TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM SCHOOL OF ENGINEERING AND DESIGN

FLEXIBILITY-ORIENTED DESIGN GUIDELINES FOR PEDESTRIAN LEG IMPACT ON THE BASIS OF A LOW-FIDELITY VEHICLE FRONT-END MODEL

STEFANO CHIAPEDI

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Kai-Uwe Bletzinger

Prüfer der Dissertation:

1. Prof. Dr.-Ing. habil. Fabian Duddeck
2. Prof. Giovanni Belingardi

Die Dissertation wurde am 12.07.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 09.11.2021 angenommen.

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM SCHOOL OF ENGINEERING AND DESIGN

FLEXIBILITY-ORIENTED DESIGN GUIDELINES FOR PEDESTRIAN LEG IMPACT ON THE BASIS OF A LOW-FIDELITY VEHICLE FRONT-END MODEL

STEFANO CHIAPEDI

2021

Doktor-Ingenieur (Dr.-Ing.)
Associate Professorship of Computational Mechanics

SUPERVISOR:
Prof. Dr.-Ing. habil. Fabian Duddeck

Dedicated to mamma and papà,
my first and most important *doctores*, from Latin teachers, of life.

Dedicato a mamma e papà,
i miei primi e più importanti *doctores*, dal latino insegnanti, di vita.

One of the things that just blows me away about universities is that
no one ever tells students why they should write something.
*"*Write this assignment.*" But why are you writing?*
Well, you need the grade. No! You need to learn to think.
Because thinking makes you act effectively in the world.
Thinking makes you win the battles you undertake. [. . .]
It is the most powerful weapon you could possibly provide someone with.

— Jordan B. Peterson[1]

1 During *Lecture 6: Story and MetaStory (Part 2)* of 2017 class *Maps of Meaning: The Architecture of Belief*.

ABSTRACT

The ever-rising complexity of product requirements raises challenges in the development process: complex technical problems must be solved in the concept phases with an often incomplete or uncertain set of information. For instance, the recent introduction of a new legform impactor for pedestrian protection, called aPLI, represents an additional challenge in the design of the vehicle front-end. Not designing a robust concept may generate major issues in the subsequent development phases. I propose a method to support the concept designers of the vehicle front-end with guidelines that offer flexibility of design, while fulfilling the system requirements for pedestrian leg impact. The guidelines are based on a computationally-efficient low-fidelity model of the front-end and represent, mathematically, multiple axis-aligned boxes, lying in the feasible region of the design space. They empower the designers with know-how on the required structural properties of the front-end sub-systems. Offering multiple compromises among the required properties, this know-how can greatly help to reduce future design iterations. Despite the significant computational effort in high-dimensional spaces, the proposed method was successfully applied on an industrial design problem.

ZUSAMMENFASSUNG

Die ständig steigende Komplexität der Produktanforderungen stellt den Entwicklungsprozess vor Herausforderungen: komplexe technische Probleme müssen in den Konzeptphasen mit häufig unvollständigen oder unsicheren Informationen gelöst werden. Beispielsweise stellt der kürzlich eingeführte Bein-Impaktor für den Fußgängerschutz aPLI eine zusätzliche Herausforderung bei der

Gestaltung der Fahrzeugfront dar. Wenn kein robustes Konzept ausgelegt wird, kann dies in den nachfolgenden Entwicklungsphasen zu erheblichen Problemen führen. Ich schlage eine Methode vor, um die Konzeptdesigner der Fahrzeugfront mit Richtlinien zu unterstützen, die Flexibilität bei der Auslegung bieten und gleichzeitig die Systemanforderungen für den Aufprall von Fußgängern erfüllen. Die Richtlinien basieren auf einem rechnerisch effizienten Low-Fidelity-Modell der Fahrzeugfront und entsprechen mathematisch mehreren achsenorientierten Boxen, die im zulässigen Bereich des Entwurfsraums liegen. Sie vermitteln den Designern Know-how über die erforderlichen strukturellen Eigenschaften der Fahrzeugfront-Subsysteme. Dieses Know-how bietet mehrere Kompromisse bei den erforderlichen Eigenschaften und kann erheblich dazu beitragen, zukünftige Iterationen im Auslegungsprozess zu reduzieren. Trotz des relativ hohen Rechenaufwands in hochdimensionalen Räumen wurde das vorgeschlagene Verfahren erfolgreich auf ein industrielles Auslegungsproblem angewendet.

ACKNOWLEDGMENTS

The research presented in this thesis was conducted during my time as Research Assistant at the Associate Professorship of Computational Mechanics of the Technical University of Munich, between March 2017 and February 2020. The research was funded by and carried out in cooperation with Audi AG.

In the thesis, I often make use of the singular first-person pronoun, *I*, to take responsibility for the work done and the opinions therein expressed. In fact, only *one* author is named on the cover page. The plural version, *we*, would be, then, quite pretentious: *pluralis maiestatis* should be left to Cicero and few others. However, *we* would be more appropriate: not for pretentiousness, rather for honesty and gratitude. The thesis is, indeed, single-author, yet multi-contributor. The contributions were so many and so strong that this thesis would simply not have been possible without them.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Dr.-Ing. habil. Fabian Duddeck. By giving me the opportunity to work, first, on a Master's thesis and, then, a PhD under his supervision, he probably influenced my life in the past five years in a way no one else did. Not only that, he enriched me with countless input and feedback, not least the revision of this thesis. I will especially remember the long discussions we had in his office, which always finished in the same way: with me taking lots of notes and asking myself "*Stefano, why didn't you talk to him earlier?*". Those conversations were true sources of inspiration.

My deepest gratitude also goes out to my supervisor at Audi AG, Dr.-Ing. Andreas Koukal. He took the first steps that lead to this research and, once I started, constantly guided me through the tortuous path of the PhD. His readiness to find time to answer my frequent questions was extraordinary. He also made sure that everything was running smoothly both inside- and outside-the-office. Finally, he supported me even when he wasn't completely

convinced about my proposals. I must say that I, then, often started the next meeting with "*Andreas, you were right...*".

I would like to thank my mentor, Prof. Giovanni Belingardi. I first met him in 2013, during my Bachelor's studies, where I had the chance to appreciate both his technical competency and interpersonal skills: my oral exam did not go as I wanted and he gave me a second chance. Several years later, during this research, I had further proof of both qualities: he devoted time to carefully understand my research and supported me with both technical and motivational feedback.

I would like to thank the head of the pedestrian protection department at Audi AG, Birgit Arneth. She trusted me during the job interview, as well as over the duration of the work, actively showing appreciation for it. Furthermore, her encouragement to learn German proved valuable for my acclimatization in Germany and professional development.

I would like to thank Enrico Vogelsang, Christian Teichmann and the rest of the pedestrian protection team at Audi AG. Their experience on the topic and active testing of the method on real design problems were useful in identifying weak spots and determining corrections and improvements.

I would like to thank Koushyar Komeilizadeh, fellow Research Assistant at the chair. He advised me on surrogate models and helped me navigate the obscure world of mathematical approximations of black-box functions. Without him, I would have probably not dared entering that world.

I would like to thank the students who contributed to this research with internships and theses: Nilesh Thakare, Siddharth Murali, Konstantinos Kamariotis, Thi vu Linh Ngo. I learned a great deal both with and from them. Furthermore, their passion, curiosity and brilliant input enhanced this research significantly.

I would like to thank the fellow Research Assistants at the chair. Although their research topics were, in some cases, far from mine, the discussions with them were always stimulating –and funny.

I would like to thank Christoph Stöckle, fellow doctoral candidate at Audi AG. He introduced me to tree-traversal algorithms and, in so doing, helped me to solve a problem –and generate ten others.

I would like to thank my brother Umberto and my friends. It is because of the energy-recharging time extra-thesis that I spent with them, that I could stand the energy-consuming time on-the-thesis.

The gratitude to my parents goes way beyond –and before– this thesis. However, besides their unconditional love and support during this period, I can mention two ways they affected this work. Firstly, no longer having to answer their recurring question "*How long before you finish writing the thesis?*" gave me a strong motivation to actually finish writing it. Secondly, I remember mamma telling me, when I was a child, "*I wish you become a doctor*". I am sure, this isn't the type of doctor she meant, but, eventually, it might happen.

Munich, May 2021 Stefano Chiapedi

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACRONYMS

ADAS	Advanced Driver-Assistance Systems
ACL	Anterior Cruciate Ligament
ANN	Artificial Neural Network
aPLI	advanced Pedestrian Legform Impactor
BA	Balanced Accuracy
BFS	Breadth-First Search
BLUE	Best Linear Unbiased Estimator
CAD	Computer-Aided Design
C-NCAP	Chinese New Car Assessment Programme
CORA	Correlation and Analysis
CPU	Central Processing Unit
DDS	Depth-bounded Discrepancy Search
DFS	Depth-First Search
DOE	Design of Experiments
DOF	Degree of Freedom
EI	Expected Improvement
Euro NCAP	European New Car Assessment Programme

FE	Finite Element
FN	False Negative
FP	False Positive
FlexPLI	Flexible Pedestrian Legform Impactor
GP	Gaussian Process
GUI	Graphical User Interface
HBM	Human Body Model
HFM	High-Fidelity Model
HPC	High-Performance Computing
ILDS	Improved Limited Discrepancy Search
LDS	Limited Discrepancy Search
LFM	Low-Fidelity Model
LHS	Latin Hypercube Sampling
MAE	Mean Absolute Error
MCC	Matthews Correlation Coefficient
MCL	Medial Collateral Ligament
MLE	Maximum Likelihood Estimation
MOR	Model Order Reduction
MSE	Mean Squared Error
NMMSO	Niching Migratory Multi-Swarm Optimiser
NVH	Noise, Vibration and Harshness
PCA	Principal Component Analysis
PCL	Posterior Cruciate Ligament
PDF	Probability Density Function
PPV	Positive Predictive Value
PSO	Particle Swarm Optimization
RMSE	Root Mean Squared Error
SUV	Sport Utility Vehicle
SVD	Singular Value Decomposition

SVM	Support-Vector Machine
THUMS	Total Human Model for Safety
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
TRL	Transport Research Laboratory
UBM	Upper Body Mass

Part I

THE MOTIVATION.

1 INTRODUCTION

[. . .] self-propelled vehicles on public roads
must be preceded by a man on foot
waving a red flag and blowing a horn.

— The Locomotive Act, 1865[1]

Vehicle development is a complex process lasting several years. Broadly speaking, it can be divided in five phases: product definition, concept development, series development, industrialization/testing and production [133], as schematized in Figure 1.1. This thesis addresses issues that span across concept and series development –with a stronger focus on the former one– and aims to reduce some of the complexity that arises here.

The sources of complexity in vehicle development are various. Certainly, one is the *technical complexity*: for instance, minimizing the aerodynamic drag coefficient to reduce the power needed for propulsion raises some technical challenges [14]. In order to solve them, some design iterations are unavoidable: a virtual model is built, simulated and analyzed and measures for improvement are identified, leading, thus, to the construction of a new model. Therefore, some trial-and-error can be considered inherent in the design process.

To the technical complexity of a functional requirement –e. g., aerodynamic drag– a *system complexity* adds up. The system, intended as the vehicle, must satisfy several functional requirements

1 Enacted by the Parliament of the United Kingdom, it was arguably the very first "pedestrian protection" regulation.
The "*man on foot*", known as the *stalker*, had to walk 60 yards (55 m) ahead of the vehicle to ensure the safety of pedestrians and animals. In total, three people were required for each vehicle: one to steer, one to stoke the engine and one to stalk. The law was not repealed until 1896; nevertheless, the use of the red flag was made optional from 1878. The Act also limited the vehicle speed to 4 mph (6 km/h). Vehicles at that time were mainly steam carriages, as the first motorcar was Benz Patent-Motorwagen, patented in Mannheim (Germany) in 1886.

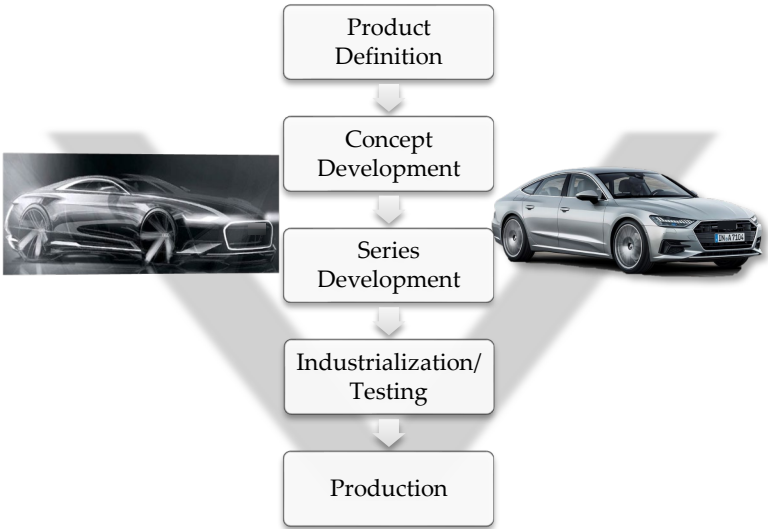


Figure 1.1: Schematic overview of the vehicle development process.

–applying frequently to the same package– as well as cost requirements, manufacturing requirements, etc., constituting alltogether the *system requirements*. Considering, for instance, the vehicle front-end, various functional requirements can be named: aerodynamics, thermal management, passive safety, active safety, reparability, propulsion, lighting and so on. Hence, not only one technical aspect must be solved: the combination of all technical and non-technical ones multiplies the complexity. The design of high-complex systems is target of systems engineering [43, 127]. The fundamental approach of systems engineering is not to design parts fitting together into a system, but to design functioning systems composed of parts. Here, the system is conceived as a set of parts acting together to achieve the system goals. Systems engineering focuses, hence, on the system as a whole, in all its interdisciplinary facets: it guides the development process through a clear identification of system goals, an efficient coordination of the multiple disciplines and a careful tracking of the goals fulfillment. A model often employed in systems engineering is the V-model [125], or Vee model, originated in software development.

It takes its name from its recognizable schematic arrangement, depicted as a "V". In the V-model, the requirements on the system are decomposed into requirements on the sub-systems and, then, on the components, in a hierarchical approach. Design occurs on the components-level; then, validation and integration follows bottom-up. Technical complexity is reduced, because larger problems are broken down into smaller sub-problems, ultimately easier to be solved. For each system requirement, decoupling of the requirements on sub-systems and components is essential to ensure independent design and testing. Still, due to the multiple system requirements, some components may be relevant for many of them.[2] Hence, teams from different disciplines must collaborate together to co-design and analyze these sub-systems/components.

Related to the above two complexities, there is, then, the *process complexity*. The fact that:

- design iterations are unavoidable,
- multiple design teams must collaborate on the same sub-systems and may have contrasting requirements,
- many decisions on sub-systems and components must be still finalized in the concept phase,

implies that the initial design will evolve during the development process. This is certain. Uncertain is *how* it will evolve. Therefore, the product development process itself raises uncertainty in the early stages, fundamentally associated with a lack of knowledge [140].[3]

To tackle the above mentioned complexities, working on the concept phase is crucial. Here, concept engineers must take major, conceptual decisions on the package arrangement and properties. The decision-process should be supported with guidelines that:

2 Think of the chassis: relevant, e. g., for passive safety, but also Noise, Vibration and Harshness (NVH)[23].

3 N.B.: Uncertainty is also present in later stages, not only in the concept ones: e. g., caused by manufacturing tolerances. Yet, this can be modeled with a probabilistic approach.

- prescribe the technical properties that the designs must have in order to fulfill the system requirements –*technical complexity*,
- decouple, as much as possible, the system requirements on the various sub-systems/components –*system complexity*,
- quantify how much flexibility the designs have to tolerate future changes[4] –*process complexity*.

Such guidelines are the object of this work. I will focus on the system requirements concerning pedestrian leg impact.

1.1 PEDESTRIAN LEG IMPACT

According to the World Health Organization [137], about 311 thousand pedestrians worldwide lost their lives in 2016 as a result of road traffic accidents. This represents about 23% of all fatalities on the roads. In the European region, the percentage rises to 27%, meaning 23 thousand pedestrian fatalities or 2.5 per 100 thousand inhabitants. In Germany, in 2019, 417 pedestrians died on the roads and about 30 thousand ones got injured [116]. In the last two decades, from 2000 to 2019, pedestrian fatalities decreased in Germany of 58%, while injuries of 23%. Hence, an encouraging trend can be seen, at least in Germany, about pedestrian fatalities.

The last two decades have, indeed, seen a lot of advancements in the field of passive pedestrian protection. In 2003, the first regulation regarding pedestrian protection was enacted by the European Commission [28]. It specified four subsystem tests: lower legform to bumper, upper legform to bonnet leading edge, child headform to bonnet top and adult headform to bonnet top. The respective impactors are illustrated in Figure 1.2. I will focus in the following on the lower leg impact, abbreviated simply as leg impact. For further information on pedestrian protection see [31, 75], and on pedestrian impact bio-mechanics see [138, 139].

Studies on the interaction between the human leg and the vehicle front-end during pedestrian impact started already in the

4 In other words, how much the designs can change and still satisfy the system requirements.

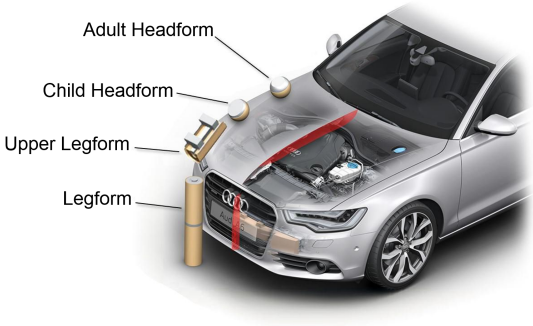
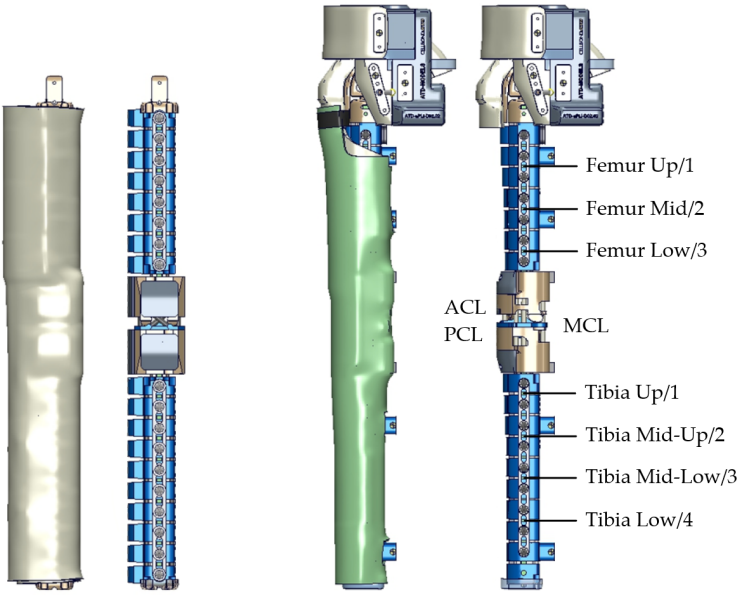


Figure 1.2: Sub-system impactors for pedestrian protection, as prescribed by [28]. From [73].

80s: for instance, Cesari et al. [15] used 20 cadavers to analyze pedestrian leg injuries, the effect of bumper height and the forces exchanged during impact. However, it was only with the development of sub-system impactors and the definition of repeatable and reproducible tests that vehicle design for the pedestrian leg impact gained momentum. The first legform impactor, the TRL impactor, takes its name from UK's Transport Research Laboratory that developed it. This impactor, shown in Figure 1.2, is the one adopted in 2003 regulation. It constitutes of two rigid, steel rods representing tibia and femur connected by a joint representing the knee. The entire legform is wrapped with foam, represent the flesh, and neoprene, representing the skin. The injury criteria measured by the impactor are bending and shearing of the knee and the acceleration of the upper tibia [26].

Rigid rods cannot represent the characteristic bending of human bones [68]. The lack of bio-fidelity of the Transport Research Laboratory (TRL) impactor induced, hence, researchers of the Japan Automobile Research Institute to develop a more bio-fidelic pedestrian legform impactor, then called Flexible Pedestrian Legform Impactor (FlexPLI) [69, 70]. This impactor is shown in Figure 1.3a. As the name suggests, it presents flexible bones, made of fiberglass reinforced plastic, which show a human-like bending characteristics under impact loading. A mechanical knee replicates, with springs and cables, the human knee ligaments. Finally, flesh and



(a) FlexPLI (b) aPLI

Figure 1.3: Legform impactors currently in use for the sub-system test of the pedestrian leg impact. From [117].

skin are represented by multiple layers of rubber and neoprene. FlexPLI's weight amounts to about 13 kg. Three strain gauges are positioned on the femur to measure the bending moment at different sections and four on the tibia. Potentiometers measure the elongations of Medial Collateral Ligament (MCL), Anterior Cruciate Ligament (ACL) and Posterior Cruciate Ligament (PCL). Hence, the FlexPLI allows to measure all relevant human-like leg injuries. It was adopted, first, in 2014 by the European New Car Assessment Programme (Euro NCAP) consumer test [27] and in 2015 by the UN Regulation No 127 [124], valid also in Europe.

The FlexPLI is very bio-fidelic in representing, mechanically, the human leg. Yet, it misses completely what is on top of the leg, namely the upper human body. However, the upper body plays a significant role during impact with passenger cars. In fact, its mass induces, by inertia, a large bending of the femur before it reaches contact with the vehicle bonnet or windshield. Therefore, the FlexPLI may give rise to very unrealistic kinematics with low- or with high-bumper vehicles [54]. The same researchers of the Japan Automobile Research Institute, who worked on the FlexPLI, developed an improved pedestrian legform impactor, so-called advanced Pedestrian Legform Impactor (aPLI) [55]. This is largely based on the FlexPLI, but adds a 12kg simplified upper-body part, which is connected to the femur via a hip-joint. Furthermore, it presents an improved mass distribution, ligaments positioning and routing and rounded knee impact area [53]. A major change with respect to the FlexPLI in the loading condition is the height from ground: 25 mm instead of 75. The aPLI will be adopted in 2022 by the Chinese New Car Assessment Programme (C-NCAP) [18].

The introduction of the aPLI raises some challenges in vehicle design: first of all, it is new and exhibits a new legform kinematics that must be analyzed; then, at least for the Chinese market, both FlexPLI –for regulation– *and* aPLI requirements must be fulfilled. Supporting the concept engineers with quantitative guidelines on the sub-systems/components requirements for both aPLI and FlexPLI would lead to a smoother and more robust concept phase.

A detailed, high-fidelity model of the vehicle front-end is very complex and takes a long time to simulate. Furthermore, most of

the parameters that can be conveniently defined have little effect on the legform injury criteria [16]. Therefore, such a model is not suitable to conceptual investigations: rather, a reduced, low-fidelity model of the vehicle front-end for pedestrian leg impact should be used in order to derive the desired guidelines.

1.2 STRUCTURE OF THE THESIS

The thesis will be structured as follows:

- in Chapter 2, I will report which studies addressed the topic of this work and highlight which points remain unclear or unsolved;
- in Chapter 3, I will state the high-level research question that this work aims to answer, as well as the low-level ones that define the single objectives;
- in Chapter 4, I will propose a low-fidelity modeling technique of the vehicle front-end for pedestrian leg impact and a possible parametrization approach;
- in Chapter 5, I will propose a work-flow to compute flexibility-oriented design guidelines with computationally-expensive black-box functions;
- in Chapter 6, I will present a validation approach for all proposed methods;
- in Chapter 7, I will apply all proposed methods on a real-world industrial design problem;
- in Chapter 8, I will reflect on the achievement of the thesis' objectives and discuss advantages and disadvantages of the proposed methods;
- in Chapter 9, finally, I will summarize the work done, condense the key take-away messages of this work and share my outlook for possible further research.

2 STATE OF THE ART

There is no conversation more boring
than the one where everybody agrees.

— Attributed to Michel de Montaigne[1]

As introduced in Chapter 1, the focus in this thesis lies in the identification of a method to develop design guidelines for the pedestrian leg impact, on the basis of a Low-Fidelity Model of the vehicle front-end. In this chapter, I investigate to what extent previous methods address the topic and which questions remain unanswered that this work considers relevant to undertake. In the first part, I concentrate on computationally-efficient approaches for the modeling of the vehicle front-end for pedestrian leg impact. In the second part, I focus on methods to establish flexibility-oriented design guidelines for black-box functions.

2.1 VEHICLE FRONT-END MODELING FOR PEDESTRIAN LEG IMPACT

Computer models of car-pedestrian impact go back approximately three decades [52]. Since then, pedestrian protection has undergone several changes, from the introduction of regulations to the development of new subsystem impactors. Therefore, several studies have addressed the design of front-ends for pedestrian protection. Here, I will focus on those concerning the leg impact.

Depending on the scope of the investigation and the impactor used, different categories of simulation models are used to represent the vehicle front-end:

DETAILED, HIGH-FIDELITY FINITE ELEMENT MODEL: it consists of a Finite Element (FE) discretization of all the components present in the vehicle front-end.

1 In: Michel de Montaigne. *Essais*. 1595.

REDUCED, HIGH-FIDELITY FINITE ELEMENT MODEL: it consists of a FE discretization of *a subset of* the components present in the vehicle front-end, which are considered the most relevant ones for the analysis.

REDUCED, LOW-FIDELITY FINITE ELEMENT MODEL: it consists of finite, virtual elements –not faithful to the real-word– representing the stiffness characteristics of a subset of the vehicle front-end components in specific loading conditions.

