence plots for the mean value and standard deviation of the STM outputs themselves, can be found in figure 4.11. In this case, a 4rd-order PCE suffices in order to achieve convergence of the aforementioned statistics.

However, the convergence of the investigated variables does not imply that they converge to the correct value. Consequently, convergence of the mean value and the standard deviation of the output variables is tested against a QMC sampling-based method here, and the same procedure will be followed afterwards for the validation of the Sobol' indices.

Figure 4.12 shows this comparison for the lateral acceleration (a_y) at $t = 2.5\ s$. Further images on this topic can be found in appendix B.

As it can be observed, again a 3rd-order PCE already converges to the sampling-based values and even a 2nd-order PCE already offers a good accuracy for these statistics, the PC-based method requiring much fewer model evaluations and thus offering a good alternative in terms of performance.

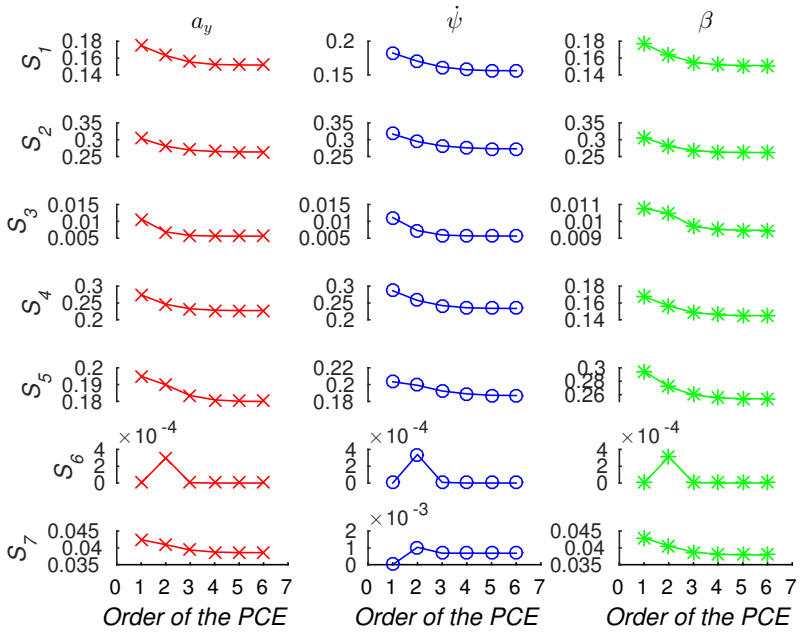


Figure 4.8: Convergence of the first order Sobol' indices for different model outputs of the STM at $t = 4\ s$ with respect to the order of the PCE

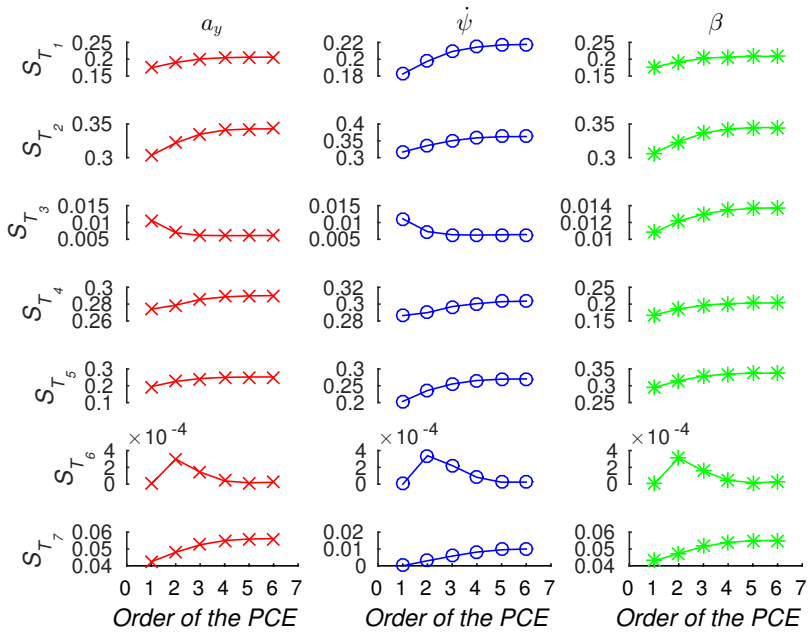


Figure 4.9: Convergence of the total Sobol' indices for different model outputs of the STM at $t = 4\ s$ with respect to the order of the PCE

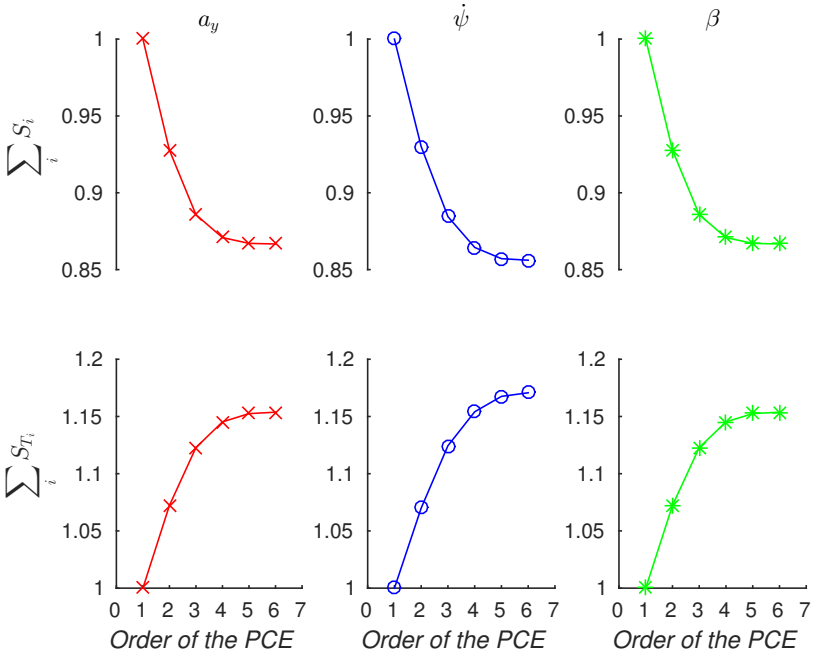


Figure 4.10: Convergence of the sum of first order and total Sobol' indices for different model outputs of the STM at $t = 4$ s with respect to the order of the PCE

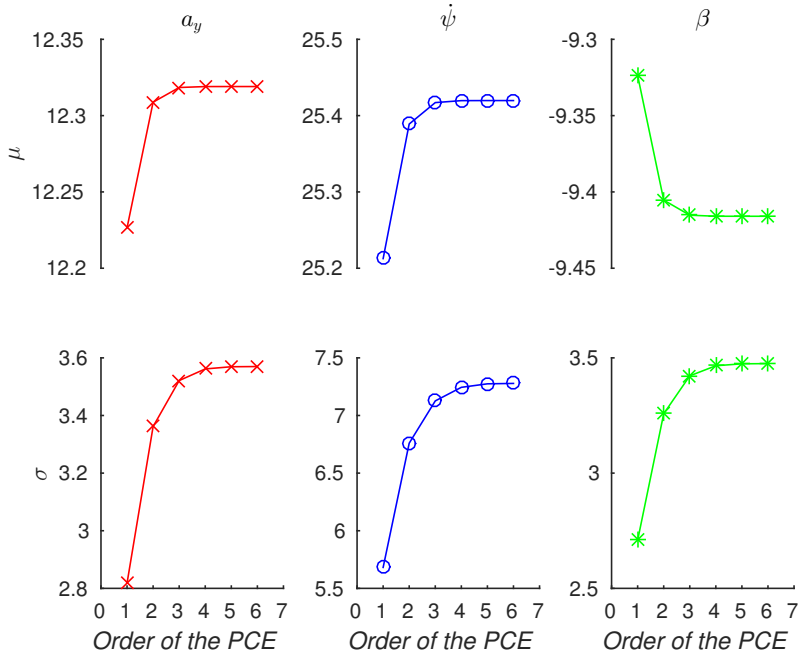


Figure 4.11: Convergence of the mean and standard deviation for different model outputs of the STM at $t = 4$ s with respect to the order of the PCE

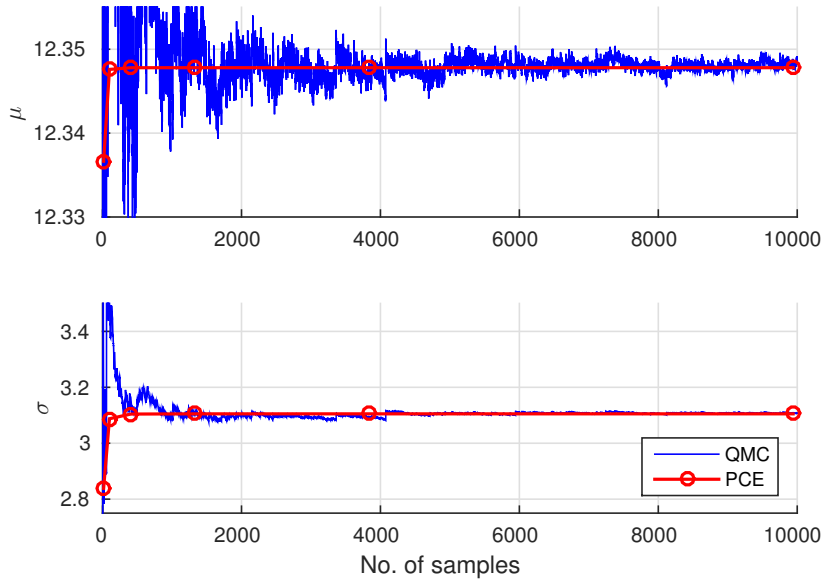


Figure 4.12: PCE vs. QMC, convergence of the mean and standard deviation for a_y at $t = 2.5\ s$

4.4.2.3 Confidence intervals of the moments and KDE of the model outputs

The convergence of the PC-based mean and the standard deviation is actually guaranteed by the Cameron-Martin theorem (Cameron & Martin, 1947), and although convergence of higher-order moments is not guaranteed at all, it is interesting to consider their (95%) confidence intervals, computed here by means of a bootstrapping strategy.

The bootstrapping is carried out here by sampling the 6th-order PCEs of the model outputs. The confidence intervals for the yaw rate ($\dot{\psi}$) can be found in figure 4.13, being computed at $t = 2.5\ s$. The corresponding KDEs can be found in figure 4.14. For additional images the reader is referred to appendix B.

As can be observed in those figures, confidence intervals for the first two moments (i.e. mean and standard deviation), are imperceptible, practically coincident with the value of the moments themselves. The confidence intervals for the 3rd-, 4th- and 5th-order moments are relatively narrow, but scatter as the order increases (compare with 6th-order moments), since their convergence is not strictly guaranteed.

Furthermore, the kernel density estimates may be compared with the moments for each of the variables. Qualitatively, both the mean and the standard deviation coincide with the values seen in the KDE. Finally, the skewness (given by the 3rd-order moment) is positive for the yaw rate, and this statement can be confirmed by looking at the KDE of the corresponding variable: the yaw rate has a positive skew, i.e. it is right-skewed.

Finally, it is noted that in figure 4.14 the KDE of the 6th-order PC-based outputs perfectly match the KDE of the MC-based output, computed using 200000 samples/model evaluations. The PC-based method using only 9941 model evaluations in comparison.

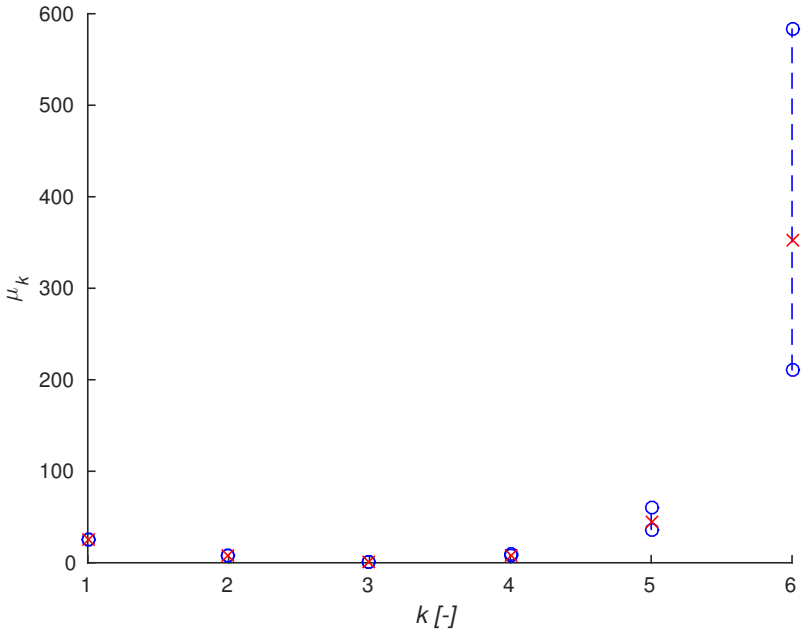


Figure 4.13: Confidence Intervals for the moments of $\dot{\psi}$ at $t = 2.5\ s$ for the STM

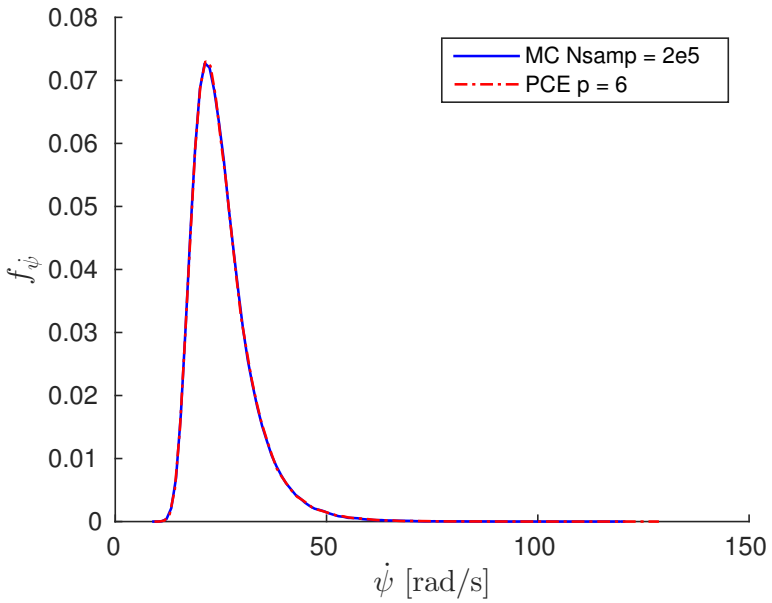


Figure 4.14: KDE of $\dot{\psi}$ at $t = 2.5\ s$ for the STM

4.4.2.4 PC-based vs. sampling-based Sobol' indices

To finish this section, the PC-based (first order and total) Sobol' indices are compared to the ones obtained by using a QMC sampling-based approach.

Figure 4.15 shows this comparison for the first order Sobol' indices for the lateral acceleration (a_y), whilst figure 4.16 does the same for the total Sobol' indices. In both cases, 7 indices are computed, each one corresponding to a different uncertain parameter (the order follows the one presented in table 4.1, e.g. parameter number 3 corresponds to the total mass).

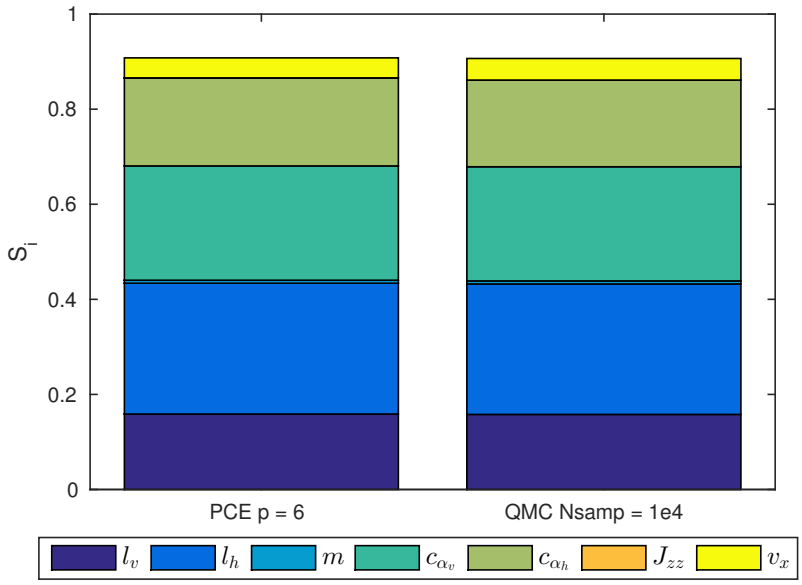


Figure 4.15: Comparison of PC-based and sampling-based first order Sobol' indices of a_y at $t = 3\ s$ for the STM

As shown, in both figures, the relative error for all of the computed indices is lower than 1%, and thus negligible in practice. Furthermore, the QMC sampling-based method (using a Sobol sequence as described in section 3.2.2) was computed with 10000 samples, and thus $N(k+2) = 10000(7+2) = 90000$ model evaluations were required for the computation of the Sobol' indices, while the PC-based method of 6th-order (computed by means of sparse grid integration) requires only 9941 model evaluations, that is, approximately 11.05% as many.

These results thus support the idea that a PCE of an appropriate order using sparse grid integration provides sufficient accuracy not only in the time evolution of the output variables themselves, but also in derived measures such as Sobol' indices and are a good alternative to sampling-based methods in terms of performance and number of model evaluations.

