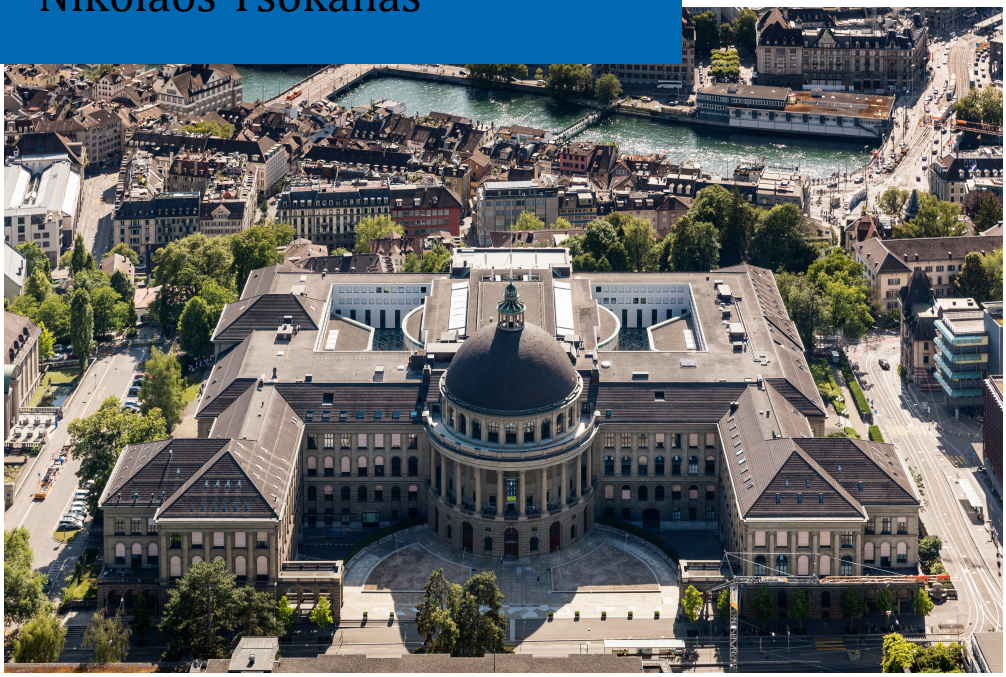
ETH zürich

Real-Time and Stochastic Hybrid Simulation

Nikolaos Tsokanas



Diss.-No. ETH 27957

D BAUG

NIKOLAOS TSOKANAS

REAL-TIME AND STOCHASTIC HYBRID SIMULATION

DISS. ETH NO. 27957

REAL-TIME AND STOCHASTIC HYBRID SIMULATION

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

NIKOLAOS TSOKANAS

Dipl., University of Patras

born on 03.07.1993

citizen of

Greece

accepted on the recommendation of

Prof. Dr. Božidar Stojadinović, examiner
Prof. Dr. Eleni Chatzi, co-examiner
Prof. Dr. Shirley Dyke, co-examiner
Dr. Roland Pastorino, co-examiner
Prof. Dr. David Wagg, co-examiner

2021

DOI 10.3929/ethz-b-000504284
ISBN 978-3-907234-67-9

Front cover page image: Copyrights in © ETH Zürich / Alessandro Della Bella. Recovered from:
https://ethz.ch/en/news-and-events/media-information/images/standort-eth-zentrum.html

To my parents,
Kyriaki and Panagiotis

Acknowledgements

This dissertation outlooks my activities as a doctoral student at the Chair of Structural Dynamics and Earthquake Engineering at the Institute of Structural Engineering (IBK) of the ETH Zurich, in Switzerland.

First and foremost I would like to thank my supervisor, Prof. Božidar Stojadinović, for offering me the opportunity to join his Chair and be a member of his research group. I am grateful for the kind support and trust as well as for the scientific advice that I received from him throughout these years. Additionally, I would like to thank the co-examiners of my doctoral examination committee, namely Prof. Eleni Chatzi, Prof. Shirley Dyke, Roland Pastorino and Prof. David Wagg, for reviewing my thesis, along with the chairperson of the committee Prof. Ioannis Anastasopoulos. Additionally, I sincerely thank Prof. Giuseppe Abbiati for his assistance and the valuable discussions that we had, which helped me to better understanding the principles of hybrid simulation and uncertainty quantification.

In addition, I gratefully acknowledge the funding from the DyVirt (Dynamic Virtualization: modeling performance of engineering structures) program. DyVirt is a project funded from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 764547. The sole responsibility of this publication lies with the author. The European Union is not responsible for any use that may be made of the information contained herein.

Within the DyVirt project I had the chance to conduct part of my research work in the Dynamics Research Group at the University of Sheffield, in the UK and in the Model-based System Testing group at Siemens Digital Industries Software, in Belgium. In the aforementioned groups, I was co-supervised by Prof. David Wagg and Roland Pastorino, respectively. Their constant availability to support and advise me, helped me overcome numerous challenges I faced within my research project. I am grateful for their kind supervision.

I would also like to thank the team of the IBK Structural Testing Laboratory at ETH Zurich, and in particular Dominik Werne and Pius Herzog, for their interest in the research project and for assisting me in the conducted experimental campaigns.

Furthermore, I would also like to thank all the friends and colleagues from the groups of Prof. Stojadinović, Prof. Chatzi and Prof. Sudret as well as the

members of the Dynamics Research Group at the University of Sheffield and the Model-based System Testing group at Siemens Digital Industries Software for their helpful discussions and the successful collaborations.

Finally, I want to express my gratitude to my family and my partner, Erifyli, for their continuous encouragement, patience and support throughout my doctoral studies.

Zurich, October 2021

Tsokanas Nikolaos

Abstract

It is common practice that prior to any engineering decision a high-fidelity model is required to provide information regarding the performance of the investigated system and to account for any potential risks embedded in its design. Building trusted virtual models for structures subjected to dynamic loads is a process known as dynamic virtualization. Nevertheless, development of such high-fidelity models premises model validation and calibration. In this regard, hybrid simulation has emerged as a cost-effective method that can be used to develop trusted virtual models. In hybrid simulation, the response of a prototype system subjected to realistic excitation is obtained by combining numerical and physical substructures into a hybrid model of the prototype. It is established as an alternative technique to full-scale physical testing for studying the dynamic response of a structural system.

The coupling of numerical and physical substructures formulates the hybrid model and is achieved by means of transfer systems. When hybrid simulation is conducted on a distorted time scale the dynamic response of the system under consideration may be biased. This is most acute for hybrid models encompassing loading-rate-sensitive substructures. Nonetheless, trust in hybrid simulation outcomes is paramount and therefore conducting hybrid simulation in real-time is essential. Albeit attractive, real-time hybrid simulation is a complex procedure incorporating many challenges that can compromise the simulation fidelity.

Additionally, in current practices, substructures and excitation that characterize the hybrid model are often considered to be deterministic. However, nominally identical specimens are, in practice, never actually identical, while excitation is usually stochastic. Furthermore, uncertainties are always present in experimental campaigns and their effects may be significant on the dynamic response of the examined structural system. Therefore, there is an increased necessity to understand how uncertain structural systems perform under uncertain operating conditions.

The subject of this research project is to conduct real-time and stochastic hybrid simulations. To conduct real-time hybrid simulations, compensation techniques are applied to eliminate time delays originating both from the physical and numerical substructures, enabling thus high-fidelity hybrid simulation results. Classical and adaptive model predictive control is em-

ployed to compensate for time delays introduced by the inherent dynamics of the transfer system used. Accordingly, model order reduction techniques based on polynomial chaos expansion and neural networks are utilized to reduce the order of high-dimensional nonlinear numerical substructures and hence to decrease the required computational power.

To conduct stochastic hybrid simulations, an uncertainty quantification framework is employed. It is based on the design of probabilistic hybrid models, uncertainty propagation as well as surrogate modeling and global sensitivity analysis. Several surrogate modeling techniques are investigated, namely polynomial chaos expansion, Kriging, polynomial chaos Kriging and generalized lambda models. Global sensitivity analysis utilizing Sobol' indices is performed to investigate the sensitivity of the simulation results to certain input parameters, uncovering the inner workings of the hybrid model in both epistemic and aleatory sense.

The main accomplishments presented in this dissertation are twofold. The first accomplishment corresponds to establishing new methods for conducting reliable hybrid simulations, while the second concerns on advancing the frontiers of the current state-of-the-art deterministic hybrid simulation practices to account for the stochastic nature of uncertain structural systems, that is how associated uncertainties propagate and affect the performance of the structural system. Therefore, the major contribution of this dissertation is to provide a high-fidelity simulation tool that can be used to apply uncertainty quantification techniques to examine the dynamic response of uncertain systems operating under uncertain conditions, with the overreaching goal of dynamic validation and calibration of models used in engineering design.

Zusammenfassung

Es ist gängige Praxis, dass vor jeder technischen Entscheidung ein realitätsgetreues Modell erforderlich ist, um Informationen über die Leistung des untersuchten Systems zu erhalten sowie mögliche Risiken in der Konstruktion zu berücksichtigen. Die Erstellung zuverlässiger virtueller Modelle für Strukturen, die dynamischen Belastungen ausgesetzt sind, ist ein Prozess, der als dynamische Virtualisierung bekannt ist. Die Entwicklung solcher Modelle setzt jedoch eine Modellvalidierung und -kalibrierung voraus. In diesem Zusammenhang hat sich die hybride Simulation als kosteneffiziente Methode erwiesen, mit der zuverlässige virtuelle Modelle entwickelt werden können. Bei der hybriden Simulation wird die Reaktion eines Prototyps unter realistische Anregung, durch die Kombination numerischer und physikalischer Teilstrukturen in einem hybriden Modell des Prototyps ermittelt. Sie hat sich als Alternative zu physikalischen Tests in Originalgröße für die Untersuchung der dynamischen Reaktion eines strukturellen Systems etabliert.

Das Hybridmodell besteht aus der Kopplung von numerischen und physikalischen Unterstrukturen mit der Hilfe von Übertragungssystemen. Wenn die hybride Simulation auf einer verzerrten Zeitskala durchgeführt wird, kann die dynamische Antwort des betrachteten Systems verzerrt werden. Dies ist bei Hybridmodellen, die belastungsempfindliche Unterstrukturen umfassen, besonders akut. Nichtsdestotrotz ist das Vertrauen in die Ergebnisse der hybriden Simulation von größter Bedeutung, weshalb die Durchführung der hybriden Simulation in Echtzeit unerlässlich ist. Die hybride Echtzeitsimulation ist zwar attraktiv, aber ein komplexes Verfahren mit vielen Herausforderungen, die die Simulationstreue beeinträchtigen können.

Darüber hinaus werden in der gegenwärtigen Praxis die Unterstrukturen und die Erregung, die das Hybridmodell charakterisieren, oft als deterministisch angesehen. Nominell identische Proben sind jedoch in der Praxis nie wirklich identisch, während die Erregung in der Regel stochastisch ist. Darüber hinaus sind Unsicherheiten in experimentellen Kampagnen immer vorhanden, und ihre Auswirkungen auf die dynamische Reaktion des untersuchten Struktursystems können erheblich sein. Daher ist es umso wichtiger zu verstehen, wie sich ungewisse Struktursysteme unter ungewissen Betriebsbedingungen verhalten.

Gegenstand dieses Forschungsprojekts ist die Durchführung von Echtzeit- und stochastischen Hybridsimulationen. Um hybride Echtzeitsimulationen durchzuführen, werden Kompensationstechniken angewandt, um Zeitverzögerungen zu eliminieren, die sowohl von den physikalischen als auch von den numerischen Substrukturen herrühren, und so hochgenaue hybride Simulationsergebnisse zu ermöglichen. Klassische und adaptive modellprädiktive Steuerungen werden eingesetzt, um Zeitverzögerungen zu kompensieren, die durch die inhärente Dynamik des verwendeten Übertragungssystems entstehen. Dementsprechend werden Techniken zur Reduzierung der Modellordnung auf der Grundlage polynomialer Chaos-Expansion und neuronaler Netze eingesetzt, um die Ordnung hochdimensionaler nichtlinearer numerischer Unterstrukturen zu reduzieren und damit die erforderliche Rechenleistung zu verringern.

Zur Durchführung stochastischer hybrider Simulationen wird ein Rahmenwerk zur Quantifizierung der Unsicherheit eingesetzt. Es basiert auf dem Entwurf von probabilistischen Hybridmodellen, der Unsicherheitsfortpflanzung sowie der Surrogatmodellierung und der globalen Sensitivitätsanalyse. Es werden mehrere Surrogat-Modellierungstechniken untersucht, nämlich polynomiale Chaos-Expansion, Kriging, polynomiales Chaos-Kriging und verallgemeinerte Lambda-Modelle. Eine globale Sensitivitätsanalyse unter Verwendung von Sobol-Indizes wird durchgeführt, um die Empfindlichkeit der Simulationsergebnisse gegenüber bestimmten Eingangsparametern zu untersuchen und die innere Funktionsweise des Hybridmodells sowohl im epistemischen als auch im aleatorischen Sinne aufzudecken.

Die wichtigsten Errungenschaften dieser Dissertation sind zwei. Erstens die Entwicklung neuer Methoden für die Durchführung zuverlässiger hybrider Simulationen, und zweitens, die Erweiterung der Grenzen des derzeitigen Stands der Technik bei deterministischen hybriden Simulationsverfahren, um die stochastische Natur ungewisser struktureller Systeme zu berücksichtigen, d. h. wie sich die damit verbundenen Unsicherheiten ausbreiten und die Leistung des strukturellen Systems beeinflussen. Daher besteht der Hauptbeitrag dieser Dissertation darin, ein High-Fidelity-Simulationswerkzeug bereitzustellen, das zur Anwendung von Unsicherheitsquantifizierungstechniken verwendet werden kann, um die dynamische Reaktion von ungewissen Systemen zu untersuchen, die unter ungewissen Bedingungen arbeiten, mit dem übergreifenden Ziel der dynamischen Validierung und Kalibrierung von Modellen, die in der technischen Planung verwendet werden.

Table of Contents

List of Figures

List of Tables

Nomenclature

AIC	Akaike Information Criterion
AMPC	Adaptive Model Predictive Control
BIC	Bayesian Information Criterion
CDF	Cumulative Distribution Function
DMD	Dynamic Mode Decomposition
CMS	Component-Mode Synthesis
CS	Case Study
CSI	Control-Structure-Interaction
CV	Coefficient of Variation
DAQ	Data Acquisition
DoF	Degrees of Freedom
eCVT	electrically Continuously Variable Transmission
ED	Experimental Design
EoM	Equation of Motion
FFNN	Feedforward Neural Networks
FOM	Full-Order Model
FORM	First-Order Reliability Method
GLAM	Generalized Lambda Model
GLD	Generalized Lambda Distribution
GSA	Global Sensitivity Analysis
HHT	Hilber-Hughes-Taylor
HiL	Hardware-in-the-Loop
HS	Hybrid Simulation
IR lamp	Infrared lamp
LAR	Least Angle Regression
LHS	Latin Hypercube Sampling
LOO	Leave-One-Out
LQR	Linear Quadratic Regulator

LTI	Linear Time-Invariant
MiL	Model-in-the-Loop
MIMO	Multi-Input-Multi-Output
MISO	Multi-Input-Single-Output
MOR	Model Order Reduction
MPC	Model Predictive Control
MRD	Magnetorheological Damper
NRMSE	Normalized Root Mean Square Error
NS	Numerical Substructure
ODE	Ordinary Differential Equations
OLS	Ordinary Least-Squares
OS	Operator-Splitting
PCA	Principal Component Analysis
PCE	Polynomial Chaos Expansion
PCK	Polynomial Chaos Kriging
PDF	Probability Density Function
PGD	Proper Generalized Decomposition
POD	Proper Orthogonal Decomposition
PS	Physical Substructure
QoI	Quantity of Interest
QP	Quadratic Programming
RK4	Runge-Kutta fourth-order
RO	Research Objective
RO-NS	Reduced-Order Numerical Substructure
RTHS	Real-Time Hybrid Simulation
SiL	Software-in-the-Loop
SIMO	Single-Input-Multi-Output
SISO	Single-Input-Single-Output
SORM	Second-Order Reliability Method
SVD	Singular Value Decomposition
TS	Testing Scenario
vPS	virtual Physical Substructure
vRTHS	virtual Real-Time Hybrid Simulation

V&V	Verification and Validation
WBZ	Wood-Bossak-Zienkiewicz

Chapter 1

Introduction

1.1 Motivation

It is clear that prior to any engineering decision, a high-fidelity model is needed in order to give insight into the performance of the examined structure and to account for any potential risks embedded in its design. Modeling the performance of an engineering structure that operates in a dynamic environment results in a so-called *virtual model*. Building trusted virtual models for structures subjected to dynamic loads is a process known as *dynamic virtualization*. A dominant benefit of dynamic virtualization is the ability to trust computer models for design cycle decision-making without having more complete test data from prototypes or full-scale physical testing. In fact, in the absence of such trusted models, current industrial practices are limited to either conservatively overdesign components and structures that are subjected to large dynamic loading or accept high probability of failure and uncertainty rates within the operational performance. However, both latter practices have a negative aftermath regarding cost and performance.

Through dynamic virtualization it is feasible to reduce costs and uncertainties associated with the design phase of engineering structures but also to progress on their performance and safety, resulting in more resilient and reliable systems. Additionally, with dynamic virtualization it is possible to investigate extreme scenarios, such as intense loadings and system behavior beyond failure, as these scenarios are not reasonably affordable on full-scale experimental campaigns. Furthermore, creation of trusted virtual models can be subsumed into the development of the so-called *digital twins*, as the basis of any digital twin is a numerical model, validated for a specific application context [1, 2].

For creating trusted virtual models, *verification and validation* (V&V)[1] techniques are essential. A key element of V&V is experimental testing. Currently, several well-established experimental techniques exist to investigate the dynamic response of a structural system. A widely used technique is the *quasi-static testing*. In the latter, predefined force or displacement time histories loadings are applied via an arrangement of actuators to the exam-

1 A more comprehensive review of V&V is provided in Chapter 2.

ined structure. This technique is beneficial since the hysteretic behavior of structural elements along with characteristics of the examined structure, such as loading-rate dependencies and material properties, can be explored at relatively low economical cost and effort. Nonetheless, such types of tests are restricted to the nature of the predefined imposed loadings and hence they typically fail to adequately capture the dynamic response of a structural system subjected to dynamic excitation.

An alternative to quasi-static testing is the *full-scale dynamic testing* of an entire structural system, e.g. shaking table testing in earthquake engineering. With these kinds of tests, it is feasible to investigate the structural behavior of a system at extreme scenarios like yielding, in the event of geometrical nonlinearities as well as beyond failure, taking into account the structure's inertia effects and energy dissipation properties. Outcomes of such tests can sufficiently resemble realistic scenarios that could occur in real-world applications. Nevertheless, conducting full-scale experimental testing is associated with large costs regarding constructing the physical specimen. Furthermore, limitations in laboratory capacity and equipment add a further barrier, increasing the challenges and complexity related to such tests. A promising workaround is the scaled specimen testing adhering to the respective rules of dynamic similitude [3]. Although scaled tests can accurately represent the overall response of the respective full-scale structure, local effects such as geometric imperfections and nonlinearities, able to significantly contribute to the structure's response, are not captured precisely. Therefore, the fidelity of scaled testing is questioned.

Another experimental method is the so-called *hybrid simulation*. Hybrid simulation (HS) is a method to obtain the response of a prototype system subjected to realistic excitation by combining numerical and physical substructures into a hybrid model of the prototype and refers to the simulation setup and simulation procedures involved, including individual elements of the hybrid model and response simulations. It is established as a high-fidelity tool for studying the dynamic response of a structural system [4, 5, 6], combining the advantages of numerical simulations with the realism of experimental testing. HS may be characterized as an advanced actuator-based testing scheme in which the imposed loading is not predefined, but determined online during the simulation process, for a structural system subjected to a particular excitation. Additionally, it can be used to explore a structural component beyond its linear regime without testing an entire structural assembly. As a result, the cost of experimentation is substantially reduced. Although beneficial, HS is also linked with several challenges as addressed later on. HS constitutes the core of this dissertation as it can be employed to satisfy the overarching goal; to support the validation and

calibration of virtual models. Chapter 2 offers a thorough review on HS fundamentals.

In all the above experimental methodologies, it is often the case that systems and related excitation are conceived as deterministic. However, in the majority of cases encountered in real-world applications, loading is stochastic, while the systems' parameters may be highly uncertain. An exhaustive exploration of all possible load cases is not an option given the experimental cost associated with a single experiment evaluation. Rather than that, present experimental campaigns focus on studying the dynamic response of structural systems only to a limited set of variant excitation. Although such an approach could unveil the performance of the structure beyond failure, it is not sufficient to estimate the structural sensitivity of the prototype structure response, i.e. to reveal what influences what within a structural system, uncovering its inner workings in both epistemic and aleatory sense. Such missing information could be valuable not only for future planning of additional experimental campaigns but also in the context of modeling and analysis choices in the overall structural design process.

The goal of this dissertation is to go beyond the classical process of designing computer-based simulation models toward hybrid computer-physical models by developing methodologies for V&V of trusted virtual models that can guarantee representative dynamic responses even in the case of uncertainties in the system parameters and/or in the system operational conditions. This research can contribute to broaden the limits of current technologies which can guarantee trust in models in more restricted circumstances.

1.2 Concept of hybrid simulation

Before introducing the concept of HS, it should be noted that across different engineering disciplines, HS may be referred to under a different terminology. In mechanical engineering for example, the term *real-time dynamic substructuring* [7] is commonly used while in earthquake engineering, the term *hybrid simulation* [8] is used more often. In multibody dynamics community, the terms *co-simulation* [9] and *model-based system testing* [10] are more dominant. In co-simulation each substructure may be simulated at different time rates resulting in the so-called *multi-rate simulation* [11]. The latter is also starting to evolve in earthquake engineering applications [12]. Nevertheless, co-simulation may consist purely of NS and not PS. Finally, the *cyber-physical empirical method* [13] and the *hardware-in-the-loop (HiL)* testing [14] are also referring to the same simulation concept as HS.

HS is a dynamic response simulation technique. The unique characteristic of HS is that the system under consideration, the reference system thereof, can be decomposed into multiple numerical (NS) and physical (PS) substructures. The coupling of NS and PS formulates the hybrid model and is achieved by means of transfer systems (a.k.a. actuation systems), which in most cases are an arrangement of electric/hydraulic actuators or servo-controlled motors. The transfer systems are responsible for continuously synchronizing the interfaces of the substructures' boundary conditions and are commanded in a closed-loop control setting. The HS outcome is based on a gradual numerical solution of the equations governing the motion of the reference system. In more detail, in each time-step of the HS, a predefined excitation is applied to the hybrid model and a time integration scheme computes the NS response due to that excitation. Subsequently, the computed NS response is imposed to the PS via the utilized transfer system, to preserve synchronization of the substructures' boundary conditions. The dynamic response of the PS is then measured via, e.g. force transducers or displacements sensors, and fed back to the time integration scheme, completing the unknown terms of the governing equations of motion of the reference system and advancing HS to the next time-step. This feedback loop continues until the end of the HS process. If the computed NS response is displacement or force, then HS is conducted under displacement or force control, respectively. In structural HS applications, displacement and/or force control are usually used. However, velocity, acceleration or torque control can also be employed, depending on the application needs.

1.2.1 *Advantages of hybrid simulation*

HS is a beneficial experimental tool. Some of its most important advantages are listed below:

1. Modeling of complex nonlinear structural components is often prone to errors due to deliberate assumptions and simplifications made in the process. In HS it is possible to choose such complex components as the PS, allowing for realistic measurements of the respective responses and thus avoiding potential modeling errors.

2. In large-scale physical testing, constructing a full-scale specimen is associated with high costs. On the contrary, in HS only the hybrid model components of interest are constructed and hence treated as the PS. Therefore, representative results are obtained at a much lower experimentation cost.

3. Full-scale specimen testing, especially large-scale testing, strongly depends on the laboratory capacity and size. With HS such requirements can dramatically decrease as it is mainly constrained from the capacity of the utilized transfer system. The latter depends on the strength of the PS.

4. HS can be conducted either at real-time or at an extended time scale (typical values can range up to 1000 slower than real-time [15]). Therefore, HS can facilitate both quasi-static and dynamic testing. Additionally, because of the ability to conduct tests at a slower rate, it is possible to examine in more detail the damage evolution occurring in the structure, particularly for events near collapse, and thus unveil insights of the system's behavior.

5. HS can be conducted either locally, engaging a single laboratory facility, or geographically distributed [16], involving several experimental centers. In this case, the individual substructures' boundary conditions are synchronized via an internet connection. A benefit of such testing is the usage of several laboratories, offering a range of different capabilities and capacities.

6. In industrial manufacturing, e.g. automotive industry, HS can significantly contribute towards system integration de-risking. Quality checks can be performed at the component level yet emulating the overall system but risking to damage only the tested component, the PS, and not the overall reference system.

1.2.2 *Challenges of hybrid simulation*

Despite HS being a beneficial experimental tool, it is associated with various challenges, many of which rest in today's research frontiers both in academic and industrial domains. Some of these are outlined below:

1. Fidelity and trust in HS outcomes are paramount. Yet there exist numerous sources of errors and uncertainties which are potential to arise in a HS, risking falsifying the obtained results. Therefore, awareness and minimization of such errors and uncertainties are necessary. Chapter 2 presents a source overview of the latter.

2. One advantage of HS, as previously mentioned, is the ability to conduct simulations at different time scales. However, in cases that the hybrid model consists of loading-rate-sensitive substructures, it is of great importance to conduct HS in real-time. Real-time hybrid

simulation (RTHS) is an extension of HS, in which the dynamic boundary conditions at the interfaces between NS and PS are being synchronized in real-time [17]. Nevertheless, RTHS is correlated with barriers risking distorting the time scale and outcomes of HS and thus its fidelity may be questioned. These are:

a) The inherent dynamics of the transfer system, utilized to couple NS and PS, introduce time delays along with tracking errors. In slow HS, the latter may be negligible but in RTHS they could significantly alter the system's dynamic response. To tackle such issues advanced control strategies are employed.

b) The computational power needed to sufficiently compute the NS responses can contribute considerably to the time scale of HS. This is most acute for hybrid models encompassing high-dimensional nonlinear NS. In particular, to capture the dynamic behavior of such numerical models, it is often the case to further decrease the time-step of the numerical solver. However, the smaller the time-step of the simulation, the larger the computational power needed to compute it. In RTHS, when the required computational power becomes larger than the actual simulation time, time delays are introduced to the hybrid model. Hence, in such cases, the only viable solution is to reduce the order of the NS to be able to conduct the HS in real-time. An alternative solution, not always relatively affordable, is the usage of high-performance computing environments consisting of multiple processors and cores.

c) The data acquisition system (DAQ) along with the transfer system, e.g. actuators, should be able to operate at least up to the same frequency as the conducting HS. Measurement delays due to low sampling frequency of the DAQ or actuator inability to meet certain velocity requirements would highly affect the HS loop.

d) On top of the above-mentioned points, geographically distributed HS are affected also by communication interruptions such as network delays and breakdowns, which are randomly present within a HS. In this regard, network emulators are investigated to further understand how such delays occur and to minimize them [18].

3. An additional challenge is the establishment of fidelity indicators, able to quantitatively appraise online and a posteriori the truthfulness

and performance of HS in terms of how well it matches the response of the reference studied system.

4. HS is strongly application dependent and each HS configuration could be a quite time-consuming procedure. Towards this direction, the middleware common frameworks *OpenFresco* [19] and *UI-SIMCOR* [20] were developed to standardize the deployment of such tests. However, to date there exist no readily available platform to conduct general purpose HS.

5. For hybrid models with stiff PS, it may be the case that a single control mode is not sufficient to conduct the HS and therefore switch between different control modes online may be required. For example, hybrid testing of specimen with large stiffness in earthquake engineering may require mixed force/ displacement control. However, such switching control strategies are yet not that mature in the HS domain.

1.2.3 *Example applications*

In this section, some examples of HS, applied across different engineering disciplines, are presented. In earthquake engineering, HS has been used for testing of large structures such as bridges, e.g. [21, 22] and multistory buildings, e.g. [23, 24, 25]. Similarly, HS has been recently proposed to test mooring lines of offshore structures for which hydrodynamic tests with sizeable scale are prohibitive, e.g. [26, 27], as well as for offshore wind turbines testing, e.g. [28, 29]. HS is gaining popularity for component-level testing also in fire engineering, e.g. [30, 31]. More recently, HS has been combined with centrifuge testing for investigating soil-structure interaction problems [32]. Additionally, HS has been also used in aerospace applications, e.g. for testing of helicopters lag dampers [33, 34]. The latter highly affect the vibrations imposed on the helicopter's fuselage but since they are complex to model, it is not trivial to evaluate their performance within the helicopter's design phase. Treating them as PS of the hybrid model could elevate this burden. Finally, HS has also been used in automotive applications, e.g. [35], as well as for docking simulation of spacecraft [36].

1.3 Research objectives

This project is part of the European Union-funded Marie Skłodowska-Curie Action Innovative Training Network (ITN) entitled "DyVirt: Dynamic virtualization; modeling performance of engineering structures" led by Prof. David Wagg at the University of Sheffield (UK) [37]. This doctoral

dissertation research aims to develop methods for hybrid simulation of the dynamic response of uncertain structural systems that quantify the uncertainties inherent to the excitation, operation, and modeling. In order to do so, the research objectives (RO) will be divided into two main sectors; *Real-Time Hybrid Simulation* and *Stochastic Hybrid Simulation*.

1.3.1 *RO1: Real-time hybrid simulation*

RTHS is paramount to ensure high fidelity of the simulations, yet poses several challenges arising both from the PS and the NS (challenges 2.a and 2.b from Section 1.2.2, respectively). Therefore, the first research objective is to develop new methodologies and mechanisms capable to conduct HS in hard real-time, increasing the fidelity of and trust in HS outcomes. These are investigated using control schemes, based on classical and adaptive model predictive control theory, as well as model order reduction techniques, based on polynomial chaos expansion and feedforward neural networks.

1.3.2 *RO2: Stochastic hybrid simulation*

The second research objective is to extend the deterministic state-of-the-art HS to stochastic HS, where the parameters of the hybrid model are treated as random variables with known probability distributions. Furthermore, stochastic HS attempts to also account for latent variables that do not appear in the known random variables but are present, making the hybrid model response uncertain. The aim is to cover a more realistic situation where the physical substructure response is not deterministic, as nominally identical specimens are, in practice, never actually identical. Propagating uncertainties and implementing sampling strategies in HS campaigns results in probability distributions of the structural response quantities of interest. The arising question is to understand which uncertain variables affect hybrid model responses the most and in which way. This will give us insight on how to improve the robustness of HS in addition to increasing both the performance and safety levels of engineering structures. The above are investigated using uncertainty quantification practices with focus on surrogate modeling techniques and global sensitivity analysis.

1.4 Relevance to science and economy

1.4.1 *Scientific significance*

This project is part of the European-Union funded ITN action "DyVirt" and aims to enable a new generation of early-stage researchers to face the urgent challenge of how to model the performance of engineering structures that operate in dynamic and uncertain environments. The overall goal of the working package in which this project belongs is to create enhanced and accelerated methods for modeling and testing protocols for V&V of models of the dynamic response of structures that incorporate inherent uncertainties of the excitation, operation, and modeling. The major contribution of this doctoral dissertation research itself will be an extension of today's deterministic approach of hybrid simulation to a more stochastic perspective giving insight to conditions and parameters that exist in experimental testing and cannot be fully explained or observed till today. On top of that, implementation of classical and adaptive model predictive control as well as model order reduction techniques in the hybrid simulation loop will be a step beyond enabling more future research in this field.

1.4.2 *Economic significance*

Damage to structures causes not only loss of human lives and disruption of lifelines, but also long-term impact on the local, regional and sometimes national and international economies. These damages could be limited if we were able to design safer infrastructures. This could be achieved if we gain a better understanding of the behavior of various classes of structures under different dynamic load types from the elastic range through to developing collapse mechanisms and failure. There is where the significance of experimental testing lies. However, experimental testing in full-scale is an expensive and not always repeatable method of conducting simulations. On the other hand, hybrid simulation combines advantages of computer simulation with the realism of experimental testing. And this applies not only to infrastructures and structural engineering but to a wider range of applications as well, e.g. aerospace, automotive engineering. Furthermore, designing experimental simulations with deterministic parameters requires a lot of tests in order to account for every single deterministic case. In this regard, stochastic simulations could be proved to be more economically effective as a single test could cover a range of deterministic parameters, reducing significantly the number of tests conducted and enabling a more accurate method for model V&V. This project aims to develop new methods

of conducting more accurate real-time and stochastic hybrid simulations taking into account uncertainties that occur in real-world applications expanding our knowledge and understanding of how physical systems behave in order to design them in a more safe and reliable manner.

1.5 Outline of the dissertation

This dissertation is structured in two parts, aligned with the two research objectives presented in Section 1.3. Chapters 3-5 focus on the first research objective, namely methodologies to conduct real-time hybrid simulation. In particular, Chapters 3, 4 address challenges arising from the physical substructures of the hybrid model (challenge 2.a from Section 1.2.2), while Chapter 5 challenges from the numerical substructures of the hybrid model (challenge 2.b from Section 1.2.2). Chapters 7, 8 focus on the second research objective, namely the development of stochastic hybrid simulations with focus on global sensitivity analysis and surrogate modeling. In more detail, the chapters of this dissertation are organized as follows:

- **Chapter 2** provides background knowledge on HS fundamentals to establish the foundation to better comprehend the work presented thereafter.
- **Chapter 3** tackles challenges arising from the physical substructures of the hybrid model and in particular from the utilized transfer system. A novel control strategy, based on model predictive control, is proposed for time delay and tracking error compensation, introduced from the inherent actuator dynamics. The proposed tracking controller is presented first, followed by two parametric case studies, which serve to validate the performance and robustness of the novel control strategy.
- **Chapter 4** extends the tracking controller presented in Chapter 3 to an adaptive scheme, namely the adaptive model predictive controller. The controller is introduced first, followed by one parametric case study, used to validate its performance.
- **Chapter 5** continues to address the first research objective but this time the focus lies on challenges arising from the numerical substructures of the hybrid model. In particular, hybrid models with high-dimensional nonlinear numerical substructures risk distorting the time scale of hybrid simulation due to the increased necessary computational power. A model order reduction framework is proposed for real-time hybrid simulation, based on polynomial chaos

expansion and feedforward neural networks. The employed techniques are introduced and a parametric case study is then used to validate the framework.

- **Chapter 6** introduces the second part of this dissertation, which is aligned with the second research objective, i.e. conducting stochastic hybrid simulations with focus on global sensitivity analysis. This chapter serves as a background introduction to uncertainty quantification techniques, mainly emphasizing on probabilistic modeling, uncertainty propagation and sensitivity analysis.

- **Chapter 7** introduces the uncertainty quantification framework for global sensitivity analysis of hybrid models with deterministic physical substructures (at this stage, it is assumed that nominally identical specimen have identical responses, plus some negligible measurement noise). Global sensitivity analysis is performed with Sobol' indices, using surrogates modeling techniques. In particular, three different surrogate modeling methods, namely polynomial chaos expansion, Kriging and polynomial chaos Kriging, are utilized. The proposed framework is firstly addressed and two parametric case studies are presented afterwards to validate it.

- **Chapter 8** extends the framework presented in Chapter 7 to the more realistic case of hybrid models with stochastic physical substructures (nominally identical specimens are, in practice, never actually identical). The framework is based on the quantile-based extension of the classical Sobol' indices, while generalized lambda models are used as surrogates to account for the random nature of the stochastic hybrid models. A parametric case study follows for the validation of the proposed framework.

- **Chapter 9** draws conclusions and highlights future research directions.

Chapter 2

Hybrid simulation background

This chapter provides an introduction to HS fundamentals to establish the foundation to better comprehend the work presented thereafter. Firstly, the hybrid simulation components are addressed and then a short introduction is taking place on uncertainties and errors that are present in HS. Finally, a brief overview of model V&V techniques takes place.

2.1 Hybrid simulation components

To conduct a successful HS, certain procedures and elements are necessary. Mainly these include dynamic substructuring, time integration schemes, real-time operating systems, transfer systems and reference tracking control techniques along with DAQ systems.

2.1.1 *Dynamic substructuring*

The dynamic response of a reference structural system defined by the Ω^R domain, Figure 2.1(a), can be obtained in terms of degrees of freedom (DoF) and for $\mathbf{x}(t) \forall t \in [0, t_f]$ by computing its equation of motion (EoM), which admits:

$$\Omega^R: \quad \mathbf{M}^R \ddot{\mathbf{x}}(t) + \mathbf{C}^R \dot{\mathbf{x}}(t) + \mathbf{K}^R \mathbf{x}(t) = \mathbf{f}_g(t), \tag{2.1}$$

where $\mathbf{x}$, $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$ corresponds to the displacement, velocity and acceleration vectors of the structural system respectively, $\mathbf{M}^R$, $\mathbf{C}^R$ and $\mathbf{K}^R$ are the mass, damping and stiffness matrices respectively, $\mathbf{f}_g$ the applied excitation expressed in force units and t_f the simulation duration. In earthquake engineering, the term $\mathbf{f}_g$ is commonly expressed as $\mathbf{f}_g(t) = -\mathbf{i}\ddot{x}_g(t)$, where $\mathbf{i}$ denotes the seismic participation vector and $\ddot{x}_g$ the ground acceleration.

Dynamic substructuring refers to the process of decomposing the reference structural system into multiple subdomains, at least two. Some of them are treated as the NS, i.e. Ω^N in Eq. (2.2), and the remaining as the PS, i.e. Ω^P in Eq. (2.3). Figure 2.1(b) illustrates the substructuring of a reference system into two substructures. The reference structure can be defined as the union of all the encompassed substructures; $\Omega^R = \Omega^N \cup \Omega^P$. Each

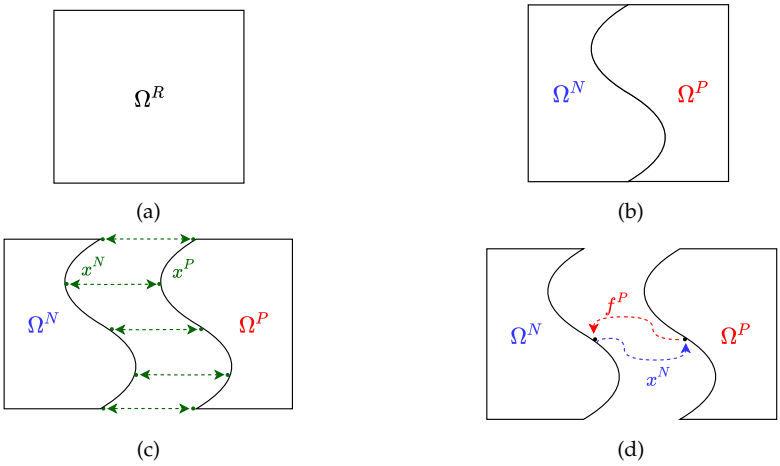


FIGURE 2.1: Dynamic substructuring and coupling of a structural system.

substructure can be then simulated independently, assuming that coupling of their boundary conditions is enforced in terms of compatibility and equilibrium conditions at their interfaces. Figure 2.1(c) offers an example of such coupling, in which the substructures' displacements at their interface are enforced to be synchronized, i.e. $x^N = x^P$ at every time-step of the HS. Apart from displacement signals, velocity, acceleration or force can be also chosen to enforce the substructures' coupling. The selection depends mainly on the context of the application.

$$\Omega^N: \quad \mathbf{M}^N\ddot{\mathbf{x}}(t) + \mathbf{C}^N\dot{\mathbf{x}}(t) + \mathbf{K}^N\mathbf{x}(t) = \mathbf{f}^N(t), \tag{2.2}$$

$$\Omega^P: \quad \mathbf{M}^P\ddot{\mathbf{x}}(t) + \mathbf{C}^P\dot{\mathbf{x}}(t) + \mathbf{K}^P\mathbf{x}(t) = \mathbf{f}^P(t), \tag{2.3}$$

where

$$\mathbf{M}^R = \mathbf{M}^N + \mathbf{M}^P, \quad \mathbf{C}^R = \mathbf{C}^N + \mathbf{C}^P, \quad \mathbf{K}^R = \mathbf{K}^N + \mathbf{K}^P. \tag{2.4}$$

Substitution of Eq. (2.4) into Eq. (2.1) results in:

$$\begin{aligned}\mathbf{M}^N\ddot{\mathbf{x}}(t) + \mathbf{C}^N\dot{\mathbf{x}}(t) + \mathbf{K}^N\mathbf{x}(t) &= \mathbf{f}_g - \left(\mathbf{M}^P\ddot{\mathbf{x}}(t) + \mathbf{C}^P\dot{\mathbf{x}}(t) + \mathbf{K}^P\mathbf{x}(t)\right) \\ &= \mathbf{f}_g(t) - \mathbf{f}^P(t).\end{aligned} \tag{2.5}$$

The latter equation can be visualized from Figure 2.1(d). At each HS time-step, the utilized time integration scheme computes the NS displacement, using Eq. (2.2), and via the employed transfer system transmits the computed energy to the PS. The force generated from the latter due to that motion exchange is measured, via e.g. load cells, and fed back to the time integration scheme to compute the NS displacements for the next HS time-step. This feedback loop continues as long as $t < t_f$, where t the current time of HS. As mentioned above, the displacement/force signal combination is used only for explanatory purposes. A combination of other types of signals is also possible.

To date, several dynamic substructuring techniques exist in the literature. To state a few, Rixen further developed the Craig-Bampton method [38] to a dual assembly, the so-called dual Craig-Bampton [39]. Hashemi and Mosqueda proposed a substructuring method in which the boundary conditions are overlapping between the NS and PS domains [40]. Abbiati et al. proposed a framework for fast HS based on partitioned time integration and state-space modeling [41]. Later this framework was extended to facilitate dynamic relaxation as an alternative to dynamic simulation method [42].

2.1.2 *Time integration schemes*

The EoM of structural systems, e.g. Eq. (2.5), are mostly second-order ordinary differential equations (ODE). To solve them, usage of time integration algorithms is essential. The characteristics of these algorithms can directly affect the HS response and therefore they are of great importance for HS. Time integration algorithms are defined by their numerical stability and accuracy properties. An algorithm is numerical stable when it can restrain the initial error under a specific level within the gradual integration. When the numerical stability is depending on the time-step, the algorithm is referred as conditionally stable and unconditionally stable if otherwise. Accuracy is related to how much the time integration result deviates from the analytical solution of the underlying equation. The accuracy of time integration algorithms can be determined by the so-called order of accuracy, denoted by k; $\tau = \mathcal{O}(\Delta t^k)$, where Δt is the time-step, a.k.a. time-interval, and τ the truncation error. Another property of time integration schemes is the numerical damping, a.k.a. numerical dissipation, which refers to the ability of the algorithm to restrain fictitious oscillations. The latter is considered as an artifact introduced from the spatial discretization of the examined system, without representing any realistic behavior of it.

Additionally, time integration algorithms are categorized into explicit and implicit. An explicit algorithm computes the system's response at a

future time-step based on information available up to the current time-step. Whereas in an implicit algorithm, the response at a future time-step is computed based on information at the current and future time-step. Considering that the system's response at a future time-step is expressed by $y(t+\Delta t)$, then an explicit algorithm would have to solve:

$$y(t+\Delta t) = F(y(t)), \tag{2.6}$$

while an implicit:

$$G(y(t), y(t+\Delta t)) = 0, \tag{2.7}$$

where $F(\cdot)$, $G(\cdot)$ are functions obtained by implementation of the explicit and implicit algorithms. Most implicit algorithms are unconditionally stable, while explicit are conditionally stable.

To date, several families of time integration methods exist. Below some methods, already used in HS, are shortly presented.

2.1.2.1 *Central difference method*

One widely used time integration algorithm is the central difference method. It is an explicit, second-order accurate method with no numerical damping, commonly used due to its simplicity. Without loss of generality, let the EoM of Eq. (2.1) be:

$$\mathbf{M}\ddot{\mathbf{x}}_k + \mathbf{C}\dot{\mathbf{x}}_k + \mathbf{K}\mathbf{x}_k = \mathbf{f}_{g_k}. \tag{2.8}$$

Considering now the Eq. (2.8), the solution from the central difference method follows:

$$\dot{\mathbf{x}}_k = \frac{\mathbf{x}_{k+1} - \mathbf{x}_{k-1}}{2\Delta t}, \tag{2.9}$$

$$\ddot{\mathbf{x}}_k = \frac{\mathbf{x}_{k+1} - 2\mathbf{x}_k + \mathbf{x}_{k-1}}{\Delta t^2}. \tag{2.10}$$

Some of the HS studies that the central difference method was used are [17, 43], among others.

2.1.2.2 *Newmark's family methods*

Newmark's family methods [44] is a group of time integration schemes, defined by the parameters β and γ. Widely used algorithms of this group is the trapezoidal rule and the linear acceleration algorithm. Considering the EoM of Eq. (2.8), the solution from the Newmark's family method follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t\dot{\mathbf{x}}_k + (0.5-\beta)\Delta t^2\ddot{\mathbf{x}}_k + \beta\Delta t^2\ddot{\mathbf{x}}_{k+1}, \tag{2.11}$$

$$\dot{\mathbf{x}}_{k+1} = \dot{\mathbf{x}}_k + \Delta t(1-\gamma)\ddot{\mathbf{x}}_k + \Delta t\gamma\ddot{\mathbf{x}}_{k+1}. \tag{2.12}$$

The algorithm is unconditionally stable for $1/2 \leqslant \gamma \leqslant 2\beta$, but for $\gamma < 1/2$ is conditionally stable. For $\beta = 0$ and $\gamma = 1/2$, the algorithm coincides with the central difference method. While a drawback of Newmark's method is that the numerical damping is not the desirable, studies towards the direction of improving this property were made. Some of them, within the HS domain, are in [45, 46, 47].

2.1.2.3 *α-family methods*

The α-family methods is a generalization of the Newmark's family methods to improve the numerical dissipation property by introducing the α term in the EoM. Such a method is the Hilber-Hughes-Taylor (HHT)-α method [48] in which the α term is introduced in the EoM of Eq. (2.8), admitting:

$$\mathbf{M}\ddot{\mathbf{x}}_{k+1} = (1-\alpha)\left[\mathbf{f}_{g_{k+1}} - \mathbf{C}\dot{\mathbf{x}}_{k+1} - \mathbf{K}\mathbf{x}_{k+1}\right] - \alpha\left[\mathbf{f}_{g_k} - \mathbf{C}\dot{\mathbf{x}}_k - \mathbf{K}\mathbf{x}_k\right]. \tag{2.13}$$

The solution from the HHT-α method follows Eqs. (2.11) and (2.12) with $\beta = \frac{1}{4}(1-\alpha)^2$, $\gamma = \frac{1}{2} - \alpha$ and $\alpha \in [-1/3, 0]$. Examples of HHT-α in HS are in [49, 50].

Associating the α parameter of Eq. (2.13) only with the mass term, results to the Wood-Bossak-Zienkiewicz (WBZ)-α method [51], with EoM following:

$$(1-\alpha)\mathbf{M}\ddot{\mathbf{x}}_{k+1} = -\alpha\mathbf{M}\ddot{\mathbf{x}}_k - \mathbf{C}\dot{\mathbf{x}}_{k+1} - \mathbf{K}\mathbf{x}_{k+1} + \mathbf{f}_{g_{k+1}} \tag{2.14}$$

and solution as of Eqs. (2.11) and (2.12). Both HHT-α and WBZ-α methods are unconditionally stable, second-order accurate and with optimal numerical dissipation.

Combination of the HHT-α and the WBZ-α methods, results in the generalized-α method [52]. Characteristics of this method is the unconditional stability, the second-order accuracy and the optimal numerical dissipation. Two additional parameters, namely the α_m and α_f, are introduced formulating the EoM of Eq. (2.8) as follows:

$$\mathbf{M}\ddot{\mathbf{x}}_{k+1-\alpha_m} + \mathbf{C}\dot{\mathbf{x}}_{k+1-\alpha_f} + \mathbf{K}\mathbf{x}_{k+1-\alpha_f} = \mathbf{f}_{g_{k+1-\alpha_f}}. \tag{2.15}$$

The solution of Eq. (2.15) using the generalized-α method admits:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t\dot{\mathbf{x}}_k + (0.5-\beta)\Delta t^2\ddot{\mathbf{x}}_k + \beta\Delta t^2\ddot{\mathbf{x}}_{k+1}, \tag{2.16}$$

$$\dot{\mathbf{x}}_{k+1} = \dot{\mathbf{x}}_k + \Delta t(1-\gamma)\ddot{\mathbf{x}}_k + \Delta t\gamma\ddot{\mathbf{x}}_{k+1}, \tag{2.17}$$

$$\mathbf{x}_{k+1-\alpha_f} = (1-\alpha_f)\mathbf{x}_{k+1} + \alpha_f\mathbf{x}_k, \tag{2.18}$$

$$\dot{\mathbf{x}}_{k+1-\alpha_f} = (1-\alpha_f)\dot{\mathbf{x}}_{k+1} + \alpha_f\dot{\mathbf{x}}_k, \tag{2.19}$$

$$\ddot{\mathbf{x}}_{k+1-\alpha_m} = (1-\alpha_m)\ddot{\mathbf{x}}_{k+1} + \alpha_m\ddot{\mathbf{x}}_k, \tag{2.20}$$

$$\mathbf{f}_{k+1-\alpha_f} = (1-\alpha_f)\mathbf{f}_{k+1} + \alpha_f\mathbf{f}_k. \tag{2.21}$$

2.1.2.4 *Operator-Splitting method*

The operator-splitting (OS) method is a combination of implicit and explicit methods, concurrently implemented [53, 54]. It is a partitioned analysis algorithm that divides the structure into two geometrical meshes. The one is solved from the implicit algorithm while the other from the explicit. This method originated from the need to handle subsystems that are governed from stiff EoM. A stiff equation is numerically unstable unless the solver is implicit. Therefore, the idea behind OS method is to divide the system into stiff and soft subsystems. The former is solved using implicit algorithms, whereas the latter from explicit for efficiency purposes.

In the OS method the EoM admits:

$$\mathbf{M}\ddot{\mathbf{x}}_{k+1} + \mathbf{C}^I\dot{\mathbf{x}}_{k+1} + \mathbf{C}^E\tilde{\dot{\mathbf{x}}}_{k+1} + \mathbf{K}^I\mathbf{x}_{k+1} + \mathbf{K}^E\tilde{\mathbf{x}}_{k+1} = \mathbf{f}_{g_{k+1}}. \tag{2.22}$$

And the solution follows:

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_k + \Delta t\dot{\mathbf{x}}_k + (0.5-\beta)\Delta t^2\ddot{\mathbf{x}}_k, \tag{2.23}$$

$$\tilde{\dot{\mathbf{x}}}_{k+1} = \dot{\mathbf{x}}_k + \Delta t(1-\gamma)\ddot{\mathbf{x}}_k, \tag{2.24}$$

$$\mathbf{x}_{k+1} = \tilde{\mathbf{x}}_{k+1} + \beta\Delta t^2\ddot{\mathbf{x}}_{k+1}, \tag{2.25}$$

$$\dot{\mathbf{x}}_{k+1} = \tilde{\dot{\mathbf{x}}}_{k+1} + \gamma\Delta t\ddot{\mathbf{x}}_{k+1}, \tag{2.26}$$

where $\mathbf{M} = \mathbf{M}^I + \mathbf{M}^E$, $\mathbf{C} = \mathbf{C}^I + \mathbf{C}^E$, $\mathbf{K} = \mathbf{K}^I + \mathbf{K}^E$ and I, E superscripts correspond to the implicit and explicit methods, respectively. The $\tilde{\mathbf{x}}$, $\tilde{\dot{\mathbf{x}}}$ are commonly referred to as predictor-corrector variables. Examples of the OS method in HS are in [55, 56].