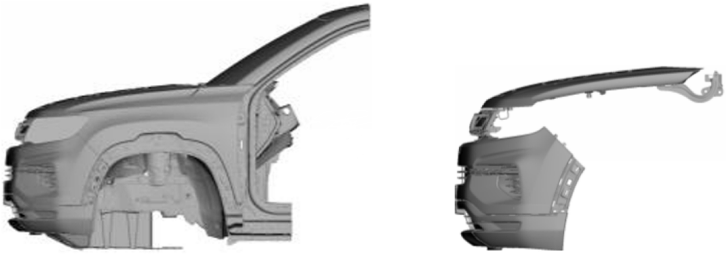
REDUCED, LOW-FIDELITY MULTI-BODY MODEL: it consists of idealized inter-connected rigid and flexible bodies, representing a subset of the vehicle front-end components. It is generally employed with pedestrian dummy models for full pedestrian kinematics. Not being suitable to legform impactor and analysis of its injury criteria, I will not discuss it further in the following. For information, see [1, 62, 79].

DETAILED, HIGH-FIDELITY FE MODEL It is the standard model used for assessment of the vehicle performance in pedestrian leg impact throughout the development process, as exemplarily shown in Figure 2.1a. In literature, it is used mostly for targeted optimization of geometrical and stiffness characteristics of the lower stiffener and bumper energy absorber, such as thickness, deformation space and relative position between the two. Han and Lee [45], Svoboda and Kuklik [119] and Karimullah et al. [61] focus on the TRL legform impactor's injury criteria, Lv et al. [81] consider both TRL and FlexPLI while Scattina et al. [107] use a full pedestrian model, consisting of a combination of the Hybrid III dummy and the Lower Limb Model for Safety [2].
All studies agree on the importance of lower stiffener and bumper energy absorber design for the pedestrian leg impact. Yet, those studies are quite limited –in other words, very targeted– in scope. Due to the complexity of the models, few parameters could be defined. Furthermore, these parameters could not implement broad conceptual changes, while the long computational time does not allow to run a large sampling in a reasonable amount of time. Thanks to its accuracy, a detailed, High-Fidelity Model (HFM) is suitable to fine shape optimization of some relevant front-end

components for final design adjustments at rather late stages of the development process, as done for instance by Wetzstein et al. [135].

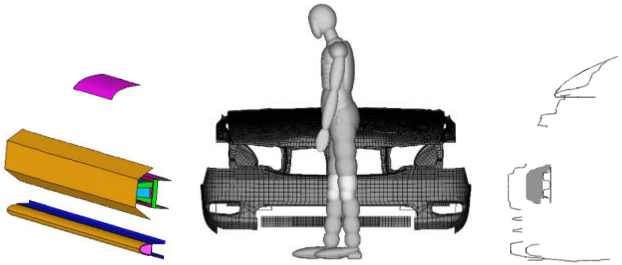
REDUCED, HIGH-FIDELITY FE MODEL It decreases the number of degrees of freedom of the model and, therefore, increases its computational efficiency. This is done by simply neglecting components non-significant to the pedestrian leg impact, as depicted, e. g., in Figure 2.1b. Typical components included in the model are the lower stiffener, bumper energy absorber and bonnet. Investigations generally involve the optimization of the relative positions among these three load regions and of the deformation space in front of the bumper beam. Nanda et al. [91] –whose model is shown in Figure 2.2a– optimize TRL impactor's injury criteria, Nakane et al. [90] –in Figure 2.2b– compare the influence of the design parameters on a pedestrian dummy and the Total Human Model for Safety (THUMS), Lee et al. [77] –in Figure 2.2c– optimize FlexPLI's injury criteria adding the bonnet angle of attack as design variable. The work of Fu et al. [32] is the first optimization attempt with the aPLI. Their Sport Utility Vehicle (SUV) model, shown in Figure 2.1b, comprises also a mid-up stiffener, between the bumper beam and the bonnet, and consists of about 310 thousand elements. They optimize the longitudinal and vertical positions of lower stiffener, bumper beam and mid-up stiffener with independent, rigid shifts and find that the longitudinal positions of lower stiffener and bumper beam are the most significant parameters for aPLI's injury criteria.

The models above gain in computational efficiency with respect to detailed HFMs, yet they lose some accuracy. The comparison of tibia bending moment and ligament elongation curves between the two models in [32] shows a non-negligible loss of accuracy. Furthermore, reduced HFMs retain the parametrization limitations of detailed ones: the stiffness characteristics can hardly undergo substantial changes and, in order to implement geometrical changes, either morphing needs to be used or the load regions must be split so as to be able to move independently. In both cases, accuracy may be further compromised. Lastly, the computational effort is still significant so that only few parameters may be defined.



(a) Detailed HFM, from [32] (b) Reduced HFM, from [32]

Figure 2.1: Detailed and reduced FE High-Fidelity Models (HFMs) of a vehicle front-end for pedestrian leg impact simulation, from Fu et al. [32].

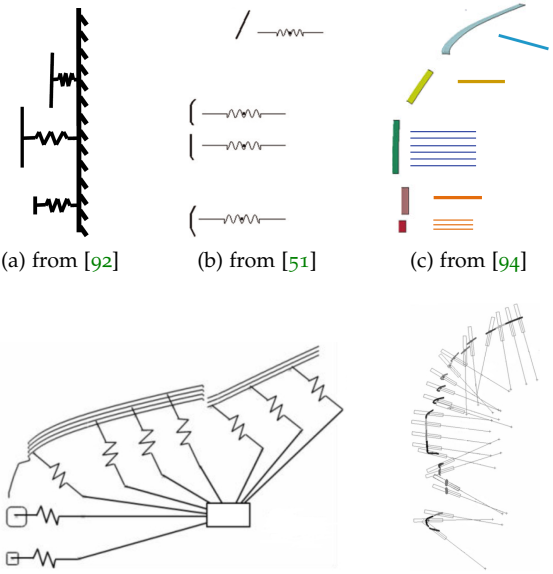


(a) from [91] (b) from [90] (c) from [77]

Figure 2.2: Reduced FE High-Fidelity Models (HFMs) of vehicle front-ends for pedestrian leg impact simulation, from literature.

REDUCED, LOW-FIDELITY FE MODEL It further decreases the computational effort of pedestrian leg impact simulations and facilitates the definition of parameters. Early work with lumped mass-spring models was carried out for front crash by Kamal [60] with a linear spring force-deformation characteristics and by Hollowell [49] with a non-linear one. For pedestrian leg impact, Neal [92] develops a vehicle front-end model consisting of three rigid plates supported by non-linear translational springs, as shown in Figure 2.3a, to run investigations with the TRL impactor. The combination of rigid plate and deformable spring represents a front-end load level: in this case, they correspond to lower stiffener, bumper energy absorber and grille. The force-deformation characteristics of each spring is calibrated to the one of the related detailed HFM during impact with the impactor. He optimizes both geometrical and structural parameters: longitudinal and vertical position of each load level, deformation space and crush force of the middle one and Young's modulus of the lower one. The abstractness of the model allows to easily implement changes. Neal et al. [93] update the model with an inclined grille and introduce a Graphical User Interface (GUI) to provide guidelines to concept engineers: based on multiple Kriging response surfaces, it graphically shows the effect of any change in design parameter values on the TRL impactor's injury criteria. Huang et al. [51] identify the impactor's knee as delicate spot and split the middle load region in two at the TRL impactor's knee height, as shown in Figure 2.3b, to better reproduce the forces acting above and below it.

The lack of bio-fidelity of the TRL legform impactor induced many researchers to evaluate the pedestrian-friendliness of vehicle front-end designs directly with FE Human Body Models (HBMs). Accordingly, the LFMs have been enhanced with additional load levels. Takahashi et al. [121] adopt a fourth load level representing the bonnet, modeled as a longitudinal translational spring, free to move in the vertical direction. Nie et al. [94] add a fifth load level to represent a mid-low one between lower stiffener and bumper energy absorber, as shown in Figure 2.3c. To account for local deformation, they model lower and middle load levels with shell elements –instead of a rigid plate– supported by distributed beam



(a) from [92] (b) from [51] (c) from [94]

(d) from [3] (e) from [89]

Figure 2.3: Reduced FE Low-Fidelity Models (LFMs) of vehicle front-ends for pedestrian leg impact simulation, from literature.

elements. The bending behavior of the beam elements is, though, not discussed. The force-deformation characteristics is recorded by impact of small, rigid cylindrical impactors against each level separately. However, the prescribed impactor motion may not be representative of the actual one of the HBM. Nie et al. [95] use this model to compute Sobol's sensitivity indices [115] for both geometrical and stiffness parameters: longitudinal and vertical position of lower and upper load levels and height of the middle one, yield force and deformation space for lower and middle levels and a scaling factor for the remaining ones. Furthermore, they extend the GUI tool in [93] with the new vehicle LFM and the HBM, in place of the TRL impactor. Finally, they perform a stochastic analysis of the uncertainty propagation between input and output variables, so that, given a Probability Density Function (PDF) of the design parameters, they compute both the average resulting injury criteria values and their confidence intervals. Asanuma et al. [3], whose model is shown in Figure 2.3d, represent both bonnet and windshield with multiple translational springs and rigid plates that contact specific parts of the human body to account for the local stiffness in these areas.

The introduction of the Flexible Pedestrian Legform Impactor (FlexPLI) brought to an enhancement of the modeling of the vehicle front-end local deformation to cope with the flexible behavior of the legform. Mößner et al. [89] show that an LFM consisting of only five load levels moving in the longitudinal direction is not accurate enough for FlexPLI. They use, therefore, 34 load levels –in Figure 2.3e– oriented according to their direction of deformation, as recorded during HFM impact with FlexPLI. Mößner [88] uses such model to investigate the ideal loading and unloading stiffness distribution for a particular front-end styling. He finds that a large intrusion in the upper region and a low intrusion in the lower one help keeping the leg straight and that the unloading phase in most regions is crucial for FlexPLI. He does not investigate the influence of geometrical parameters.

The models above, constituting of only few elements, provide a significant boost of computational efficiency. Most of the effort is due to the impactor or HBM. Accuracy is, clearly, affected by the simplification, yet the comparison of injury criteria with LFM

and HFM in [89] is very promising. The simplification tailors the accuracy on a specific pedestrian leg impact's dynamics and is not suitable to different front-end investigations. For instance, in [89], the model is validated for small variations of the impactor's kinematics: e. g., −30 to 50mm of impactor's height variation. Nevertheless, no LFM has been validated with aPLI yet.
Additionally, a clear trend can be seen in literature to constantly raise the number of load levels used to represent the front-end. Nonetheless, no investigation prescribes yet how many levels are necessary to reach satisfactory accuracy results.
Furthermore, a reduced LFM allows an easy, quick and understandable definition of parameters. However, most studies have either focused on specific load levels or used ideal stiffness distributions. Therefore, a parametrization with realistic geometrical and stiffness variations distributed over the whole front-end has not been investigated yet.
Finally, the GUI tool to provide guidelines in [93, 95] is very useful to understand how much influence each design parameter has on the injury criteria –as well as their uncertainty– or whether *a single design* satisfies the requirements; however, they do not provide information on which *set of designs* satisfies the requirements and which the limit values for the parameters are.

2.2 FLEXIBILITY-ORIENTED DESIGN GUIDELINES FOR BLACK-BOX FUNCTIONS

One, single design candidate that fulfills the requirements –as it can be obtained by a classical optimization run or by simple trial-and-error– is certainly an interesting information, yet not very useful in the early phases of the development process. Here, in fact, the design still undergoes many changes until the final design is achieved.[2] This makes it difficult to foresee how the final design will look like. Hence, uncertainty emerges in the initial development phases due to the lack of knowledge about the evolution of the design during the subsequent ones. This uncertainty can be mitigated, more than by the information that *one*

2 Even more so if the design candidate is a reduced Low-Fidelity Model.

design candidate satisfies the requirements, by the information *in which interval a set of* design candidates satisfies the requirements. Furthermore, in order to guide the designers through the development process, the interval should provide enough flexibility to account for possible variations. Therefore, I am interested in *flexibility-oriented design guidelines*.

A renowned approach that considers sets of designs, instead of single ones, is named *set-based concurrent engineering*, described by Sobek II et al. [114]. Here, a broad set of feasible[3] designs is chosen in the early stages of the development process and gradually refined over time as new information arises. If, on the one hand, a set of designs provides more than just one solution to the feasibility problem, on the other hand, it does not span any *interval*. It contains finite, discrete design candidates that allow no or little deviation, hence no flexibility.

The term *flexibility* is first introduced by Grossmann and Morari [42] in the field of design of chemical processes under uncertain conditions. Halemane and Grossmann [44] formulate the *flexibility test problem* that consists in evaluating whether the system is feasible *in a specific parameter interval*. An interval of values for each parameter generates an axis-aligned hyper-rectangle[4] in the parameter space –as for, e. g., the design space. Instead of evaluating whether a specific hyper-rectangle is contained in the feasible region –as done in the flexibility test problem– Swaney and Grossmann [120] evaluate the size of the largest hyper-rectangle inscribed in the feasible region and centered *around a specific design*, as shown in Figure 2.4. The hyper-rectangle is proportional to the original parameter ranges –in other words, it keeps the same aspect ratio of the design space. The measure of the size of the hyper-rectangle is named *flexibility index*. This index is very useful to assess the flexibility of a specific design. It can be used, for instance, to quantify the robustness of a design at late stages of the development process. However, it does not assess the overall flexibility inside the design space. In the early stages, instead, it is relevant to identify the interval where the flexibility is largest, independent on any specific design.

3 That is, fulfilling the requirements.

4 For sake of simplicity, I will imply axis-aligned in the following.

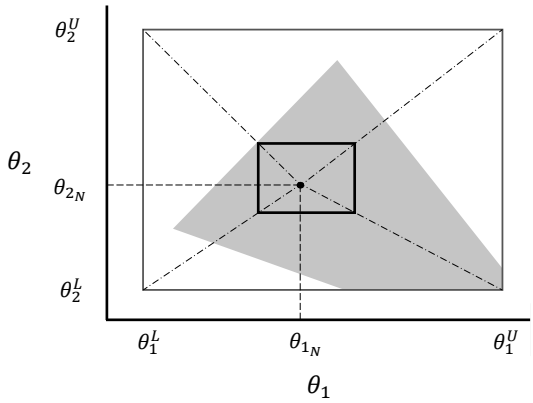


Figure 2.4: Flexibility index problem in 2D, based on [120]. θ_1 and θ_2 are two uncertain parameters. The resulting hyper-rectangle (solid line) is inscribed in the feasible region (in gray), centered around the nominal design θ_N (dot) and with the same aspect ratio of the design space (dotted line).

Goyal and Ierapetritou [36] build an approximation of the entire feasible region, by inscribing a convex hull in it and circumscribing a convex polytope around it, and using the two as lower and upper bounds. Finding a mathematical formulation for the feasible region offers, clearly, the greatest available flexibility. Nevertheless, it is not practical to serve as a design guideline: it does not provide an indication on how to design one single parameter, as all parameters are coupled together. Concurrent engineering is, thereby, inconvenient. Decoupling among parameters is achieved with the shape of an axis-aligned hyper-rectangle: each parameter is associated with an interval that is completely defined by its lower and upper bound and, thus, independent on the specific value of the other parameters. This empowers designers with a simple guideline: to design each parameter inside a prescribed range. Although it is easier said than done, a prescribed range of values is certainly more valuable than a prescribed single design.

Robust design optimization [9] and reliability-based design optimization [98] –besides looking for single designs– assume, in general, a mathematical description of the variability of the

parameters, e. g., through a probability distribution. However, a model of the design parameter uncertainty is not available in the early stages. Hyper-rectangles, instead, do not express the variability in terms of the values a parameter is *likely* to assume, rather of the values it is *allowed* to assume. The limits of variation are the focus here.

To sum up, effective design guidelines for the early development phases require flexibility, in the form of a hyper-rectangle.

Zimmermann and von Hoessle [141] optimize the flexibility –to start with, say size– of a hyper-rectangle inscribed in the feasible region and call it *box-shaped solution space* or solution box.[5] In a sense, this problem relates to an inverse flexibility test problem: instead of evaluating whether a certain hyper-rectangle is feasible, it aims to find the largest feasible hyper-rectangle. It relates also to the flexibility index problem, though it drops the consideration of a nominal design: the resulting hyper-rectangle is the largest feasible one, overall. Due to its foundation, I refer to this problem as *design flexibility optimization* and to the solution box as *flexibility box*.

In [141], a stochastic algorithm is introduced to perform design flexibility optimization for black-box, high-dimensional functions. Starting from a random feasible design –obtained for instance by classical optimization techniques– a candidate box is, first, initialized and, then, iteratively updated by increasing its size, sampling the new space and relocating its boundaries so as to remove the infeasible designs. The iterations continue until the box does not significantly move anymore. Graff et al. [39] analyze the performance of such algorithm and find very good results in low-dimensions, while both accuracy and convergence deteriorate as dimension increases.

The algorithm is computationally expensive, as the number of necessary function evaluations is, in general, greater than 10,000. For applications where each function evaluation is in the order of a dozen minutes, the computational effort becomes disproportionate and the use of a surrogate model, thus, necessary. Furthermore, the iterative approach zooms into a specific area of the feasible

5 The solution space is the set of all feasible designs –i. e., solutions to the feasibility problem– namely, it corresponds to the feasible region.

region and focuses here most of the samples. In case the feasible region is non-convex or there are multiple disconnected ones, this may lead to a local optimum.

Fender [29] introduces a mathematical formulation for problems that can be analytically described by linear constraints and, at most, quadratic objective function: finding the flexibility box becomes a linear, convex optimization problem. He applies it to the design of vehicle front crash structure. This approach is significantly more accurate and computationally efficient than the stochastic one, yet applicable to a limited number of problems. The problem addressed in this thesis does not belong to those, due to the lack of an analytical model. Hence, this approach will not be further discussed.

Graf et al. [37] compute the flexibility box without an iterative procedure, rather with a global approach. They compute the minimum axis-aligned bounding box around the feasible region, then sample it uniformly and split each dimension in intervals. In this way, the bounding box is divided into sub-hyper-rectangles. Each of them is, then, classified as feasible or infeasible, depending on whether it contains only feasible designs. By considering all possible combinations of sub-hyper-rectangles, they can find the combination that constitutes the largest box. The accuracy of this approach lies in both the number of intervals in which each dimension is divided and the number of samples –for both, the larger the number, the more precise the box. However, the number of intervals greatly impacts the computational effort, since, as dimension increases, the number of possible combinations rapidly grows to unreasonable values. Precise information about accuracy and computational efficiency are not reported, yet this approach does not seem practical in high-dimensions with tens of thousands of samples. Therefore, if the consideration of the entire feasible region in [37] –instead of an iterative zooming in a specific area– may benefit the convergence towards a global optimum, the shrinking –or trimming– algorithm in [39, 141] seems more computational efficient to find the largest feasible box from a given set of samples.

Even provided that the optimal flexibility box is obtained, several feasible designs may be excluded by this box, depending on

the complexity of the feasible region shape. This problem has been addressed in several ways in literature. Erschen et al. [24] couple pairs of design parameters, so that a polygon –instead of a hyper-rectangle– is identified for these two parameters. The pair is still decoupled from the remaining pairs or single parameters. By coupling *two* parameters at the time, visualization on a 2D surface is still possible. Vogt et al. [126] split the design parameters into early- and late-decision ones and initially compute the box only with the early-decision parameters. This approach increases the flexibility of the box in the early phases, yet it is only applicable when the late-decision parameters can be *exactly* set at later stages, as their range of flexibility may be here very low. Daub et al. [20] couple the design parameters component-wise, yet, when there are more than two parameters per component, visualization may become difficult.

All above approaches have focused on finding *one* box –or product of polygons. No study so far has tried to find *multiple boxes* at the same time. This would, however, find good reasons:

- the feasible region could be better approximated and more feasible designs included in the output,
- the approach would keep the simplicity of the hyper-rectangular shape and the completeness of the design parameters,
- designers would be able to choose among multiple guidelines based on their experience or switch among them during the development process, in case a major issue arises,
- designers would be able to extend their understanding of the feasible region and the interactions among parameters in the design space.

3 RESEARCH QUESTION

Some years ago, the head of the Industrial Engineering Department of Yale University said:
"If I had only one hour to solve a problem,
I would spend up to two-thirds of that hour
in attempting to define what the problem is."

— William H. Markle[1]

In Chapter 1, I introduced the present research topic and why it is relevant to the scientific community. In Chapter 2, I reported to what extent previous studies addressed the topic and discussed where I see potential for further research. In this chapter, I sum up the intention of this thesis: I explicitly formulate the question the current research aims to answer and the narrower sub-questions it can be broken down into.

The research question of this thesis is:

> *Which method is appropriate to compute design guidelines for pedestrian leg impact on the basis of a low-fidelity vehicle front-end model?*

I aim to propose a method that can be used by concept engineers as a support during the early design stages. The method should provide indications on the recommended properties of the relevant front-end structures, so that pedestrian leg impact requirements can be fulfilled.

The formulated research question defines the high-level, broad scope of the thesis. To reach this scope, smaller problems must be solved. They are more limited in extent and represent the single objectives of the proposed method. I present them in the following in the form of low-level sub-questions:

1 From: William H. Markle. "The Manufacturing Manager's Skills." In: *The Manufacturing Man and His Job*. Edited by Robert E. Finley and Henry R. Ziobro. 1966.

1. *Which low-fidelity modeling technique of the vehicle front-end is appropriate for the impact with the advanced Pedestrian Legform Impactor (aPLI)?*

   As reported in Chapter 2, no study has run investigations with front-end LFMs and aPLI yet. Accordingly, no analysis of the relevant front-end components –or load levels– to be modeled for this impactor is available so far.

2. *Which parametrization approach is recommended for the representation of design variability and handling of complexity?*

   As reported in Chapter 2, front-end design parameters used in previous studies either focused on specific load levels or assumed ideal force-deformation characteristics. In this thesis, the attention lies in the consideration of the entire front-end and in the definition of design modifications as realistic as possible. Still, a large number of parameters is strongly discouraged in order to avoid too-high dimensional spaces. This further raises the challenge.

3. *Which method for the evaluation of design feasibility is appropriate with computationally-expensive black-box functions?*

   Feasibility analysis of a large number of designs is a time-consuming task when function evaluations are computationally-expensive. Previous studies propose efficient methods to perform feasibility analysis with purposefully-trained surrogate models. However, the performance of such methods with the increase of dimensionality is open.

4. *Which type of guideline is recommended to support the design process?*

   As reported in Chapter 2, previous studies on pedestrian leg impact aimed to find either one single feasible design or parameters' sensitivity indices. However, none of those offers any design flexibility. Methods from other disciplines aim to find, instead, one set of parameter ranges. This can serve as effective guideline during the design process. Providing, then, more than one set of parameter ranges –i. e., more

boxes– would further support the designers. A formulation for this problem is, though, not available yet.

5. *Which optimization procedure is appropriate to maximize the flexibility level of the guidelines?*

   As reported in Chapter 2, current optimization approaches rely on either an iterative procedure –sensitive to local optima– or a large-scale combinatorial problem. The first is not suitable to find more boxes at the same time, the second may become too computationally expensive. The optimization algorithm is, therefore, still unclear.

6. *Which range of values is recommended for front-end parameters to fulfill the pedestrian leg impact requirements?*

   This objective represents the final outcome of the method, rather than a piece of it. It is obtained by combining all the parts of the method together. As outcome, it constitutes the piece of information to be delivered to the designers.

Part II

THE METHOD.

4

LOW-FIDELITY MODEL OF VEHICLE FRONT-END FOR PEDESTRIAN LEG IMPACT

What is simple is always wrong.
What is not is unusable.

— Paul Valéry[1]

In the early phase of a vehicle development process, engineers dealing with pedestrian protection have to assess whether the styling of the front-end is pedestrian-friendly, i. e., whether it can satisfy pedestrian protection requirements. Clearly, any styling could be pedestrian-friendly; it depends on the rigidity of the structures mounted in the front-end. However, not every stiffness is possible, as other requirements need to be taken into account as well; among others:

- material quality, e. g., reliability, durability and aesthetics,
- material costs,
- resistance to misuse,
- reparability costs in case of low-speed impacts,
- styling, e. g., short overhang,
- manufacturability,
- integration of components used for active safety and occupant protection,
- regulation of the inlet air flow.

1 In: Paul Valéry. *Mauvaises pensées et autres*. 1942.

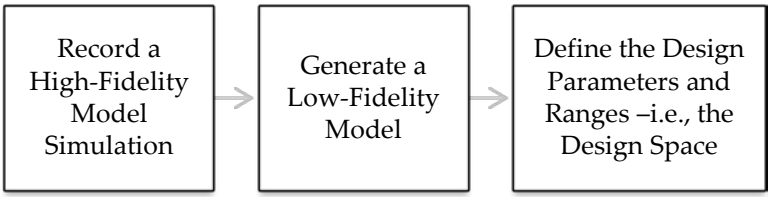


Figure 4.1: Broad overview of steps described in Chapter 4.

Therefore, concept engineers are challenged to understand, first, which structures are necessary to satisfy the pedestrian protection requirements and, then, if those can satisfy also the other requirements. In this thesis, I discuss methods to accomplish the first task; the second one is mostly related to experience and other disciplines' simulations and is taken into account only partly in the Low-Fidelity Model (LFM) parametrization.