Finally, the convergence of the Sobol' indices in terms of the number of model evaluations is exemplified by computing the 2nd total Sobol' index (corresponding to l_h) of each model output computed at $t = 2.5\ s$. Figure 4.17 compares the convergence of the PC-based method with that of the QMC sampling-based method.

As can be observed in the figure, a 3rd-order PCE already converges to the same value as the QMC sampling-based method with as little as 407 model evaluations, compared to approx. 60000 needed for the convergence of the QMC method.

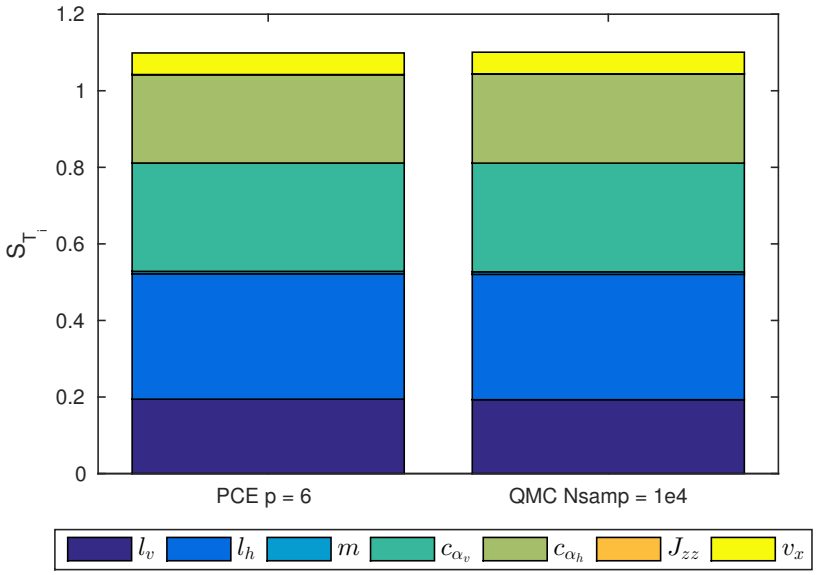


Figure 4.16: Comparison of PC-based and sampling-based total Sobol' indices of a_y at $t = 3\ s$ for the STM

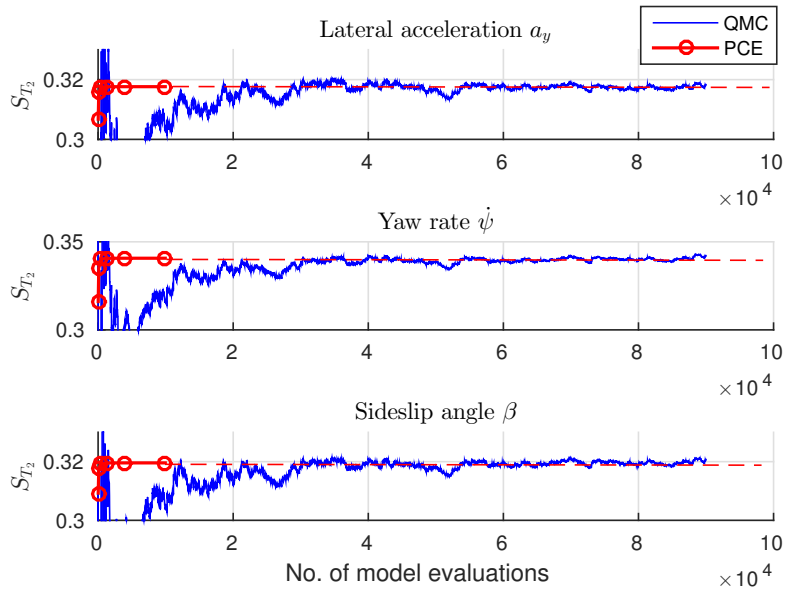


Figure 4.17: PCE vs. QMC convergence of the 2nd total Sobol' index at $t = 2.5\ s$ for the different model outputs

4.4.2.5 Time evolution of the Sobol' indices

Finally, the time evolution of the PC-based first order and total Sobol' indices of the yaw rate ($\dot{\psi}$) for the STM are presented in figure 4.18. Additionally, figure 4.19 shows their QMC sampling-based counterparts.

These stacked area graphs allow for the simultaneous graphical representation of the 7 indices, as well as their cumulated values, over time.

As discussed previously, the Sobol' indices enable us to quantify the influence of the uncertainty in a model parameter on the uncertainty of a model output, and in this case we can observe this influence over time. In both figures two clearly distinct regions can be distinguished, a transient region (from $t = 1\ s$ to $t = 2\ s$ approx.) and a stationary state region (starting at $t = 2\ s$ approximately), corresponding to their analogous regions of the outputs themselves (see figure 4.4). In the stationary region, the relative contribution of each parameter to the variance in the output does not change.

The trends shown by the PC-based Sobol' indices coincide with those which one would intuitively expect. For example, the yaw moment of inertia (J_{zz}) has great influence (almost 33%) on the yaw rate ($\dot{\psi}$) right at the beginning of the transient, and its influence quickly vanishes, as the yaw angular acceleration ($\ddot{\psi}$) vanishes, corresponding to a constant yaw rate ($\dot{\psi}$). The cornering stiffness of the front wheels (c_{α_v}) also has a great impact on the yaw rate at the beginning of the transient, this can be explained by the fact that the front wheels are the ones used for the steering of the vehicle (in this case) and so affect the angular response of the vehicle right after the step steering maneuver takes place.

For further images on this subject the reader is referred to appendix B.

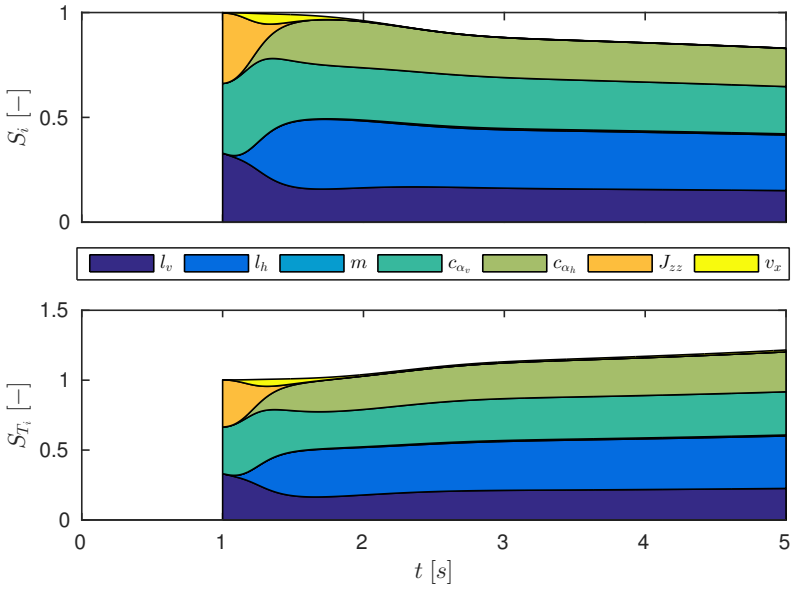


Figure 4.18: PC-based first order and total Sobol' indices of $\dot{\psi}$ for the STM

As can be observed by comparing the PC-based Sobol' indices to their QMC sampling-based counterparts, even their time evolution match very accurately. That is, the PCE method implemented in this work and the Sobol' indices derived therefrom match the results obtained with other sampling-based techniques (such as QMC) with a much lower compu-

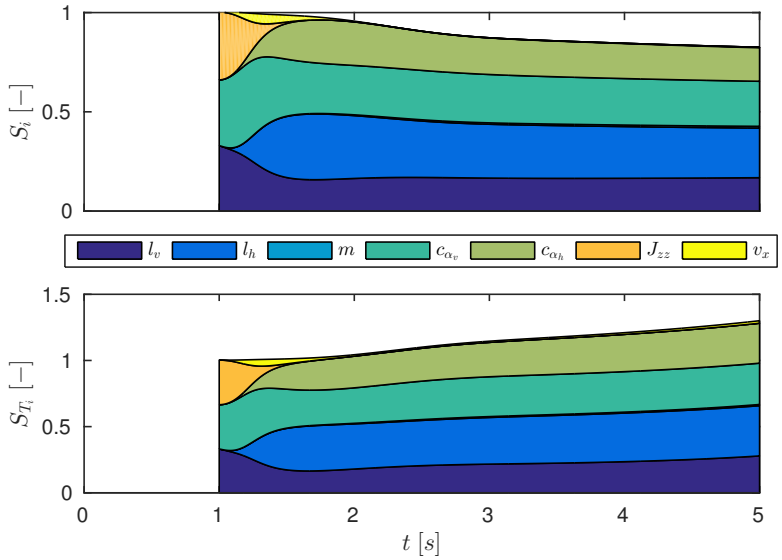


Figure 4.19: Sampling-based first order and total Sobol' indices of $\dot{\psi}$ for the STM

tational cost when the order of the PCE is appropriately chosen. Usually, a 4th-order PCE is accurate enough for these output variables and this model, as experience has shown.

4.5 Two-track Model

In this section, the results obtained by using a TTM as the model for vehicle dynamics will be presented. The TTM that will be used is built using TESIS DYNAware DYNA4, a modular Simulink-based simulation framework that contains several models for the automotive development process. DYNA4's Car Professional offers real-time vehicle dynamics simulation, suitable for suspension concept studies and vehicle dynamics controller development. The vehicle model is again a modular multi-body system based on components, containing different multi-body axle models including axle geometry via hardpoint definition or tabled axle kinematics, various steering and drivetrain configurations and the TM-Easy and Pacejka96 tire models. Furthermore, the braking system can be implemented by means of a brake pressure distribution or using a hydraulic brake system model. Car Professional also offers different drive train configurations, such as manual or automatic gearboxes and front-, rear- or all-wheel drive. It also incorporates different Electronic Control Unit (ECU) models, as well as a 3D road model including road geometry and road surface properties. Finally, open-loop as well as closed-loop maneuver controls and a catalog of typical vehicle dynamics tests (e.g. ISO tests) are also provided in the framework. DYNA4 allows one to easily access and edit all of these modules, making full customization of the vehicle properties, test scenarios and maneuvers possible.

Furthermore, the selection of the DYNA4 framework is motivated by its usage in previous works on related subjects. For example, the work by C. Bartsch on sampling methods for uncertainty quantification in vehicle dynamics, see Bartsch (2014–2015). In that work, a pre-processing tool written in Python and called UPSim was developed and implemented which allows for the Monte Carlo (MC) simulation of vehicle dynamics with DYNA4 by using its XML-Remote Procedure Call (RPC) Application Program Interface (API). Using UPSim's

Graphical User Interface (GUI) a DYNA4 project may be chosen and loaded and the uncertain parameters and their distributions may be selected, finally a number of MC simulations can be easily run at the end of the configuration process.

4.5.1 *Python-MATLAB-DYNA4 framework*

DYNA4 provides access to its MATLAB engine through its aforementioned XML-RPC API, nonetheless, the API does not work as one would expect for all MATLAB functions, e.g., it does not support functions returning structures or cells. These limitations were recognized by TESIS DYNAware when requesting support for some related problems that occurred while implementing the UQ capability into our framework. Consequently, there was a need for a simple solution that would allow for the implementation of general MATLAB functions and libraries. In figure 4.20 the proposed solution for the design of the framework is represented.

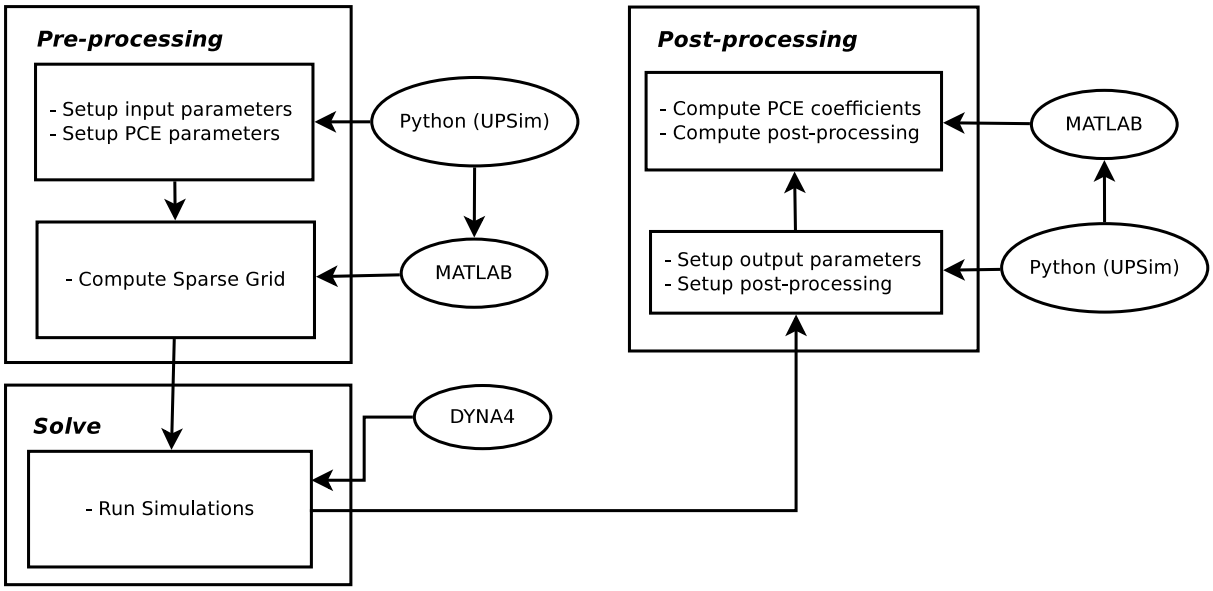


Figure 4.20: Python-MATLAB-DYNA4 Framework

Luckily, MATLAB provides an API which allows running a full-working MATLAB engine within Python. Since UPSim is written in Python it was relatively straightforward to implement that engine into the core of UPSim. Thus, every time UPSim starts up, an instance of the MATLAB engine is automatically started and waits ready for any necessary computations.

The framework as presented in figure 4.20 consists of three basic components working together, Python, a MATLAB engine (run from Python) and DYNA4 itself. UPSim running in Python is used for the post-processing and the selection of the uncertain input parameters and their corresponding distributions. UPSim's GUI could also be used for the setup of the PCE parameters (its implementation was out of the scope of the present work, instead the PCE parameters are hard-coded into UPSim). Then the Python's MATLAB engine is used to compute the necessary sparse grid for the integration taking into account the PCE settings (using UQTk and the modifications described in section 4.2). Then DYNA4 is iteratively called and the TTM is evaluated at each of the sparse grid nodes, producing the corresponding result files. Finally, the results could be loaded into UPSim and the GUI could be used for the post-processing allowing the user to select the different output parameters and post-processing actions that should be taken, such as the computation of the spectral coefficients for each of the selected model outputs or the computation of the PC-based Sobol' indices.

Again, UPSim's GUI was not modified for this purpose, but the post-processing was instead performed for the outputs of interest using a series of MATLAB scripts and functions.

In this work two open-loop maneuvers will be considered for the validation of the results obtained with this framework, first a simple “step steering input" maneuver, similar to the one used with the STM. The second scenario is a more complex “braking in a turn" maneuver.

Next each of these scenarios will be thoroughly described and their corresponding results will be presented.

4.5.2 *Step steering input*

In this test case, 3 random normally distributed parameters will be used, namely the mass of the vehicle's body, the position of the center of gravity in x-direction (measured from the front axle in the vehicle's reference frame) and the spring stiffness of the anti-roll bar (front axle). Their corresponding means and standard deviations are shown in table 4.2.

Parameter description	Units	Mean (μ)	Std. dev. (σ)
Vehicle body mass	[kg]	1134.8	75
Vehicle body center of gravity position x	[m]	-1.285	0.147224
(Front axle) Spring stiffness of anti-roll bar	[N/m]	78400	11547

Table 4.2: Uncertain parameters and their distribution for the TTM step steering input scenario

In this case the vehicle accelerates from rest at $t = 1\ s$ to 15 [m/s] in 7.5 seconds (i.e. until $t = 8.5\ s$) and the speed is maintained thereafter. At $t = 9.5\ s$ (having reached constant speed) a steering step input of $\delta = 1.5\,\mathrm{rad}$ (approx. $86°$) takes place. At some point steady state is reached if the vehicle is stable and the vehicle travels in a circle. The simulation stops at $t = 14.5\ s$. For figures on the time evolution of some model outputs the reader is referred to appendix B (see B.12, B.13 and B.14).

The simulations are carried out with three different PCEs of orders 2, 4 and 6, and a QMC sampling-method based on a Sobol' sequence (just as for the STM).

As an example, figure 4.21, shows the convergence of mean value and standard deviation of the lateral acceleration, yaw rate, sideslip angle and lateral force of the front right wheel at $t = 14\ s$ as computed by means of a 2nd-, 4th- and 6th-order PCE and a QMC simulation (with 735 samples), respectively. As can be seen, a 2nd order PCE already offers a high accuracy in all investigated variables, whilst requiring a much lower number of model evaluations compared to a QMC simulation (19 versus approx. 300). Further similar images on this subject can be found in appendix B.

Moreover, the PC-based Sobol' indices of the different PCEs (2nd, 4th and 6th order) are compared to the sampling-based Sobol' indices obtained by means of QMC simulation, computed using again 735 samples (recall that for the computation of the Sobol' indices that means $735 \cdot (3 + 2)$ = 3675 model evaluations). As an example, this comparison is shown, over the number of model evaluations, in figure 4.22 and 4.23 1st-order and total Sobol' Indices of the vehicle body mass (S_1 and S_{T_1}, respectively) and of the yaw rate at $t = 14\ s$, respectively.