2.1.2.5 *Runge-Kutta methods*

The Runge-Kutta methods is a family of explicit and implicit algorithms, suitable for solving ODEs of nonlinear systems. They are based on single-step integration schemes, substituting higher-order derivatives with finite difference approximations of the system between two successive time-steps. A widely used method of this family is the four-order Runge-Kutta (RK4) method. In particular, for a system governed by the following EoM:

$$\dot{\mathbf{x}}_k = \mathbf{f}(t_k, \mathbf{x}_k, \mathbf{f}_{g_k}), \tag{2.27}$$

the solution from RK4 admits:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{1}{6}\Delta t(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \tag{2.28}$$

where

$$\mathbf{k}_1 = \mathbf{f}(t_k, \mathbf{x}_k, \mathbf{f}_{g_k}), \tag{2.29}$$
$$\mathbf{k}_2 = \mathbf{f}(t_k + \Delta t/2, \mathbf{x}_k + (\Delta t/2)\mathbf{k}_1, \mathbf{f}_{g_k}), \tag{2.30}$$
$$\mathbf{k}_3 = \mathbf{f}(t_k + \Delta t/2, \mathbf{x}_k + (\Delta t/2)\mathbf{k}_2, \mathbf{f}_{g_k}), \tag{2.31}$$
$$\mathbf{k}_4 = \mathbf{f}(t_k + \Delta t, \mathbf{x}_k + \Delta t\mathbf{k}_3, \mathbf{f}_{g_k}). \tag{2.32}$$

Examples of RK4 in HS are in [6, 57].

The textbooks from Butcher [58] and Griffiths et al. [59] offer a more comprehensive review on time integration algorithms. For algorithms used specifically in HS, the reader is encouraged to consult [41, 60, 61, 62] as well.

2.1.3 *Real-time operating systems*

To conduct HS in real-time, the time integration schemes presented in Section 2.1.2 need to be implemented in real-time. Therefore, a real-time operation system needs to be employed in the HS loop. The real-time operating systems is responsible to coordinate the overall HS software, i.e. to execute the time integration scheme, command the transfer system and read the measurements from the PS via the DAQ. Conducting a HS in real-time means that certain tasks of the simulation must be performed under specific time deadlines. Failing to perform such a task in the given time frame would result in time scale distortion and HS performance degradation, risking the stability and fidelity of the overall HS.

In this regard, RTHS can be divided into two groups; the *hard* and *soft* RTHS. The former refers to the case that HS should be conducted at hard real-time, meaning that no task within the HS should miss the prescribed time deadlines. The soft RTHS refers to HS in which frequent missing of time deadlines can be accepted without risking the HS outcome. The textbook from Buttazzo [63] offers a thorough review of real-time computing systems.

Some real-time operating systems already used in HS are: Speedgoat in [4], Simcenter real-time platform in [10], National Instrument in [20], dSpace in [57], MTS in [64] and INDEL in [65].

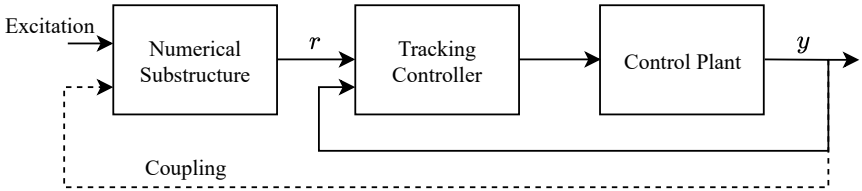
2.1.4 *Transfer systems and reference tracking control*

As mentioned above, the hybrid model results from the coupling of NS and PS. The latter is achieved by transfer systems, a.k.a. actuation systems. In most cases, these transfer systems are an arrangement of actuators or servo-controlled motors. Recall that the transfer systems are paramount for HS as they are responsible for continuously synchronizing the interfaces of the substructures' boundary conditions. As can be appreciated from Figure 2.1(d), there is the need to transmit a signal from the NS, in this case, the x^N displacement, to the PS via a transfer system in order to preserve synchronization of the substructures coupling. However, due to the inherent dynamics of the utilized transfer system, time delays and tracking errors are introduced in the HS loop, risking altering the HS outcomes and diverging from the reference system response. Hence, compensation of these time delays and tracking errors is of critical importance, as HS's goal is to emulate the dynamics of the reference examined system. In control engineering disciplines, controller design for such kind of compensation is referred to as *reference tracking control*. Figure 2.2 serve as an example of the reference tracking problem. Figure 2.2(a) illustrates a common block diagram of the hybrid model. The control plant refers to the system we want to control. In HS, this corresponds to either the transfer system interconnected with the PS or solely to the transfer system, depending on the application. The core idea nonetheless, is that in each HS time-step a command signal r is generated from the NS and needs to be followed by the PS. The tracking controller is operating in a closed-loop fashion, responsible to adequately compensate for tracking errors and time delays. In Figure 2.2(b), an example is provided of the hybrid model response y, with and without tracking control, namely open-loop and closed-loop response, respectively. From Figure 2.2(b), it can be appreciated that large tracking errors are associated with the open-loop response, whereas very small for

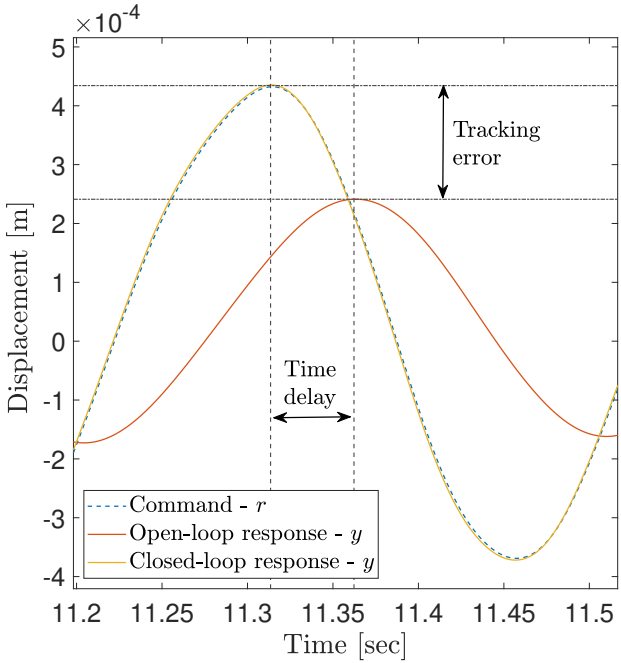
the closed-loop response. Therefore, it can be shown that, especially in RTHS, proper design and implementation of control techniques is of great importance for HS. Furthermore, the tracking controller should be robust enough, in order to maintain the desired performance under disturbances, that may be introduced during an experiment, and to measurement noise.

Recently, several control approaches have been proposed for transfer system dynamics compensation in RTHS. A selection of these approaches is highlighted below. Horiuchi developed a compensation technique using a polynomial extrapolation methodology to overcome time delays [66], which was later modified into an adaptive scheme [7, 67]. Phase-lead compensators were also proposed by several authors. These work by compensating for the phase shift of the transfer system [49, 68, 69]. Another popular compensation method is the inverse compensation, in which an inverse model of the transfer function is used as a feedforward compensator, see for example [70] and references therein. Following the initial work of [71, 72], adaptive compensation strategies were employed to improve the robustness of RTHS by online estimation of controller parameters [73, 74]. Many authors adapted general control methods to RTHS. For example, Carrion and Spencer developed a method using model-based and LQG algorithms [15]. Phillips and Spencer further enhanced this method by adding feedforward and feedback terms, accounting for multi-actuator schemes as well [75, 76]. H_∞ loop shaping controller designs were also proposed as an additional technique to improve the performance and robustness of RTHS under the presence of uncertainties in the experimental procedure [4, 77, 78]. Lately, a self-tuning nonlinear controller based on a combined robust-adaptive scheme was proposed, aiming at capturing nonlinearities of the dynamic interaction between transfer systems and physical substructures [79]. Condori et al. [80] proposed also a robust control approach with a nonlinear Bayesian estimator to address uncertain nonlinear systems. A model-based sliding mode control approach has also been developed for RTHS making use of a reduced control plant [81]. Recently, a benchmark control problem for RTHS was published by Silva et al. [23] with several state-of-the-art contributions towards this research frontier, see [82] for references therein.

Part of this dissertation is the development of reference tracking control strategies for RTHS. Chapter 3 introduces a control approach based on model predictive control for RTHS [83] and Chapter 4 an extension of it to adaptive model predictive control [84].



(a)



(b)

FIGURE 2.2: Reference tracking problem: (a) hybrid model common block diagram, (b) reference tracking example response.

2.1.5 *Data acquisition system*

Another essential component of HS is the data acquisition (DAQ) system. With this we refer to the procedure of signal sampling, measuring real-world physical phenomena and transforming them to digital numerical values, able to be processed in a computer environment. A DAQ system mainly encompasses sensors, instrumentation devices and analog to digital converters. The role of the DAQ system is to measure the dynamic response of the tested PS and to transmit the respective measured values to the time integration algorithm, closing the HS feedback loop. Experimental errors strongly depend on the quality and configuration of the DAQ system. Therefore correct specification of the desired resolution, sensor calibrations as well as suitable sensor placement are critical procedures for the quality of the HS outcomes.

2.1.6 *Uncertainties and errors in hybrid simulation*

Typically, errors in experimental testing are grouped into *systematic* and *random*. Systematic errors refer to errors that are introduced in the system's response in a consistent and repeatable fashion. In other words, they can admit a regular pattern of appearance and even mathematically modeled. They are commonly originating from regular instrument inaccuracies and/or from experimental setup malfunction. Systematic errors can be decreased or eliminated after proper usage of the utilized experimental equipment. Some potential sources of such errors may be faulty sensor calibration, unidentified setup friction and conversions between analog and digital signals. Random errors can also originate from various sources but, on the contrary, they have no regular pattern of occurrence and therefore it is more challenging to control and foresee them than the systematic errors. An example of random error is the measurement noise due to some random electrical noise in the DAQ system.

In engineering contexts, uncertainties are commonly classified into *aleatory* and *epistemic*. Aleatory uncertainties, a.k.a. physical uncertainties, are unknown and are not repeated when the same experiments are conducted twice. They are caused by natural variability in the phenomenon under consideration, e.g. wind speed. Hence, rejection of such uncertainties cannot be achieved. In contrast, epistemic uncertainties arise as a result of lack of knowledge of a system and its environment and can be reduced, supposing that more accurate models and data are accessible [85, 86]. Examples of epistemic uncertainties are imperfections in numerical models or limited

number of measurements. It is often the case that epistemic uncertainties are referred to as systematic errors or vice versa.

In HS, uncertainties and errors are always present. Some common errors are for instance modeling errors because of discretization and numerical errors originating from the time integration scheme. Also present in HS, are tracking errors introduced from the utilized transfer system and measurements errors generated by the DAQ system and the instrumentation devices. Studies have shown that performing HS disregarding the presence of such uncertainties and errors will affect the quality of the results [45, 87]. In fact, small errors may not significantly affect the gradual response of the examined system but when accumulated and propagated at each simulation time-step, then they could considerably alter the investigated response. Therefore, proper handling of such uncertainties and errors, from the early phase of the HS design, is crucial to preserve the HS's fidelity.

2.2 Model verification & validation

V&V consists of a series of actions used to ascertain whether a computational model can accurately describe the underlying equations of its respective mathematical model, i.e. *verification*, and to determine the extent to which it accurately represents the system being modeled, and thus the degree of trust that can be given to its predictions of real-world events, i.e. *validation* [88, 89]. In particular, verification concerns with elimination of coding errors, i.e. code verification, as well as with minimization of numerical errors that inevitably arise due to discretization of physical laws, i.e. calculation verification. Validation concerns with experimental campaigns with the objective of model validation. It also includes accuracy assessment in terms of quantitatively identifying the level the experimental and model outcomes match. The validation process additionally involves quantification of experimental variation and repeatability together with error classification, e.g. random or systematic. The outcomes of a V&V process is metrics of how much a model can be trusted, evaluated by how much its predictions diverge from the observed behavior. Figure 2.3 illustrates activities associated with the V&V process [88].

The *conceptual model* corresponds to a physical phenomenon along with the assumptions and simplification made and represents the solid mechanics which govern it. The *mathematical model* includes the mathematical equations governing the motion of the conceptual model together with the boundary and initial conditions. Numerical implementation of the mathematical model, taking into account numerical discretization along with the time integration scheme, results in the *computational model*. This is the model

used to generate the simulation outcomes which are afterwards compared to the experimental outcomes in the validation process. *Calibration* refers to the procedure of adjusting modeling parameters to increase convergence with experimental data.

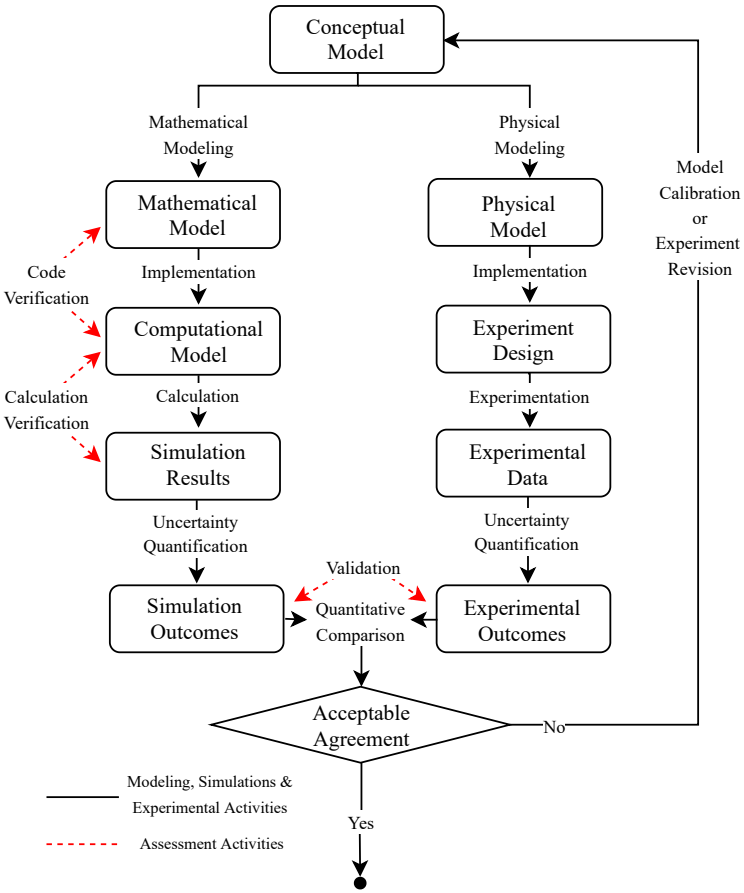


FIGURE 2.3: Activities associated with the model verification and validation (from [88]).

2.3 Conclusions

This chapter outlined the main components/procedures needed to perform a HS, namely dynamic substructuring, time integration schemes, real-time

operating systems, transfer systems, reference tracking control and DAQ systems. A short overview of types of errors and uncertainties that are possible to arise in a HS also took place. The chapter concluded with an introduction to model verification and validation techniques and practices. The following chapters will start addressing the research objectives of this dissertation.

Chapter 3

Model predictive control in real-time hybrid simulation

The material presented in this chapter is published in the Frontiers in Built Environment journal [83]. Part of the work is also presented in the XI International Conference on Structural Dynamics (EURODYN) held in Athens, Greece, in 2020 [90] and in the 17th World Conference on Earthquake Engineering (17WCEE) held in Sendai, Japan, in 2021 [91].

3.1 Introduction

This chapter focuses on the first research objective, namely methodologies to conduct real-time hybrid simulation, and tackles challenges arising from the PS of the hybrid model and in particular from the utilized transfer system. Recall that the inherent dynamics of the transfer system used, introduce time delays and tracking errors in the HS loop, altering the dynamic response of the tested structural system. As a result, implementation of adequate control techniques to compensate for such delays and errors is necessary (challenge 2.a from Section 1.2.2). An overview of already existing control techniques for RTHS was presented in Section 2.1.4.

In this chapter, a novel control method is proposed, in which the tracking controller consists of a model predictive controller (MPC) along with a polynomial extrapolation algorithm and a Kalman filter. One important advantage of MPC is its capability to adapt the control law online, compensating for time delays and uncertainties for a set of specific simulation time-steps. This is of significant importance for RTHS since experimental errors and actuator dynamics introduce arbitrarily delays in the system, which need to be compensated for online. Another significant advantage of MPC is the fact that it can perform online optimization, handling at the same time constraints of the system under consideration. Following the design formulation of the proposed tracking controller, two virtual RTHS (vRTHS) parametric case studies are examined in order to validate the performance and robustness of the proposed control scheme. Variations in the parameters of the hybrid model will prove the robustness of the proposed controller to uncertainties introduced throughout the RTHS

procedure. RTHS using the proposed control scheme is demonstrated to be effective for structural seismic performance assessment.

This chapter is organized as follows. Section 3.2 describes the tracking controller. Section 3.3 introduces the case studies with the vRTHS along with the obtained results, used to validate the performance of the controller. Section 3.4 presents the overall conclusions of this chapter.

3.2 Outline of the tracking controller

In this section, the architecture of the proposed tracking controller is explained. The controller consists of a MPC along with a polynomial extrapolation algorithm and a Kalman filter. In Figure 3.1 the tracking controller's block diagram is shown. In the following sections, the main parts of the controller are described in detail. The control plant corresponds to the system under control. The subsequent vRTHS case studies will give more insight into the control plant dynamics and architecture.

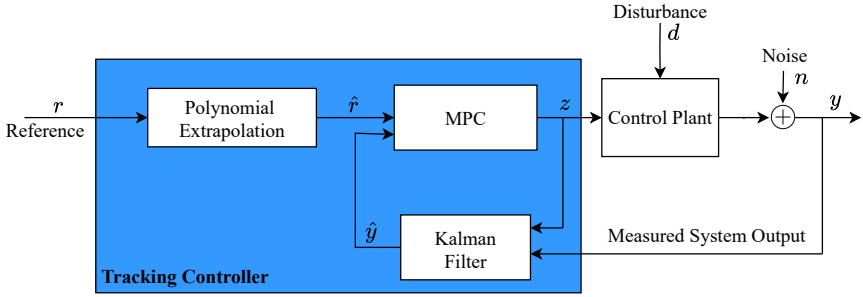


FIGURE 3.1: Architecture of the proposed tracking controller based on MPC.

3.2.1 *Model predictive control*

MPC is a control strategy in which the ongoing control law is adapted by computing, at every control interval, a finite horizon optimization problem, applying the ongoing state of the control plant as the initial state. The optimization generates an optimal control sequence consisting of a series of individual control laws, out of which the first one is applied to the control plant for the current control interval [92]. The control interval is defined as a sampling frequency or, in other words, as a set of continuous time-steps of the simulation, serving as an internal time-step for the MPC. MPC solves essentially a finite horizon optimization problem similar to H_∞

and LQR control techniques. A fundamental difference nevertheless lies in the fact that MPC solves during the simulation, at each control interval, a new optimization problem in a receding horizon approach. Therefore, the control actions are updated online, while in classical optimal control, e.g. LQR, the optimization is solved offline and thus the control law is defined prior to the simulation and remains constant. This is the fundamental difference between MPC and classical control theory. Additionally, in LQR control, for example, the optimal problem could be of the infinite horizon as well, while that's not the case for MPC. Online control law derivation is also a feature of adaptive control theory. However, classical MPC is not an adaptive control scheme as the controller itself does not change/adapt within the simulation; only the control actions are updated.

Every application imposes mandatory (hard) constraints. For example i) actuators are of limited stroke/capacity meaning that the produced displacement/force is limited; and/or ii) safety limits are applied in almost every experimental setup. The problem of meeting hard constraints in control applications is well established in the literature. MPC has proven to be one of few adequate control methodologies to suitably satisfying constraints on the inputs, states and/or outputs of the system under consideration while maintaining concurrently the desired performance [93].

The proposed tracking MPC controller consists of four elements, namely the prediction model, the performance index or cost function, the constraints and a solver to derive the control laws. The prediction model constitutes the core of MPC as it is used to predict the future control plant outputs, based on information available up to the ongoing time-step, e.g. previous and current inputs/outputs, and on the future control actions. As the overall performance of MPC is directly influenced by the responses of the prediction model, the latter should be as accurate as possible in order to be able to sufficiently capture the control plant dynamics and its behavior. Therefore, a detailed prediction model could improve MPC performance. However, a detailed prediction model could also increase the required computational power, needed to compute it at every control interval. Care must be taken in order not to introduce delays due to numerical calculations, especially in real-time applications, as RTHS, in which timing is crucial. Updating the prediction model online would result in the so-called adaptive MPC (see Chapter 4 for more details).

Figure 3.2 displays the structure and methodology of MPC. In more detail, the prediction model predicts at each control interval k the future control plants outputs $\hat{y}(k+i|k),\ i = 0,\dots,P$ for every time instant i within a prediction horizon P. The notation $\hat{y}(k+i|k)$ correspond to the output at the time instant $k+i$ evaluated at k. The control sequence $z_k^T =$

$[u(k|k)^T \dots u(k+i|k)^T \dots u(k+P-1|k)^T]$ includes the sequence of future control actions $u(k+i|k)$. At each k, a cost function is optimized and a new control sequence is computed. The cost function embodies the tracking error, i.e. the error between the reference trajectory and the predicted output values of the control plant, and is expressed as follows:

$$\begin{aligned} \mathbf{J}^*(\hat{r}_k, \hat{y}_k, z_k) = &\sum_{j=1}^{n_y} \sum_{i=1}^{P} \left\{ w^{y_j} [\hat{r}_j(k+i|k) - \hat{y}_j(k+i|k)] \right\}^2 + \\ &\sum_{j=1}^{n_u} \sum_{i=0}^{P-1} \left\{ w^{u_j} [u_j(k+i|k) - u_j(k+i-1|k)] \right\}^2, \end{aligned} \tag{3.1}$$

where n_y is the number of control plant outputs, n_u the number of control plant inputs, $\hat{r}_j(k+i|k)$ the reference value to be tracked at the i-th time instant step from the j-th control plant output, $\hat{y}_j(k+i|k)$ the predicted value of the j-th control plant output at the i-th time instant, $u_j(k+i|k)$ the j-th control plant input at the i-th time instant, w^{y_j} the tuning weight of the j-th control plant output and w^{u_j} the tuning weight of the j-th control plant input. The P, w^{y_j} and w^{u_j} parameters of Eq. (3.1) are computed offline, through trial and error and in the context of this dissertation, they remain constant during the simulation.

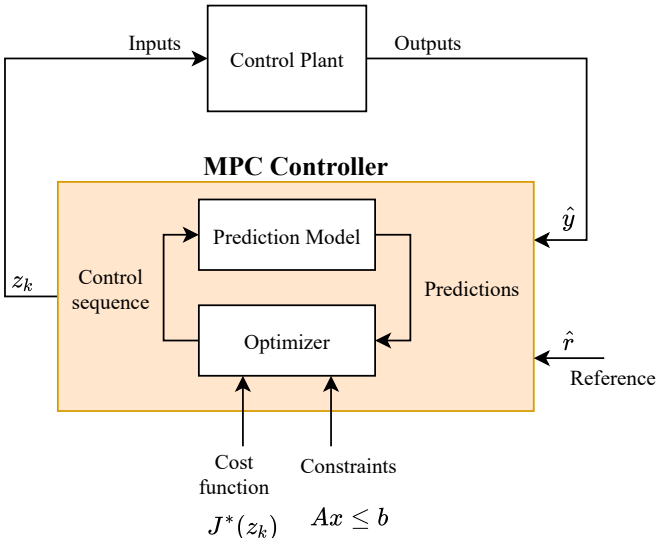
In addition, an output disturbance d and a measurement noise model n are used in the proposed tracking controller as described in Eqs. (3.2) and (3.3), respectively. They are implemented as additive to the control plant outputs, representing potential disturbances and sensor noise that could be present in RTHS. The inputs u_d and u_n are assumed to be Gaussian variables with zero mean and unit variance.

$$\begin{aligned} x_d(k+1) &= A_d x_d(k) + B_d u_d(k) \\ d(k) &= C_d x_d(k) + D_d u_d(k), \end{aligned} \tag{3.2}$$

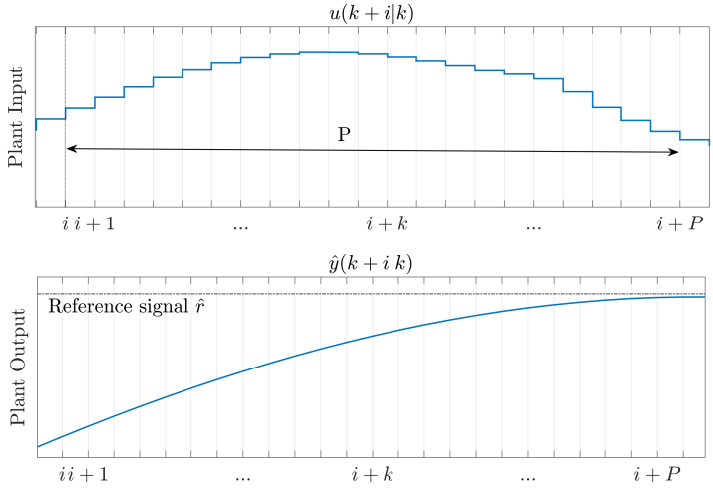
$$\begin{aligned} x_n(k+1) &= A_n x_n(k) + B_n u_n(k) \\ n(k) &= C_n x_n(k) + D_n u_n(k), \end{aligned} \tag{3.3}$$

where A_d, B_d, C_d, D_d and A_n, B_n, C_n, D_n are matrices associated with the disturbance d and noise n, respectively.

The control sequence z_k is computed at each control interval in the optimizer (see Figure 3.2(a)), which takes into account the cost function (and in



(a)



(b)

FIGURE 3.2: (a) MPC structure of the tracking controller, (b) MPC methodology.

essence the tracking error, as it is embedded in the cost) and the constraints. The cost function of Eq. (3.1) can be formulated into a Quadratic Programming (QP) problem [94, 95] and this is what the optimizer essentially solves. The QP formulation admits:

$$\min_{x}\left(\frac{1}{2}x^{T}Hx + f^{T}x\right) \tag{3.4}$$

$$\text{subject to} \quad Ax \leqslant b, \tag{3.5}$$

where the $Ax \leqslant b$ inequality represents the applied constraints, x is the solution vector, H the Hessian matrix, vectors A and b define the constraints and f is a vector computed by:

$$f = Kx(k|k)^{T}x(k|k) + K\hat{r}(k|k)^{T}\hat{r}(k|k) + Ku(k|k-1)^{T}u(k|k-1), \tag{3.6}$$

where $\hat{r}(k|k)$ is the reference signal at the current control interval, $u(k|k-1)$ is the applied control action in the previous control interval and K a weighting factor.

In the proposed tracking controller, the QP problem of Eqs. (3.4) and (3.5) is solved employing an active-set solver which applies the KWIK algorithm [96]. The latter is a built-in QP solver from the Model Predictive Toolbox of MATLAB, used in this study to compute the control law sequence [97].

The MPC algorithm used in the proposed tracking controller can be summarized as follows:

1. **Prediction step:** Considering a discrete time multiple-input-multiple-output (MIMO) system that represents a linearized model of the control plant, the prediction model in state-space formulation at the control interval k follows:

$$\begin{aligned} x_p(k+1) &= A_p x_p(k) + B_p u(k) \\ y_p(k) &= C_p x_p(k) + D_p u(k), \end{aligned} \tag{3.7}$$

   where A_p, B_p, C_p and D_p are matrices associated with the prediction model, representing the control plant dynamics. The disturbance and noise models of Eqs. (3.2) and (3.3), respectively, are assumed to be additive to the measured control plant output (y signal in Figure 3.1) and hence the latter admits: $y(k) = y_p(k) + d(k) + n(k)$.

2. **Optimization step:** Using the current state estimates and predicted control plant outputs from the Kalman filter (Eqs. (3.15) and (3.18)), MPC solves the optimization problem described by Eq. (3.1) for the current control interval k:

$$\min_{z_k} \mathbf{J}^*(\hat{r}_k, \hat{y}_k, z_k) \tag{3.8}$$

$$\text{subject to} \quad u_{j_{min}} \leqslant u_j \leqslant u_{j_{max}} \quad \text{and} \tag{3.9}$$

$$\dot{u}_{j_{min}} \leqslant \dot{u}_j \leqslant \dot{u}_{j_{max}}, \tag{3.10}$$

where the above constraints correspond to the physical limitations of the actuator regarding displacement u and velocity $\dot{u}$ capacity and $\mathbf{J}^*(\hat{r}_k, \hat{y}_k, z_k)$ to the cost of Eq. (3.1).

3. **Control step:** The control sequence for the current control interval $z_k^T = [u(k|k)^T \dots u(k+i|k)^T \dots u(k+P-1|k)^T]$ is computed and its first control action is applied to the control plant:

$$z_k := \arg\min \mathbf{J}^*(\hat{r}_k, \hat{y}_k, z_k) \tag{3.11}$$

4. Steps 1-3 are repeated till the completion of the RTHS.

If the control interval coincides with the sampling frequency of RTHS, then at the next time-step of RTHS a new optimization will be performed and a new control sequence will be computed, taking into account the new information available, e.g. disturbance, noise, introduced in that time-step. When the control interval is larger than the RTHS time-step, i.e. includes several time-steps, then the first control action of step 3 will be applied as long as the ongoing control interval lasts. Therefore, reducing the time-steps in the control interval would result in a more robust MPC. However, the smaller the control interval the more the optimizations that must be performed, thus increasing the computational load. Additionally, as mentioned before, a higher dimensional prediction model, that could potentially capture more accurately the control plant dynamics, would increase the performance of MPC as the predictions would be more accurate. Nevertheless, a higher dimensional model would require additional computational power. Consequently, taken the aforementioned points into consideration, there exists a trade-off between controller performance and computational effort needed.

In RTHS, the uncertainties and experimental errors are neither constant nor predictable. MPC enables computing a new control law for every control interval within the simulation time, making it possible to compensate

specifically for the incurred time delays, uncertainties and/or experimental errors that are introduced in RTHS at each specific control interval. In contrast, classical control techniques utilize a single pre-computed control law that is robust enough to compensate for all the delays coming into play in the entire simulation process. In addition, RTHS always involves experimental equipment, which has physical boundaries, e.g. limited actuator force capacity. Hence, the command signals must be limited to satisfy these boundaries. MPC can solve optimization problems and concurrently satisfy hard constraints, which in RTHS case, can be laboratory limitations. Therefore, MPC is desirable and suitable for RTHS applications.

MPC theory is quite extensive, covering various subjects, e.g. convex optimization, optimal control theory, computational solvers, that are taken into account during the design and implementation process of MPC and are not described in full detail in the above section. For a more comprehensive literature review in MPC, the reader is encouraged to consult [98, 99, 100, 101, 102].

3.2.2 *Kalman filter*

As mentioned above, the performance of MPC depends on the accuracy of the predicted control plant outputs. In order to provide MPC with low-noise control plant outputs, a Kalman filter is used. The control plant output predictions are required to estimate how the current control sequence will regulate the future control plant outputs and hence utilize these estimations to optimize the control actions. Furthermore, the Kalman filter provides estimates of the plant outputs when there are none available from the plant sensors. Considering Eqs. (3.7) and (3.2), the state-space formulation including the control plant dynamics and the disturbance model follows:

$$\begin{aligned} x_c(k+1) &= A_c x_c(k) + B_c u_c(k) \\ \hat{y}(k) &= C_c x_c(k) + D_c u_c(k), \end{aligned} \tag{3.12}$$

where

$$A_c = \begin{bmatrix} A_p & 0 \\ 0 & A_d \end{bmatrix}, \quad B_c = \begin{bmatrix} B_p & 0 \\ 0 & B_d \end{bmatrix}, \tag{3.13}$$
$$C_c = \begin{bmatrix} C_p & C_d \end{bmatrix}, \quad D_c = \begin{bmatrix} D_p & D_d \end{bmatrix},$$
$$x_c^T = \begin{bmatrix} x_p^T & x_d^T \end{bmatrix} \text{ and } u_c^T = \begin{bmatrix} u^T & u_d^T \end{bmatrix}.$$

The weighting coefficients Q, R, N of the Kalman filter are derived from the following expectations:

$$Q = \mathbb{E}\left[dd^T\right], \quad R = \mathbb{E}\left[nn^T\right], \quad N = \mathbb{E}\left[dn^T\right]. \tag{3.14}$$

Figure 3.1 displays the interconnection of the Kalman filter within the proposed tracking controller. In more detail, at each control interval k, the x_c states are estimated as follows:

1. **Measurement update step:** The current state estimate $x_c(k|k)$ is adjusted with the latest measurements:

$$x_c(k|k) = x_c(k|k-1) + M\left[y(k) - C_c x_c(k|k-1)\right], \tag{3.15}$$

   where $x_c(k|k-1)$ denotes the state estimate from the $k-1$ control interval.

2. **Estimation step:** The state for the next, $k+1$, control interval is estimated as:

$$x_c(k+1|k) = A_c x_c(k|k-1) + B_{c_1} u(k) + L\left[y(k) - C_c x_c(k|k-1)\right], \tag{3.16}$$

   where $u(k)$ the control action used from $(k-1)$ till k, $y(k)$ the control plant output measured at k and B_{c_1} the first column of B_c. M, L from Eqs. (3.15) and (3.16) respectively, denote the Kalman filter gain vectors.

Once the current states are estimated, the controller predicts low-noise control plant outputs for the entire prediction horizon of the current control interval.

1. For any successive time instant of the prediction horizon, $i = 1, \ldots, P$, the predicted control plant states are obtained by::

$$x_c(k+i|k) = A_c x_c(k+i-1|k) + B_{c1} u(k+i-1|k). \tag{3.17}$$

2. The predicted control plant output for $i = 1, \dots, P$ is obtained by:

$$\hat{y}(k+i|k) = C_c x_c(k+i|k). \tag{3.18}$$

3.2.3 *Polynomial extrapolation*

MPC can guarantee adequate tracking performance and robustness under uncertainties and disturbances. However, since in RTHS even small tracking errors can significantly alter the simulation outcome, a fourth-order polynomial extrapolation [7, 66, 67, 77] is integrated in the tracking controller as illustrated in Figure 3.1, in order to further compensate for time delays and additionally improve the MPC performance. Its formulation follows:

$$\hat{r}_k = a_0 r_{(0,k)} - a_1 r_{(1,k)} + a_2 r_{(2,k)} - a_3 r_{(3,k)} + a_4 r_{(4,k)}, \tag{3.19}$$

where $r_{(i,k)} = r(t_k - iT_d)$ is the discrete reference signal by adding shifts of a pure time delay T_d by integer values of i. The polynomial coefficients a_0 - a_4 are obtained using the Lagrange basis function and by trial and error.

3.3 Case Studies

The following two virtual RTHS (vRTHS) parametric case studies serve as validation for the performance and robustness of the proposed tracking controller. The case studies are virtual in that both the PSs of the hybrid models are implemented numerically in software, not physically as specimens in a laboratory. This was done to facilitate the development and testing of the proposed tracking controller. For each case study (CS), the dynamics of the tested system are explained, then the tracking controller design properties are addressed and finally, results are presented.

To assess the performance of the controller three metrics are defined. These are:

1. Tracking time delay, defined as:

$$J_1 = \left(\arg\max_k \left(\mathrm{Corr}(r(i), y(i-k)) \right) \right) f_{\mathrm{RTHS}} \quad \text{[msec]}, \tag{3.20}$$

where f_{RTHS} is the sampling frequency of RTHS.

2. Normalized Root Mean Square Error (NRMSE) of the tracking error, defined as:

$$J_2 = \sqrt{\frac{\sum_{i=1}^{N}\left[y(i)-r(i)\right]^2}{\sum_{i=1}^{N}\left[r(i)\right]^2}} \times 100 \quad [\%]. \tag{3.21}$$

3. Peak Tracking Error (PTE), defined as:

$$J_3 = \frac{\max|y(i)-r(i)|}{\max|r(i)|} \times 100 \quad [\%]. \tag{3.22}$$

J_1 is established as the maximum cross-correlation between the reference and the measured signal, multiplied by the sampling frequency of RTHS. It is a metric of how different in time these two signals are. The cross-correlation describes how many time-steps the measured signal should be shifted in order to match the reference. When $J_1 > 0$ the measured signal is delayed with respect to the reference (tracking delay), whilst when $J_1 < 0$, the measured signal is leading the reference (overcompensation). The desire is to have zero time tracking delay, meaning the value of J_1 to be as close to zero as possible, without overcompensating. J_2 represents how quantitatively different the reference and measured signals are accounting for the whole simulation period, whilst J_3 accounts only for the maximum value of the tracking error. The performance of the tracking controller is assessed by how close to zero J_1, J_2 and J_3 are [23].

3.3.1 *Case Study 1: virtual RTHS of a prototype structure with an attached pendulum*

3.3.1.1 *Problem Formulation*

The reference system under consideration for case study 1 (CS1) corresponds to a vertical cantilever beam with mass concentrated at its top, and a pendulum attached to the center of gravity of the cantilever mass, as shown in Figure 3.3(a). The NS is the cantilever beam, Figure 3.3(b), described by Eq. (3.23), while the virtual PS (vPS) is the pendulum, Figure 3.3(c).

The EoM for the reference structure follows:

$$M^N\ddot{x} + C^N\dot{x} + K^N x = -M^N\ddot{x}_g + f^P, \tag{3.23}$$

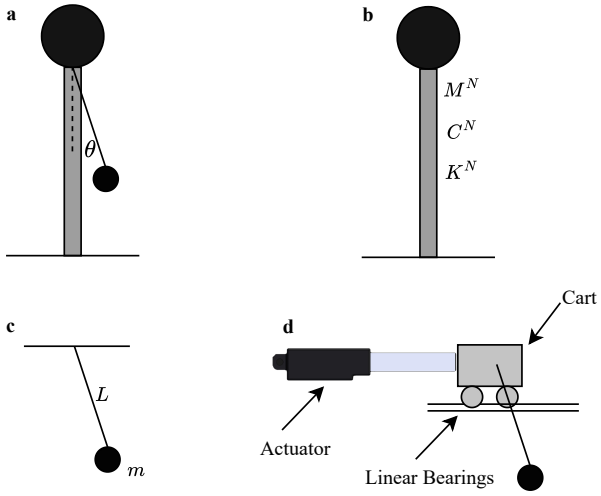


FIGURE 3.3: Hybrid model of CS1: (a) reference structure, (b) NS, (c) vPS, (d) control plant.

where $\ddot{x}$, $\dot{x}$ and x correspond to acceleration, velocity and displacement of the NS relative to the ground, $M^N = 100$ [Kg], $C^N = 100 \left[\frac{Ns}{m}\right]$ and $K^N = 10e4 \left[\frac{N}{m}\right]$ are the mass, damping and stiffness of the NS respectively, $\ddot{x}_g$ is the ground motion applied to the hybrid model and f^P the force measured from the vPS.

The vPS corresponds only to the pendulum. However, to move the pendulum pivot point horizontally in a lab, an actuator could be attached to a cart mounted on a horizontal rail. Thus, the cart and the actuator would be the transfer system. As a result, the cart dynamics and its interaction with the pendulum are taking into account for solving the equations for the vPS. The vPS is described by the Eqs. (3.24) and (3.25). Moreover, in order to reduce as much as possible the friction due to the cart movement μ, low friction linear bearings are assumed to be implemented. The friction at the pendulum pivot, b, is assumed to be small.

The EoM for the cart with the pendulum follows:

$$\ddot{x} = \frac{mL^2\dot{\theta}^2 \sin(\theta) + mLg\sin(\theta)\cos(\theta) + \dot{\theta}b\cos(\theta) - \mu L\dot{x} + f^P L}{L(m\sin^2(\theta) + M)}, \quad (3.24)$$

$$\ddot{\theta} = \frac{-mL^2\dot{\theta}^2 \sin(\theta)\cos(\theta) - \dot{\theta}b(1 + \frac{M}{m}) - gL(M+m)\sin(\theta)}{L^2(m\sin^2(\theta) + M)} + \quad (3.25)$$

$$+ \frac{\dot{x}L\mu\cos(\theta) + f^P L\cos(\theta)}{L^2(m\sin^2(\theta) + M)},$$

where the parameters from Eqs. (3.24) and (3.25) correspond to:

- Pendulum angle, angular velocity, angular acceleration $\rightarrow$ θ, $\dot{\theta}$, $\ddot{\theta}$, respectively,
- Cart position, velocity, acceleration $\rightarrow$ x, $\dot{x}$, $\ddot{x}$, respectively,
- Force generated from the pendulum $\rightarrow$ f^P,
- Pendulum mass $\rightarrow$ $m = 0.15$ [Kg],
- Cart mass $\rightarrow$ $M = 2$ [Kg],
- Rod length $\rightarrow$ $L = 0.7$ [m],
- Cart friction coefficient $\rightarrow$ $\mu = 0.001$ [-],
- Pendulum friction coefficient $\rightarrow$ $b = 0.0001$ [-],
- Acceleration of gravity $\rightarrow$ $g = 9.81$ [m/s^2].

Since this is a virtual simulation, an actuator model needs to be implemented representing the dynamics of the real actuator. For CS1, a linear hydraulic actuator was chosen. Its model consists of three transfer functions; i) G_{sv} represents the servo-valve dynamics as in Eq. (3.26), ii) G_a the actual actuator dynamics as in Eq. (3.27) and iii) G_{CSI} the control-structure-interaction (CSI) [103] as in Eq. (3.28). The way these transfer functions are interconnected is shown using a block diagram of the actuator model in Figure 3.4. Taking the above into consideration, the control plant for CS1, corresponds to the actuator model along with the cart and the pendulum. A graphical representation of the control plant is illustrated in Figure 3.3(d) and its block diagram in Figure 3.4. The control plant is a single-input-multiple-output (SIMO) model with input the displacement of the actuator z and four outputs; x, $\dot{x}$, θ and $\dot{\theta}$. In the tracking controller, only the first

output, the cart position x is used, described by a single-input-single-output (SISO) transfer system as in Eq. (3.29).

$$G_{sv} = \frac{2.128e13}{s^2 + 425s + 99976} \tag{3.26}$$

$$G_a = \frac{1}{s + 3.3} \tag{3.27}$$

$$G_{CSI} = 7.26e5s \tag{3.28}$$

$$G_{cp} = \frac{0.5s^2 + 0.0007823s + 8.058}{s^4 + 0.001963s^3 + 15.07s^2 + 0.007s} \tag{3.29}$$

The block diagram of the overall hybrid model is presented in Figure 3.5. It consists of the numerical and vPS, the proposed tracking controller and the control plant. The RTHS is conducted in displacement control; in every time-step the measured displacement of the cart (same as the horizontal displacement of the pendulum), x, is fed back in the tracking controller, while the measured force generated from the movement of the pendulum, f^P is fed back to the NS to compute the next displacement command r. The coupling of the two substructures is achieved through force f^P. The matrices of the disturbance model, Eq. (3.2), admit: $A_d = 1, B_d = 0.0009766, C_d = 1$ and $D_d = 0$. Furthermore, apart from the latter, additional measurement noise n, see Eq. (3.3), is added in the calculated displacement x and force f^P, representing measurement noise from the displacement and force sensors respectively. The noise for displacement and force is generated from two correlated standard Gaussian distributions amplified each by $1.5e-7$ m and $6e-5$ N, respectively, which approximately equals to 0.01% of their full span. The matrices of Eq. (3.3) admit the following values in CS1:

$$A_n = 0_{2\times2}, B_n = 0_{2\times2}, C_n = 0_{2\times2} \text{ and } D_n = \begin{bmatrix} 1.5e-7 & 0 \\ 0 & 6e-5 \end{bmatrix}. \tag{3.30}$$

The reference ground motion of the hybrid model $\ddot{x}_g$ is a historical acceleration record from the El Centro 1940 earthquake downscaled by 0.4, as shown in Figure 3.6. The sampling frequency of RTHS was set to f_{RTHS} = 4096 Hz. For the time integration scheme, the RK4 method is used with a fixed time-step of 1/4096 sec.

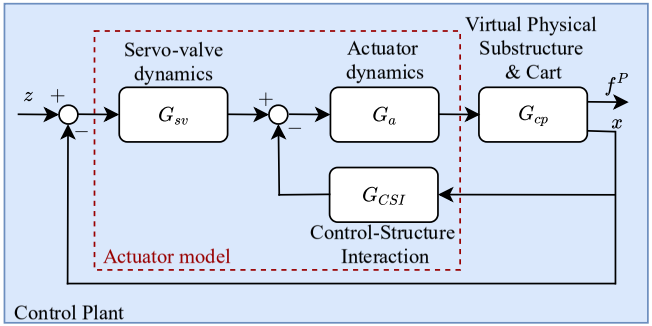


FIGURE 3.4: Block diagram of the control plant in CS1.

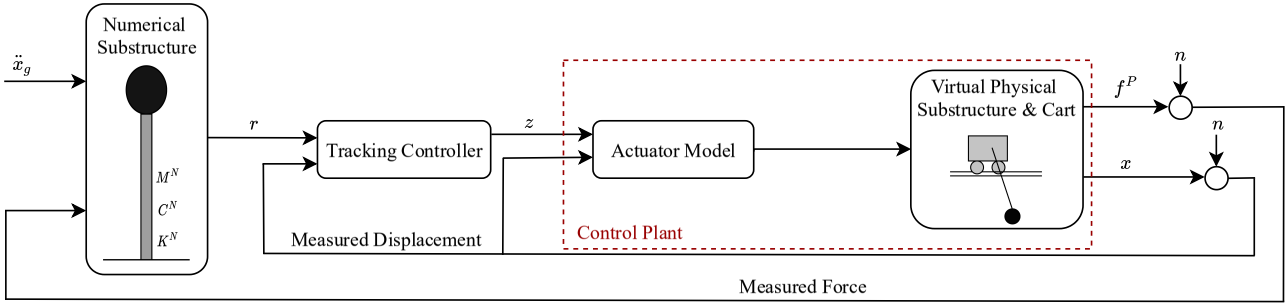
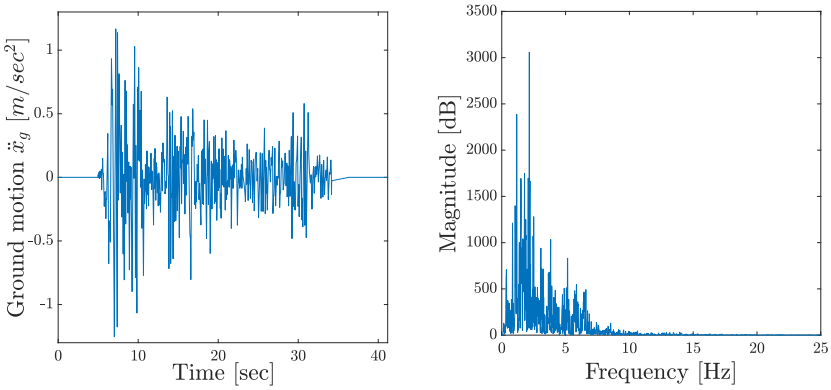


FIGURE 3.5: Block diagram of the hybrid model in CS1.



(a) (b)

FIGURE 3.6: Reference ground motion used in CS1 & CS2: El Centro 1940 earthquake North-South ground motion record, downscaled by 0.4: (a) time history, (b) frequency spectra.

3.3.1.2 *Tracking controller design properties*

The prediction model used in MPC for CS1 is a linearized model of the control plant. Since MPC functions in discrete time, the linearized model of the control plant is discretized with the sampling frequency of RTHS, f_{RTHS}. Essentially it is a discrete linear time-invariant (LTI) SISO model described by Eq. (3.7). The A_p, B_p, C_p and D_p correspond to the prediction model matrices and are equal to:

$$A_p = \begin{bmatrix} 0.4762 & -0.3993 & -0.2475 & -0.2307 & -0.2279 & -0.0009 & -0.0143 \\ 0.7532 & 0.7912 & -0.1315 & -0.1222 & -0.1272 & -0.0005 & -0.008 \\ 0.2103 & 0.4645 & 0.9775 & -0.0209 & -0.0223 & 0 & -0.0014 \\ 0.0184 & 0.0603 & 0.2486 & 0.9987 & -0.0014 & 0 & 0 \\ 0.0012 & 0.0051 & 0.0312 & 0.2499 & 1 & 0 & 0 \\ 0 & 0 & 0.0002 & 0.002 & 0.0156 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0.0156 & 1 \end{bmatrix},$$

$$B_p = \begin{bmatrix} 0.0118 \\ 0.0066 \\ 0.0012 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, C_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 19.86 & 0.002 & 1.25 \end{bmatrix}, D_p = \begin{bmatrix} 0 \end{bmatrix}. \tag{3.31}$$

Additionally, the Kalman filter gain vectors follow:

$$L = [0.0096 \quad -0.0062 \quad -0.0269 \quad 0.0021 \quad 0.0148 \quad 0.0004 \quad -0.0003 \quad 0.0008]^T, \tag{3.32}$$

$$M = [0.0071 \quad -0.0147 \quad -0.0215 \quad 0.0082 \quad 0.0135 \quad 0.0002 \quad -0.0003 \quad 0.0008]^T. \tag{3.33}$$

The MPC weight coefficients used in Eq. (3.1) are selected to be $w^y = 15.26$ and $w^u = 0.63$. The number of control plant outputs n_y is 1 and the number of control plant inputs n_u, is also 1. The prediction horizon was set to $P = 8$ and each control interval k was obtained at a sampling frequency of 1024 Hz, one fourth of the RTHS sampling rate. The applied constraints represent the physical limitations of the actuator to provide bounded displacements and velocity. It is assumed that the virtual actuator has a maximum stroke of ± 250 [mm] and maximum velocity of ± 100 $\left[\frac{mm}{sec}\right]$. So the constraints follow:

$$-250 \leqslant u \leqslant 250 \quad \left[\text{mm}\right], \tag{3.34}$$
$$-100 \leqslant \dot{u} \leqslant 100 \quad \left[\frac{\text{mm}}{\text{sec}}\right].$$

The polynomial extrapolation coefficients used in CS1 for the proposed tracking controller follow:

$$\hat{r}_k = 5r_{(0,k)} - 10r_{(1,k)} + 10r_{(2,k)} - 5r_{(3,k)} + r_{(4,k)}. \tag{3.35}$$

3.3.1.3 *Case study 1 results*

In order to test the robustness of the proposed tracking controller, six dominant parameters of the hybrid model were chosen to vary. The first three parameters originate from the control plant and correspond to its M, m and L, while the remaining three originate from the NS and correspond to M^N, C^N and K^N. These parameters are treated as random with known probability distributions. Their distribution characteristics are described in Table 3.1. CV refers to the coefficient of variation.

Parameter	Probability Distribution	Mean Value	Stand. Dev.	CV (%)	Parameter Description	Units
M	Lognormal	2	0.4	20	Cart mass	Kg
m	Lognormal	0.15	0.03	20	Pendulum mass	Kg
L	Lognormal	0.7	0.21	30	Rod length	m
M^N	Lognormal	100	20	20	Cantilever mass	Kg
C^N	Lognormal	100	30	30	Cantilever damping	$\frac{\text{Ns}}{\text{m}}$
K^N	Lognormal	10e4	3e4	30	Cantilever stiffness	$\frac{\text{N}}{\text{m}}$

TABLE 3.1: CS1: Hybrid model parameter variations with their assigned distributions.

Using the Latin hypercube sampling (LHS) [104] methodology, 200 samples were generated from all six parameters and 200 runs of the vRTHS were conducted using combinations of all parameters in each iteration. The tracking controller was kept the same for each one of the 200 runs.