Figure 4.1 shows an overview of the steps taken in the current chapter. In the first section, I propose a procedure to generate a Low-Fidelity Model (LFM) of a passenger vehicle front-end for the pedestrian leg impact. The aPLI is used as exemplary legform impactor. Nevertheless, the procedure can be equally applied to the FlexPLI. In Section 4.2, I define parameters of the LFM that could be possibly relevant for design investigations. As the fundamental reason to build an LFM of a vehicle front-end is the simplicity in the implementation of changes, the parametrization is of the utmost importance.

4.1 LOW-FIDELITY MODEL GENERATION

In the early phase of the development process, concept engineers must identify the measures that needs to be taken to satisfy the pedestrian protection requirements. In order to tackle this task, it is essential to have an FE simulation model that allows to rapidly perform changes and gather know-how on how the development should evolve. In early phases, high-fidelity FE simulation models of the vehicle front-end may not be already available. Even if

they were, they would be quite unsuitable for the task. In fact, they take long to simulate and long to be changed. They usually consist of a great number of elements, about 4 million for a medium-sized sedan, as in Figure 4.3a, and take about 6 h to run on a High-Performance Computing (HPC) cluster with 48 processors. Furthermore, to implement substantial modifications, either geometry- or stiffness-related, that do not involve simple thickness variations or different material model definitions, is rather troublesome and time-consuming. The complexity lies in the many components, shapes and attachment points that are present in a front-end and require careful design to avoid geometrical conflicts. Therefore, especially inside an automatic framework, as required, for instance, by a Design of Experiments (DOE) or structural optimization techniques, High-Fidelity Models (HFMs) do not fit well.

A simpler FE simulation model of the vehicle front-end is desirable. It should combine the following features:

ABSTRACTION: the material and mechanical properties of the front-end structures should be modeled with a high degree of abstraction, in order to minimize the number of assumptions and leave freedom for new structural concepts;

DEGREES OF FREEDOM: the model should retain only the DOFs relevant to the pedestrian leg impact;

PARAMETRIZATION: parameters, affecting the geometry or the stiffness of the front-end, should, first, be easy to implement and link to an automatic framework and, then, cover the desired range of change;

RUNTIME: the computational effort involved with an FE simulation should be as low as possible;

ACCURACY: last, but not least, the model should be accurate *enough* to perform meaningful and realistic investigations.

In this chapter, I discuss the first three requirements, while the last two in Section 6.1.

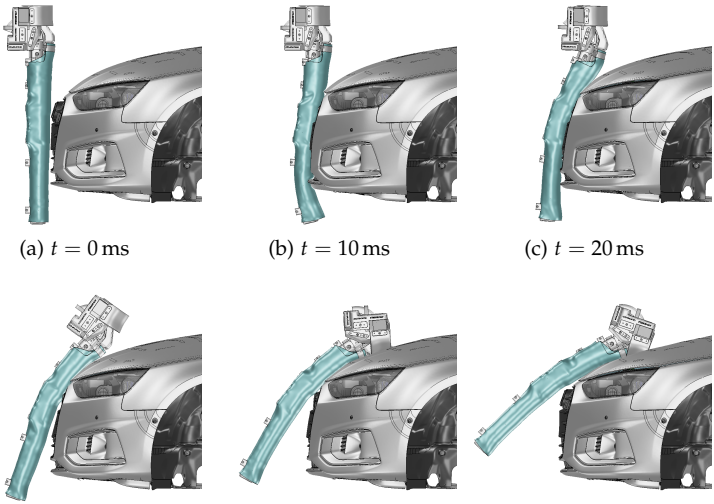
4.1.1 *Abstraction*

The highest degree of abstraction would be achieved by not modeling any of the vehicle front-end components. However, proper contact representation between impactor and vehicle is crucial for simulation accuracy and must be, hence, ensured.

Figure 4.2 shows the kinematics of the pedestrian leg impact for a medium-sized sedan with the aPLI. We can note that the impactor tends to wrap around the outer surface of the vehicle front-end. To use this surface as contact surface seems appropriate. Mößner et al. [89] adopt this approach and define the outer surface as a massless rigid body. This is done to decouple the contact and the energy-absorption functions in the LFM. Nonetheless, a rigid body allows no local deformation; instead, the contact between the impactor and the front-end has a very local behavior. Therefore, they split the outer surface vertically in several independent strips, equally distributed over the surface. They do not specify, though, how many strips are necessary. As for the stiffness representation of the front-end structures, they record the contact force and the deformation of each strip from a HFM FE simulation and map them on massless, translational, non-linear springs, one per strip, as done, e. g., also in [92, 121]. In this way, they get rid completely of the tedious detailed modeling of components and replace it with a basic representation of stiffness, i. e., a one-dimensional, possibly non-linear, force-deformation characteristics. Each spring is rigidly connected to the respective strip and is inclined as the regression line fitting the deformation in the xz-plane. As it is a computationally efficient modeling technique for the pedestrian leg impact, I adopt a similar approach for the LFM proposed in this thesis. Differences with respect to [89] will be highlighted at the end of chapter.

4.1.2 *Load Levels Definition*

As the information on the deformation is at a node level, while that on the contact force at a strip level, it is not possible to map force and deformation directly. Out of all the nodes defined in one strip, the most representative deformation has to be chosen. It is not a



(a) $t = 0\,\text{ms}$ (b) $t = 10\,\text{ms}$ (c) $t = 20\,\text{ms}$

(d) $t = 30\,\text{ms}$ (e) $t = 40\,\text{ms}$ (f) $t = 50\,\text{ms}$

Figure 4.2: Kinematics of the aPLI for a medium-sized sedan at different time steps in central shooting position.

simple task, as, depending on how large a strip is, deformations among the nodes may vary greatly. The outer surface of the vehicle front-end may, in fact, undergo significant local deformations. Many approaches could be used, such as average, minimum or maximum displacement. In order to keep the error at a minimum, I define small strips of 10 mm height in z-direction, as shown in Figure 4.3b, and choose the node with maximum displacement during contact as most representative, as it is usually the one that stayed the longest in contact with the impactor. It is also the one that gave the best results in terms of accuracy in my investigations. For sake of convenience, I define a width of 250 mm for the outer LFM surface and center it around the shooting position, as shown in in Figure 4.3b for the position $y = 0$ mm. The width, however, as long as it covers the contact surface, does not play a role in the computation. The combination of a strip and the respective force-deformation characteristics is called load level in this thesis. It is assumed that each load level moves independently from the others.

4.1.3 *Degrees of Freedom*

Three physical quantities should be taken into account in the generation of a mechanical LFM:

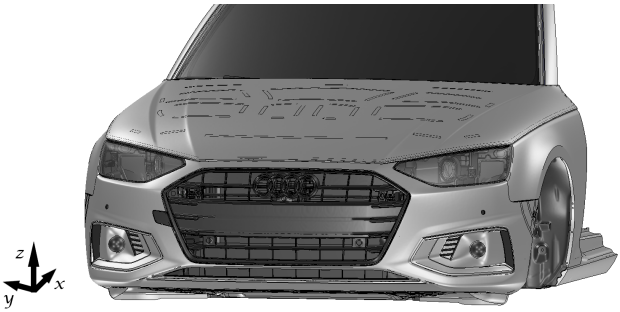
DEFORMATION: $\mathbf{u} = \mathbf{u}(x, y, z)$ is a vector in $\mathbb{R}^3$. It is saved by the FE solver PAM-CRASH v2018 for every node defined as output of a simulation.

FORCE: $\mathbf{F} = \mathbf{F}(\mathbf{u})$ is a vector in $\mathbb{R}^3$. Unfortunately, the contact force cannot be saved by the FE solver PAM-CRASH v2018 on a single node as output of a simulation, rather on a group of nodes [25]. I define all the nodes in a strip.

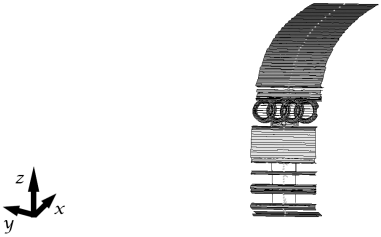
MECHANICAL WORK: W is a scalar and can be computed as:

$$W = \int \mathbf{F}(\mathbf{u}) \cdot d\mathbf{u} \tag{4.1}$$

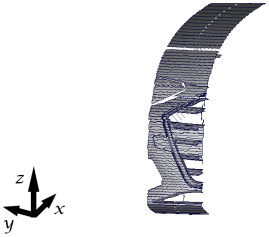
where $\mathbf{F}$ and $\mathbf{u}$ are defined above. Considering the whole system, following the work–energy principle, the mechanical work equals the variation of kinetic energy E_k of the



(a) HFM



(b) LFM at $y = 0\,\text{mm}$



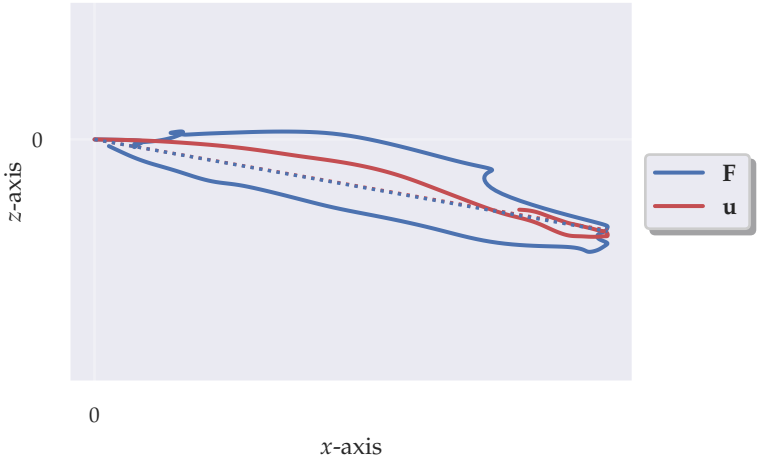
(c) LFM at $y = 450\,\text{mm}$

Figure 4.3: Comparison of an FE High-Fidelity Model (HFM) of a medium-sized sedan and its respective Low-Fidelity Model (LFM) at the shooting positions $y = 0\,\text{mm}$ and $y = 450\,\text{mm}$.

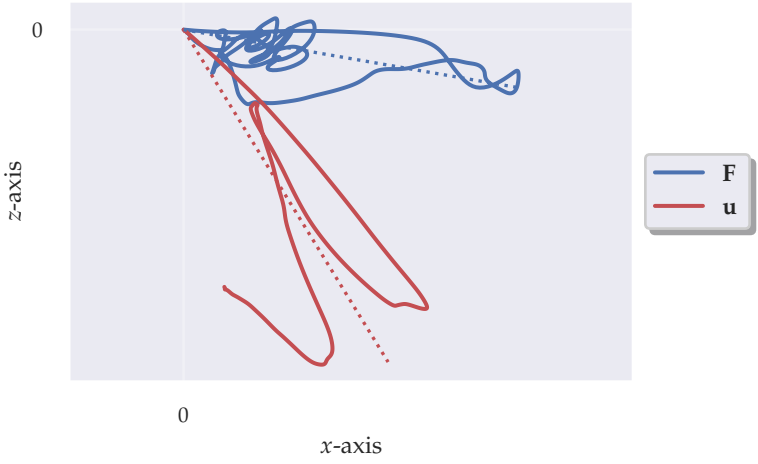
impactor and is the sum of the energy absorbed by the front-end structures and the energy dissipated by friction at the contact interface[2].

Translational springs act, by definition, on a straight line. If both **F** and **u**, recorded during the high-fidelity model simulation, were linear and coincident, all force, deformation and mechanical work could be exactly reproduced and the springs could be perfectly accurate. Unfortunately, this is not always the case. Figure 4.4 shows the evolution of normalized **F** and **u** in the xz-plane for two different load levels. The level in Figure 4.4a presents a very linear deformation and a quite linear force, whose regression line is almost coincidental to the displacement one; we can expect the translational spring to approximate really well the high-fidelity model here. This is usually the case for the load levels in front of the bumper beam, as they deform mainly in x-direction. The level shown in Figure 4.4b, instead, presents a much more complex deformation and the deviation from the regression line is more evident. In addition, force and deformation vectors develop in different directions: the first one still acts mainly in x-direction, while the second one has a strong z-component. We can expect the spring to be a rougher estimation here. Furthermore, it raises the question which inclination should be taken for the spring. A translational spring, if fully constrained at one end, cannot deform in its orthogonal directions. To prevent this movement, it can exert an artificially large force. Furthermore, the location where the force acts, defined by the deformation of the spring, is especially important for flexible impactors as aPLI and FlexPLI. Therefore, to orient the spring according to the direction of the displacement averaged over time is, in general, a better choice. This is also confirmed by better accuracy of the results, as reported in [89]. Figure 4.5 shows the orientation of the springs for the LFMs in Figure 4.3b and Figure 4.3c.

2 In the LFM, friction is taken into account in the contact definition between impactor and vehicle outer surface. It should be, therefore, excluded from the force-deformation characteristics of the spring. This is, however, not trivial, as the friction force is integrated into the contact force. All in all, I consider this effect negligible, due to the fact that it takes place in the direction tangential to the vehicle surface, which is not the main one for energy-absorption.

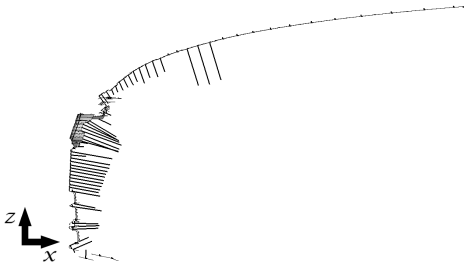


(a)

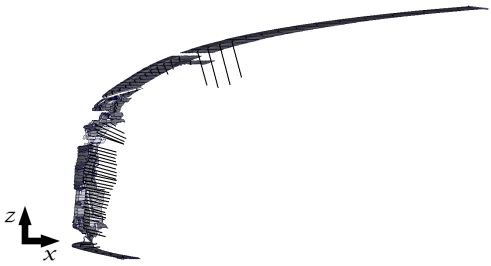


(b)

Figure 4.4: Comparison of the evolution of normalized force and deformation vectors in the xz-plane and their respective regression lines for two load levels of the model in Figure 4.3a: (a) taken from the license plate (lev26), (b) taken from the bonnet (lev51), where, during unloading, re-loading takes place.



(a) LFM at $y = 0\,\text{mm}$



(b) LFM at $y = 450\,\text{mm}$

Figure 4.5: Side view of the Low-Fidelity Model (LFM) of the medium-sized sedan in Figure 4.3a at the shooting positions $y = 0\,\text{mm}$ and $y = 450\,\text{mm}$.

A deviation of $\mathbf{u}$ from the regression line during the HFM simulation, as the one shown in Figure 4.4b, would cause the related load-level to exert an artificially large force in the orthogonal direction during the LFM simulation to prevent movement. Furthermore, changes in the LFM during parametrization may induce load levels to move out-of-line and, thus, give rise to locking. This is inconvenient, especially if the force caused by artificial stiffness is comparable –or even larger– than the original one exerted by the load level. A possible solution to this problem is to define springs also in the orthogonal directions. As the $y-$component of the force-deformation characteristics gets relevant for non-central shooting positions, I define a system of three orthogonal, translational, non-linear springs rigidly attached to the respective front-end strip, at one extremity, and supported by one roller each, at the other extremity, as shown in Figure 4.6. I could use also one single spring with three translational Degrees of Freedom (DOFs) and hinged at one end. However, the reference frame attached to the spring may rotate during deformation and the directions along which the force-deformation characteristics act would not correspond to the initial ones anymore. Additionally, the angles of rotation would depend on the position of the constrained end, causing this to be a relevant parameter. Therefore, to always keep the springs with the same orientation, I use rollers, able to move freely in the spring orthogonal plane.

Let $\mathbf{U}$ be the $n \times 3$ matrix that contains n vectors $\mathbf{u}$ of a single load level, where n is the number of recorded time-steps during the HFM simulation in which the load level is in contact with the impactor. The orientation of the three orthogonal, translational springs for that load level is obtained by Principal Component Analysis (PCA) of $\mathbf{U}$. PCA is computed by standard Singular Value Decomposition (SVD) [35] with the `python` library `NumPy` [96]:

$$\mathbf{U} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T \tag{4.2}$$

where

$\mathbf{W}$ is an $n \times n$ matrix containing the left singular vectors of $\mathbf{U}$,

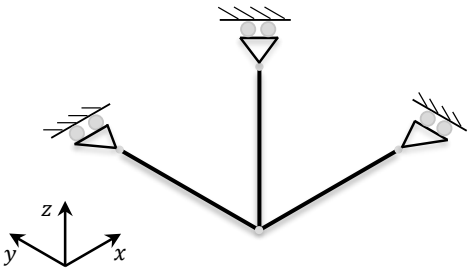


Figure 4.6: System of three orthogonal translational springs –represented as connector elements– rigidly attached at one common end and supported by rollers at the other end. The rollers can freely move in their orthogonal plane.

$\mathbf{\Sigma}$ is an $n \times 3$ diagonal matrix containing the singular values of $\mathbf{U}$,

$\mathbf{V}$ is a 3×3 matrix containing the right singular vectors of $\mathbf{U}$.

$\mathbf{V}$ forms an orthogonal basis for $\mathbb{R}^3$ and contains the cosines of rotation into the principal components. Therefore, in order to transform the vectors $\mathbf{F}$ and $\mathbf{u}$ –recorded at each time-step– in the new basis, these are multiplied by $\mathbf{V}$ by means of matrix multiplication:

$$\mathbf{F}' = \mathbf{F}^T\mathbf{V} \qquad \mathbf{u}' = \mathbf{u}^T\mathbf{V} \tag{4.3}$$

where $\mathbf{F}'$ and $\mathbf{u}'$ are, respectively, the force and deformation vectors expressed in principal components. Each component i of $\mathbf{F}'$ and $\mathbf{u}'$ is then mapped together –as $\left(F_i', u_i'\right)$– to the i-th translational spring. For each spring, a subset of the n recorded time-steps is defined: those pairs $\left(F_i', u_i'\right)$ belonging to the loading phase and defined in the first and, possibly, third quadrants.

It is worth noting that the force recorded from the high-fidelity model simulation is a function of the deformation in x-, y- and z-directions. Therefore, it can be considered a mapping $f : \mathbb{R}^3 \to \mathbb{R}^3$. The force defined in each translational spring, instead, is only a function of the deformation in the corresponding principal

component: $f_i : \mathbb{R} \to \mathbb{R}$. This mapping is repeated three times, once per each component, to generate three *independent* force-deformation characteristics. Some loss of information happens here. This is direct cause of the choice to replace most of the elements and DOFs of the HFM with translational springs. This topic will be discussed more in detail in Chapter 8.

4.1.4 *Unloading*

The force-deformation characteristics of a non-linear spring can be divided into two phases: the loading phase and the unloading one. While the first one can be precisely defined in FE solvers, the second one requires some special treatment. As a matter of fact, as far as the deformation is monotonically increasing, the recorded, possibly non-linear, force-deformation mapping can be used directly for the loading. For what concerns the unloading, instead, the recorded data return only *one possible* unloading curve, i. e., the one starting at the recorded maximum intrusion. However, the FE solver needs to know how to unload the structure also for maximum intrusions different from the recorded one. Mößner [88] –with the explicit FE solver ABAQUS– prescribes two sample unloading curves and compute the actual one as interpolation between the two. PAM-CRASH allows to choose between two strategies [25]:

1. The unloading phase consists of two stages: a sample unloading curve and a transition line, which starts from the point of maximum intrusion and reaches the unloading curve. The unloading curve must be monotonically decreasing and lower at any point than the loading one.

2. The unloading phase is implemented via a power function:

$$F(u) = a\,(u - \varepsilon u_{max})^b \tag{4.4}$$

where u_{max} is the maximum intrusion, ε the permanent deformation factor[3] and a and b two coefficients that are

3 εu_{max} is the permanent deformation.

automatically calculated by PAM-CRASH to match a user-defined dissipation factor D. This is computed as:

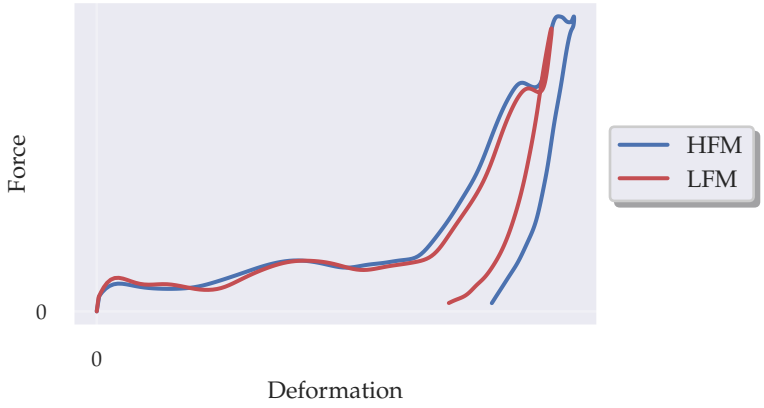
$$D = 1 - \frac{\left| \int_{u_{max}}^{\varepsilon u_{max}} F(u)\ du \right|}{\int_0^{u_{max}} F(u)\ du} \tag{4.5}$$

where the denominator is the energy absorbed during the loading phase, while the numerator the one recovered during the unloading phase. In other words, $D = 0$ describes a perfectly elastic material –i. e., with no dissipation– and $D = 1$ a perfectly inelastic one –i. e., purely dissipative, such as one following a fracture.

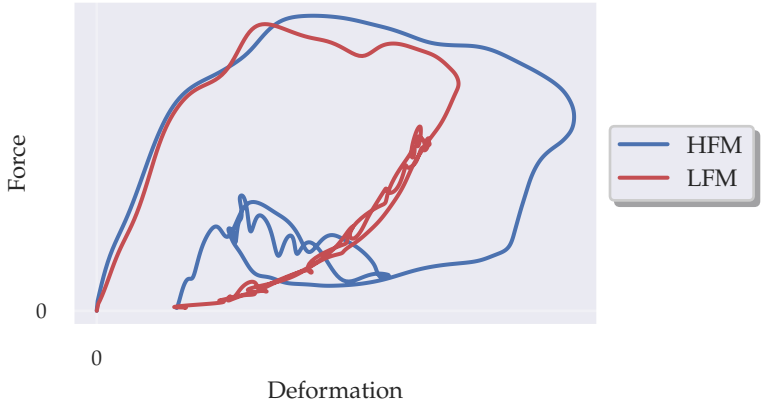
For sake of both simplicity and accuracy, I adopt the second strategy and set ε to zero. First, the use of a dissipation factor as only parameter is a convenient way to control the unloading phase and is exploited in Section 4.2 for the LFM parametrization. Then, the power law shape approximates well most of the recorded curves. Figure 4.7 shows a comparison of the force-deformation characteristics between HFM and LFM for two different load levels. In Figure 4.7a, both the loading and the unloading phases appear to be sufficiently well approximated. In Figure 4.7b, instead, we see a typical behavior that occurs on the first part of the bonnet area: this is first loaded while the impactor is still vertically straight; then, once the Upper Body Mass (UBM) of the aPLI rotates over the bonnet, it is loaded again. Therefore, during the unloading phase, re-loading takes place. An analogous issue is experienced by Mößner et al. [89] with the FlexPLI. PAM-CRASH uses, in both of the above mentioned strategies, the same curve for unloading and re-loading.

4.1.5 *Extrapolation*

Both levels in Figure 4.7 have a smaller maximum intrusion in the LFM than in the HFM simulation, with the effect being more significant in Figure 4.7b. This hints that the load levels in these regions are, in that particular simulation, stiffer than their HFM counterparts. Sometimes, the contrary happens and some load



(a)



(b)

Figure 4.7: Comparison of the recorded force-deformation characteristics in x-direction between the High-Fidelity Model (HFM) in Figure 4.3a and the Low-Fidelity Model (LFM) in Figure 4.3b for two load levels: (a) taken from the license plate (lev26), (b) taken from the bonnet (lev51).

levels intrude more in the LFM than in the HFM. In this case, extrapolation of the recorded data is necessary. Extrapolation is particularly relevant for simulations where variations are applied to the LFM. Here, deviations in the LFM deformation with respect to the HFM are not only due to model inaccuracy, but especially to the changes applied. PAM-CRASH automatically extrapolates the force-deformation curve on the basis of the slope between the last two points defined [25]. Some engineering know-how should be introduced here to prevent unrealistic intrusions. In fact, the deformation space reserved for pedestrian protection is rather limited and relatively-hard structures, such as the bumper beam and the engine cover, are placed afterwards. However, they are not generally hit during a pedestrian leg impact; hence, there is no force-deformation information for those structures. A possibility to gather more information is to increase the impactor initial kinetic energy. This ensures more deformation and, thus, more recorded data. Nevertheless, this may change the impactor kinematics and the responses of the structures due to inertial effects and material strain-rate dependency. Therefore, the force-deformation characteristics in the originally recorded area may differ. Without changing the initial kinetic energy, the impactor shooting height could be varied. This causes more intrusion in some areas and less in others. The information coming from different shooting heights should then be combined together. However, the different impactor position and kinematics lead the upper body mass of the aPLI to hit the bonnet in different regions. The stiffness of the bonnet, usually made of steel or aluminum, generates a rather global deformation –i. e., also regions, or load levels, not in contact with the impactor are deformed. This deformation contributes to the stiffness –i. e., the force-deformation characteristics– of the regions in contact. By changing the impactor position, new load levels go in contact with the upper body mass. By combining together the information coming from different simulations, the risk is to consider the stiffness of some load levels more than once, overestimating, then, the overall bonnet stiffness.