The rest of the indices for all model outputs exhibit similar trends, although they are not shown here in the interest of concision.

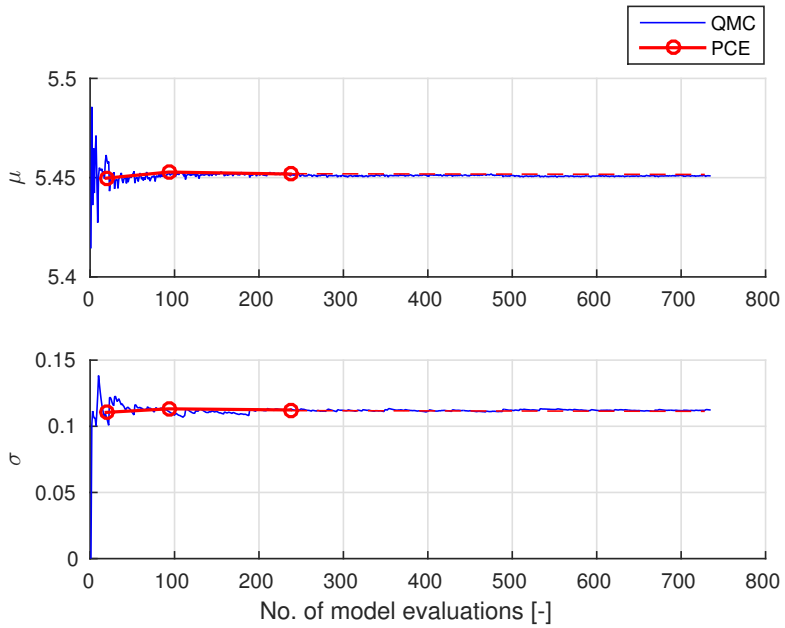


Figure 4.21: PCE vs. QMC convergence of the mean and standard deviation for the lateral acceleration a_y at $t = 14\ s$

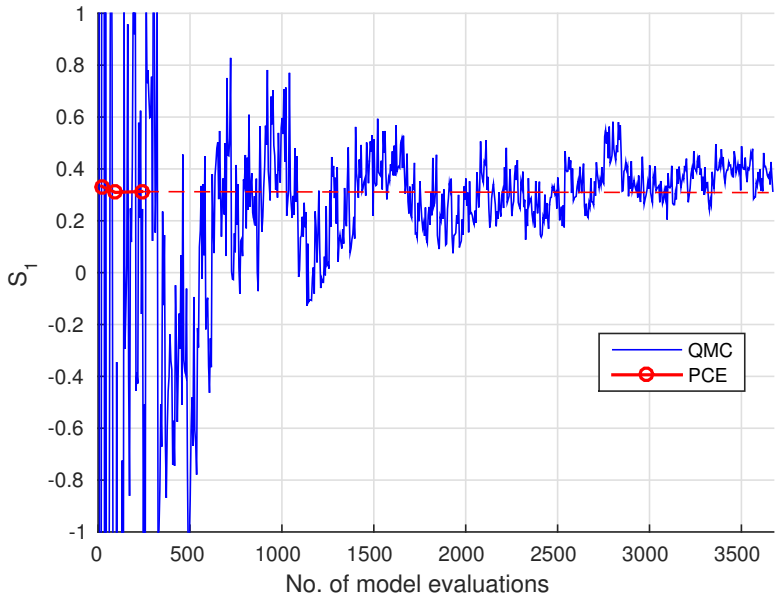


Figure 4.22: PCE vs. QMC convergence of the 1st first order Sobol' index at $t = 14\ s$ for the yaw rate $\dot{\psi}$

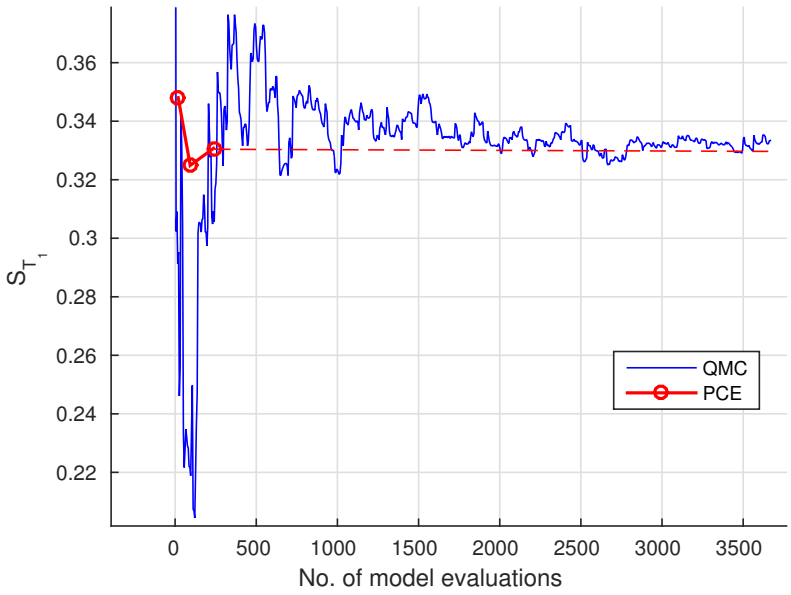


Figure 4.23: PCE vs. QMC convergence of the 1st total Sobol' index at $t = 14\ s$ for the yaw rate $\dot{\psi}$

As can be seen, the PC-based Sobol' indices match their sampling-based counterpart very accurately at a much lower computational cost, just as they did for the STM.

Figure 4.24 shows a direct comparison of the 6th-order PC-based and the QMC sampling-based (value of the last sample, i.e. 735) total Sobol' indices for the lateral acceleration at $t = 14\ s$.

As it can be observed, the results obtained through the PCE again match their sampling-based counterpart almost exactly. The relative error is less than 3% for each of the indices and, in fact, this error is probably due to a lack of convergence in the sampling-based results, since *only* 735 samples were used.

Furthermore, the KDE of the (6th-order) PC-based model outputs also show a high accuracy compared to their sampling-based counterpart. As an example, figure 4.25 and 4.26 show a comparison of both KDEs for the sideslip angle (β) and the lateral acceleration (a_y), respectively, at $t = 10.9$ (little after the step steering input).

The small differences in the KDEs of the PDFs can be attributed to a slight lack of convergence in the sampling-based QMC approach (since *only* 1470 samples were used in this case). Nonetheless, the relative error is minimal and the PC-based approach shows, once again, a good accuracy and a better performance than classical MC and QMC methods (recall that the 6th-order PCE required only 237 model evaluations in this case).

The results shown so far confirm that even for more complex non-linear models such as the TTM used in this case, PC is a good alternative to sampling-based techniques in terms of performance for UQ, both for the computation of model output statistics (mean and standard deviation), as well as for the computation of derived sensitivity measures, such as the Sobol' indices.

In what follows, the last test scenario explored within the present work will be described and the corresponding results will be analyzed.

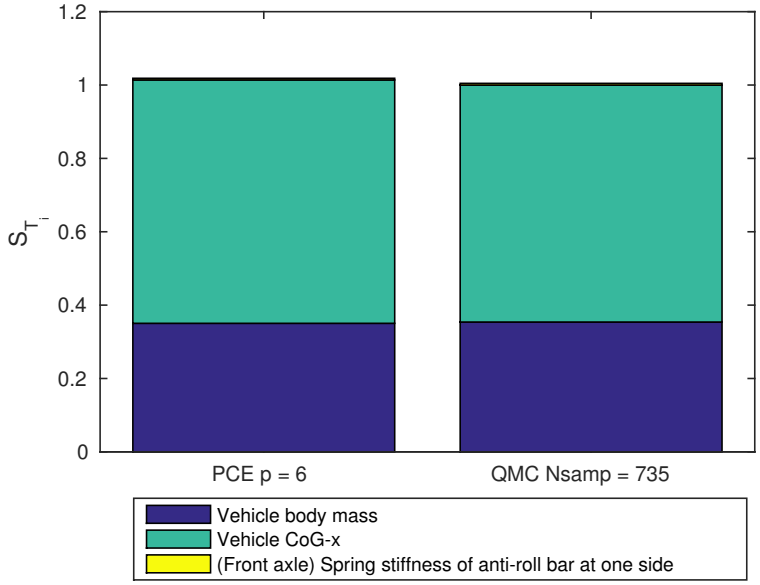


Figure 4.24: PCE vs. QMC comparison of the total Sobol' indices at $t = 14\ s$ for the lateral acceleration a_y

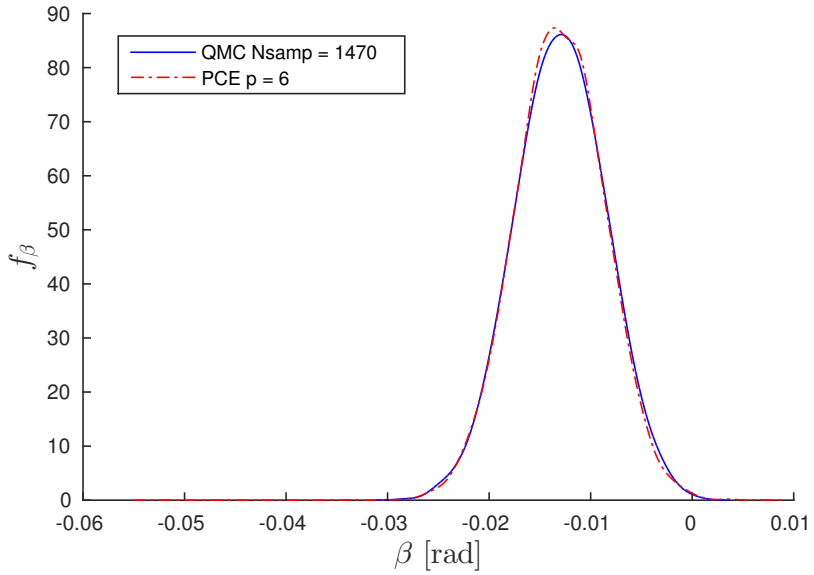


Figure 4.25: PCE vs. QMC comparison of the KDE at $t = 10.9\ s$ for the sideslip angle β

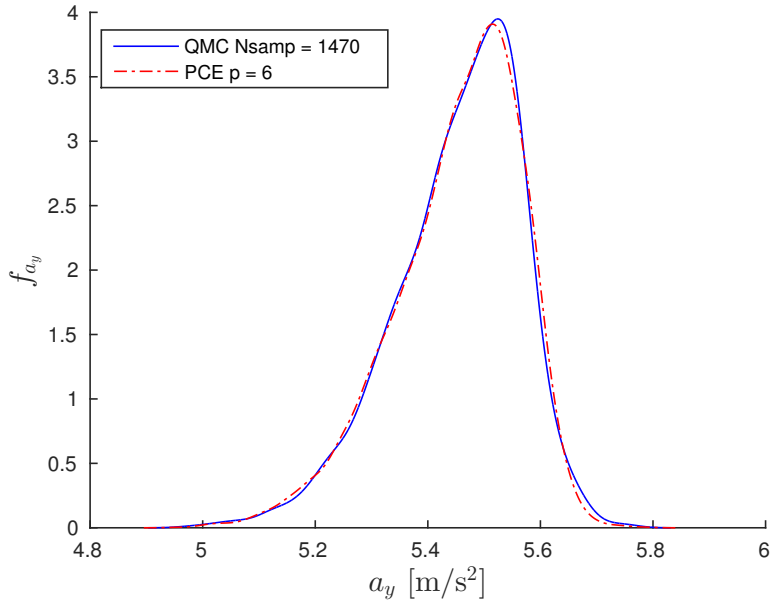


Figure 4.26: PCE vs. QMC comparison of the KDE at $t = 10.9$ s for the lateral acceleration a_y

4.5.3 *Braking in a turn*

The second and last maneuver for the TTM used in this work is *braking in a turn*. In this case 5 normally distributed parameters are used: the friction scaling factor, which describes the coefficient of friction between the wheels and the road (affected for example by the presence of rain, snow or ice on the road or by tire wear, among other factors); the maximum lateral force scaling factor, used to simulate the effects of different inflation pressures of the tires; the mass of the vehicle's body; the position of the center of gravity in x-direction (measured from the front axle in the vehicle's reference frame) and the rear axle wheel cylinder brake pressure delta, that is, the difference in the brake pressure applied to the rear-right and rear-left wheels. This last parameter is used to simulate the effects of a torque vectoring system, which allows for the variation of the power input to each wheel and thus provides a way to counteract undesired effects in the dynamics, such as oversteer (or understeer). The corresponding means and standard deviations used for these parameters are shown in table 4.3. The parameters are again selected in such a way that the vehicle remains stable during the whole maneuver, i.e. no unstable simulations occur.

Parameter description	Units	Mean (μ)	Std. dev. (σ)
Friction scaling factor	[-]	0.8	0.045
Maximum lateral force scaling factor	[-]	1.0	0.0065
Vehicle body mass	[kg]	1425	50
Vehicle body center of gravity position x	[m]	-1.26	0.0125
Rear axle wheel cylinder brake pressure delta	[N/m^2]	0	100000

Table 4.3: Uncertain parameters and their distribution for the TTM braking in a curve scenario

In this test scenario the vehicle accelerates from rest at $t = 1$ s to 20 [m/s] in 7.5 seconds (i.e. until $t = 8.5$ s) and the speed is maintained thereafter. At $t = 11.5$ s (having reached constant speed) a steering step input of $\delta = 0.55\,\text{rad}$ (approx. $31.51°$) takes place. At $t = 21$ s, having reached steady state, while the vehicle is traveling in a circular path, a

step braking input is carried out and, finally, at $t = 25\ s$ the simulation ends. Again, for figures on the time evolution of some model outputs the reader is referred to appendix B (see B.18, B.19, B.20 and B.21).

Once more the PC-based approach is compared to a sampling-based QMC method. Nevertheless, in spite of some recurring problems with the simulations stopping prematurely (that is, before the desired number of samples could be reached), only 397 samples could be computed with the QMC approach for the computation of the Sobol' indices (i.e., $397 \cdot (5 + 2) = 2779$ model evaluations; compared to the actual desired number of samples between 2000 and 3000 samples, that is, 14000 to 21000 model evaluations). Nevertheless, it shall be noted that 794 samples can actually be used for the computation of means, standard deviations, KDEs, etc. since in the sampling-based approach two independent sets of random parameters are first generated, which can be directly used for that purpose. Additionally, combinations thereof are required for the computation of conditional variances and, ultimately, for the estimation of the Sobol' indices as discussed in 3.2. The source of the problems seems to be that Python loses the connection with DYNA4 after some (apparently) undefined amount of time and thus the simulations stop after they timeout. In this case, the PC-based computations were carried out with 3rd-, 4th-, 5th- and 6th-order PCEs. That being said, the results presented in the following section shall be studied carefully and first qualitatively, since not all variables have necessarily converged with such a small amount of samples for the QMC simulations. Furthermore, classic MC simulations were also carried out in this case with 1897 samples (number again limited by the problems and the available time) in order to understand the convergence better. Nevertheless, the problem with the simulations stopping shall be investigated and is left as further research.

Firstly, as an example convergence plots for the mean of the yaw rate ($\dot{\psi}$) at $t = 15\ s$ and $t = 22\ s$ are presented in figures 4.27 and 4.28, respectively.

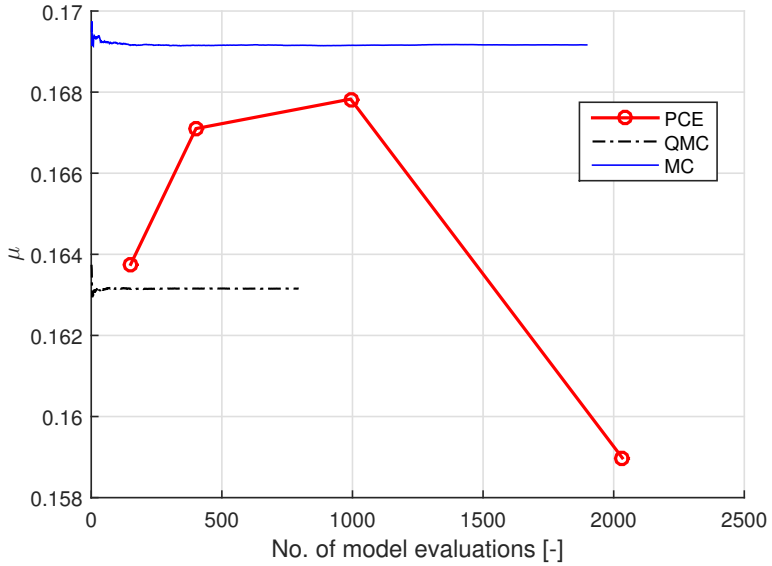


Figure 4.27: PCE, MC and QMC comparison for the mean of the yaw rate ($\dot{\psi}$) at $t = 15\ s$

As can be seen in both cases, for the statistical moments (the mean in this case) the QMC simulations converge quite fast, although they don't seem to converge to the right values predicted by the MC simulations. This can be explained by the fact that QMC methods are usually not as robust against non-smooth integrands as classical MCS, as was already mentioned in section 3.1.3. Additionally, as noted by Crestaux et al. (2009, p. 1165) and as

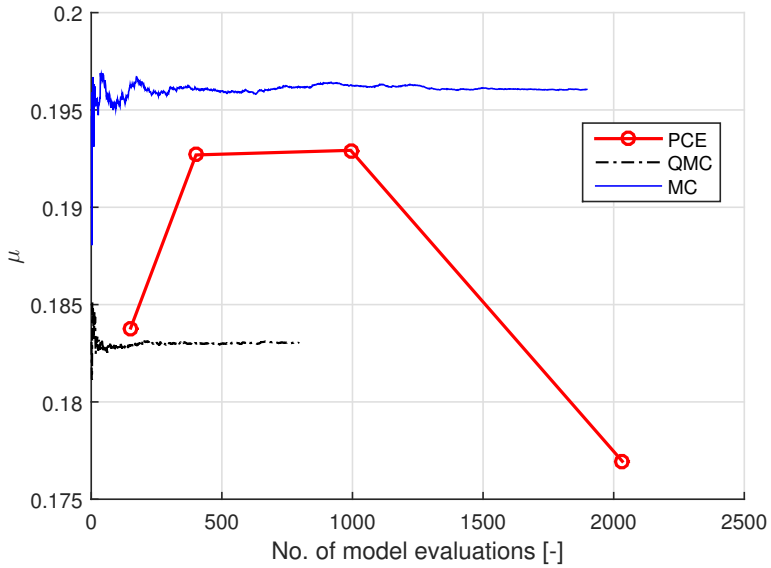


Figure 4.28: PCE, MC and QMC comparison for the mean of the yaw rate ($\dot{\psi}$) at $t = 22\ s$

mentioned earlier in this work, there seems to exist an optimal order for the PCE in this case, as can be seen, the 5th-order PCE. However, if one computes the relative errors for the different methods it is clear that the differences are, in practice, negligible for the statistics of all the studied variables (although not all of them are presented here in the interest of concision), for example, at $t = 15\ s$: $e_{rr}(PCE_{N=5}, MC) = |0.1678 - 0.1691|/0.1691 \cdot 100 \approx 0.769\%$. The small differences ($< 1\%$ in most cases) can be explained by the lower robustness of SGI against non-smooth integrands as compared to other integration methods, as mentioned in 3.1.3. Nevertheless, the results obtained my means of a 5th-order PCE computed by means of SGI are thus, in practice, accurate enough even for more complex statistics such as the Sobol' indices or the KDEs, as will be shown later.