The simulation of the 200 vRTHS runs is referred to as stochastic vRTHS hereafter. For the scope of the presented study, 200 sample points are considered sufficient to adequately capture the underlying probability space and thus to expose the tracking controller to wide parameter variations. The resulting J_1, J_2 and J_3 outputs of the nominal and the stochastic vRTHS are shown in Table 3.2 for both CS1 and CS2 for brevity. The nominal values correspond to the parameters used in Eqs. (3.23), (3.24) and (3.25). The normalized histograms of the J_1, J_2 and J_3 out of the 200 vRTHS are shown in Figure 3.7. The aforementioned histograms are a more comprehensive, graphical representation of the values presented in Table 3.2, illustrating the mean values as well as the deviation from them. It is also a metric of robustness; more robust tracking controllers would result in lower deviation in the histograms.

	CS1			**CS2**		
	Nominal	Stochastic		Nominal	Stochastic	
		Mean Values	Stand. Dev.		Mean Values	Stand. Dev.
J_1 [msec]	0.24	0.11	0.12	0	0	0
J_2 [%]	1.88	1.96	0.67	1.88	2.93	0.63
J_3 [%]	1.86	1.99	0.57	2.07	3.02	0.60

TABLE 3.2: Tracking controller performance and robustness results for CS1 and CS2. *Note:* CS2 is addressed in the next section 3.3.2. CS2 results are presented here for brevity.

To check if the proposed tracking controller remains stable as the hybrid model parameters vary, vRTHS simulations using the minimum and the maximum values of the random variables were conducted first. No instabilities were observed. Furthermore, none of the conducted 200 simulations was unstable. The same holds for CS2.

The reference, command and measured signals of the hybrid model in the nominal vRTHS are illustrated in Figure 3.8. The reference signal corresponds to the displacement response of the reference model (one with integrated PSs and NSs). The command signal corresponds to the displacement response r computed from the NS at each given time-step of vRTHS and is the one that should be followed from the control plant. Finally, the measured signal corresponds to the measured displacement response x of the vPS. An ideal tracking controller should be able to compensate the hybrid model in such a way that the command r and measured x to be identical. As it is shown from Figure 3.8, those two signals are, indeed,

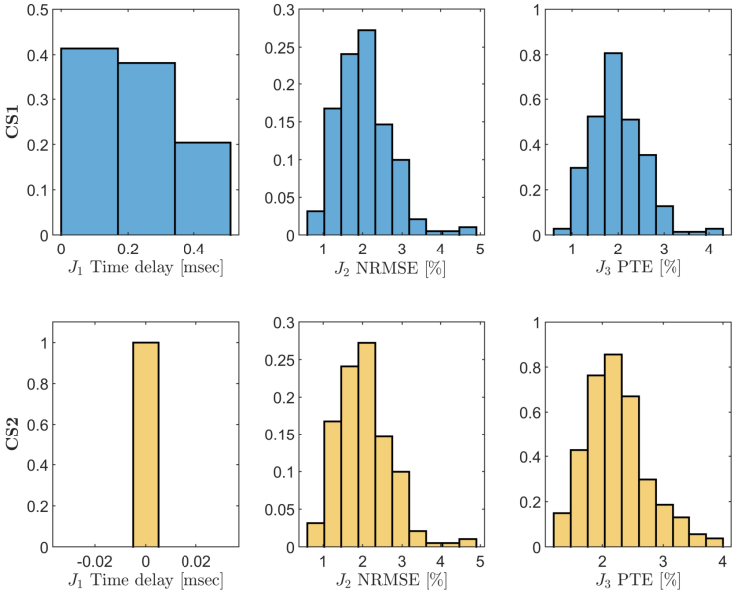


FIGURE 3.7: Normalized histograms of J_1, J_2 and J_3 for CS1 and CS2, obtained from 200 vRTHS runs.

very close. The comparison with the reference signal is provided in order to validate the fidelity of the hybrid model with respect to the reference structure.

In Figure 3.9, the performance index of the MPC versus time for the nominal case is displayed. This graph illustrates how well MPC managed to minimize the given objective cost function of Eq. (3.1) in every time-step of the simulation. A zero value would mean that the cost function was minimized as desired and the "best" optimal control sequence was computed for the given time-step. From Figure 3.9, we can observe that the performance index is almost zero during the entire vRTHS, while it is not zero in the time-steps in which the highest peaks of the reference signal are attained. This is expected, as the peaks of the command signal are approached, the controller is challenged more and more and has to adapt.

Since the performance of the tracking controller is assessed by how close to zero J_1, J_2 and J_3 are, it is clear from Table 3.2 and Figure 3.7-3.9

that the proposed tracking controller can provide the desired performance under the presence of any combination of all six random parameters of the hybrid model chosen here, which also demonstrates its robustness. The effects of these stochastic parameters could represent the effect of potential uncertainties (aleatory and/or epistemic) that could be introduced during RTHS. On top of that, it should be pointed out that the controller maintains its performance even in the presence of the additional noise n and disturbance d that were added in the hybrid model.

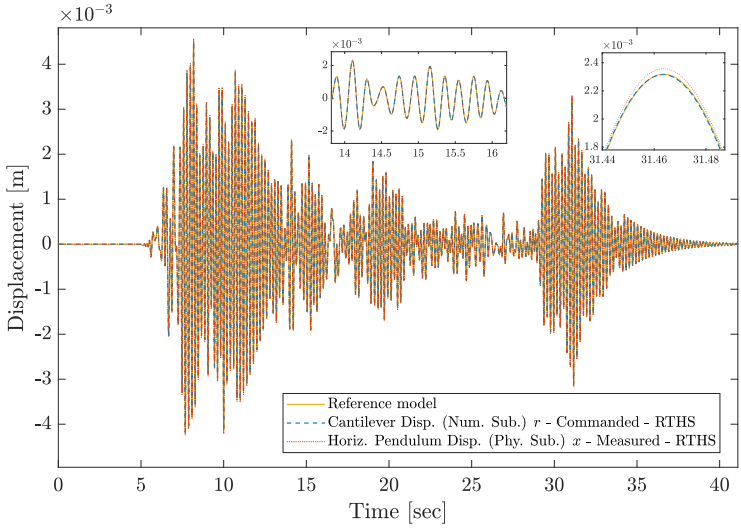


FIGURE 3.8: Displacement response of the reference model and the numerical and physical vRTHS substructures in CS1.

3.3.2 *Case study 2: virtual RTHS of a magnetorheological damper attached to a 3-story structure*

3.3.2.1 *Problem Formulation*

The reference structure in CS2 corresponds to a 3-story structure equipped with a magnetorheological damper (MRD), installed between the ground and first floor [105] as shown in Figure 3.10(a). The NS corresponds to the 3-story structure, Figure 3.10(b), while the virtual physical to the MRD, Figure 3.10(c).

The EoM of the reference model reads:

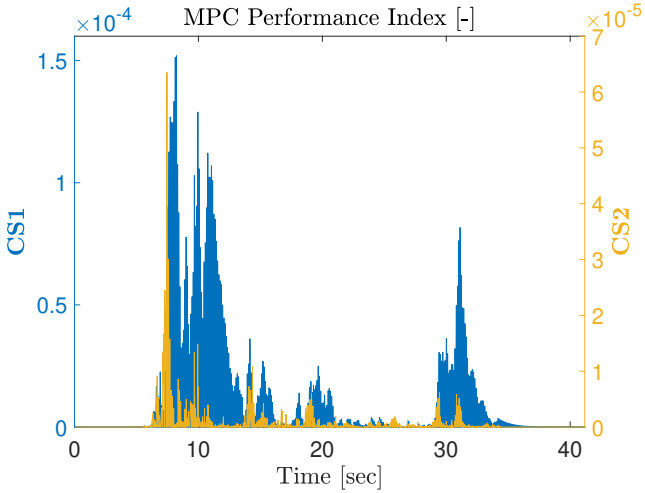


FIGURE 3.9: MPC optimization performance index for CS1 and CS2.

$$\mathbf{M}^N\ddot{\mathbf{x}} + \mathbf{C}^N\dot{\mathbf{x}} + \mathbf{K}^N\mathbf{x} = -\mathbf{M}\boldsymbol{\Lambda}\ddot{x}_g + \boldsymbol{\Gamma}f^P, \tag{3.36}$$

where $\mathbf{x} = [x_1, x_2, x_3]^T$, $\dot{\mathbf{x}} = [\dot{x}_1, \dot{x}_2, \dot{x}_3]^T$ and $\ddot{\mathbf{x}} = [\ddot{x}_1, \ddot{x}_2, \ddot{x}_3]^T$ correspond to the displacement, velocity and acceleration relative to the ground, $\ddot{x}_g$ is the ground motion and f^P corresponds to the force generated from the MRD. The $\mathbf{M}^N, \mathbf{C}^N, \mathbf{K}^N$ matrices represent the mass, damping and stiffness of the 3-story structure, respectively, as follows:

$$\begin{aligned}
\mathbf{M}^N &= \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix} [\mathrm{Kg}], \\
\mathbf{C}^N &= 1e4\times \begin{bmatrix} 1.408 & -0.787 & 0.044 \\ -0.787 & 1.494 & -0.635 \\ 0.044 & -0.635 & 0.722 \end{bmatrix} [\mathrm{Ns/m}], \\
\mathbf{K}^N &= 1e7\times \begin{bmatrix} 2.605 & -2.313 & 0.594 \\ -2.313 & 3.256 & -1.442 \\ 0.594 & -1.442 & 0.927 \end{bmatrix} [\mathrm{N/m}].
\end{aligned} \tag{3.37}$$

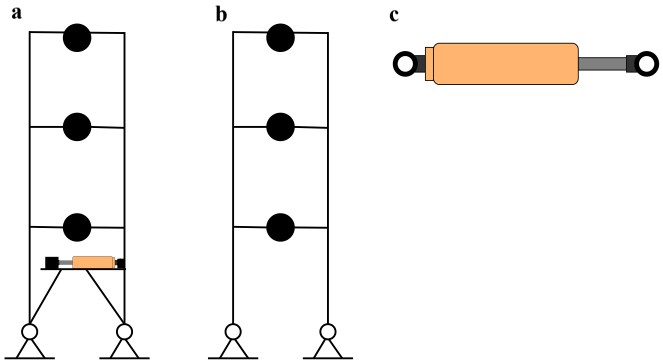


FIGURE 3.10: Hybrid model of CS2: (a) reference structure, (b) NS, (c) vPS.

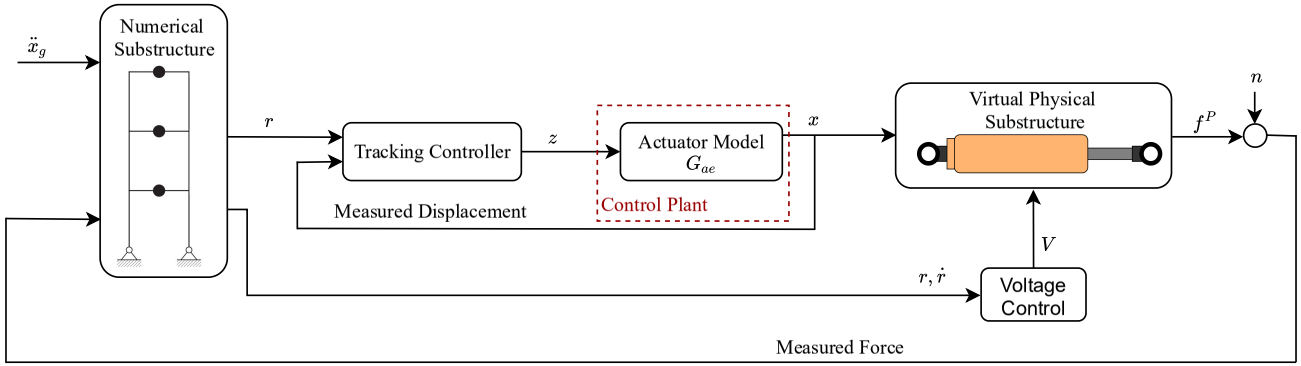


FIGURE 3.11: Block diagram of the hybrid model in CS2.

Vector $\boldsymbol{\Lambda} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$ is the ground motion influence vector, while vector $\boldsymbol{\Gamma} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ represents the effect of the MRD to the structure. In RTHS, a state-space representation of the Eq. (3.36) is used, which follows:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \end{aligned} \tag{3.38}$$

where $\mathbf{u} = [\ddot{x}_g, f^P]^T$, $\mathbf{y} = [x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3, \ddot{x}_1, \ddot{x}_2, \ddot{x}_3]^T$, $\mathbf{x} = [x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3]^T$ and

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{M}^{N^{-1}}\mathbf{K}^N & -\mathbf{M}^{N^{-1}}\mathbf{C}^N \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ -\boldsymbol{\Lambda} & -\mathbf{M}^{N^{-1}}\boldsymbol{\Gamma} \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{I} \\ -\mathbf{M}^{N^{-1}}\mathbf{K}^N & -\mathbf{M}^{N^{-1}}\mathbf{C}^N \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\boldsymbol{\Lambda} & -\mathbf{M}^{N^{-1}}\boldsymbol{\Gamma} \end{bmatrix}. \end{aligned} \tag{3.39}$$

The block diagram of the hybrid model of CS2 is shown in Figure 3.11. The reference signal r, in Figure 3.11, corresponds to the displacement of the first story x_1. Respectively, $\dot{r} = \dot{x}_1$. RTHS is conducted in displacement control, as in CS1. The ground motion applied to the hybrid model is the same as in CS1, a historical acceleration record from the El Centro 1940 earthquake downscaled by 0.4. As in CS1, apart from the additive disturbance described in Eq. (3.2) with values: $A_d = 1, B_d = 0.0009766, C_d = 1$ and $D_d = 0$, additional white noise n is added in the calculated force from MRD f^P, which represents measurement noise from the load cell. The noise is generated from a standard Gaussian distribution amplified by 0.15 N, which approximately equals to 0.01% of its full span. The matrix values of Eq. (3.3) admit in CS2: $A_n = 0, B_n = 0, C_n = 0$ and $D_n = 1$. The sampling frequency of RTHS was set to f_{RTHS} = 4096 Hz. For the time integration scheme, the RK4 method is used with a fixed time-step of 1/4096 sec.

To model the vPS, the MRD in CS2, the Viscous + Dahl model [106] was employed. Its dynamics are described as follows:

$$
\begin{aligned}
f^P(t) &= \left[k_{x_a} + k_{x_b} V(t)\right] \dot{x}(t) + \left[k_{w_a} + k_{w_b} V(t)\right] W(t), \\
\dot{W}(t) &= \rho\left(\dot{x}(t) - |\dot{x}| W(t)\right), \\
W(0) &= \frac{f^P(0) - \left[k_{x_a} + k_{x_b} V(0)\right] \dot{x}(0)}{k_{w_a} + k_{w_b} V(0)},
\end{aligned}
\tag{3.40}
$$

where $\dot{x}(t)$ denotes the MRD piston velocity, $V(t)$ the voltage input command, $f^P(t)$ the damping force, W the damper's nonlinear behavior, k_{x_a} and k_{x_b} the viscous friction coefficient, k_{w_a} and k_{w_b} the dry friction coefficient and t refers to the simulation time. The parameter ρ is calculated as in [107] and selected to be $\rho = 4795\ (m^{-1})$, while the friction parameters admit $k_{x_a} = 978\ (Nsm^{-1})$, $k_{x_b} = 4075\ (Nsm^{-1}V^{-1})$, $k_{w_a} = 60.11\ (N)$ and $k_{w_b} = 344.78\ (NV^{-1})$. The latter parameters are identified as described in [108]. The inputs of the MRD model are the displacement $x(t)$ and the voltage $V(t)$, while the output is the force f^P. The latter is the variable that couples the two substructures of the hybrid model in CS2.

In a MRD, a relatively small electric current applied to the MR valve can change the behavior from very high to very low resistance to motion over a very short time period. In order to ensure an optimal response, a bang-bang voltage controller is designed and implemented as illustrated in Figure 3.11. More specific, when $\mathrm{sgn}(r(t)) = \mathrm{sgn}(\dot{r}(t))$ then the controller provides the MRD with the maximum input voltage, resulting in maximum MRD force f^P. Otherwise, the MRD force is minimum. This bang-bang controller is part of the MRD and it is exclusively responsible for the internal behavior of the MRD.

In CS2, a different approach of the control plant is investigated compared to CS1, since in this case, the control plant corresponds only to the actuator model. The latter is in contrast with the control plant in CS1, which was represented by the actuator model in series with the vPS. Results presented later on prove that the compensation of time delays is sufficient and the performance of RTHS desired. Moreover, in this way, the dynamics of the control plant are much simpler. Hence, the complexity of the tracking controller is reduced. Therefore, in CS2 the control plant is a SISO model described by Eq. (3.41) with input the desired displacement of the actuator z and output x, the achieved displacement of the actuator. So, in the J_1, J_2 and J_3 criteria, the measured signal $y(i)$ of Eqs. (3.20), (3.21) and (3.22) corresponds to the actuator achieved displacement x. Furthermore, in order to expose the proposed tracking controller under different actuator scenarios, the actuator model used in CS2 corresponds to an electric actua-

tor, represented by a second-order transfer function, G_{ae}, described by the dynamics of Eq. (3.41).

$$G_{ae} = \frac{3060}{s^2 + 267s + 3060}. \tag{3.41}$$

3.3.2.2 *Tracking Controller Design Properties*

As in CS1, the prediction model used in MPC is the control plant, discretized by the sampling frequency of RTHS. The state-space formulation of the discretized model follows Eq. (3.7) with:

$$A_p = \begin{bmatrix} 0.7693 & -0.0411 \\ 0.055 & 1 \end{bmatrix}, B_p = \begin{bmatrix} 0.0069 \\ 0.0002 \end{bmatrix}, C_p = \begin{bmatrix} 0 & 5.98 \end{bmatrix}, D_p = 0. \tag{3.42}$$

The Kalman filter gain matrices in this case are:

$$\begin{aligned} L &= \begin{bmatrix} -0.001 & 0.723 & 0.974 \end{bmatrix}^T \times 1e-3, \\ M &= \begin{bmatrix} 0.037 & 0.721 & 0.974 \end{bmatrix}^T \times 1e-3. \end{aligned} \tag{3.43}$$

In CS2, the MPC parameters are selected as follows:

- $n^y = 1$ and $n^u = 1$
- $w^y = 64.073$ and $w^u = 0.002$
- $P = 10$
- Each control interval k is obtained on a sampling frequency of 1024 Hz.
- The constraints remain the same with Eq. (3.34).

The polynomial extrapolation coefficients are the same as in Eq. (3.35).

3.3.2.3 *Case study 2 results*

As in CS1, in order to test the robustness of the tracking controller, six dominant parameters are selected to be random variables with known probability distributions. The first four originate from the NS:

- Beam length $\rightarrow L_b$
- Column length $\rightarrow L_c$
- Floor mass $\rightarrow M^N$
- Damping ratio $\rightarrow Z$

The remaining two parameters correspond to the vPS and more specific to K_{x_a} and K_{w_a}. These parameters are of particular importance for the MRD model since they are responsible for its nonlinear behavior. All six parameters along with their distribution characteristics are displayed in Table 3.3. As in the previous case study, 200 samples are generated with the LHS method from the six parameters, and 200 vRTHS runs are conducted accounting for the variability of all parameters in each run. Again the tracking controller was kept the same in all vRTHSs. The nominal case for CS2 are the parameter values from Eqs. (3.37) and (3.40). The arithmetic results for J_1, J_2 and J_3 can be found in Table 3.2. Their corresponding normalized histograms for the stochastic vRTHS are illustrated in Figure 3.7.

Parameter	Probability Distribution	Mean Value	Stand. Dev.	CV (%)	Parameter Description	Units
L_b	Lognormal	0.762	0.1524	20	Beam length	m
L_c	Lognormal	0.635	0.127	20	Column length	m
M^N	Lognormal	1000	200	20	Floor mass	Kg
Z	Lognormal	0.05	0.01	20	Damping ratio	-
K_{x_a}	Lognormal	978	195.6	20	Viscous frict. coef. of MRD	$\frac{Ns}{m}$
K_{w_a}	Lognormal	60.11	12.022	20	Dry frict. coef. of MRD	N

TABLE 3.3: CS2: Hybrid model parameter variations with their assigned distributions.

In Figure 3.12, the force generated by the MRD in the reference model is compared against the one obtained from the vRTHS framework. We can observe that the forces are almost identical. This serves as a demonstration that, although the vPS was not included in the control plant, since the latter consists only of the actuator model, its response is compensated sufficiently

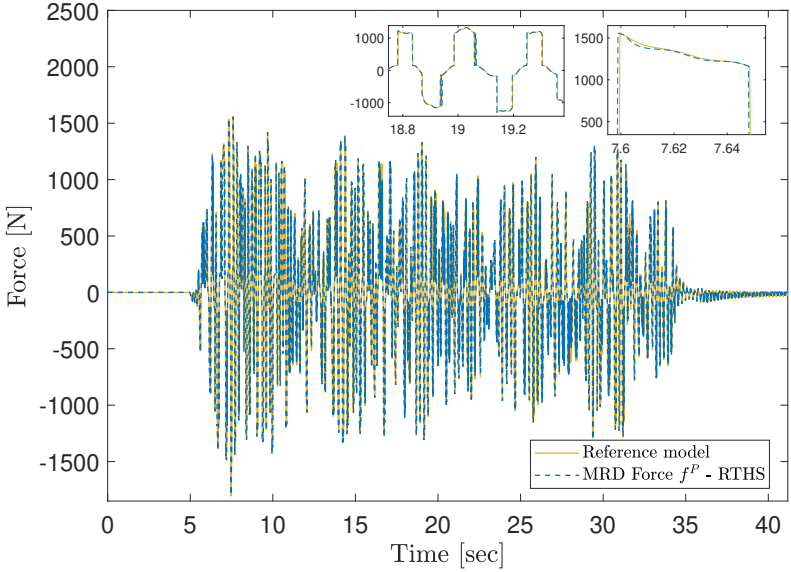


FIGURE 3.12: MRD generated force in the reference model and in vRTHS in CS2.

from time delays and tracking errors. In Figure 3.13 a comparison between three displacement responses is shown; the displacement response of the reference model, the displacement response of the first floor of the NS r (this is the command signal to the control plant), and the displacement response x, measured from the control plant (this should track r). The latter two signals prove that the performance of the tracking controller is as desired, as Figure 3.13 serves as a graphical illustration of the nominal results shown in Table 3.2. We can observe that due to the proposed controller, x follows the commanded r with minimum delay and tracking error. Finally, in Figure 3.9, the performance index of MPC for CS2 is illustrated.

As in the previous case study, from the Table 3.2, Figures 3.7 and 3.13, it is shown that the controller performance does not get affected by the presence of the introduced random variables, and it provides the requested performance in all considered cases.

3.4 Conclusions

In this chapter, a novel control method for RTHS is presented. The proposed tracking controller aims to conduct RTHS in hard-real time while compen-

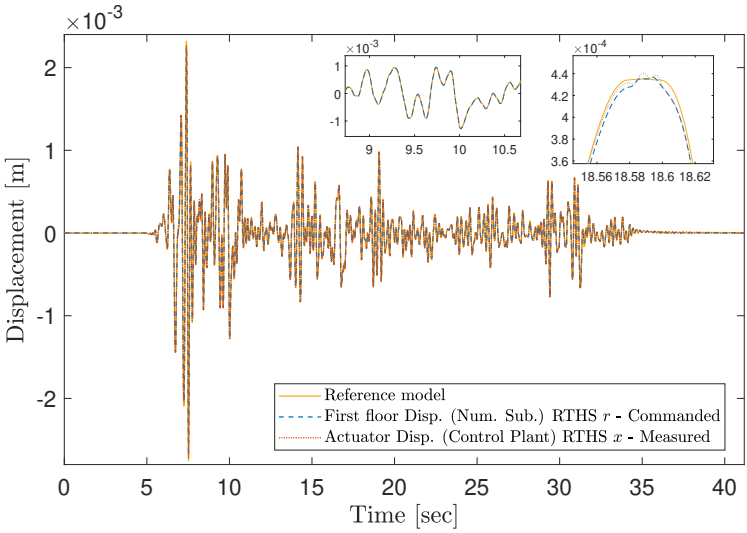


FIGURE 3.13: Displacement response of the reference model, the NS and control plant in CS2.

sating for potential time delays and tracking errors, under the uncertainties that may arise during RTHS. The tracking controller consists of a classical MPC along with a polynomial extrapolation algorithm and a Kalman filter. The fact that MPC can solve optimization problems online, adapt the new control laws during RTHS using the same model of the system, and simultaneously handle constraints for the system under consideration, indicates that the proposed novel control method is promising for RTHS applications. Polynomial extrapolation was employed to further assist MPC performance, as even small tracking errors can alter the hybrid model's dynamic response. A Kalman filter was used so as to provide MPC with low-noise responses of the system, in order to derive optimal control laws. In this chapter, the proposed tracking controller formulation was addressed first, followed by two vRTHS parametric case studies to assess the performance and robustness of the tracking controller. Dominant parameters of the hybrid model in both case studies were selected and given random perturbations via prescribed probability distributions, varied with at least a 20% coefficient of variation. In each case study, 200 samples were generated from the random parameters and 200 vRTHS runs were conducted in order to verify if the proposed tracking controller was robust enough to maintain the desired performance under the introduced uncertainties. Furthermore,

a random disturbance was added in the hybrid model loop along with an additional white noise additive to the measured signals. The added disturbance and noise represent systematic or random errors occurring in a real experiment. Since the two case studies were virtual, actuator models had to be developed in order to simulate actuator dynamics. Two different actuators models were employed in order to assess the tracking controller performance in a wider range of potential experimental equipment. Results from the two case studies illustrate that the proposed tracking controller can guarantee very small time delays and tracking errors under uncertainties that may be introduced in RTHS. Notably, the delays and errors were very close to zero in both case study reference models. Therefore, RTHS using the proposed tracking controller scheme is demonstrated to be effective for structural performance assessment.

Chapter 4

Adaptive model predictive control in real-time hybrid simulation

The material presented in this chapter is submitted for publication and is currently under review. A preprint can be found in [84]. Part of the work is also presented in the International Symposium on Co-Simulation and Solver Coupling in Dynamics (COSIM) held in Ferrol, Spain, in 2021 [109].

4.1 Introduction

As shown in the latter chapter, model predictive control (MPC) is demonstrated to be an effective control strategy for RTHS, as a key advantage of MPC is its ability to perform online optimization as well as satisfying constraints of the examined system, e.g. actuation system capacity limitations. However, in MPC the online optimization is performed based on predictions of the future dynamic response of the system under control. Therefore, the performance of the optimization depends on the accuracy of the predictions. Classical MPC utilizes a prediction model, i.e. a LTI model, for these predictions. In case the dynamics of the examined system are high nonlinear or linear with time-varying parameters, the predictions from a LTI model may not be accurate and thus the performance of MPC degrades. In this regard, *adaptive MPC* (AMPC) is a suitable solution, since the LTI model used to predict the dynamic response of the controlled system is adapting during the simulation in order to capture the changing dynamics of the system under control. Some of the common practices for model updating in AMPC include online system identification and successive linearization [110, 111].

Therefore, this chapter continues to address the first research objective of this dissertation. In particular, in this chapter a novel tracking controller for dynamics compensation in RTHS is presented, extending the classical MPC introduced in Chapter 3, to an adaptive scheme. It is based on AMPC, linear time-varying Kalman filter and on an online model identification algorithm. With the latter, auto-regressive exogenous (ARX) polynomial models are identified online, at each control interval, and are used to estimate the changing control plant dynamics and to update the prediction model for the AMPC. A parametric virtual RTHS (vRTHS) case study is employed to

demonstrate and validate the performance and robustness of the proposed controller.

This chapter is organized as follows. Section 4.2 describes the tracking controller. Section 4.3 introduces the vRTHS case study where the proposed controller is developed, validated and demonstrated. The obtained results are also presented in this section. Section 4.4 presents the overall conclusions of this chapter.

4.2 Outline of the tracking controller

In this section, the individual components of the proposed tracking controller are described. As mentioned above, these correspond to the AMPC, the linear time-varying Kalman filter and an online model identification algorithm, which is used to update the prediction model of AMPC. Figure 4.1 depicts the block diagram of the controller. The control plant corresponds to the system under control. The latter is addressed in more detail in Section 4.3.

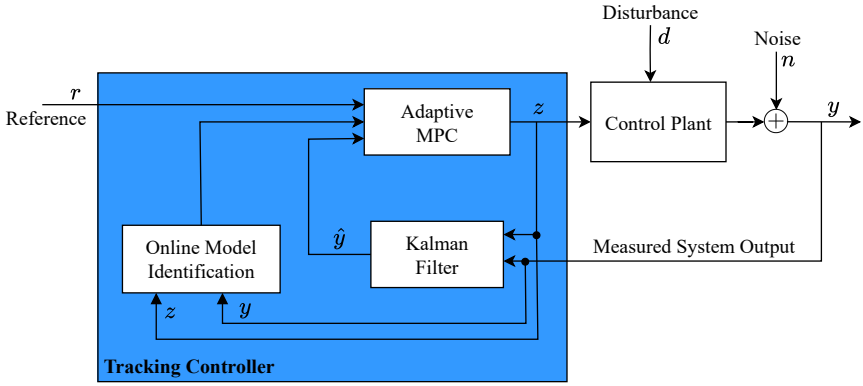


FIGURE 4.1: Architecture of the proposed tracking controller based on AMPC.

4.2.1 *Adaptive model predictive control*

The structure and methodology of AMPC are the same as in the classical MPC, which was presented in Chapter 3. What differs nonetheless, is that in AMPC the prediction model used internally in the controller is updating, i.e. adapting, online during the simulation. Recall that in classical MPC, the prediction model was defined offline and was kept constant throughout

the simulation. The AMPC control strategy is rather desirable for control plants with high nonlinear or linear with time-varying dynamics.

In the context of this dissertation, the remaining components of MPC, namely the cost function, constraints formulation and the solver used to derive the control laws, are the same for MPC and AMPC. For brevity, the cost function, that is optimized in each control interval k, is presented also here:

$$J^*(r_k, \hat{y}_k, z_k) = \sum_{j=1}^{n_y} \sum_{i=1}^{P} \left\{ w^{y_j} [r_j(k+i|k) - \hat{y}_j(k+i|k)] \right\}^2 + \sum_{j=1}^{n_u} \sum_{i=0}^{P-1} \left\{ w^{u_j} [u_j(k+i|k) - u_j(k+i-1|k)] \right\}^2, \tag{4.1}$$

where n_y is the number of control plant outputs, n_u the number of control plant inputs, $r_j(k+i|k)$ the reference value to be tracked at the i-th time instant step from the j-th control plant output, $\hat{y}_j(k+i|k)$ the predicted value of the j-th control plant output at the i-th time instant, $u_j(k+i|k)$ the j-th control plant input at the i-th time instant, w^{y_j} the tuning weight of the j-th control plant output and w^{u_j} the tuning weight of the j-th control plant input. As in the classical MPC, the P, w^{y_j} and w^{u_j} parameters of Eq. (4.1) are computed offline and remain constant during the simulation.

Note that in the MPC-based controller of Chapter 3, a polynomial extrapolation algorithm was used that is not utilized in the AMPC-based controller, introduced in this chapter. This is because during the controller's design phase it was found that the polynomial extrapolation for this case was not significantly contributing to improve the performance of the overall tracking controller. Hence, in order to simplify the controller architecture, the polynomial extrapolation was not implemented. This can be justified since, by the online adaption of the prediction model in AMPC, its performance is already improved.

In addition, as in the classical MPC of the previous chapter, an output disturbance d and a measurement noise model n are used in the proposed tracking controller. Their state-space formulation is the same and for simplicity they are presented here as well, in Eqs. (4.2) and (4.3), respectively. Recall, that they are implemented as additive to the control plant output, representing potential disturbances and sensor noise that could be present in RTHS.

$$\begin{aligned} x_d(k+1) &= A_d x_d(k) + B_d u_d(k) \\ d(k) &= C_d x_d(k) + D_d u_d(k), \end{aligned} \tag{4.2}$$

$$\begin{aligned} x_n(k+1) &= A_n x_n(k) + B_n u_n(k) \\ n(k) &= C_n x_n(k) + D_n u_n(k), \end{aligned} \tag{4.3}$$

where A_d, B_d, C_d, D_d and A_n, B_n, C_n, D_n are matrices associated with the disturbance d and noise n respectively. The inputs to these state-space models, namely u_d for the disturbance and u_n for the noise, are Gaussian variables with zero mean and unit variance.

The utilized AMPC algorithm is summarized as follows:

1. **Prediction step:** Considering a discrete time MIMO system that represents a linearized model of the control plant, the prediction model in state-space formulation at the control interval k follows:

$$\begin{aligned} x_p(k+1) &= A_p^k x_p(k) + B_p^k u(k) \\ y_p(k) &= C_p^k x_p(k) + D_p^k u(k), \end{aligned} \tag{4.4}$$

   where A_p^k, B_p^k, C_p^k and D_p^k are matrices associated with the prediction model at the control interval k, representing the control plant dynamics. The disturbance and noise models of Eqs. (4.2) and (4.3) respectively, are assumed to be additive to the measured control plant output, y signal in Figure 4.1, and hence the latter admits: $y(k) = y_p(k) + d(k) + n(k)$.

2. **Identification step:** At each control interval k, an ARX polynomial model of the control plant is identified and used to adapt the prediction model of AMPC, i.e. the matrices A_p^k, B_p^k, C_p^k and D_p^k, (see Section 4.2.3 for a more detail description of the algorithm).

3. **Update step:** At each control interval k, the L and M vectors of the Kalman filter are updated to be consistent with the adapted prediction model (see Section 4.2.2).

4. **Optimization step:** Using the current state estimates and predicted control plant outputs from the Kalman filter (Eqs. (4.15) and (4.18)), AMPC solves the optimization problem described by Eq. (4.1) for the current control interval k:

$$\min_{z_k} J^*(r_k, \hat{y}_k, z_k) \tag{4.5}$$

$$\text{subject to} \quad u_{j_{min}} \leqslant u_j \leqslant u_{j_{max}} \quad \text{and} \tag{4.6}$$

$$\dot{u}_{j_{min}} \leqslant \dot{u}_j \leqslant \dot{u}_{j_{max}}, \tag{4.7}$$

where the above constraints represent safety limits regarding input signals to the actuation system.

5. **Control step:** The control sequence for the current control interval $z_k^T = [u(k|k)^T \dots u(k+i|k)^T \dots u(k+P-1|k)^T]$ is computed and its first control action is applied to the control plant:

$$z_k := \arg\min J^*(r_k, \hat{y}_k, z_k) \tag{4.8}$$

6. Steps 1-5 are repeated till the completion of the RTHS.

Updating the prediction model of AMPC during the simulation requires implementation of online system identification techniques. Depending on the application context, the latter may require a significant amount of computational power, risking the time scale of RTHS. In this regard, care must be taken not to introduce time delays due to such computational needs. A way to overcome such challenges is to decrease the frequency that each control interval is sampled and hence reduce the frequency the online model identification algorithm is activated to adapt the AMPC's prediction model. On the contrary, decreasing the frequency each control interval is obtained could degrade the controller's robustness and performance.

4.2.2 *Linear time-varying Kalman filter*

As mentioned in Chapter 3, the performance of the MPC depends on the accuracy of the predicted control plant outputs. In order to provide the MPC with low-noise plant outputs, a Kalman filter is used. Furthermore, the Kalman filter provides estimates of the plant outputs when there are none available from the plant sensors. Since the prediction model of AMPC is adapted online, the gain vectors of the Kalman filter also need to adapt online, resulting in a linear time-varying Kalman filter. Considering Eqs. (4.4) and (4.2), the state-space formulation including the control plant dynamics and the disturbance model follows:

$$\begin{aligned} x_c(k+1) &= A_c^k x_c(k) + B_c^k u_c(k) \\ \hat{y}(k) &= C_c^k x_c(k) + D_c^k u_c(k), \end{aligned} \tag{4.9}$$

where

$$\begin{aligned} A_c^k &= \begin{bmatrix} A_p^k & 0 \\ 0 & A_d \end{bmatrix}, \quad B_c^k = \begin{bmatrix} B_p^k & 0 \\ 0 & B_d \end{bmatrix}, \\ C_c^k &= \begin{bmatrix} C_p^k & C_d \end{bmatrix}, \quad D_c^k = \begin{bmatrix} D_p^k & D_d \end{bmatrix}, \\ x_c^T &= \begin{bmatrix} x_p^T & x_d^T \end{bmatrix} \text{ and } u_c^T = \begin{bmatrix} u^T & u_d^T \end{bmatrix}. \end{aligned} \tag{4.10}$$

A_p^k, B_p^k, C_p^k and D_p^k matrices are updated online, at each control interval k, while A_d, B_d, C_d, D_d of Eq. (4.2) are considered constant during the RTHS. The same holds for A_n, B_n, C_n, D_n of the noise model in Eq. (4.3). The x_c, u_c vectors contain the states and inputs of the plant p and the disturbance d, respectively. As in Chapter 3, the weighting coefficients Q, R, N of the Kalman filter are constant during the RTHS and are computed from the following expectations:

$$Q = \mathbb{E}\left[dd^T\right], \quad R = \mathbb{E}\left[nn^T\right], \quad N = \mathbb{E}\left[dn^T\right]. \tag{4.11}$$

Figure 4.1 displays the interconnection of the Kalman filter within the proposed tracking controller. In more detail, at each control interval k, the x_c states are estimated as follows:

1. **Gain computations:** The gain vectors L, M of the Kalman filter are updated to be consistent with the adapted prediction model:

$$L^k = \left(A_c^k P^{k|k-1} C_c^{k^T} + N\right)\left(C_c^k P^{k|k-1} C_c^{k^T} + R\right)^{-1}, \tag{4.12}$$

$$M^k = P^{k|k-1} C_c^{k^T}\left(C_c^k P^{k|k-1} C_c^{k^T} + R\right)^{-1}, \tag{4.13}$$

$$P^{k+1|k} = A_c^k P^{k|k-1} A_c^{k^T} - \left(A_c^k P^{k|k-1} C_c^{k^T} + N\right) L^{k^T} + Q, \tag{4.14}$$

where L^k, M^k denote the L, M gain vectors at the control interval k and $P^{k+1|k}$ the state estimate error covariance matrix at $k+1$, calculated with information obtained from the k control interval.

2. **Measurement update step:** The current state estimate $x_c(k|k)$ is adjusted with the latest measurements:

$$x_c(k|k) = x_c(k|k-1) + M^k\left[y(k) - C_c^k x_c(k|k-1)\right], \tag{4.15}$$

where $x_c(k|k-1)$ denotes the state estimate from the $k-1$ control interval.

3. **Estimation step:** The state for the next, $k+1$, control interval is estimated as:

$$x_c(k+1|k) = A_c^k x_c(k|k-1) + B_{c_1}^k u(k) + L^k\left[y(k) - C_c^k x_c(k|k-1)\right], \tag{4.16}$$

where $u(k)$ the control action used from $(k-1)$ till k, $y(k)$ the control plant output measured at k and $B_{c_1}^k$ the first column of B_c^k.

Once the current state of the control plant is estimated, the controller predicts control plant outputs for the entire prediction horizon of the current control interval.

1. For any successive time instant of the prediction horizon, $i = 1, \ldots, P$, the predicted control plant states are obtained by:

$$x_c(k+i|k) = A_c^k x_c(k+i-1|k) + B_{c1}^k u(k+i-1|k). \tag{4.17}$$

2. The predicted control plant output for $i = 1, \ldots, P$ is obtained by:

$$\hat{y}(k+i|k) = C_c^k x_c(k+i|k). \tag{4.18}$$

4.2.3 *Online model identification algorithm*

At each control interval k, an online model identification algorithm is employed and ARX polynomial models [112] of the control plant are identified and then used to update the prediction model of the AMPC. The formulation of ARX polynomial models follow:

$$A(q)y(t) = B(q)u(t) + e(t), \tag{4.19}$$

where $A(q) = 1 + \alpha_1 q^{-1} + \alpha_2 q^{-2} + \cdots + \alpha_{na} q^{-na}$, $B(q) = b_1 + b_2 q^{-1} + b_3 q^{-2} + \cdots + b_{nb} q^{-(nb-1)}$, q is a time shift operator and $u(t)$, $y(t)$ and $e(t)$ are the inputs, outputs and error respectively.

The model identification is based on the recursive infinite-history algorithm [113] following:

$$\theta(t) = \theta(t-1) + K(t)\big(y(t) - y_{pred}(t|\theta)\big), \tag{4.20}$$

where

$$\theta(t) = \begin{bmatrix} \alpha_1(t) & \alpha_2(t) & \dots & \alpha_{na}(t) & b_1(t) & \dots & b_{bn}(t) \end{bmatrix}, \tag{4.21}$$

corresponds to the identified parameters at time t, $y(t)$ the measured control plant output at t, $y_{pred}(t|\theta) = \psi^T(t)\theta(t-1)$ the prediction of $y(t)$ accounting for measurements up to $t-1$ and $K(t) = Q(t)\psi(t)$, where $\psi(t)$ is the gradient of $y_{pred}(t|\theta)$ and $Q(t)$ admits:

$$Q(t) = \frac{P(t-1)}{1 + \psi^T(t)P(t-1)\psi(t)}, \tag{4.22}$$

where

$$P(t) = P(t-1) + 1 - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{1 + \psi^T(t)P(t-1)\psi(t)}. \tag{4.23}$$

$P(t)$ represents the covariance of parameter identification error with $P(0) = 1$.

At each control interval, the parameter $\theta(t)$ are identified and the $A(q)$, $B(q)$ polynomial are converted to state-space formulation resulting in the A_p^k, B_p^k, C_p^k and D_p^k matrices used in Eq. (4.4).

As mentioned earlier, a Kalman filter is used in the control scheme for state estimation and the recursive infinite-history algorithm for online parameter identification of the prediction model. As displayed in Figure 4.1, these are implemented as two separate blocks, in a decentralized approach. This approach offers the flexibility of independent design of the state and the parameter estimation algorithms. Consequently, a variety of combinations of different online model identification algorithms can easily be tested. A single joint algorithm, a centralized approach, featuring both state and parameter estimation could also be implemented, but this is not examined in this study.

4.3 Case study: virtual RTHS of a prototype motorcycle

To validate the performance and robustness of the proposed AMPC tracking controller, a parametric virtual case study is utilized. To facilitate the design

and testing of the controller, the hybrid model is simulated virtually in the sense that all of its substructures are simulated numerically and thus virtual PS (vPS) are employed instead of physical specimens. To assess the performance of the controller, the three performance metrics J_1, J_2 and J_3, defined in Eqs. (3.20)-(3.22), are utilized here as well. In this section, the problem formulation of the case study is introduced first, followed by the design properties of the tracking controller. Results are addressed afterwards.

4.3.1 *Problem formulation*

The case study prototype is a motorcycle. The hybrid model of the motorcycle is made of four NS and one vPS. The NS are: i) the engine, ii) the motorcycle body dynamics, iii) the rear wheel braking system and iv) the front wheel braking system. The vPS is the electrically continuously variable transmission (eCVT) of the motorcycle.

The eCVT vPS corresponds to a MIMO model with two sets of one input / one output. The first set is connected to the motorcycle engine NS and the second one to the motorcycle body NS. The latter connection corresponds to the transmission output shaft of the motorcycle. The engine NS simulates the dynamics of the combustion engine. It is represented by a multi-input-single-output (MISO) model, with its inputs being the throttle percentage thr and the angular velocity of the engine ω_{en} and its single output being the torque of the engine τ_{en}. The motorcycle body NS represents the body or chassis dynamics of the motorcycle together with the dynamics of the wheels, tires and suspensions along with the road profile and the environmental driving conditions. It is represented by a MIMO model with 3 sets of one input / one output. The first set is connected to the eCVT vPS with the torque τ_{vd} as input and the angular velocity ω_{vd} of the transmission output shaft as output. The second and third sets are linked to the rear and front wheel braking system NS respectively. Both braking systems are MISO models. The rear wheel braking system NS inputs are the angular velocity of the rear wheel ω_{rw} and the applied force on the brake pedal $F_{br_{rw}}$, while its single output is the braking torque of the rear wheel τ_{rw}. Respectively, the front wheel braking system NS inputs are the angular velocity of the front wheel ω_{fw} and the applied force on the brake lever $F_{br_{fw}}$ and its single output is the braking torque in the front wheel τ_{fw}. Figure 4.2 illustrates the interconnections between NS and the vPS, while Figure 4.3 shows the reference full-scale prototype motorcycle. Figure 4.4 depicts the real eCVT PS that would be utilized in a non-virtual RTHS. To develop the substructures of the hybrid model, the multi-physics

simulation software *Simcenter Amesim* was used. The report of [114] offers a thorough description of the utilized substructures along with the equations governing their motion. To interconnect and co-simulate all substructures as well as the RTHS algorithm in real-time, the *Simcenter real-time platform* was used.

The forces applied on the brake pedal (to activate the rear wheel brakes) and the brake lever (to activate the front wheel brakes) are considered equal and expressed in Newton, $F_{br_{rw}} = F_{br_{fw}}$, and thr, $F_{br_{rw}}$ are defined by Eq. (4.24) and Eq. (4.25) respectively. The maximum applied throttle is 0.5 (50%), which corresponds to half-open throttle. In Figure 4.5 the applied driving scenario of the case study is shown. The road profile, i.e. height of the ground, is expressed by the $h(x)$ sinusoidal signal and follows Eq. (4.26), where x is the current motorcycle position in meters. The ambient wind velocity is considered to be zero. The simulation duration of the case study is 45 sec and the RK4 method with a fixed time-step of 0.1 msec is used as the numerical integration scheme in the conducted RTHS of the dynamic response of the motorcycle virtual hybrid model during the driving scenario. Figure 4.6 displays some indicative dynamic responses of the virtual hybrid model, excited under the driving scenario on the given road profile and in the given wind conditions.

$$thr(t) = \begin{cases} 0.125t & , 5 \leqslant t < 9 \\ 0.5 & , 9 \leqslant t \leqslant 13 \\ -0.125t & , 13 < t \leqslant 17 \\ 0 & , \text{elsewhere} \end{cases} \tag{4.24}$$

$$F_{br_{rw}}(t) = \begin{cases} 10t & , 20 \leqslant t < 25 \\ 50 & , 25 \leqslant t \leqslant 32 \\ -10t & , 32 < t \leqslant 37 \\ 0 & , \text{elsewhere} \end{cases} \tag{4.25}$$

$$h(x) = \begin{cases} 0 & , 0 \leqslant x \leqslant 2 \\ 0.02\Big(\cos\left(\frac{2\pi x}{3}\right) - 1\Big) & , \text{elsewhere} \end{cases} \tag{4.26}$$

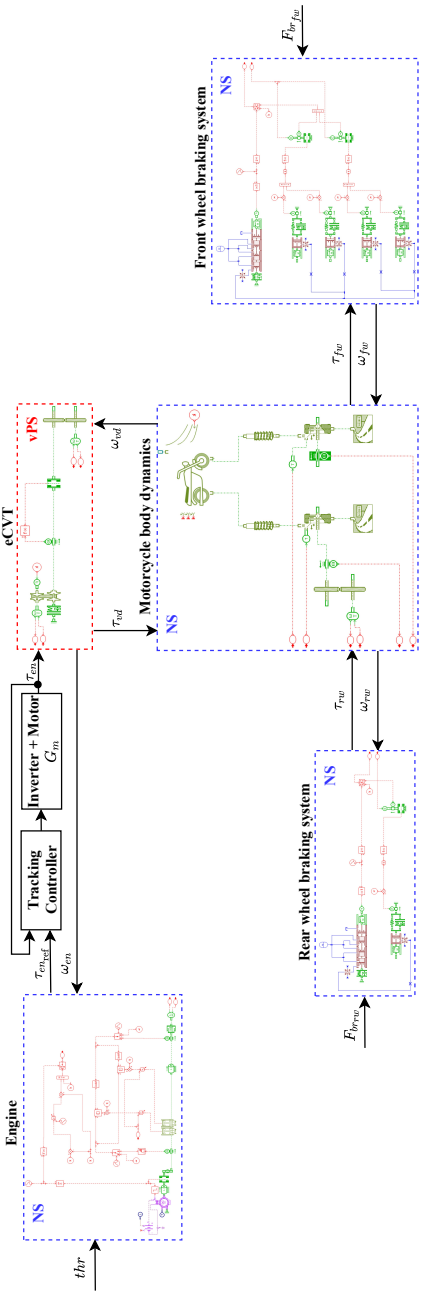


FIGURE 4.2: Motorcycle hybrid model block diagram.

FIGURE 4.3: Reference full-scale prototype motorcycle.

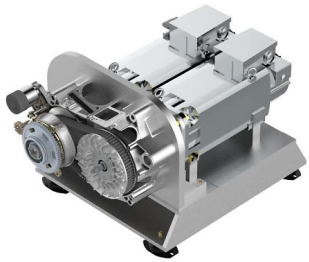


FIGURE 4.4: The eCVT test bench at the testing facilities of Siemens Industry Software.

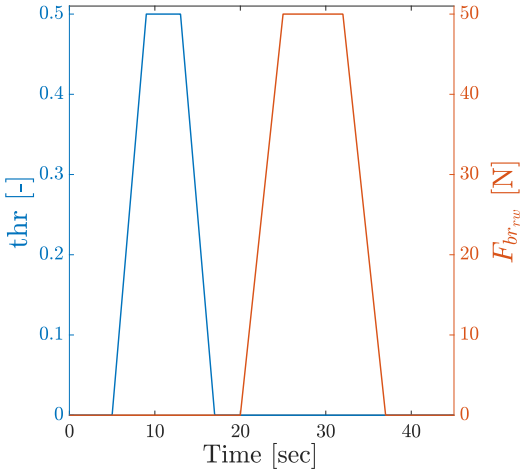


FIGURE 4.5: Case study driving scenario.

Because the eCVT vPS (the motorcycle transmission) has two inputs, τ_{en} and ω_{vd}, two actuation systems would be required if the RTHS would be conducted physically; hence two individual tracking controllers would be needed. However, without loss of generality and for the sake of simplicity, one actuation system/tracking controller is utilized as illustrated in Figure 4.2. It is applied between the eCVT vPS and the Engine NS as the

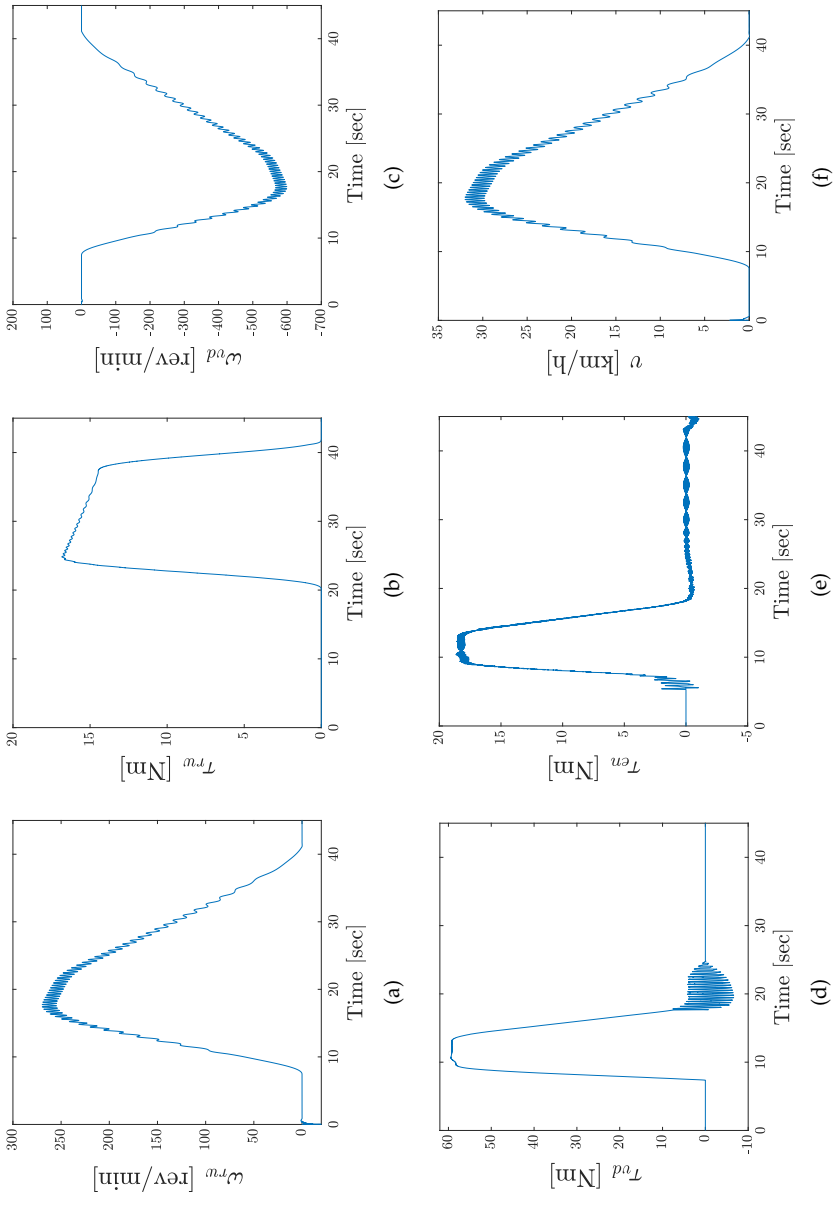


FIGURE 4.6: Indicative hybrid model responses: (a) angular velocity of the rear wheel, (b) braking torque of the rear wheel, (c) angular velocity of the transmission output shaft, (d) torque of the transmission output shaft, (e) torque of the engine, (f) motorcycle velocity.

interconnection between the two has the highest frequency content thus requiring the compensation for the RTHS to be as good as possible.