In light of the above reasons, I define the extrapolation on the basis of the information provided by the recorded data of a single reference simulation. A linear extrapolation, because of its sim-

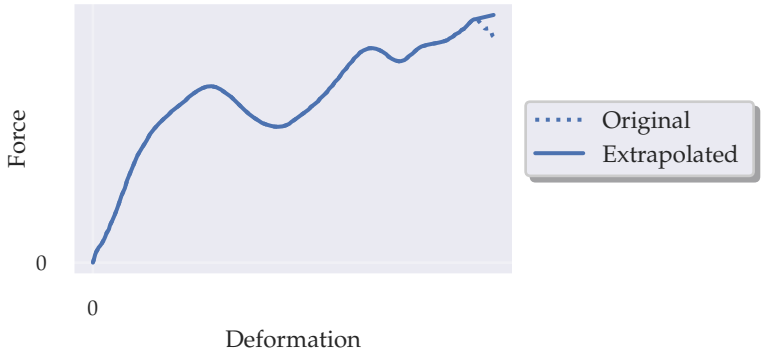


Figure 4.8: Example of the linear extrapolation of a loading force-deformation curve.

plicity, seems reasonable. Neglecting cases where material failure applies, it is rather unlikely that structures decrease their exerted load as deformation goes on. Therefore, similarly to [89], I set the minimum extrapolation slope to zero. Due to many components interacting and contacting each other and plasticity, the recorded force-deformation curve may be very non-linear. To rely on a small portion of the curve may be risky. Therefore, I compute the average slope in the last 25% of the deformation of the loading curve and use the point of maximum force in the portion considered as starting point for the linear extrapolation. Figure 4.8 shows an exemplary extrapolation curve. If the computed slope is negative, the extrapolation follows a constant line from the point of maximum deformation.

4.1.6 *Load Levels Clustering*

A height of 10 mm for each load level is necessary to keep the mismatch between recorded force and deformation sufficiently low. However, this implicates a large number of load levels to be defined in the LFM. The model of the medium-sized sedan in Figure 4.3b, for instance, consists of 80 load levels. Out of these, only 53 are non-zero –i. e., get in contact with the impactor. A

reduction of the number of load levels does not speed-up the simulation, as the elements of the impactor take most of the simulation time. Rather, it is beneficial for the parametrization: on the one hand, because the fewer the load levels, the easier is to apply the changes; on the other hand, because load levels of just 10 mm height are not representative of any structural member underneath the vehicle outer surface. In order to define valuable parameters, these should be as closely related to real components as possible. Therefore, it would be appropriate to merge some load levels together, or in other words, to *cluster* them, so as to get closer to the force-deformation characteristics of the single structures. The downside is that a rougher representation of the front-end generally may reduce the accuracy.

Cluster analysis is a popular data mining technique to find regularity in a set of observations and arrange them into groups according to some similarity criteria. There exists several clustering algorithms, which differ for how they measure similarity or how they group observations together. A thorough overview can be found in [102]. In order to recognize load levels belonging to the same structure, it is necessary to define one or more criteria that describe how a structure works. I use three features:

1. the maximum intrusion in the first principal component $u'_{1_{max}}$,
2. the inclination of the deformation vector $\mathbf{u}$ in the xz-plane,
3. the inclination of the force vector $\mathbf{F}$ in the xz-plane.

Essentially, I assume that a structure is mainly characterized by how it deforms. Furthermore, a load level may only be merged with its adjacent levels. Therefore, a connectivity constraint has to be taken into account. This is possible with the hierarchical clustering algorithm, which I apply in its agglomerative form with the `python` library `scikit-learn` [97]. Agglomerative hierarchical clustering starts with each observation in its own single cluster and merges together a pair of clusters at each iteration according to the metric and linkage criterion used. I specify Ward's criterion, originally introduced by Ward [132], that, at each iteration, selects the pair of clusters to merge that minimizes the increase

in variance within all clusters. With this linkage criterion, only Euclidean distance can be used as similarity measure. In order to allow the computation of meaningful distances, the features are first normalized in the interval $[0, 1]$. The variance is used as similarity metric also by the *k*-means clustering algorithm, though this uses a different approach to form the clusters, which does not allow the definition of connectivity constraints. Both the *k*-means and hierarchical clustering algorithms require the user to specify the number of clusters k.

The connectivity matrix $\mathbf{C}$ is a symmetric matrix with rows and columns labeled by observations, with a 1 or 0 in position (i, j) according to whether the i-th observation may be connected with the j-th one. Since a load level may be only connected with its adjacent ones, I define an $n \times n$ connectivity matrix that is the sum of an upper shift matrix and a lower shift matrix:

$$\mathbf{C_n} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ & & & & 1 \\ 0 & & \dots & & 0 \end{pmatrix} + \begin{pmatrix} 0 & & \dots & & 0 \\ 1 & & & & \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} =$$

$$= \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & & 1 \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}$$

where n is the number of load levels in the LFM.

Once similar load levels are grouped together into clusters, their equivalent force-deformation characteristics must be computed. The load levels in a cluster are analogous to a system of springs in parallel. Unlike such springs, though, the load levels do not share the same deformation. In order to compute the equivalent characteristics a shared quantity has to be found. I use *time*. In fact, data are recorded along the same time axis. Therefore, at each

time step at which the HFM output is saved –i. e., every 0.05 ms– I compute the equivalent force and displacement vectors in the global reference frame, $\mathbf{F_{eq}}$ and $\mathbf{u_{eq}}$, respectively as the sum of the force vector and the weighted average of the displacement vector of the load levels in the cluster:

$$\begin{aligned} \mathbf{F_{eq}} &= \sum_{i=1}^{n} \mathbf{F_i} \\ \mathbf{u_{eq}} &= \frac{\sum_{i=1}^{n} \left(\int_0^{u_{max}} \mathbf{F_i}(\mathbf{u_i}) \cdot d\mathbf{u_i} \right) \mathbf{u_i}}{\sum_{i=1}^{n} \int_0^{u_{max}} \mathbf{F_i}(\mathbf{u_i}) \cdot d\mathbf{u_i}} \end{aligned} \tag{4.6}$$

where n is the number of load levels in the cluster. The weighting factor is the energy absorbed during the loading phase. By means of Equation 4.6, from a set of load levels in a cluster, I obtain an equivalent single load level for each cluster.

4.1.7 *Validation Metrics*

In order to quantify the accuracy of the LFM, several quantities can be measured, ranging from the force-deformation characteristics of the load levels to the displacement at different sections of the impactor [89]. I use directly the output responses that are considered in the regulation and consumer test protocols, as ultimately these are the quantities of interest. For aPLI, they are the bending moment around the y-axis at four sections of the tibia, the ACL, the MCL, the PCL and the bending moment around y at three sections of the femur, for a total of 10 injury criteria, as shown in Figure 1.3b.[4] Although other quantities may be more directly related to the impactor's kinematics, the injury criteria represent a good estimate of the LFM performance in different regions of the impactor, as they are measured over the entire leg.

It is not only relevant *what* to compare, rather also *how*. I consider two approaches:

4 ACL and PCL are currently not taken into account by the C-NCAP [18]. Anyway, I consider them in the validation, as they measure physical quantities related to the leg's kinematics.

RELATIVE ERROR: the maximum value of the i-th injury criterion over time, $IC_i(t)$, of the LFM is compared relatively to the HFM:

$$\begin{aligned} IC_{i_{HFM_{max}}} &= \max_t \left(\left| IC_{i_{HFM}}(t) \right| \right) \\ IC_{i_{LFM_{max}}} &= \max_t \left(\left| IC_{i_{LFM}}(t) \right| \right) \\ E_{r_{@max_i}} &= \frac{IC_{i_{LFM_{max}}} - IC_{i_{HFM_{max}}}}{IC_{i_{HFM_{max}}}} \end{aligned} \tag{4.7}$$

where E_r is the relative error and the subscript $_{@max}$ highlights that what is computed is not the maximum relative error, rather the relative error on the maximum values of the two curves. The relative error could be computed also at different time points of the curves –e. g., at fixed time intervals– to get a better overview of the whole simulation. Nevertheless, I consider the maximum point most relevant, because it is the quantity steering the design changes.

CORA ERROR: Correlation and Analysis (CORA) is a tool developed by Gehre et al. [33] to objectively compare two time-signals, typically from simulation and experimental test. It combines a corridor and a cross-correlation rating: the first evaluates how the curve fits in a corridor, while the second takes into account phase shift, shape and area under the curve. The CORA rating of the i-th injury criterion, R_{CORA_i}, quantifies how similar two curves are with a number between 0 and 1. In order to express an error, I simply compute:

$$E_{CORA_i} = 1 - R_{CORA_i}. \tag{4.8}$$

Although E_{CORA_i} is a more comprehensive evaluation of the similarity of two curves, the relative error on the maximum value is what matters for the evaluation of the pedestrian leg impact. Therefore, as validation metric I *measure what matters* –i. e., $E_{r_{@max_i}}$.

Finally, it is practical to validate the simulation on the basis of a single value. Out of the 10 injury criteria, then, I consider the worst case:

$$E_{r_{@max_{max}}} = \max_i \left(E_{r_{@max_i}} \right) \tag{4.9}$$

where i expresses the error on the i-th injury criterion.

4.1.8 *Workflow*

After the discussion of the single facets that compose an LFM, it is helpful to outline the actual sequence of steps I follow to generate an LFM.

1. INPUT: Import the HFM and the outer surface of the vehicle front-end.

   The outer surface must include all elements that go in contact with the impactor. An automatic outer surface generation is possible, as done, for instance, in [118].

2. PROCESS: Prepare the HFM for simulation.

   2.1. Slice the front-end outer surface around the shooting position, as shown in Figure 4.3b.

   2.2. Split the front-end outer surface in strips of 10 mm height.

3. PROCESS: Run HFM simulation.

   I use the explicit FE solver PAM-CRASH [25].

4. PROCESS: Post-process the HFM simulation output.

   4.1. Read the HFM force and deformation output curves for each load level.

   4.2. Rotate force and deformation vectors in principal components.

   4.3. Cluster the load levels together in k groups.

   4.4. Compute the equivalent force and deformation vectors for each group in global coordinates.

   4.5. Rotate new force and deformation vectors in new principal components.

   4.6. Split the loading and unloading curves.

   4.7. Compute the dissipation factor during unloading.

4.8. Extrapolate the loading curves.

5. PROCESS: Prepare the LFM for simulation.

   5.1. Define the strip of each load level as a rigid body.

   5.2. Connect each strip to three orthogonal, translational springs.

   5.3. Map the corresponding loading curve and dissipation factor to each spring.

6. PROCESS: Run LFM simulation.

   I use the explicit FE solver PAM-CRASH [25].

7. DECISION: Check the LFM accuracy.

   I compute the maximum relative error among the injury criteria, calculated on the maximum value, $E_{r_{@max_{max}}}$, according to Equation 4.9.
   If $\left|E_{r_{@max_{max}}}\right| \leq 0.1$, move on; otherwise, increase the number of clusters and repeat from step 4.[5]

8. OUTPUT: Export the LFM.

   The LFM is ready for parametrization.

This workflow is schematized in Figure 4.9.

4.2 LOW-FIDELITY MODEL PARAMETRIZATION

To state the importance of this section, I could simply claim that the procedure described in the previous section would be pointless without this one. The LFM is a simplification. This is justified only if useful. As George E. P. Box wisely summarized, "*all models are wrong but some are useful*"[6]. The *usefulness* of the LFM lies in

5 In case clustering is not the problem, a more detailed understanding of the error source is required in order to improve the model.

6 The expression is used in a technical report as title of the following section:

> Now it would be very remarkable if any system existing in the real world could be *exactly* represented by any simple model. However, cunningly chosen parsimonious models often do provide remark-

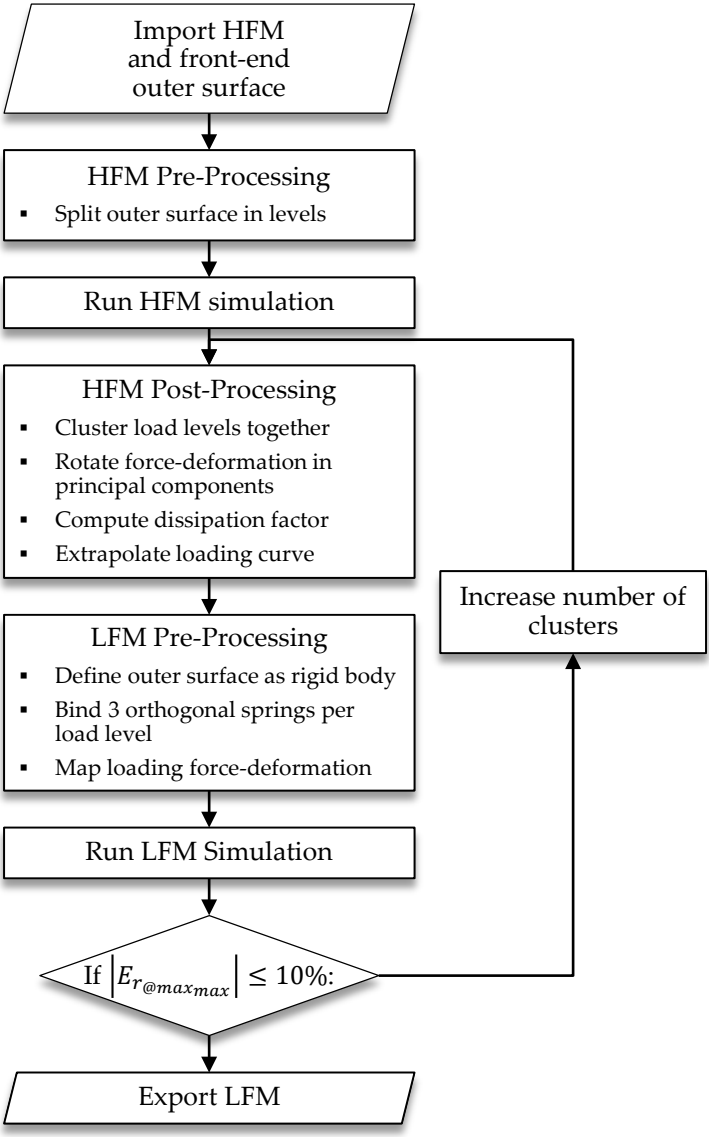


Figure 4.9: Workflow to generate a Low-Fidelity Model (LFM), as described in Section 4.1.8.

its capability to be modified while preserving the validity of its output. It is clear, then, that *what* is modified and *how* are crucial to the model validity. With output, I mean the set of information that can be gathered from one or more simulations: typically, for the pedestrian leg impact, trends and sensitivities of some design parameters with respect to the impactor injury criteria.

What should be modified is strictly related to the questions engineers are posed during the vehicle development process. In the early phase, questions regarding both the position and the stiffness characteristics of the load levels arise. At later stages, questions about the stiffness of the structures present in the front-end become more precise. Therefore, both *geometry-* and *stiffness-*related modifications should be investigated. Since where one load level acts and how much force it exerts are strictly linked, both categories of parameters should be used at the same time. An effort should be done, then, to keep the number of parameters as low as possible. The *curse of dimensionality* is a well-known problem [8] and should not be underestimated. Even more so considering the computational run time of an LFM simulation, which limits the number of simulations completed in a reasonable time period.

How the modifications should be –that is, which range of variation they should cover– is related to both the phase of the vehicle development process under consideration and the range of validity of the model. In early stages, general, conceptual investigations are carried out. Here, modifications should be rather broad to allow for sufficient flexibility. In later stages, big changes usually

---

ably useful approximations. For example, the law $PV = RT$ relating pressure P, volume V and temperature T of an "ideal" gas via a constant R is not exactly true for any real gas, but it frequently provides a useful approximation and furthermore its structure is informative since it springs from a physical view of the behavior of gas molecules.

For such a model there is no need to ask the question "Is the model true?". If "truth" is to be the "whole truth" the answer must be "No". The only question of interest is "Is the model illuminating and useful?".

From: George E. P. Box. “Robustness in the Strategy of Scientific Model Building.” In: *Robustness in Statistics*. Ed. by Robert L. Launer and Graham N. Wilkinson. Academic Press, 1979, pp. 201–236, excerpt pp. 202–203.

involve high costs. Here, investigations aim to fine-tune the model and study local sensitivities. However large the variation, it should not exceed the range, where the model is trustworthy. Further detail on this topic is provided in Section 6.1. Furthermore, the range of variation can affect also the amount of simulations required by the investigation. In fact, in case the number of parameters cannot be decreased, reducing range of variation of each parameter lowers, in general, the non-linearities in the design space.

In addition to *what* and *how*, the modifications should be:

- automatically implemented through the definition of parameters,
- uniquely defined by the value of the parameters,
- realizable in real-world vehicle front-ends,
- easy to understand and to communicate.

The first two points enable the usage of structural optimization techniques. The last two focus on the *usefulness* of the parametrization. In fact, we must not forget that the outcome of the investigations carried out with the proposed LFM should be transferred to the front-end design of real cars. This transfer is, however, outside the scope of this thesis and left to experienced CAD engineers.

In order to keep the number of parameters low, I concentrate the modifications to four main load regions of the front-end, as highlighted in Figure 4.10:

LOW: the lower region including the front spoiler;

MID: the middle region including the license plate;

MID-UP: the middle-upper region including the emblem;

UP: the upper region including the bonnet.

A middle-lower region is missing, as that space is usually reserved for Advanced Driver-Assistance Systems (ADAS) sensors and air inlet.

By means of clustering, the number of load levels is reduced in such a way as to better represent the characteristics of single components. Nevertheless, to preserve accuracy, this number cannot

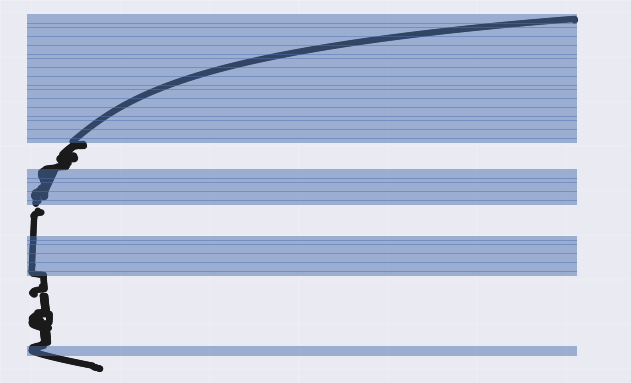


Figure 4.10: The four load regions used in the LFM parametrization.

get as low as four. Therefore, each load region being parametrized may consist of a few load levels, manually selected. The parameters are, then, applied to all levels in the region.

4.2.1 *Geometry Parameters*

Geometrical variations modify the x- and z-positions of the load regions. The aim does not lie in finding a new front-end styling; rather, in identifying the location where the load regions should act. How these are *covered* is mainly a creative task and is up to designers. The variations are implemented as rigid *translations* in the xz-plane, respectively Δx and Δz, where each load region is moved independently. This may create either overlap between regions or some empty areas. However, this does not cause numerical issues. In fact, contact is not defined among load levels, but only between load levels and impactor. Therefore, an overlap simply simulates a situation where the sum of the stiffnesses of the load levels in the overlapping portion is applied and an empty area one where no structurally relevant components are present.

The lower load region is kept fixed in x and used as reference position. This avoids to obtain the same relative positions with different sets of parameter values. In z this issue does not apply, as,

although the load regions have the same relative positions, their position would be different relative to the impactor, thus giving a different kinematics. Nevertheless, the height of the bumper beam is not linked to pedestrian protection requirements, rather to front and low-speed crash ones. Therefore, the z-position of the middle load region, usually placed in front of the bumper beam, is kept fixed. Instead, the *deformation space* in front of the bumper beam is strictly linked to pedestrian leg impact requirements. Hence, an additional parameter, $u_{DS_{mid}}$, is implemented for the middle load region that defines the deformation at which a rigid behavior starts. The stiffness here is set constant to $1\,\mathrm{kN\,mm^{-1}}$. It is worth noting, that this deformation may not be reached during the simulation, depending on the actual intrusion.

The load levels in the upper load region have recorded the bonnet stiffness according to only one specific impactor kinematics –i. e., with the aPLI UBM impacting at a precise location. In general when changes are applied, but especially when shifting the bonnet, the area of impact of the UBM changes and the LFM may not be valid anymore. Therefore, I discourage large geometrical variations of the upper load region.

In outwards shooting positions, the angle of tapering of the front-end –i. e., from a top view the angle with respect to the y-axis– affects the impactor rotation around the z-axis during the simulation. Therefore, that angle could be a relevant parameter for the pedestrian leg impact; however, it is not implemented in this work.

To sum up, I define a maximum of 7 geometry-related parameters: Δz_{low}, Δx_{mid}, $u_{DS_{mid}}$, $\Delta x_{mid\text{-}up}$, $\Delta z_{mid\text{-}up}$, Δx_{up}, Δz_{up}.

4.2.2 *Stiffness Parameters*

Variations related to the stiffness of a load region modify the shape of the force-deformation characteristics of the levels in that region. For the loading phase, two parameters, s_1 and s_2, are defined: they scale the curve, respectively, at zero deformation and at the last point of deformation recorded during the HFM simulation, similarly to what is proposed in [88]. If only s_1 is used, the curve

is scaled globally, i. e., the parameter acts as a *scaling factor* of the whole curve. Each entry of the force for the i-th spring defined in the load level is simply given by:

$$F_i^* = s_1 F_i \tag{4.10}$$

where the star symbol indicates the scaled quantity. If both s_1 and s_2 are used, a linear scaling function over the deformation is defined and each entry of the force is computed as:

$$s = s_1 + \frac{s_2 - s_1}{u_{i_{max}}} u_i \qquad F_i^* = s F_i \tag{4.11}$$

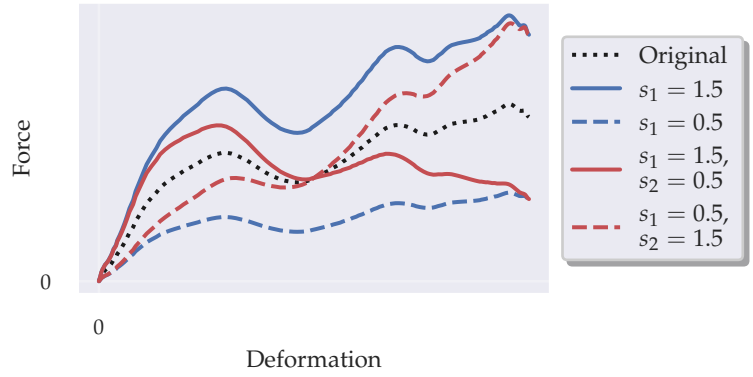
where $u_{i_{max}}$ is the maximum intrusion of the i-th component. Figure 4.11a shows an exemplary curve, scaled with either one or two parameters, and Figure 4.11b the respective scaling functions over the deformation.

Instead of expressing the variations relative to an original shape, a completely new force-deformation curve could be defined. I prefer using scaling parameters for the following reasons:

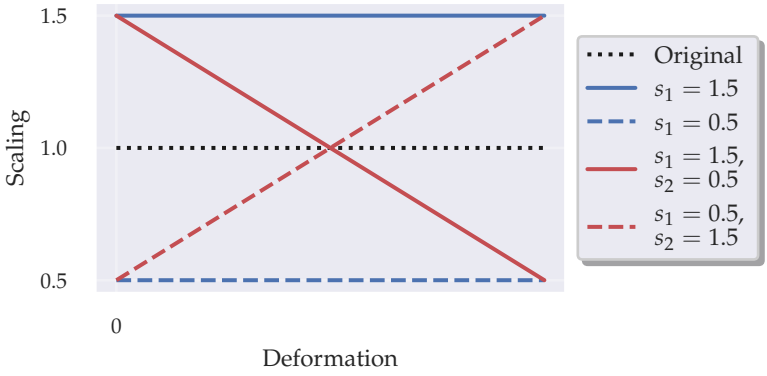
- the need to apply the same parameter to each level defined in the load region,
- the need to keep the number of parameters low,
- the simplicity to express variations relative to the original curve,
- the flexibility in shapes it still enables.

In fact, it is rather unlikely that all the levels in a load region have the same force-deformation characteristics. Therefore, a curve for each one should be defined, increasing significantly the number of parameters. A cubic curve, for instance, needs three points to be specified –the fourth one being the origin– and often offers too much flexibility in shape for the limited deformation space to be accurately realized in real components. Furthermore, in order to communicate the variations, it is easier to express them in relation to a known construction, rather than starting from scratch.

In the early stages of the vehicle development process, rather general indications on the force-deformation characteristics are



(a)



(b)

Figure 4.11: Example of scaling parameters on the force-deformation characteristics: (a) shows the scaled curves, (b) shows the respective scaling functions over the deformation.

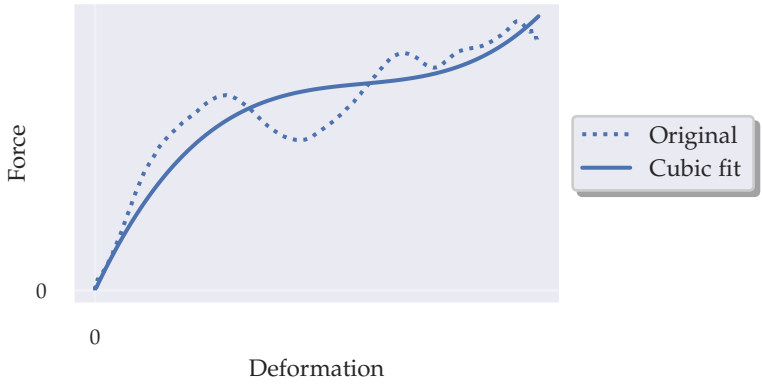


Figure 4.12: Example of cubic polynomial fit of a force-deformation characteristics.

sought. The reference characteristics may be taken even from previous similar vehicle models. Therefore, it is convenient to *smooth out* small oscillations in the reference curve and improve generality of the results. A cubic polynomial fits sufficiently well the reference curve, as shown exemplarily in Figure 4.12. The curve is, then, extrapolated as discussed in Section 4.1.5. Simulations show that the effect of small oscillations in the reference curve is rather small. Nevertheless, in later stages, precise information about the force-deformation characteristics are sought. Here, the reference curve typically belongs to the model under development and the indications should be as close as possible to reality; in these cases, I recommend using the original reference curve.