Next, a comparison of the PC-based Sobol' indices with their QMC sampling-based counterpart will be presented. Nevertheless, it shall be noted that while the sampling-based total Sobol' indices have almost already converged with 397 samples (see figure 4.29 for an example), the same statement does not hold in general for the first order Sobol' indices (see figure 4.30) and although the relative importance of each parameter is relatively constant, only some of the total Sobol' indices will be compared here as an example.

Figure 4.31 compares the PC-based (using a 5th-order PCE) total Sobol' indices to their QMC sampling-based counterpart (using 397 samples) for the sideslip angle at $t = 22\ s$.

As can be observed, the differences in the computed total Sobol' indices are in this case minimal and are probably larger than they actually should be in spite of the lack of convergence for the QMC method.

Furthermore, a similar graph is shown for total Sobol' indices of the lateral acceleration at $t = 22\ s$ in figure 4.32. Again, the differences are small and probably due to the lack of convergence in the QMC approach. Moreover, the relative contributions of each parameter are also qualitatively accurate.

Recall that a 5th-order PCE using SGI requires, in this case, only 993 model evaluations whilst the QMC-based approach needs $(5 + 2) \cdot 397 = 2779$ model evaluations (and it would actually need around 7000 model evaluations for full convergence of the results).

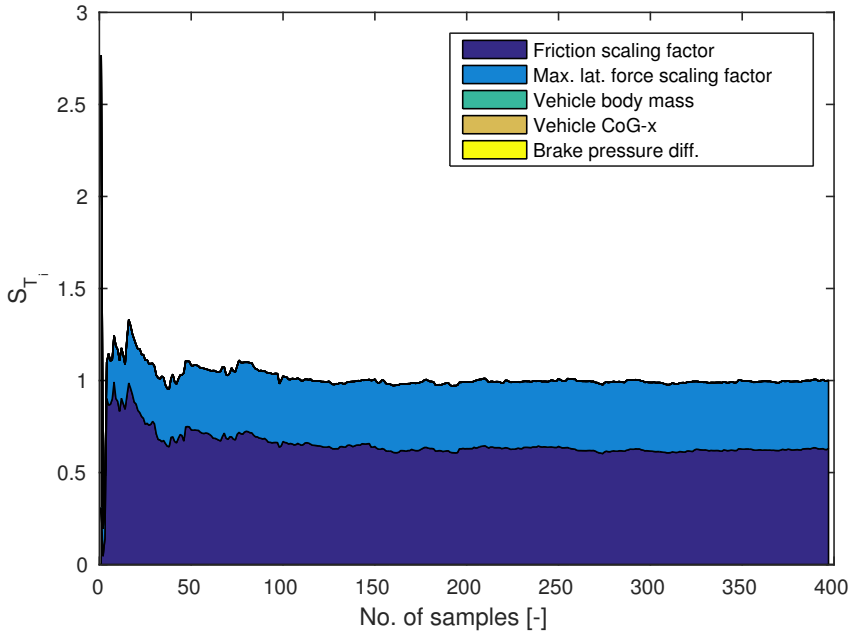


Figure 4.29: Example of convergence for the total Sobol' indices of the lateral acceleration (a_y) at $t = 15\ s$ computed by means of QMC simulation

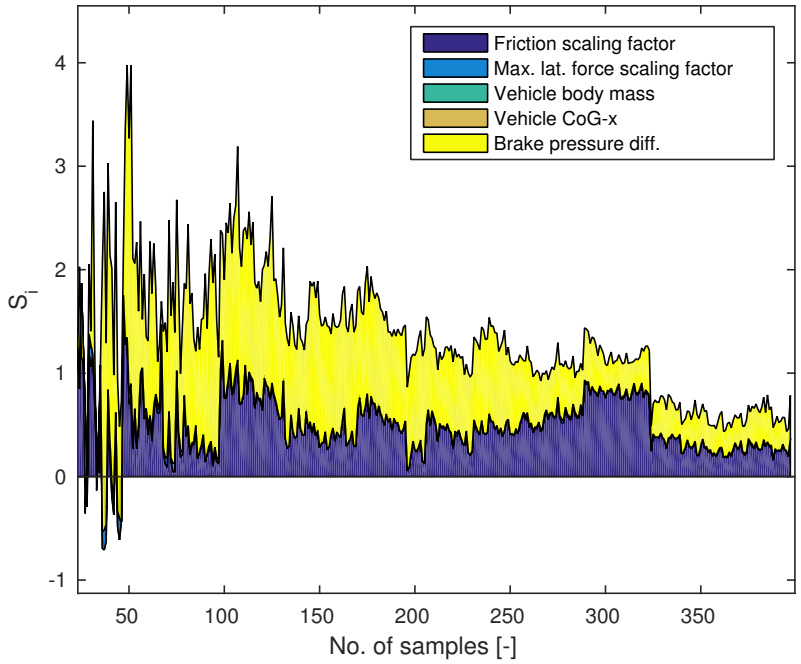


Figure 4.30: Example of non-convergence for the first order Sobol' indices of the yaw rate ($\dot{\psi}$) at $t = 22\ s$ computed by means of QMC simulation

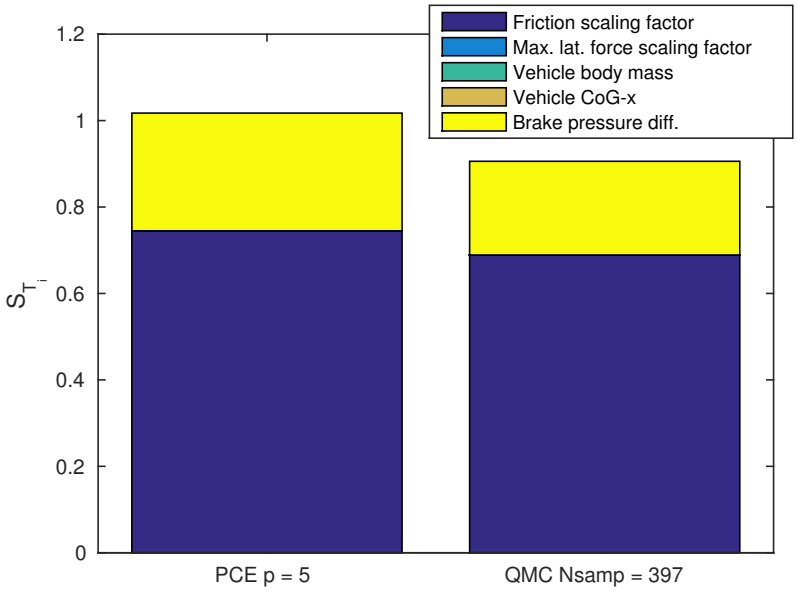


Figure 4.31: PCE vs. QMC total Sobol' indices of the sideslip angle (β) at $t = 22\ s$

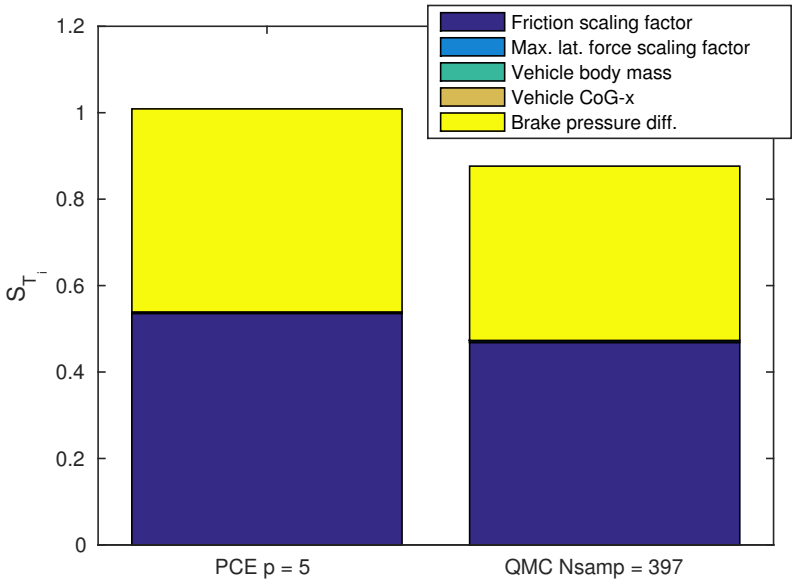


Figure 4.32: PCE vs. QMC total Sobol' indices of the lateral acceleration (a_y) at $t = 22\ s$

In addition, a comparison of some PC-based KDEs with their MC sampling-based counterpart is carried out as an example.

As can be seen in figure 4.33, the shapes of the KDEs match relatively accurately, although the lower robustness of SGI used for the computation of the spectral coefficients is probably the main reason for the small differences that can be appreciated. The same can be concluded for other model outputs, for example for the yaw rate ($\dot{\psi}$) at $t = 22\ s$, by inspecting figure B.22, found in appendix B.

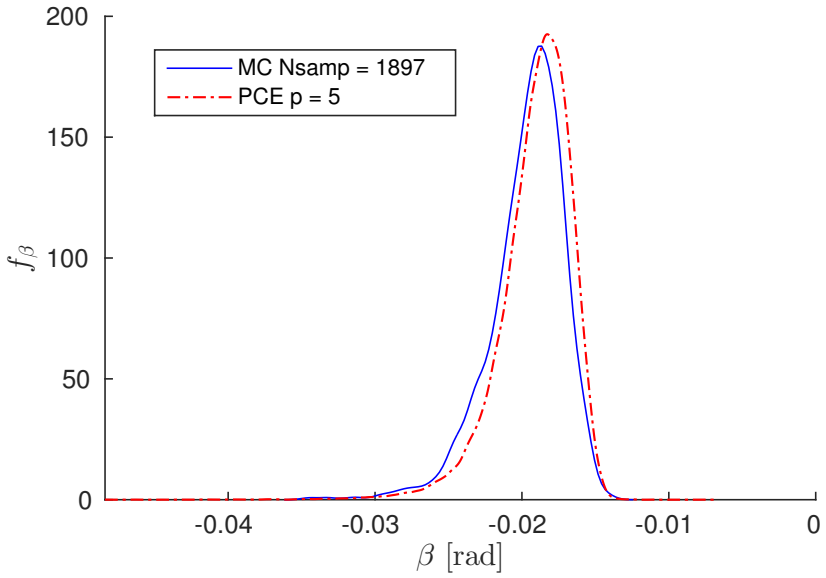


Figure 4.33: PCE vs. MC-based KDE for the sideslip angle β at $t = 22\ s$

Finally, all of the usual statistics used throughout this work are also shown for a more complex model output, namely the ratio of the yaw rate at $t = 22$ and $t = 21\ s$, i.e. $\frac{\dot{\psi}(22\ s)}{\dot{\psi}(21\ s)}$.

First, convergence plots for the 1st and 5th total Sobol' indices are shown in figures 4.34 and 4.35, respectively.

As can be observed in the figures, a 5th-order PCE shows again the best convergence compared with the QMC predictions (which are again lacking convergence), although the differences are relatively small. The comparison of all five total Sobol' indices for the 5th-order PCE and the QMC approach are shown in figure 4.36.

As can be seen, the results exhibit minor differences and qualitatively the influences of each parameter are correctly captured by the 5th-order PCE with far fewer model evaluations, the differences being again probably caused by the lack of convergence in the QMC simulations.

Furthermore, the KDE predicted by using a 5th-order PCE is also very similar to the one computed by means of MCS, as can be seen in figure 4.37, and again any differences might probably be attributed to the lack of convergence in the MC-based approach.

In conclusion, the PCE approach shows again accurate and relatively robust results at a lower computational cost than MC and QMC methods and is a valuable tool for Uncertainty Quantification (UQ) and the computation of variance-based sensitivity measures such as the Sobol' sensitivity indices, outperforming sampling-based methods if the number of stochastic dimensions is relatively low. Nevertheless, it should be noted that after the investigation of all the results, a convergence analysis appears to be of utmost importance if the model outputs are relatively complex and the SGI method is being used for the computation of the spectral

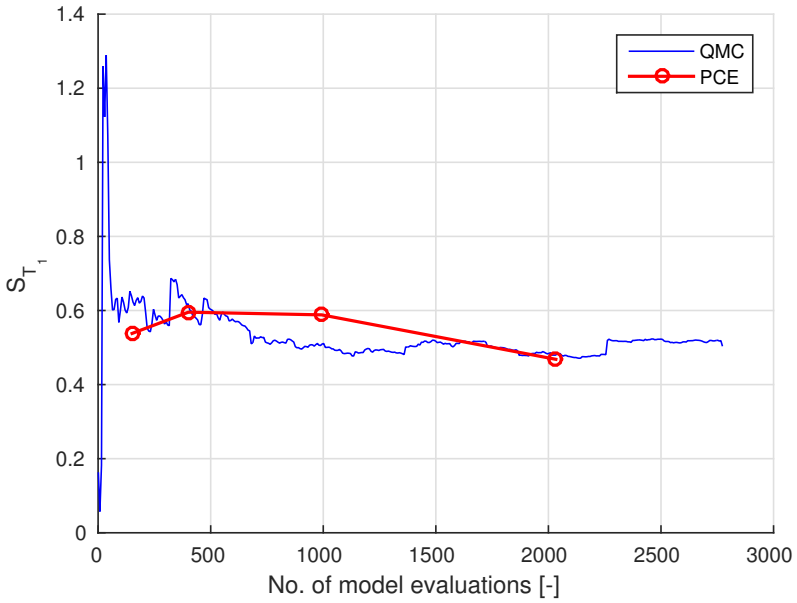


Figure 4.34: PCE vs. QMC 1st total Sobol' index for the ratio $\dot{\psi}(22\ s)/\dot{\psi}(21\ s)$

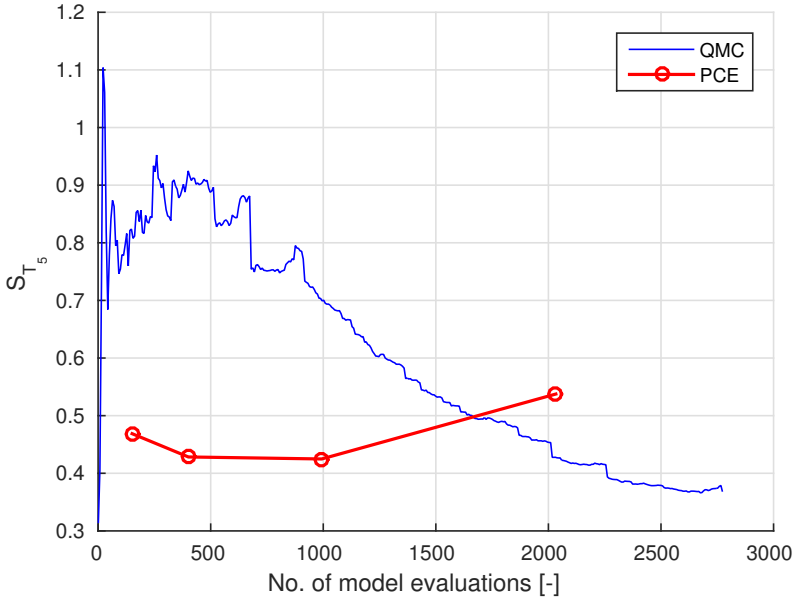


Figure 4.35: PCE vs. QMC 5th total Sobol' index for the ratio $\dot{\psi}(22\ s)/\dot{\psi}(21\ s)$