The control plant of the case study corresponds to the eCVT vPS actuation system, i.e. the inverter and the electric motor of Figure 4.2. Therefore, the control plant output y of Figure 4.1 corresponds to the τ_{en} signal and the reference r to $\tau_{en_{ref}}$. Measurements were conducted on the corresponding test bench at the testing facilities of Siemens Industry Software for system identification of the inverter and electric motor. This actuation system was identified as a second-order transfer function. Such system is represented in continuous time as follows:

$$G_m = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0}, \tag{4.27}$$

where

$$b_1 = 0,\ b_0 = a_0 = 5657,\ a_2 = 1 \text{ and } a_1 = 78.62. \tag{4.28}$$

It can also be expressed as $G_m = \omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$, with ω_n and ζ its natural frequency and damping respectively.

In a study by Silva et al. [23], a coefficient of variation of up to 5% and 7% of the control plant's natural frequency and damping respectively, were identified after conducting several HS runs. These were mainly due to physical changes of the experimental setup after repeated testing. Therefore, in order to simulate potential variations of the control plant G_m used in this case study, its natural frequency ω_n and damping ζ were selected to vary. To be conservative and expose the tracking controller to more severe conditions, ω_n and ζ vary with a coefficient of variation of 15% and 17% respectively, online during the simulation at every 0.25 second of the vRTHS. The variations are implemented by changing the values of b_0, a_0. Particularly, b_0, $a_0 \sim \mathcal{N}(5657, 1640.53)$ and b_0 equals a_0. Therefore, the goal of the online model identification algorithm, presented in Section 4.2.3 is to capture the applied variations of G_m and to correctly identify its changing dynamics and provide the AMPC with the updated prediction model. Figures 4.8a-d illustrate in blue color the applied variations of G_m over the simulation duration. The initial values of G_m are those of Eq. (4.28).

Furthermore, to assess the robustness of the proposed tracking controller under various hybrid model configurations, eleven parameters of the motorcycle virtual hybrid model are selected to vary. Table 4.1 provides an overview of these parameters. A uniform distribution is assigned to each of the chosen parameters. Their mean values and standard deviations are identified from [115, 116, 117] to reflect a range of possible parameter varia-

tions of the corresponding motorcycle components. The nominal parameter values for this case study are their mean values. Using the Latin hypercube sampling (LHS) method [104], 200 samples are obtained from each parameter and hence 200 hybrid model evaluations are performed. In contrast with a_0, b_0 parameters of G_m that are varying online, the parameters of Table 4.1 remaining constant during each vRTHS run, but their values are changed in every successive vRTHS. The stability of the hybrid model was not affected by the variation of the control plant dynamics and Table's 4.1 parameters. This can be appreciated from the results presented in Table 4.2 and Figure 4.7 of Section 4.3.3, as in each case the hybrid model response was stable.

Param.	Prob. Distrib.	Mean Value	Stand. Dev.	CV (%)	Parameter Description	Units
K_{rt}	Unif.	58570	11714	20	Vertical stiffness rear tire	$\frac{N}{m}$
Z_{rt}	Unif.	11650	3495	30	Vertical damping rear tire	$\frac{Ns}{m}$
K_{ft}	Unif.	25000	5000	20	Vertical stiffness front tire	$\frac{N}{m}$
Z_{ft}	Unif.	2134	640.2	30	Vertical damping front tire	$\frac{Ns}{m}$
K_{rs}	Unif.	125000	25000	20	Stiffness rear suspension	$\frac{N}{m}$
Z_{rs}	Unif.	10000	3000	30	Damping rear suspension	$\frac{Ns}{m}$
K_{fs}	Unif.	19000	3800	20	Stiffness front suspension	$\frac{N}{m}$
Z_{fs}	Unif.	1250	375	30	Damping front suspension	$\frac{Ns}{m}$
M	Unif.	300	6	2	Motorcycle mass	Kg
J	Unif.	0.0115	0.0023	20	Engine moment of inertia	Kgm^2
μ	Unif.	0.001	0.00005	5	Engine coefficient of viscous friction	$\frac{Nm}{rev/min}$

TABLE 4.1: Motorcycle virtual hybrid model parameters with their assigned statistical distributions.

4.3.2 *Tracking controller design properties*

The initial matrices of the prediction model ($A_p^0, B_p^0, C_p^0, D_p^0$ of Eq. (4.4)) coincide with the control plant model of Eq. (4.27) and follow:

$$A_p^0 = \begin{bmatrix} 0.96859 & -0.0348 \\ 0.02519 & 0.9996 \end{bmatrix}, B_p^0 = \begin{bmatrix} 0.0031 \\ 0 \end{bmatrix}, \tag{4.29}$$
$$C_p^0 = \begin{bmatrix} 0 & 11.0488 \end{bmatrix}, D_p^0 = \begin{bmatrix} 0 \end{bmatrix}.$$

The disturbance d and noise n models, described in Eqs. (4.2) and (4.3), respectively, are additive to the control plant output and admit:

$$\begin{aligned} x_d(k+1) &= x_d(k) + 0.0004 u_d(k) \\ d(k) &= x_d(k), \end{aligned} \tag{4.30}$$

$$n(k) = u_n(k), \tag{4.31}$$

with $A_d = 1, B_d = 0.0004, C_d = 1$ and $D_d = 0$, while $A_n = 0, B_n = 0, C_n = 0$ and $D_n = 1$. Recall that the inputs to the disturbance and noise models, i.e. u_d and u_n, respectively, follow the standard Gaussian distribution, namely $u_d, u_n \sim \mathcal{N}(0,1)$. According to Eqs. (4.9) and (4.10), the state-space formulation of the control plant dynamics for $k = 0$, including the disturbance model admits:

$$A_c = \begin{bmatrix} 0.96859 & -0.0348 & 0 \\ 0.02519 & 0.9996 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B_c = \begin{bmatrix} 0.0031 & 0 \\ 0 & 0 \\ 0 & 0.0004 \end{bmatrix}, \tag{4.32}$$
$$C_c = \begin{bmatrix} 0 & 11.0488 & 1 \end{bmatrix}, D_c = \begin{bmatrix} 0 & 0 \end{bmatrix},$$

The initial Kalman filter gain vectors follow:

$$L^0 = \begin{bmatrix} 0.157 & 0.917 & 0.398 \end{bmatrix}^T \times 1e-3, \tag{4.33}$$
$$M^0 = \begin{bmatrix} 0.195 & 0.912 & 0.398 \end{bmatrix}^T \times 1e-3.$$

The tuning weights of Eq. (4.1) are selected to be $w^y = 5.204$ and $w^u = 0.096$, while n_u, n_y equal to 1 as the control plant is a SISO model. Each control interval is obtained with a sampling frequency of 2.5 kHz and the prediction horizon is $P = 10$. The constraints of Eqs. (4.6) and (4.7) represent safety limits regarding the control plant input and admit:

$$\begin{aligned} -200 \leqslant u &\leqslant 200 \quad \left[\mathrm{Nm}\right], \\ -38 \leqslant \dot{u} &\leqslant 38 \quad \left[\frac{\mathrm{Nm}}{\mathrm{sec}}\right]. \end{aligned} \tag{4.34}$$

4.3.3 *Results*

Table 4.2 presents the performance metrics, J_1, J_2 and J_3, of the proposed tracking controller for the aforementioned simulations and Figure 4.7 displays their normalized histograms.

	Nominal	**Stochastic**	
		Mean Values	Standard Deviation
J_1 [msec]	2.5	2.5	0
J_2 [%]	0.32	0.34	0.01
J_3 [%]	2.79	2.82	0.21

TABLE 4.2: Tracking controller performance metric results.

The nominal results of Table 4.2 refer to the case when the mean values of the parameters in Table 4.1 were used and hence they correspond to a single deterministic vRTHS. The stochastic results refer to the outcomes from the 200 vRTHS evaluations. In both deterministic (nominal) and stochastic, the parameters a_0, b_0 of G_m vary as displayed in Figure 4.8.

The performance of the tracking controller is assessed by how close to zero the performance metrics are. From Table 4.2 and Figure 4.7, it can be appreciated that J_1, J_2 and J_3 are quite small and hence the proposed control scheme can adequately regulate the desired plant output even under the presence of hybrid model parameter and control plant dynamics variations, proving as well its robustness.

Figure 4.8 also presents in red color the parameters of the control plant dynamics that the online model identification algorithm estimates at each control interval. As stated in Eq. (4.24), for the first 5 seconds of the simulation there is no input (zero values) to the hybrid model and hence the output is also zero. As a result, the identification algorithm returns zero

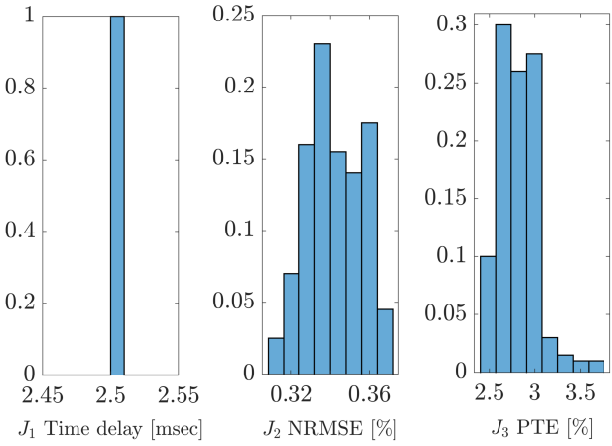


FIGURE 4.7: Normalized histograms of J_1, J_2 and J_3, obtained from 200 vRTHS evaluations.

values. It can be observed that after the first 5 seconds, the algorithm starts to respond. Recall that there were no applied variations for a_1 and b_1. However, the algorithm fails to accurately identify these two parameters (Figures 4.8b,d). Nevertheless, their deviation is relatively small. Regarding a_0 and b_0, the predictions are satisfactory as shown from Figures 4.8a,c. Figure 4.9 displays the overall identification error of the online model identification algorithm, defined as:

$$\varepsilon(t) = \frac{y(t) - y_{\text{pred}}(t)}{\max|y(t)|} \times 100 \quad [\%]. \tag{4.35}$$

From Figure 4.9, it can be stated that the overall identification error is quite small and thus the parameter identifications are acceptable. It should be noted that the performance of the online model identification algorithm is crucial for the overall performance of the AMPC. The model that the algorithm identifies is used internally in AMPC as its prediction model. Hence, faulty system identification of the control plant could yield to large tracking errors.

Figure 4.10 displays, for the nominal vRTHS case, the reference values of the control plant, namely the output of the engine NS $\tau_{\text{en}_{\text{ref}}}$, and the control plant output, namely the input to the vPS τ_{en}. An ideal tracking controller should be able to compensate for the actuation system dynamics so that $\tau_{\text{en}_{\text{ref}}}$ and τ_{en} are as close as possible. As shown from Figure 4.10, the tracking error between the two signals is very small indeed.

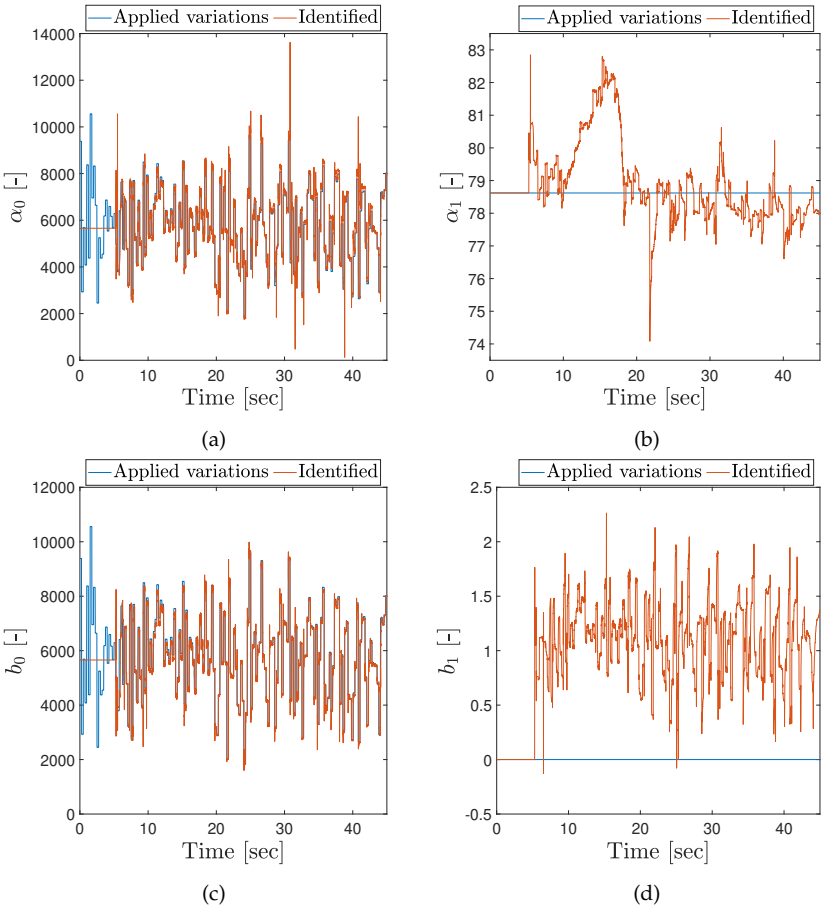


FIGURE 4.8: Applied variations in the parameters of the control plant and the identified values from the online model identification algorithm: (a)a_0, (b)a_1, (c)b_0, (d)b_1.

Based on the above results, the proposed tracking controller can provide the desired performance and at the same time be robust to hybrid model and actuation system variations, as well as to the introduced disturbances and measurement noise. Additionally, using the proposed tracking controller, the tracking errors and time delays that are introduced due to the inherent dynamics of the actuation system used, are satisfactorily compensated, enabling thus RTHS outcomes of high fidelity. As a result, the proposed control scheme is demonstrated to be suitable for RTHS.

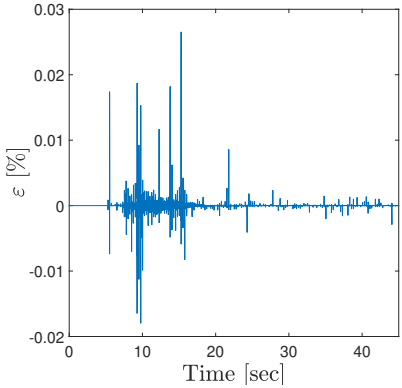


FIGURE 4.9: Control plant system identification error.

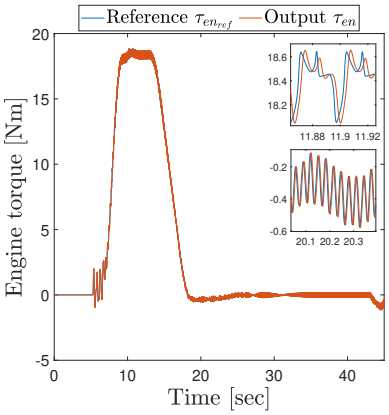


FIGURE 4.10: Engine torque responses: reference $\tau_{en_{ref}}$ and measured output τ_{en}.

4.4 Conclusions

In this chapter, a novel tracking controller for dynamics compensation in RTHS was proposed. The proposed control scheme is an extension of the MPC presented in Chapter 3 to include an adaptive behavior. It is based on AMPC, a linear time-varying Kalman filter and an online model identification algorithm. For the latter, ARX polynomial models are identified online to estimate the changing control plant dynamics and used to update the prediction model of the MPC. Such a control strategy is desirable for control plants with high nonlinear or linear with time-varying dynamics. A parametric vRTHS case study is used to validate the performance and robustness of the proposed control scheme. In particular, the control plant dynamics were varying during the simulation in the presence of disturbance and measurement noise, additive to its output. Additionally, parameters of the hybrid model were chosen and assigned with random values via prescribed probability distributions. A total of 200 samples were generated and 200 hybrid simulation evaluations were performed to assess whether the proposed control scheme is robust and preserves the desired performance. Results indicate that the controller can guarantee small tracking errors under uncertainties that may be present during the hybrid simulation. As a result, the effectiveness of the proposed control methodology was demonstrated proving that the AMPC is a suitable approach for RTHS.

Chapter 5

Model order reduction framework in real-time hybrid simulation

The material presented in this chapter is submitted for publication and is currently under review. A preprint can be found in [118].

5.1 Introduction

This chapter continues to address the first research objective but this time the focus lies on challenges arising from the NS of the hybrid model. Recall that hybrid models with high-dimensional nonlinear NS risk distorting the time scale of NS due to the increased necessary computational power. In particular, in order to capture the dynamic behavior of interest of a high-dimensional nonlinear numerical model, it is often necessary to use small time-steps for the numerical solver. However, the smaller the time-step of the simulation, the larger the computational power needed to compute it. In RTHS, when the required computation time becomes larger than the actual simulation time, time delays are introduced to the hybrid model risking not only to decrease the fidelity of the HS results but also to drive the overall HS into instability. Therefore, in such cases, the only viable solution is to reduce the order of the NS to be able to conduct the HS in real-time (challenge 2.b from Section 1.2.2).

Several model order reduction (MOR) techniques can be found in the literature. Proper orthogonal decomposition (POD) [119, 120, 121], also known as principal component analysis (PCA), and its discrete version, namely the singular value decomposition (SVD), are mostly combined with the method of snapshots [122, 123] and have been extensively used for extraction of mode shapes from high-dimensional systems, among other techniques. A technique similar to POD, also using the method of snapshots but following a data-based instead of a model-based approach, is the dynamic mode decomposition (DMD) [124]. In particular, the main difference between POD and DMD is that the former relies upon a time-averaged spatial correlation matrix while the latter represents the temporal dynamics with high-degree polynomials [125]. An advancement of POD was also proposed based on the use of separated representations, the so-called proper generalized decomposition (PGD) [126]. More recently, the

component-mode synthesis (CMS) approach [127, 128] has been used for non classically damped linear systems [129] as well as in HS as a method to reduce the order of the comprising NS and PS [130]. Furthermore, a quadratic manifold [131] and a non-intrusive [132] approach were also proposed for MOR of nonlinear structural systems. MOR techniques have also been used for Bayesian finite element model updating [133] and for reliability-based design problems [134].

In this chapter, a MOR framework originating from data-driven regression is proposed to reduce the order of high-dimensional nonlinear NS in HS. Two different methodologies for MOR are addressed; a polynomial chaos expansion (PCE) and a feedforward neural network (FFNN). In both cases, the NS, whose order we want to reduce, is treated as a *black box*, meaning that no prior knowledge of its full-order model (FOM) dynamics is required. Instead, the outputs of the respective NS are mapped to its inputs by simpler-to-evaluate functions than those of the original FOM. The goal of the MOR framework is to adequately capture the dynamic behavior of high-dimensional NS at a much lower computational cost and in shorter computation time, and thus enable RTHS. A parametric case study encompassing a virtual hybrid model is employed to validate the proposed MOR framework. Selected NS are replaced with their respective FFNN- and PCE-based reduced-order models while specific parameters of the hybrid model are varied to investigate the framework's robustness to different hybrid model configurations. The corresponding reduced-order hybrid model responses are compared to the ones from the FOM. Results demonstrate the effectiveness of the proposed MOR framework.

The chapter is organized as follows. Section 5.2 introduces the MOR framework describing the PCE and FFNN. Section 5.3 presents the case study with the virtual hybrid model used to validate the framework along with the obtained results. Section 5.4 presents the overall conclusions of this chapter.

5.2 Outline of the model order reduction framework

Considering an input vector $\mathbf{X} \in \mathcal{D}_{\mathbf{X}} \subset \mathbb{R}^N$ and a computational model $\mathbf{Y} = \mathcal{M}(\mathbf{X})$ with $\mathbf{Y} \in \mathbb{R}^N$, data-driven regression algorithms formulate a map $\mathcal{M}^s : \mathbf{X} \mapsto \mathbf{Y}$ based on an obtained sample set of input points $\mathbf{X} = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right\}^T$ and of the respective output values, i.e. model evaluations, $\mathbf{Y} = \left\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}\right\}^T$. The set of $\mathbf{X}, \mathbf{Y}$ realizations corresponds to the so-called experimental design (ED). Regression techniques encompass linear regression, kernel methods, neural networks and graphical models

among others. An overview of these techniques can be found in [135, 136]. As mentioned above, the proposed MOR framework of this chapter explores use of PCE and FFNN to this end, with these methods elaborated in what follows.

5.2.1 *Polynomial chaos expansion*

PCE is a well-known uncertainty quantification spectral method used to substitute the dynamics of an expensive-to-compute numerical model with an inexpensive-to-compute surrogate (a.k.a. metamodel), representing the outputs of the model by a polynomial function of its inputs [137, 138]. It is proven to be a powerful surrogate technique used in a wide variety of engineering contexts, to replicate the dynamic response of complex high-dimensional models [65, 139, 140]. Hence, it can be seen as a promising technique for MOR of NS in HS.

In more detail, given a random input vector $\mathbf{X}$ with independent components expressed by the joint PDF $f_{\mathbf{X}}$ and a finite variance computational model $Y = \mathcal{M}(\mathbf{X})$, such that $\mathbb{E}[Y^2] = \int_{\mathcal{D}_{\mathbf{X}}} \mathcal{M}^2(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x} < \infty$, the PCE of $\mathcal{M}(\mathbf{X})$ follows:

$$Y = \mathcal{M}(\mathbf{X}) \approx \mathcal{M}^{PCE}(\mathbf{X}) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^N} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\mathbf{X}). \tag{5.1}$$

The PCE function is built on the $\Psi_{\boldsymbol{\alpha}}(\mathbf{X})$ multivariate orthonormal polynomial basis with respect to the input vector $f_{\mathbf{X}}$. The degree of the $\Psi_{\boldsymbol{\alpha}}$ polynomials components is identified by the $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N), \boldsymbol{\alpha} \in \mathbb{N}^N$ multi-index for each of the input variables, while $y_{\boldsymbol{\alpha}}$ corresponds to the polynomial coefficients.

5.2.1.1 *Polynomial basis*

The multivariate polynomials are constructed as a tensor product of their univariate orthonormal polynomials $\phi_k^{(i)}(x^{(i)})$, such that:

$$\Psi_{\boldsymbol{\alpha}}(\mathbf{X}) = \prod_{i=1}^{N} \phi_{\alpha_i}^{(i)}(x^{(i)}). \tag{5.2}$$

The latter meets the orthonormality criteria:

$$\left\langle \phi_j^{(i)}(x^{(i)}), \phi_k^{(i)}(x^{(i)}) \right\rangle = \int_{\mathcal{D}_{X_i}} \phi_j^{(i)}(x^{(i)}) \phi_k^{(i)}(x^{(i)}) f_{X_i}(x^{(i)})\, dx_i = \delta_{jk}, \tag{5.3}$$

where i corresponds to the input variable, j and k to the polynomial degree, $f_{X_i}(x^{(i)})$ to the i^{th}-input marginal PDF and δ_{jk} to the Kronecker symbol. The selection of the univariate orthonormal polynomial families depends on the marginal PDF of each input variable to which they are orthogonal, e.g. if an input variable follows the uniform/ Gaussian distribution then the Legendre/ Hermite orthogonal polynomial family is used respectively for this specific input variable [137, 141].

5.2.1.2 *Truncation schemes*

Once the univariate polynomials families are selected for each input variable, the next step is the construction of the PCE following Eq. (5.1). However, because the sum of Eq. (5.1) consists of infinite terms, it is often truncated to a finite number of terms for practical reasons. Hence, the truncated basis is defined as $\mathcal{A} \subset \mathbb{N}^N$ and the PCE of $\mathcal{M}(\mathbf{X})$ admits:

$$\mathcal{M}^{PCE}(\mathbf{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\mathbf{X}). \tag{5.4}$$

The performance of PCE is closely connected with the truncation scheme used. Underfitting or overfitting is possible to occur when too many terms are discarded or introduced respectively [139]. In the standard basis truncation scheme [137] the maximum degree $p \in \mathbb{N}^+$ is defined, such that the degree of each polynomial is capped by this value. Therefore, the standard basis truncation scheme consists of $\binom{N+p}{p} = \frac{(N+p)!}{N!\,p!}$ elements and follows:

$$\mathcal{A}^{N,p} = \left\{ \boldsymbol{\alpha} \in \mathbb{N}^N : |\boldsymbol{\alpha}| \leqslant p \right\}, \tag{5.5}$$

where N corresponds to the number of input variables of $\mathcal{M}(\mathbf{X})$ and $|\boldsymbol{\alpha}| = \sum_{i=1}^{N} \alpha_i$ to the total degree of all polynomials in $\boldsymbol{\Psi}$. To further reduce the polynomial basis size, additional truncation schemes have been developed, namely the maximum interaction and hyperbolic truncation scheme.

In the maximum interaction truncation scheme [141], the basis of Eq. (5.5) is reduced such that the $\boldsymbol{\alpha}$ indices to include at most r non-zero components and thus the rank of $\boldsymbol{\alpha}$ is decreased [141]. Accordingly, Eq. (5.5) is written as:

$$\mathcal{A}^{N,p,r} = \left\{ \boldsymbol{\alpha} \in \mathcal{A}^{N,p} : \|\boldsymbol{\alpha}\|_0 \leqslant r \right\}, \tag{5.6}$$

where $\|\boldsymbol{\alpha}\|_0 = \sum_{i=1}^{N} \mathbb{1}_{\{\alpha_i>0\}}$ and $\mathbb{1}_{\{\alpha_i>0\}}$ is the indicator function.

The hyperbolic (a.k.a. q-norm) truncation scheme [142] reforms the basis of Eq. (5.5) such that:

$$\mathcal{A}^{N,p,q} = \left\{\boldsymbol{\alpha} \in \mathcal{A}^{N,p} : \|\boldsymbol{\alpha}\|_q \leqslant p\right\}, \tag{5.7}$$

where $\|\boldsymbol{\alpha}\|_q = \left(\sum_{i=1}^{N} a_i^q\right)^{\frac{1}{q}}$ and $q \in (0, 1]$.

5.2.1.3 *PCE coefficient calculation*

Once the truncation scheme is determined, the next step is the calculation of coefficients $\mathbf{y} = \{\mathbf{y}_{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathcal{A}\}$ of Eq. (5.4). Various methods have been developed to do this [141]. One of these techniques is the least-squares minimization [143]. This is an non-intrusive method, meaning that the coefficients are obtained after post-processing the ED points. More specifically, Eq. (5.4) is reformed as:

$$\mathcal{M}^{PCE}(\mathbf{X}) = \sum_{j=0}^{P-1} y_j \Psi_j(\mathbf{X}) + \varepsilon_P = \mathbf{y}^T \boldsymbol{\Psi}(\mathbf{X}) + \varepsilon_P, \tag{5.8}$$

where $\mathbf{y} = (y_0, \ldots, y_{P-1})^T$ denote the PCE coefficients, ε_P the truncation error, $\boldsymbol{\Psi}(\mathbf{X}) = \{\Psi_0(\mathbf{X}), \ldots, \Psi_{P-1}(\mathbf{X})\}^T$ the multivariate orthonormal polynomials, and $P = \binom{N+p}{p}$. Then, the PCE coefficients are obtained by solving:

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y} \in \mathbb{R}^{|\mathcal{A}|}} \mathbb{E}\left[\left(\mathbf{y}^T \boldsymbol{\Psi}(\mathbf{X}) - \mathcal{M}(\mathbf{X})\right)^2\right]. \tag{5.9}$$

The solution of Eq. (5.9) is attained by Ordinary Least-Squares (OLS) and follows:

$$\hat{\mathbf{y}} = \left(\mathbf{A}^T \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{Y}, \tag{5.10}$$

where the experimental matrix is $A_{ij} = \Psi_j(\mathbf{x}^{(i)})$ with $i = 1, \ldots, N$ and $j = 0, \ldots, P-1$.

However, for $N < P-1$ the $\mathbf{A}^T\mathbf{A}$ matrix is non-invertible and thus the OLS problem cannot be solved [139, 144]. Nonetheless, through sparse regression accurate surrogates can be constructed with fewer coefficients. A way to achieve sparse regression in high-dimensions is by a modification of the least-squares minimization method. In more detail, a penalty term is introduced in Eq. (5.9) and the latter is modified as:

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y} \in \mathbb{R}^{|\mathcal{A}|}} \mathbb{E}\left[\left(\mathbf{y}^{\mathrm{T}}\boldsymbol{\Psi}(\mathbf{X}) - \mathcal{M}(\mathbf{X})\right)^2\right] + \lambda\|\mathbf{y}\|_1, \tag{5.11}$$

where $\|\mathbf{y}\|_1 = \sum_{\boldsymbol{\alpha}\in\mathcal{A}}|y_{\boldsymbol{\alpha}}|$ and λ a parameter of the penalty term. This penalty term causes the minimization to facilitate sparse solutions. One of the widely utilized techniques, used to solve the minimization problem of Eq. (5.11) is the Least Angle Regression (LAR) algorithm [142].

5.2.2 *Feedforward neural networks*

A widely used group of versatile methods employed both for regression and classifications problems, are the neural networks. One frequently employed variety of such is the so-called fully connected FFNN. In particular, FFNNs comprise layers of neurons and in each layer, the neuron's vector of values is computed by multiplying a weight matrix with the vector of values from its previous layer, followed by applying an element-wise nonlinear activation function. FFNNs have been well-established as capable emulators of a broad variety of application functions and can be recognized, in the boundaries of infinite neurons, as universal approximators [145, 146]. Therefore, FFNNs can be clearly considered as a metamodeling technique as it has been shown in various contexts [147, 148, 149]. A feature of FFNNs is their capability to be scaled to large and high dimensional datasets and particularly taking into account deep learning methods, they are proven powerful to be able to deal with highly nonlinear dynamic relations [150, 151]. As such, FFNNs are a promising metamodeling alternative in a hybrid simulation context, although it has only been demonstrated in limited cases to date [152, 153].

5.2.2.1 *Network Architecture*

A reason that makes neural networks such powerful, is their ease to adapt their architecture to systems of various sizes and complexity [154]. Figure 5.1 illustrates the basic architecture of a 3 layer FFNN. In the first layer, i.e. input layer, the neurons admit the values of the input vector $\mathbf{X}$. The dimensionality of the latter equals the number of neurons in the input layer. The output layer is the last layer of the network, in which the neuron values correspond to the elements of the output vector $\boldsymbol{Y}$. As before, the dimensionality of the output vector equals the number of neurons in the output layer. A hidden layer lies in between the input and output layer and consists of an arbitrary number of neurons, the choice of which can significantly influence the network's performance. The neurons of the hidden layer admit the activation values, represented by the vector $\boldsymbol{A}$. Multiple hidden layers

would result in the so-called deep neural networks. Although the latter are deemed as even more powerful to emulate highly complex functions [155], in the scope of this work only a single hidden layer is utilized as it is considered sufficient.

Eqs. (5.12) and (5.13) represent mathematically the architecture shown in Figure 5.1. The activation values in the hidden layer of the FFNN are calculated following Eq. (5.12). In detail, the input vector $\mathbf{X}$ is matrix multiplied with the weight matrix $\mathbf{W^1}$ and the bias vector $\mathbf{b^1}$ is then added. Afterwards, the activation vector $\mathbf{A}$ is computed by applying the element-wise activation function g^1. The FFNN output vector $\mathbf{Y}$ is similarly calculated following Eq. (5.13). The activation vector $\mathbf{A}$ is matrix multiplied with a second weight matrix $\mathbf{W^2}$ and a second bias vector $\mathbf{b^2}$ is added before applying the element-wise second activation function g^2. In the aforementioned equations, the parameters of the weight matrices $\mathbf{W^1}$ and $\mathbf{W^2}$, as well as the parameters of the bias vectors $\mathbf{b^1}$ and $\mathbf{b^2}$, are identified, i.e. learned, during the training of the network. On the contrary, the activation functions g^1 and g^2 are hyperparameters, selected by the network designer. Typical candidates include the $\tanh$, RELU and linear activation functions, each with different associated properties [5].

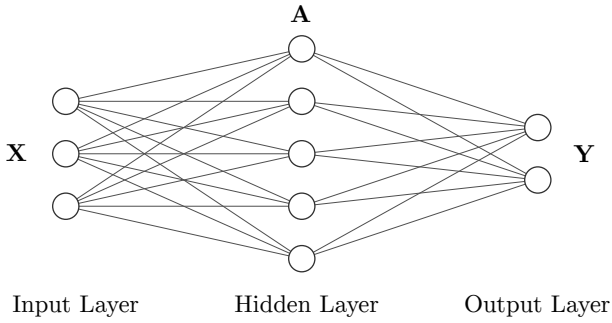


FIGURE 5.1: Architecture of a 3 layer FFNN.

$$\mathbf{A} = g^1(\mathbf{W^1}\mathbf{X} + \mathbf{b^1}), \tag{5.12}$$

$$\mathbf{Y} = g^2(\mathbf{W^2}\mathbf{A} + \mathbf{b^2}). \tag{5.13}$$

5.2.2.2 *Network Training*

The first step to create a FFNN metamodel consists of generating a dataset that will include the input and output pairs of vectors for the system being emulated. This dataset is used to train, validate and test the network and

hence the dataset is further divided into training, validation and testing datasets. The training dataset includes the input-output pairs used for the optimization of the learned parameters of the network, namely the weight matrices and the bias vectors. The validation dataset consists of a set of held back points that are employed to compare networks developed with differing hyperparameters, e.g. hidden layer size or activation function. Such a method is useful since it enables comparison of the network generalization, generally to a more complex and thus flexible network, and hence yield larger performance on a training dataset. Utilization of separate validation datasets can evidence whether the obtained performance can transfer to points outside of the training dataset or whether the network is overfitting to the training points [154]. Finally, the testing dataset corresponds to those pairs in which the final trained network of the selected architecture is tested.

For this work, the MATLAB deep learning toolbox [156] was used to train the neural network. Because in the context of metamodeling the problem is one of regression, i.e. prediction of a continuous output variable, the mean squared error value was chosen as the loss function of the network to be minimized and it is described by Eq. (5.14). In the latter equation, the loss value J is computed as the mean value of all the N training data points of the squared error between the measured (true) output value Y_i for the training point i and the predicted output value from the FFNN $\hat{Y}_i$. For the network weight training, the Bayesian regularization algorithm was used. The latter is considered to be more resistant to overfitting and tends to train networks with greater generalization [157]. A validation dataset was utilized to nominate the final architecture of the FFNN as the one with the better validation performance. Therefore, the final network architecture chosen for this study was a 3 layer network with a hidden layer size of 10 neurons. The activation function at the hidden layer was selected as a $tanh$ function and for the output layer a $linear$ function.

$$J = \frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2 \tag{5.14}$$

5.3 Case study: virtual RTHS of a prototype motorcycle

Given that the focus of this work is a MOR framework for RTHS, the case study employs a virtual hybrid model to conduct virtual RTHSs (vRTHSs). All of the components of the virtual hybrid model are implemented numerically. Thus, virtual PSs (vPSs) are used in place of physical specimens. In this regard, without loss of generality and for the sake of clarity, transfer

systems that would be required between NS and PS in a physical HS are omitted from this case study.

5.3.1 *Problem formulation*

The virtual hybrid model used in this case study represents a prototype motorcycle. The hybrid model is the same as the one presented in the case study of Chapter 4. However, since no transfer system is implemented in this case study, the hybrid model is simplified. Therefore, for clarity and for the ease of reading, it is repeated here as well. It consists of one vPS, the electronically-controlled continuously variable transmission (eCVT) of the motorcycle, and four NS: i) the engine, ii) the motorcycle body dynamics, iii) the rear wheel braking system and iv) the front wheel braking system. The interconnections between NS and the vPS are illustrated in Figure 5.2.

The eCVT vPS corresponds to a MIMO model with two sets of one input / one output. The first set is connected to the motorcycle engine NS and the second one to the motorcycle body NS. The latter connection corresponds to the transmission output shaft of the motorcycle. The engine NS simulates the dynamics of the combustion engine. It is represented by a MISO model, with its inputs being the throttle percentage thr and the angular velocity of the engine ω_{en} and its single output being the torque of the engine τ_{en}. The motorcycle body NS represents the body or chassis dynamics of the motorcycle together with the dynamics of the wheels, tires and suspensions along with the road profile and the environmental driving conditions. It is represented by a MIMO model with 3 sets of one input / one output. The first set is connected to the eCVT vPS with the torque τ_{vd} as input and the angular velocity ω_{vd} of the transmission output shaft as output. The second and third sets are linked to the rear and front wheel braking system NS respectively. Both braking systems are MISO models. The rear wheel braking system NS inputs are the angular velocity of the rear wheel ω_{rw} and the applied force on the brake pedal $F_{br_{rw}}$, while its single output is the braking torque of the rear wheel τ_{rw}. Respectively, the front wheel braking system NS inputs are the angular velocity of the front wheel ω_{fw} and the applied force on the brake lever $F_{br_{fw}}$ and its single output is the braking torque in the front wheel τ_{fw}. Recall that the development of the substructures of the hybrid model was done in the multi-physics simulation software *Simcenter Amesim*. The report of [114] offers a thorough description of the utilized substructures along with the equations governing their motion.

The performance of the motorcycle prototype is examined by testing its virtual hybrid model under predefined driving, road and wind scenarios.

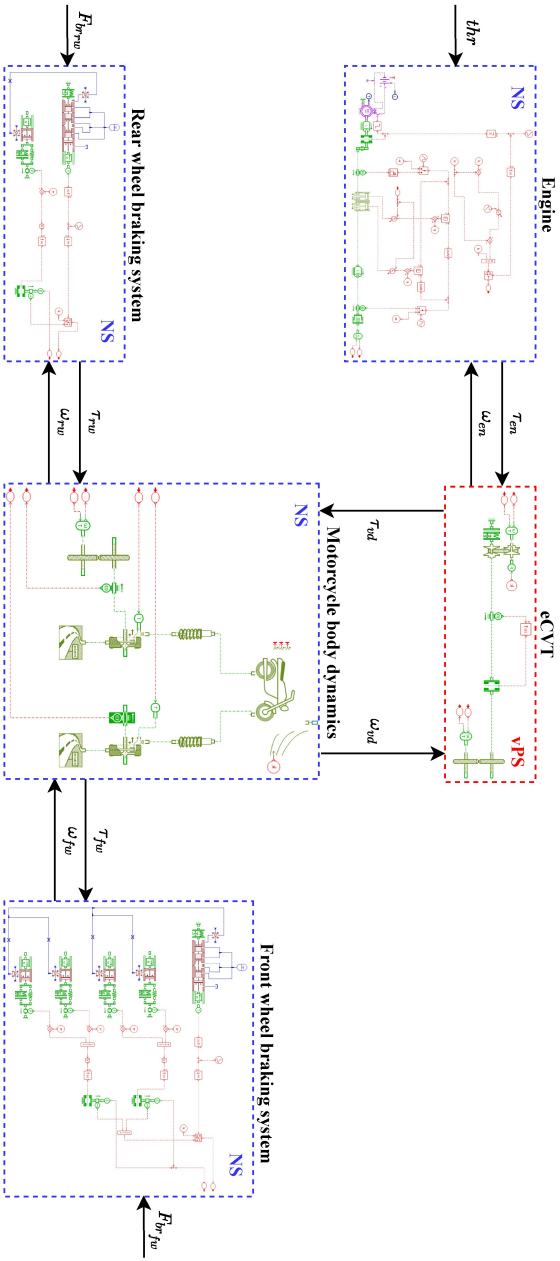


FIGURE 5.2: Motorcycle hybrid model block diagram.

These tests are, in fact, vRTHSs. A Simcenter real-time simulation platform is used for data exchange between substructures and for running the vRTHSs.

The case study involves a 45 sec long motorcycle driving scenario on a road with a defined profile and in defined wind conditions. The driving scenario is described by the variation of the throttle thr and the braking forces $F_{br_{rw}} = F_{br_{fw}}$ (front and rear wheel braking forces are assumed to be equal) described by Eq. (5.15) and Eq. (5.16), respectively, and shown in Figure 5.3. Note that the braking forces are measured in Newton and the maximum applied throttle is 0.5 (50%), corresponding to a half-open throttle. The road profile, i.e. height variation of the road, is defined by the sinusoidal function $h(x)$ in Eq. (5.17), where x denotes the current position of the motorcycle in meters. The ambient wind velocity is assumed to be zero.

Each case study motorcycle driving test involves "driving" the motorcycle virtual hybrid model following the predefined driving scenario along the given road in the given wind conditions. The position of the motorcycle is computed by solving its equation of motion. In this case study, the RK4 (fourth-order Runge–Kutta) method with a fixed time-step of 0.1 msec was used to numerically solve this equation of motion. In addition to the position of the motorcycle, a number of indicative performance parameters are computed and monitored. These include global response parameters, such as the motorcycle velocity v, as well as the input and output parameters of the NSs, i.e. the angular velocities and torques shown in Figure 5.2.

$$thr(t) = \begin{cases} 0.125t & , 5 \leqslant t < 9 \\ 0.5 & , 9 \leqslant t \leqslant 13 \\ -0.125t & , 13 < t \leqslant 17 \\ 0 & , \text{elsewhere} \end{cases} \tag{5.15}$$

$$F_{br_{rw}}(t) = \begin{cases} 10t & , 20 \leqslant t < 25 \\ 50 & , 25 \leqslant t \leqslant 32 \\ -10t & , 32 < t \leqslant 37 \\ 0 & , \text{elsewhere} \end{cases} \tag{5.16}$$

$$h(x) = \begin{cases} 0 & , 0 \leqslant x \leqslant 2 \\ 0.02\Big(\cos\left(\frac{2\pi x}{3}\right) - 1\Big) & , \text{elsewhere} \end{cases} \tag{5.17}$$

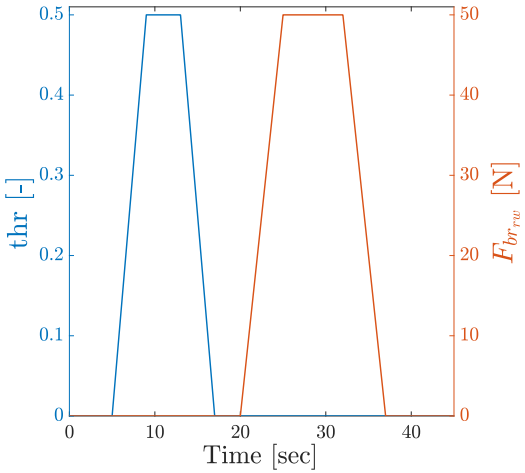


FIGURE 5.3: Case study driving scenario.

5.3.2 *Reduced-order numerical substructures*

To demonstrate the proposed MOR framework, two of the four NS in the virtual hybrid model of the motorcycle, specifically the rear and front wheel braking systems, were replaced with their respective reduced-order models, applying the PCE and FFNN techniques addressed in Section 5.2. The braking systems were selected because they include the brake hydraulic systems and friction pads. Adequately emulating the dynamic behavior of such systems using numerical models requires small integration time-steps, making them ideal candidates for MOR. The reduced-order models of the two braking system NSs will be referred to as reduced-order NS (RO-NS) hereafter.

To train the front and rear brake system PCE and FFNN RO-NSs, two sets of 200 points were generated for the input variables listed in Table 5.1, assuming the inputs are statistically independent, using the Latin hypercube

sampling (LHS) method [104]. For each of the 200 generated points, full-order models (FOMs) of the front and rear braking systems were used to (accurately but expensively) compute the resulting front and rear braking torques τ_{rw} and τ_{fw}, respectively. The obtained experimental designs (EDs) for the rear and the front braking systems are:

$$\begin{aligned} \mathbf{X}_{rw} &= \left\{F_{br_{rw}}, \omega_{rw}\right\}^T \\ \mathbf{Y}_{rw} &= \{\tau_{rw}\}, \end{aligned} \tag{5.18}$$

$$\begin{aligned} \mathbf{X}_{fw} &= \left\{F_{br_{fw}}, \omega_{fw}\right\}^T \\ \mathbf{Y}_{fw} &= \{\tau_{fw}\}. \end{aligned} \tag{5.19}$$

Two types of RO-NSs were developed for each braking system, one featuring a single PCE and the other being a FFNN. The same ED was used to train, validate and test the PCE and the FFNN RO-NSs for each braking system, as discussed in Section 5.2.

Input	Probability Distribution	Mean Value	Standard Deviation	Units
$F_{br_{rw}}$	Uniform	25	14.43	N
ω_{rw}	Uniform	235	135.68	rpm
$F_{br_{fw}}$	Uniform	25	14.43	N
ω_{fw}	Uniform	245	141.45	rpm

TABLE 5.1: Marginal PDFs and their characteristics per input variable of the front and rear brake system RO-NSs. Used to construct the EDs of Eq. (5.18) and Eq. (5.19).

5.3.3 *Virtual hybrid models*

Four different virtual hybrid models of the motorcycle were constructed in the case study, differing in the selection of braking system NSs. The base-line virtual hybrid model features the FOM-NSs of both the front and the rear braking systems. The remaining three virtual hybrid models have RO-NSs. In the PCE virtual hybrid model, both braking systems are represented using PCE RO-NSs. Analogously, the FFNN virtual hybrid model uses FFNN RO-NSs. Finally, the mixed virtual hybrid model of the

motorcycle uses the FFNN RO-NS for the front brakes and the PCE RO-NSs for the rear brakes.

The performance of the thus generated motorcycle virtual hybrid models was tested statistically. To this end, the 11 parameters of the motorcycle model, listed in Table 5.2, were assigned uniform distributions with parameters identified from [115, 116, 117] to reflect a range of possible parameter variations of the corresponding motorcycle components. A total of 30 different samples of the 11 motorcycle parameters were generated using the LHS method, essentially resulting in 30 different motorcycle prototypes. These prototypes were modeled using four virtual hybrid models described above. Using the driving scenario described above, a total of 120 motorcycle model response time histories were computed in this case study.

The variability of motorcycle prototype response in the selected driving scenario, computed using the (accurate but expensive) FOM virtual hybrid model, is demonstrated in Figures 5.4. The 30 time histories of two indicative virtual hybrid model response variables, the front wheel brake torque τ_{fw} and the motorcycle velocity $v = \dot{x}$, are compared to their mean value. All results presented hereafter correspond to the mean values of the 30 virtual hybrid model response simulations.

Additionally, the time history responses of Figures 5.4 can be intuitively interpreted. Figure 5.4a depicts the time history of the front wheel brake torque τ_{fw}, generated by the employed driving scenario. As can be appreciated from Figure 5.3 and Eq. (5.16), the motorcycle brakes are activated between the 20^{th} and 37^{th} second of the simulation and therefore the braking torque is non-zero only inside this time interval. Figure 5.4b illustrates the evolution of the motorcycle's velocity response v using the aforementioned driving scenario. According to Figure 5.3 and Eq. (5.15), for the first five seconds of the simulation, no throttle is applied to the motorcycle and hence its velocity is zero. Between the 5^{th} and 17^{th} second of the simulation, the driving scenario involves stepping on the throttle and therefore the motorcycle starts to accelerate. As a result, its velocity is increasing. Following Eq. (5.16), as mentioned before, after the 20^{th} second, the driving scenario involves pressing on the brakes and thus the motorcycle's velocity begins to decrease and it is brought to almost a full stop in the 45 seconds of the simulation.

5.3.4 *Results*

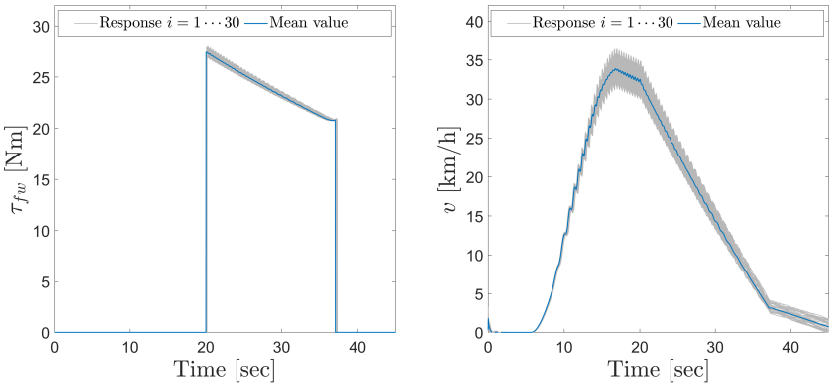
The performance of the motorcycle prototype was evaluated in tests involving a 45 sec long driving scenario. RTHS lasted approximately 42 sec when the FOM virtual hybrid model was used, while they took approximately

Param.	Prob. Distrib.	Mean Value	Stand. Dev.	CV (%)	Parameter Description	Units
K_{rt}	Unif.	58570	11714	20	Vertical stiffness rear tire	$\frac{N}{m}$
Z_{rt}	Unif.	11650	3495	30	Vertical damping rear tire	$\frac{Ns}{m}$
K_{ft}	Unif.	25000	5000	20	Vertical stiffness front tire	$\frac{N}{m}$
Z_{ft}	Unif.	2134	640.2	30	Vertical damping front tire	$\frac{Ns}{m}$
K_{rs}	Unif.	125000	25000	20	Stiffness rear suspension	$\frac{N}{m}$
Z_{rs}	Unif.	10000	3000	30	Damping rear suspension	$\frac{Ns}{m}$
K_{fs}	Unif.	19000	3800	20	Stiffness front suspension	$\frac{N}{m}$
Z_{fs}	Unif.	1250	375	30	Damping front suspension	$\frac{Ns}{m}$
M	Unif.	300	6	2	Motorcycle mass	Kg
J	Unif.	0.0115	0.0023	20	Engine moment of inertia	Kgm^2
μ	Unif.	0.001	0.00005	5	Engine coefficient of viscous friction	$\frac{Nm}{rev/min}$

TABLE 5.2: Motorcycle virtual hybrid model parameters with their assigned statistical distributions.

19 sec, 20 sec and 21 sec for the FFNN, PCE and mixed (FFNN&PCE) virtual hybrid models, respectively. Thus, by reducing the order of two braking system NSs, the computation time was halved. This is notable because the computation time of the FOM virtual hybrid model is close to the duration of the driving scenario: it is conceivable that if the driving scenario was more complex, the FOM virtual hybrid model would not complete its calculations in real-time, risking time delays and distortion of the HS timescale. Therefore, reducing the order of the critical NSs using the techniques discussed in Section 5.2 is justified.

To further assess the performance of the MOR framework, the normalized root mean square error (NRMSE) is used to compare the selected hybrid model dynamic response parameters. Note that the compared responses



(a) Front wheel braking torque.

(b) Motorcycle velocity.