For the unloading phase, I vary directly the *dissipation factor D*, defined in Section 4.1.4. I assign the same dissipation factor to each level in the load region. Although being an approximation, it is a very convenient way, with a single parameter, to control the energy that the load region returns.

Linked to how the load levels deform is the inclination of the springs. This may play a large role, especially in the lower and upper load regions. Nevertheless, it would be difficult to precisely transfer the inclination back to the HFM and, therefore, I do not

employ it as a parameter in this thesis. I suggest its usage in targeted investigations to understand the way load regions should deform.

To sum-up, I define a maximum of 12 stiffness-related parameters: $s1_{low}$, $s2_{low}$, D_{low}, $s1_{mid}$, $s2_{mid}$, D_{mid}, $s1_{mid\text{-}up}$, $s2_{mid\text{-}up}$, $D_{mid\text{-}up}$, $s1_{up}$, $s2_{up}$, D_{up}.

4.3 NOVELTY OF THE PROPOSED METHOD

The idea of using a simplified model of the vehicle front-end for investigations of the pedestrian leg impact is nothing new. Though, to the best of my knowledge, the way I generate and use it has some original aspects. The major ones are:

DEGREES OF FREEDOM: state-of-the-art approaches use one spring per load level [51, 89]; the use of three orthogonal, translational springs allows a more faithful approximation of both forces and deformations of the HFM. Especially when modifications are applied to the original model, a system of three springs avoids the introduction of artificially-large stiffness in the orthogonal direction of deformation.

CLUSTERING: the approach in [89] selects the load levels of the model and groups some of them for the parametrization. The definition of 10 mm high load levels, instead, allows a much finer record of the local deformations in the HFM. Furthermore, the subsequent clustering combines together those levels deforming similarly so as to better approximate the force-deformation characteristics of single energy-absorbing structures in the front-end.

PARAMETRIZATION: state-of-the-art approaches mostly use idealized force-deformation characteristics or focused on specific load regions [88, 94]; I define simple –yet quite comprehensive– parameters that involve both geometry and stiffness of the front-end, are easy to understand and are suitable to both early and late design stages.

IMPACTOR: no previous work has developed an LFM and run front-end design analyses with the aPLI.

Some minor aspects that are peculiar of my work concern the *unloading* –through the definition of the dissipation factor directly in the LFM– the *extrapolation* –by averaging the slope of the last 25% of the force-deformation curve– and the *node* selected for the deformation in the load level –that is, the one with largest intrusion.

5

DESIGN FLEXIBILITY OPTIMIZATION OF COMPUTATIONALLY EXPENSIVE BLACK-BOX FUNCTIONS

Your answers ought to be general, not generic.

— Rossana Negri[1]

In Chapter 4, I proposed a method to generate and parametrize a simplified vehicle model for the pedestrian leg impact. Concept engineers can use it in early phases of the development process to run first investigations and determine what is necessary to fulfill the requirements. In this regard, a global sensitivity analysis would be useful to rank the importance of the parameters and identify a subset of the most relevant ones. Even more useful would be to find out, not only on which parameters to focus, but also how their value should be; in other words, to provide design engineers with *guidelines* about optimal parameters set-up. According to the Cambridge Dictionary, guidelines are "*information intended to advise people on how something should be done*". Ergo, they would represent a good starting point for the design process of vehicles front-end for the pedestrian leg impact and, thus, support the initial development stages. Furthermore, they would boost understanding of the complex link between vehicle and impactor. This know-how can, then, guide some trial-and-error that may take place in later stages.

The conventional approach is to identify *one* feasible set of parameters, i. e., one that fulfills the requirements. However, in the early stages, this is of little help; essentially, because of two reasons:

1. Lots of uncertainties are present at the beginning of the development process about its evolution. Several construc-

1 Directed at me, during an oral exam of the Philosophy class at high-school.

tion decisions must still be taken and the initial design may undergo significant changes.

2. The parameter values belong to the LFM. It is unlikely that these values can be *exactly* matched by a HFM. No information is provided about how the impactor's response changes when those values are not met.

Therefore, one single feasible set-up provides little help. Useful guidelines would recommend, instead, *a range* of values for each design parameter, where the requirements are satisfied. Namely, useful guidelines should offer *flexibility*.

In the context of design under uncertainty, the term flexibility was initially introduced by Grossmann and Morari [42] to describe the ability of a chemical plant to satisfy specifications despite the variations that may occur in the parameter values during operation. According to the Merriam-Webster Dictionary, flexibility is the "*ready capability to adapt to new, different, or changing requirements*". Therefore, the aim of this chapter is the *optimization* of the capability of the design parameters to adapt to new requirements, not specific to the pedestrian leg impact, that may arise during the development process, while still fulfilling the pedestrian leg impact ones.

The word optimization comes from the Latin *optimus*, superlative of *bonus*, good, and means to take the quality of an object to its highest degree. *True optimization* is outside the scope of this work. In fact, due to the relatively high-dimensionality of the problem under investigation and the stochastic approach used to assess flexibility –described in Section 5.2– the highest degree of flexibility of a design is not known. More pragmatically, I aim for a *bonization*, that is, for a satisfactory degree of flexibility given the circumstances. Nonetheless, for sake of clarity, the word optimization is used in the rest of the thesis.

Figure 5.1 shows an overview of the steps taken in the current chapter. In the first section, I describe an adaptive strategy to train a surrogate model for feasibility analysis; in the second one, I propose a revision of an existing algorithm to compute flexibility-oriented design guidelines.

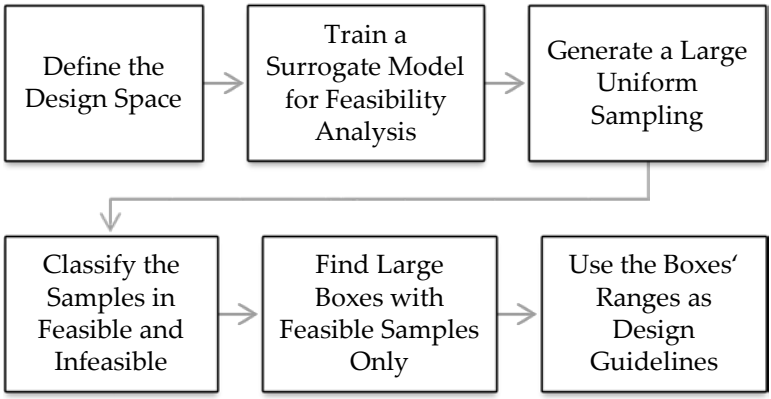


Figure 5.1: Broad overview of steps described in Chapter 5.

5.1 BAYESIAN FEASIBILITY ANALYSIS

In order to identify feasible intervals of parameter values, it is first necessary to mathematically define what feasible means. A convenient way to do this is by means of the feasibility function ψ [44].

Let $x_i \in \mathbb{R}$ be a real-numbered value of the design parameter i.

Let $\mathbf{X} := \prod_{i=1}^{d} \left[x_{i_L}, x_{i_U}\right] \subset \mathbb{R}^d$ be a compact, convex subset of the Euclidean space of dimension d, $\mathbb{R}^d$, obtained by Cartesian product of d intervals. $\mathbf{X}$ is commonly called design space.

Let $\mathbf{x} := (x_1, \ldots, x_d) \in \mathbf{X}$ be a vector in $\mathbf{X}$, i. e., a d-tuple of input parameter values. $\mathbf{x}$ is commonly called sample or, simply, point.

Let $f : \mathbf{X} \to \mathbb{R}$ be an output scalar quantity. $f_j(\mathbf{x})$, then, refers specifically to the output response f_j of the sample $\mathbf{x}$. For the pedestrian leg impact, f_j represents, in general, the j-th injury criterion.

Let f_{j_t} be the threshold value for the response f_j. Therefore, $f_j \leq f_{j_t}$ is a requirement to fulfill. This can be rewritten in the form of an inequality constraint g:

$$g_j(\mathbf{x}) = \frac{f_j(\mathbf{x})}{f_{j_t}} - 1 \leq 0. \tag{5.1}$$

Hence, if g_j is non-positive, the j-th requirement is satisfied. It follows that, if the condition $g_j \leq 0$ is true for each $j \in \{1, \ldots, m\}$, where m is the number of requirements, $\mathbf{x}$ is said to be feasible. It comes in handy to define a function that *quantifies* the feasibility of a sample $\mathbf{x}$. That is the feasibility function ψ, introduced by Halemane and Grossmann [44]:

$$\psi(\mathbf{x}) = \max_{j \in \{1,\ldots,m\}} g_j(\mathbf{x}) \tag{5.2}$$

which measures how far the most critical response is, relative to its threshold value. By way of example, a value $\psi = 0.1$ means that the most critical response is 10% higher than its threshold. Accordingly, if $\psi > 0$, there exists at least one requirement that is not fulfilled, and if $\psi \leq 0$, every requirement is fulfilled. All the samples $\mathbf{x} \in \mathbf{X}$, for which the condition $\psi(\mathbf{x}) \leq 0$ holds, belong to the feasible region of the design space. $\psi = 0$ identifies, then, the *boundary* of the feasible region. The aim of feasibility analysis is to determine the feasible region of the design space [130].

The algorithm to obtain the guidelines –described in the second part of the chapter– requires, first, a large number of samples and, second, the category to which they belong: feasible or infeasible. One FE simulation of the LFM for pedestrian leg impact, introduced in Chapter 4, still takes too long to run large data sets in a reasonable amount of time. For computationally expensive models, mathematical, data-driven, approximation methods are often introduced to describe the input-output behavior of the simulation model. The physical nature of the problem is assumed to be a black-box and only the relationship between output responses and input parameters is investigated, as shown in Figure 5.2. The challenge lies in the precise identification of this relationship on the basis of a limited number of training samples. The described approximation methods are known under several names, among which surrogate models [100], response surface models [64] or metamodels [129]. In this thesis, I use the name surrogate model.

The efficiency of surrogate models is exploited for many purposes, such as:

GLOBAL APPROXIMATION: to describe the model behavior in the whole input space,

Figure 5.2: Sketch of a single-output black-box model.

GLOBAL OPTIMIZATION: to direct the search towards the global optimum,

RELIABILITY ANALYSIS: to compute the failure probability of a system,

FEASIBILITY ANALYSIS: to classify the samples based on their feasibility status.

Models used for the global approximation require high global accuracy, with no preferred region of the space to focus on. Those used for the last three purposes, instead, do still require a general global accuracy, but present region of interest in the space, where high accuracy is more important. In global optimization, this is the region close to the global optimum. In reliability analysis, it is the one where limit states contribute to system failure. In feasibility analysis, it is the boundary of the feasible region.

Depending on the purpose of the surrogate model, the way to train it differs. There are, essentially, two sampling approaches:

UNIFORM The sampling plan is designed to fill out the entire input space, maximizing the uniformity; e. g., a common space-filling design is Latin Hypercube Sampling (LHS) [7, 85]. A thorough discussion on computer experimental designs can be found in [105]. The sampling budget is, generally, used in one solution. For deterministic computer simulations, the goal of an experimental plan should be to reduce systematic errors, rather than random errors [104]. Therefore, uniformity is desired for global accuracy.

ADAPTIVE The sampling strategy is divided into two phases. The initial phase generates a uniform sampling; however, this

time not all the sampling budget is used. Then, an iterative phase starts, where the samples are specifically chosen so as to provide maximum information on the region of interest, based on current knowledge. It is, therefore, a greedy approach. Usually, a figure of merit, also called *infill criterion*, is defined that quantifies the quality of a sample. It aims to balance *exploitation* of the surrogate model prediction and *exploration* of the design space. The former tries to locate samples where the prediction value is interesting, while the latter where the prediction error is high. An overview of infill criteria is given in [58]. After each iteration, the surrogate model is updated with the newly evaluated samples. The iterative phase stops when either the sampling budget is over or a satisfactory accuracy is reached. Adaptive sampling, also called sequential sampling in statistics, aims to reduce the computational effort –i. e., the number of evaluations– necessary to reach the same accuracy in the region of interest with respect to a uniform approach. Popularity of adaptive sampling has grown remarkably since the work of Jones et al. [59] in the context of Bayesian global optimization. In machine learning, the key principle related to adaptive sampling –namely to let a learning model query for new data points at self-chosen locations– is called *active learning* [109].

In this thesis, I use surrogate models for feasibility analysis, i. e., to assess the feasibility status of a sample. Since this is a binary classification task, a uniform degree of accuracy over the whole design space is not necessary. Rather, the prediction accuracy should be especially high in the area between feasible and infeasible categories –that is, close to the feasible region boundary. I adopt, therefore, an adaptive sampling technique.

Similarly to the Bayesian approaches for global optimization [50, 59, 86], I use Gaussian processes to approximate the objective function. The definition *Bayesian* is due to the probabilistic approach of the approximation: Gaussian process can be considered a Bayesian statistical method. In fact, they assume a *prior distribution*[2] on the

2 Details on Gaussian process theory will follow in Section 5.1.1.

underlying function, analogously to the a-priori probability in Bayesian decision theory [87]. Therefore, to be consistent with the global optimization nomenclature, I call the described feasibility analysis strategy, Bayesian feasibility analysis.

5.1.1 *Gaussian Process Regression*

Theories related to Gaussian processes go back more than a century.[3] During this period, the mathematical foundation has been re-invented several times and termed with many different names; this already suggests how powerful the theory is. Generally accepted names are Kriging, stochastic process, Best Linear Unbiased Estimator (BLUE) and Wiener-Kolmogoroff prediction. Gaussian process is the most commonly used name in the context of machine learning, presumably because it hints to the mathematical foundation of the model. The theory became popular thanks to the formulation in the field of geostatics by French mathematician Matheron [83], which builds on the Master's thesis of South African mining engineer Krige [74] on the prediction of gold distribution on the basis of few boreholes. The modern theory refers to the book of Rasmussen and Williams [99].

The historical digression points out the significance of the theory in several fields. This finds two main explanations. For a given finite set of observations, there is an infinite number of functions passing through them. In simple terms, Gaussian processes:

1. assign a *probability* to each possible function. That can be used to predict, on the one hand, which function most likely interpolates the data –given by the mean of the probability distribution– on the other hand, the error associated with this prediction –given by the standard deviation.

2. are versatile, as they can generate functions with any shape, and so are suitable to interpolate *non-linear* and multimodal problems.

3 Early work dates back to 1880 by Danish astronomer Thorvald N. Thiele [76]. The primitive theory was, then, formalized by American mathematician Wiener [136] and Soviet mathematician Kolmogoroff [67] for time series analysis.

The first strength comes from the probabilistic approach typical of Gaussian distributions; the second one from the *non-parametric* modeling approach. While parametric approaches –e. g., polynomial regression– assume *a certain* shape of the function and optimize the parameters –e. g., the coefficients of the polynomial– that ensure the best fit to the training data, non-parametric approaches –e. g., Gaussian processes, Support-Vector Machine (SVM) or Artificial Neural Network (ANN)– may assume *any* shape and automatically learn the actual shape of the function from the training data, *after* optimization of so-called *hyper-parameters*. On the one hand, the model learns the best fitting parameters for a certain shape; this is called parametric machine learning. On the other hand, the model learns the best fitting shape for a certain set of hyper-parameters; this is called non-parametric machine learning. The latter is way more powerful, as it enables much more flexibility and can fit any data, though at the cost of many more samples required to be accurate.

Gaussian processes have an important drawback, particularly relevant in high-dimensional spaces. In fact, their computational complexity is $\mathcal{O}(n^3)$, i. e., cubic with the number of training samples n [5]. Therefore, as they are not sparse –namely they use the whole amount of training samples to compute a prediction– they may become very expensive for large data sets, as it is often required in high-dimensional spaces.

After an introduction on pros and cons of Gaussian processes, some math is needed to describe the fundamentals of the regression method.

Let $\{\mathbf{x_1}, \dots, \mathbf{x_n}\} \in \mathbf{X}$ be a finite set of n arbitrary points in the design space.

Let $y \colon \mathbf{X} \to \mathbb{R}$ be an output scalar quantity. $y(\mathbf{x})$, for any point $\mathbf{x} \in \mathbf{X}$, is the function to model. In Gaussian process theory, the quantity $y(\mathbf{x_i})$, at location $\mathbf{x_i}$, is considered a random variable.

Here the complex bit: a Gaussian process is defined as a set of random variables $\{y(\mathbf{x_1}), \dots, y(\mathbf{x_n})\}$ that has a multivariate Gaussian distribution [99]. This is a generalization of the univariate Gaussian distribution for more than one variable. As the one-dimensional Gaussian distribution is fully determined by mean and variance, the multivariate one is fully determined by the mean

function, denoted as $\mu(\mathbf{x})$, and the covariance function, also called *kernel*, $k(\mathbf{x_i}, \mathbf{x_j})$:

$$\begin{aligned} \mu(\mathbf{x}) &= \mu(y(\mathbf{x})) = \mathbb{E}\left[y(\mathbf{x})\right] \\ k(\mathbf{x_i}, \mathbf{x_j}) &= \operatorname{cov}(y(\mathbf{x_i}), y(\mathbf{x_j})) = \\ &= \mathbb{E}\left[\left(y(\mathbf{x_i}) - \mu(\mathbf{x_i})\right)\left(y(\mathbf{x_j}) - \mu(\mathbf{x_j})\right)\right]. \end{aligned} \tag{5.3}$$

In Bayesian statistics, the distribution of possible functions $y(\mathbf{x})$, before any observation is taken into account, is called prior. Exemplary functions from this distribution are shown in Figure 5.4a. As there is no information on the underlying function yet, namely no observations, the prior distribution is often assumed to have mean constant and equal to zero. This means that the average of all possible functions $y(\mathbf{x})$ at any point $\mathbf{x}$ equals zero. With this assumption, mean and kernel of the Gaussian Process (GP) prior, in Equation 5.3, simplify to:

$$\begin{aligned} \mu(\mathbf{x}) &= 0 \\ k(\mathbf{x_i}, \mathbf{x_j}) &= \mathbb{E}\left[y(\mathbf{x_i})y(\mathbf{x_j})\right]. \end{aligned} \tag{5.4}$$

Assuming a zero mean does not affect the shape of the possible functions. Specifying the covariance function does. In fact, this defines the similarity between pairs of function values, $\left(y(\mathbf{x_i}), y(\mathbf{x_j})\right)$, and so the function's degree of smoothness. The covariance is assumed to be a function of the distance between $\mathbf{x_i}$ and $\mathbf{x_j}$ only. Therefore, in case of isotropy, as assumed in this thesis, it assumes the form:

$$k(\mathbf{x_i}, \mathbf{x_j}) = k(\left\|\mathbf{x_i} - \mathbf{x_j}\right\|) \tag{5.5}$$

where $\|\cdot\|$ indicates the Euclidean norm. With the kernel it is possible to specify which shape the function most likely has, based on the a priori knowledge. A GP, whose prior has constant mean and covariance function that depends only on the distance between two points, is said to be *stationary*.

A common kernel, because of its flexibility, is the Matérn covariance function [99], introduced by the Swedish statistician Matérn [82]:

$$k(\mathbf{x_i}, \mathbf{x_j}) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{\left\|\mathbf{x_i} - \mathbf{x_j}\right\|}{l}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\frac{\left\|\mathbf{x_i} - \mathbf{x_j}\right\|}{l}\right) \tag{5.6}$$

where

ν is a positive parameter called smoothness coefficient,

l is a positive parameter called length-scale,

$\Gamma(\nu)$ is the gamma function,

K_ν is the modified Bessel function of the second kind.

At zero distance, $k(\mathbf{x_i}, \mathbf{x_i}) = 1$, meaning that the function interpolates the points.

The smoothness coefficient ν defines the class of differentiability of the function at zero distance, $\|\mathbf{x_i} - \mathbf{x_j}\| = 0$. For $\nu > p$, the function $y(\mathbf{x})$ is p-times continuously differentiable. The smoothness affects also how the prediction uncertainty grows close to a point. The higher ν, the more similar two distinct points are, the smoother the function is, the less uncertain the prediction close to a point is. For $\nu = 1/2$, the well-known exponential covariance function is obtained, which is C^0 continuous, while for $\nu \to \inf$, it tends to another well-known covariance function, the squared exponential, also called Gaussian, which is infinitely differentiable. The decay in similarity for different values of ν as function of the distance between two points is shown in Figure 5.3. Popular choices of ν are $3/2$ and $5/2$. Due to the nature of the functions that I aim to approximate, namely responses of a non-linear FE simulation, I use $\nu = 3/2$, which generates C^1 continuous functions. For this choice, the covariance function simplifies to:

$$k(\mathbf{x_i}, \mathbf{x_j}) = \left(1 + \sqrt{3}\frac{\|\mathbf{x_i} - \mathbf{x_j}\|}{l}\right) \exp\left(-\sqrt{3}\frac{\|\mathbf{x_i} - \mathbf{x_j}\|}{l}\right). \quad (5.7)$$

The length-scale, as the name suggests, scales the distance between two points. Ergo, it controls how close two points $(\mathbf{x_i}, \mathbf{x_j})$ must be to consider their function values $(y(\mathbf{x_i}), y(\mathbf{x_j}))$ similar. The higher l, the more similar two points at unit distance are, the smoother the function is, the less uncertain the prediction close to a point is. The length-scale is the hyper-parameter of Equation 5.7.

A squared term σ_f^2, called signal variance, is sometimes used to multiply Equation 5.6 and Equation 5.7. It defines the variance of

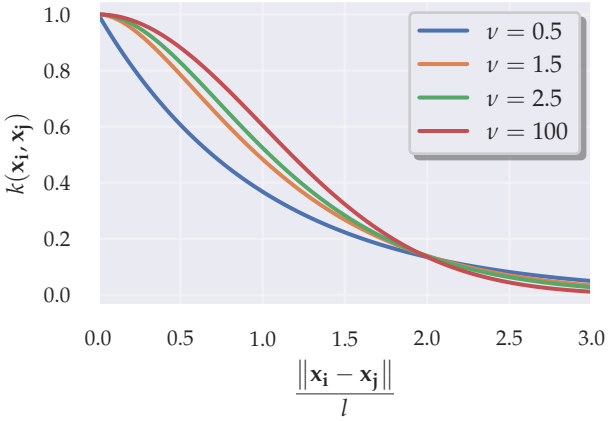


Figure 5.3: Matérn covariance as function of the distance between two points for different values of the smoothness coefficient ν, according to Equation 5.6.

the function $y(\mathbf{x})$ –whose mean is zero, as defined in Equation 5.4– and would be another hyper-parameter of the covariance function. However, in order to keep low the number of hyper-parameters to infer from the training samples, I assume a unitary variance.

Figure 5.4a shows some randomly sampled realizations of the prior distribution with zero mean and kernel as defined in Equation 5.7 for a one-dimensional case. For the plot, a length-scale of 0.1 is assumed. The functions have different shape, but same smoothness. However, they do not approximate any specific problem, as no observation is taken into account. In other words, the model is still untrained.

Let $X = [\mathbf{x_1} \ \dots \ \mathbf{x_N}]^T$ be a $N \times d$ matrix of N training samples –where d is the dimensionality of the design space– and $\mathbf{x}_*$ be a test sample.

Let $\mathbf{y} = [y(\mathbf{x_1}) \ \dots \ y(\mathbf{x_N})]^T$ be a column vector of N observations, called training outputs, and $y_* = y(\mathbf{x}_*)$ the unknown function value to predict, called test output.

Let $\mathbf{K}_{XX}$ be the $N \times N$ covariance matrix –whose entry $K_{\mathbf{x_i x_j}}$ is equal to $k(\mathbf{x_i}, \mathbf{x_j})$– and $\mathbf{k}_{X\mathbf{x}_*} = [k(\mathbf{x_1}, \mathbf{x}_*) \ \dots \ k(\mathbf{x_N}, \mathbf{x}_*)]^T$ the col-

umn vector of covariances between training points and test point. Then, the multivariate probability distribution of the training outputs $\mathbf{y}$ and test output y_* is Gaussian and given by:

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K}_{XX} & \mathbf{k}_{X\mathbf{x}_*} \\ \mathbf{k}_{X\mathbf{x}_*}^T & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right). \tag{5.8}$$

Finally, the Gaussian probability distribution of y_* is obtained by conditioning the prior, right hand-side of Equation 5.8, on the observations $\mathbf{y}$. In simple words, the conditioning operation is equivalent to force the distribution of functions given by the prior, such as those in Figure 5.4a, to pass through the observations. Details on this operation are more cryptic and can be found in [99]. The resulting expected value, $\hat{y}_*$, and variance, σ_*^2, of the prediction at location $\mathbf{x}_*$ are computed as:

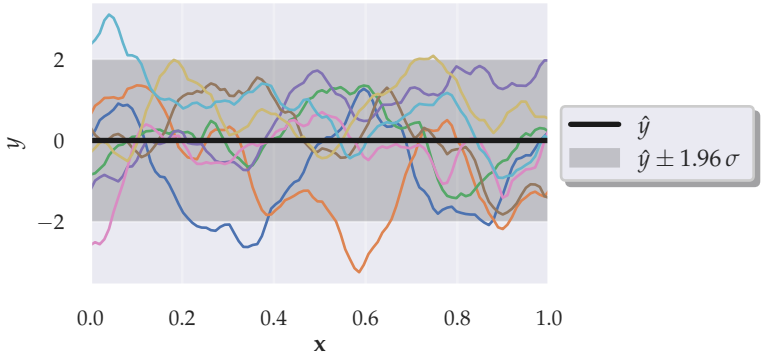
$$\begin{aligned} \hat{y}_* &= \hat{y}(\mathbf{x}_*) = \mathbf{k}_{X\mathbf{x}_*}^T \mathbf{K}_{XX}^{-1} \mathbf{y} \\ \sigma_*^2 &= \sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_{X\mathbf{x}_*}^T \mathbf{K}_{XX}^{-1} \mathbf{k}_{X\mathbf{x}_*}. \end{aligned} \tag{5.9}$$

Equation 5.9 describes the posterior distribution of y_* at location $\mathbf{x}_*$, i. e., the distribution obtained once information is inferred from the available observations. The posterior distribution of functions $y(\mathbf{x})$ is shown in Figure 5.4b with a $\nu = 3/2$ Matérn covariance function. The length-scale is chosen to best fit the data. The shape of the functions is much more similar, once training points are taken into account, and the uncertainty overall smaller. Therefore, the model has, partially, *learned* the shape of the underlying function.