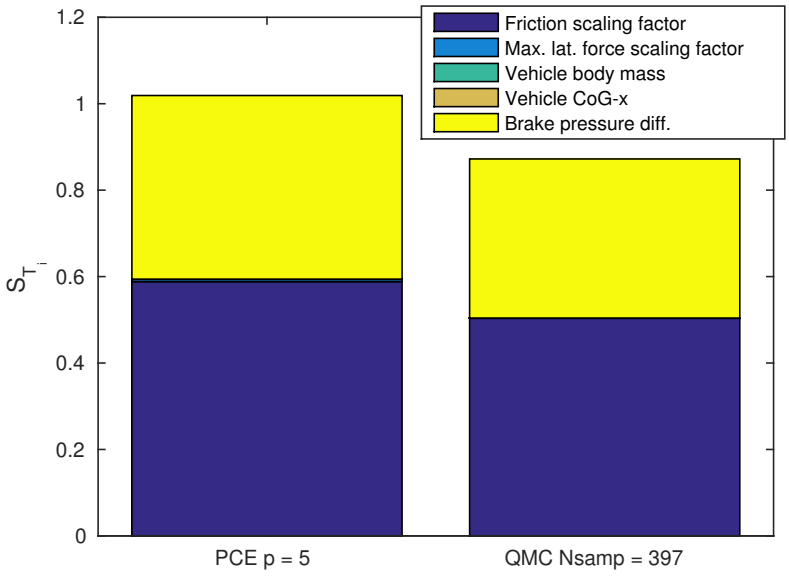


Figure 4.36: PCE vs. QMC total Sobol' indices for the ratio $\dot{\psi}(22\ s)/\dot{\psi}(21\ s)$

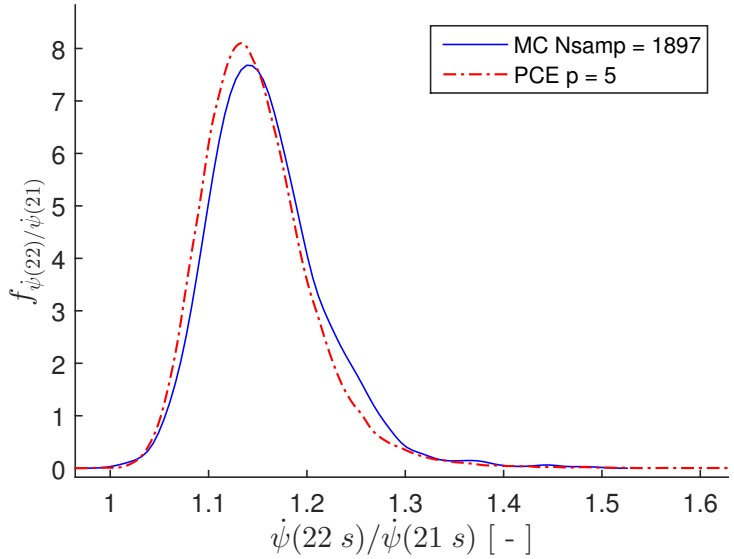


Figure 4.37: PCE vs. MC-based KDE for the ratio $\dot{\psi}(22\ s)/\dot{\psi}(21\ s)$

coefficients. The use of 4th- or 5th-order PCE is suggested considering the results shown and probably a study of the results obtained by using the Linear Regression Method (LRM) for the computation of the spectral coefficients would also be advisable in complex scenarios like this last one.

5 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

In this work the study of a method for the computation of the global Sobol' sensitivity indices based on the theory of Polynomial Chaos (PC) has been carried out and the method has been successfully applied to the propagation of uncertainty in the simulation of vehicle dynamics.

First of all, different approaches to the computation of the spectral coefficients of the Polynomial Chaos Expansion (PCE) have been investigated, more precisely non-intrusive methods: the Linear Regression Method (LRM) and two different Non-intrusive Spectral Projection (NISP) methods, namely the tensor product rule and sparse grid integration. The resulting PC-based Sobol' indices and other statistics of interest (means, standard deviations and Kernel Density Estimations (KDEs) of Probability Density Functions (PDFs)) have been studied and compared to the results based on classical sampling methods such as Monte Carlo (MC) or Quasi-Monte Carlo (QMC) Simulation.

The methods have been verified and validated using test functions that are typically used in the field of Uncertainty Quantification (UQ). As has been shown, the three methods for the computation of the spectral coefficients that have been investigated have resulted in very similar statistics and Sobol' indices and have matched their analytical counterparts accurately.

In addition, the methods have been applied to a transient linear Single-Track Model (STM) for vehicle dynamics by means of an open-loop step steering maneuver with 3 uncertain parameters. The results have shown a high accuracy of the PC-based approach both in the prediction of statistics of the model outputs as well as in the estimation of first order and total Sobol' sensitivity indices. Moreover, the PCE has proven to outperform classical sampling-based techniques (such as MC and QMC) requiring far fewer model evaluations in this case and thus achieving a substantial gain in performance. Particularly, the use of Sparse Grid Integration (SGI) for the computation of the spectral coefficients has proven to be very efficient in terms of accuracy versus computation time and thus it has been used for the computation of the PCEs for the rest of the work.

Finally, the methods studied in this work have been put to use with a more complex transient non-linear Two-Track Model (TTM) (with TESIS Dynaware's DYNA4 Framework) employing two different maneuvers and a framework combining Python, MATLAB and DYNA4, specifically designed for this task. The results have shown that on relatively simple scenarios, such as the step steering maneuver, PCE using SGI again outperforms sampling-based methods providing a very good accuracy for all of the aforementioned statistics.

However, if the scenario is complex and the model outputs are no longer smooth, special care must be taken in studying the results if Sparse Grid Integration (SGI) is used, in spite of its robustness limitations. Convergence analyses are suggested in this case, comparing increasing orders of the PCE at different time instants with sampling-based techniques (ideally Monte Carlo (MC) and Quasi-Monte Carlo (QMC) simulations) and the use of LRM is

recommended if the results are not satisfactory, since it also offers a good accuracy at a low computational cost.

Nevertheless, it must be noted that a PC-based approach is usually only feasible when the stochastic dimension of the problem (i.e. the number of uncertain parameters) is relatively low (usually < 10) and a sampling-based approach is suggested otherwise.

Finally, it should be mentioned that there is still room for improvement in some parts of the code and the framework developed throughout this work and that other PC-based techniques which are recently appearing in the literature, such as sparse PCEs, could be investigated. In addition, future work could be carried out in the parallelization of the studied methods which could also lead to further reductions in execution time and consequently result in great improvements in performance.

In conclusion, this work has shown that the PC-based approach for the computation of the global Sobol' sensitivity indices usually offers a good alternative to classical sampling-based methods in terms of accuracy and performance, specially if the number of stochastic dimensions is relatively low. In particular, it has been shown that the NISP method for the computation of the spectral coefficients using Sparse Grid Integration (SGI) achieves a very good performance provided that the integrand is relatively smooth and that its accuracy level and the order of the PCE are chosen appropriately. Finally, the investigated methods have been successfully applied to the simulation of vehicle dynamics.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Master's Thesis in Industrial Engineering

Alejandro Valero Andreu

Variance-based sensitivity analysis in vehicle dynamics simulation: development and application of a polynomial chaos expansion method

BUDGET

Contents of the budget

1 INTRODUCTION

In order for a university or an external company to evaluate the costs and study the possibility of executing a project an estimation of the budget is required.

Therefore, in the next chapters a budget for the whole project is presented. It contains all estimated costs required for the development of this work, thus it is structured following the logical order of the tasks carried out in the project (except for the meetings, which are considered an independent unit, taking place in parallel to the development of the project).

First of all, a detailed and disaggregated budget containing all the items and tasks developed during this project, their description and the required resources and personnel for each one of them is introduced in chapter 2.

Next, a summary of the chapters including the usual coefficients such as the overhead expenses, the industrial profit and the V.A.T., as well as the overall budget, is presented in chapter 3.

2 MEASUREMENTS & BUDGET

Order No.	Description of the construction units	Units		Measurement	Price	Subtotal

1 RESEARCH & LABOR

1.1 MEETING

Meeting between the engineer and the supervisor for general information exchange regarding the project, feedback and review of the progress.

E01	h	ENGINEER	5.00	45.00	225.00
P01	h	PROFESSOR	5.00	100.00	500.00

Total of item 1.1 ..... 10.00 .... 72.50 € ..... 725.00 €

1.2 IMPLEMENTATION

Programming, implementation and setup of the required programs and models in MATLAB, Python and DYNA4.

1.2.1 LINEAR REGRESSION METHOD

Implementation of the Linear Regression Method for the computation of the Polynomial Chaos Expansion Coefficients.

E01	h	ENGINEER	40.00	45.00	1,800.00
C01	h	COMPUTER	40.00	0.57	22.80
L01	h	SOFTWARE LICENSES	40.00	2.27	90.80

Total of item 1.2.1 ..... 1.00 1,913.60 € ..... 1,913.60 €

1.2.2 SPARSE GRID INTEGRATION METHOD

Implementation of the Sparse Grid Integration Method for the computation of the Polynomial Chaos Expansion Coefficients.

E01	h	ENGINEER	16.00	45.00	720.00
C01	h	COMPUTER	16.00	0.57	9.12
L01	h	SOFTWARE LICENSES	16.00	2.27	36.32

Total of item 1.2.2 ..... 1.00 .. 765.44 € ..... 765.44 €

1.2.3 TEST FUNCTIONS

Implementation of the test functions for the validation and verification of the different methods for the computation of the Polynomial Chaos Expansion Coefficients.

E01	h	ENGINEER	5.00	45.00	225.00
C01	h	COMPUTER	5.00	0.57	2.85
L01	h	SOFTWARE LICENSES	5.00	2.27	11.35

Total of item 1.2.3 ..... 1.00 .. 239.20 € ..... 239.20 €

Order No.	Description of the construction units	Units		Measurement	Price	Subtotal

1.2.4 SINGLE-TRACK MODEL

Implementation and setup of the Single-Track Model and the corresponding maneuver scenarios.

E01	h	ENGINEER	16.00	45.00	720.00
C01	h	COMPUTER	16.00	0.57	9.12
L01	h	SOFTWARE LICENSES	16.00	2.27	36.32

Total of item 1.2.4 ..... 1.00 .. 765.44 € ..... 765.44 €

1.2.5 PYTHON-MATLAB-DYNA4 FRAMEWORK

Implementation and setup of the Python-MATLAB-DYNA4 Framework for the simulation of the Two-Track Model test cases.

E01	h	ENGINEER	56.00	45.00	2,520.00
C01	h	COMPUTER	56.00	0.57	31.92
L01	h	SOFTWARE LICENSES	56.00	2.27	127.12

Total of item 1.2.5 ..... 1.00 2,679.04 € ..... 2,679.04 €

1.2.6 TWO-TRACK MODEL

Implementation and setup of the Two-Track Model and its corresponding test scenarios.

C01	h	COMPUTER	40.00	0.57	22.80
E01	h	ENGINEER	40.00	45.00	1,800.00
L01	h	SOFTWARE LICENSES	40.00	2.27	90.80

Total of item 1.2.6 ..... 1.00 1,913.60 € ..... 1,913.60 €

Total of chapter 1.2 .....8,276.32 €

1.3 TESTING & DEBUGGING

Testing and debugging of the required programs and models in MATLAB, Python and DYNA4.

1.3.1 LINEAR REGRESSION METHOD

Testing and debugging of the Linear Regression Method for the computation of the Polynomial Chaos Expansion Coefficients.

E01	h	ENGINEER	16.00	45.00	720.00
C01	h	COMPUTER	16.00	0.57	9.12
L01	h	SOFTWARE LICENSES	16.00	2.27	36.32

Total of item 1.3.1 ..... 1.00 .. 765.44 € ..... 765.44 €

Order No.	Description of the construction units	Units		Measurement	Price	Subtotal
1.3.2	SPARSE GRID INTEGRATION METHOD					
	Testing and debugging of the Sparse Grid Integration Method for the computation of the Polynomial Chaos Expansion Coefficients.					
E01	h	ENGINEER		8.00	45.00	360.00
C01	h	COMPUTER		8.00	0.57	4.56
L01	h	SOFTWARE LICENSES		8.00	2.27	18.16
		Total of item 1.3.2	.....	1.00	.. 382.72 €	..... 382.72 €
1.3.3	TEST FUNCTIONS					
	Testing and debugging of the test functions for the validation and verification of the different methods for the computation of the Polynomial Chaos Expansion Coefficients.					
E01	h	ENGINEER		3.00	45.00	135.00
C01	h	COMPUTER		3.00	0.57	1.71
L01	h	SOFTWARE LICENSES		3.00	2.27	6.81
		Total of item 1.3.3	.....	1.00	.. 143.52 €	..... 143.52 €
1.3.4	SINGLE-TRACK MODEL					
	Testing and debugging of the Single-Track Model and the corresponding maneuver scenarios.					
E01	h	ENGINEER		4.00	45.00	180.00
C01	h	COMPUTER		4.00	0.57	2.28
L01	h	SOFTWARE LICENSES		4.00	2.27	9.08
		Total of item 1.3.4	.....	1.00	.. 191.36 €	..... 191.36 €
1.3.5	PYTHON-MATLAB-DYNA4 FRAMEWORK					
	Testing and debugging of the Python-MATLAB-DYNA4 Framework for the simulation of the Two-Track Model test cases.					
E01	h	ENGINEER		56.00	45.00	2,520.00
C01	h	COMPUTER		56.00	0.57	31.92
L01	h	SOFTWARE LICENSES		56.00	2.27	127.12
		Total of item 1.3.5	.....	1.00	2,679.04 €	..... 2,679.04 €
1.3.6	TWO-TRACK MODEL					
	Testing and debugging of the Two-Track Model and its corresponding test scenarios.					
L01	h	SOFTWARE LICENSES		24.00	2.27	54.48
C01	h	COMPUTER		24.00	0.57	13.68
E01	h	ENGINEER		24.00	45.00	1,080.00
		Total of item1.3.6	.....	1.00	1,148.16 €	..... 1,148.16 €
	Total of chapter 1.3		**.....**			**5,310.24 €**

Order No.	Description of the construction units	Units		Measurement	Price	Subtotal
1.4	DATA ANALYSIS Analysis of the data acquired during the simulations of the test cases for both the Single-Track and the Two-Track Model.					
E01	ENGINEER	h		80.00	45.00	3,600.00
	Total of item 1.4			1.00	3,600.00 €	3,600.00 €
1.5	REPORT ELABORATION Elaboration of a complete report of the simulations with the obtained results, including the data analysis and the conclusions derived therefrom.					
E01	ENGINEER	h		160.00	45.00	7,200.00
	Total of item 1.5			1.00	7,200.00 €	7,200.00 €
	Total of chapter 1.3					**25,111.56 €**
	Total budget					**25,111.56 €**

3 CHAPTER SUMMARY

Order No.	Description of the chapters	Subtotal	%
1	RESEARCH & LABOR	25,111.56	100.00 %
1.2	IMPLEMENTATION	8,276.32	32.96 %
1.3	TESTING & DEBUGGING	5,310.24	21.15 %
1.4	DATA ANALYSIS	3,600.00	14.34 %
1.5	REPORT ELABORATION	7,200.00	28.68 %

BUDGET FOR MATERIAL IMPLEMENTATION OF THE PROJECT	**25,111.56 €**
15 % Overhead Expenses	3,766.73 €
6 % Industrial Profit	1,506.69 €
BUDGET FOR PROVISION OF THE CONTRACTED SERVICES	**30,384.98 €**
21 % V.A.T.	6,380.85 €
OVERALL BUDGET W/VAT	**36,765.83 €**

The projected budget adds up to the expressed amount of:
THIRTY-SIX THOUSAND, SEVEN-HUNDRED SIXTY-FIVE EUROS AND EIGHTY-THREE CENTS

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Master's Thesis in Industrial Engineering

Alejandro Valero Andreu

Variance-based sensitivity analysis in vehicle dynamics simulation: development and application of a polynomial chaos expansion method

PROJECT SPECIFICATIONS

Contents of the project specifications

1 PROJECT SPECIFICATIONS

During the early stages of any new product development, many parameters of the product being developed will change and are therefore subject to uncertainty. In particular, in the automotive industry, these uncertainties in the parameters will ultimately affect the behavior of the vehicle being developed. If particular vehicle dynamics are targeted, knowing which specific input parameter of the vehicle is more important in determining the uncertainty in the output of interest is of utmost importance.

Vehicle dynamics models have been known and used for a long time and they can be very useful for simulating the behavior of the vehicle under certain maneuvers if its parameters are known. Nonetheless, even when some parameters are uncertain, reasonable assumptions can usually be accepted regarding their variance, based on experience or previous similar products, for example. Consequently, the propagation of uncertainty from the parameters to the output variables and thus the stochastic behavior of the model can be determined in principle.

In this project, variance-based sensitivity measures and methods for global Sensitivity Analysis will be investigated. In particular, the Polynomial Chaos Expansion method (with many applications in other fields of science and engineering) and the computation of Sobol' sensitivity indices based on polynomial chaos expansions will be applied to the simulation of vehicle dynamics.

The project is divided into the following steps:

- State of the art: research of existing literature on the subjects
- Fundamentals of the polynomial chaos method for the computation of variance-based sensitivity measures (Sobol' indices)
- Implementation of polynomial chaos for the computation of the Sobol' indices
- Implementation of different methods for the computation of the expansion coefficients: non-intrusive spectral projection method, linear regression and sparse grid integration
- Reproduction of typical numerical examples (test functions)
- Application of the forementioned methods to the Car Professional model of Tesis' Dyna4 simulation framework
- Analysis and interpretation of the parametric studies
- Benchmarking of the different methods for the computation of the polynomial chaos expansion coefficients
- Benchmarking of polynomial chaos against classical algorithms

The project report is intended to document the individual steps in a clear manner.