FIGURE 5.4: Thirty FOM virtual hybrid model responses time histories and their respective mean values for the (a) front wheel braking torque and the (b) motorcycle velocity.

were computed from the four sets of 30 driving scenario simulations using four virtual hybrid models of the motorcycle prototype described above.

First, the RO-NSs of the front and rear braking systems are compared to the full-order model (FOM) prototype. Recall that the same EDs (Eqs. (5.18) and (5.19)), comprising 200 input-output data points generated by the FOM of the front and rear braking systems, were used to train the RO-NSs. To assess the RO-NSs performance, the quantities that characterize the braking systems, namely the rear τ_{rw} and front wheel braking torque τ_{fw}, are compared to the same outputs of the FOM-NSs. Table 5.3 displays the NRMSE between FOM-NSs and RO-NSs for τ_{rw} and τ_{fw}. The NRMSE is small, less than 3% for the real wheel torque and less than 7% for the front wheel, without a clear advantage of the FFNN or the PCE RO-NSs. The NRMSE error difference likely originates with the motorcycle dynamics. Figures 5.5a-b present the mean time history response of the braking torques obtained from the FOM, FFNN and PCE virtual hybrid models. The comparison indicates that the responses from the reduced-order virtual hybrid models are similar and only slightly different from the response of the FOM virtual hybrid model.

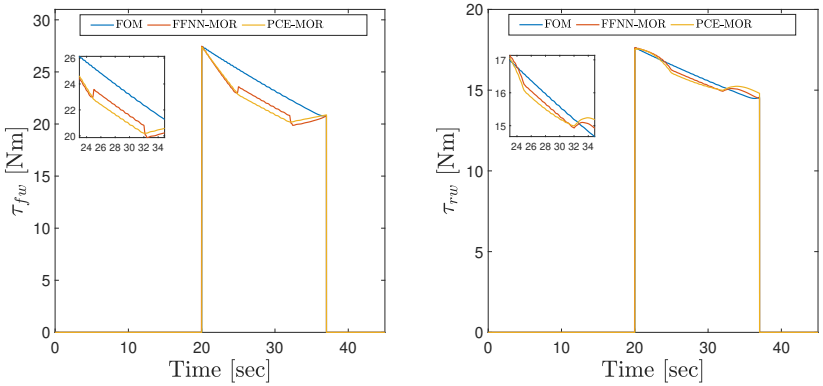
Second, the response of the four virtual hybrid models is compared by examining the NRMSEs and the time histories of the indicative model response quantities, namely the angular velocities and torques that couple the substructures of the hybrid models (ω_{rw}, ω_{fw}, ω_{en}, ω_{vd}, τ_{en} and τ_{vd})

and the motorcycle velocity v. The variables that couple the substructures of the hybrid models were investigated since deviations from their nominal values, would risk the HS outcomes and could drive the HS loop into instability. In addition, for non-virtual RTHS that include real test benches and PS, large deviations could damage the corresponding physical specimen and laboratory equipment. On the contrary, the motorcycle velocity v is chosen to be examined as it corresponds to a global response quantity that informs about the overall dynamic behavior of the motorcycle during the driving scenario. Table 5.4 provides an overview of the NRMSE and Figures 5.6, 5.7 show the response quantity time histories for the four virtual hybrid models. Evidently, the difference between the virtual hybrid models is small: the response time history graphs virtually overlap, while the NRMSEs are all smaller than 2%. Therefore, both the PCE and the FFNN MOR techniques described in Section 5.2 produce valid and efficient reduced-order models for RTHS. It is also notable that a combination of RO-NSs developed using different MOR approaches also works, giving a much-needed ability to combine reduced-order models in practice.

Finally, given that the performance of PCE-based and FFNN-based reduced-order models is practically the same, model designers need to make their modeling choice based on additional consideration. High-dimensional models, with long input and output vectors, are more suitable for FFNN-based MOR, since PCEs may require utilization of sparse schemes and thus incur additional errors. On the other hand, PCE-based MOR offers a distinct advantage in that the PCE coefficients can be used to compute the statistics of the relevant model response quantities, as well as assess their sensitivity to model input quantities via the Sobol' indices approach, at no additional cost [158, 159]. The latter is not a feature of neural networks. Notably, the software for developing both FFNN and PCE reduced-order models exists in the form of MATLAB toolboxes [156, 160], making practical implementation of these to MOR approaches relatively easy.

Rear and front wheel braking torque	**NRMSE FOM - FFNN**	**NRMSE FOM - PCE**
τ_{rw}	2.75	2.04
τ_{fw}	6.1	6.54

TABLE 5.3: NRMSE of rear and front wheel braking torque between FOM-NSs and RO-NSs. The values are in percent and correspond to the mean values of the 30 HS evaluations.



(a) Front wheel braking torque.

(b) Rear wheel braking torque.

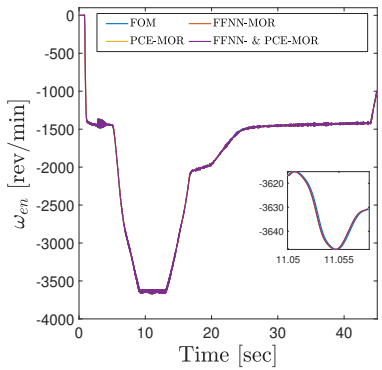
FIGURE 5.5: Front and rear wheel braking torques for the cases of FOM, FFNN-based and PCE-based MOR. Mean values of the 30 HS evaluations.

Hybrid model response	**NRMSE FOM - FFNN**	**NRMSE FOM - PCE**	**NRMSE FOM - FFNN&PCE**
ω_{rw}	1.39	1.64	1.43
ω_{fw}	1.38	1.63	1.42
ω_{en}	0.13	0.17	0.16
ω_{vd}	1.39	1.64	1.43
τ_{en}	1.43	1.85	1.68
τ_{vd}	0.09	0.09	0.07
v	1.47	1.74	1.51

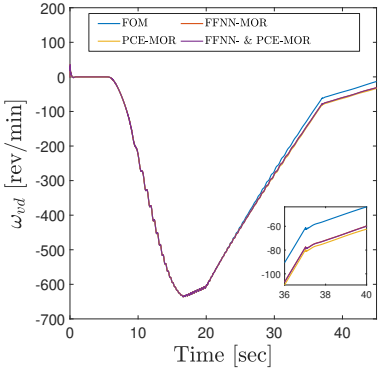
TABLE 5.4: NRMSEs for the mean values of the selected virtual hybrid model response quantities. The values are in percent.

5.4 Conclusions

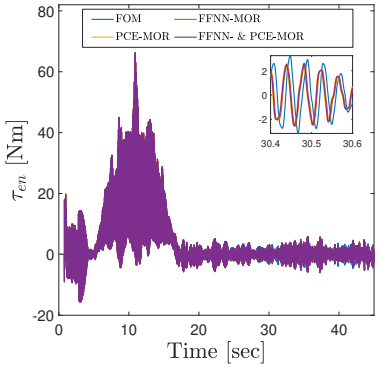
In this chapter, several MOR techniques are bench-marked for RTHS. The use of MOR techniques in RTHS is especially important for hybrid models encompassing high-dimensional nonlinear NS, risking distorting the time scale of HS by introducing time delays due to the extended computational power needed. The selected MOR techniques are based on data-driven regression techniques and particularly on PCE and FFNN. A parametric case study consisting of a virtual hybrid model is used to validate the framework.



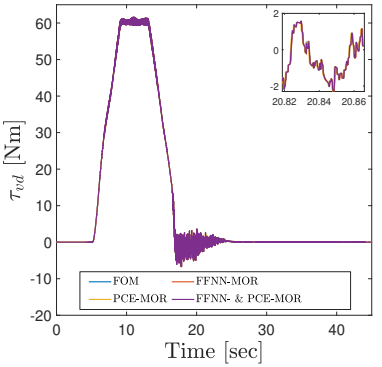
(a) Engine angular velocity.



(b) Transmission output shaft angular velocity.



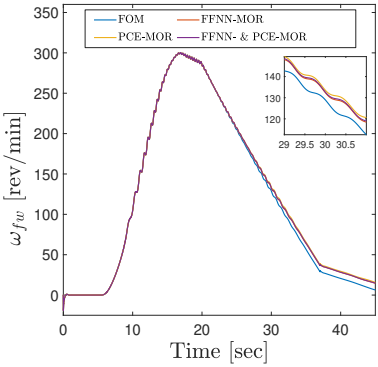
(c) Engine torque.



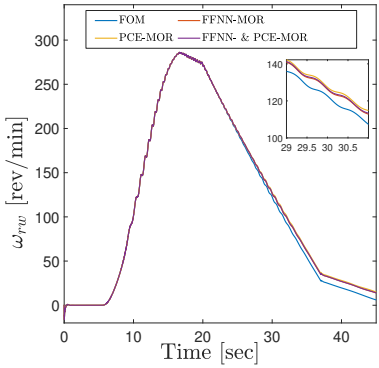
(d) Transmission output shaft torque.

FIGURE 5.6: Selected hybrid model responses evaluated in each testing scenario. Mean values of the 30 HS evaluations. Zoomed axes are added to highlight the difference between the shown response time histories.

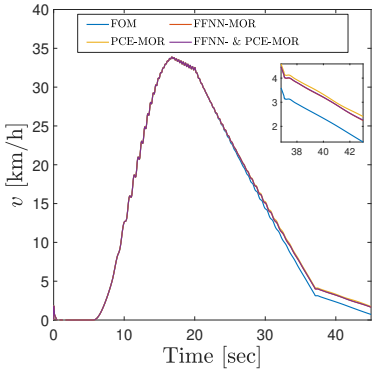
FFNN and PCE surrogates are trained to replicate the dynamic response of selected NS. Substitution of their respective full-order components forms the reduced-order hybrid model. Specific parameters of the hybrid model are varied and multiple HS evaluations are conducted to determine the robustness of the MOR framework. Comparisons with the full-order hybrid model responses are made to assess whether the reduced-order hybrid models can accurately replicate the initial dynamic responses while the corresponding errors are proven to be negligibly small. Results confirm



(a) Front wheel angular velocity.



(b) Rear wheel angular velocity.



(c) Motorcycle velocity.

FIGURE 5.7: Selected hybrid model responses evaluated in each testing scenario. Mean values of the 30 HS evaluations. Zoomed axes are added to highlight the difference between the shown response time histories.

the validity, benefits and performance of the proposed MOR framework. In particular, for the case study presented in this work, MOR techniques allowed for a $\approx 52\%$ reduction of the computational cost, for a maximum of 6.54% loss on absolute accuracy versus FOM. MOR techniques are therefore capable of producing hybrid models for more reliable RTHS without a significant loss of accuracy. The use of MOR techniques pushes the boundaries of hybrid simulation further by enabling the use of more complex and computationally involved NSs.

This chapter concludes the first research objective of this dissertation. The next chapter will serve as an introduction to the second research objective.

Chapter 6

Uncertainty quantification background

6.1 Introduction

The dynamic response of a physical system can deviate from its nominal performance, evaluated at its design phase, as a consequence of the presence of uncertainties. In a trial of quantitative modeling of these uncertainties, the science of *uncertainty quantification* has emerged. It incorporates both mathematical modeling and simulations as well as applied statistics and probability theory, with the overreaching goal to investigate and understand how uncertainties propagate through physical systems and how they can be reduced. Section 2.1.6 provided an overview of potential sources of uncertainties and errors that are present in HS.

The goal of this chapter is to introduce the uncertainty quantification framework that is used later on, in Chapters 7 and 8, to address the second research objective of this dissertation. Recall that the second research objective is to understand which uncertain variables affect hybrid model responses the most and in which way. The main objective of the uncertainty quantification framework is to conduct global sensitivity analysis (GSA) with Sobol' indices, using surrogate modeling techniques. Section 6.2 will firstly outline the utilized uncertainty quantification framework, while Sections 6.3-6.5 offer a brief overview of quantification of sources of uncertainty, uncertainty propagation and sensitivity analysis techniques.

This chapter introduces only shortly the background material to understand the tools that are utilized later on and does not serve as an extensive review for uncertainty quantification techniques and practices. For a more comprehensive review, the reader is encouraged to consult [141, 159, 161, 162, 163].

6.2 Uncertainty quantification framework

In the scope of this dissertation, an uncertainty quantification framework is utilized, based on GSA with Sobol' indices and surrogate modeling. The outline of the framework used in Chapters 7 and 8 is the same. However, different techniques are employed in each chapter in order to tackle the respective raised challenges. Chapters 7 and 8 will provide in more detail the

mathematical formulation of each utilized tool. The outline of the utilized uncertainty quantification framework consists of mainly fours steps, as follows:

1. Identification of the input uncertainties and outputs quantities of interest (QoI) of the hybrid model. The outcome of this step is a vector of random input variables and a vector of output hybrid model QoI.
2. Quantification of the sources of uncertainties; probabilistic distribution assignment to the system parameters of interest in order to investigate their variance.
3. Propagation of the input uncertainty through the model. Surrogate modeling techniques are key methodologies at this stage.
4. GSA with Sobol' indices is conducted, indicating the most sensitive input uncertainties by quantifying how much they affect the output QoI. The outcome of this step is Sobol' indices per QoI.

The development and implementation of the surrogate modeling, as well as the GSA, in both Chapters 7 and 8 was performed with the UQLab software framework developed by the Chair of Risk, Safety and Uncertainty Quantification in ETH Zurich [160].

6.3 Quantification of sources of uncertainty

To quantify the sources of uncertainties, or in other words to probabilistic model the input uncertainties under consideration, several techniques exist in the literature, depending on whether there are available data. For the case that there is no available data related to the uncertain input variable we want to characterize, *expert judgment* and the *principle of maximum entropy* may be employed. If large datasets are available, *statistical methods* can be used. For the case that only limited data is available, *Bayesian inference* can be applied, combining both expert judgment and the limited data available. The overall objective of this step is to identify the probabilistic distribution that best describes the random input variable we want to characterize.

6.3.1 *Expert judgment and principle of maximum entropy*

When no data is available, the probabilistic distribution may be prescribed by expert judgment; prior knowledge of similar systems or information available in the literature. Additionally, physical interpretation of the considered random variable, e.g. sign or physical bounds of the parameter, can be

of assistance in order to characterize it. To offer some examples, according to the JCSS Probabilistic Model Code [164], it is a good practice to assign Lognormal distributions for material properties, Gaussian distributions for measurement noise, Gumbel distributions for loads and extreme events and Beta distributions for bounded parameters.

The principle of maximum entropy [165] can also be utilized in combination with the available information mentioned above. In more detail, it is a measure which states that the PDF f_X which best characterizes the random variable X is the one that maximizes the entropy H, following:

$$H = -\int_{\mathcal{D}_X} f_X(x) \log f_X(x) dx. \tag{6.1}$$

6.3.2 *Statistical methods*

When data is available, statistical methods can be used to find the "most suitable" probability distribution, i.e. *descriptive statistics*, and to estimate the hyperparameters θ of this distribution, i.e. *statistical inference*. Methods for descriptive statistics are the *numerical summaries*, e.g. sample mean, sample variance, and the *graphical representations*, e.g. histogram, box plot, while methods for statistical inference are the *methods of moments* and the *maximum likelihood estimation*. With the method of moments, the hyperparameters of a distribution are determined by equating them to the sample moments, i.e. the empirical moments of the available data. On the contrary, in the maximum likelihood estimation method, the hyperparameters are determined by those which maximize the likelihood function, which is the probability of having observed the collected data. More specifically, given the observation dataset $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$ of a random variable $\boldsymbol{X}$, described by the joint PDF $f_{\boldsymbol{X}}(\boldsymbol{x}, \theta)$, the likelihood function admits:

$$L\left(\theta; \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right) = \prod_{i=1}^{n} f_{\boldsymbol{X}}(\boldsymbol{x}^{(i)}, \theta), \tag{6.2}$$

providing that the observations are independent and identically distributed. Therefore, the optimal hyperparameters $\hat{\theta}^{ML}$ are the ones that maximize the likelihood function with respect to θ:

$$\hat{\theta}^{ML} = \arg\max_{\theta} L\left(\theta; \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right). \tag{6.3}$$

Finally, *goodness-of-fit* criteria can be used to choose the "most suitable" model among various potential. Such criteria are the Akaike information criterion (AIC) [166] and the Bayesian information criterion (BIC) [167].

6.3.3 *Bayesian inference*

In Bayesian inference [168], a *prior* probability distribution $p'_{\theta}(\theta)$ is selected based on expert judgment and by incorporating the available data, a *posterior* distribution $p''_{\theta}(\theta)$ of the random variable is computed. Given the observation dataset $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ of a random variable $\mathbf{X}$, described by the joint PDF $f_{\mathbf{X}}(\mathbf{x}, \theta)$, the posterior distribution of θ admits:

$$p''_{\theta}(\theta) = \frac{1}{c} p'_{\theta}(\theta) L\left(\theta; \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right), \tag{6.4}$$

where $L\left(\theta; \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right)$ the likelihood function defined in Eq. (6.2) and c a normalized constant denoted by:

$$c = \int_{\mathcal{D}_{\theta}} p'_{\theta}(\theta) L\left(\theta; \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right) d\theta. \tag{6.5}$$

In the case studies presented in Chapters 7 and 8, no data are available and therefore the expert judgement method is used.

6.4 Uncertainty propagation

Consider a system $\mathbf{Y} = \mathcal{M}(\mathbf{X})$ with $Y \in \mathbb{R}$ and $\mathbf{X} \in \mathcal{D}_X \subset \mathbb{R}^N$ a random vector including the input parameters, with $f_Y(y)$ the PDF of $\mathbf{Y}$. Uncertainty propagation focuses on investigating how the uncertainty in $\mathbf{Y}$ relates to the uncertainty in $\mathbf{X}$.

Sudret and Der Kiureghian [169] proposed classes of various methods for uncertainty propagation, depending on the context of the application. In this regard, for the case that only the mean value μ_Y and standard deviation σ_Y of the system response are of interest, *second moment analysis* methods shall be employed. Among others, these include the *perturbation method* [170], the *weighted integral method* [171] and the *quadrature method* [172]. For the case that the interest is concentrated only on the tail of the response PDF $f_Y(y)$, this can relate to *reliability analysis* methods such as first-order reliability method (FORM), second-order reliability method (SORM) and importance sampling. Finally, for the case that the whole PDF $f_Y(y)$ is of interest, Monte Carlo simulation is the most well-established method. However, conducting several Monte Carlo simulations may be not reasonably affordable due to

the high computational/experimentation cost associated. To alleviate this burden, *surrogates* (a.k.a. metamodels or response surface) are a promising alternative. Surrogate models emulate the response of the system $\mathcal{M}$ with simpler-to-evaluate functions, and hence several model evaluations can be conducted at a much lower cost. Various surrogate modeling techniques are addressed in detail in Chapters 7 and 8.

6.5 Sensitivity analysis

The objective of sensitivity analysis is to quantify the contribution of each input variable to the uncertainty on the system's output response, taking into account their singular effects as well as their interactions with the other input variables. The outcome of the sensitivity analysis is a quantitative hierarchy of the input variables with respect to their influence on the system's response.

Typically, methods for sensitivity analysis can be classified into *local* and *global*. Local sensitivity analysis methods assess the local impact of the input uncertainty on the system's response. Such methods aim to compute the gradient of the system response with respect to its parameters over a nominal value. Examples of such methods are the finite-difference scheme and the direct differentiation [161, 173].

On the contrary, GSA methods take into account the whole input domain. A widely used class of GSA methods is the so-called *variance-based methods*. In the latter, the variance of the system's response is decomposed as a sum of contributions of each input parameter independently or in interaction thereof. Examples of such methods are the *Sobol' indices* [174], the *Borgonovo indices* [175] and the *Kucherenko indices* [176], among others. Another class of methods is the so-called *linearization methods* as they assume that the system under consideration is either linear or can be linearized around a nominal value. Such methods are the *perturbation* and *Cotter* method [177]. The report of Marelli et al. [178] offers a more comprehensive review of methods for sensitivity analysis. In this dissertation, GSA will be conducted using only Sobol' indices. GSA of both deterministic (Chapter 7) and stochastic hybrid models (Chapter 8) will be performed. In the latter chapter, extensions of the classical Sobol' indices will be discussed, to account for GSA of stochastic hybrid models.

6.6 Conclusions

This chapter serves as an introduction to the second research objective of the dissertation, namely to extend the deterministic state-of-the-art HS to

stochastic HS, in terms of performing GSA and identifying the effect of uncertainties to hybrid model QoI. In this regard, an uncertainty quantification framework is employed focusing on GSA and surrogate modeling. The outline of the framework was presented and its main steps were highlighted. A brief review of each step also took place. The framework's outline remains the same for both Chapters 7 and 8. However, different tools are utilized in each chapter in order to properly address the respective challenges, namely whether a hybrid model response is deterministic or stochastic. These are presented in the following two chapters.

Chapter 7

Global sensitivity analysis of hybrid models with deterministic physical substructures

Parts of the material presented in this chapter are published in the Mechanical Systems and Signal Processing journal [65] and others are submitted for publication and are currently under review. A preprint of the unpublished material can be found in [179]. Part of the work is also presented in the Engineering Mechanics Institute (EMI) Conference, held in Pasadena CA, USA, in 2019 [180] and in the 38th IMAC Conference and Exposition on Structural Dynamics, held in Houston TX, USA, in 2020 [181].

7.1 Introduction

The first part of this dissertation (Chapters 3-5) focuses on the first research objective, namely tackling challenges that arise within a real-time hybrid simulation (RTHS), with the overall objective to create a simulation tool that is able to produce high fidelity simulation results. In parallel, work is concentrated on how to extend this state-of-the-art deterministic HS to stochastic HS, with the overall objective to reveal the inner workings of the hybrid model by quantitatively identifying the effect of input uncertainties on selected output QoI of the hybrid model. Recall that the latter corresponds to the second research objective of this dissertation.

To do so, global sensitivity analysis (GSA) utilizing Sobol' sensitivity indices is performed using the uncertainty quantification framework, whose outline was briefly presented in Section 6.2. In this chapter, the mathematical formulation of the tools used in here is presented in more detail and the framework is applied to two case studies. Both case studies encompass hybrid models whose physical substructures (PS) are treated as deterministic[1]. Following the outline of the framework, firstly the input uncertainties are identified and the hybrid model output QoI are selected. Afterwards, through expert judgment, probability distributions are assigned to the input uncertainties. Multiple HS evaluations are made to generate sufficient

1 With deterministic PS it is meant that nominally identical specimen have identical responses, plus some negligible measurement noise. This assumption is however alleviated in the next chapter, Chapter 8, in which a hybrid model with stochastic PS is examined.

data in order to develop surrogates of the original hybrid model with a negligibly small validation error. Surrogates are of particular importance at this stage, since in order to perform GSA, large datasets would be needed. Nevertheless, evaluating several times the dynamic response of either an experimental prototype or a highly complex computational model is not reasonably affordable. On the contrary, performing Monte Carlo simulations of the surrogate model can be achieved at a much lower computational cost. Additionally, when a polynomial chaos expansion (PCE) is chosen to be developed as the surrogate of the hybrid model QoI, the Sobol' indices can be derived analytically as a by-product of the PCE coefficients at no extra cost [158]. In this chapter, apart from PCE, also Kriging (a.k.a. Gaussian process) and polynomial chaos Kriging (PCK) are considered as surrogate modeling techniques. PCE was also used in Chapter 5 in order to emulate the response of highly nonlinear numerical substructures (NS) within the hybrid model and thus, reduce their complexity. In that case, PCE was used as a surrogate of specific only NS and not of the whole hybrid model. The response of PCE was also evaluated online, during the HS, and it was conceived as a concrete part of the hybrid model. In this chapter, PCE is used only offline, after several HS evaluations are performed. In this case, it emulates a specific output QoI of the overall hybrid model in order to perform GSA. Nevertheless, the mathematical formulation and principles of PCE are the same in both chapters. Therefore, the PCE background would not be repeated in this chapter as it was presented in Chapter 5. The work presented in this chapter serves also as a survey of performing GSA employing different surrogate modeling methods. Comparison of the accuracy between the examined surrogates is performed and potential deviations in the Sobol' indices obtained from each surrogate are highlighted.

This chapter is organized as follows. Sections 7.2 and 7.3 provide the mathematical formulation to develop the considered surrogates[2] and to perform GSA using Sobol' indices, respectively. Section 7.4 presents the utilized case studies and Section 7.5 draws conclusions.

7.2 Surrogate modeling methods

As mentioned in Section 5.2, considering an input vector $\mathbf{X} \in \mathcal{D}_{\mathbf{X}} \subset \mathbb{R}^N$ and a computational model $\mathbf{Y} = \mathcal{M}(\mathbf{X})$ with $Y \in \mathbb{R}^N$, surrogates formulate a map $\mathcal{M}^s : \mathbf{X} \mapsto \mathbf{Y}$ based on an obtained sample set of input points $\mathbf{X} = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right\}^T$ and of the respective output values, i.e. model

2 For the mathematical formulation of PCE, consult Section 5.2.1.

evaluations or experimental measurements, $\mathbf{Y} = \left\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)}\right\}^T$. The set of $\mathbf{X}, \mathbf{Y}$ realizations corresponds to the so-called experimental design (ED).

7.2.1 *Kriging*

Kriging (a.k.a. Gaussian process) is a widely used surrogate modeling method. It considers the output of a model $\mathbf{Y} = \mathcal{M}(\mathbf{X})$ as a realization of a Gaussian process, with $\mathbf{x} \in \mathcal{D}_{\mathbf{X}} \subset \mathbb{R}^N$ and follows [182, 183]:

$$\mathcal{M}^{Krig}(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{f}(\mathbf{x}) + \sigma^2 Z(\mathbf{x}, \omega), \tag{7.1}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_P)$ and $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_P(\mathbf{x}))$ are coefficients and regression functions, respectively. The product of the latter two corresponds to the mean values of the Gaussian process, i.e. the Kriging trend, while σ^2 to its variance. $Z(\mathbf{x}, \omega)$ is a stationary Gaussian process with zero mean and unit variance, defined by an autocorrelation function $R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ and its hyperparameters $\boldsymbol{\theta}$. The coefficients $\boldsymbol{\beta}$ are computed from the generalized least-square solution following:

$$\boldsymbol{\beta} = \left(\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F}\right)^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{Y}, \tag{7.2}$$

where $\mathbf{F}_{ij} = f_j(\mathbf{x}^{(i)})$ and $\mathbf{R}_{ik} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(k)}; \boldsymbol{\theta})$ the correlation matrix with $i, k = 1, \ldots, N$ and $j = 1, \ldots, P$.

In Kriging, the model Y and surrogate $\widehat{Y}(\mathbf{x}) = \mathcal{M}^{Krig}(\mathbf{x})$ response is assumed to have a joint Gaussian distribution and therefore the mean value and variance of the prediction $\widehat{Y}(\mathbf{x})$ read [182, 184]:

$$\mu_{\widehat{Y}}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} + \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1} \left(\mathbf{Y} - \mathbf{F}\boldsymbol{\beta}\right), \tag{7.3}$$

$$\sigma^2_{\widehat{Y}}(\mathbf{x}) = \sigma^2 \left(1 - \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}) + \mathbf{u}(\mathbf{x})^T \left(\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F}\right)^{-1} \mathbf{u}(\mathbf{x})\right), \tag{7.4}$$

where $\mathbf{r}_i(\mathbf{x}) = R(\mathbf{x}, \mathbf{x}^{(i)}; \boldsymbol{\theta})$ consists of the cross-correlations between predictions $\mathbf{x}$ and sample points $\mathbf{x}^{(i)}$ with $i = 1, \ldots, N$ and $\mathbf{u}(\mathbf{x}) = \mathbf{F}^T \mathbf{R}^{-1} \mathbf{r}(\mathbf{x}) - \mathbf{f}(\mathbf{x})$.

7.2.1.1 *Trend families*

The first step in constructing a Kriging surrogate is to determine its trend family. Three common family types can be found in the literature [184, 185], namely the simple Kriging expressed by $\boldsymbol{\beta}^T \mathbf{f}(\mathbf{x}) = \sum_{j=1}^{P} f_j(\mathbf{x})$, the

ordinary with $\boldsymbol{\beta}^T \mathbf{f}(\mathbf{x}) = \beta_0$ and the universal which follows $\boldsymbol{\beta}^T \mathbf{f}(\mathbf{x}) = \sum_{j=1}^{P} \beta_j f_j(\mathbf{x})$. The latter is a generic type admitting a variety of formations, e.g. multivariate polynomials (see Section 7.2.2).

7.2.1.2 *Autocorrelation functions*

The autocorrelation functions used in Kriging metamodeling represent the relative position of two input sample points $\mathbf{x}$ and $\mathbf{x}'$. Some common autocorrelation function families [184, 185] are the linear:

$$R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \prod_{i=1}^{N} \max\left(0, 1 - \frac{|x^{(i)} - x'^{(i)}|}{\theta_i}\right), \tag{7.5}$$

the exponential:

$$R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \exp\left(-\sum_{i=1}^{N} \frac{|x^{(i)} - x'^{(i)}|}{\theta_i}\right), \tag{7.6}$$

the Gaussian:

$$R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \exp\left(-\sum_{i=1}^{N} \left(\frac{|x^{(i)} - x'^{(i)}|}{\theta_i}\right)^2\right) \tag{7.7}$$

and the Matérn:

$$R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}, \nu) = \prod_{i=1}^{N} \frac{1}{2^{\nu-1}\Gamma(\nu)} \left(2\sqrt{\nu}\frac{|x^{(i)} - x'^{(i)}|}{\theta_i}\right)^{\nu} \mathcal{K}_\nu \left(2\sqrt{\nu}\frac{|x^{(i)} - x'^{(i)}|}{\theta_i}\right), \tag{7.8}$$

where $\nu \geqslant 1/2$ corresponds to the shape parameter, Γ is the Euler Gamma function and $\mathcal{K}_\nu$ is the modified Bessel function of the second kind.

7.2.1.3 *Hyperparameters estimation*

Following the selection of trend and autocorrelation family, the next step in building the Kriging surrogate consists of estimating the unknown hyperparameters $\boldsymbol{\theta}$. One of the existing methodologies is the maximum-likelihood estimation. The goal of the latter is to maximize the likelihood of model evaluations $\mathbf{Y}$ by determining the appropriate set of the Kriging parameters $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\theta}$. The likelihood function of $\mathbf{Y}$ admits:

$$\mathcal{L}(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}; \mathbf{Y}) = \frac{\left(\det \mathbf{R}\right)^{-1/2}}{\left(2\pi\sigma^2\right)^{N/2}} \exp\Big(-\frac{1}{2\sigma^2}\left(\mathbf{Y}-\mathbf{F}\boldsymbol{\beta}\right)^{\mathsf{T}}\mathbf{R}^{-1}\left(\mathbf{Y}-\mathbf{F}\boldsymbol{\beta}\right)\Big). \quad (7.9)$$

The hyperparameters $\boldsymbol{\theta}$ are computed by solving the optimization problem:

$$\begin{aligned}\boldsymbol{\theta} &= \arg\min_{\boldsymbol{\theta}\in\mathcal{D}_{\boldsymbol{\theta}}} \left(-\log \mathcal{L}(\boldsymbol{\theta}; \mathbf{Y})\right) \\ &= \arg\min_{\boldsymbol{\theta}\in\mathcal{D}_{\boldsymbol{\theta}}} \frac{1}{2}\big(\log(\det \mathbf{R}) + N\log(2\pi\sigma^2) + N\big). \end{aligned} \quad (7.10)$$

For a more detailed description of the derivation of Eq. (7.10) along with additional methods for hyperparameter estimation, e.g. cross-validation estimation, the reader should refer to [182, 185].

7.2.1.4 *Optimization methods*

The solution of Eq. (7.10) is obtained by employing an optimization algorithm. Two main categories of such algorithms are the local, e.g. gradient-based, and global, e.g. evolutionary algorithms, methods. Local methods need less objective functions evaluations and can converge faster. However, their performance may be comprised due to multiple local minima. Integrating characteristics from both local and global methods, results in a third category, the hybrid methods. Such a method is for example the hybrid genetic algorithm [184, 186].

The Kriging surrogate used later in the case studies of this chapter is of the ordinary trend type using the Matérn autocorrelation function family, while the hyperparameters are estimated with the maximum-likelihood method and optimized using the hybrid genetic algorithm method.

7.2.2 *Polynomial chaos Kriging*

PCK is another surrogate modeling method that combines features from PCE and Kriging into one. In more detail, PCK is a Kriging model with a universal trend type. Its trend though is formulated from sparse orthogonal polynomials. The output of PCK combines Eq. (5.4) and Eq. (7.1) and follows [187, 188, 189]:

$$\mathcal{M}^{PCK}(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\mathbf{X}) + \sigma^2 Z(\mathbf{x}, \omega). \quad (7.11)$$

The first term, $\sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\mathbf{X})$, is the trend of the Kriging model as described in Section 7.2.1 and is computed from sparse PCE as described in Section 5.2.1. The second term of Eq. (7.11) corresponds to the variance of the Gaussian process σ^2 and $Z(\mathbf{x}, \omega)$ is a stationary Gaussian process with zero mean and unit variance as of Eq. (7.1).

Therefore, building a PCK involves a step of constructing the PCE and an additional one to determine the Kriging model, i.e. to estimate its parameters. Two ways of combining the above two steps are the so-called sequential and optimal approaches [187, 188, 190]. In the sequential approach, the PCK is constructed sequentially, meaning that a sparse PCE is constructed first, using the LAR algorithm, and then the parameters of Eq. (7.11) are determined as in the common Kriging modeling. Whereas in the optimal approach, the PCK is constructed iteratively. More precisely, the set of multivariate orthogonal polynomials are identified as before using the LAR algorithm, but in this case, a list of ranked polynomials are computed from the LAR algorithm and are selected depending on their correlation to the current residual in each iteration. Afterwards, they are added one-by-one to the trend of PCK and the coefficients of the latter are computed for each case, resulting in a $|\mathcal{A}|$ number of individual PCK surrogates. The optimal PCK is the one with the minimum leave-one-out cross-validation error (see Section 7.2.3 for more details on this error) among the others.

The PCK surrogate used later in the case studies of this chapter is developed using the optimal approach and the Matérn autocorrelation function family is used along with the maximum-likelihood estimation and the hybrid genetic algorithm optimization method.

7.2.3 *Leave-one-out cross-validation error*

To assess the performance of the developed surrogates the leave-one-out (LOO) cross-validation error is used, defined by:

$$\varepsilon_{LOO} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{M}(\mathbf{x}^{(i)}) - \mathcal{M}^{s \setminus i}(\mathbf{x}^{(i)}) \right)^2, \tag{7.12}$$

where $\mathcal{M}^{s \setminus i}(\mathbf{x}^{(i)})$ corresponds to the surrogates developed from the subset of the ED, acquired by removing the $\mathbf{x}^{(i)}$ point.

7.3 Global sensitivity analysis with Sobol' indices

Broadly speaking, GSA aims to quantitatively identify the level each input variable in $\mathbf{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}^T$, $\mathbf{X} \in \mathcal{D}_X \subset \mathbb{R}^N$ affects the response of $Y = \mathcal{M}(\mathbf{X})$, $Y \in \mathbb{R}$. As mentioned previously, in this dissertation, the Sobol' indices are used for GSA. It should be noted that this method is only valid for independent input variables.

7.3.1 *Sobol'-Hoeffding decomposition*

For $\mathbf{X} \in \mathcal{D}_X \subset \mathbb{R}^N$ with independent components, $Y = \mathcal{M}(\mathbf{X})$ with $\mathrm{Var}[Y] < +\infty$ can be rewritten as [65, 174]:

$$\begin{aligned}\mathcal{M}^s(\mathbf{X}) &= \mathcal{M}_0 + \sum_{i=1}^{N} \mathcal{M}_i(x^{(i)}) + \sum_{1 \leqslant i \leqslant j \leqslant N} \mathcal{M}_{ij}(x^{(i)}, x^{(j)}) + \cdots + \mathcal{M}_{1,2,\ldots,N}(\mathbf{X}) \\ &= \mathcal{M}_0 + \sum_{\substack{\mathbf{u} \subset \{1,\ldots,N\} \\ \mathbf{u} \neq \emptyset}} \mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}),\end{aligned} \tag{7.13}$$

where $\mathcal{M}_0$ is constant and denotes the mean value of Y, $\mathbf{u} \neq \emptyset$, $\mathbf{u} \subset \{1, \ldots, N\}$ and $\mathbf{X}_{\mathbf{u}}$ is a sub-vector of $\mathbf{X}$, including the $\mathbf{u}$-indexed elements. The summands in the latter equation corresponds to 2^{N-1} terms. The elementary functions are defined by conditional expectations:

$$\mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}) = \sum_{\mathbf{v} \subset \mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \mathbb{E}\left[Y \mid \mathbf{X}_{\mathbf{v}} = \boldsymbol{x}_{\mathbf{v}}\right], \tag{7.14}$$

where $|\mathbf{u}|$ gives the cardinality of $\mathbf{u}$. When the following holds:

$$\int_{\mathcal{D}_{X_m}} \mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}) f_{X_m}(x_m)\, \mathrm{d}x_m = 0, \text{ if } m \in \mathbf{u}, \tag{7.15}$$

the Sobol'-Hoeffding decomposition of Eq. (7.13) is unique. Due to this uniqueness, the orthogonality property:

$$\mathbb{E}\left[\mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}) \cdot \mathcal{M}_{\boldsymbol{v}}(\mathbf{X}_{\boldsymbol{v}})\right] = 0, \text{ if } \mathbf{u} \neq \boldsymbol{v}, \tag{7.16}$$

also holds. Because of Eqs. (7.15) and (7.16) the total variance of $\mathcal{M}(\mathbf{X})$ decomposes as:

$$D = \mathrm{Var}\left[\mathcal{M}(\mathbf{X})\right] = \sum_{\substack{\mathbf{u}\subset\{1,\ldots,N\}\\ \mathbf{u}\neq\emptyset}} D_{\mathbf{u}}, \tag{7.17}$$

where

$$D_{\mathbf{u}} = \mathrm{Var}\left[\mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}})\right] = \mathbb{E}\left[\mathcal{M}^2_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}})\right], \tag{7.18}$$

corresponds to the partial variance.

7.3.2 *Sobol' indices*

The Sobol' index $S_{\mathbf{u}}$ is determined as the ratio between the partial and total variance that represents the $\mathbf{u}$-indexed set of input variables and follows:

$$S_{\mathbf{u}} = \frac{D_{\mathbf{u}}}{D}. \tag{7.19}$$

For $|\mathbf{u}| = 1$, the separate impact of each input variable $x^{(i)}$ to the response $Y = \mathcal{M}(\mathbf{X})$ is described by the indices respective to this specific variable, namely the first-order indices $S_i^{(1)} = D_i/D$. The impact originating from the interaction of pairs of the input variables $(x^{(i)}, x^{(j)})$ and not already accounted in the S_i, S_j is described by the second-order indices $S_{ij}^{(2)} = D_{ij}/D$. The impact from interaction of a larger set of input variables is described from the higher-order indices. Total indices S_{T_i} describe the overall impact of each input variable, considering its first-order index and all interactions with the other input variables and admit:

$$S_{T_i} = 1 - S_{-i}, \tag{7.20}$$

where S_{-i} denotes the sum of all $S_{\mathbf{u}}$ of $\mathbf{u}$ but i. It follows that $\sum_{\mathbf{u}\neq\emptyset} S_{\mathbf{u}} = 1$.

7.3.3 *Sobol' indices estimation*

7.3.3.1 *Monte Carlo-based estimation*

The variances of Eqs. (7.17) and (7.18) can be obtained by estimators from Monte Carlo simulations. These read [178]:

$$\begin{aligned}\hat{\mathcal{M}}_0 &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{M}^s\left(x^{(i)}\right)\\ \hat{D} &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{M}^s\left(x^{(i)}\right)^2 - \hat{\mathcal{M}}_0^2\\ \hat{D}_{\mathbf{u}} &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{M}^s\left(x_n^{(i)}, x_{\sim n}^{(i)}\right)\mathcal{M}^s\left(x_n^{(i)}, x_{\sim n}^{\prime(i)}\right) - \hat{\mathcal{M}}_0^2,\end{aligned} \tag{7.21}$$

where $x_{\sim n}^{(i)}$ denotes the i-th realization of x that does not involve the n input variable and x' corresponds to a realization of $\mathbf{X}$ which is independent of $x = \left(x_n^{(i)}, x_{\sim n}^{(i)}\right)^T$. Once the total and partial variance estimators are computed, the Sobol' indices can be derived from Eqs. (7.19) and (7.20).

7.3.3.2 *Sobol' indices from polynomial chaos expansion*

Recall from Eq. (5.4), that the PCE of $\mathcal{M}(\mathbf{X})$ admits:

$$\mathcal{M}^{PCE}(\mathbf{X}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\mathbf{X}). \tag{7.22}$$

Due to the orthonormality of the PCE basis, the mean and variance of $\mathcal{M}^{PCE}(\mathbf{X})$ can be computed analytically from the $\mathbf{y}$ coefficients at no extra cost and follow [159]:

$$\hat{\mu} = \mathbb{E}\left[\mathcal{M}^{PCE}(\mathbf{X})\right] = \mathbb{E}\left[\sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\mathbf{X})\right] = y_0 \tag{7.23}$$

$$\hat{\sigma}^2 = \mathrm{Var}\left[\mathcal{M}^{PCE}(\mathbf{X})\right] = \mathbb{E}\left[\left(\mathcal{M}^{PCE}(\mathbf{X}) - y_0\right)^2\right] = \sum_{\substack{\boldsymbol{\alpha}\in\mathcal{A}\\ \boldsymbol{\alpha}\neq 0}} y_{\boldsymbol{\alpha}}^2. \tag{7.24}$$

The above equations hold since $\Psi_0 \equiv 1$. Therefore, for the case that a PCE of Y is already constructed, the Sobol'-Hoeffding decomposition of Eq. (7.13) can be rewritten as:

$$\mathcal{M}^{PCE}(\mathbf{X}) = y_0 + \sum_{\substack{\mathbf{u}\subset\{1,\ldots,N\}\\ \mathbf{u}\neq\emptyset}} \mathcal{M}_{\mathbf{u}}^{PCE}(\mathbf{X}_{\mathbf{u}}), \tag{7.25}$$

where $\mathcal{A}_{\mathbf{u}} = \{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_m \neq 0 \text{ if and only if } m \in \mathbf{u}\}$ corresponds to the set of multi-indices including only $\mathbf{u}$ and $\mathcal{M}^{PCE}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_{\mathbf{u}}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\mathbf{X}_{\mathbf{u}})$, as of Eq. (7.22). Thus, because the Sobol'-Hoeffding decomposition is unique, there exists an analytical representation of $\mathcal{M}^{PCE}_{\mathbf{u}}$ [158]. Accordingly to Eqs. (7.23), (7.24), the total and partial variances of $\mathcal{M}^{PCE}(\mathbf{X})$ admit:

$$\begin{aligned} D &= \mathrm{Var}\left[\mathcal{M}^{PCE}(\mathbf{X})\right] = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y^2_{\boldsymbol{\alpha}} \\ D_{\mathbf{u}} &= \mathrm{Var}\left[\mathcal{M}^{PCE}_{\mathbf{u}}(\mathbf{X})\right] = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_{\mathbf{u}}} y^2_{\boldsymbol{\alpha}}. \end{aligned} \tag{7.26}$$

Hence, the first-order and total Sobol' indices are obtained from:

$$\begin{aligned} S^{(1)}_i &= \frac{\sum_{\boldsymbol{\alpha}\in\mathcal{A}_i} y^2_{\boldsymbol{\alpha}}}{\sum_{\boldsymbol{\alpha}\in\mathcal{A}} y^2_{\boldsymbol{\alpha}}}, \quad \mathcal{A}_i = \left\{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i > 0,\ \alpha_{i\neq j} = 0\right\} \\ S_{T_i} &= \frac{\sum_{\boldsymbol{\alpha}\in\mathcal{A}^T_i} y^2_{\boldsymbol{\alpha}}}{\sum_{\boldsymbol{\alpha}\in\mathcal{A}} y^2_{\boldsymbol{\alpha}}}, \quad \mathcal{A}^T_i = \left\{\boldsymbol{\alpha} \in \mathcal{A} : \alpha_i > 0\right\}, \end{aligned} \tag{7.27}$$

respectively. Therefore, the Sobol' indices can be obtained as a by-product of the PCE coefficients at no extra cost [158].

For a more comprehensive review of GSA using Sobol' indices as well as of the development of Kriging and PCK surrogates, the reader is encouraged to consult [141, 159, 163, 178, 184, 187].

7.4 Case studies

In order to provide validation of the proposed uncertainty quantification framework for HS, two case studies are presented. The first one is experimental, conducted using a 3-DoFs prototype structure subjected to earthquake ground motion excitation. The second case study corresponds to a virtual HS encompassing a prototype motorcycle subjected to a prescribed driving scenario. In both case studies, three surrogate techniques are utilized for GSA in HS, namely PCE, Kriging and PCK, serving as a review of how GSA can be performed using different surrogates.

7.4.1 *Case study 1: HS of a prototype structure*

7.4.1.1 *Problem formulation*

The prototype structure of case study 1 (CS1) consists of a simply-supported beam coupled at the ends with two rotational elastic restraints, two lumped masses, and one dashpot. The beam is excited with the ground motion excitation $a_g(t)$, whose direction is orthogonal to its axis. The prototype structure is characterized by a geometrical nonlinearity since the beam's bending and axial deformation are coupled for large displacements. In Figure 7.1 a schematic view of the prototype structure is displayed along with its hybrid model, substructured into one PS and two NS.

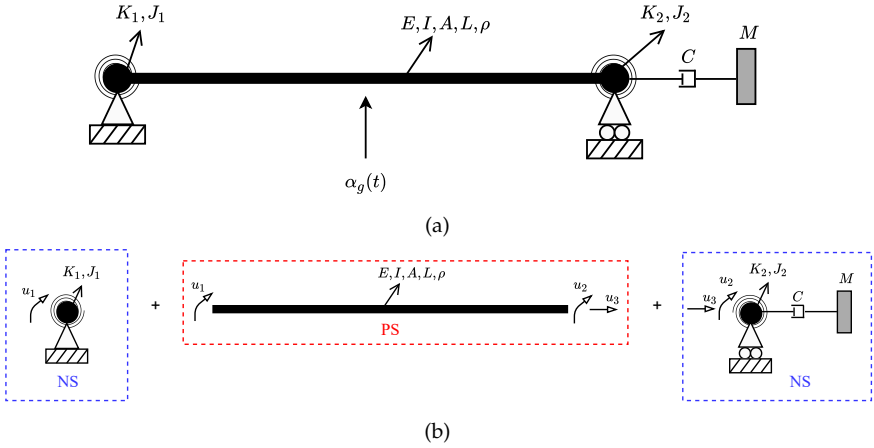


FIGURE 7.1: CS1: (a) prototype structure, (b) hybrid model consisting of two NS and one PS.

The u_1 and u_2 variables of Figure 7.1 correspond to the two rotational DoFs of the simply-supported beam, whereas u_3 indicate its axial DoF. The PS corresponds to an iron plate characterized by length $L = 470\,\text{mm}$, Young modulus $E = 100$ GPa, density $\rho = 7850\ \text{kg/m}^3$, cross-section of 200×2 mm, area $A = 400\ \text{mm}^2$ and moment of inertia $I = 133\ \text{m}^4$. Since that in this specific case study the HS are performed with a testing time scale equal to 50, a numerical mass matrix accounts for the PS inertia. The NS of the hybrid model consists of two rotational masses $J_1 = J_2 = 0.9\ \text{kgm}^2$ and a translation mass $M = 20000$ kg, lumped at the supports. The linear dash-pot is characterized by a damping coefficient $C = 300000\ \text{Ns/m}$. It is

the sole source of numerical damping and averts the excitation of the axial eigenmode of the hybrid model. The stiffness K_1 and K_2 of the two linear elastic rotation springs represent the non-deterministic properties of the prototype structure supports and are treated as input variables for the GSA.

The excitation $a_g(t)$ is defined by a velocity-pulse model, which offers an intuitive representation of near-fault earthquake ground motions [191]. The velocity-pulse model encompasses a cosine waveform of period T_p, modulated by a pulse function. The latter is a truncated cosine function of period γT_p and amplitude V_p, which coincides to the peak velocity of the pulse at time $t_{max,p}$. The velocity-pulse admits:

$$v_g(t) = \left(\frac{V_p}{2}\cos\left(2\pi\left(\frac{t-t_{max,p}}{T_p}\right)+\nu\right)-\frac{D_r}{\gamma T_p}\right)* \\ *\left(1+\cos\left(\frac{2\pi}{\gamma}\left(\frac{t-t_{max,p}}{T_p}\right)\right)\right), \tag{7.28}$$

where ν is the phase angle between the cosine waveform and the pulse function. D_r represents the non-zero residual displacement at the end of the pulse, given by:

$$D_r = V_p T_p \frac{\sin(\nu+\gamma\pi)-\sin(\nu-\gamma\pi)}{4\pi(1-\gamma^2)}. \tag{7.29}$$

The transverse inertial acceleration $a_g(t)$, which excites the prototype structure, is determined as the time derivative of the velocity-pulse model. The velocity-pulse model is defined for the time interval of $t \in \{0, \gamma T_p\}$ and zero otherwise. The pulse peak velocity V_P and period T_P are chosen as input variables for the GSA, while $t_{max,p} = \frac{\gamma T_p}{2}$ sec, $\gamma = 2$ and $\nu = 0.1$ rad. Figure 7.2 illustrates a realization of the velocity-pulse model defined in Eqs. (7.28) and (7.29) with $V_P = 2\,\frac{m}{sec}$ and $T_P = 1.25$ sec.

Two hybrid model output response QoI are selected for GSA, namely the maximum absolute rotation at the left support of the beam $u_{1,max}$ and the maximum absolute bending moment at the same support $r_{1,max}$. The QoI are measured using the vertical actuator displacement sensor and the vertical actuator load cell, respectively as shown in Figure 7.1. Accordingly, to the notation presented in Sections 7.2 and 7.3, the vector of input variables and QoI follow:

$$\begin{aligned} \mathbf{X} &= \{V_P, T_P, K_1, K_2\} \\ \mathbf{Y} &= \{u_{1,max}, r_{1,max}\} \end{aligned} \tag{7.30}$$

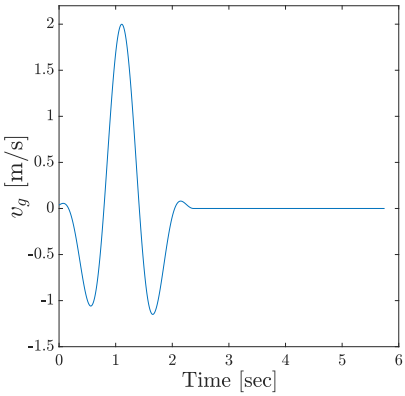


FIGURE 7.2: Sample velocity-pulse time history excitation used in the CS1 with $V_P = 2\,\frac{m}{sec}$ and $T_P = 1.25\,sec$.

Table 7.1 reports the characteristics of the probability distributions assigned to each input variable. The intervals of Table 7.1 are chosen as such to result in a linear response of the hybrid model, compatible with the force and displacement capacities of the HS test setup. In more detail, for the case that K_1 equals K_2 and both vary within the range reported in Table 7.1, the period of the first bending mode of the hybrid model lies in the range between 0.68 and 0.45 sec. This range overlaps with the period range of the velocity-pulse excitation, as the latter varies between 0.50 and 2.00 sec. In this regard, this frequency range overlap is anticipated to influence the dynamic amplification of both response QoI.

7.4.1.2 *Hybrid simulation setup*

The 3-DoFs HS test rig used to conduct the HS for this case study is located in the IBK Structural Testing Laboratory at ETH Zurich and it is a stiff loading frame equipped with four electromechanical actuators interfaced to an INDEL real-time computer. The 3-DoFs HS test rig is designed to test plate specimens with an approximate footprint of 200 × 500 mm and thickness varying between 1 and 3 mm. Figure 7.3 illustrates the architecture of the HS setup, including a close-up view of the plate specimen accommodation. Two axonometric views of the 3-DoF test rig, consisting of the main hardware components are shown in Figure 7.4.