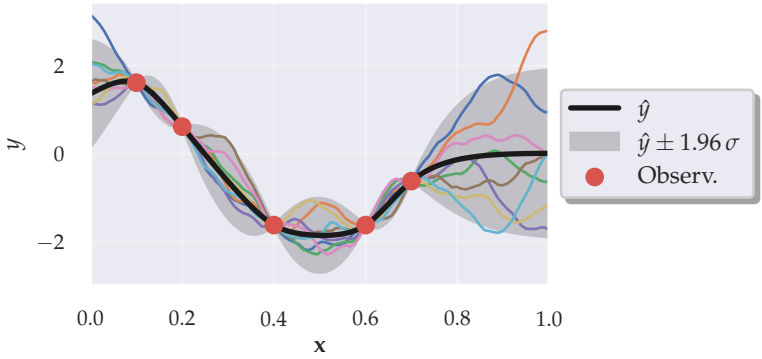
Looking at Equation 5.9, it is now clear where the drawback of Gaussian processes comes from. The larger the number of training samples N, the larger the size of the covariance matrix $\mathbf{K}_{XX}$, the more computationally expensive it is to invert it. As told already in undergraduate math courses, matrix inversion is no fun.

$\hat{y}_*$ in Equation 5.9 can be interpreted as a linear combination of N covariance functions centered, respectively, at $\mathbf{x_1}, \ldots, \mathbf{x_N}$:

$$\begin{aligned} \hat{y}(\mathbf{x}_*) &= \sum_{i=1}^{N} \alpha_i \, k(\mathbf{x_i}, \mathbf{x}_*) \\ \boldsymbol{\alpha} &= \begin{bmatrix} \alpha_1 & \ldots & \alpha_N \end{bmatrix}^T = \mathbf{K}_{XX}^{-1} \mathbf{y}. \end{aligned} \tag{5.10}$$



(a)



(b)

Figure 5.4: Randomly sampled functions $y(\mathbf{x})$, mean and interval at 95% confidence level of Gaussian process: (a) prior distribution, as in Equation 5.4, with Matérn covariance function ($\nu = 3/2$, $l = 0.1$); (b) posterior distribution, as in Equation 5.9, with Matérn covariance function ($\nu = 3/2$, $l = 0.182$).

Noteworthy, although the mean of the prior is assumed to be zero, the mean of the posterior is not necessarily zero. Nevertheless, this tends to zero in the area where no information –i. e., no observation– is available, as in the right side of Figure 5.4b.

Most numerical simulations are noisy. Therefore, no observations of the actual underlying function are available, but only noise-corrupted values. In this case, it is inconvenient to interpolate the data, as it is likely to lead to a problem known as *overfitting*. This occurs when the model overestimates the changes in the underlying function.[4] In such cases, to pass *close* to the observations may be a better approach than passing through them. This can be achieved by introducing a term σ_n^2, called noise variance, in the covariance matrix of the training inputs $\mathbf{K}_{XX}$:

$$\mathbf{K}_{XX_n} = \mathbf{K}_{XX} + \sigma_n^2 \mathbf{I} \tag{5.11}$$

where $\mathbf{I}$ is the identity matrix. Noise is assumed to have a Gaussian distribution with zero mean and variance σ_n^2.

Equation 5.11 adds a variance to the diagonal of the covariance matrix, namely to the covariance of each training point with itself, $k(\mathbf{x_i}, \mathbf{x_i})$. This is also called nugget effect. By substituting $\mathbf{K}_{XX_n}$ with $\mathbf{K}_{XX}$ in Equation 5.9, the posterior distribution for GP regression for noisy training data is obtained [99].[5] A tiny noise term may be beneficial also for deterministic observations, as it enhances the numerical stability of matrix computation by preventing singular matrices, similarly to the Tikhonov regularization [40]. Furthermore, Gramacy and Lee [41] argue that it is better to add a noise term and use a flexible kernel, rather than interpolating the data with a very smooth kernel.

To sum up, the Gaussian process learns the input-output relationship of the underlying model only through the available set of observations and the prior distribution. As the former may be limited, the only way to bring some know-how into the predictions is to work on the prior. Because the mean is assumed

4 Overfitting may be especially relevant when the training points are non-uniform, with some very close to each other –as it happens, for instance, around the region of interest with adaptive sampling.

5 Note that the prediction is noise-free, unless the noise variance is added to the predicted variance. That is fine, because goal of the prediction is to approximate the noise-free underlying function.

zero, specifying a proper covariance function is of the utmost importance.

In this thesis, I use the implementation of GP regression in the `python` library `scikit-learn` [97]. It is often recommended to not assume too much smoothness in the data and include a regularization term [21, 40, 99]. For this reason and on the basis of the above discussion, I employ the Matérn covariance function with smoothness coefficient $\nu = 3/2$, as in Equation 5.7, together with a noise term, as in Equation 5.11. The two hyper-parameters, l and σ_n^2, are optimized by Maximum Likelihood Estimation (MLE) [110], which aims at finding the hyper-parameters that most likely explain the data. As likelihood function, the logarithm of the marginal likelihood is used [99]. Due to the multimodal form of this function, the optimizer is started in 15 different initial configurations. Despite the computational overhead, hyper-parameters MLE is performed each time new observations are acquired, namely at each adaptive sampling iteration, in order to enhance prediction accuracy, as suggested in [113]. The hyper-parameters are optimized inside the range:

LENGTH-SCALE: $l \in \left[10^{-1}, 10^{3}\right]$,

NOISE VARIANCE: $\sigma_n^2 \in \left[10^{-7}, 10^{-3}\right]$.

According to the discussion following Equation 5.7, it is clear that specifying too small length-scales is dangerous. In fact, the approximation function may get too flexible and likely cause overfitting [21]. Flexibility is especially risky in high-dimensional spaces, where, because of limited sampling budget, distances among samples tend to get larger. In this case, smooth functions are preferable. Since I normalize the design space used for the GP to the unit hyper-cube, a lower limit of 0.1 for the length-scale looks appropriate. 10^{-7} as lower limit of the noise variance is considered enough for regularization. As I use Gaussian processes to approximate normalized constraints, of the type in Equation 5.1, an upper limit of 10^{-3} expresses a standard deviation of about 3% of the threshold of the measured response and looks appropriate.

5.1.2 *Expected Improvement for Feasibility Analysis*

The predictions on mean and variance, in Equation 5.9, can be used to iteratively improve the GP accuracy close to the feasible region boundary, and so in feasibility analysis. New samples should be chosen at each iteration on the basis on an infill criterion, which is required to guide the search towards the feasible region boundary. On the one hand, the criterion should exploit the current information and favor points whose expected value lies on the boundary. On the other hand, it should promote the exploration of unsampled areas and favor points whose prediction –although not expected to lie on the boundary– have a high uncertainty, because of which they may still lie on the boundary. Therefore, new samples should contribute to relocate the feasible region boundary more precisely –or in other words, with less uncertainty.

In this thesis, I use the Expected Improvement (EI) for feasibility analysis infill criterion, EI_{feas}, introduced by Boukouvala and Ierapetritou [11]. This is based on the EI criterion, first formulated by Močkus [86] and successfully applied by Jones et al. [59] for efficient, surrogate-based, global optimization. The EI founds on the definition of an improvement.

Let $\mathbf{x}$ be a candidate sample for the next iteration.

Let $y = y(\mathbf{x})$, at location $\mathbf{x}$, be a random variable that belongs to a GP, as described in the previous subsection. Therefore, it has a Gaussian distribution, with mean $\hat{y}$ and standard deviation σ, according to Equation 5.9.

GLOBAL OPTIMIZATION The improvement, I, is non-zero for values $y < f_{\min}$, where y is the quantity to minimize and $f_{\min} = \min(y(\mathbf{x_1}), \ldots, y(\mathbf{x_n}))$ is the current best observation. Simply put, the optimum is *improved* if the sample's value is lower than the current ones. I is defined as [108]:

$$I = \begin{cases} f_{\min} - y & \text{if } y < f_{\min} \\ 0 & \text{otherwise} \end{cases} = \max\left(0, f_{\min} - y\right) \tag{5.12}$$

i. e., if $y < f_{\min}$, the further y from $f_{\min}$, the more the optimum is improved. Accordingly, the *EI* at location $\mathbf{x}$, $EI = EI(\mathbf{x})$, is [59]:

$$\begin{aligned} EI &= \mathbb{E}[I] = \mathbb{E}[\max(0, f_{\min} - y)] = \\ &= (f_{\min} - \hat{y})\,\Phi\left(\frac{f_{\min} - \hat{y}}{\sigma}\right) + \sigma\varphi\left(\frac{f_{\min} - \hat{y}}{\sigma}\right) \end{aligned} \tag{5.13}$$

with $EI = 0$ if $\sigma = 0$, where $\varphi(\cdot)$ and $\Phi(\cdot)$ indicate, respectively, the probability density function and the cumulative distribution function of the standard normal distribution. Equation 5.13 is derived in Section A.1.

FEASIBILITY ANALYSIS The improvement, I_{feas}, is non-zero for values $y \leq 0$, if $\hat{y} > 0$, and for values $y \geq 0$, if $\hat{y} \leq 0$, where y approximates the feasibility function ψ of the underlying problem, defined in Equation 5.2. Simply put, if the sample is predicted as infeasible, $\hat{y} > 0$, the feasibility analysis capability of the surrogate model is *improved* only in case the sample turns out to be feasible, $y \leq 0$; and vice versa. I_{feas} is defined as [131]:

$$I_{feas} = \begin{cases} \hat{y} - y & \text{if } y \leq 0 < \hat{y} \\ y - \hat{y} & \text{if } \hat{y} \leq 0 \leq y \\ 0 & \text{otherwise} \end{cases} \tag{5.14}$$

i. e., if the sample is predicted as infeasible, $\hat{y} > 0$, *and* the sample is actually feasible, $y \leq 0$, the further y from $\hat{y}$, the more feasible the sample, the larger the prediction error, the more the feasibility analysis capability of the surrogate model is improved; and vice versa. Noteworthy, if $\hat{y} \neq 0$, I_{feas} equals $\mid \hat{y} \mid$ at $y = 0$ and presents a discontinuity. Accordingly, the *EI* for feasibility analysis at location $\mathbf{x}$, $EI_{feas} = EI_{feas}(\mathbf{x})$, can be conveniently computed as [11]:

$$EI_{feas} = \mathbb{E}\left[I_{feas}\right] = \sigma\varphi\left(\frac{\hat{y}}{\sigma}\right) \tag{5.15}$$

with $EI_{feas} = 0$ if $\sigma = 0$. Equation 5.15 is derived in Section A.2.

For the sake of completeness, it is worth mentioning that Shintani et al. [112] use the standard *EI* as infill-criterion to improve

the feasibility analysis capability of the surrogate model. They set then the current value to improve $-f_{\min}$ in Equation 5.13– to zero. However, with this approach, samples are not added around the feasibility boundary, but rather mostly inside the feasibility region. Hence, the feasibility analysis capability of the surrogate model may improve, but this is not the direct goal of the EI formulation –as it is, instead, for EI_{feas}.

Figure 5.5a shows, by way of example, I_{feas} for the case $\hat{y} > 0$, together with a possible probability density function of y. Accordingly, the probability density function of I_{feas} –and its expected value EI_{feas}– can be determined, as shown in Figure 5.5b.[6]

$\Phi(t)$ expresses the probability that the function's value is lower or equal to t and monotonically increases as t increases. Therefore, the first term of EI, in Equation 5.13, increases as $f_{\min} - \hat{y}$ increases or as the prediction uncertainty σ decreases. This term exploits the surrogate model prediction to seek the optimum.

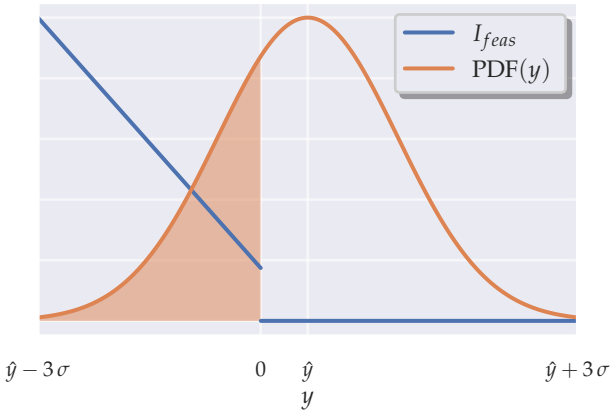
$\varphi(t)$ expresses the probability that the function's value is equal to t; it is symmetrical and maximum for $t = 0$. Therefore, the second term of EI increases as $\hat{y}$ tends to $f_{\min}$ or as σ increases: it explores the design space to seek points whose value is similar to the current optimum, but with high uncertainty.

EI_{feas}, in Equation 5.15, has the same form of the second term of EI, yet does not only perform exploration. Rather, it combines exploitation and exploration in a single term. In fact, EI_{feas} increases as $\hat{y}$ tends to zero, which is the region of interest –namely the feasibility boundary– or as σ increases, which occurs in unexplored regions. This is confirmed by the partial derivatives of EI_{feas} with respect to $\hat{y}$ and σ.

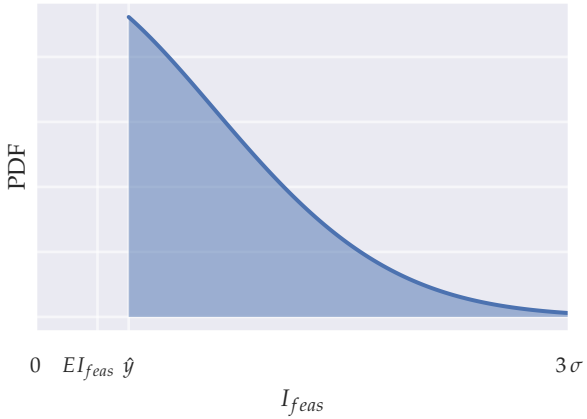
Recalling the expression of the probability density function for a standard Gaussian distribution, I can rewrite EI_{feas} as:

$$EI_{feas} = \sigma \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\hat{y}}{\sigma}\right)^2\right) \tag{5.16}$$

6 If $\hat{y} - 3\sigma \geq 0$, EI_{feas} would be, to a first approximation, zero.



(a)



(b)

Figure 5.5: (a) Improvement for feasibility analysis, I_{feas}, and possible Probability Density Function (PDF) of y, for the case $\hat{y} > 0$, according to Equation 5.14.
(b) PDF of I_{feas} in (a): the corresponding expected value EI_{feas} is also reported.

and compute the partial derivatives:

$$\begin{aligned}\frac{\partial EI_{feas}}{\partial \hat{y}} &= -\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\hat{y}}{\sigma}\right)^2\right)\frac{\hat{y}}{\sigma} \\ \frac{\partial EI_{feas}}{\partial \sigma} &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\hat{y}}{\sigma}\right)^2\right)\left(1+\left(\frac{\hat{y}}{\sigma}\right)^2\right).\end{aligned} \tag{5.17}$$

$\partial EI_{feas}/\partial \hat{y}$ and $\hat{y}$ are of opposite signs: if $\hat{y}$ is positive, $\partial EI_{feas}/\partial \hat{y}$ is negative, so EI_{feas} increases as $\hat{y}$ decreases; vice versa if $\hat{y}$ is negative. Therefore, EI_{feas} increases as $\hat{y}$ tends to zero. $\partial EI_{feas}/\partial \sigma$ is strictly positive. Therefore, EI_{feas} increases monotonically as σ increases. Although a noise term is defined in the covariance function, as in Equation 5.11, σ is very low at sampled locations; thus, EI_{feas} does not tend to focus on a very local area over the iterations. This is crucial, as the feasible region boundary is, in general, a hyper-surface, and not a single point similarly to the global optimum.

Figure 5.6 shows a comparison of EI and EI_{feas} for the one-dimensional case in Figure 5.4b. As $\hat{y}$ barely goes below the best observation, EI is mainly related to the second term in Equation 5.13 and driven by high prediction uncertainty. EI_{feas} presents two peaks in proximity of $\hat{y} = 0$, whose magnitude mainly depends on the standard deviation in these locations.

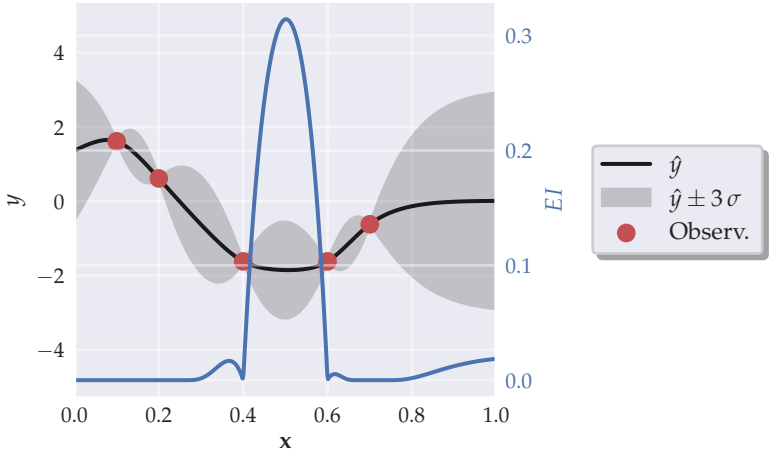
The most promising location to sample is the one, $\mathbf{x}^*$, that maximizes EI_{feas}:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{X}} EI_{feas}(\mathbf{x}). \tag{5.18}$$

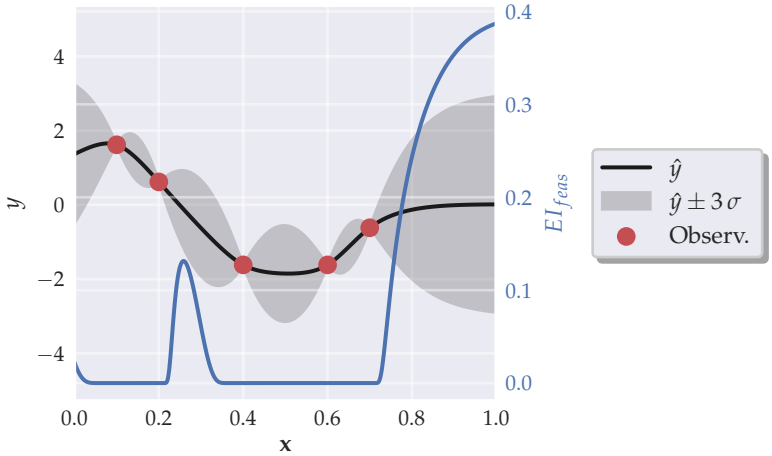
Therefore, an optimization algorithm is required.

5.1.3 *Multimodal Optimization*

Recent advances in large-scale computing, such as, HPC or cloud computing, offer the capability to efficiently run more than one simulation at the time. Several observations could be, therefore, added at each iteration and run *in parallel*. This would accelerate the adaptive sampling process, because, on the one hand, more



(a)



(b)

Figure 5.6: Comparison of expected improvement formulations for the 1-dimensional case in Figure 5.4b: (a) EI for global optimization; (b) EI_{feas} for feasibility analysis.

simulations would be completed in the same time period, and on the other hand, less iterations would be necessary to reach a satisfactory level of accuracy. When many samples are used to fit the surrogate model, the training phase gets computationally expensive; thus, reducing the number of iterations would be advantageous. Nevertheless, not all the sampling budget should be utilized in a single iteration in order to capitalize on the Bayesian approach and infer information from the observations as they become available.

As feasibility analysis is interested in a region, rather than a single point, sampling the design space in multiple locations at each iteration is potentially even more beneficial than for global optimization. For this one, a special form of Expected Improvement is first defined by Schonlau [108] and used by Ginsbourger et al. [34] to find the best set of q points that improves the estimate of the global optimum at each iteration; it is called multi-points EI or q-EI. As each sample contributes to the global optimum only to the extent that it is actually better than the current optimum, the multi-points improvement, q-I, is defined as the maximum of the improvements of each sample:

$$q\text{-}I = \max\left(I(\mathbf{x_1}), \ldots, I(\mathbf{x_q})\right) \tag{5.19}$$

where $I(\mathbf{x})$ is specified in Equation 5.12.

The expected value of q-I, q-EI, is not a simple scalar as the standard EI, but requires high-dimensional numerical integration. For instance, the 2-EI is derived explicitly in [34] and consists of the sum of the Expected Improvements of the two samples plus some correlation factors that depend on their relative distance. For larger q, Ginsbourger et al. [34] propose a sequential approach where q sub-iterations are carried out at each adaptive sampling iteration: at each sub-iteration, the standard EI is optimized and the surrogate model is updated with the estimated value of the obtained sample. However, the surrogate model requires q training phases per each iteration, which may be a major computational effort.

Feasibility analysis looks for a region, rather than a single point. Therefore, the multi-points improvement for feasibility analysis cannot be simply defined as the maximum of the improvements

I_{feas} of each sample. In fact, more than one sample contributes, in general, to the representation of the feasible region boundary. An alternative would be to define the multi-points improvement for feasibility analysis as the sum of the improvements I_{feas} of each sample. This is not correct either: if two points are coincident, the improvement would be double the real one. A definition of the multi-points improvement for feasibility analysis does not seem straightforward. To my knowledge, there is no definition for it in literature.

Rogers and Ierapetritou [101] add several observations per iteration by clustering large values of EI_{feas} in the design space and using the clusters' centers as starting points for local optimization. This approach assumes, implicitly, the multi-points improvement for feasibility analysis as the sum of I_{feas} of each sample, yet prevents the samples from being too close to each other. Furthermore, as no sub-iteration is performed, only one training phase is required per iteration. I employ this approach in the present work; though, instead of a do-it-yourself combination of clustering and local optimization, I apply directly a multi-modal optimizer to EI_{feas} to find the set of samples for the iteration.

As demonstrative example, I use a modified version of the Branin function, defined as:

$$f(x_1, x_2) = \frac{\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1)}{250} \tag{5.20}$$

where $x_1 \in [-5, 10]$, $x_2 \in [0, 15]$. Figure 5.7a shows the contour plot of f. The Branin function is a popular choice as test function in multi-modal optimization problems, especially due to the presence of three global optima [108]. With respect to the version in [106, 108], I shift and normalize f so as to, first, create three feasible regions of reasonable size and, second, make it more representative of a feasibility function.

Maximizing EI_{feas} is not a trivial task. Figure 5.7b shows EI_{feas} for the modified Branin function –used as feasibility function– and a set of 60 uniformly-distributed training samples. EI_{feas} is:

- asymptotically zero in large areas of the space distant from the feasible region boundary,
- highly non-linear due to the variation of the predicted variance in proximity of training samples.

A multi-modal or a multi-start local optimizer, as in [10, 130], is, therefore, required. I employ the Niching Migratory Multi-Swarm Optimiser (NMMSO) developed by Fieldsend [30], freely available via the `python` package `pynmmso`.[7]

NMMSO is based on Particle Swarm Optimization (PSO), an approach first introduced by Kennedy and Eberhart [63] and become very popular in global optimization due to its simplicity and rapid convergence. Each particle of a swarm is a point that moves iteratively in the design space based on its previous movement direction, its personal best location and the swarm best location. By balancing these three components, a compromise between exploration and exploitation in the optimization search can be found.

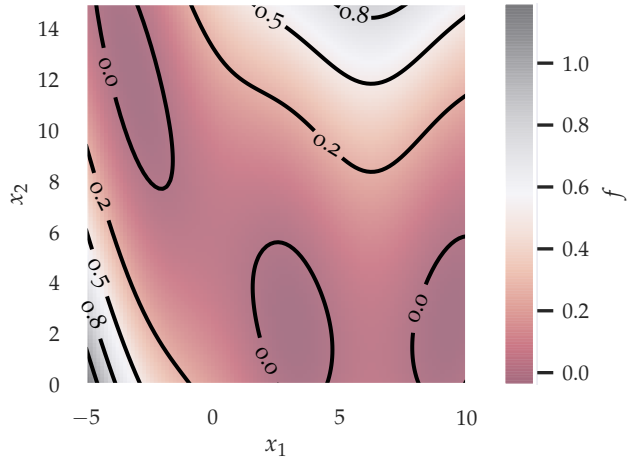
Niching multi-modal optimization does not search for one solution, $\mathbf{x}^*$, as in Equation 5.18, but for multiple ones, namely for all the sampling locations that maximize the objective function in isolated peak regions, also called *niches*. The first niching PSO technique was introduced by Brits et al. [13].