The project is an adaptation from previous work presented at the Technical University of Munich as part of the T.I.M.E. Double Degree Program and the author has received explicit permission by the Technical University of Munich for its use and adaptation, therefore no confidentiality agreement has been violated.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES VALENCIA

Master's Thesis in Industrial Engineering

Alejandro Valero Andreu

Variance-based sensitivity analysis in vehicle dynamics simulation: development and application of a polynomial chaos expansion method

LISTS, BIBLIOGRAPHY & APPENDICES

Symbols

a_y	Lateral acceleration
A_{ij}	Information matrix
c_{α_h}	Front wheel cornering stiffness
c_{α_v}	Rear wheel cornering stiffness
d	Stochastic dimension of the PCE
D	Total variance of the model output Y
$\hat{D}$	Total sampling-based variance of the model output Y
$D_{\boldsymbol{u}}$	Conditional variance of the model output Y
$\hat{D}_{\boldsymbol{u}}$	Conditional sampling-based variance of the model output Y
$\hat{f}_h$	KDE
$\mathbb{E}$	Expected value
F	Unknown distribution
$\hat{F}_n$	Empirical distribution
F_Z	joint CDF of Z
F_{Z_i}	marginal CDF of Z_i
F_{vr}^y	Lateral force of the front right wheel
$\mathcal{F}$	Set of possible events
h	Bandwidth for the KDE
$\boldsymbol{i}$	Multi-index
I_d	d-variate integral
J_{zz}	Yaw moment of inertia (moment of inertia about the vertical axis of the vehicle)
k	Single-index
K	Kernel
$\mathcal{L}$	Differential operator in space and/or time
l_h	Distance rear axle-center of gravity
l_v	Distance front axle-center of gravity
L^p	Lebesgue function space of pth-order
m	Total mass of the vehicle
m_k	k-th statistical moment
N	Order of the PCE
$\mathbb{N}$	Set of natural numbers (positive integers)
$\mathbb{N}_0$	Set of natural numbers and the zero (non-negative integers)
$\mathcal{N}$	Gaussian/normal distribution
$\mathcal{O}$	Big O notation
P	Number of PCE terms
Q_L^d	d-variate quadrature rule of accuracy level L
$\mathcal{P}$	Probability measure
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of positive real numbers
$\mathbb{R}_0^+$	Set of positive real numbers and the zero
S_i	First order Sobol' indices
S_{T_i}	Total Sobol' indices

$S_{u_1,\ldots,u_s}$	sth-order Sobol' indices
$\hat{S}_i$	First order sampling-based Sobol' indices
$\hat{S}_{T_i}$	Total sampling-based Sobol' indices
$\hat{S}_{u_1,\ldots,u_s}$	sth-order sampling-based Sobol' indices
T_l	Lower limit of confidence
T_u	Upper limit of confidence
$\boldsymbol{u}$	Multi-index
$\mathcal{U}$	Uniform distribution
$\boldsymbol{v}$	Multi-index
v_x	Vehicle velocity in x-direction
Var	Variance
w^i	Weighting function for node ξ^i of the Gaussian quadrature
y_k	Spectral expansion coefficient
X_i	Random parameter/variable
X_{ij}	Design of experiment, quadrature nodes matrix
Y	Model output
Z_i	Random parameter/variable
α	Positive measure
$\boldsymbol{\alpha}$	Multi-index
β	Sideslip angle
$\dot{\beta}$	Sideslip angular velocity
γ	Normalizing constant or Confidence level
δ	Steering angle
δ_{nm}	Kronecker delta function
ϵ	Residual in the linear regression method
$\boldsymbol{\zeta_u}^{(i)}$	Combined ith-sample
$\boldsymbol{\eta_u}^{(i)}$	ith-sample for the sampling-based Sobol' indices
θ	Unknown statistic or parameter
$\hat{\theta}$	Statistic estimator
μ	Mean value
$\boldsymbol{\xi}$	Random vector
$\boldsymbol{\xi_u}^{(i)}$	ith-sample for the sampling-based Sobol' indices
ξ_k	Univariate random variable
ρ	Probability Density Function (PDF)
σ	Standard deviation
$\dot{\psi}$	Yaw rate
$\ddot{\psi}$	Yaw acceleration
Ψ_k	Orthogonal polynomial of kth-order
Ω	Sample space

Abbreviations

ANOVA Analysis of Variance

STM Single-Track Model

TTM Two-Track Model

gPC generalized Polynomial Chaos

MC Monte Carlo

QMC Quasi-Monte Carlo

MCS Monte Carlo Simulation

MSE Mean Squared Error

KDE Kernel Density Estimation

PC Polynomial Chaos

PCE Polynomial Chaos Expansion

SA Sensitivity Analysis

PDF Probability Density Function

CDF Cumulative Distribution Function

LRM Linear Regression Method

SGI Sparse Grid Integration

SDE Stochastic Differential Equation

UQ Uncertainty Quantification

UQTk Uncertainty Quantification Toolkit

DAKOTA Design Analysis Kit for Optimization and Terascale Applications

NISP Non-intrusive Spectral Projection

XML Extensible Markup Language

RPC Remote Procedure Call

API Application Program Interface

GUI Graphical User Interface

ECU Electronic Control Unit

List of Figures

List of Tables

Bibliography

Askey, R. & Wilson, J. A. (1985). *Some basic hypergeometric orthogonal polynomials that generalize jacobi polynomials*. Memoirs of the American Mathematical Society. American Mathematical Society. doi:http://dx.doi.org/10.1090/memo/0319

Augustin, F. M. (2012). *Generalized wiener expansions for the numerical solution of random ordinary differential equations* (Doctoral dissertation, Technische Universität München).

Bartsch, C. (2014–2015). *Implementierung von sampling methoden zur quantifizierung von unsicherheiten in das fahrdynamik simulationsframework dyna4* (Master's Thesis, Technische Universität München, Fakultät für Maschinenwesen).

Berveiller, M. (2005). *Éléments finis stochastiques: approches intrusive et non intrusive pour des analyses de fiabilité* (Doctoral dissertation, Université Blaise Pascal, Clermont-Ferrand, France).

Cameron, R. H. & Martin, W. T. (1947, April). The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, *48*(2), 385–392.

Campolongo, F., Cariboni, J., Saltelli, A., & Schoutens, W. (2005). Enhancing the morris method. *European Commission, Joint Research Centre*, *31319*, 369–379. Retrieved from http://publications.jrc.ec.europa.eu/repository/handle/JRC31319

Chihara, T. (1978). *An introduction to orthogonal polynomials*. American Mathematical Society, Providence.

Crestaux, T., Le Maitre, O., & Martinez, J. M. (2009, July). Polynomial chaos expansion for sensitivity analysis. *Reliability Engineering & System Safety*, *94*(7), 1161–1172. doi:10.1016/j.ress.2008.10.008

Debusschere, B. J., Najm, H. N., Pébay, P. P., Knio, O. M., Ghanem, R. G., & Le Maître, O. P. (2004). Numerical challenges in the use of polynomial chaos representations for stochastic processes. *SIAM Journal on Scientific Computing*, *26*(2), 698–719.

Debusschere, B., Sargsyan, K., & Safta, C. (2014, June). *Uqtk version 2.1 user manual*. Sandia National Laboratories. Retrieved from http://www.sandia.gov/UQToolkit/UQTk_v2.1_manual.pdf

Efron, B. (1979, January). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, *7*(1), 1–26. Retrieved from http://www.stat.cmu.edu/~fienberg/Statistics36-756/Efron1979.pdf

Efron, B. & Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, *1*(1), 54–77. Retrieved from https://projecteuclid.org/download/pdf_1/euclid.ss/1177013815

Efron, B. & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman & Hall/CRC.

Engel, J. & Grübel, R. (2008, March). Bootstrap – oder die kunst, sich selbst aus dem sumpf zu ziehen. *Mathematische Semesterberichte*, *55*(2), 113–130.

Ghanem, R. & Spanos, P. (1991). *Stochastic finite elements: a spectral approach*. Springer Verlag. doi:10.1007/978-1-4612-3094-6

Hanss, M. (2002). The transformation method for the simulation and analysis of systems with uncertain parameters. *Fuzzy Sets and Systems*, *130*(3), 277–289. doi:http://dx.doi.org/10.1016/S0165-0114(02)00045-3

Härdle, W. & Simar, L. (2007). *Applied multivariate statistical analysis* (2nd ed.). Springer.

Heiss, F. [F.] & Winschel, V. [V.]. (2015, August). Quadrature on sparse grids. Retrieved from http://www.sparse-grids.de

Heiss, F. [Florian] & Winschel, V. [Viktor]. (2006, April). Estimation with numerical integration on sparse grids. *Munich Discussion Papers in Economics*, *15*. Retrieved from http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-916-5

Held, L. & Bové, D. S. (2014). *Applied statistical inference*. Springer Verlag.

Homma, T. & Saltelli, A. [Andrea]. (1996, April). Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, *52*(1), 1–17. doi:10.1016/0951-8320(96)00002-6

Isermann, R. (2006). *Fahrdynamik-regelung: modellbildung, fahrerassistenzsysteme, mechatronik* (1st ed.). Vieweg, Teubner Verlag.

Klenke, A. (2014). *Probability theory* (2nd ed.). Universitext. Springer London. doi:10.1007/978-1-4471-5361-0_5

Le Maitre, O. & Knio, O. (2010). *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer Verlag.

Lienkamp, M. (2015). Dynamik der straßenfahrzeuge. Vorlesungsskript. Technische Universität München.

Matousek, J. (1998). On the l2-discrepancy for anchored boxes. *Journal of Complexity*, *14*(4), 527–556.

Metropolis, N. & Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, *44*(247), 335–341.

Niederreiter, H. (1988). Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, *30*(1), 51–70. doi:10.1016/0022-314X(88)90025-X

Niederreiter, H. (1992). *Random number generation and quasi-monte carlo methods*. Society for Industrial and Applied Mathematics.

Oakley, J. E. (2015, August). Coefficient vectors and matrices of the oakley & o'hagan function. Retrieved from http://www.jeremy-oakley.staff.shef.ac.uk/psa_example.txt

Oakley, J. E. & O'Hagan, A. (2004, August). Probabilistic sensitivity analysis of complex models: a bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodolgy)*, *66*(3), 751–769. doi:10.1111/j.1467-9868.2004.05304.x

Paffrath, M. & Wever, U. (2007). Adapted polynomial chaos expansion for failure detection. *Journal of Computational Physics*, *226*(1), 263–281.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, *33*(3), 1065–1076. doi:10.1137/S1064827503427741

Rahman, S. & Xu, H. (2004). A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics. *Probabilistic Engineering Mechanics*, *19*(4), 393–408. doi:http://dx.doi.org/10.1016/j.probengmech.2004.04.003

Riekert, P. & Schunk, T. (1940). Zur fahrmechanik des gummibereiften kraftfahrzeugs. *Ingenieur-Archiv*, *11*(3), 210–224.

Saltelli, A. [A.], Annoni, P., Azzini, I., Campolongo, F., Ratto, M., & Tarantola, S. (2010, February). Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, *181*(2), 259–270. doi:10.1016/j.cpc.2009.09.018

Saltelli, A. [A.], Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., ... Tarantola, S. (2008). *Global sensitivity analysis: the primer* (L. John Wiley & Sons, Ed.). Wiley Online Library.

Saltelli, A. [Andrea], Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity analysis in practice: a guide to assessing scientific models*. John Wiley & Sons, Ltd.

Schoutens, W. (2000). *Stochastic processes and orthogonal polynomials* (1st ed.). Springer Verlag.

Shields, M., Teferra, K., Hapij, A., & Daddazio, R. (2013). Bootstrap monte carlo method utilizing a refined stratified sampling routine for uncertainty quantification of very large transient dynamic computational models. In *International conference on structural safety and reliability*.

Smolyak, S. A. (1963). Quadrature and interpolation formulas for tensor products of certain classes of functions. *Doklady Akademii nauk SSSR*, *4*, 240–243.

Sobol, I. M. (2001, February). Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, *55*(1–3), 271–280. doi:10.1016/S0378-4754(00)00270-6

Stadler, C. (2014–2015). *Anwendung von polynomial chaos zur quantitativen bewertung von parameterunsicherheiten in der fahrdynamiksimulation* (Bachelor's Thesis, Technische Universität München, Fakultät für Maschinenwesen).

Sudret, B. (2008, July). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, *93*(7), 964–979. doi:10.1016/j.ress.2007.04.002

Szegö, G. (1939). *Orthogonal polynomials*. American Mathematical Society, Colloquium Publications.

Wiener, N. (1938, October). The homogeneous chaos. *American Journal of Mathematics*, *60*(4), 897–936.

Xiu, D. (2010). *Numerical methods for stochastic computations - a spectral methods approach*. Princeton University Press.

Xiu, D. & Karniadakis, G. (2002). The wiener–askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, *24*(2), 619–644.

Xu, H. & Rahman, S. (2004). A generalized dimension-reduction method for multidimensional integration in stochastic mechanics. *International Journal for Numerical Methods in Engineering*, *61*(12), 1992–2019. doi:10.1002/nme.1135

Appendix A

UQTk Functions

The UQTk functions that can be found in version 2.1.1 and that were used in this work are presented in table A.1.

Table A.1: UQTk functions used in this work

`qrule.m`	Gauss quadrature rules for the computation of 1D nodes and x_j weights w_j
`uq_evalBasisNormsSquared.m`	Computes the inner product of the basis polynomials $\langle \Psi_i^2 \rangle$
`uq_get1DNodesAndWeights.m`	Set up 1D nodes x_j and weights w_j for the Gaussian Quadrature
`uq_getNISP.m`	Returns the NISP matrix, used to project a vector of data to a PC basis
`uq_getSensitivity.m`	Computes first or second order Sobol' sensitivity indices
`uq_getTotalSensitivity.m`	Computes total Sobol' sensitivity indices
`uq_initMultiIndex.m`	Computes the multi-indices α_k
`uq_meanvar.m`	Computes mean and variance of a time series
`uq_PCBasis.m`	Computes multidimensional basis polynomials $\Psi_i(x_j)$ at node x_j
`uq_pcset.m`	Sets up the basic PCE parameters
`uq_psi.m`	Computes 1D basis polynomials Ψ_i
`uq_quadrature.m`	Generation of the multidimensional integration nodes and weights by using the tensor product rule from their unidimensional counterpart
`uq_quadtable.m`	Sets up lookup table of quadrature nodes
`uq_sample.m`	Samples a PCE by using the spectral coefficients

Table A.2 presents further methods implemented in this work to improve or adapt the capabilities of UQTk.

Table A.2: Functions implemented or adapted in this work for their use with UQTk

`plotPDF.m`	Computes the KDE for the PDF (using MATLAB's `ksdensity`) of the input variable at the specified time in the time series and plots the result
`regressionPCE.m`	Computes the necessary matrices of the Linear Regression Method for the computation of the PCE spectral coefficients
`samplePCE.m`	Samples the PCE at the specified time (uses `uq_sample.m`)
`sg_quadrature.m`	Adapted from original code by Heiss & Winschel for its use with UQTk, returns the nodes and weights for sparse grid integration
`sobolIndex.m`	Generalization of `uq_getSensitivity.m` for the computation of nth-order Sobol' sensitivity indices

Appendix B

Additional figures

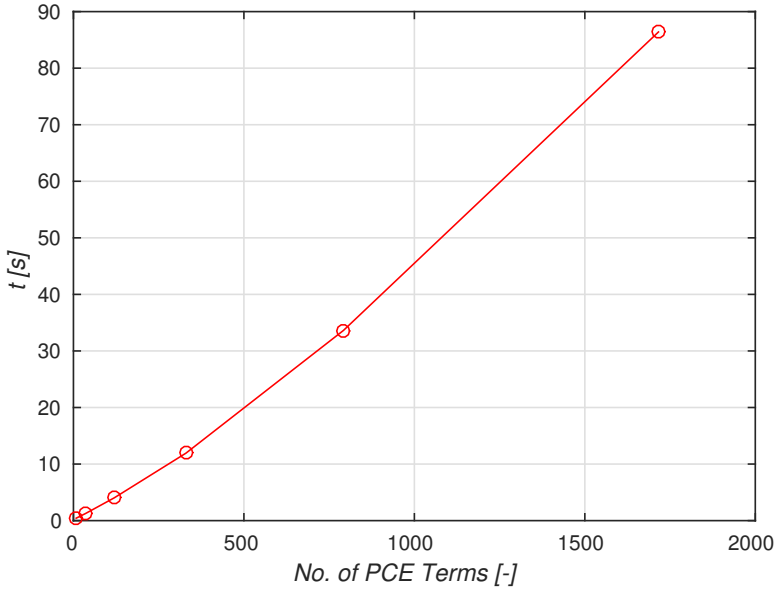


Figure B.1: Computation time vs. Number of PCE terms for the STM using SGI (Kronrod-Patterson rule) for the computation of the PCE coefficients