Parameter	Probability Distribution	Lower Bound	Upper Bound	Parameter Description	Units
K_1	Uniform	20	120	Rotational stiffness left spring	$\frac{Nm}{rad}$
K_2	Uniform	20	120	Rotational stiffness right spring	$\frac{Nm}{rad}$
V_p	Uniform	1.00	3.00	Peak of velocity pulse	$\frac{m}{sec}$
T_p	Uniform	0.50	2.00	Period of velocity pulse	sec

TABLE 7.1: Input variable intervals and assumed probability distributions.

The moving parts of the test rig are colored in yellow, the plate specimen in brown and the fixed parts in gray. The latter are fixed to a reaction frame. In order to impose the u_1 and u_2 rotations, two rack-pinion systems (10) are installed along the vertical actuator y_1 and y_2 (1). The rack-pinion systems aim at transforming the commanded displacements from the actuators to rotational DoFs, applied to the short edges of the plate specimen (6) through aluminum clamps (3). The two horizontal actuators x_1 and x_2 control the position of the moving frame mounted on profiled rail guides (4) and correspond to the axial DoF u_3 of the plate specimen (6). A laser sensor measures the out-of-plane deflection at the mid-span of the plate specimen (labeled u_L in Figure 7.3).

The GINLink bus connects actuator servo drivers INDEL SAC4 and all DAQ modules to the real-time computer INDEL SAM4, which executes the HS software. The latter is developed in MATLAB/SIMULINK, compiled, and downloaded to the INDEL SAM4 from the Host-PC. At each simulation time-step, the HS software imposes displacements u_1, u_2 and u_3 to the plate specimen, the PS, reads the corresponding restoring forces r_1, r_2 and r_3 measured using force transducers, and solves the coupled equation of motion of the hybrid model. For the time integration scheme used in the HS of this case study, a fixed time-step of 1 msec was used. A comprehensive description of the scheme is reported in [41].

7.4.1.3 *Case study 1 results*

Using the LHS method [104], 200 samples of the vector $\mathbf{X}$, which includes the input variable distributions presented in Table 7.1, were generated. For each sample, a HS was conducted utilizing the hybrid model of Figure 7.1,

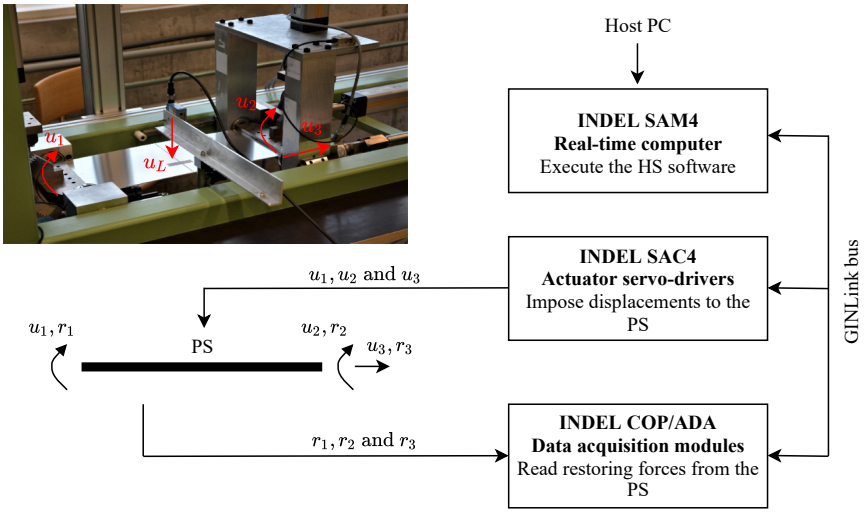


FIGURE 7.3: Architecture of the 3-DoFs HS test rig.

and the response QoI were collected in vector $\mathbf{Y}$. Figure 7.5 depicts a sample dynamic response of the rotation u_1 and bending moment r_1 at the left support of the beam of the hybrid model, evaluated with $K_1 = K_2 = 70\,\text{Nm/rad}$, $V_P = 2\,\text{m/sec}$ and $T_P = 1.25\,\text{sec}$. The QoI $u_{1,\max}$ and $r_{1,\max}$ are also highlighted in red color. Following the methodologies presented in Sections 5.2.1, 7.2.2 and 7.2.1, the ED $\{\mathbf{X}, \mathbf{Y}\}$ was used to develop a PCE, a PCK and a Kriging surrogate for each response QoI, namely for $u_{1,\max}$ and $r_{1,\max}$.

From Figure 7.5 it can be appreciated that the response of the bending moment r_1 is much noisier than rotation u_1. This can be interpreted as when the stiffness of the PS is much smaller than the stiffness of the actuation system, displacement control is adequately accurate, whereas restoring force measurements are more noisy. Therefore, measurement signal noise affects $r_{1,\max}$ more than $u_{1,\max}$ and hence has a further negative impact at the convergence of the corresponding surrogates. The latter can be confirmed from Figure 7.6, which illustrates validation scatter plots by comparing the measured QoI with the corresponding PCE, PCK and Kriging predictions. More specifically, for each QoI, the examined surrogates were developed using 150 samples of the ED and the remaining 50 samples were used to validate them.

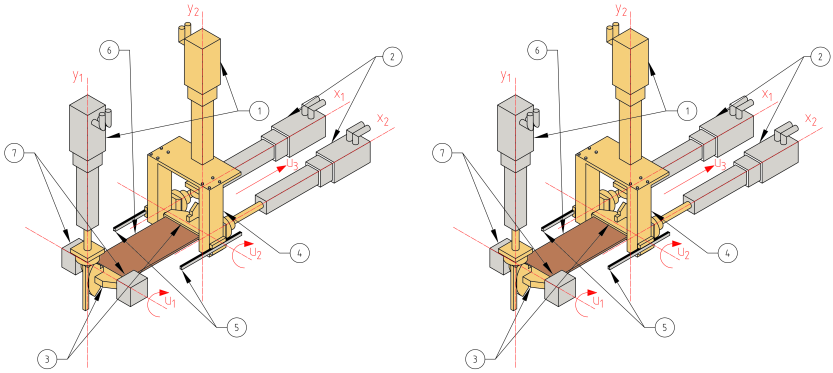
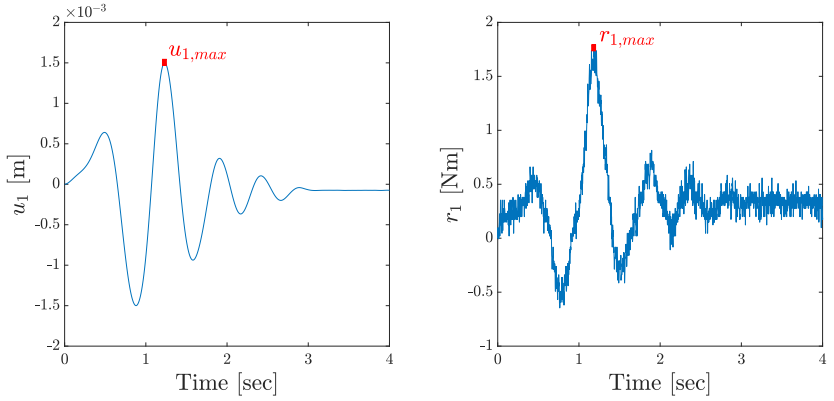


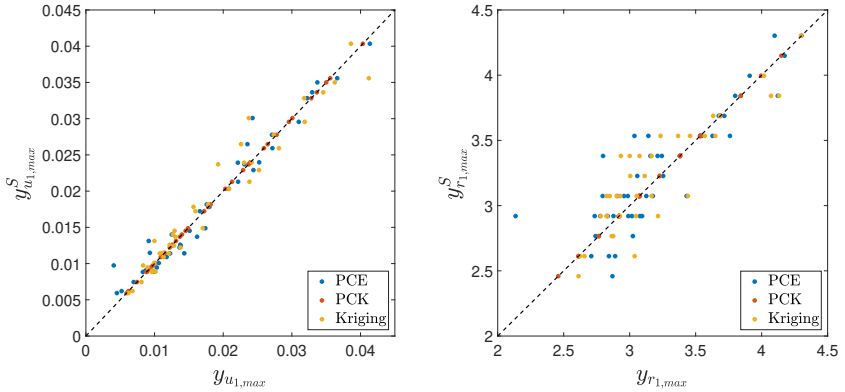
FIGURE 7.4: Axonometric views of the 3-DoFs HS test rig with main components: (1) vertical actuators, (2) horizontal actuators, (3) installation clamps, (4) moving frame, (5) profiled rail guides, (6) plate specimen, (7) hinges, (8) vertical actuator load cells, (9) horizontal actuator load cells and (10) rack-pinion systems. The moving parts are colored in yellow, the plate specimen is brown, while those parts fixed to the reaction frame are gray. The latter is omitted in this figure.

Figure 7.7 depicts the convergence of the LOO errors of the developed surrogates for both QoI. The LOO errors are calculated using EDs of increasing size from 15 to 200 samples. It should be noted that both force measurement noise and displacement control errors are sources of aleatory uncertainty, which propagate throughout the hybrid model response and is inherently stochastic. In this case study, the surrogates serve as a data denoiser and as a result, the LOO errors of Figure 7.7 do not converge to zero. In Figures 7.8 and 7.9 the convergence plots of the PCE-based moments estimates, namely the mean, standard deviation and coefficient of variation (CV), are illustrated for both QoI, respectively. Clearly, each PCE-based moment estimate is converging in a stable manner to reliable values attained for ED sizes larger than 50 samples. Recall that these moments are computed analytically from the PCE coefficients at no extra cost, as described by Eqs. (7.23) and (7.24). The latter though is not a feature of PCK or Kriging and hence Figures 7.8 and 7.9 do not include moments estimates from these two surrogates. Nevertheless, to further evaluate the performance of the developed surrogates, Figure 7.10 compares the predic-



(a) (b)

FIGURE 7.5: Sample HS dynamic response of the hybrid model with $K_1 = K_2 = 70\,\mathrm{Nm/rad}$ to a velocity-pulse ground motion excitation with $V_P = 2\,\mathrm{m/sec}$, $T_P = 1.25\,\mathrm{sec}$: (a) u_1 rotation at the left support of the beam and (b) r_1 bending moment at the same point. The QoI $u_{1,max}$ and $r_{1,max}$ are highlighted in red color.



(a) (b)

FIGURE 7.6: Comparison between QoI measurements and corresponding surrogate predictions for (a) $u_{1,max}$ and (b) $r_{1,max}$.

tion PDF of each surrogate with the histograms of the measurements for each QoI. The surrogate predictions are evaluated based on 200 random samples, generated from the distributions described in Table 7.1 using the LHS method.

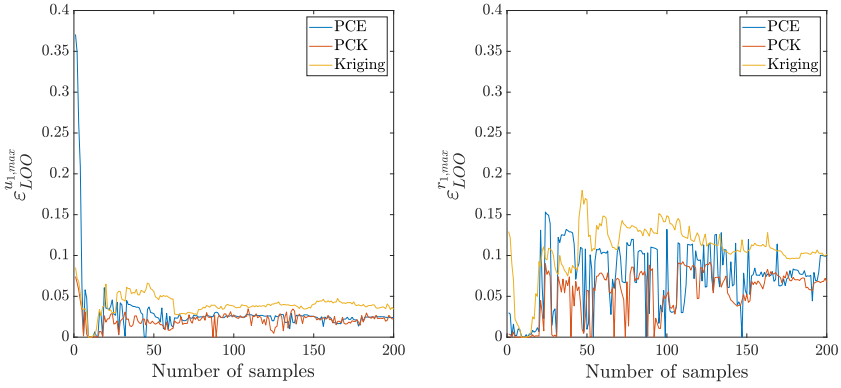


FIGURE 7.7: Convergence of LOO error estimates of (a) $u_{1,max}$ and (b) $r_{1,max}$ surrogates.

From Figures 7.6, 7.7 and 7.10 it can be appreciated that the performance and accuracy prediction of PCE, PCK and Kriging are similar. However, it should be noted that the computational time associated to train each surrogate differs. For the case of PCE and Kriging, the training computational time for each surrogate was approximately 1.2 and 2.3 sec, respectively. Nevertheless, for the case of PCK, it was approximately 21.7 sec. The latter finding can be intuitively interpreted as in the PCK case, the training phase consists of two steps; one constructing the PCE and a second one to determine the Kriging model. Note that these timings depend heavily on the ED sample size as well as on the settings each surrogate is developed.

Once the surrogates of each QoI are developed, the Sobol' indices can be computed. For the case of PCE, the Sobol' indices are computed analytically at no extra cost [158], as described in Section 7.3.3.2. For the case of PCK and Kriging though, this is not possible and thus the Sobol' indices are computed using Monte Carlo evaluations of the surrogate response, as described in Section 7.3.3.1. It should be noted that more than 10^4 Monte Carlo evaluations of the Kriging and PCK surrogates are needed to ensure convergence of the Sobol' indices. In CS1 (and in CS2 as addressed later on),

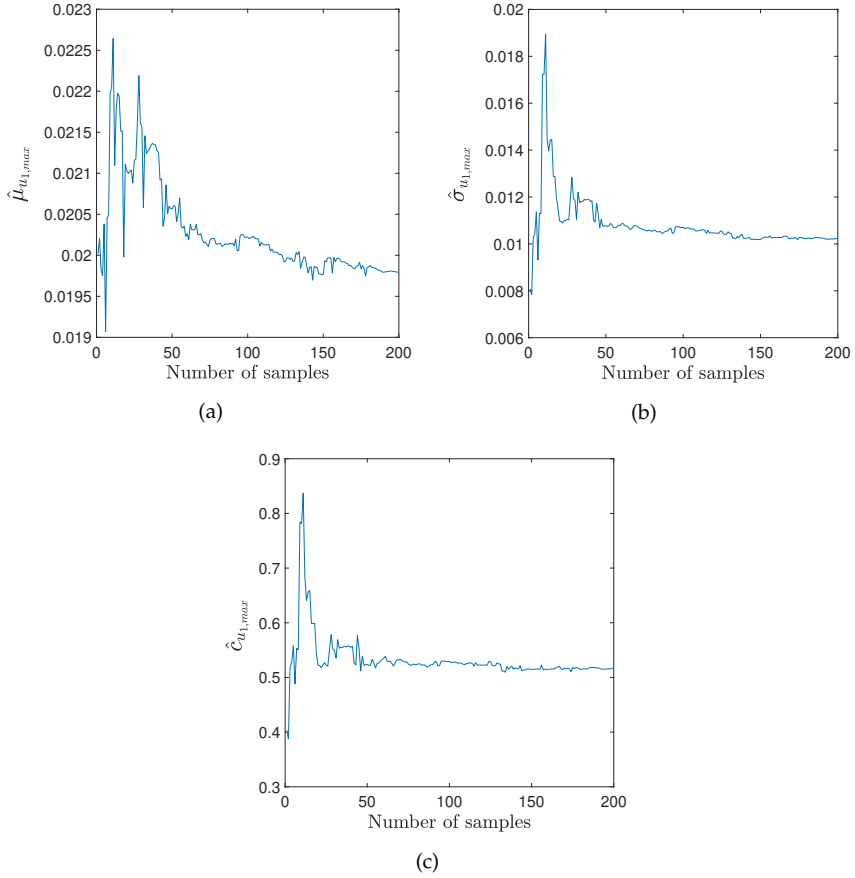


FIGURE 7.8: Convergence of PCE-based moments estimates of (a) mean, (b) standard deviation and (c) CV of QoI $u_{1,max}$.

10^5 evaluations of the response of the Kriging and PCK surrogates were performed to compute the respective Sobol' indices. However, it is worth mentioning that the computational cost of running these evaluations was negligibly small ($\approx$ 1 sec for each surrogate).

In Figure 7.11 the first-order and total Sobol' indices for both QoI $u_{1,max}$ and $r_{1,max}$ are reported. Clearly, both first-order and total Sobol' indices computed utilizing the different examined surrogates are admittedly close with each other. This confirms that the indices obtained from the PCK

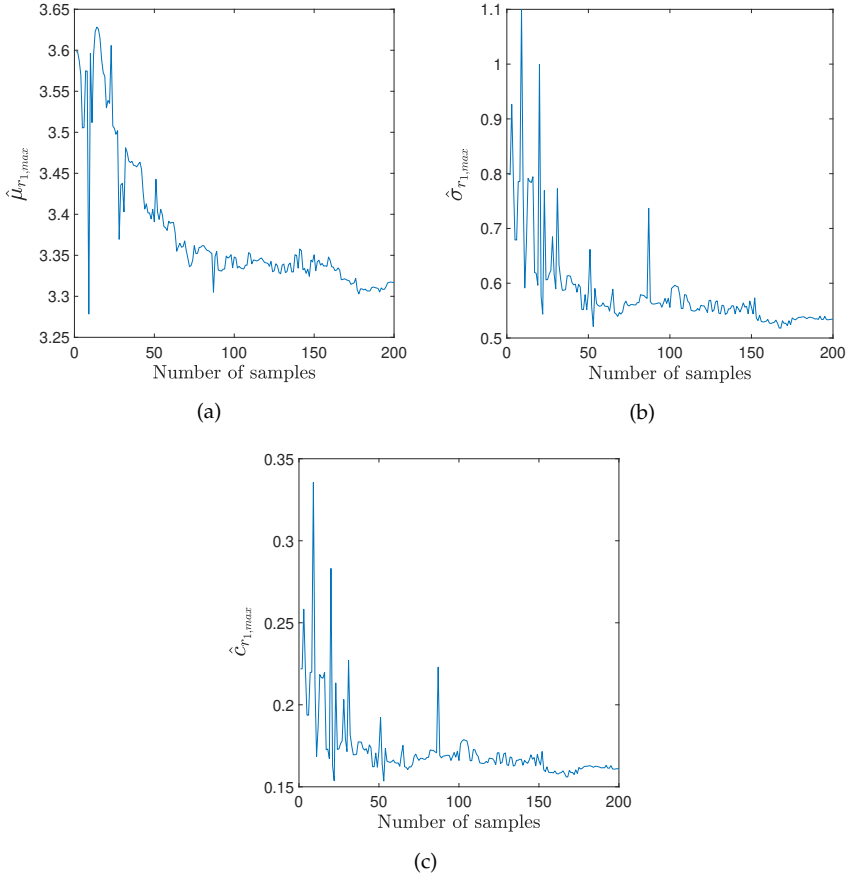
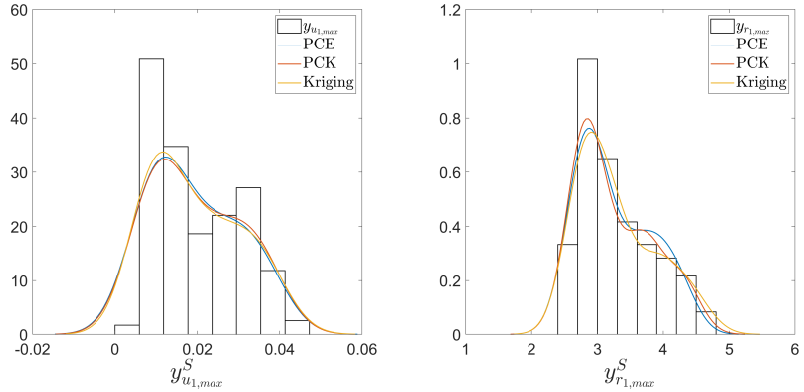


FIGURE 7.9: Convergence of PCE-based moments estimates of (a) mean, (b) standard deviation and (c) CV of QoI $r_{1,max}$.

and Kriging converge to the analytical values computed from the PCE coefficients. From Figure 7.11 it can be appreciated that the most sensitive input variables for both QoI are the period T_p of the velocity pulse and the stiffness K_1 of the left rotational spring. This outcome can be intuitively interpreted as the range of the velocity-pulse period T_p was chosen as such to overlap with the range of periods of the first bending mode of the hybrid model at the given bounds of the rotation springs' stiffness. Additionally, the QoI correspond to the hybrid model responses measured on the left



(a) (b)

FIGURE 7.10: Comparison between histogram measurements and corresponding surrogate predictions for (a) $u_{1,max}$ and (b) $r_{1,max}$.

support of the beam. Since the rotation spring stiffness K_1 is a boundary condition at the left support, it is anticipated to affect more the QoI than the rotation spring's stiffness at the right support, namely the K_2. Furthermore, from the fact that there exists only a small difference between pairs of first-order and total Sobol' indices, it can be perceived that the effect from high-order interactions between the input variables is negligible.

7.4.2 *Case study 2: virtual HS of a prototype motorcycle*

7.4.2.1 *Problem Formulation*

The hybrid model of CS2 is a prototype motorcycle. The hybrid model and the driving scenario of this case study coincide with the case study of Chapter 5. The same holds for the time integration scheme used to conduct the HSs. Therefore, the presentation of the hybrid model architecture is not repeated here and the reader should consult Section 5.3 for a detailed description.

In CS2, the input variables for GSA represent parameter variation of the hybrid model. The selected input variables with their assigned probability distributions are reported in Table 7.2. From these distributions, 200 sample points were generated using the LHS method [104] and accordingly 200 HS were conducted, utilizing in each iteration a different sample set. The

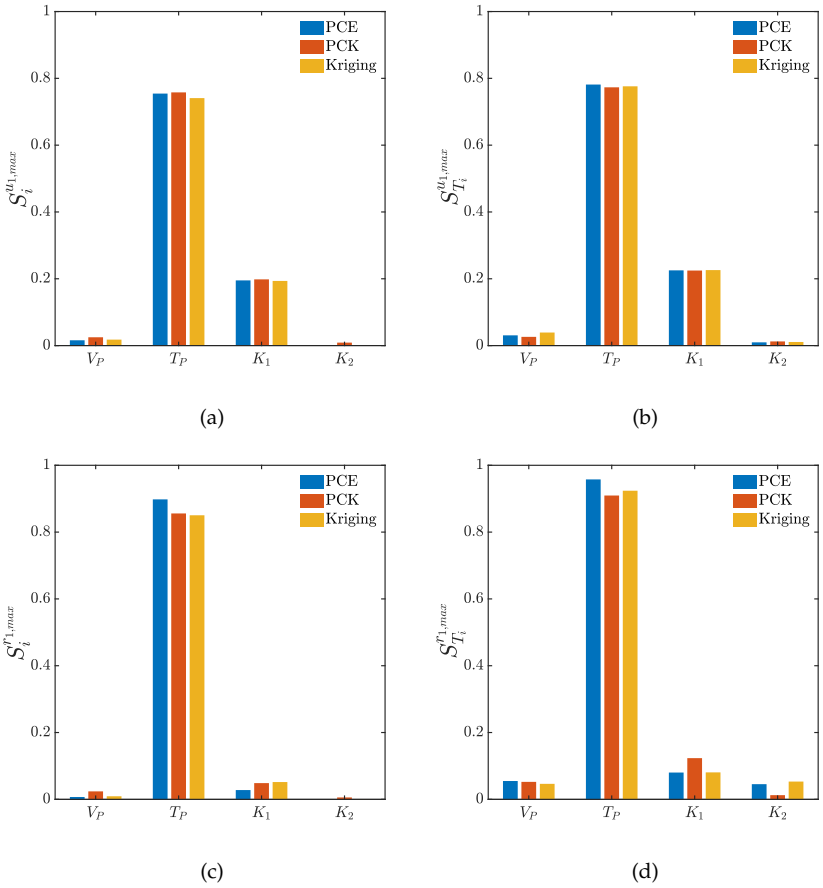


FIGURE 7.11: First-order and total Sobol' indices for (a-b) $u_{1,max}$ and (c-d) $r_{1,max}$.

bounds of the input variables were selected to reflect a range of possible parameter variations of the corresponding motorcycle components [115, 116, 117, 192]. The QoI of the hybrid model in CS2 are the max and mean value of the motorcycle velocity, namely v_{max} and v_{mean}, respectively. Both the latter are expressed in km/h. Therefore, the ED for this case study follows:

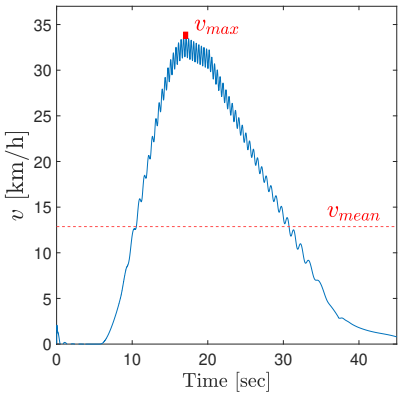


FIGURE 7.12: Sample time history response of CS2 hybrid model. The QoI are highlighted in red color.

$$\begin{aligned} \mathbf{X} &= \{K_{rt}, Z_{rt}, K_{ft}, Z_{ft}, K_{rs}, Z_{rs}, K_{fs}, Z_{fs}, M, J, E\mu, B\mu\} \\ \mathbf{Y} &= \{v_{max}, v_{mean}\}. \end{aligned} \tag{7.31}$$

Figure 7.12 illustrates a sample HS time history response of the hybrid model, indicating the QoI for CS2, which are highlighted in red color. Using the ED of Eq. (7.31), a PCE, a Kriging and a PCK surrogate is developed for each QoI, as described in Sections 5.2.1, 7.2.1 and 7.2.2.

7.4.2.2 *Case study 2 results*

Figure 7.13 displays the convergence of the LOO errors of the PCE, PCK and Kriging surrogates for both QoI. The LOO errors are computed using EDs of increasing size from 15 to 200 samples. From Figure 7.13, it can be appreciated that the LOO errors of PCE and PCK are quite close, while the error of Kriging is slightly larger. Yet, all the errors are negligibly small as they converge to values in the magnitude of 10^{-3} or smaller. The satisfying performance of the surrogates can be also confirmed from Figure 7.14, which compares the measurements of the two QoI with the respective predictions from the corresponding PCE, PCK and Kriging surrogates. In more detail, for each QoI, the surrogates are trained on 150 samples of the ED and the remaining 50 samples are used to validate them.

Param.	Prob. Distrib.	Mean Value	Stand. Dev.	CV (%)	Parameter Description	Units
K_{rt}	Unif.	58570	11714	20	Vertical stiffness rear tire	$\frac{N}{m}$
Z_{rt}	Unif.	11650	3495	30	Vertical damping rear tire	$\frac{Ns}{m}$
K_{ft}	Unif.	25000	5000	20	Vertical stiffness front tire	$\frac{N}{m}$
Z_{ft}	Unif.	2134	640.2	30	Vertical damping front tire	$\frac{Ns}{m}$
K_{rs}	Unif.	125000	25000	20	Stiffness rear suspension	$\frac{N}{m}$
Z_{rs}	Unif.	10000	3000	30	Damping rear suspension	$\frac{Ns}{m}$
K_{fs}	Unif.	19000	3800	20	Stiffness front suspension	$\frac{N}{m}$
Z_{fs}	Unif.	1250	375	30	Damping front suspension	$\frac{Ns}{m}$
M	Unif.	300	15	5	Motorcycle mass	Kg
J	Unif.	0.0115	0.0023	20	Engine moment of inertia	Kgm^2
E_μ	Unif.	0.0012	0.00018	15	Engine coefficient of viscous friction	$\frac{Nm}{rev/min}$
B_μ	Unif.	0.13	0.0195	15	Brake coefficient of viscous friction	$\frac{Nm}{rad/s}$

TABLE 7.2: Input variables of the hybrid model for CS2 and their characteristics.

In addition, Figures 7.15 and 7.16 depict the convergence plots of the PCE-based moments estimates, i.e. mean, standard deviation and CV values, for both QoI. Recall that the PCE-based moments estimates can be computed from the PCE coefficients as post-process at no additional computational cost. However, the latter is not a feature of Kriging or PCK. Thus Figures 7.15 and 7.16 do not include moments estimates from these two surrogates. Nevertheless, as in CS1, to further evaluate the performance of the developed surrogates, Figure 7.17 compares the prediction PDF of each surrogate with the histograms of the measurements for each QoI. The surrogate predictions are evaluated based on 200 random samples, generated from the distributions described in Table 7.2 using the LHS method.

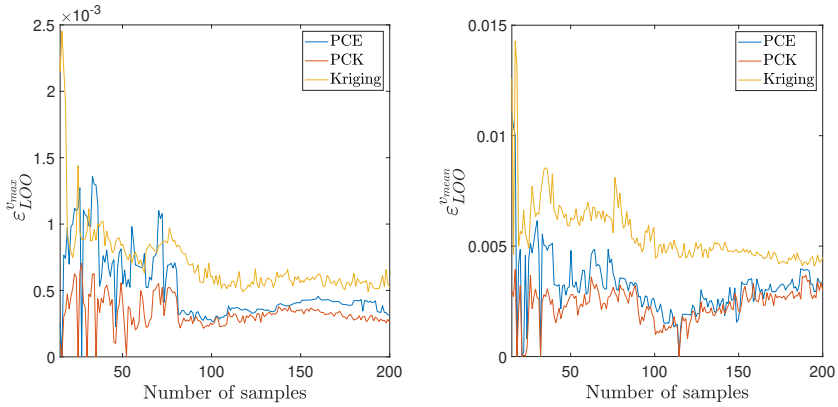


FIGURE 7.13: Convergence of LOO error estimates of (a) $\nu_{\max}$ and (b) ν_{mean} surrogates.

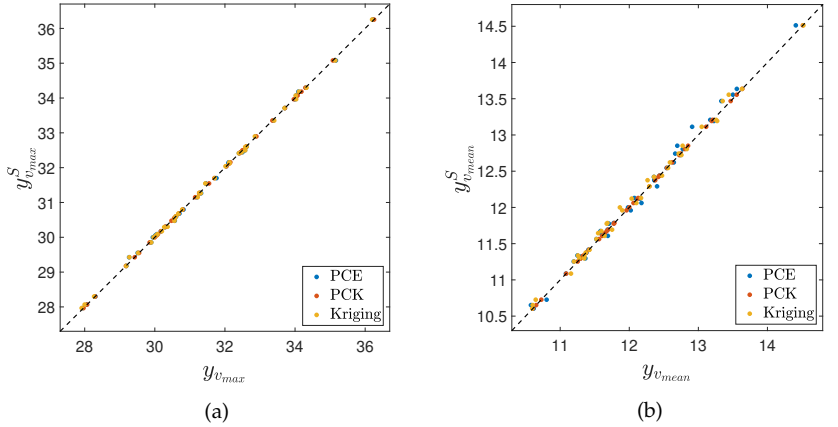


FIGURE 7.14: Comparison between QoI measurements and corresponding surrogate predictions for (a) $\nu_{\max}$ and (b) ν_{mean}.

Accordingly, to CS1, once the PCE, PCK and Kriging surrogates are developed, the Sobol' indices can be computed as described in Sections 7.3.3.1 and 7.3.3.2. Figure 7.18 reports the first-order and total Sobol' indices for

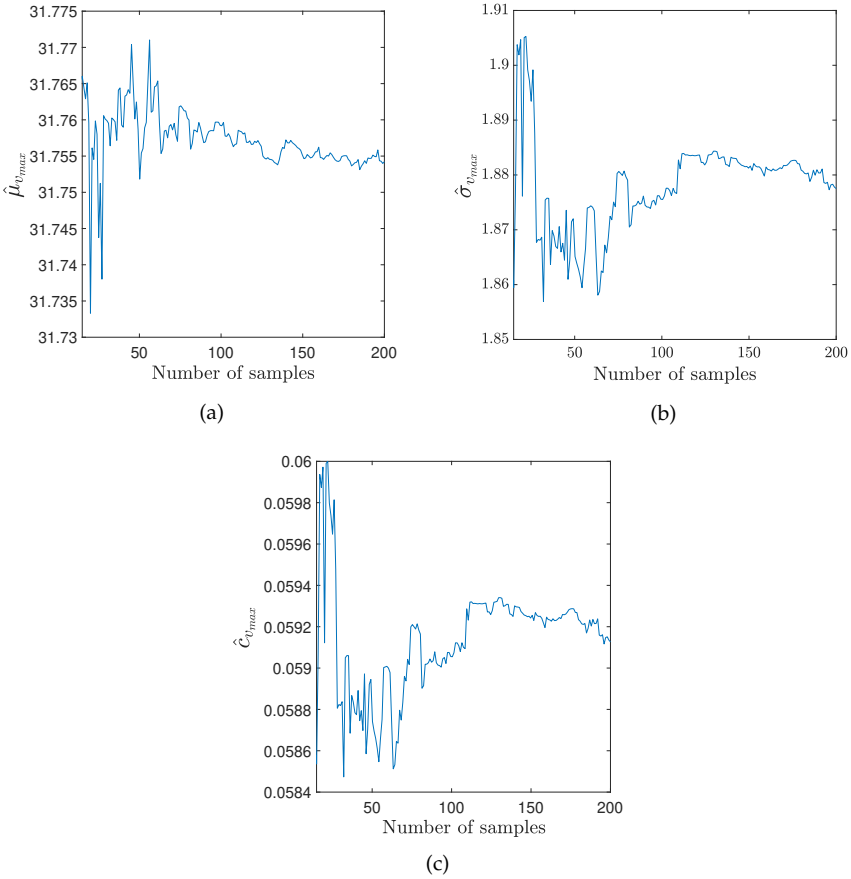


FIGURE 7.15: Convergence of PCE-based moments estimates of (a) mean, (b) standard deviation and (c) CV of QoI v_{max}.

both QoI v_{max} and v_{mean}, computed using the trained PCE, PCK and Kriging surrogates. As in CS1, both first-order and total Sobol' indices computed utilizing the different examined surrogates are close with each other. From Figure 7.18 it can be stated that the input variables related to the motorcycle mass M as well as to the engine coefficient of viscous friction $E\mu$, are the most sensitive variables for both QoI. In particular, for the QoI related to the maximum motorcycle velocity, namely v_{max}, the motorcycle mass has a greater effect than the engine friction coefficient. Whereas, for the QoI

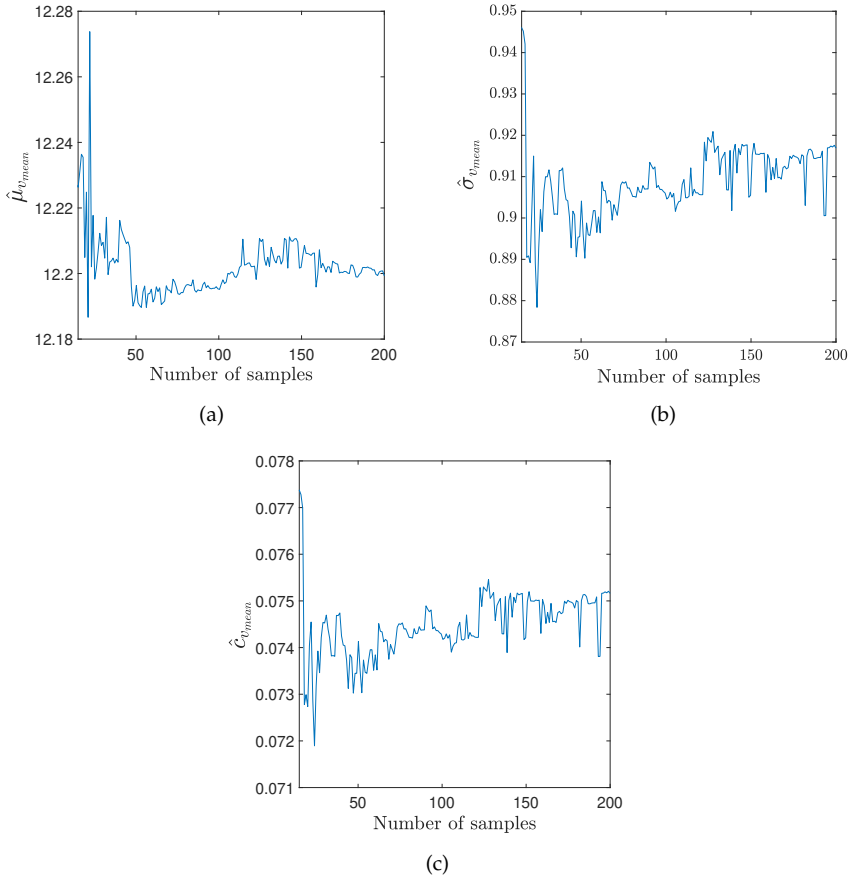
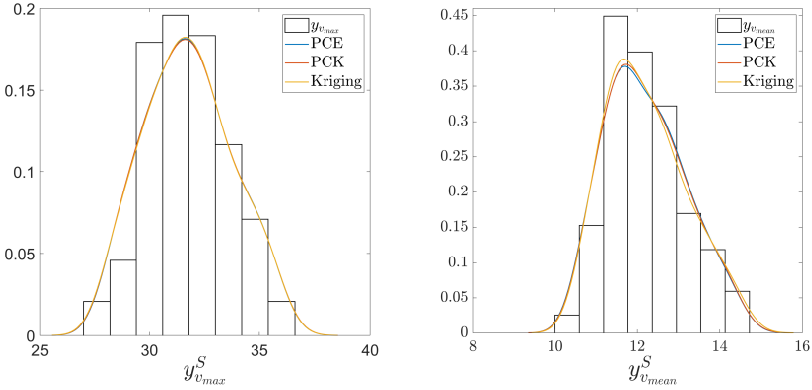


FIGURE 7.16: Convergence of PCE-based moments estimates of (a) mean, (b) standard deviation and (c) CV of QoI v_{mean}.

related to the mean motorcycle velocity, namely v_{mean}, the engine friction coefficient contributes more than the motorcycle mass. On the contrary, the remaining input variables of Table 7.2, do not affect significantly the selected QoI. Furthermore, since the first-order and total Sobol' indices for both QoI are quite close, it can be noticed that the effect from high-order interactions between the input variables is negligible.

Finally, as reported in Section 7.4.1.3, in CS1, convergence of the LOO errors can be observed for ED size larger than 50 samples. Nevertheless, in



(a) (b)

FIGURE 7.17: Comparison between histogram measurements and corresponding surrogate predictions for (a) $v_{\max}$ and (b) v_{mean}.

CS2 acceptably small LOO errors can be observed for ED of smaller size ($\approx$ 20 samples). This difference lies in the fact that in CS2 a virtual HS case study is employed and thus no measurement noise is introduced and propagated throughout the hybrid model. Recall that in CS1 (Section 7.4.1.3) the measurement noise had a significant negative effect on the convergence of the developed PCE. The difference in measurement noise can be also observed by comparing the sample time history responses of the hybrid models in CS1 and CS2, Figures 7.5 and 7.12, respectively.

The development and implementation of the surrogate modeling, as well as the GSA, for both case studies, were performed using the UQLab software framework developed by the Chair of Risk, Safety and Uncertainty Quantification in ETH Zurich [160].

7.5 Conclusions

The uncertainty quantification framework, initially outlined in Chapter 6, was presented in this chapter in more detail and it was applied to two case studies. The presented framework focuses on GSA using Sobol' indices. The latter are computed utilizing PCE, PCK and Kriging surrogate models. The goal of GSA is to quantify the effect of input variables on generic response QoI of the hybrid model. Typically, large datasets are required to perform GSA. Nonetheless, evaluation of the dynamic response of either an

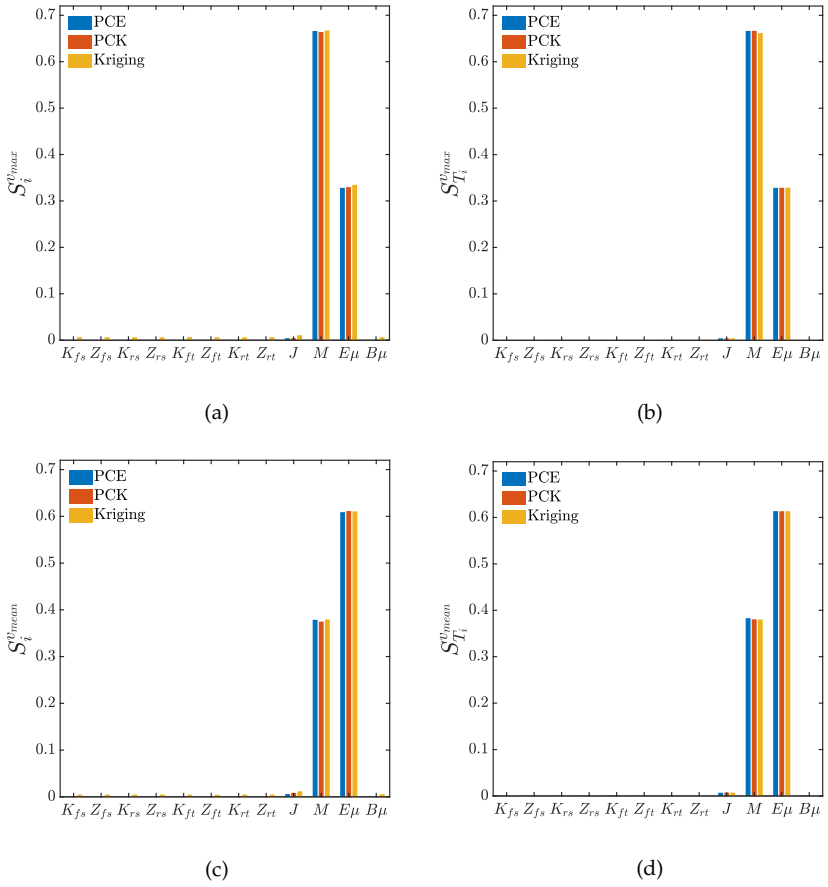


FIGURE 7.18: First-order and total Sobol' indices for (a-b) $\nu_{\max}$ and (c-d) ν_{mean}.

experimental prototype or a highly complex computational model multiple times is not reasonably affordable. However, this burden can be alleviated by developing surrogates of the hybrid model output QoI, whose response can be evaluated numerous times in a Monte Carlo simulation fashion at a much lower computational cost. Especially for the case that a PCE of the output QoI is developed, the Sobol' indices can be analytically computed from the PCE coefficients, omitting further Monte Carlo simulations.

To validate the presented framework two case studies were addressed; an experimental HS (CS1) and a virtual HS (CS2). As an overall remark,

due to the presence of measurement noise in CS1, the LOO error of the corresponding surrogates in CS1 was not converging to zero asymptotically. In addition, more samples were needed to achieve a negligibly small validation error of the surrogate than in CS2, since in the latter the hybrid model responses were noise-free and thus error convergence to zero was achieved with significantly fewer samples. In more detail, stable surrogates and thus stable Sobol' indices were attained with 50 HS evaluations for the CS1, whereas with as few as 20 for the CS2. Nonetheless, the stable plateau values of the CS1 indicate that the examined surrogates act as a powerful denoiser.

In addition, the work presented in this chapter serves as a survey of performing GSA with Sobol' indices utilizing different surrogate modeling methods. Comparisons of the performance of each developed surrogate as well as potential differences in the Sobol' indices obtained from each surrogate are made to highlight possible deviations. Results demonstrate the excellent accuracy of each surrogate developed, while their validation error is similar. Additionally, the Sobol' indices obtained from each surrogate are almost identical for every examined QoI.

GSA enables HS to be advanced into a simulation tool for investigating the structural sensitivity of the examined prototype structure to parameter variations and to the presence of uncertainties. The goal of GSA within HS is to unveil the inner workings of the prototype structure under consideration, providing us with quantitative information of what influences what and how much. Such information is valuable not only for future planning of additional HS campaigns for model V&V but also in the context of modeling and analysis choices in the overall structural design process.

Chapter 8

Global sensitivity analysis of hybrid models with stochastic physical substructures

The material presented in this chapter is published in the Frontiers in Built Environment journal [193].

8.1 Introduction

In Chapter 7, an uncertainty quantification framework was presented, focusing on performing global sensitivity analysis (GSA) with Sobol' indices using different surrogate modeling techniques. The goal was to reveal *what influences what and how much* within the HS: This entails uncovering the inner workings of the hybrid model in both epistemic and aleatory sense. In that study, the substructures of the hybrid model, and specifically the respective physical substructures (PS), were treated as deterministic, that is, aleatory uncertainty was neglected by assuming that nominally identical specimens have identical responses plus some negligible measurement noise. This assumption, however, is still far from a realistic scenario in structural testing. Nominally identical specimens are, in practice, never actually identical. Also, some sources of loading exerted through the PS is inherently stochastic, e.g. fire or hydrodynamic loading. As a result, uncertainty, both aleatory or epistemic, always affects the PS structural behavior and hence the hybrid model's dynamic response.

This chapter extends the framework presented in Chapter 7 to the case of PS with non-deterministic behavior. Similarly to the original framework, the idea is to surrogate the hybrid model response as a function of the input variables that can be controlled by the experimenter and originate from numerical substructures and loading. However, latent variables that do not appear in the input parameter vector make the hybrid model response stochastic. To account for the latter, the generalized lambda model (GLAM) recently developed by Zhu and Sudret [194, 195] is used here to directly surrogate the PDF of the response QoI. This is achieved by means of the family of generalized lambda distributions (GLD), which are suitable to approximate a wide class of distributions commonly found in engineering contexts [196]. The parameters of the GLD are cast as functions of the input variables of the hybrid model and approximated via polynomial

chaos expansion (PCE) [137, 142]. Normally, several repetitions of every single sample of the inputs variables, i.e. replications, would be required to replicate the response of a stochastic hybrid model. However, acquiring such repetitions would be impossible in a HS. Instead, by using the GLAM presented in [195] this problem can be solved, as the GLAM can be computed in a non-intrusive manner and does not require repeated HS for a single sample of input variables. For these reasons, the GLAM is well-suited to surrogate the response of a *stochastic hybrid model*. Variance-based GSA is uniquely defined for deterministic emulators [163]. In the context of stochastic emulators, three alternative variants of Sobol' indices are discussed in [197], namely *classical*, *QoI-based* and *trajectory-based* Sobol' indices. In this work, QoI-based, and in particular quantile-based, Sobol' indices are used for the GSA of the hybrid model response QoI, as explained in Section 8.3

The effectiveness of the proposed framework is demonstrated for a case study encompassing a 3-DoF stochastic hybrid model subjected to mechanical and thermal loading. Specifically, thermal loading is experimentally exerted on the PS so that temperature fluctuations (out of control of the experimenter) entail a stochastic response of the hybrid model.

This chapter is organized as follows. Section 8.2 introduces the GLAM, while Section 8.3 describes extensions of the Sobol' indices to account for the random nature of stochastic hybrid models. Section 8.4 presents a case study used to test the framework and discusses the obtained results of the GSA for the stochastic hybrid model response. Finally, Section 8.5 presents the overall conclusions of this chapter.

8.2 Generalized lambda model

Following the notation of Chapter 7, let a random variable vector $\mathbf{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\} \in \mathcal{D}_{\mathbf{X}} \subset \mathbb{R}^N$, where $\mathcal{D}_{\mathbf{X}}$ denotes the range of definition of $\mathbf{X}$, represent the input variables of a hybrid model. Nevertheless, as now hybrid model are conceived as stochastic, due to the random nature of their response, for a given set of variables $\mathbf{X}$, the corresponding QoI $Y(\mathbf{X})$ is a random variable rather than a deterministic value. This is because some latent variables $\boldsymbol{z}$ cannot be identified or measured in the process, which makes it impossible to include all the relevant variables in $\mathbf{X}$. Therefore, a stochastic hybrid model can be expressed as a mapping of:

$$\begin{aligned} \mathcal{M}^s : \mathcal{D}_{\mathbf{X}} &\subset \mathbb{R} \\ \mathbf{X} &\mapsto \mathcal{M}^s(\mathbf{X}, \mathbf{Z}). \end{aligned} \tag{8.1}$$

The latent variables z are grouped into a random vector $\mathbf{Z}$, representing the stochasticity of the hybrid model. Note that for a fixed realization of $\mathbf{Z} = z_0$, the output $Y(\mathbf{X})$ would become a deterministic function of $\mathbf{X}$. Such functions correspond to the so-called *trajectories*, where a single z_0 value represents a single trajectory $\mathbf{X} \mapsto \mathcal{M}^s(\mathbf{X}, z_0)$ [197]. On the contrary, with a fixed $\mathbf{X}$ and $\mathbf{Z}$ varying according to some probability distribution, the hybrid model output $Y(\mathbf{X})$ remains random.

The input variables x are treated as random, modeled by known probability distributions and grouped into a random vector $\mathbf{X} = \left\{x^{(1)}, \ldots, x^{(N)}\right\}$. $\mathbf{X}$ is characterized by its joint distribution with the PDF denoted by $f_{\mathbf{X}}$. Furthermore, we assume that X_i's are mutually independent, and thus the joint PDF is the product of marginal PDFs, i.e. $f_{\mathbf{X}}(x) = \prod_{i=1}^{N} f_{X_i}(x_i)$ with f_{X_i} being the marginal PDF of the i-th variable.

For a given set of input variables $\mathbf{X}$, the QoI $Y(\mathbf{X})$ is a random variable characterized by an unknown conditional probability distribution. Therefore, representation of the stochastic behavior of a hybrid model consists in estimating the response distribution for any variable $\mathbf{X} \in \mathcal{D}_{\mathbf{X}}$. However, one simulation for $\mathbf{X}$ does not provide the whole probability distribution but rather a single realization of $Y(\mathbf{X})$. Hence, it is usually necessary to repeatedly conduct experiments for the same $\mathbf{X}$, i.e. replications, to have enough insight into the resulting hybrid model response probability distribution. This quickly becomes intractable when the number of x's to be investigated increases. To alleviate the burden, surrogate models can be constructed to emulate the stochastic behavior of a hybrid model. Once a surrogate model is constructed, we can perform further analysis of the hybrid model response at a low cost, namely the GSA.

The simplest surrogate model of a stochastic hybrid model involves additive Gaussian noise:

$$Y^s(\mathbf{X}) = h(\mathbf{X}) + Z, \quad Z \sim \mathcal{N}(0, \sigma^2). \tag{8.2}$$

To build such a surrogate, one needs to estimate the mean function h and the noise variance σ^2. In this case, PCE, Kriging and PCK can be directly applied. However, Eq. (8.2) can be rather restrictive. To cover a wider group of problems, the recently developed statistical model called the *generalized lambda model* (GLAM) [194, 195] is chosen to be employed.

A GLAM assumes that the probability distribution of hybrid model response QoI $Y(\mathbf{X})$ can be approximated by a generalized lambda distribution (GLD) [196]. The latter is a highly flexible four-parameter distribution family, which is able to approximate many common distributions such as normal, lognormal, uniform and extreme value distributions. A GLD is

defined by its quantile function $Q(u) = F^{-1}(u)$, where $F(u)$ corresponds to its cumulative distribution function (CDF). In the presented work, GLD of the Freimer–Kollia–Mudholkar–Lin family [198] are utilized and admit:

$$Q(u;\boldsymbol{\lambda}) = \lambda_1 + \frac{1}{\lambda_2}\left(\frac{u^{\lambda_3}-1}{\lambda_3} - \frac{(1-u)^{\lambda_4}-1}{\lambda_4}\right), \tag{8.3}$$

where $u \in [0,1]$ and $\boldsymbol{\lambda} = \{\lambda_l : l = 1,\dots,4\}$ are the four distribution parameters. More precisely, λ_1 is the location parameter, λ_2 is the scaling parameter, and λ_3 and λ_4 are the shape parameters. To have valid quantile functions, λ_2 should be positive. From Eq. (8.3), the associated PDF of the random variable Y follows:

$$f_Y(y;\boldsymbol{\lambda}) = \frac{\lambda_2}{u^{\lambda_3-1}+(1-u)^{\lambda_4-1}}\,\mathbb{1}_{[0,1]}(u), \text{ with } u = Q^{-1}(y;\boldsymbol{\lambda}), \tag{8.4}$$

where $\mathbb{1}_{[0,1]}$ is the indicator function. From the above equation, it is clear that evaluating the PDF for a particular y requires solving numerically the equation $u = Q^{-1}(y;\boldsymbol{\lambda})$.