NMMSO uses multiple PSOs with strong local search, i. e., greater exploitation than exploration, to rapidly optimize separate local maxima. The sub-swarms do not work independently. Rather, particles may migrate to another sub-swarm if they find a peak different than the one of the parent sub-swarm. Furthermore, sub-swarms may be merged if they converge on the same peak or if the distance among them is smaller than a tolerance one, default set to 10^{-6}. Additionally, new regions can be identified by splitting particles of existing sub-swarms.

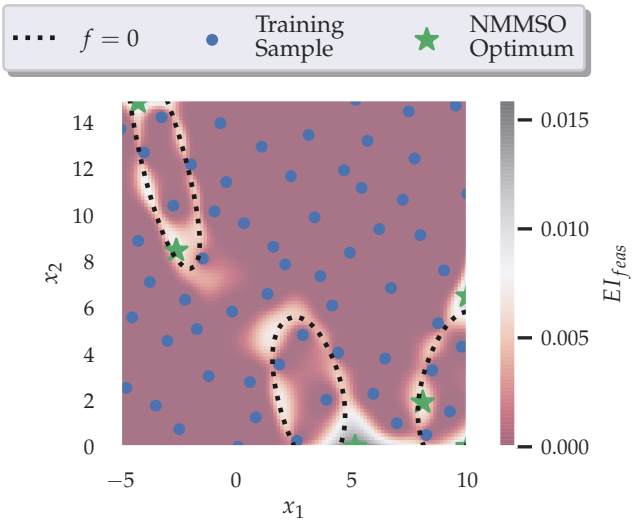
In conclusion, NMMSO presents three important features that make it an effective multi-modal optimizer:

- it dynamically identifies the number of modes,

7 The NMMSO algorithm won the 2015 IEEE CEC Competition on Niching Methods for Multimodal Optimization [80].



(a)



(b)

Figure 5.7: (a) Contour plot of the Branin function, as defined in Equation 5.20;
(b) Contour plot of EI_{feas} according to a set of 60 uniformly-distributed training samples for the Branin function in (a); the optimal sampling locations computed with NMMSO are reported as well.

- it self-adapts to the problem type, with few meta-parameters to be set a priori,
- it rapidly climbs the peak regions [30].

I use the default meta-parameters of the NMMSO algorithm to optimize the EI_{feas} function. Figure 5.7b reports the optima returned by NMMSO for the EI_{feas} of the Branin function in Figure 5.7a. The optimizer successfully identified the most significant 6 peaks.

I define a maximum amount of points to be sampled at each iteration, fixed across the iterations in order to have an approximately uniform expansion of the data set in the adaptive phase. If the number of samples returned by the optimizer is larger than the maximum allowed, those with lowest EI_{feas} are discarded.

Furthermore, in order not to sample points with disproportionate and too low EI_{feas} values, I define a minimum EI_{feas} threshold below which a sample is discarded: EI_{feas} must be larger than one third of the maximum value found in the iteration, $\mathbf{x}^* \mid EI^i_{feas}(\mathbf{x}^*) > EI^i_{feas_{max}}/3$, where the superscript i indicates the i-th adaptive phase iteration.

5.1.4 *Validation Metrics*

A critical step in training a surrogate model is its validation, i. e., the determination of its degree of accuracy in the domain of application [65]. One or more quantitative measures should be defined to assess whether the model can be used or requires further training data –as long as further data can be collected. In case of adaptive sampling, the assessment should be carried out both before and after the adaptive phase.

Due to the cardinal role validation metrics play in the methodology, they are examined in details in the following. As the GP regression model is used in this thesis for both regression –during the adaptive sampling phase– and classification –for feasibility analysis– validation metrics for both tasks should be considered.

REGRESSION is a predictive mapping from the design space to a continuous scalar quantity. Therefore, it is possible to measure

the scalar differences between true and predicted values, usually called residuals.
Let y be the observed value of a continuous variable, $\hat{y}$ its predicted value and n the number of observations.
The simplest metric that evaluates the accuracy of the prediction is the Mean Absolute Error (MAE):

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{5.21}$$

which is an equally-weighted average of the residuals. For this reason, the MAE is robust against outliers. If, instead, large deviations should not be considered outliers and are particularly undesirable, more weight can be attributed to them. This is obtained, e. g., with the Mean Squared Error (MSE):

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} \tag{5.22}$$

or its square-rooted variant, the Root Mean Squared Error ($RMSE$):

$$RMSE = \sqrt{MSE} \tag{5.23}$$

which, over the MSE, has the important property of being expressed in the same unit of measure of y and represents the standard deviation of the residuals. It holds the condition $RMSE \geq MAE$.
Both MAE and $RMSE$ are absolute measures of the goodness of fit and are, therefore, related to the specific quantity y. Nevertheless, in order to compare the surrogate model accuracy among different applications, a relative measure is required. A well-known relative measure is the coefficient of determination, usually denoted R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{5.24}$$

where $\bar{y} = \sum_{i=1}^{n} y_i / n$ is the mean of the observations. R^2 represents the proportion of variance of the output parameter y that is explained by the input parameters in the set of observations [22].

More intuitively, it measures how good the surrogate model predicts the observed data set with respect to a model that constantly predicts the mean of the observations.
The coefficient R^2 is defined in the range $(-\infty, 1]$. If:

$R^2 < 0$, the surrogate model prediction is worse than the mean prediction,

$R^2 = 0$, the surrogate model prediction is as good as the mean prediction,

$0 < R^2 < 1$, the surrogate model prediction is better than the mean prediction,

$R^2 = 1$, the surrogate model makes no prediction error *in the observed data set*.

R^2 strongly depends on the size of the data set and its uniformity in the design space. In fact, few observations that accidentally have similar values are very well predicted by the mean, but may be slightly mispredicted by a rather globally accurate surrogate model; thus, giving negative R^2. A similar situation happens in case samples are not uniformly distributed, but mostly concentrated in one region of the space with similar output values: R^2 may be low, but not representative of the global accuracy of the surrogate model. This is the case of adaptive sampling for feasibility analysis, where the added observations mostly lie around $y = 0$.
A variant of R^2, the adjusted R^2 –indicated as $\bar{R}^2$– is often considered in feature selection to adjust for the number of explanatory input parameters d relative to the number of observations n, via the term $(n-1)/(n-d-1)$ [123]. Nevertheless, as $n \gg d$ in my applications and feature selection is outside the scope of this work, I simply consider R^2.
Further information on the evaluation of regression models can be found in [22].

CLASSIFICATION is a predictive mapping from the design space to a category, i. e., a discrete quantity. Therefore, the prediction may either be *true*, if the predicted category is the same as the

		OBSERVED CATEGORY	
		POSITIVE	NEGATIVE
PREDICTED CATEGORY	POSITIVE	True Positive (TP)	False Positive (FP)
	NEGATIVE	False Negative (FN)	True Negative (TN)

Table 5.1: Confusion matrix for binary classification.

observed one, or *false*, otherwise. In case of binary classification, as for feasibility analysis, the categories are usually called positive and negative.[8]

Metrics mostly focus on the positive class, regarded as the relevant one and usually smaller in size than the negative one. For sake of convenience with the metrics definition, I label the feasible samples as *positive* –although they have non-positive feasibility function ψ values– and the infeasible ones as *negative* –although they have positive ψ.

In order to define the evaluation metrics, it is useful to introduce the so-called confusion matrix, in Table 5.1. Accordingly, True Positive (TP) cases are samples both predicted and observed feasible, False Positive (FP) ones are predicted feasible but observed infeasible, False Negative (FN) ones are predicted infeasible, but observed feasible and, lastly, True Negative (TN) ones are both predicted and observed infeasible. In the metrics equations, the number of samples in each case is considered.

The *completeness* of the positive prediction is measured by the True Positive Rate (TPR), also called recall in computer science context and sensitivity in medical one:

$$TPR = \frac{TP}{TP + FN} \tag{5.25}$$

which is the portion of True Positives among all the observed positive cases.

8 This terminology originates in medical context, where patients are diagnosed either positive or negative to a disease.

The completeness of the negative prediction is measured by the True Negative Rate (TNR), also called specificity in medical context:

$$TNR = \frac{TN}{TN + FP} \tag{5.26}$$

which is the portion of True Negatives among all the observed negative cases.
The *exactness* of the positive prediction is measured by the Positive Predictive Value (PPV), also called precision in computer science context:

$$PPV = \frac{TP}{TP + FP} \tag{5.27}$$

which is the portion of True Positives among all the predicted positive cases.
The information from the three above-defined measures is often condensed in two scores, Balanced Accuracy (BA) and $F1$, regularly used for the evaluation of binary classification.

BA performs the arithmetic average of TPR and TNR:

$$BA = \frac{TPR + TNR}{2} \tag{5.28}$$

and represents the balanced completeness of the correct predictions, both positive and negative. It ranges from 0 to 1. It is an important measure for my application, as I aim to correctly predict both as many feasible and as many infeasible as possible. Nevertheless, it says little on the precision of the predictions if the categories have very different sizes.

$F1$ performs the harmonic mean of PPV and TPR:

$$F1 = 2\frac{PPV \cdot TPR}{PPV + TPR} \tag{5.29}$$

and represents the average of precision and completeness of the correctly predicted positive class. It ranges from 0 to 1. It is also an important measure, as I aim to predict as feasible samples that are actually feasible -thus, to be precise– and also as many of them as possible –thus, to be complete. Nevertheless, it is biased towards the positive class and does not consider the True Negatives.

An *all-in-one* solution is the Matthews Correlation Coefficient (*MCC*), formulated by Matthews [84]:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{5.30}$$

which takes into account the four cases of the confusion matrix in a way to balance categories of different sizes. It measures the correlation between predicted and observed data and, thus, ranges from -1 to 1. If:

$MCC = -1$, the predictions are in perfect disagreement with the observations,

$MCC = 0$, the predictions are in random agreement with the observations,

$MCC = 1$, the predictions are in perfect agreement with the observations.

In statistics, *MCC* is known as *phi coefficient*. *BA* and *MCC* are independent on which category is defined as positive or negative; *F1* is, instead, category-dependent. Chicco and Jurman [17] recommend the usage of *MCC* over accuracy and *F1*. Further information on the evaluation of classification models can be found in [122].

While for regression, the comparison of observation and prediction gives a real-valued output –i. e., the residual– in classification it gives only four outcomes, shown in Table 5.1. Therefore, classification measures cannot express an absolute error, such as, for instance, the *RMSE*, but only a relative one, which is computed on the basis of the number of samples with each outcome. Consequently, they strongly depend on the number of samples and their uniformity in the design space. If points lie close to the boundary between the categories, it is more likely that the classifier makes a mistake. This is the case, for instance, of adaptive sampling for feasibility analysis.

It is not only relevant which validation metric is used, but also *which data set* is used to compute it. In this thesis, I use two approaches, depending on the computational effort involved in a single function evaluation.

- For cheap-to-evaluate functions, such as mathematical functions, I generate a new LHS to test the accuracy of the trained surrogate model against the newly observed function values. Due to the uniformity of the samples, I evaluate both regression and classification metrics: *RMSE*, R^2 and *MCC*.

- For expensive-to-evaluate functions, such as LFM FE simulations, I use k-fold cross-validation. This technique is particularly useful to test the accuracy of a predictor *directly* on the available set of observations. In fact, this is split into k subsets of equal size: the model is trained with the union of $k-1$ subsets and tested against the remaining one. This procedure is repeated k times, so that each sample is once a test sample and $k-1$ times a training sample. Therefore, a prediction is obtained for each sample in the available set of observations. Noteworthy, the actual accuracy of the surrogate model trained with the entire data set is actually unknown. Nevertheless, the higher k, the better the approximation. As training the surrogate model k times may be computationally expensive with a large data set, I use $k = 10$, meaning that the model is repeatedly trained with 90% of the data. For the initial uniform sampling, I evaluate the accuracy of the surrogate model with the R^2 score. Since in the adaptive phase the samples are not uniformly distributed, I use the *RMSE* to check the accuracy at the end of this phase. Further details on cross-validation can be found in [66].

The metrics values obtained with new uniform test data and with cross-validation are compared in Section 6.2 by means of mathematical functions.

5.1.5 *Workflow*

After the description of the single pieces of the Bayesian feasibility analysis puzzle –namely surrogate model, infill criterion, new samples selection and validation metrics– I explain in the following how I compose the puzzle. In other words, I go through the steps that I use to train a surrogate model for feasibility analysis.

1. INPUT: Choose design parameters and respective ranges.

   It is an obvious task, though a fundamental one. In fact, it is the foundation of the entire analysis. Shan and Wang [111] examine strategies to deal with the curse of dimensionality [8] for computationally-expensive black-box functions, i. e., those for which only relatively few samples can be available.[9] An often-used approach is to screen irrelevant parameters with a sensitivity analysis. In this thesis, I choose those parameters for which I aim to get guidelines. Therefore, although a parameter may be less relevant than others, it is still important to consider it, if a guideline for it is required. In fact, the guideline fixes the interval of values that each parameter should assume in order to fulfill the requirements. If a parameter is not considered in the analysis, it is actually not proved that it may assume *any* value.
   Less effective than dimensionality reduction, yet still useful is the reduction of the parameter ranges. Since I aim to optimize the flexibility of the parameters, too small ranges would already be a limiting factor and may exclude interesting regions of the design space.
   Therefore, for both number of parameters, d, and associated ranges, $\left\{\left[x_{1_L}, x_{1_U}\right], \ldots, \left[x_{d_L}, x_{d_U}\right]\right\}$, in case of design flexibility optimization, a compromise based on engineering knowledge and requirements is recommended.

2. INPUT: Define sampling budget, ratio uniform/adaptive sampling and number of iterations in the adaptive phase.

   The amount of samples necessary to reach a satisfactory level of surrogate model accuracy depends on the problem dimensionality and the non-linearities of the underlying function in the design space –which are, however, difficult to foresee. Since each LFM FE simulation is computationally expensive, I define a maximum sampling budget, which represents a time and cost limit. I allocate more time, as the dimensionality d grows. This is rather natural: the bigger

9 Such strategies are problem decomposition, variables screening, variables mapping and design space reduction.

DIMENSION	SAMPLES	DIMENSION	SAMPLES
2	100	5	625
8	1600	12	3600
15	5625		

Table 5.2: Sampling budget for different dimensions computed with Equation 5.31.

the problem, the longer I am inclined to wait. To use an exponential function, even with base two, i. e., 2^d, would be too computationally expensive in high-dimensional spaces –say with 15 variables $2^{15} = 32{,}768$. Often-used rules of thumb for the initial sampling are either 10^d [10] or $10d$ [131]; nevertheless, the former is too expensive in high-dimensions, the latter too sparse. Therefore, I use a quadratic curve with base $5d$:

$$n = (5d)^2 \tag{5.31}$$

where n is the maximum number of samples allowed for the analysis. Table 5.2 reports the sampling budget for some dimensions used in this thesis.

The surrogate model accuracy after the initial sampling is crucial to be able to precisely identify the feasible region boundary during the adaptive phase. Furthermore, the focus in the present work is not to find the strategy to get the best surrogate model accuracy with the lowest possible amount of samples. Rather, it is to find a robust strategy to get good accuracy with a limited amount of samples. Robust is intended here in the sense that it should work for several dimensions and types of function. In fact, the most efficient strategy depends on the problem at hand [130]. For the above mentioned reasons, I assign more importance to the uniform phase, exploratory, than to the adaptive one, exploitative.

In *low-dimensions*, say up to 7D, I use 60% of the budget for the uniform sampling and a maximum of 40% for the adap-

tive one in 25 iterations, which means 1.6% per iteration.[10] In *high-dimensions*, say from 8D, the amount of samples given by Equation 5.31 is relatively scarce, considering the dimensionality, yet large in absolute sense. Therefore, on the one hand, the exploratory phase should be enhanced and, on the other hand, the number of iterations should be limited, as surrogate model training takes time. Accordingly, I assign 80% of the sampling budget for the uniform sampling and a maximum of 20% for the adaptive one in 10 iterations, which means 2% per iteration.

3. PROCESS: Generate uniform sampling.

   I use an improved LHS, originally introduced by Beachkofski and Grandhi [7] and implemented in the `python` library `diversipy` [134].

4. PROCESS: Run simulations and collect output quantities.

   I run FE simulations of the LFM for pedestrian leg impact described in Chapter 4.

5. PROCESS: Train surrogate models for feasibility function.

   In order to perform Bayesian feasibility analysis, the predictions of mean and standard deviation of the feasibility function, ψ, are needed. The simplest approach would be to directly train the surrogate model with ψ, as done in [10, 101, 130]. However, in case more than one constraint, g, is active inside the design space, ψ is C^0 continuous –i. e., is continuous, but its first derivative, $\psi\prime$, is not. Gaussian processes cannot accurately interpolate C^0 functions. Even though a C^0 covariance function is used, such as the exponential one, the points of non-differentiability are at the location of the training samples –see Section 5.1.1– and may not be where $\psi\prime$ presents the discontinuities. Adding a noise term improves the situation, as the function is not forced to over-fit the samples. Nevertheless, a much more significant improvement is obtained by training one surrogate

10 Depending on the outcome of NMMSO, the number of samples selected at each iteration may be less than the prescribed one.

model per each normalized constraint[11] and then compute ψ according to Equation 5.2. Nonetheless, the computational effort associated with the training of several surrogate models is considerable. Therefore, in case of LFM FE simulations with the aPLI, shown in Figure 1.3b, I split its 8 injury criteria in three groups and apply Equation 5.2 to each one. The groups consist of:

a) the four tibia bending moments,

b) the Medial Collateral Ligament (MCL) elongation[12],

c) the three femur bending moments.

The current GP implementation in `scikit-learn` does not allow to get the standard deviation for multiple outputs.[13] Hence, I train a single GP, with related hyper-parameter optimization, for each group's ψ. Noteworthy, this function is still, in general, non-differentiable, but with less points of discontinuity in its first derivative than the global ψ. The expected value and standard deviation of the global ψ are, then, equal to those of the group with the most critical –highest– expected value.
As zero is the value of the feasible region boundary –hence, is a meaningful value– it is often recommended to not use it as mean of the GP prior distribution [21]. Therefore, the observations are centered around the mean of the observations from the uniform sampling.

6. DECISION: Check surrogate model accuracy with cross-validation.

   I use the R^2 score, as samples are uniformly distributed.
   If $R^2 \geq 0.8$, move on; otherwise, increase amount of samples.

7. PROCESS: Select samples for iteration.

   I apply NMMSO to EI_{feas}. The number of selected samples is limited to the value chosen in step 2. The number of

11 This effect will be shown in Figure 6.13.

12 This group can be extended to consider also ACL and PCL, in case they should also be evaluated.

13 As of version 0.22.2

returned optima may be, however, lower. Therefore, the adaptive phase may use less sampling budget, than the allocated one.

8. PROCESS: Repeat steps 4 and 5.

9. DECISION: Check if iterations are over.

   If $iter > iter_{max}$, move on; otherwise, repeat steps 7 and 8.

10. DECISION: Check surrogate model accuracy with cross-validation.

    I compute the *RMSE* for observed feasibility function values ψ in range $[-0.1, 0.1]$. The global accuracy of the surrogate model was already tested in step 6. Here, the interest lies in its accuracy for feasibility analysis, namely close to the feasible region boundary $\psi = 0$. I do not check surrogate model accuracy at each iteration, since cross-validation is expensive.
    If $RMSE_{0.1} \leq 0.05$, move on; otherwise, consider increasing number of iterations for adaptive phase.

11. OUTPUT: Store trained surrogate models and training data.

    The surrogate models are ready to be used for feasibility analysis.

The described steps are condensed in Figure 5.8.

Rogers and Ierapetritou [101] adjust the feasibility function prediction by its prediction error. In other words, they do not consider feasible samples for which $\hat{\psi} \leq 0$, rather $\hat{\psi} + \sigma \leq 0$. However, since the method I propose mainly addresses the early stages of the development process, I consider such approach too conservative; some error can be still corrected in later stages. In fact, I aim to equip concept engineers with design guidelines, not strict rules to follow.

5.2 STOCHASTIC OPTIMIZATION OF FLEXIBILITY BOXES

At the beginning of the chapter, I stated the need for flexibility of design for each design parameter value in the early phases of

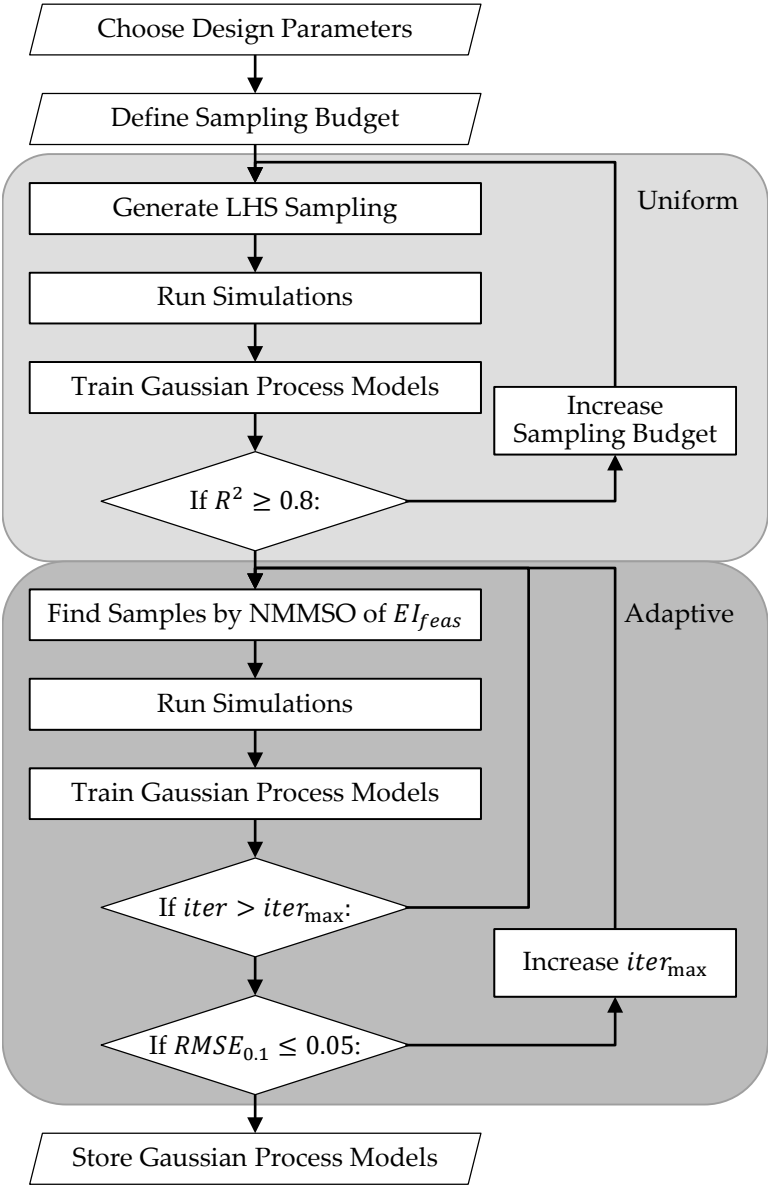


Figure 5.8: Workflow to train surrogate models for feasibility analysis. A detailed description can be found in Section 5.1.5.

the development process. In this section, I explain in detail what flexibility boxes are and how I compute them.

Let $x_i \in \mathbb{R}$ be a real-numbered value of the design parameter i.

Let $\mathbf{X} := \prod_{i=1}^{d} \left[x_{i_L}, x_{i_U}\right] \subset \mathbb{R}^d$ be a compact, convex subset of the Euclidean space of dimension d, obtained by Cartesian product of d intervals $\left[x_{i_L}, x_{i_U}\right]$. $\mathbf{X}$ is commonly called design space.

Let $\mathbf{x} := (x_1, \ldots, x_d) \in \mathbf{X}$ be a vector in $\mathbf{X}$, i. e., a d-tuple of input parameter values. $\mathbf{x}$ is commonly called sample or, simply, point.

Let $\psi \colon \mathbf{X} \to \mathbb{R}$ be the feasibility function, as defined in Equation 5.2.

Let $\mathbf{B} := \prod_{i=1}^{d} \left[x^B_{i_L}, x^B_{i_U}\right] \subseteq \mathbf{X} \mid \forall \mathbf{x} \in \mathbf{B} : \psi(\mathbf{x}) \leq 0$ be a compact, convex subset of the design space, obtained by Cartesian product of d intervals $\left[x^B_{i_L}, x^B_{i_U}\right] \subseteq \left[x_{i_L}, x_{i_U}\right]$, such that any included point is feasible. Just like $\mathbf{X}$, $\mathbf{B}$ is an axis-aligned hyper-rectangle, also called hyper-box and abbreviated as box in the following for sake of convenience. Unlike $\mathbf{X}$, however, $\mathbf{B}$ is a *feasible box*. $\mathbf{X}$ *may* be a feasible box if it does not contain any infeasible region.

Design flexibility optimization aims at quantifying the maximum limits of variation $\left[x^B_{i_L}, x^B_{i_U}\right]$ of each design parameter x_i such that the requirements are always fulfilled. In other words, it aims at maximizing the flexibility of $\mathbf{B}$. Quantitative flexibility measures are described in Section 5.2.2.