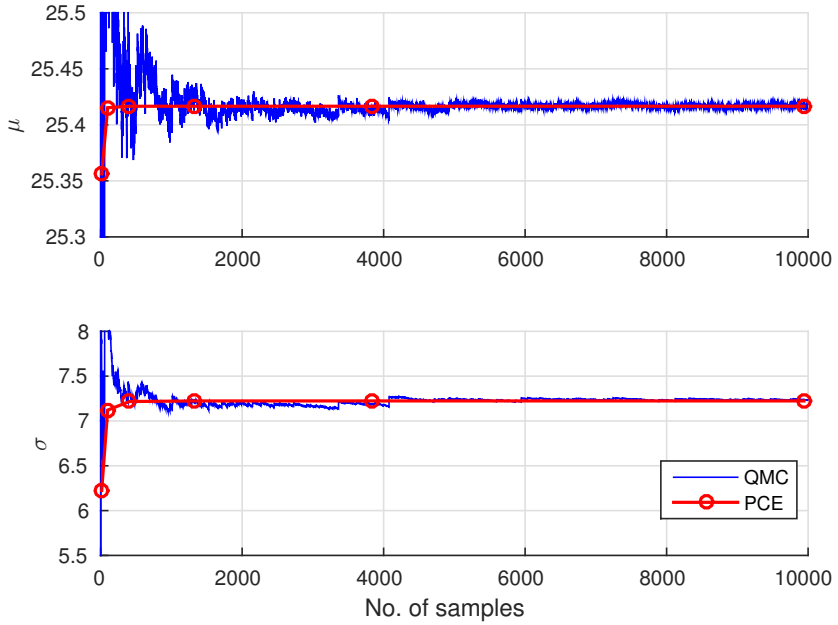


Figure B.2: PCE vs. QMC, convergence of the mean and standard deviation for $\dot{\psi}$ at $t = 2.5\ s$

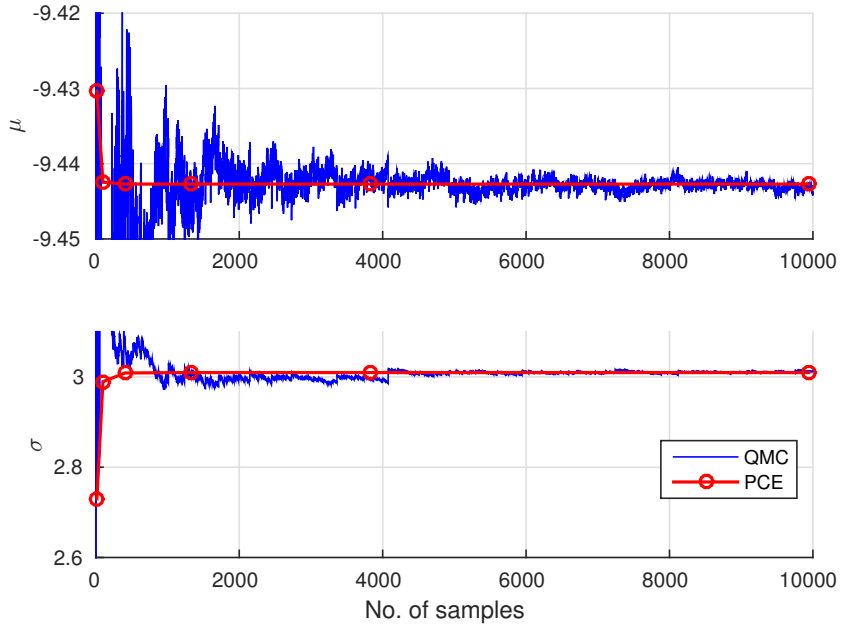


Figure B.3: PCE vs. QMC, convergence of the mean and standard deviation for β at $t = 2.5\ s$

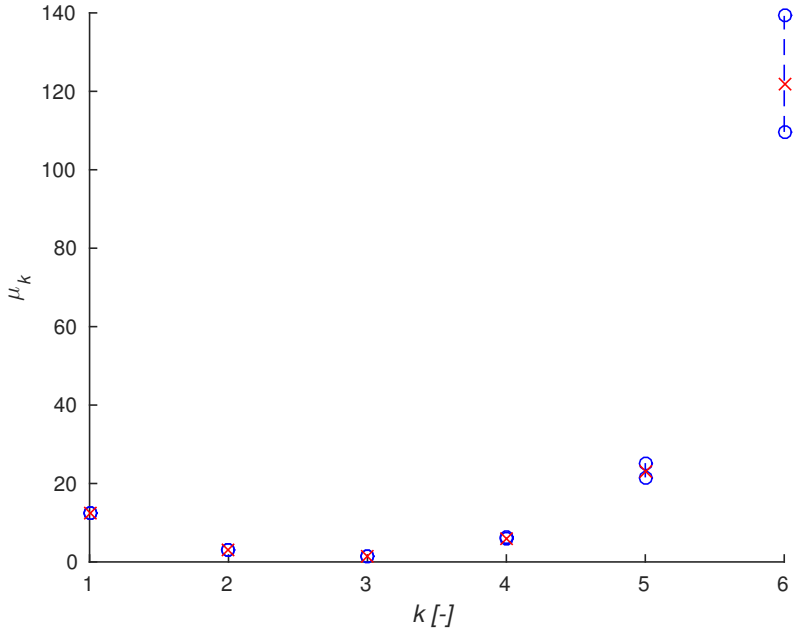


Figure B.4: Confidence Intervals for the moments of a_y at $t = 2.5$ s for the STM

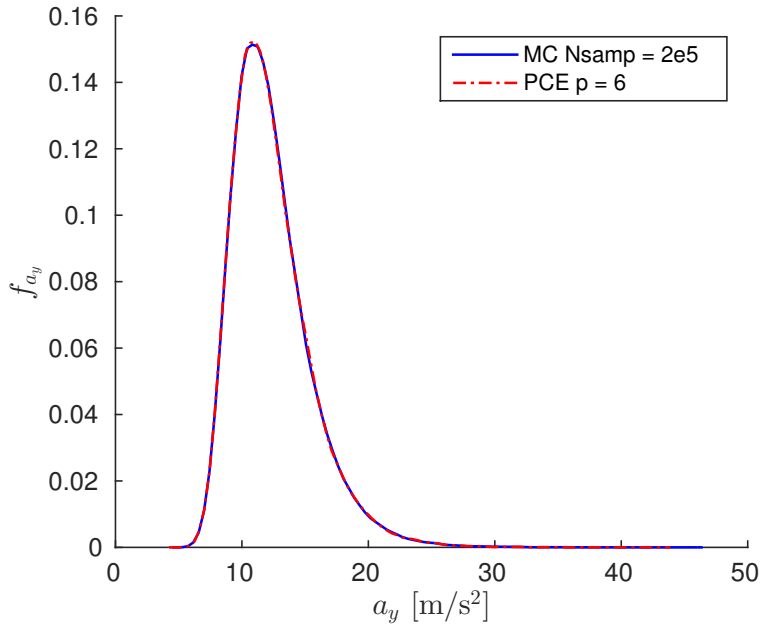


Figure B.5: KDE of a_y at $t = 2.5$ s for the STM

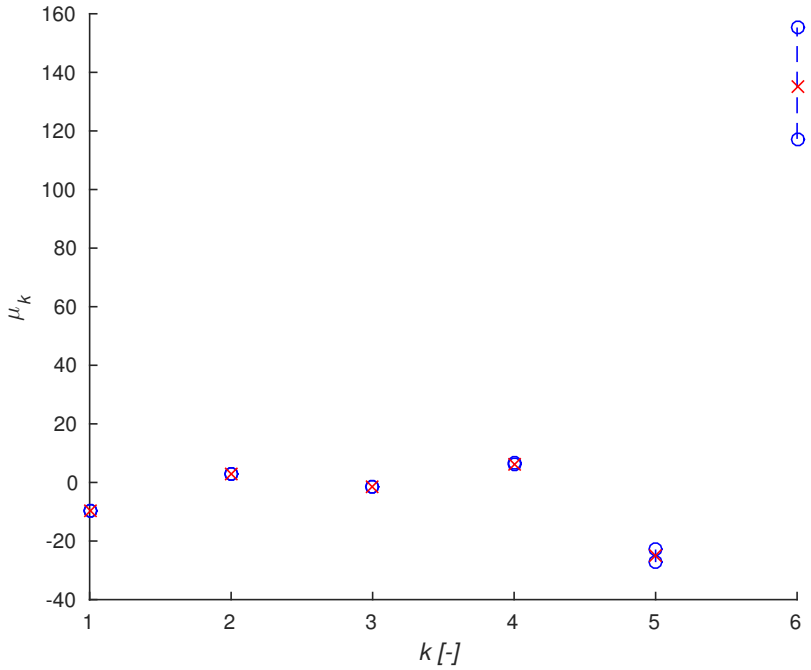


Figure B.6: Confidence Intervals for the moments of β at $t = 2.5\ s$ for the STM

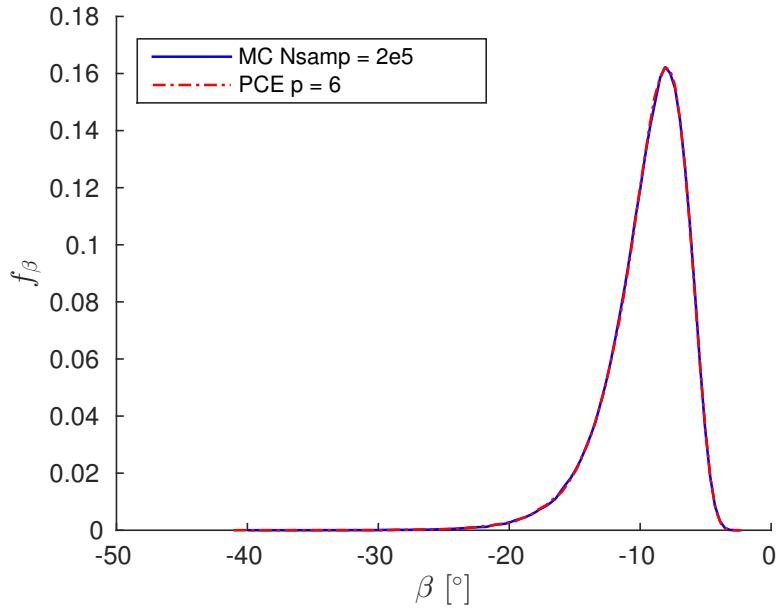


Figure B.7: KDE of β at $t = 2.5\ s$ for the STM

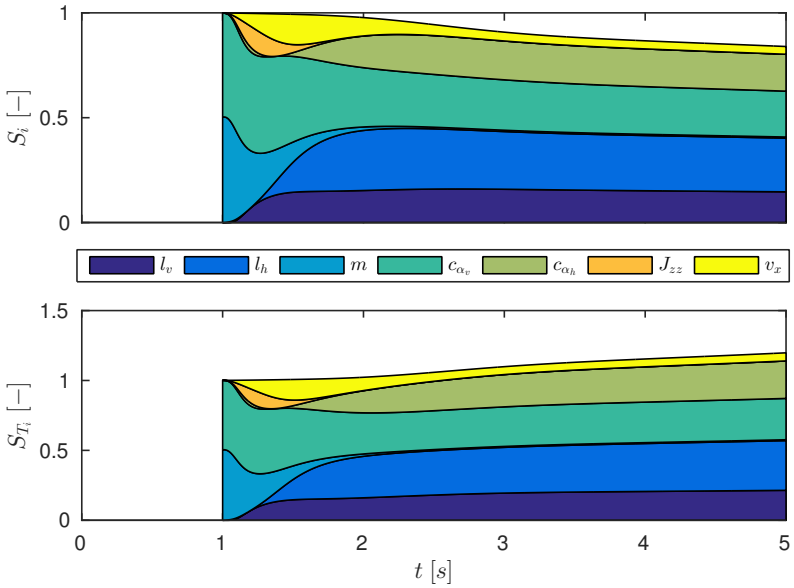


Figure B.8: PC-based first order and total Sobol' indices of a_y for the STM

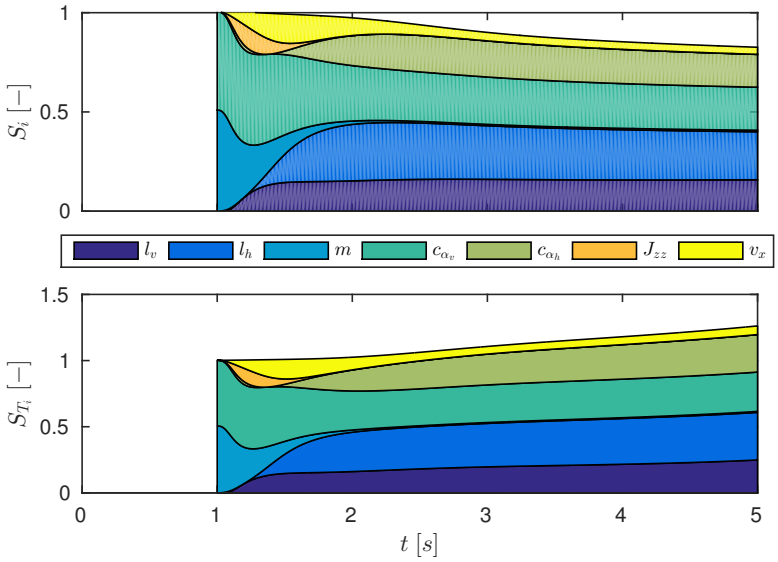


Figure B.9: Sampling-based first order and total Sobol' indices of a_y for the STM

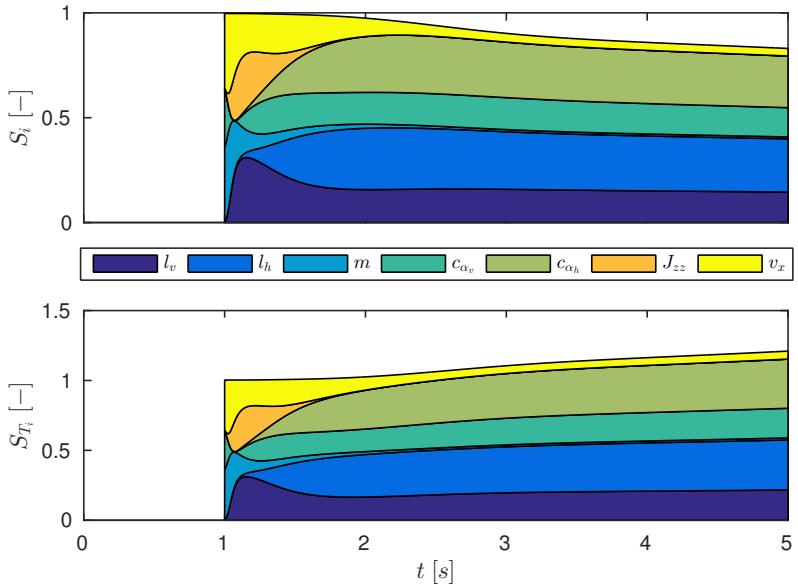


Figure B.10: PC-based first order and total Sobol' indices of β for the STM

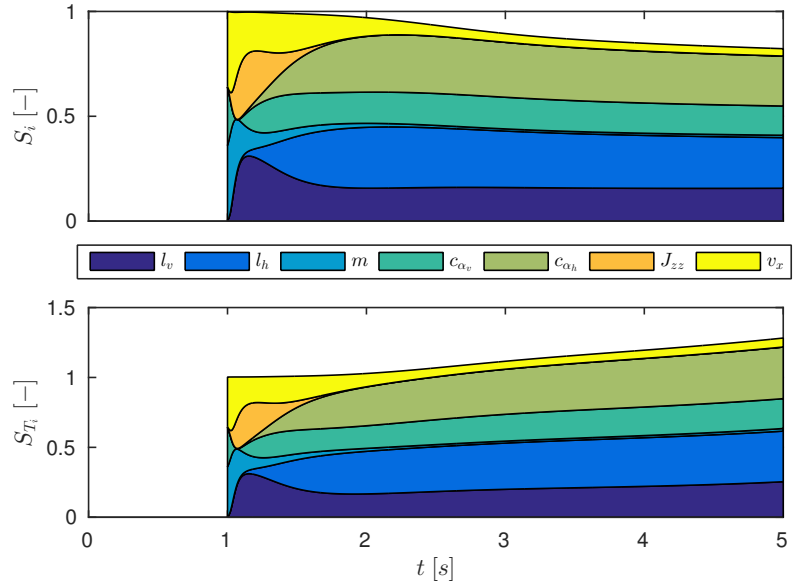


Figure B.11: Sampling-based first order and total Sobol' indices of β for the STM

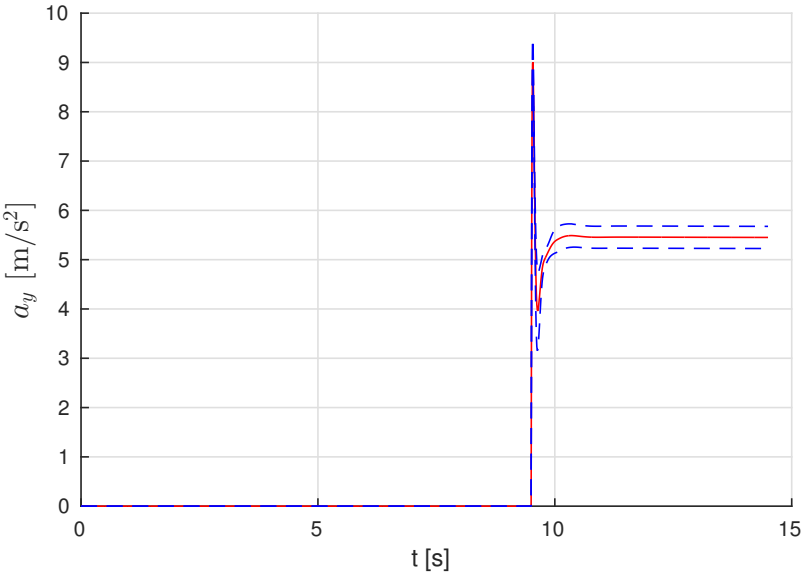


Figure B.12: Time evolution for the lateral acceleration (a_y) as computed by means of a 6th-order PCE (TTM, Step steering input maneuver)