Under this setup, varying $\boldsymbol{x}$ will lead to $Y(\boldsymbol{x})$ following a GLD with different distribution parameters $\boldsymbol{\lambda}$. In other words, λ_l's are functions of $\boldsymbol{x}$, which allows us to express Y as:

$$Y(\boldsymbol{x}) \sim \mathrm{GLD}\left(\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x}), \lambda_3(\boldsymbol{x}), \lambda_4(\boldsymbol{x})\right). \tag{8.5}$$

Recall that the input variables $\boldsymbol{x}$ are modelled as independent random variables $\boldsymbol{X}$ with joint PDF $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{M} f_{X_i}(x_i)$. Under appropriate assumptions [199], each component of $\boldsymbol{\lambda}(\boldsymbol{x})$ admits a PCE representation following:

$$\begin{aligned}
\lambda_l(\boldsymbol{x}) &\approx \lambda_l^{\mathrm{PCE}}(\boldsymbol{x};\boldsymbol{c}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}_l} c_{l,\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}), \quad l = 1,3,4,\\
\lambda_2(\boldsymbol{x}) &\approx \lambda_2^{\mathrm{PCE}}(\boldsymbol{x};\boldsymbol{c}) = \exp\left(\sum_{\boldsymbol{\alpha}\in\mathcal{A}_2} c_{2,\boldsymbol{\alpha}}\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})\right),
\end{aligned} \tag{8.6}$$

where $\{\boldsymbol{\Psi}_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}^N\}$ is a basis of multivariate polynomials that are mutually orthogonal with respect to the probability measure of $\boldsymbol{X}$, $\boldsymbol{\alpha}$ is a multi-index that identifies the polynomial degree in each of the input variables, $\mathcal{A}_l \subset \mathbb{N}^N$ is a truncated set defining a finite set of basis functions for $\lambda_l(\boldsymbol{x})$ and $\boldsymbol{c} = \{c_{1,\boldsymbol{\alpha}},\dots,c_{l,\boldsymbol{\alpha}}\}$ denotes the associated PCE coefficients

(see Section 5.2.1 for more details on PCE[1]). Note that the PCE for $\lambda_2(\boldsymbol{x})$ is built on the logarithmic transform to ensure that $\lambda_2^{\text{PCE}}(\boldsymbol{x})$ is always positive. Combination of Eq. (8.5) with Eq. (8.6) results in the generalized lambda surrogate model:

$$Y^{\text{GLAM}}(\boldsymbol{x}) \sim \text{GLD}\left(\lambda_1^{\text{PCE}}(\boldsymbol{x};\boldsymbol{c}), \lambda_2^{\text{PCE}}(\boldsymbol{x};\boldsymbol{c}), \lambda_3^{\text{PCE}}(\boldsymbol{x};\boldsymbol{c}), \lambda_4^{\text{PCE}}(\boldsymbol{x};\boldsymbol{c})\right). \quad (8.7)$$

To build a GLAM, we need to determine the associated coefficients $\boldsymbol{c}$. As mentioned above, typically for developing stochastic surrogate models several replications are required. Recall that replications are obtained by evaluating the response of the random QoI $\boldsymbol{Y}(\boldsymbol{X})$ with $\boldsymbol{X}$ fixed to constant values. Nevertheless, generation of replications would result in a large number of experiments. Therefore, to avoid the need for replications, a method developed by Zhu and Sudret [195] which combines feasible generalized least-squares with maximum conditional likelihood estimations, is utilized. Firstly, a set of realizations of the input random vector $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right\}$ is generated, called the experimental design (ED). For each point of the ED, a HS is conducted and the corresponding QoI are collected into $\boldsymbol{Y} = \left\{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N)}\right\}$. Note that each HS may correspond to a different realization of the latent variable $\boldsymbol{Z}$, which does not need to be explicitly known in the analysis. In a second step, the $\boldsymbol{c}$ coefficients are estimated by maximizing the conditional likelihood, i.e. minimizing the negative log-likelihood, that is:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} L(\boldsymbol{c}) = \arg\min_{\boldsymbol{c}} \sum_{i=1}^{N} -\log\left(f^{\text{GLD}}\left(y^{(i)}; \boldsymbol{\lambda}^{\text{PCE}}\left(\boldsymbol{x}^{(i)};\boldsymbol{c}\right)\right)\right), \quad (8.8)$$

where f^{GLD} is the PDF of the GLD defined in Eq. (8.4). To solve the optimization problem, it is necessary to determine the support of $\boldsymbol{c}$, which is equivalent to finding the truncation set $\mathcal{A}_l$ for each λ_l^{PCE}. To this end, the hybrid-LAR algorithm [142] is plugged into the modified feasible generalized least-squares framework, described in [195]. The basis functions selected for these two functions are then used to represent $\lambda_1^{\text{PCE}}(\boldsymbol{x})$ and $\lambda_2^{\text{PCE}}(\boldsymbol{x})$, respectively. As λ_3 and λ_4 mainly control the PDF shape of a GLD, which is expected not to change much when $\boldsymbol{x}$ is changed, polynomials with low degree can be picked, namely 0 or 1, for $\lambda_3^{\text{PCE}}(\boldsymbol{x})$ and $\lambda_4^{\text{PCE}}(\boldsymbol{x})$. After

1 In Section 5.2.1, the PCE coefficients were denoted with $\boldsymbol{y}$. Without loss of generality, here the $\boldsymbol{c}$ notation is used instead.

specifying the basis functions, the associated GLAM is built by solving Eq. (8.4).

In this section, only the basic background of GLAM was covered. For a thorough review on GLD distributions and construction of the GLAM as well as for methods of estimating the distribution parameters $\boldsymbol{\lambda}$ of the GLAM, consult [194, 195].

8.3 Sobol' indices extensions for stochastic hybrid models

The Sobol' indices for deterministic hybrid models have been examined in Chapter 7. A short recap is presented here as well. As described in Section 7.3, for a random vector $\mathbf{X}$ with independent components, any deterministic mapping $Y = \mathcal{M}(\mathbf{X})$ with $\mathrm{Var}[Y] < +\infty$ can be decomposed as [174]:

$$\begin{aligned}\mathcal{M}^s(\mathbf{X}) &= \mathcal{M}_0 + \sum_{i=1}^{N} \mathcal{M}_i(x^{(i)}) + \sum_{1 \leqslant i \leqslant j \leqslant N} \mathcal{M}_{ij}(x^{(i)}, x^{(j)}) + \cdots + \mathcal{M}_{1,2,\ldots,N}(\mathbf{X}) \\ &= \mathcal{M}_0 + \sum_{\substack{\mathbf{u} \subset \{1,\ldots,N\} \\ \mathbf{u} \neq \emptyset}} \mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}),\end{aligned} \tag{8.9}$$

where $\mathcal{M}_0$ is a constant, and $\mathcal{M}_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}})$ is a function of the subset of input variables $\mathbf{x}_{\mathbf{u}}$ specified by $\mathbf{u} \neq \emptyset$, $\mathbf{u} \subset \{1, \ldots, N\}$. The elementary functions are defined by conditional expectations:

$$\mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}) = \sum_{\mathbf{v} \subset \mathbf{u}} (-1)^{|\mathbf{u}|-|\mathbf{v}|} \mathbb{E}\left[Y \mid \mathbf{X}_{\mathbf{v}} = \mathbf{x}_{\mathbf{v}}\right], \tag{8.10}$$

where $|\mathbf{u}|$ gives the cardinality of $\mathbf{u}$. The decomposition of Eq. (8.9) is unique and hence the total variance of $\mathcal{M}(\mathbf{X})$ decomposes as:

$$D = \mathrm{Var}\left[\mathcal{M}(\mathbf{X})\right] = \sum_{\substack{\mathbf{u} \subset \{1,\ldots,N\} \\ \mathbf{u} \neq \emptyset}} D_{\mathbf{u}}, \tag{8.11}$$

where

$$D_{\mathbf{u}} = \mathrm{Var}\left[\mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}})\right] = \mathbb{E}\left[\mathcal{M}_{\mathbf{u}}^2(\mathbf{X}_{\mathbf{u}})\right], \tag{8.12}$$

corresponds to the partial variance. The first- and higher-order are obtained from:

$$S_{\mathbf{u}} = \frac{D_{\mathbf{u}}}{D}, \tag{8.13}$$

while the total indices from:

$$S_{T_i} = 1 - S_{-i}, \tag{8.14}$$

where S_{-i} denotes the sum of all $S_{\mathbf{u}}$ of $\mathbf{u}$ but i.

Nevertheless, for the case of stochastic hybrid models, due to their random nature, decomposition similar to Eq. (8.9) is generally impossible. Therefore, it is necessary to represent a stochastic model by a deterministic function to obtain the associated Sobol' indices. Based on the choice of the deterministic representation, different extensions of the Sobol' indices can be used. In this regard, three Sobol' indices extensions are found in the literature, as described in [197].

The most straightforward approach is to include the latent variables $\mathbf{Z}$ within the input variables $\mathbf{X}$, as proposed by Iooss and Ribatet [200]. In that way, the latent variables are handled similarly with the input variables. This leads to the underlying (yet unknown in practice) deterministic model $\mathcal{M}^s(\mathbf{X}, \mathbf{Z})$, defined in Eq. (8.1). Decomposition of this function results in:

$$Y = \mathcal{M}^s(\mathbf{X}, \mathbf{Z}) = \mathcal{M}_0 + \sum_{\substack{\mathbf{u} \subset \{1,\dots,N\} \\ \mathbf{u} \neq \emptyset}} \mathcal{M}_{\mathbf{u}}(\mathbf{X}_{\mathbf{u}}) + \mathcal{M}_{\mathbf{Z}}(\mathbf{Z}) + \mathcal{M}_{\mathbf{X},\mathbf{Z}}(\mathbf{X}, \mathbf{Z}). \tag{8.15}$$

Since the definition of $\mathcal{M}_{\mathbf{u}}$ is the same as of Eq. (8.10), the Eq. (8.12) for $D_{\mathbf{u}}$ still holds and thus the Sobol' indices of this approach are referred to as *classical* Sobol' indices, as their extension is direct [197]. $S_{\mathbf{u}}$ can be determined by the statistical dependence between Y and $\mathbf{X}_{\mathbf{u}}$. The definition of the total index S_{T_i} requires including the interactive effect between $x^{(i)}$ and the latent variables $\mathbf{Z}$. However, interactions with $\mathbf{Z}$ cannot be determined by the response distribution but rather depend on the precise data generation process (i.e. how $\mathbf{Z}$ is present in the function $\mathcal{M}^s$). Since the latent variables $\mathbf{Z}$ are generally impossible to characterize and control in a real experiment, the total Sobol' indices cannot be assessed.

Another alternative approach for extending the Sobol' indices to account for stochastic models is the so-called *trajectory-based* Sobol' indices (see [197] for their mathematical formulation). Recall that a trajectory, in which $\mathbf{Z} = z_0$, is a deterministic function of $\mathbf{X}$. Nevertheless, in the trajectory-based Sobol' indices, it is necessary to control the latent variable $\mathbf{Z}$. However, this is not feasible in experimental testing as $\mathbf{Z}$ is unknown.

Alternatively to Eq. (8.15), certain summary quantities of the response random variable $Y(\mathbf{X})$ can be employed as a deterministic representation of $Y(\mathbf{X})$, resulting in the *QoI-based* Sobol' indices [197]. This is particularly helpful when the selected summary quantity itself is of interest. Typical quantities are mean $m(\mathbf{X}) = \mathbb{E}[Y(\mathbf{X})]$, variance $v(\mathbf{X}) = \mathrm{Var}[Y(\mathbf{X})]$ [200], and α-quantiles $q_\alpha(\mathbf{X})$ [201]. As these functions are well-defined as deterministic functions of $\mathbf{X}$ (since the effect of the latent variables $\mathbf{Z}$ has been marginalized), the associated Sobol' indices follow directly from Eq. (8.9). In the case study of Section 8.4, QoI-based Sobol' indices are chosen to be utilized. In particular, α-quantiles are selected as the respective summary QoI and hence the corresponding Sobol' indices are referred to as *quantile-based*. The selection of the QoI-based Sobol' indices instead of the classical or the trajectory-based is twofold. A first reason lies in the fact that quantile functions of the QoI for the presented case study is more of interest and secondly, for the computation of the respective indices no knowledge nor control of the latent variables $\mathbf{Z}$ is required, as this would be otherwise impossible.

A generalized lambda surrogate model emulates the response distribution of a stochastic model, which fully captures the statistical dependence between the input variables $\mathbf{X}$ and the QoI Y. Therefore, such a surrogate allows for evaluating both types of classical and QoI-based Sobol' indices. More precisely, either Monte Carlo simulations or PCE estimates can be used to compute the respective Sobol' indices, suchlike as explained in Sections 7.3.3.1 and 7.3.3.2 for the deterministic hybrid models. The detailed mathematical expressions to derive the respective Sobol' indices for the stochastic hybrid models can be found in [197].

Recall that the presented GSA framework assumes that the input parameters in $\mathbf{x}$ are statistically independent. Nevertheless, a generalized lambda surrogate model can emulate the response of a stochastic hybrid model even in the case of dependent input parameters [195]. This holds since the dependence of Y on the input parameters are not affected by the dependence within the input variables. Nevertheless, for the case of dependent input parameters, generalized Sobol' indices or alternative variance-based sensitivity analysis methods should be employed as described in [178, 202].

8.4 Case study: HS of a prototype structure

8.4.1 *Problem formulation*

The proposed GSA framework is validated using a stochastic 3-DoFs hybrid model subjected to both thermal and mechanical loading, as illustrated

in Figure 8.1. As can be appreciated from Figure 8.1a, the hybrid model consists of a simply-supported beam provided with two rotational elastic restraints. In this regard, u_1 and u_2 indicate the two rotational DoFs, while u_3 is the axial DoF. Figure 8.1b describes the substructuring of the hybrid model into PS and NS. The NS comprises the two elastic rotational restraints, the axial spring and the linear dashpot whereas the PS coincides with the beam element. Specifically, the axial spring is characterized by a constant stiffness of $K_3 = 8100 \times 10^3$ N/m. Two lumped rotational masses are defined by $J_1 = J_2 = 10$ kgm² while the lumped translational mass is defined by $M_3 = 5000$ kg. The linear dash-pot is characterized by $C_3 = 1129 \times 10^3$ Ns/m. The PS consists of an aluminum plate of 0.2×0.002 m rectangular cross-section and length $L = 0.47$ m. Accordingly, the cross-section of the plate is characterized by an area $A = 4 \times 10^{-4}$ m² and a moment of inertia $I = 66.67 \times 10^{-12}$ m⁴. The Young's modulus, density and thermal expansion coefficient of aluminum are $E = 69.5$ GPa, $\rho = 2700$ kg/m³, and $\alpha = 23 \times 10^{-6}$ °C⁻¹, respectively. Since in this specific case study HS are conducted using a testing time scale equal to 50, the PS mass (rotational and translational) does not contribute to the hybrid model inertia. For the sake of clarity, all equations and plots in the following refer to simulation time, which is virtual and 50 times slower than the wall-clock time.

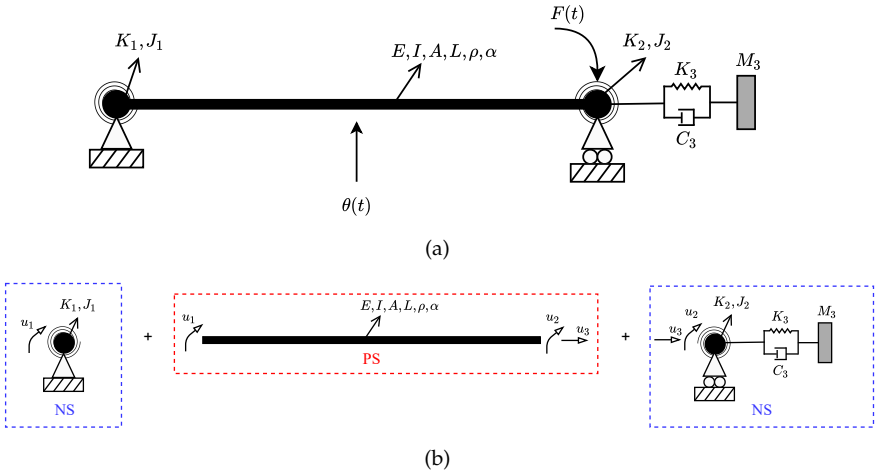


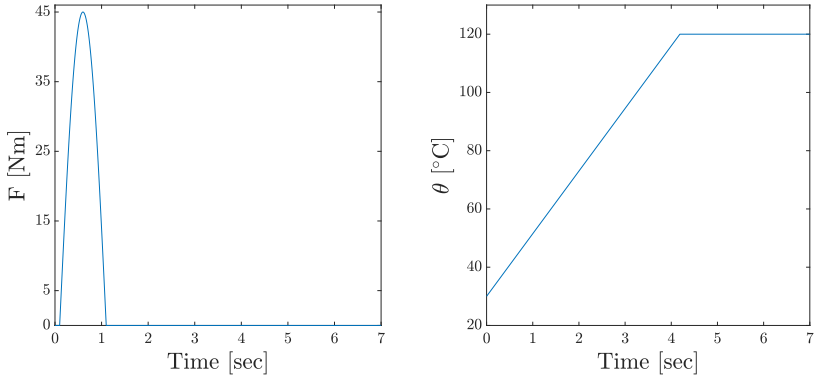
FIGURE 8.1: (a) The prototype structure of the case study and (b) its hybrid model.

Mechanical loading is supplied as a bending moment history $F(t)$ applied to the right rotational DoF u_2 while thermal loading, applied by a heating lamp, is defined by a ramp & hold temperature history $\theta(t)$. The expressions of both read:

$$F(t) = \begin{cases} F_{max} \sin(\frac{\pi(t-t_0)}{T}) \text{ , } t_0 \leqslant t \leqslant t_0 + T/2 \\ 0 \text{ , elsewhere} \end{cases} \tag{8.16}$$

$$\theta(t) = \begin{cases} \dot{\theta} t \text{ , } \theta(t) < \theta_{max} \\ \theta_{max} \text{ , elsewhere} \end{cases} \tag{8.17}$$

where F_{max} and T are peak value and period of the half-side bending moment pulse, respectively, applied to DoF u_2 with a time shift of $t_0 = 0.1$ s; $\dot{\theta}$ and θ_{max} are temperature rate and plateau, respectively characterizing the temperature history imposed to the PS. For the sake of this example, Figure 8.2a depicts the bending moment history computed for $F_{max} = 45$ Nm, $T = 1$ s. Similar, Figure 8.2b depicts the temperature history computed for $\dot{\theta} = 21.5$ °C/s and $\theta_{max} = 120$ °C.



(a) (b)

FIGURE 8.2: (a) bending moment history for $F_{max} = 45$ Nm, $T = 1$ sec and (b) temperature time history for $\dot{\theta} = 21.5$ °C/s and $\theta_{max} = 120$ °C.

Consistent with the motivations underlying the development of the GSA framework, the stiffness of the two elastic rotational springs, which play the role of boundary conditions to the PS, as well as the loading parameters, are selected as input variables for the surrogate modeling phase. In line with the procedure described in Sections 8.2 and 8.3, the input variables are described by independent uniform distributions, whose bounds are summarized in Table 8.1.

Parameter	Probability Distribution	Lower Bound	Upper Bound	Parameter Description	Units
K_1	Uniform	110.58	1105.80	Rotational stiffness left spring	$\frac{Nm}{rad}$
K_2	Uniform	110.58	1105.80	Rotational stiffness right spring	$\frac{Nm}{rad}$
F_{max}	Uniform	40.00	50.00	Bending moment pulse peak	Nm
T	Uniform	0.50	2.00	Bending moment pulse period	sec
θ_{max}	Uniform	100.00	130.00	Temperature plateau	°C

TABLE 8.1: Input variable intervals and assumed probability distributions.

It is important to remark that, in order to reduce the experimental effort required to validate the proposed framework, the hybrid model and loading excitation were designed such that the PS always remained in the linear response regime. As a result, HS were conducted using a single aluminum plate.

The response QoI selected for the GSA corresponds to the maximum absolute out-of-plane deflection of the tested aluminum plate, which is denoted as $u_{L,max}$.

8.4.2 *Hybrid simulation setup*

The 3-DoFs HS test rig used to conduct the HS for this case study is located in the IBK Structural Testing Laboratory at ETH Zurich and it is the same as the one presented in Section 7.4.1.2. However, adjustments in the hardware are made to enable exciting the prototype structure with thermal loading. For the ease of reading, the HS test rig is described in full here as well.

It corresponds to a stiff loading frame equipped with four electromechanical actuators and an infrared (IR) lamp module interfaced to an INDEL real-time computer. The 3-DoFs HS test rig is designed to test plate specimens with an approximate footprint of 200×500 mm and thickness varying

between 1 and 3 mm. Figure 8.3 illustrates the architecture of the HS setup, including a close-up view of the plate specimen accommodation. Two axonometric views of the 3-DoF test rig, consisting of the main hardware components are shown in Figure 7.4.

The moving parts of the test rig are colored in yellow, the plate specimen in brown and the fixed parts in gray. The latter are fixed to a reaction frame. In order to impose the u_1 and u_2 rotations, two rack-pinion systems (10) are installed along the vertical actuator y_1 and y_2 (1). The rack-pinion systems aim at transforming the commanded displacements from the actuators to rotational DoFs, applied to the short edges of the plate specimen (6) through aluminum clamps (3). The two horizontal actuators x_1 and x_2 control the position of the moving frame mounted on profiled rail guides (4) and correspond to the axial DoF u_3 of the plate specimen (6). A linear variable differential transformer (LVDT) measures the out-of-plane deflection at the mid-span of the plate specimen (labeled u_L in Figure 8.3). A Type K thermocouple installed at the center of the plate provides the feedback signal for the control of the IR lamp, which imposes the temperature history $\theta(t)$.

The GINLink bus connects actuator servo drivers INDEL SAC4, IR lamp and all DAQ modules to the real-time computer INDEL SAM4, which executes the HS software. The latter is developed in MATLAB/SIMULINK, compiled, and downloaded to the INDEL SAM4 from the Host-PC. At each simulation time-step, the HS software imposes displacements u_1, u_2 and u_3 to the plate specimen, the PS, reads the corresponding restoring forces r_1, r_2 and r_3 measured using force transducers, and solves the coupled equation of motion of the hybrid model. Also, the HS software generates the temperature command for the IR lamp. For the time integration scheme used in the HS of this case study, a fixed time-step of 1 msec was used. A comprehensive description of the scheme is reported in [41].

Forces were manually set to zero before starting each HS. An electric fan cools down the PS at the end of each test. Room temperature was quite stable and equal to 30 °C for the entire testing campaign.

8.4.3 *Results*

The response of the hybrid model described in Section 8.4.1 was evaluated using the HS setup described in Section 8.4.2 on 200 samples of the input parameter vector generated using Latin hypercube sampling (LHS) [104]. The resulting ED $\mathbf{X}, \mathbf{Y}$ was used for computing surrogate models. In this regard, Figure 8.4 reports out-of-plane and axial displacement histories obtained via HS for a single sample of input variables, where $u_{L,max}$,

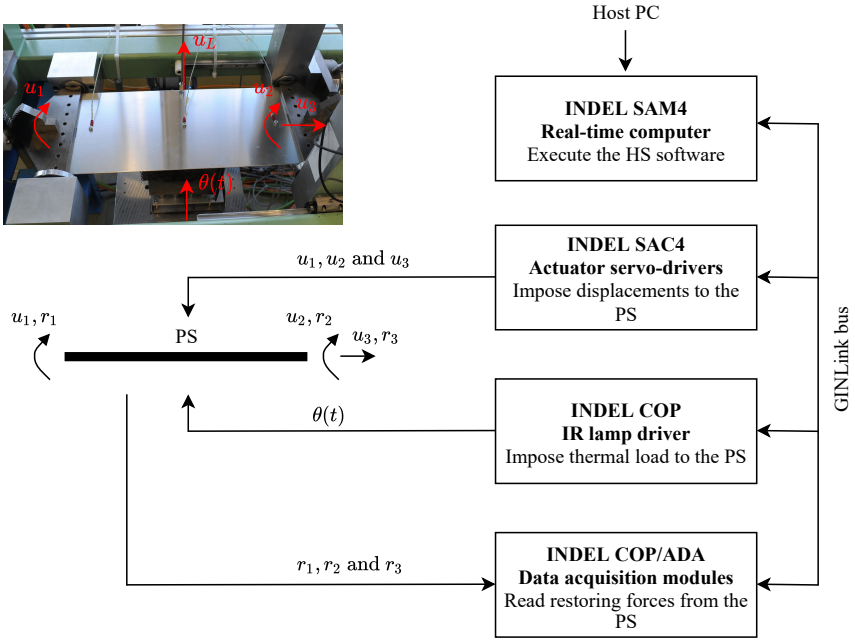


FIGURE 8.3: Architecture of the 3-DoFs HS test rig.

$u_{3,max}$, $u_{L,0}$ and $u_{3,0}$ scalar quantities are highlighted. Specifically, $u_{L,max}$ corresponds to the QoI (see Eq.(8.18)) and $u_{3,max}$ to the absolute maximum axial displacement while $u_{L,0}$ and $u_{3,0}$ indicate the initial position of the out-of-plane displacement and axial axes respectively, relative to the value measured during the first HS. Figure 8.5 describes the evolution of $u_{L,0}$ and $u_{3,0}$ scalar quantities over the entire experimental campaign.

8.4.3.1 *Drift observed in measurement data*

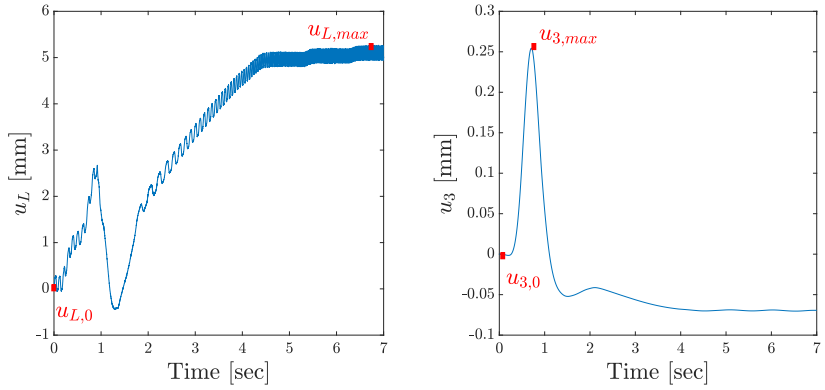
From Figure 8.5 it is clear that both $u_{L,0}$ and $u_{3,0}$ quantities have a constant drift, which results in a total accumulated out-of-plane and axial displacements of 3.0 mm and 0.3 mm, respectively. Such a drift can be reasonably ascribed to the cumulative slippage of plate fixtures produced by heating/cooling cycles. This drift occurs regardless of the type of analysis that the HS setup was used for, and hence it should be removed from the acquired raw data before any further post-processing. Accordingly, prior to the calculation of surrogate models, the effect of drift on $u_{L,max}$ was eliminated via linear detrending with respect to $u_{L,0}$. The detrended QoI is referred to as $\hat{u}_{L,max}$ and compared to original values in Figure 8.6. Notably, $u_{L,0}$ is independent of the parameters of the hybrid model since the initial position of the PS was set by zeroing actuator forces.

Consistent with the notation introduced in Sections 8.2 and 8.3, surrogate modeling was performed considering the following input parameter vector and response QoI:

$$\begin{aligned} \mathbf{X} &= \{K_1, K_2, F_{max}, T, \theta_{max}\}, \\ \mathbf{Y} &= \{\hat{u}_{L,max}\}. \end{aligned} \tag{8.18}$$

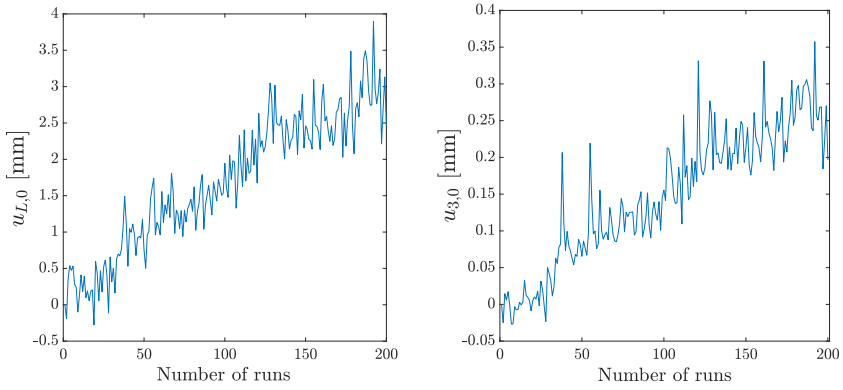
8.4.3.2 *Global sensitivity analysis results*

A GLAM of the hybrid model dynamic response QoI was computed as explained in Section 8.2. In this study, we set the candidate degrees up to 5 for λ_1^{PCE} and 3 for λ_2^{PCE}. In order to validate the GLAM, 10 repeated HS were performed for two validation ED points, namely 58 and 157, associated with different regions of the input parameter space and characterized by appreciably different QoI values. For each validation ED point, Figure 8.7 compares the GLAM prediction to the empirical distribution of the 10 related repetitions. It is observed that the GLAM correctly captures the empirical distribution of the QoI for both points. It is interesting to note that the computed GLAM model converged to zero-order polynomials for λ_2^{PCE},



(a) (b)

FIGURE 8.4: Sample time history response of the hybrid model with $K_1 = 224.454$ Nm/rad, $K_2 = 118.235$ Nm/rad, $F_{max} = 48$ Nm, $T = 0.618$ sec and $\theta_{max} = 116.73$ °C: (a) out-of-plane displacement, (b) axial displacement.



(a) (b)

FIGURE 8.5: Drift in the hybrid model response: (a) u_L and (b) u_3.

and that the two PDFs of Figure 8.7 are similar in shape, thus suggesting a homoscedastic stochasticity of the hybrid model, which was further verified using a PCE surrogate model. Specifically, a residual analysis was performed on the difference between the measured QoI and its PCE.

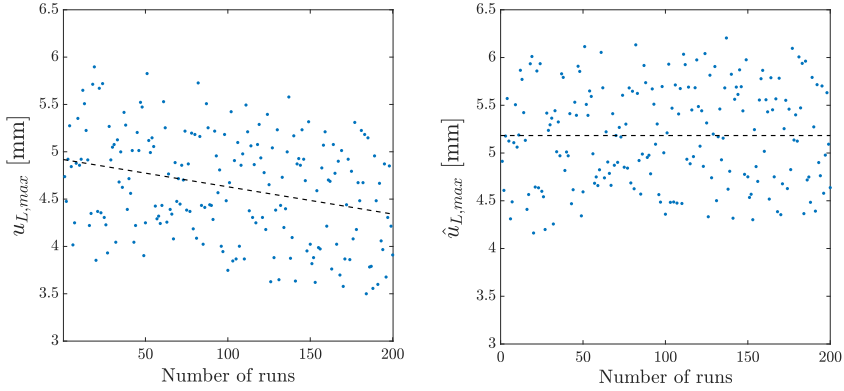


FIGURE 8.6: Effect of detrending on the QoI: (a) original values ($u_{L,max}$) and (b) values after detrending ($\hat{u}_{L,max}$). Dashed lines indicate a linear trend of data.

As highlighted in Chapter 7, in presence of noisy data, PCE is a powerful denoiser, which naturally provides a surrogate of the average model response $\mathbb{E}_Z[Y|\mathbf{x}]$. In this regard, the Tukey-Anscombe plot [203] of Figure 8.8a compares the PCE output to the corresponding residual for each sample of the ED. The Q-Q plot of Figure 8.8b compares the empirical quantiles of the residuals normalized to unit standard deviation to the theoretical values of a standard normal distribution $\mathcal{N}(0,1)$. The zero-average uniform scattering of residuals highlighted by the Tukey-Anscombe plot and the fairly good agreement between empirical and theoretical quantiles highlighted by the Q-Q plot confirms that the hybrid model response was affected by a Gaussian homoscedastic additive noise.

As reported in Section 8.3, only first- and higher-order Sobol' indices but not total Sobol' indices can be obtained from the GLAM of the QoI. Instead, first and total Sobol' indices can be computed for the QoI quantiles. Accordingly, Figure 8.9 provides first and total Sobol' indices of the 5, 50 and 95 % quantiles of $\hat{u}_{L,max}$. The results of the GSA indicate that the temperature plateau value θ_{max} is the most sensitive input parameter for the selected QoI. Additionally, the equal Sobol' indices values for each quantile unveil the homoscedastic response of the stochastic surrogate.

The development and implementation of the surrogate modeling, as well as the GSA, was performed using the UQLab software framework

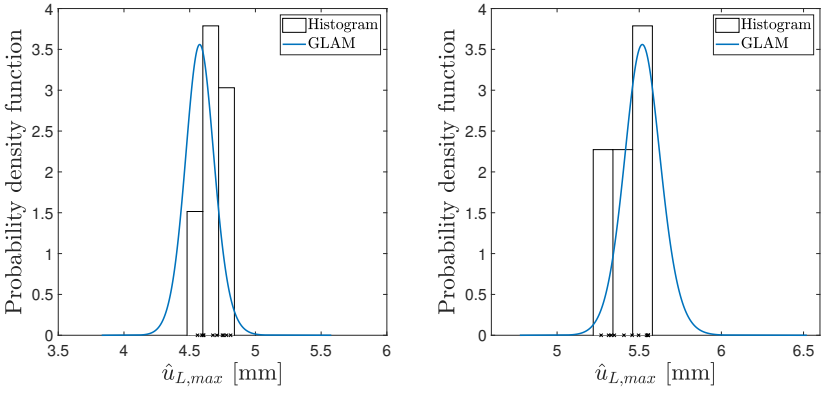


FIGURE 8.7: PDF of $\hat{u}_{L,max}$ predicted by the GLAM versus empirical distributions for ED points: (a) 58, and (b) 157. Cross markers denote the 10 related repetitions for each point.

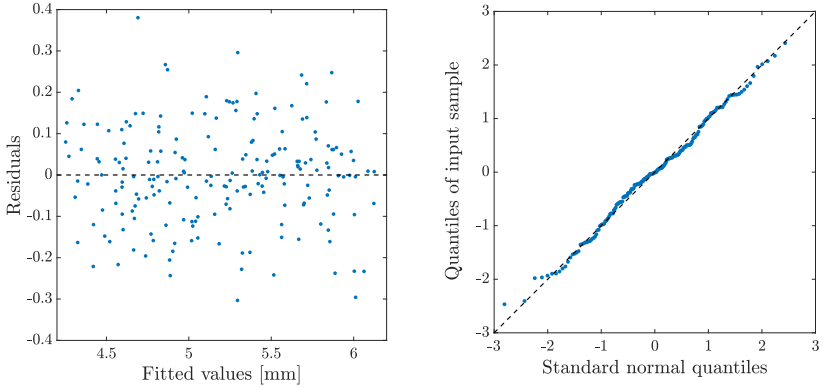
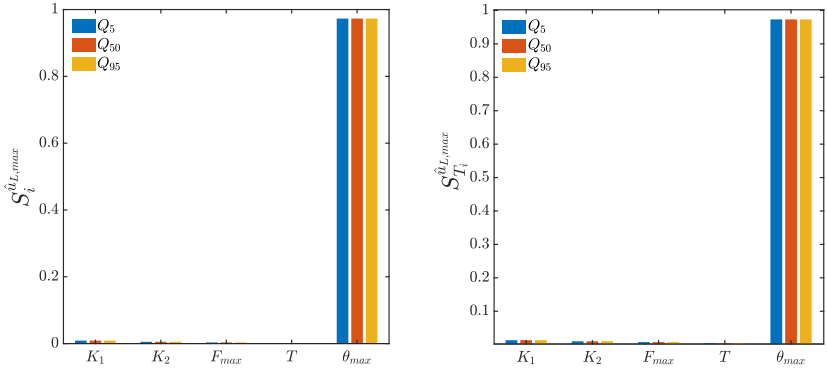


FIGURE 8.8: Analysis of QoI residuals with respect to PCE: (a) Tukey-Anscombe plot and (b) Q-Q plot.

developed by the Chair of Risk, Safety and Uncertainty Quantification in ETH Zurich [160].



(a) (b)

FIGURE 8.9: Sobol' indices of 5, 50 and 95 % quantiles of $\hat{u}_{L,max}$: (a) First order and (b) Total.

8.5 Conclusions

This chapter described a framework for GSA of stochastic hybrid models. A generalized lambda surrogate modeling technique recently developed by Zhu and Sudret [194, 195] is used to compute Sobol' indices for the quantiles of a response QoI. The idea of using surrogate modeling to enable global sensitivity studies with few expensive-to-evaluate hybrid simulations was already presented in Chapter 7. The novelty of this work lies in the extension of GSA to the case of hybrid models with stochastic PS, to cover the more realistic situation where the hybrid model response for two nominally identical PS is not repeatable.

The effectiveness of the proposed framework is demonstrated in an experimental application consisting of a hybrid model with five parameters and subjected to mechanical and thermal loading. A great advantage of the proposed framework is that the generalized lambda surrogate model does not require repeated evaluations of the same sample. The results of the demonstration study highlight that the stochasticity of the particular hybrid model under consideration is homoscedastic with respect to the hybrid model parameters. Accordingly, both first-order and total Sobol' indices of 5, 50, and 95 % quantiles are almost identical. Moreover, the temperature plateau value of the thermal loading is the most sensitive parameter for the selected response QoI. The outcome of the experiment demonstrates

the effectiveness of the proposed GSA framework in revealing the inner workings of the hybrid model.

Chapter 9
Conclusions and future research directions

9.1 Conclusions

9.1.1 *Outlook*

Prior to any engineering decision, it is common to consult a high-fidelity model in order to give insight into the performance of the examined system and to account for any potential risks included in its design. The development of trusted models for systems that operate in a dynamic environment is a process known as *dynamic virtualization*. An essential step in the development of such models is the so-called *verification and validation*. A vital element of validation itself is experimental testing. To date, the *quasi-static* and *full-scale dynamic testing* are two widely used experimental techniques to investigate the response of a structural system and to further validate the corresponding model. Nevertheless, the quasi-static testing fails to adequately capture the dynamic response of a structural system subjected to dynamic excitation. On the contrary, this is not the case for full-scale dynamic testing. In the latter, however, the cost associated with constructing the full-scale physical specimen as well as with providing the required experimental capacity to conduct such a type of test is often quite large.

A more recently developed experimental technique is the so-called *hybrid simulation*. The latter is a well-established and cost-effective method to investigate the dynamic response of a system, alternatively to full-scale physical testing. In hybrid simulation, the response of a prototype system subjected to realistic excitation is obtained by combining numerical and physical substructures into a hybrid model of the prototype. The hybrid model results from the coupling of the encompassing numerical and physical substructures. Transfer systems, e.g. an arrangement of actuators, are employed to accomplish this coupling and hence to synchronize the boundary conditions of each utilized substructure. Since trust in hybrid simulation results is paramount, conducting the latter in real-time is essential. This is most acute for hybrid models with loading-rate-sensitive substructures. Nonetheless, various challenges arise in real-time hybrid simulation risking compromising the simulation fidelity.

In this regard, the *first research objective* of this dissertation emerges. It relies on developing methodologies for performing hybrid simulations in real-time. In particular, two types of challenges are considered. The first one originates from the physical substructures and more specifically from the utilized transfer system. Due to the inherent dynamics of the latter, time delays and tracking errors are introduced within the hybrid simulation loop comprising the time scale of the simulation and its outcomes. To tackle this challenge, advanced control methodologies are implemented in order to compensate for the arising time delays and tracking errors. In the scope of this dissertation, two control schemes based on classical and adaptive model predictive control are investigated. The second considered challenge in real-time hybrid simulation initiates from the numerical substructures and especially from nonlinear high-dimensional models. In detail, in order to adequately capture the dynamic response of such models, it is often the case to further reduce the time scale of the simulation. Yet, the smaller the time-step of the simulation, the higher the computational time required to solve it. Therefore, for the case that the required computational time exceeds the actual simulation time, the time scale of hybrid simulation is distorted risking falsifying the obtained results. To overcome this challenge, a model order reduction framework is proposed based on polynomial chaos expansion and feedforward neural networks.

Apart from performing hybrid simulations in real-time, this research project also focuses on extending the current state-of-the-art deterministic hybrid simulation into the concept of stochastic hybrid simulation. The latter composes the *second research objective* of this dissertation. The motivation of this objective is the fact that often substructures of the hybrid model and respective excitation are conceived as deterministic, while in reality, uncertainties are always present affecting the performance of the examined systems. Therefore, the second research objective aims to unveil the inner workings of structural systems by means of understanding how uncertainties evolve, propagate and influence the system response. The valuable information of revealing the sensitivity of structural systems to a broad range of input uncertainties could not only guideline future additional experimental campaigns but also assist the design phase and assessment of structural systems. In this regard, an uncertainty quantification framework is proposed, based on global sensitivity analysis using Sobol' indices. Surrogates are employed to alleviate the computational/experimentation burden associated with the generation of large datasets that are required for global sensitivity analysis. Specific parameters of the hybrid model are selected to represent the input uncertainties of the framework, whereas

their contribution to the uncertainty on the hybrid model output response is quantified.

9.1.2 *Summary of each chapter*

This dissertation could be divided into two parts. The first part is aligned with the first research objective and corresponds to Chapters 3-5. Respectively, the second part addresses the second research objective and consists of Chapters 7-8. Chapters 2 and 6 serve as introduction to hybrid simulation and uncertainty quantification background, respectively. In particular, each chapter covers the following:

- **Chapter 1** introduces the motivation of this research project. The concept of hybrid simulation is firstly here presented and the need to utilize it for model validation and calibration is highlighted. The research objectives as well as the main contribution of the dissertation are defined. The scientific and economic significance of the latter is also discussed.

- **Chapter 2** offers a thorough introduction into the world of hybrid simulation. The concept of dynamic substructuring as well as the main components of hybrid simulation are explained. Additionally, potential sources of uncertainties and errors that could be present in a hybrid simulation are indicated and their effect is emphasized. Finally, model verification and validation strategies are shortly addressed.

- **Chapter 3** begins to address the first research objective by tackling challenges that arise in a real-time hybrid simulation due to the employed transfer system that is utilized within the hybrid simulation loop. A novel control strategy is investigated and proposed for compensation for the time delays and tracking errors. The controller is based on model predictive control along with a Kalman filter and a polynomial extrapolation algorithm. The controller's structure and its mathematical formulation are explained, followed by two parametric case studies, employing the proposed control scheme. The performance and robustness of the controller are evaluated by means of predefined assessment criteria.

- **Chapter 4** continues to address challenges that arise in real-time hybrid simulation due to the inherent dynamics of the utilized transfer system. In comparison with the previous chapter though, Chapter 4 investigates transfer systems that consists of time-varying parameters. To compensate for the time delays and tracking errors introduced by

such systems, an extension of the control scheme presented in Chapter 3 is proposed here. It is based on adaptive model predictive control along with a linear time-varying Kalman filter and an online model identification algorithm. With the latter, ARX polynomial models are identified online and used to estimate the changing transfer system dynamics. After presenting the proposed adaptive control scheme, a parametric case study is utilized to validate it. Results obtained both from Chapters 3 and 4, indicate that conducting hybrid simulation in real-time using the proposed control strategies is demonstrated to be effective for structural performance assessment.

- **Chapter 5** keeps focusing on challenges that arise in a real-time hybrid simulation. Nonetheless, in Chapter 5 concentration is given to time delays introduced by the required computational time to simulate high-dimensional nonlinear numerical substructures of the hybrid model. To tackle this challenge a model order reduction framework is proposed. The goal of the framework is to approximate the original models with simpler-to-evaluate functions and thus reduce their complexity and the associated required computational time. In this regard, polynomial chaos expansion and feedforward neural networks are utilized to emulate the response of the investigated numerical substructures. Replacement of the selected original numerical substructures with their corresponding polynomial chaos expansion/neural networks, within the hybrid simulation loop, results in the reduced-order hybrid model. A parametric case study is employed to investigate the performance of the framework. Testing scenarios utilizing only polynomial chaos expansion or neural networks of the respective numerical substructures, as well as combinations of the latter two within the same hybrid model, are performed. Comparisons between the full-order and reduced-order hybrid model responses are made to assess whether the latter can accurately replicate the initial dynamics. Results indicate that the corresponding errors are negligibly small, while the related computational time is significantly reduced. Additionally, differences in the performance between polynomial chaos expansion and feedforward neural networks as model order reduction techniques for the presented case study are highlighted.

- **Chapter 6** serves as an introduction to the second research objective and more specifically outlines the uncertainty quantification framework that is used in Chapters 7 and 8. Chapter 6 briefly discusses each step of the proposed framework, offering a short overview of the available methods in each step.

- **Chapter** 7 presents more comprehensively the mathematical formulation of the tools used within the proposed uncertainty quantification framework, for hybrid models with deterministic physical substructures (it is assumed that nominally identical specimen have identical responses, plus some negligible measurement noise). The idea is to surrogate the hybrid model response as a function of the input uncertainties that can be controlled by the experimenter and originate from numerical substructures and loading. In Chapter 7, the proposed framework is based on utilizing three different surrogate modeling techniques, namely polynomial chaos expansion, Kriging and polynomial chaos Kriging. Based on the developed surrogates, global sensitivity analysis with Sobol' indices is performed. For the case of polynomial chaos expansion, Sobol' indices are computed analytically as a by-product of its coefficients at no extra computational cost. For the case of Kriging and polynomial chaos Kriging, the Sobol' indices are computed in a Monte Carlo simulation fashion. Two parametric case studies are investigated and results validate the proposed framework. Comparison of the obtained Sobol' indices, using the different surrogate modeling techniques, is made and potential deviations are highlighted.

- **Chapter 8** extends the framework presented in Chapter 7 for the more realistic case of hybrid models with stochastic physical substructures, i.e. nominally identical specimens are, in practice, never actually identical. The overall idea is similar to the one explained in Chapter 7. However, in this case, some latent variables that cannot be controlled by the experimenter nor appear in the vector of input uncertainties, make the hybrid model response stochastic. To account for this non-deterministic response, generalized lambda models are employed as surrogates of the hybrid model output response. Quantile-based extension of the classical Sobol' indices is used to perform global sensitivity analysis. A parametric case study is employed to validate the framework.

Following the presentation of the dissertation's research objectives as well as the proposed methodologies to achieve these objectives, it can be stated, that the main contribution of this dissertation is: i) to propose new methodologies for rendering hybrid simulation as a high-fidelity simulation tool, that can be used ii) to apply uncertainty quantification frameworks to examine the dynamic response of uncertain structural systems that operate under uncertain conditions. Therefore, the developed methodologies and proposed frameworks can be utilized to accomplish the overreaching goal;

to support the development of trusted computer models that can be used in the design cycle decision-making process without having more complete test data from prototypes or full-scale physical testing.

9.2 Future research directions

As a continuation of this dissertation, the following future directions are conceived:

- In the developed control schemes for the compensation of time delays and tracking errors in real-time hybrid simulation, only hybrid models that encompass a single transfer system were considered. However, in practice, there are applications that require more than one actuation system, e.g. testing of a two-story structure. In such cases, multiple controllers should be designed and deployed. In this regard, future research could aim at expanding the single model predictive controller to a scheme of multiple model predictive controllers that can coordinate online with each other and control several transfer systems of a single hybrid model.
- Additionally, the proposed control schemes were validated using only virtual case studies. Therefore, next steps should aim to deploy the developed control strategies to physical test benches.
- For global sensitivity analysis, surrogates can alleviate the computational burden of obtaining the large datasets that are needed. Yet, for developing an accurate surrogate, still several hybrid model evaluations are required. However, when the physical substructure of the hybrid model cannot sustain repeated excitation, due to accumulated damage, replacing the physical specimen after each experiment would be essential. In such cases, the experimentation cost could grow significantly and be not reasonably affordable. Therefore, future research could address the issue of adaptive sampling (a.k.a. active learning) of the input parameter space of the hybrid model to minimize the experimentation cost necessary to compute an accurate surrogate model for global sensitivity analysis.
- The proposed framework for global sensitivity analysis is validated only to case studies with statistically independent input random variables. Hence, future research direction could investigate the more realistic scenario with dependent input random variables.
- In all the case studies investigated in this dissertation, the time-step of the hybrid simulation was fixed. However, there exist applications that

the dynamic response of the examined system is subjected to discontinuities. Such an example is rocking from the earthquake engineering discipline. In rocking, in each time instant that an impact occurs, the time-step of the simulation has to adapt in order to properly capture this phenomenon and the respective dynamic response of the structural system. Therefore, future research could aim at conducting hybrid simulations at variable time-steps to be able to capture such discontinuities.

Bibliography

[1] K. Worden, E. J. Cross, P. Gardner, R. J. Barthorpe, and D. J. Wagg. "On Digital Twins, Mirrors and Virtualisations". In: *Model Validation and Uncertainty Quantification, Volume 3*. Ed. by R. Barthorpe. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing, 2020, pp. 285–295. DOI: 10.1007/978-3-030-12075-7_34.

[2] D. J. Wagg, P. Gardner, R. J. Barthorpe, and K. Worden. "On Key Technologies for Realising Digital Twins for Structural Dynamics Applications". In: *Model Validation and Uncertainty Quantification, Volume 3*. Ed. by R. Barthorpe. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing, 2020, pp. 267–272. DOI: 10.1007/978-3-030-12075-7_30.

[3] S. Kumar, Y. Itoh, K. Saizuka, and T. Usami. "Pseudodynamic Testing of Scaled Models". In: *Journal of Structural Engineering* 123.4 (1997), pp. 524–526. DOI: 10.1061/(ASCE)0733-9445(1997)123:4(524).

[4] X. Gao, N. Castaneda, and S. J. Dyke. "Real time hybrid simulation: from dynamic system, motion control to experimental error". In: *Earthquake Engineering & Structural Dynamics* 42.6 (2013), pp. 815–832. DOI: https://doi.org/10.1002/eqe.2246.

[5] T. Wang, M. Nakashima, and P. Pan. "On-line hybrid test combining with general-purpose finite element software". In: *Earthquake Engineering & Structural Dynamics* 35.12 (2006), pp. 1471–1488. DOI: https://doi.org/10.1002/eqe.586.

[6] J. E. Carrion, B. F. Spencer, and B. M. Phillips. "Real-time hybrid simulation for structural control performance assessment". In: *Earthquake Engineering and Engineering Vibration* 4.8 (2009), pp. 481–492. DOI: 10.1007/s11803-009-9122-4.

[7] M. I. Wallace, J. Sieber, S. A. Neild, D. J. Wagg, and B. Krauskopf. "Stability analysis of real-time dynamic substructuring using delay differential equation models". In: *Earthquake Engineering & Structural Dynamics* 34.15 (2005), pp. 1817–1832. DOI: 10.1002/eqe.513.

[8] G. Mosqueda, B. Stojadinovic, and S. A. Mahin. "Real-Time Error Monitoring for Hybrid Simulation. Part I: Methodology and Experimental Verification". In: *Journal of Structural Engineering* 133.8 (2007), pp. 1100–1108. DOI: 10.1061/(ASCE)0733-9445(2007)133:8(1100).

[9] T. Godfrey, S. Mullen, D. W. Griffith, N. Golmie, R. C. Dugan, and C. Rodine. "Modeling Smart Grid Applications with Co-Simulation". In: *2010 First IEEE International Conference on Smart Grid Communications*. 2010, pp. 291–296. DOI: 10.1109/SMARTGRID.2010.5622057.

[10] F. L. M. dos Santos, R. Pastorino, B. Peeters, C. Faria, W. Desmet, L. C. Sandoval Góes, and H. Van Der Auweraer. "Model Based System Testing: Bringing Testing and Simulation Close Together". In: *Structural Health Monitoring, Damage Detection & Mechatronics, Volume 7*. Ed. by A. Wicks and C. Niezrecki. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing, 2016, pp. 91–97. DOI: 10.1007/978-3-319-29956-3_10.