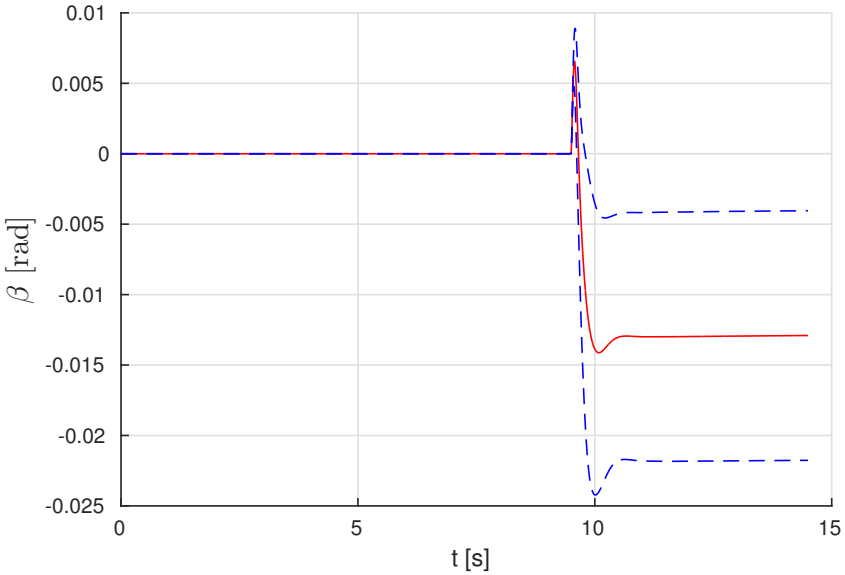


Figure B.13: Time evolution for the sideslip angle (β) as computed by means of a 6th-order PCE (TTM, Step steering input maneuver)

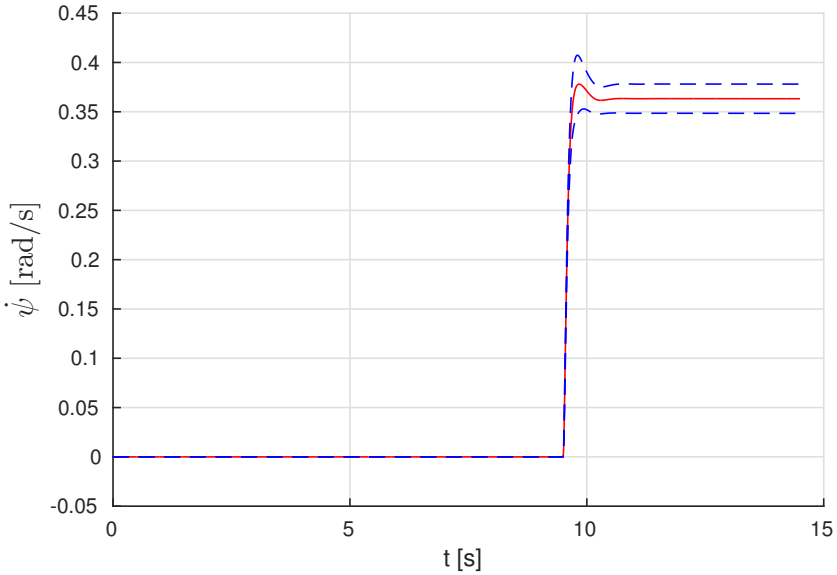


Figure B.14: Time evolution for the yaw rate ($\dot{\psi}$) as computed by means of a 6th-order PCE (TTM, Step steering input maneuver)

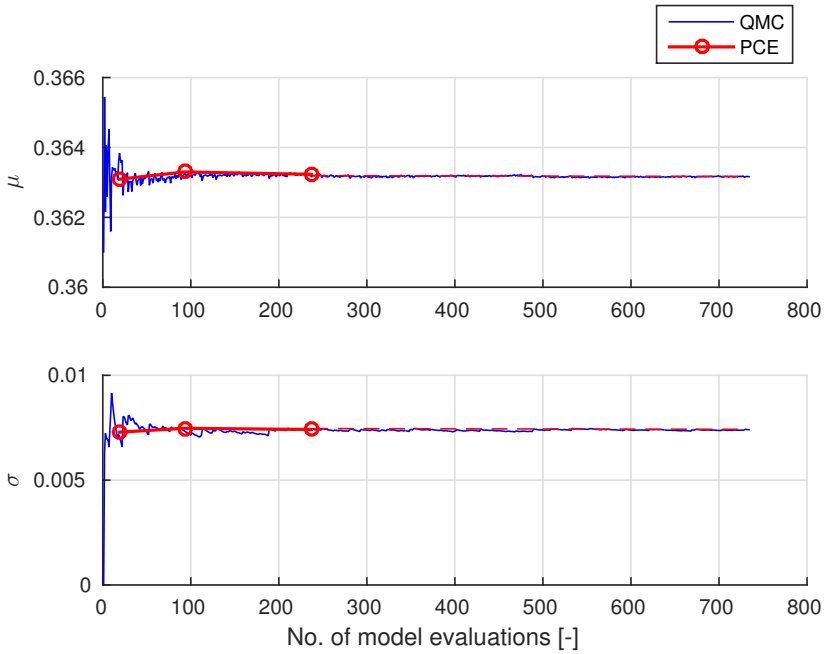


Figure B.15: PCE vs. QMC convergence of the mean and standard deviation for the yaw rate $\dot{\psi}$ at $t = 14\ s$ (TTM, Step steering input maneuver)

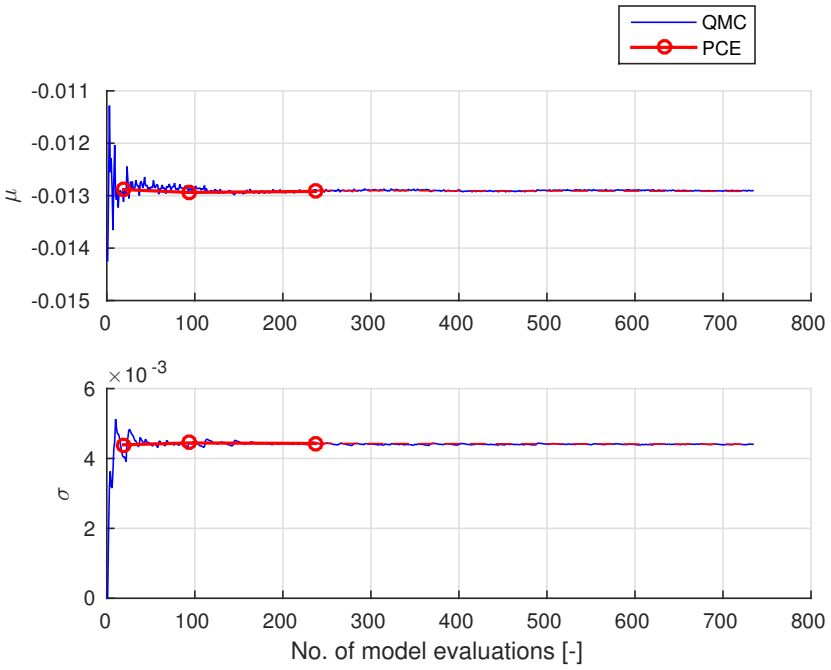


Figure B.16: PCE vs. QMC convergence of the mean and standard deviation for the sideslip angle β at $t = 14\ s$ (TTM, Step steering input maneuver)

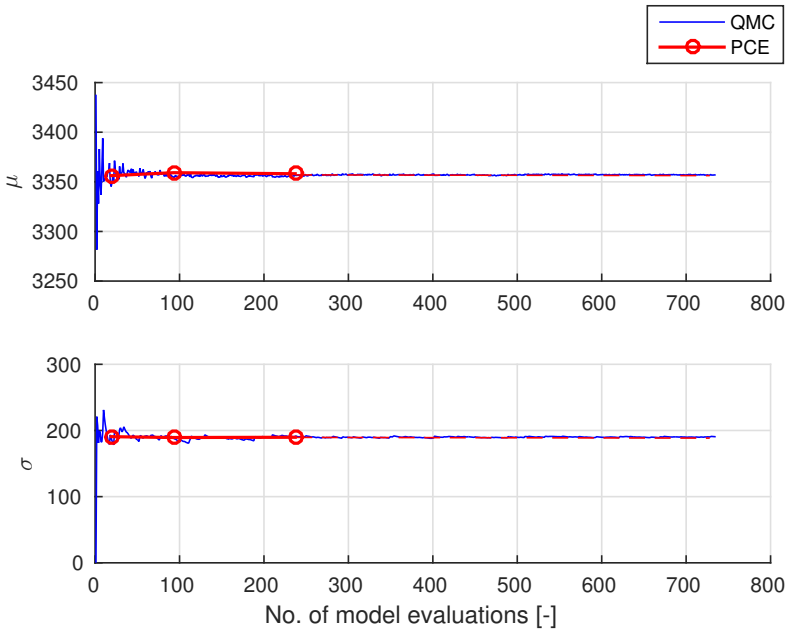


Figure B.17: PCE vs. QMC convergence of the mean and standard deviation for the lateral force of the front right wheel F_y^{vr} at $t = 14\ s$ (TTM, Step steering input maneuver)

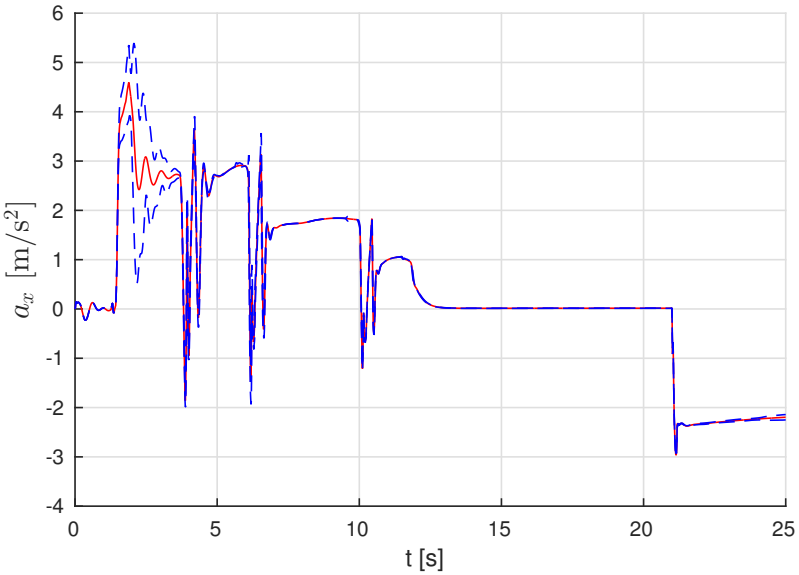


Figure B.18: Time evolution for the longitudinal acceleration (a_x) as computed by means of a 5th-order PCE (TTM, Braking in a curve maneuver)

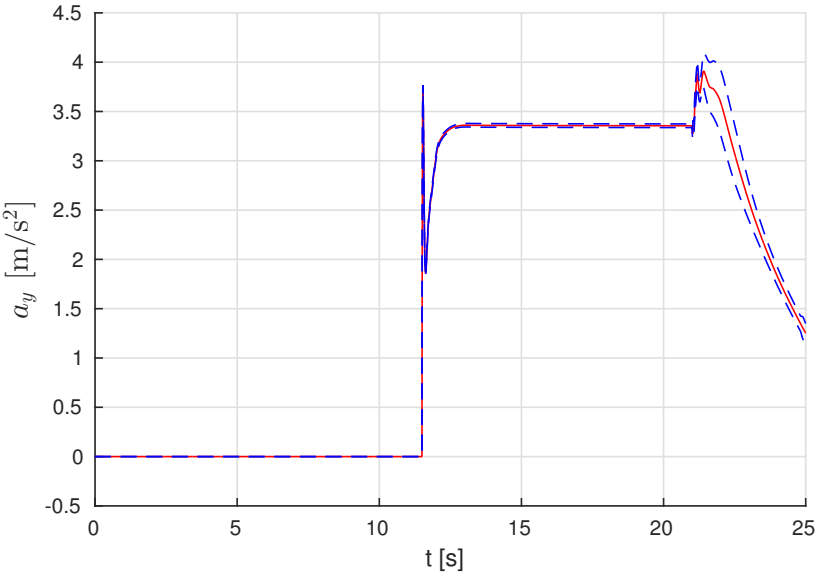


Figure B.19: Time evolution for the lateral acceleration (a_y) as computed by means of a 5th-order PCE (TTM, Braking in a curve maneuver)

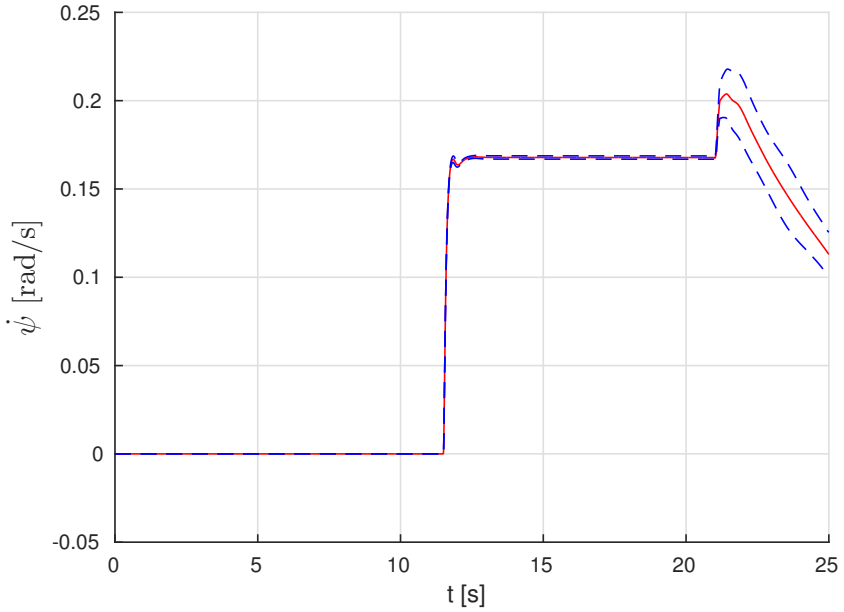


Figure B.20: Time evolution for the yaw rate ($\dot{\psi}$) as computed by means of a 5th-order PCE (TTM, Braking in a curve maneuver)

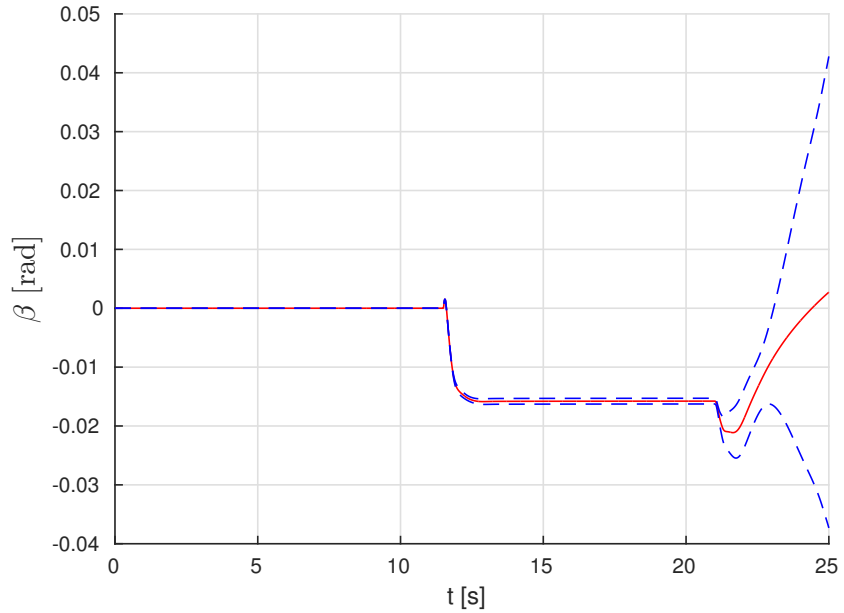


Figure B.21: Time evolution for the sideslip angle (β) as computed by means of a 5th-order PCE (TTM, Braking in a curve maneuver)

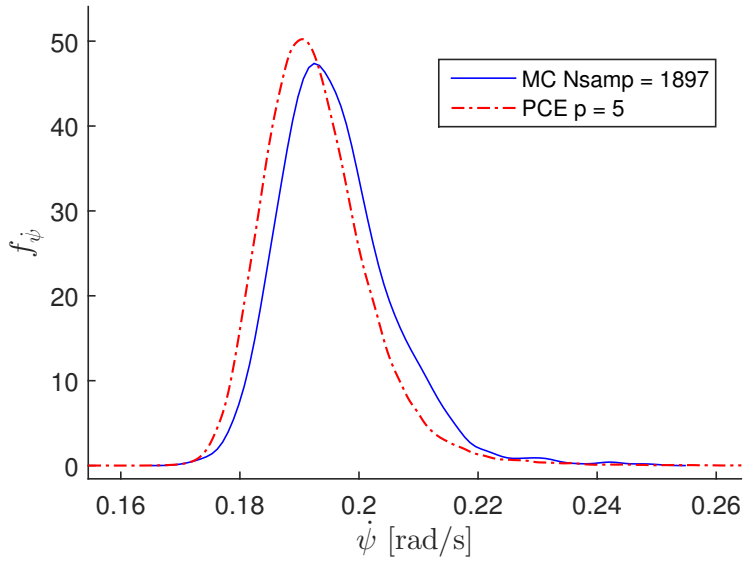


Figure B.22: PCE vs. MC KDE for the yaw rate ($\dot{\psi}$) at $t = 22\ s$ (TTM, Braking in a curve maneuver)