[11] J. G. Pearce, R. E. Crosbie, J. J. Zenor, R. Bednar, D. Word, and N. G. Hingorani. "Developments and Applications of Multi-rate Simulation". In: *2009 11th International Conference on Computer Modelling and Simulation*. 2009, pp. 129–133. DOI: 10.1109/UKSIM.2009.23.

[12] J. P. Waldbjoern, A. Maghareh, G. Ou, S. J. Dyke, and H. Stang. "Multi-rate Real Time Hybrid Simulation operated on a flexible LabVIEW real-time platform". In: *Engineering Structures* 239 (2021), p. 112308. DOI: 10.1016/j.engstruct.2021.112308.

[13] T. Sauder. "Fidelity of Cyber-Physical Empirical Methods Application to the active truncation of slender marine structures". Doctoral thesis. NTNU, 2018.

[14] O. S. Bursi and D. Wagg. *Modern Testing Techniques for Structural Systems: Dynamics and Control*. Springer Science & Business Media, 2009.

[15] J. E. Carrion and B. F. Spencer. *Model-based Strategies for Real-time Hybrid Testing*. Tech. rep. NSEL-006. Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign., 2007.

[16] B. Stojadinovic, G. Mosqueda, and S. A. Mahin. "Event-Driven Control System for Geographically Distributed Hybrid Simulation". In: *Journal of Structural Engineering* 132.1 (2006), pp. 68–77. DOI: 10.1061/(ASCE)0733-9445(2006)132:1(68).

[17] M. Nakashima, H. Kato, and E. Takaoka. "Development of real-time pseudo dynamic testing". In: *Earthquake Engineering & Structural Dynamics* 21.1 (1992), pp. 79–92. DOI: 10.1002/eqe.4290210106.

[18] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim. "CORE: A real-time network emulator". In: *MILCOM 2008 - 2008 IEEE Military Communications Conference*. 2008, pp. 1–7. DOI: 10.1109/MILCOM.2008.4753614.

[19] A. H. Schellenberg, S. A. Mahin, and G. L. Fenves. *Advanced Implementation of Hybrid Simulation*. Tech. rep. PEER 2009/104. Pacific Earthquake Engineering Research Center, UC Berkeley, California., 2009, p. 286.

[20] O.-S. Kwon, N. Nakata, A. Elnashai, and B. Spencer. "A framework for multi-site distributed simulation and application to complex structural systems". In: *Journal of Earthquake Engineering* 09.05 (2005), pp. 741–753. DOI: 10.1142/S1363246905002158.

[21] O. S. Bursi, G. Abbiati, E. Cazzador, P. Pegon, and F. J. Molina. "Nonlinear heterogeneous dynamic substructuring and partitioned FETI time integration for the development of low-discrepancy simulation models". In: *International Journal for Numerical Methods in Engineering* 112.9 (2017), pp. 1253–1291. DOI: https://doi.org/10.1002/nme.5556.

[22] G. Mosqueda, B. Stojadinovic, J. Hanley, M. Sivaselvan, and A. M. Reinhorn. "Hybrid Seismic Response Simulation on a Geographically Distributed Bridge Model". In: *Journal of Structural Engineering* 134.4 (2008), pp. 535–543. DOI: 10.1061/(ASCE)0733-9445(2008)134:4(535).

[23] C. E. Silva, D. Gomez, A. Maghareh, S. J. Dyke, and B. F. Spencer. "Benchmark control problem for real-time hybrid simulation". In: *Mechanical Systems and Signal Processing* 135 (2020), p. 106381. DOI: 10.1016/j.ymssp.2019.106381.

[24] N. Castaneda, X. Gao, and S. J. Dyke. "Computational Tool for Real-Time Hybrid Simulation of Seismically Excited Steel Frame Structures". In: *Journal of Computing in Civil Engineering* 29.3 (2015), p. 04014049. DOI: 10.1061/(ASCE)CP.1943-5487.0000341.

[25] A. Friedman, S. J. Dyke, B. Phillips, R. Ahn, B. Dong, Y. Chae, N. Castaneda, Z. Jiang, J. Zhang, Y. Cha, A. I. Ozdagli, B. F. Spencer, J. Ricles, R. Christenson, A. Agrawal, and R. Sause. "Large-Scale Real-Time Hybrid Simulation for Evaluation of Advanced Damp-

ing System Performance". In: *Journal of Structural Engineering* 141.6 (2015), p. 04014150. DOI: 10.1061/(ASCE)ST.1943-541X.0001093.

[26] T. Sauder, S. Marelli, K. Larsen, and A. J. Sørensen. "Active truncation of slender marine structures: Influence of the control system on fidelity". In: *Applied Ocean Research* 74 (2018), pp. 154–169. DOI: 10.1016/j.apor.2018.02.023.

[27] S. A. Vilsen, T. Sauder, A. J. Sørensen, and M. Føre. "Method for Real-Time Hybrid Model Testing of ocean structures: Case study on horizontal mooring systems". In: *Ocean Engineering* 172 (2019), pp. 46–58. DOI: 10.1016/j.oceaneng.2018.10.042.

[28] W. Song, C. Sun, Y. Zuo, V. Jahangiri, Y. Lu, and Q. Han. "Conceptual Study of a Real-Time Hybrid Simulation Framework for Monopile Offshore Wind Turbines Under Wind and Wave Loads". In: *Frontiers in Built Environment* 6 (2020), p. 129. DOI: 10.3389/fbuil.2020.00129.

[29] V. Chabaud, L. Eliassen, M. Thys, and T. Sauder. "Multiple-degree-of-freedom actuation of rotor loads in model testing of floating wind turbines using cable-driven parallel robots". In: *Journal of Physics: Conference Series* 1104.1 (2018), p. 012021. DOI: 10.1088/1742-6596/1104/1/012021.

[30] C. A. Whyte, K. R. Mackie, and B. Stojadinovic. "Hybrid Simulation of Thermomechanical Structural Response". In: *Journal of Structural Engineering* 142.2 (2016), p. 04015107. DOI: 10.1061/(ASCE)ST.1943-541X.0001346.

[31] A. Sauca, N. Mortensen, A. Drustrup, and G. Abbiati. "Experimental validation of a hybrid fire testing framework based on dynamic relaxation". In: *Fire Safety Journal* 121 (2021), p. 103315. DOI: 10.1016/j.firesaf.2021.103315.

[32] S. Idinyang, A. Franza, C. M. Heron, and A. M. Marshall. "Real-time data coupling for hybrid testing in a geotechnical centrifuge". In: *International Journal of Physical Modelling in Geotechnics* 19.4 (2018), pp. 208–220. DOI: 10.1680/jphmg.17.00063.

[33] M. I. Wallace, D. J. Wagg, S. A. Neild, P. Bunniss, N. A. J. Lieven, and A. J. Crewe. "Testing coupled rotor blade–lag damper vibration using real-time dynamic substructuring". In: *Journal of Sound and Vibration* 307.3 (2007), pp. 737–754. DOI: 10.1016/j.jsv.2007.07.004.

[34] D. Wagg, S. Neild, and P. Gawthrop. "Real-Time Testing With Dynamic Substructuring". In: *Modern Testing Techniques for Structural Systems: Dynamics and Control*. Ed. by O. S. Bursi and D. Wagg. CISM International Centre for Mechanical Sciences. Vienna: Springer, 2008, pp. 293–342. DOI: 10.1007/978-3-211-09445-7_7.

[35] A. Bartl, J. Mayet, M. K. Mahdiabadi, and D. J. Rixen. "Multi-DoF Interface Synchronization of Real-Time-Hybrid-Tests Using a Recursive-Least-Squares Adaption Law: A Numerical Evaluation". In: *Dynamics of Coupled Structures, Volume 4*. Ed. by M. Allen, R. L. Mayes, and D. Rixen. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing, 2016, pp. 7–14. DOI: 10.1007/978-3-319-29763-7_2.

[36] T. Boge and O. Ma. "Using advanced industrial robotics for spacecraft Rendezvous and Docking simulation". In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 1–4. DOI: 10.1109/ICRA.2011.5980583.

[37] *Dynamic virtualisation: modelling performance of engineering structures*.

[38] R. R. Craig and M. C. C. Bampton. "Coupling of substructures for dynamic analyses". In: *AIAA Journal* 6.7 (1968), pp. 1313–1319. DOI: 10.2514/3.4741.

[39] D. J. Rixen. "A dual Craig–Bampton method for dynamic substructuring". In: *Journal of Computational and Applied Mathematics*. Selected Papers from the Second International Conference on Advanced Computational Methods in Engineering (ACOMEN 2002) 168.1 (2004), pp. 383–391. DOI: 10.1016/j.cam.2003.12.014.

[40] M. J. Hashemi and G. Mosqueda. "Innovative substructuring technique for hybrid simulation of multistory buildings through collapse". In: *Earthquake Engineering & Structural Dynamics* 43.14 (2014), pp. 2059–2074. DOI: https://doi.org/10.1002/eqe.2427.

[41] G. Abbiati, I. Lanese, E. Cazzador, O. S. Bursi, and A. Pavese. "A computational framework for fast-time hybrid simulation based on partitioned time integration and state-space modeling". In: *Structural Control and Health Monitoring* 26.10 (2019), e2419. DOI: https://doi.org/10.1002/stc.2419.

[42] G. Abbiati, P. Covi, N. Tondini, O. S. Bursi, and B. Stojadinović. "A Real-Time Hybrid Fire Simulation Method Based on Dynamic Relaxation and Partitioned Time Integration". In: *Journal of Engineering Mechanics* 146.9 (2020), p. 04020104. DOI: 10.1061/(ASCE)EM.1943-7889.0001826.

[43] M. Nakashima and N. Masaoka. "Real-time on-line test for MDOF systems". In: *Earthquake Engineering & Structural Dynamics* 28.4 (1999), pp. 393–420. DOI: https://doi.org/10.1002/(SICI)1096-9845(199904)28:4<393::AID-EQE823>3.0.CO;2-C.

[44] N. M. Newmark. "A Method of Computation for Structural Dynamics". In: *Journal of the Engineering Mechanics Division* 85.3 (1959), pp. 67–94. DOI: 10.1061/JMCEA3.0000098.

[45] P. B. Shing and S. A. Mahin. "Experimental Error Effects in Pseudodynamic Testing". In: *Journal of Engineering Mechanics* 116.4 (1990), pp. 805–821. DOI: 10.1061/(ASCE)0733-9399(1990)116:4(805).

[46] S.-Y. Chang. "Improved numerical dissipation for explicit methods in pseudodynamic tests". In: *Earthquake Engineering & Structural Dynamics* 26.9 (1997), pp. 917–929. DOI: 10.1002/(SICI)1096-9845(199709)26:9<917::AID-EQE685>3.0.CO;2-9.

[47] J. Chung and J. M. Lee. "A new family of explicit time integration methods for linear and non-linear structural dynamics". In: *International Journal for Numerical Methods in Engineering* 37.23 (1994), pp. 3961–3976. DOI: 10.1002/nme.1620372303.

[48] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. "Improved numerical dissipation for time integration algorithms in structural dynamics". In: *Earthquake Engineering & Structural Dynamics* 5.3 (1977), pp. 283–292. DOI: https://doi.org/10.1002/eqe.4290050306.

[49] R.-Y. Jung, P. Benson Shing, E. Stauffer, and B. Thoen. "Performance of a real-time pseudodynamic test system considering nonlinear structural response". In: *Earthquake Engineering & Structural Dynamics* 36.12 (2007), pp. 1785–1809. DOI: 10.1002/eqe.722.

[50] O. Mercan and J. M. Ricles. "Experimental Studies on Real-Time Testing of Structures with Elastomeric Dampers". In: *Journal of Structural Engineering* 135.9 (2009), pp. 1124–1133. DOI: 10.1061/(ASCE)0733-9445(2009)135:9(1124).

[51] W. L. Wood, M. Bossak, and O. C. Zienkiewicz. "An alpha modification of Newmark's method". In: *International Journal for Numerical Methods in Engineering* 15.10 (1980), pp. 1562–1566. DOI: https://doi.org/10.1002/nme.1620151011.

[52] J. Chung and G. M. Hulbert. "A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized-α Method". In: *Journal of Applied Mechanics* 60.2 (1993), pp. 371–375. DOI: 10.1115/1.2900803.

[53] T. J. R. Hughes and W. K. Liu. "Implicit-Explicit Finite Elements in Transient Analysis: Stability Theory". In: *Journal of Applied Mechanics* 45.2 (1978), pp. 371–374. DOI: 10.1115/1.3424304.

[54] T. J. R. Hughes and W. K. Liu. "Implicit-Explicit Finite Elements in Transient Analysis: Implementation and Numerical Examples". In: *Journal of Applied Mechanics* 45.2 (1978), pp. 375–378. DOI: 10.1115/1.3424305.

[55] B. Wu, G. Xu, Q. Wang, and M. S. Williams. "Operator-splitting method for real-time substructure testing". In: *Earthquake Engineering & Structural Dynamics* 35.3 (2006), pp. 293–314. DOI: https://doi.org/10.1002/eqe.519.

[56] P. A. Bonnet, M. S. Williams, and A. Blakeborough. "Evaluation of numerical time-integration schemes for real-time hybrid testing". In: *Earthquake Engineering & Structural Dynamics* 37.13 (2008), pp. 1467–1490. DOI: https://doi.org/10.1002/eqe.821.

[57] B. M. Phillips and B. F. Spencer. *Model-Based Framework for Real-Time Dynamic Structural Performance Evaluation*. Tech. rep. NSEL-031. Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign, 2012.

[58] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. 3rd edition. Chichester, West Sussex, United Kingdom: Wiley, 2016.

[59] D. Griffiths and D. J. Higham. *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. Springer Undergraduate Mathematics Series. London: Springer-Verlag, 2010. DOI: 10.1007/978-0-85729-148-6.

[60] N. Tsokanas, G. Abbiati, K. Kanellopoulos, and B. Stojadinovic. "Multi-Axial Hybrid Fire Testing based on Dynamic Relaxation". In: *Fire Safety Journal* (2021), p. 103468. DOI: 10.1016/j.firesaf.2021.103468.

[61] P. Pan, T. Wang, and M. Nakashima. *Development of Online Hybrid Testing: Theory and Applications to Structural Engineering*. 1st edition. Oxford England ; Waltham, MA: Butterworth-Heinemann, 2015.

[62] G. A. Fermandois and B. F. Spencer. *Development and Implementation of a Multi-axial Real-time Hybrid Simulation Framework*. Tech. rep. NSEL-049. Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign, 2018.

[63] G. C. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer Science & Business Media, 2011.

[64] R. Al-Mahaidi, M. J. Hashemi, R. Kalfat, G. Burnett, and J. Wilson. *Multi-axis Substructure Testing System for Hybrid Simulation*. SpringerBriefs in Structural Mechanics. Springer Singapore, 2018. DOI: 10.1007/978-981-10-5867-7.

[65] G. Abbiati, S. Marelli, N. Tsokanas, B. Sudret, and B. Stojadinović. "A global sensitivity analysis framework for hybrid simulation". In: *Mechanical Systems and Signal Processing* 146 (2021), p. 106997. DOI: https://doi.org/10.1016/j.ymssp.2020.106997.

[66] Horiuchi T. "Development of a realtime hybrid experimental system with actuator delay compensation". In: *Proc. 11th World Conference on Earthquake Engineering*. 1996.

[67] M. Wallace, D. Wagg, and S. Neild. "An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 461.2064 (2005), pp. 3807–3826. DOI: 10.1098/rspa.2005.1532.

[68] J. Zhao, C. French, C. Shield, and T. Posbergh. "Considerations for the development of real-time dynamic testing using servo-hydraulic actuation". In: *Earthquake Engineering & Structural Dynamics* 32.11 (2003), pp. 1773–1794. DOI: 10.1002/eqe.301.

[69] P. J. Gawthrop, M. I. Wallace, S. A. Neild, and D. J. Wagg. "Robust real-time substructuring techniques for under-damped systems". In: *Structural Control and Health Monitoring* 14.4 (2007), pp. 591–608. DOI: 10.1002/stc.174.

[70] C. Chen and J. M. Ricles. "Analysis of actuator delay compensation methods for real-time testing". In: *Engineering Structures* 31.11 (2009), pp. 2643–2655. DOI: 10.1016/j.engstruct.2009.06.012.

[71] S. A. Neild, D. P. Stoten, D. Drury, and D. J. Wagg. "Control issues relating to real-time substructuring experiments using a shaking table". In: *Earthquake Engineering & Structural Dynamics* 34.9 (2005), pp. 1171–1192. DOI: 10.1002/eqe.473.

[72] D. J. Wagg and D. P. Stoten. "Substructuring of dynamical systems via the adaptive minimal control synthesis algorithm". In: *Earthquake Engineering & Structural Dynamics* 30.6 (2001), pp. 865–877. DOI: 10.1002/eqe.44.

[73] Y. Chae, K. Kazemibidokhti, and J. M. Ricles. "Adaptive time series compensator for delay compensation of servo-hydraulic actuator systems for real-time hybrid simulation". In: *Earthquake Engineering & Structural Dynamics* 42.11 (2013), pp. 1697–1715. DOI: 10.1002/eqe.2294.

[74] P.-C. Chen, C.-M. Chang, B. F. Spencer, and K.-C. Tsai. "Adaptive model-based tracking control for real-time hybrid simulation". In: *Bulletin of Earthquake Engineering* 13.6 (2015), pp. 1633–1653. DOI: 10.1007/s10518-014-9681-2.

[75] B. Phillips and B. Spencer. "Model-based feedforward-feedback actuator control for real-time hybrid simulation". In: *Journal of Structural Engineering* 139.7 (2013), pp. 1205–1214. DOI: 10.1061/(ASCE)ST.1943-541X.0000606.

[76] B. Phillips and B. Spencer Jr. "Model-Based Multiactuator Control for Real-Time Hybrid Simulation". In: *Journal of Engineering Mechanics* 139.2 (2013), pp. 219–228. DOI: 10.1061/(ASCE)EM.1943-7889.0000493.

[77] X. Ning, Z. Wang, H. Zhou, B. Wu, Y. Ding, and B. Xu. "Robust actuator dynamics compensation method for real-time hybrid simulation". In: *Mechanical Systems and Signal Processing* 131 (2019), pp. 49–70. DOI: 10.1016/j.ymssp.2019.05.038.

[78] G. Ou, A. I. Ozdagli, S. J. Dyke, and B. Wu. "Robust integrated actuator control: experimental verification and real-time hybrid-simulation implementation". In: *Earthquake Engineering & Structural Dynamics* 44.3 (2015), pp. 441–460. DOI: 10.1002/eqe.2479.

[79] A. Maghareh, S. J. Dyke, and C. E. Silva. "A Self-tuning Robust Control System for nonlinear real-time hybrid simulation". In: *Earthquake Engineering & Structural Dynamics* 49.7 (2020), pp. 695–715. DOI: 10.1002/eqe.3260.

[80] J. Condori, A. Maghareh, J. Orr, H.-W. Li, H. Montoya, S. Dyke, C. Gill, and A. Prakash. "Exploiting Parallel Computing to Control Uncertain Nonlinear Systems in Real-Time". In: *Experimental Techniques* 44.6 (2020), pp. 735–749. DOI: 10.1007/s40799-020-00373-w.

[81] H. Li, A. Maghareh, H. Montoya, J. W. Condori Uribe, S. J. Dyke, and Z. Xu. "Sliding mode control design for the benchmark problem in real-time hybrid simulation". In: *Mechanical Systems and Signal Processing* 151 (2021), p. 107364. DOI: 10.1016/j.ymssp.2020.107364.

[82] S. J. Dyke, D. Gomez, and B. F. Spencer. "Editorial: Special issue on the real-time hybrid simulation benchmark problem". In: *Mechanical Systems and Signal Processing* 142 (2020), p. 106804. DOI: 10.1016/j.ymssp.2020.106804.

[83] N. Tsokanas, D. Wagg, and B. Stojadinović. "Robust Model Predictive Control for Dynamics Compensation in Real-Time Hybrid Simulation". In: *Frontiers in Built Environment* 6 (2020), p. 127. DOI: 10.3389/fbuil.2020.00127.

[84] N. Tsokanas, R. Pastorino, and B. Stojadinovic. "Adaptive model predictive control for actuation dynamics compensation in real-time hybrid simulation". In: *Preprint submitted in engrXiv* (2021). DOI: 10.31224/osf.io/c974v.

[85] A. D. Kiureghian and O. Ditlevsen. "Aleatory or epistemic? Does it matter?" In: *Structural Safety*. Risk Acceptance and Risk Communication 31.2 (2009), pp. 105–112. DOI: 10.1016/j.strusafe.2008.06.020.

[86] I. Abdallah. "Assessment of extreme design loads for modern wind turbines using the probabilistic approach". Doctoral thesis. DTU Wind Energy, 2015.

[87] C. R. Thewalt and S. A. Mahin. *Hybrid solution techniques for generalized pseudodynamic testing*. Tech. rep. UCB/EERC-87/09. Earthquake Engineering Research Center, 1987, p. 148.

[88] L. E. Schwer. "An overview of the PTC 60/V&V 10: guide for verification and validation in computational solid mechanics". In: *Engineering with Computers* 23.4 (2007), pp. 245–252. DOI: 10.1007/s00366-007-0072-z.

[89] W. L. Oberkampf and C. J. Roy. *Verification and Validation in Scientific Computing*. Cambridge: Cambridge University Press, 2010. DOI: 10.1017/CBO9780511760396.

[90] N. Tsokanas and B. Stojadinovic. "Evaluation of the Seismic Response of a Structure with an attached Pendulum using Stochastic Real-Time Hybrid Simulation". In: ETH Zurich, Institute of Structural Engineering, 2020, p. 18748. DOI: 10.3929/ethz-b-000474269.

[91] N. Tsokanas and B. Stojadinovic. "A stochastic real-time hybrid simulation of the seismic response of a magnetorheological damper". In: 2020. DOI: 10.3929/ethz-b-000462979.

[92] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814. DOI: 10.1016/S0005-1098(99)00214-9.

[93] E. Zafiriou. "Robust model predictive control of processes with hard constraints". In: *Computers & Chemical Engineering* 14.4-5 (1990), pp. 359–371. DOI: 10.1016/0098-1354(90)87012-E.

[94] F. Delbos and J. C. Gilbert. "Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems". In: *Journal of Convex Analysis* 12.1 (2005), pp. 45–69.

[95] P. Tøndel, T. A. Johansen, and A. Bemporad. "An algorithm for multi-parametric quadratic programming and explicit MPC solutions". In: *Automatica* 39.3 (2003), pp. 489–497. DOI: 10.1016/S0005-1098(02)00250-9.

[96] C. Schmid and L. T. Biegler. "Quadratic programming methods for reduced hessian SQP". In: *Computers & Chemical Engineering*. An International Journal of Computer Applications in Chemical Engineering 18.9 (1994), pp. 817–832. DOI: 10.1016/0098-1354(94)E0001-4.

[97] A. Bemporad, N. L. Ricker, and M. Morari. *Model Predictive Control Toolbox: Getting Started Guide, The Math Works*. 2020.

[98] G. Bitsoris. "Positively invariant polyhedral sets of discrete-time linear systems". In: *International Journal of Control* 47.6 (1988), pp. 1713–1726. DOI: 10.1080/00207178808906131.

[99] E. F. Camacho and C. Bordons. *Model Predictive control*. Ed. by M. J. Grimble and M. A. Johnson. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2007.

[100] J. A. Rossiter. *Model-based predictive control: a practical approach*. Control series. Boca Raton: CRC Press, 2003.

[101] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004.

[102] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model predictive control: theory, computation, and design*. 2nd edition. Madison, Wisconsin: Nob Hill Publishing, 2017.

[103] S. J. Dyke, B. F. Spencer, P. Quast, and M. K. Sain. "Role of Control-Structure Interaction in Protective System Design". In: *Journal of Engineering Mechanics* 121.2 (1995), pp. 322–338. DOI: 10.1061/(ASCE)0733-9399(1995)121:2(322).

[104] M. D. Mckay, R. J. Beckman, and W. J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code". In: *Technometrics* 42.1 (2000), pp. 55–61. DOI: 10.2307/1268522.

[105] S. J. Dyke, B. F. Spencer, M. K. Sain, and J. D. Carlson. "An experimental study of MR dampers for seismic protection". In: *Smart Materials and Structures* 7.5 (1998), pp. 693–703. DOI: 10.1088/0964-1726/7/5/012.

[106] F. Ikhouane and S. J. Dyke. "Modeling and identification of a shear mode magnetorheological damper". In: *Smart Materials and Structures* 16.3 (2007), pp. 605–616. DOI: 10.1088/0964-1726/16/3/007.

[107] A. R. Tsouroukdissian, F. Ikhouane, J. Rodellar, and N. Luo. "Modeling and Identification of a Small-scale Magnetorheological Damper". In: *Journal of Intelligent Material Systems and Structures* 20.7 (2008), pp. 825–835. DOI: 10.1177/1045389X08098440.

[108] N. Aguirre Carvajal, F. Ikhouane, J. Rodellar Benedé, D. Wagg, and S. Neild. "Viscous + Dahl model for MR damper characterization: a real-time hybrid test (RTHT) validation". eng. In: *14th European Conference on Earthquake Engineering 2010: Ohrid, Republic of Macedonia, 30 August-3 September 2010*. 2010.

[109] N. Tsokanas, R. Pastorino, B. Peeters, and B. Stojadinovic. "Development of an adaptive model predictive reference tracking controller for hard real-time hybrid simulation". In: 2021. DOI: 10.3929/ethz-b-000473422.

[110] E. Kaiser, J. N. Kutz, and S. L. Brunton. "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2219 (2018), p. 20180335. DOI: 10.1098/rspa.2018.0335.

[111] B. R. Maner, F. J. Doyle, B. A. Ogunnaike, and R. K. Pearson. "A nonlinear model predictive control scheme using second order Volterra models". In: *Proceedings of 1994 American Control Conference - ACC '94*. Vol. 3. 1994, pp. 3253–3257. DOI: 10.1109/ACC.1994.735176.

[112] H. Akaike. "Fitting autoregressive models for prediction". In: *Annals of the Institute of Statistical Mathematics* 21.1 (1969), pp. 243–247. DOI: 10.1007/BF02532251.

[113] L. Ljung. *System identification theory for the user*. eng. 2nd ed. Prentice Hall information and system sciences series. Upper Saddle River, N.J: Prentice Hall, 1999.

[114] S. M. Pinheiro. "Motorcycle modeling for eCVT-in-the-loop real-time hybrid testing". eng. MA thesis. Porto, Portugal: University of Porto, 2020.

[115] R. Sharp, S. Evangelou, and D. Limebeer. "Advances in the Modelling of Motorcycle Dynamics". In: *Multibody System Dynamics* 12.3 (2004), pp. 251–283. DOI: 10.1023/B:MUBO.0000049195.60868.a2.

[116] M. Tanelli. *Modelling, simulation and control of two-wheeled vehicles*. eng. John Wiley & Sons, 2014.

[117] T. Kimishima, T. Nakamura, and T. Suzuki. "The Effects on Motorcycle Behavior of the Moment of Inertia of the Crankshaft". In: *SAE Transactions* 106 (1997), pp. 1993–2003.

[118] N. Tsokanas, T. Simpson, R. Pastorino, E. Chatzi, and B. Stojadinovic. "Model Order Reduction for Real-Time Hybrid Simulation: Comparing Polynomial Chaos Expansion and Neural Network methods". In: *Preprint submitted in engrXiv* (2021). DOI: 10.31224/osf.io/h2bnm.

[119] K. Kunisch and S. Volkwein. "Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics". In: *SIAM Journal on Numerical Analysis* 40.2 (2002), pp. 492–515. DOI: 10.1137/S0036142900382612.

[120] W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, eds. *Model Order Reduction: Theory, Research Aspects and Applications*. Vol. 13. Mathematics in Industry. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[121] A. Chatterjee. "An introduction to the proper orthogonal decomposition". In: *Current Science* 78.7 (2000), pp. 808–817.

[122] J. P. Bonnet, D. R. Cole, J. Delville, M. N. Glauser, and L. S. Ukeiley. "Stochastic estimation and proper orthogonal decomposition: Complementary techniques for identifying structure". In: *Experiments in Fluids* 17.5 (1994), pp. 307–314. DOI: 10.1007/BF01874409.

[123] S. Herzog. "The large scale structure in the near-wall region of turbulent pipe flow". eng. Doctoral thesis. Ithaca, NY: Cornell University, 1986.

[124] P. J. Schmid. "Dynamic mode decomposition of numerical and experimental data". In: *Journal of Fluid Mechanics* 656 (2010), pp. 5–28. DOI: 10.1017/S0022112010001217.

[125] P. J. Schmid, K. E. Meyer, and O. Pust. "Dynamic Mode Decomposition and Proper Orthogonal Decomposition of flow in a lid-driven cylindrical cavity". In: *8th International Symposium On Particle Image Velocimetry*. Melbourne, Victoria, Australia, 2009, p. 4.

[126] F. Chinesta, P. Ladeveze, and E. Cueto. "A Short Review on Model Order Reduction Based on Proper Generalized Decomposition". In: *Archives of Computational Methods in Engineering* 18.4 (2011), pp. 395–404. DOI: 10.1007/s11831-011-9064-7.

[127] R. H. MacNeal. "A hybrid method of component mode synthesis". In: *Computers & Structures*. Special Issue on Structural Dynamics 1.4 (1971), pp. 581–601. DOI: 10.1016/0045-7949(71)90031-9.

[128] R. R. Craig, A. Kurdila, and R. R. Craig. *Fundamentals of structural dynamics*. 2nd ed. Hoboken, N.J: John Wiley, 2006.

[129] F. M. Gruber and D. J. Rixen. "Dual Craig-Bampton component mode synthesis method for model order reduction of nonclassically damped linear systems". In: *Mechanical Systems and Signal Processing* 111 (2018), pp. 678–698. DOI: 10.1016/j.ymssp.2018.04.019.

[130] G. Miraglia, M. Petrovic, G. Abbiati, N. Mojsilovic, and B. Stojadinovic. "A model-order reduction framework for hybrid simulation based on component-mode synthesis". In: *Earthquake Engineering & Structural Dynamics* 49.8 (2020), pp. 737–753. DOI: 10.1002/eqe.3262.

[131] S. Jain, P. Tiso, J. B. Rutzmoser, and D. J. Rixen. "A quadratic manifold for model order reduction of nonlinear structural dynamics". In: *Computers & Structures* 188 (2017), pp. 80–94. DOI: 10.1016/j.compstruc.2017.04.005.

[132] M. Karamooz Mahdiabadi, P. Tiso, A. Brandt, and D. J. Rixen. "A non-intrusive model-order reduction of geometrically nonlinear structural dynamics using modal derivatives". In: *Mechanical Systems and Signal Processing* 147 (2021), p. 107126. DOI: 10.1016/j.ymssp.2020.107126.

[133] H. A. Jensen, E. Millas, D. Kusanovic, and C. Papadimitriou. "Model-reduction techniques for Bayesian finite element model updating using dynamic response data". In: *Computer Methods in Applied Mechanics and Engineering* 279 (2014), pp. 301–324. DOI: 10.1016/j.cma.2014.06.032.

[134] H. A. Jensen, A. Muñoz, C. Papadimitriou, and E. Millas. "Model-reduction techniques for reliability-based design problems of complex structural systems". In: *Reliability Engineering & System Safety* 149 (2016), pp. 204–217. DOI: 10.1016/j.ress.2016.01.003.

[135] C. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer-Verlag, 2006.

[136] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. 4th edition. Amsterdam: Morgan Kaufmann, 2016.

[137] D. Xiu and G. E. Karniadakis. "The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations". In: *SIAM Journal on Scientific Computing* 24.2 (2002), pp. 619–644. DOI: 10.1137/S1064827501387826.

[138] R. Ghanem and P. D. Spanos. "Polynomial Chaos in Stochastic Finite Elements". In: *Journal of Applied Mechanics* 57.1 (1990), pp. 197–202. DOI: 10.1115/1.2888303.

[139] E. Torre, S. Marelli, P. Embrechts, and B. Sudret. "Data-driven polynomial chaos expansion for machine learning regression". In: *Journal of Computational Physics* 388 (2019), pp. 601–623. DOI: 10.1016/j.jcp.2019.03.039.

[140] M. D. Spiridonakos and E. N. Chatzi. "Metamodeling of dynamic nonlinear structural systems through polynomial chaos NARX models". In: *Computers & Structures* 157 (2015), pp. 99–113. DOI: 10.1016/j.compstruc.2015.05.002.

[141] S. Marelli and B. Sudret. *UQLab user manual – Polynomial chaos expansions*. Tech. rep. # UQLab-V1.3-104. Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019.

[142] G. Blatman and B. Sudret. "Adaptive sparse polynomial chaos expansion based on least angle regression". In: *Journal of Computational Physics* 230.6 (2011), pp. 2345–2367. DOI: 10.1016/j.jcp.2010.12.021.

[143] M. Berveiller, B. Sudret, and M. Lemaire. "Stochastic finite element: a non intrusive approach by regression". In: *European Journal of Computational Mechanics* 15.1-3 (2006), pp. 81–92. DOI: 10.3166/remn.15.81-92.

[144] M. Eldred, C. Webster, and P. Constantine. "Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos." In: *49th AIAA Structures, Structural Dynamics, and Materials Conference*. 2008.

[145] K. Hornik, M. Stinchcombe, and W. Halbert. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. DOI: 10.1016/0893-6080(89)90020-8.

[146] M. Stinchcombe and H. White. "Approximating and learning unknown mappings using multilayer feedforward networks with bounded weights". In: *1990 IJCNN International Joint Conference on Neural Networks*. Vol. 3. 1990, pp. 7–16. DOI: 10.1109/IJCNN.1990.137817.

[147] R. v. d. Merwe, T. K. Leen, Z. Lu, S. Frolov, and A. M. Baptista. "Fast neural network surrogates for very high dimensional physics-based models in computational oceanography". In: *Neural Networks* 20.4 (2007), pp. 462–478. DOI: 10.1016/j.neunet.2007.04.023.

[148] B. T. Cao, M. Obel, S. Freitag, P. Mark, and G. Meschke. "Artificial neural network surrogate modelling for real-time predictions and control of building damage during mechanised tunnelling". In: *Advances in Engineering Software* 149 (2020), p. 102869. DOI: 10.1016/j.advengsoft.2020.102869.

[149] T. Wu and K. Ahsan. "Modeling hysteretic nonlinear behavior of bridge aerodynamics via cellular automata nested neural network". In: *Journal of Wind Engineering and Industrial Aerodynamics* 99.4 (2011), pp. 378–388. DOI: 10.1016/j.jweia.2010.12.011.

[150] R. K. Tripathy and I. Bilionis. "Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification". In: *Journal of Computational Physics* 375 (2018), pp. 565–588. DOI: 10.1016/j.jcp.2018.08.036.

[151] R. Snaiki and T. Wu. "Knowledge-enhanced deep learning for simulation of tropical cyclone boundary-layer winds". In: *Journal of Wind Engineering and Industrial Aerodynamics* 194 (2019), p. 103983. DOI: 10.1016/j.jweia.2019.103983.

[152] E. E. Bas and M. A. Moustafa. "Real-Time Hybrid Simulation with Deep Learning Computational Substructures: System Validation Using Linear Specimens". In: *Machine Learning and Knowledge Extraction* 2.4 (2020), pp. 469–489. DOI: 10.3390/make2040026.

[153] W. Mucha. "Application of Artificial Neural Networks in Hybrid Simulation". In: *Applied Sciences* 9.21 (2019).

[154] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[155] Y. Bengio and Y. Lecun. "Scaling learning algorithms towards AI". In: *Large-scale kernel machines*. MIT Press, 2007, pp. 321–359.

[156] *MATLAB Optimization Toolbox*. 2019.

[157] F. Burden and D. Winkler. "Bayesian Regularization of Neural Networks". In: *Artificial Neural Networks: Methods and Applications*. Ed. by D. J. Livingstone. Totowa, NJ: Humana Press, 2009, pp. 23–42.

[158] B. Sudret. "Global sensitivity analysis using polynomial chaos expansions". In: *Reliability Engineering & System Safety*. Bayesian Networks in Dependability 93.7 (2008), pp. 964–979. DOI: 10.1016/j.ress.2007.04.002.

[159] L. Le Gratiet, S. Marelli, and B. Sudret. "Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes". In: *Handbook of Uncertainty Quantification*. Ed. by R. Ghanem, D. Higdon, and H. Owhadi. Cham: Springer International Publishing, 2017, pp. 1289–1325.

[160] S. Marelli and B. Sudret. "UQLab: A Framework for Uncertainty Quantification in Matlab". In: *Proc. 2nd Int. Conf. on Vulnerability, Risk Analysis and Management (ICVRAM2014)*. Liverpool, United Kingdom, 2014, pp. 2554–2563. DOI: 10.1061/9780784413609.257.

[161] B. Sudret. "Uncertainty Propagation and sensitivity analysis in mechanical models Contributions to structural reliability and stochastic spectral methods". Doctoral thesis. Universite BLAISE PASCAL - Clermont II, 2007.

[162] B. Iooss and P. Lemaître. "A Review on Global Sensitivity Analysis Methods". In: *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*. Ed. by G. Dellino and C. Meloni. Operations Research/Computer Science Interfaces Series. Boston, MA: Springer US, 2015, pp. 101–122.

[163] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis. The Primer*. Vol. 76. 3. Chichester, UK: John Wiley & Sons, Ltd, 2007.

[164] *JCSS Probabilistic Model Code https://www.jcss-lc.org/jcss-probabilistic-model-code/*. en-US.

[165] E. T. Jaynes. "Information Theory and Statistical Mechanics". In: *Physical Review* 106.4 (1957), pp. 620–630. DOI: 10.1103/PhysRev.106.620.

[166] H. Akaike. "A new look at the statistical model identification". In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723. DOI: 10.1109/TAC.1974.1100705.

[167] G. Schwarz. "Estimating the Dimension of a Model". In: *The Annals of Statistics* 6.2 (1978), pp. 461–464.

[168] A. O'Hagan and J. J. Forster. *Kendall's Advanced Theory of Statistics, volume 2B: Bayesian Inference, second edition*. Vol. 2B. Arnold, 2004.

[169] B. Sudret and A. Der Kiureghian. *Stochastic Finite Element Methods and Reliability A State-of-the-Art Report*. Tech. rep. UCB/SEMM-2000/08. Berkeley: University of California, 2000.

[170] T. Hisada and S. Nakagiri. "Stochastic Finite Element Method Developed for Structural Safety and Reliability". In: *Proc. 3rd Int. Conf. Struct. Safety and Reliability*. Trondheim,Norway, 1981, pp. 395–408.

[171] G. Deodatis. "Weighted Integral Method. I: Stochastic Stiffness Matrix". In: *Journal of Engineering Mechanics* 117.8 (1991), pp. 1851–1864. DOI: 10.1061/(ASCE)0733-9399(1991)117:8(1851).

[172] M. Abramowitz and I. A. Stegun, eds. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. 1965.

[173] D. G. Cacuci. *Sensitivity & Uncertainty Analysis, Volume 1: Theory*. New York: Chapman and Hall/CRC, 2003. DOI: 10.1201/9780203498798.

[174] I. M. Sobol'. "Sensitivity analysis for non-linear mathematical models". In: *Mathematical Modelling and Computational Experiment* 1 (1993), pp. 407–414.

[175] E. Borgonovo. "A new uncertainty importance measure". In: *Reliability Engineering & System Safety* 92.6 (2007), pp. 771–784. DOI: 10.1016/j.ress.2006.04.015.

[176] S. Kucherenko, S. Tarantola, and P. Annoni. "Estimation of global sensitivity indices for models with dependent variables". In: *Computer Physics Communications* 183.4 (2012), pp. 937–946. DOI: 10.1016/j.cpc.2011.12.020.

[177] S. C. Cotter. "A screening design for factorial experiments with interactions". In: *Biometrika* 66.2 (1979), pp. 317–320. DOI: 10.1093/biomet/66.2.317.

[178] S. Marelli, C. Lamas, K. Konakli, C. Mylonas, P. Wiederkehr, and B. Sudret. *UQLab user manual – Sensitivity analysis*. Tech. rep. # UQLab-V1.3-106. Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019.

[179] N. Tsokanas, R. Pastorino, and B. Stojadinovic. "A comparison of surrogate modeling techniques for global sensitivity analysis in hybrid simulation". In: *Preprint submitted in engrXiv* (2021). DOI: 10.31224/osf.io/xujde.

[180] N. Tsokanas and B. Stojadinovic. "Design of a Controller for Physical Substructures in Stochastic Real-Time Hybrid Simulations". In: *Book of Abstracts: Engineering Mechanics Institute 2019 Conference*. California Institute of Technology, 2019, pp. 852–852.

[181] N. Tsokanas and B. Stojadinovic. "Design of an Actuation Controller for Physical Substructures in Stochastic Real-Time Hybrid Simulations". In: *Model Validation and Uncertainty Quantification, Volume 3*. Ed. by Z. Mao. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing, 2020, pp. 69–82. DOI: 10.1007/978-3-030-47638-0_8.

[182] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. New York: Springer, 2003.

[183] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[184] C. Lataniotis, D. Wicaksono, S. Marelli, and B. Sudret. *UQLab user manual – Kriging (Gaussian process modeling)*. Tech. rep. # UQLab-V1.3-105. Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019.

[185] V. Dubourg. "Adaptive surrogate models for reliability analysis and reliability-based design optimization". Doctoral thesis. Clermont-Ferrand, France: Université Blaise Pascal - Clermont-Ferrand II, 2011.

[186] D. E. Goldberg and J. H. Holland. "Genetic Algorithms and Machine Learning". In: *Machine Learning* 3.2 (1988), pp. 95–99.

[187] R. Schobi, B. Sudret, and J. Wiart. "Polynomial-chaos-based Kriging". In: *International Journal for Uncertainty Quantification* 5.2 (2015), pp. 171–193. DOI: 10.1615/Int.J.UncertaintyQuantification.2015012467.

[188] R. Schöbi, B. Sudret, and S. Marelli. "Rare Event Estimation Using Polynomial-Chaos Kriging". In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* 3.2 (2017), p. D4016002. DOI: 10.1061/AJRUA6.0000870.

[189] P. Kersaudy, B. Sudret, N. Varsier, O. Picon, and J. Wiart. "A new surrogate modeling technique combining Kriging and polynomial chaos expansions – Application to uncertainty analysis in computational dosimetry". In: *Journal of Computational Physics* 286 (2015), pp. 103–117. DOI: 10.1016/j.jcp.2015.01.034.

[190] R. Schöbi, S. Marelli, and B. Sudret. *UQLab user manual – Polynomial chaos Kriging*. Tech. rep. # UQLab-V1.3-109. Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019.

[191] M. Dabaghi and A. D. Kiureghian. "Simulation of orthogonal horizontal components of near-fault ground motion for specified earthquake source and site characteristics". In: *Earthquake Engineering & Structural Dynamics* 47.6 (2018), pp. 1369–1393. DOI: 10.1002/eqe.3021.

[192] S. Jia and Q. Li. "Friction-induced vibration and noise on a brake system". In: *2013 IEEE International Conference on Information and Automation (ICIA)*. Yinchuan, China, 2013, pp. 489–492. DOI: 10.1109/ICInfA.2013.6720348.

[193] N. Tsokanas, X. Zhu, G. Abbiati, S. Marelli, B. Sudret, and B. Stojadinović. "A Global Sensitivity Analysis Framework for Hybrid Simulation with Stochastic Substructures". In: *Frontiers in Built Environment* 7 (2021), p. 154. DOI: 10.3389/fbuil.2021.778716.

[194] X. Zhu and B. Sudret. "Replication-based emulation of the response distribution of stochastic simulators using generalized lambda distributions". In: *International Journal for Uncertainty Quantification* 10.3 (2020). DOI: 10.1615/Int.J.UncertaintyQuantification.2020033029.

[195] X. Zhu and B. Sudret. "Emulation of Stochastic Simulators Using Generalized Lambda Models". In: *SIAM/ASA Journal on Uncertainty Quantification* 9.4 (2021), pp. 1345–1380. DOI: 10.1137/20M1337302.

[196] Z. A. Karian and E. J. Dudewicz. *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. New York: Chapman and Hall/CRC, 2000. DOI: 10.1201/9781420038040.

[197] X. Zhu and B. Sudret. "Global sensitivity analysis for stochastic simulators based on generalized lambda surrogate models". In: *Reliability Engineering & System Safety* 214 (2021), p. 107815. DOI: 10.1016/j.ress.2021.107815.

[198] M. Freimer, G. Kollia, G. S. Mudholkar, and C. T. Lin. "A study of the generalized tukey lambda family". In: *Communications in Statistics - Theory and Methods* 17.10 (1988), pp. 3547–3567. DOI: 10.1080/03610928808829820.

[199] O. G. Ernst, A. Mugler, H.-J. Starkloff, and E. Ullmann. "On the convergence of generalized polynomial chaos expansions". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 46.2 (2012), pp. 317–339. DOI: 10.1051/m2an/2011045.

[200] B. Iooss and M. Ribatet. "Global sensitivity analysis of computer models with functional inputs". In: *Reliability Engineering and System Safety* 94.7 (2009), pp. 1194–1204. DOI: 10.1016/j.ress.2008.09.010.

[201] T. Browne. "Regression models and sensitivity analysis for stochastic simulators : applications to non-destructive examination". Doctoral thesis. Sorbonne Paris Cité, 2017.

[202] G. Chastaing, F. Gamboa, and C. Prieur. "Generalized Sobol sensitivity indices for dependent variables: numerical methods". In: *Journal of Statistical Computation and Simulation* 85.7 (2015), pp. 1306–1333. DOI: 10.1080/00949655.2014.960415.

[203] F. J. Anscombe and J. W. Tukey. "The Examination and Analysis of Residuals". In: *Technometrics* 5.2 (1963), pp. 141–160. DOI: 10.1080/00401706.1963.10490071.

[204] N. Tsokanas, L. Thielemans, B. Sputh, R. Pastorino, and B. Stojadinovic. "Three-level verification and validation approach for motorcycle subsystem testing using physical/virtual co-simulation". In: 2021. DOI: 10.3929/ethz-b-000473432.

[205] Y. Wang, Y. Li, Y. Song, and X. Rong. "The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition". In: *Applied Sciences* 10.5 (2020). DOI: 10.3390/app10051897.

Curriculum Vitae

Personal data

Name	Nikolaos Tsokanas
Date of Birth	July 03, 1993
Place of Birth	Athens, Greece
Citizen of	Greece

Education

Sept. 2018 – Dec. 2021	ETH Zurich, Zürich, Switzerland PhD student
Sept. 2012 – July 2018	University of Patras, Patra, Greece Diploma

Employment

Sept. 2020 – Oct. 2021	Visiting Researcher *Siemens Digital Industries Software*, Leuven, Belgium
Sept. 2019 – Jan. 2020	Visiting Researcher *Department of Mechanical Engineering, The University of Sheffield*, Sheffield, UK
Oct. 2017 – Apr. 2018	Internship *Airbus Defence & Space GmbH*, Friedrichshafen, Germany
Mar. 2017 – May 2017	Internship *Faculty of Aerospace Engineering, TU Delft*, Delft, The Netherlands

List of Publications

Articles in peer-reviewed journals:

[1] N. Tsokanas, D. Wagg, and B. Stojadinović. "Robust Model Predictive Control for Dynamics Compensation in Real-Time Hybrid Simulation". In: *Frontiers in Built Environment* 6 (2020), p. 127. DOI: 10.3389/fbuil.2020.00127.

[2] G. Abbiati, S. Marelli, N. Tsokanas, B. Sudret, and B. Stojadinović. "A global sensitivity analysis framework for hybrid simulation". In: *Mechanical Systems and Signal Processing* 146 (2021), p. 106997. DOI: https://doi.org/10.1016/j.ymssp.2020.106997.

[3] N. Tsokanas, X. Zhu, G. Abbiati, S. Marelli, B. Sudret, and B. Stojadinović. "A Global Sensitivity Analysis Framework for Hybrid Simulation with Stochastic Substructures". In: *Frontiers in Built Environment* 7 (2021), p. 154. DOI: 10.3389/fbuil.2021.778716.

[4] N. Tsokanas, G. Abbiati, K. Kanellopoulos, and B. Stojadinovic. "Multi-Axial Hybrid Fire Testing based on Dynamic Relaxation". In: *Fire Safety Journal* (2021), p. 103468. DOI: 10.1016/j.firesaf.2021.103468.

[5] N. Tsokanas, T. Simpson, R. Pastorino, E. Chatzi, and B. Stojadinovic. "Model Order Reduction for Real-Time Hybrid Simulation: Comparing Polynomial Chaos Expansion and Neural Network methods". In: *Preprint submitted in engrXiv* (2021). DOI: 10.31224/osf.io/h2bnm.

[6] N. Tsokanas, R. Pastorino, and B. Stojadinovic. "Adaptive model predictive control for actuation dynamics compensation in real-time hybrid simulation". In: *Preprint submitted in engrXiv* (2021). DOI: 10.31224/osf.io/c974v.

[7] N. Tsokanas, R. Pastorino, and B. Stojadinovic. "A comparison of surrogate modeling techniques for global sensitivity analysis in hybrid simulation". In: *Preprint submitted in engrXiv* (2021). DOI: 10.31224/osf.io/xujde.

Conference contributions:

[8] N. Tsokanas and B. Stojadinovic. "Design of a Controller for Physical Substructures in Stochastic Real-Time Hybrid Simulations". In: *Book of Abstracts: Engineering Mechanics Institute 2019 Conference*. California Institute of Technology, 2019, pp. 852–852.

[9] N. Tsokanas and B. Stojadinovic. "A stochastic real-time hybrid simulation of the seismic response of a magnetorheological damper". In: 2020. DOI: 10.3929/ethz-b-000462979.

[10] N. Tsokanas and B. Stojadinovic. "Design of an Actuation Controller for Physical Substructures in Stochastic Real-Time Hybrid Simulations". In: *Model Validation and Uncertainty Quantification, Volume 3*. Ed. by Z. Mao. Conference Proceedings of the Society for Experimental Mechanics Series. Cham: Springer International Publishing, 2020, pp. 69–82. DOI: 10.1007/978-3-030-47638-0_8.

[11] N. Tsokanas and B. Stojadinovic. "Evaluation of the Seismic Response of a Structure with an attached Pendulum using Stochastic Real-Time Hybrid Simulation". In: ETH Zurich, Institute of Structural Engineering, 2020, p. 18748. DOI: 10.3929/ethz-b-000474269.

[12] N. Tsokanas, L. Thielemans, B. Sputh, R. Pastorino, and B. Stojadinovic. "Three-level verification and validation approach for motorcycle subsystem testing using physical/virtual co-simulation". In: 2021. DOI: 10.3929/ethz-b-000473432.

[13] N. Tsokanas, R. Pastorino, B. Peeters, and B. Stojadinovic. "Development of an adaptive model predictive reference tracking controller for hard real-time hybrid simulation". In: 2021. DOI: 10.3929/ethz-b-000473422.

About the author

Nikolaos Tsokanas received his diploma in August 2018 from the Faculty of Electrical Engineering & Computer Science at the University of Patras, Greece. In September 2018, he joined the Chair of Structural Dynamics and Earthquake Engineering at the Institute of Structural Engineering (IBK) of ETH Zürich in Switzerland, as a doctoral student. Nikolaos is a Marie Curie fellow and his research work is part of the DyVirt (Dynamic Virtualization: modeling performance of engineering structures) program. DyVirt is a project funded from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 764547. During his Ph.D., Nikolaos visited and conducted part of his research work in the Dynamics Research Group at the University of Sheffield, in the UK and in the Model-based System Testing group at Siemens Digital Industries Software, in Belgium. Nikolaos' professional interests lie in the broad field of experimental testing and numerical simulation, automatic control system theory and uncertainty quantification.

DOI 10.3929/ethz-b-000504284
ISBN 978-3-907234-67-9