Axis-aligned feasible boxes, also called *solution boxes* [141], have an important property: they allow for decoupling of design parameters. In fact, each design parameter may assume *any* value inside its prescribed interval independently on the other parameters' values –as long as they also lie in their prescribed interval– and always fulfill the requirements. This means, on the one hand, that concept engineers have the possibility to choose the best setup for each parameter on the basis of considerations not directly considered in the feasibility analysis –e. g., manufacturability, weight or cost. On the other hand, it means that modifications of a parameter do not require modifications of the other ones as long as the modified design lies in the feasible box. This makes the box a container of a very robust set of solutions. Decoupling is especially beneficial in product development processes that involve complex interactions of design parts and synergies among

different working teams. In fact, each team can now focus only on the guideline assigned to it.

One feasible box may not include all the feasible solutions inside the design space, i. e., may not be equal to the feasible region. It would be convenient to get more than one feasible box. Concept engineers may, then, use the multiple guidelines for different purposes:

- in early stages, to choose the most suitable one based on considerations not directly taken into account in the feasibility analysis, as previously mentioned;
- in later stages, in case one or more parameters cannot lie in the prescribed range, to quickly find backup solutions or realize if feasibility can still be ensured, by simply checking whether the new values of the non-conform parameters lie in other boxes and how the other parameters are affected;
- to better understand how the feasible region expands in the design space.

In order not to carry the same information, the guidelines should be *different* among each other; I quantify this difference by the extent of overlap between two boxes and set an upper limit to it. Furthermore, they should not be contained in any others, i. e., the boxes should be *maximal*. Third, they should be limited in number, so as to be manageable: I look for the p boxes with *highest flexibility* level. On the basis of these requirements, I can formulate the optimization problem.

5.2.1 *Problem Formulation*

> Find the p maximal feasible boxes with highest flexibility level, whose relative overlap is lower than or equal to q_{max}.

Let $\mathbf{B} \subseteq \mathbf{X}$ be a feasible box inside the design space. If $\mathbf{B}$ exists –namely if there exists a feasible region in $\mathbf{X}$– and is not point-like, then there exists an infinite number of it, i. e., $\exists \mathbf{B_j} \mid \forall k \in \mathbb{N} \wedge j \neq$

$k : \mathbf{B_j} \neq \mathbf{B_k}$. In fact, as real numbers are infinite, there are infinite compact subsets of the Euclidean space.

Let $F \colon \mathbf{B} \to \mathbb{R}$ be a scalar quantity, representing a flexibility measure. $F(\mathbf{B})$, then, refers specifically to the flexibility F of the feasible box $\mathbf{B}$.

Let $\mathbb{B} := \left\{ \mathbf{B_j} \mid \forall j \in \mathbb{N} : \mathbf{B_j} \subseteq \mathbf{X} \right\}$ be the set of all, infinite feasible boxes inside the design space.

In order to derive the mathematical formulation, it is convenient to first break down the single requirements and then bring them together.

FLEXIBILITY LEVEL The objective of the optimization problem is to find the boxes with highest flexibility level. In case only one box is sought, then the formulation is simply:

$$\mathbf{B}^* = \underset{\mathbf{B} \subseteq \mathbf{X}}{\arg\max}\, F(\mathbf{B}). \tag{5.32}$$

If, instead, p boxes are sought, the formulation becomes more complex. Several objective functions can be defined. I maximize the sum of the flexibility measures of the boxes. Clearly, the p largest values in a set are those that give the largest sum. Thus, the unconstrained optimization formulation is:

$$\mathbb{B}^* = \underset{\mathbb{B}' \subset \mathbb{B},\, |\mathbb{B}'| = p}{\arg\max} \sum_{\mathbf{B} \in \mathbb{B}'} F(\mathbf{B}) \tag{5.33}$$

where $|\cdot|$ indicates the cardinality of the set.

MAXIMAL The term maximal is here adapted from graph theory, where cliques, i. e., connected graphs, are denoted as maximal if inextensible. Maximal feasible boxes are, therefore, those which cannot be enlarged without adding infeasible points. Not to be confused with the *maximum* feasible box, which is, instead, the one with the largest volume. The set of maximal feasible boxes is, then, formulated as:

$$\mathbb{B}_{max} = \left\{ \mathbf{B_j} \mid \forall \mathbf{B_j}, \mathbf{B_k} \in \mathbb{B} \wedge \mathbf{B_j} \neq \mathbf{B_k} : \mathbf{B_j} \not\subseteq \mathbf{B_k} \right\} \tag{5.34}$$

and is also an infinite set. Figure 5.9 shows an exemplary feasible region for a 2-D case and the corresponding set of maximal feasible boxes. Since the sought boxes with highest flexibility level

should not be contained in any others, maximality is a constraint for the optimization problem.

DIFFERENT The desired boxes should not only be inextensible, but also different one to the other. I measure the extent of similarity between two boxes $\mathbf{B_j}$ and $\mathbf{B_k}$ as the volume of their overlap normalized by the volume of the smallest box and denote it as $\mathbf{B_j} \cap \mathbf{B_k}$:

$$\mathbf{B_j} \cap \mathbf{B_k} = \frac{\prod_{i=1}^{d} \left(\min \left(x_{i_U}^{B_j}, x_{i_U}^{B_k} \right) - \max \left(x_{i_L}^{B_j}, x_{i_L}^{B_k} \right) \right)}{\min \left(\prod_{i=1}^{d} \left(x_{i_U}^{B_j} - x_{i_L}^{B_j} \right), \prod_{i=1}^{d} \left(x_{i_U}^{B_k} - x_{i_L}^{B_k} \right) \right)}. \quad (5.35)$$

The set of maximal feasible boxes whose relative overlap is lower than or equal to q_{max} can be, thus, formulated as:

$$\mathbb{B}_{q_{max}} = \left\{ \mathbf{B_j} \mid \forall \mathbf{B_j}, \mathbf{B_k} \in \mathbb{B}_{max} \wedge \mathbf{B_j} \neq \mathbf{B_k} : \mathbf{B_j} \cap \mathbf{B_k} \leq q_{max} \right\} \quad (5.36)$$

and is, in general, finite. Figure 5.10a shows an exemplary feasible region for a 2-D case and the corresponding set of largest maximal feasible boxes, whose maximum relative overlap is 0.5; only three boxes satisfy this condition. The maximum relative overlap is the last constraint for the optimization problem, which can be, finally, obtained by replacing $\mathbb{B}$ with $\mathbb{B}_{q_{max}}$ in Equation 5.33.

On the basis of the above described objective and constraints, the explicit mathematical formulation of the optimization problem is as follows:

$$\begin{aligned} \mathbb{B}^* = \arg\max_{\mathbb{B}'} \quad & \sum_{\mathbf{B_j} \in \mathbb{B}'} F(\mathbf{B_j}) \\ \text{subject to} \quad & \mathbb{B}' \subset \mathbb{B} \\ & |\mathbb{B}'| = p \\ & \mathbf{B_j} \not\subseteq \mathbf{B_k} \qquad \forall \mathbf{B_k} \in \mathbb{B} \wedge \mathbf{B_j} \neq \mathbf{B_k} \\ & \mathbf{B_j} \cap \mathbf{B_k} \leq q_{max} \qquad \forall \mathbf{B_k} \in \mathbb{B} \wedge \mathbf{B_j} \neq \mathbf{B_k}. \end{aligned} \quad (5.37)$$

5.2.2 *Flexibility Measures*

In order to optimize the design flexibility, this should be first quantified. A general rule is: the larger the ranges for each param-

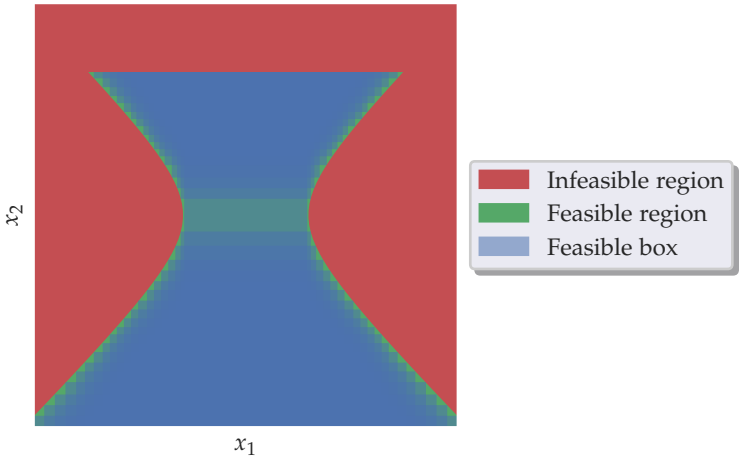


Figure 5.9: Maximal feasible boxes in an exemplary 2-D case with maximum relative overlap $q_{max} = 0.95\%$.

eter, the more flexibility the guideline offers. As a box is obtained by Cartesian product of intervals, intuitively, the larger its volume, the more flexibility it offers. Therefore, volume is a possible measure of the flexibility level of a box and used, for instance, in [39]:

$$F_V(\mathbf{B}) = \prod_{i=1}^{d} \left(x_{i_U}^B - x_{i_L}^B \right). \tag{5.38}$$

Maximizing the volume has, in general, advantages and disadvantages. First, it seeks the largest box and so the highest *combined* flexibility across the parameters. Moreover, volume is independent on the nature or unit of measure of parameters. Hence, it does not require values normalization. However, it may provide imbalanced flexibility across the parameters. In fact, some parameters may have very large ranges and others very narrow ones and *combined* give rise to the largest volume. Furthermore, it is not possible to define weights. As volume is obtained by multiplication, it does not matter how one parameter is scaled: all the boxes will be scaled equally.

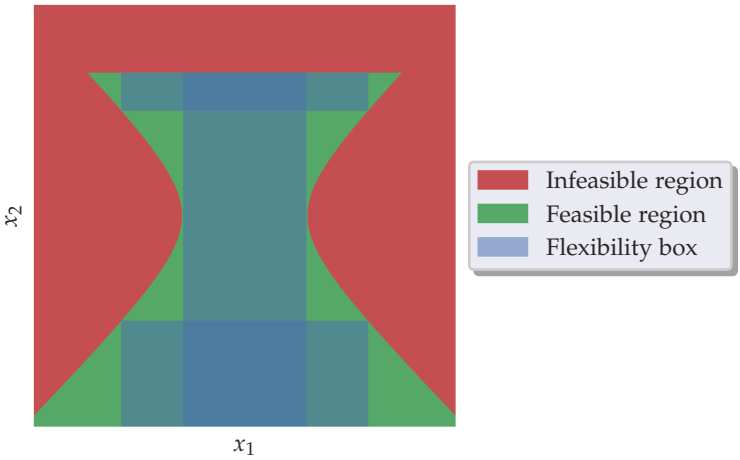
Balanced flexibility across parameters can be obtained by maximizing the minimum interval. This seeks, in general, the largest hyper-cube inside the feasible region. In fact, the minimum interval is largest when the flexibility level of the most critical parameter is as close as possible to that of the other ones. In other words, it tries to make the best compromise among the parameters and maximize uniformity. Nonetheless, as intervals are directly compared among each other, the unit of measure of the parameters does play a role. Therefore, a normalization is needed. Graff [38] uses the interval mean value as normalization factor. However, I find it rather dangerous, when the parameters have very different nature. As a matter of fact, if an interval is centered around zero, its normalized size will tend to infinite. On the contrary, if an interval mean value is very large, its normalized size will tend to a relative small value. Such a dependency is rather undesirable. Instead, I recommend to exploit the need for normalization to introduce a weight factor for the parameters. I specify the *required* minimum interval size for each parameter, $\widetilde{\Delta x_i}$, and define the flexibility measure based on the minimum interval width as:

$$F_I(\mathbf{B}) = \min_i \frac{x^B_{i_U} - x^B_{i_L}}{\widetilde{\Delta x_i}}. \tag{5.39}$$

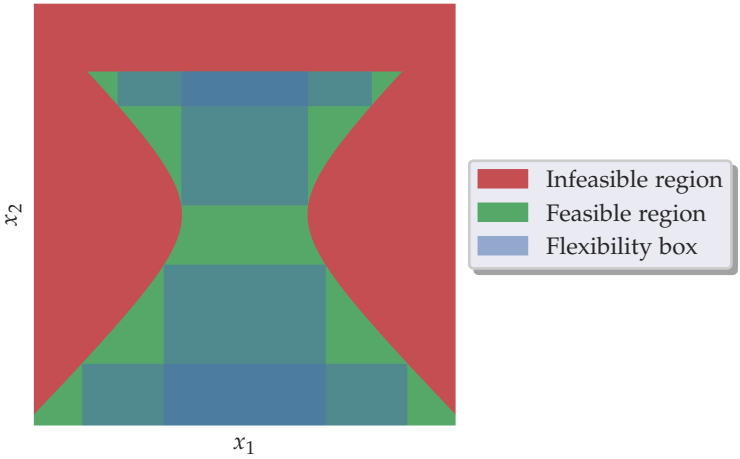
$\widetilde{\Delta x_i}$ is an user-defined quantity that expresses how large the interval size of parameter x_i should be so that x_i has a satisfactory level of flexibility. If $F_I(\mathbf{B}) \geq 1$, each parameter offers, at least, as much flexibility as required; otherwise, one or more parameters cannot ensure the required flexibility.

Figure 5.10 reports a comparison of the two flexibility measures on an exemplary 2-D case: Figure 5.10a shows the maximal boxes with highest flexibility level that satisfy the 50% maximum overlap constraint based on volume, while Figure 5.10b those based on minimum interval –in this case, the two parameters are equally-weighted. The latter presents two square-shaped boxes and two rather imbalanced ones that approximate the remaining feasible region.

Other flexibility measures definitions are, certainly, possible, though not investigated in this work.



(a) volume-based



(b) minimum-interval-based

Figure 5.10: Maximal feasible boxes with highest flexibility level –so called, flexibility boxes– whose relative overlap does not exceed $q_{max} = 0.50\%$, in an exemplary 2-D case: (a) based on volume; (b) based on minimum interval, where x_1 and x_2 are equally weighted.

5.2.3 *Stochastic Algorithm for Maximum Feasible Box*

In order to gradually increase the complexity, I first address the solution of the optimization problem in Equation 5.32, where a single box is sought.

To begin with, how do I determine whether a box is feasible? A non-point-like box contains an infinite number of points. It is, clearly, impractical to evaluate them all. I consider, then, three categories.

- In case the problem can be expressed in terms of linear constraint inequalities, the question can be answered analytically. This is done, for instance, in [29].
- In case the problem is not known in closed-form, yet the feasible region can be assumed convex, it is possible to claim that a box is feasible if its 2^d vertices are feasible.
- In case the constraints are treated as black-box functions –as in this work– evaluating only the vertices is not enough; *each* point inside the box must be tested to ascertain box's feasibility. The infinite number of them implies that *it is not possible* to assess *deterministically* whether a box is fully feasible. Even the evaluation of a coarse, equi-spaced grid becomes impractical in high-dimensional spaces.[14]

In the early phases of the development process, design guidelines should provide rather large ranges for each design parameter. It is not necessary at this point to ensure feasibility for each possible sample inside the box. A high percentage –say 85%– of feasible samples would be enough. In other words, I can state that, in an early stage, flexibility is more important than full feasibility. Because of this reason and of the impossibility to determine box's complete feasibility with black-box functions, Zimmermann and von Hoessle [141] relax the feasibility requirement by estimating the *probability of feasibility* and the respective confidence interval by means of Bayesian inference and Monte Carlo sampling. Such approach is described in more detail in Section 5.2.6.

14 An uniform grid of samples with 6 points per dimension contains $6^2 = 36$ points in 2 dimensions and $6^{10} = 60{,}466{,}176$ points in 10 dimensions.

Furthermore, they introduce a stochastic algorithm to find the maximum feasible box inside an arbitrary one on the basis of a uniform set of samples. The input box is shrunk to the largest one containing only feasible samples.

Let $\mathbf{C} := \prod_{i=1}^{d} \left[x_{i_L}^C, x_{i_U}^C\right] \subset \mathbb{R}^d$ be a compact, convex subset of the Euclidean space of dimension d, obtained by Cartesian product of d intervals $\left[x_{i_L}^C, x_{i_U}^C\right]$. $\mathbf{C}$ is the input box where the maximum feasible box is sought. It is convenient to initialize it as a unit hyper-cube: $\mathbf{C} = \prod_{i=1}^{d} [0, 1]$. It can be, then, the normalized design space or a normalized subset of the design space.

Let $\mathbb{X} := \{\mathbf{x_1}, \ldots, \mathbf{x_n}\} \in \mathbf{C}$ be a finite set of n arbitrary points uniformly distributed in the input space.

Let $\mathbb{X}_{OK} := \{\mathbf{x} \in \mathbb{X} \mid \psi(\mathbf{x}) \leq 0\} \subseteq \mathbb{X}$ be the set of feasible points in the initial set and $\mathbb{X}_{NOK} := \{\mathbf{x} \in \mathbb{X} \mid \psi(\mathbf{x}) > 0\} \subseteq \mathbb{X}$ be the set of infeasible ones. $\mathbb{X}_{OK}$ and $\mathbb{X}_{NOK}$ are, hence, two complementary subsets of $\mathbb{X}$, i. e., $\mathbb{X}_{OK} \cap \mathbb{X}_{NOK} = \emptyset$ and $\mathbb{X}_{OK} \cup \mathbb{X}_{NOK} = \mathbb{X}$.

Let n_{OK} and n_{NOK} be the cardinalities, respectively, of $\mathbb{X}_{OK}$ and $\mathbb{X}_{NOK}$.

The algorithm's cornerstone is to find, for each feasible sample, the box with the largest number of feasible samples and no infeasible ones, by repeatedly shrinking the input box until no more infeasible samples are present. Among the n_{OK} obtained boxes, the one with largest flexibility measure is regarded as maximum feasible box. The algorithm runs three nested loops: first, over the feasible samples in the initial set; second, over the infeasible samples in the current box; finally, over the dimensions.

In essence, the algorithm works as follows:

- for each feasible sample $\mathbf{x_{OK}}$ in $\mathbb{X}_{OK}$ –see Figure 5.11a:
  - while the set $\mathbb{X}_{NOK}^C$ of infeasible samples in the current box $\mathbf{C}$ is not empty:
    - select the infeasible sample $\mathbf{x_{NOK}}$ in $\mathbb{X}_{NOK}^C$ closest to $\mathbf{x_{OK}}$ –see Figure 5.11b;
    - for each dimension i:
      - count the number n_{OK} of feasible samples that would be retained if $\mathbf{C}$ was shrunk along i to

the position of $\mathbf{x_{NOK}}$, so that $\mathbf{x_{OK}}$ is possibly still in $\mathbf{C}$ –see Figure 5.11c;

* select i that retains most feasible samples or, in case of multiple maxima, the one that also removes most infeasible ones[15] –see Figure 5.11d;
* shrink $\mathbf{C}$ along i –see Figure 5.11e;
* update $\mathbb{X}^{C}_{NOK}$;

– get feasible box $\mathbf{B}$ by shrinking $\mathbf{C}$ to the minimum axis-aligned bounding box around $\mathbb{X}^{C}_{OK}$ –see Figure 5.11f;

– compute flexibility measure of $\mathbf{B}$;

- select box $\mathbf{B}^*$ with largest flexibility measure.

A more complete pseudo-code of the algorithm is reported in Algorithm 5.1.

Once all the infeasible samples are removed from the candidate box, this is further shrunk to the outermost feasible samples, as they represent the last available information on feasibility. Since the algorithm approximates the largest feasible box as that with the most feasible samples, it is crucial to have a set of uniformly distributed samples.

Zimmermann and von Hoessle [141] remove at each iteration the infeasible sample $\mathbf{x_{NOK}}$ with highest objective value[16]. I opt, instead, for the closest one to the current feasible sample $\mathbf{x_{OK}}$, as it makes the algorithm faster. In fact, shrinking the box to the position of the closest infeasible sample removes, in general, many of the remainders as well. Since all the infeasible samples must be removed from the box, the more of them are removed per iteration, the faster the algorithm is. I use the Euclidean distance to measure the distance between two samples. In order to compute meaningful distances, the design space should be normalized. I initialize the input space $\mathbf{C_{init}}$ as the minimum axis-aligned bounding box around the feasible region and normalize it to an unit hyper-cube.

To sum up, Algorithm 5.1 only provides an *approximation* of the maximum feasible box. This finds two explanations:

15 As done in Graff et al. [39].

16 As objective function, the feasibility function ψ could be used.

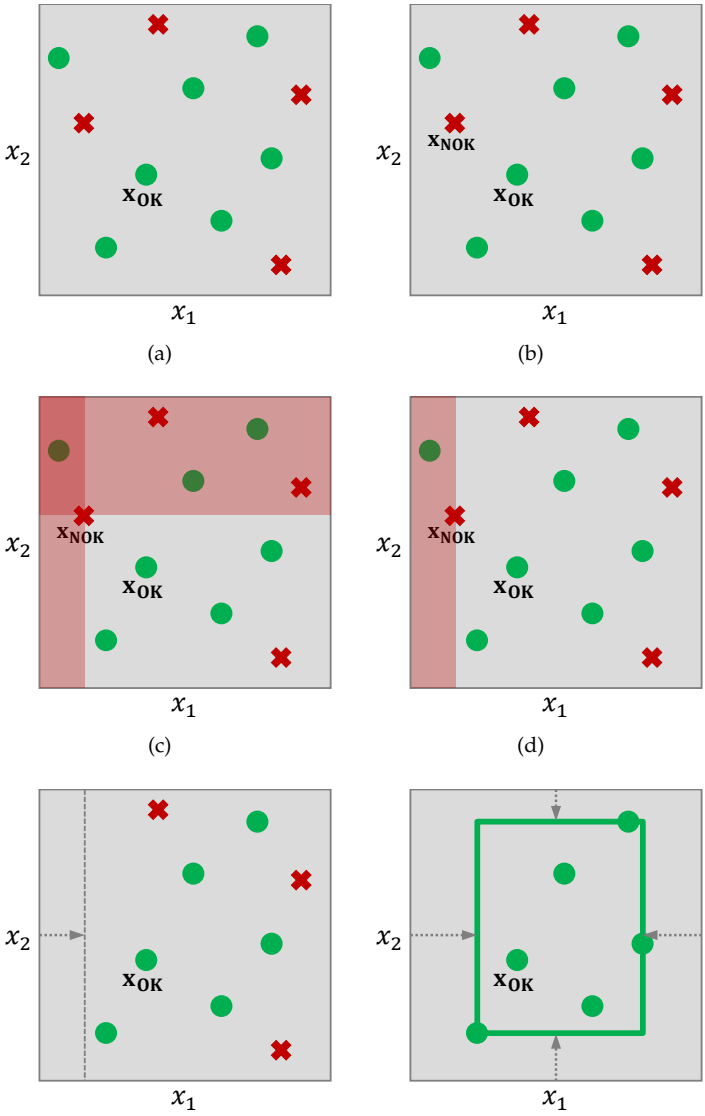


Figure 5.11: Graphical steps of the stochastic algorithm for maximum feasible box, according to Equation 5.32. Based on [141].

Input: Set of feasible samples $\mathbb{X}_{OK}$, set of infeasible samples $\mathbb{X}_{NOK}$, dimensionality d.
Output: Feasible box with largest flexibility, $\mathbf{B}^*$.

$\mathbf{C_{init}} :=$ minimum bounding box around $\mathbb{X}_{OK}$;
$F_{B^*} := 0$;
for $\mathbf{x_{OK}}$ in $\mathbb{X}_{OK}$**:**
  $\mathbf{C} := \mathbf{C_{init}}$;
  while $n^C_{NOK} \neq 0$**:**
    select $\mathbf{x_{NOK}}$ in $\mathbb{X}^C_{NOK}$ closest to $\mathbf{x_{OK}}$;
    $n^{T^*}_{OK} := 0$;
    $n^{T^*}_{NOK} := n^C_{NOK}$;
    for i in $\{1, \ldots, d\}$**:**
      $\mathbf{T} := \mathbf{C}$;
      if $x_{NOK_i} \geq x_{OK_i}$**:**
        $x^T_{i_U} := x_{NOK_i}$;
      else:
        $x^T_{i_L} := x_{NOK_i}$;
      $n^T_{OK} :=$ number of feasible samples in $\mathbf{T}$;
      $n^T_{NOK} :=$ number of infeasible samples in $\mathbf{T}$;
      if $n^T_{OK} > n^{T^*}_{OK}$ **or** $(n^T_{OK} = n^{T^*}_{OK}$ **and** $n^T_{NOK} < n^{T^*}_{NOK})$**:**
        $\mathbf{T}^* := \mathbf{T}$;
        $n^{T^*}_{OK} := n^T_{OK}$;
        $n^{T^*}_{NOK} := n^T_{NOK}$;
    $\mathbf{C} := \mathbf{T}^*$;
  $\mathbf{B} :=$ minimum bounding box around $\mathbb{X}^C_{OK}$;
  $F_B :=$ flexibility measure of $\mathbf{B}$;
  if $F_B > F_{B^*}$**:**
    $\mathbf{B}^* := \mathbf{B}$;
    $F_{B^*} := F_B$;

Algorithm 5.1: Pseudocode of a stochastic algorithm for maximum feasible box, according to Equation 5.32. Based on [141].

SAMPLING Regions of the design space without samples remain unexplored. Here no information on feasibility is available. These regions are included in or excluded from the feasible boxes so approximately, depending on the available neighboring samples. This error can only be mitigated by increasing the sampling size. The need to collect a large amount of samples in a limited amount of time explains why I employ a surrogate model in the proposed method. In high-dimensional spaces, though, extensive sampling remains not practical.

ALGORITHM The algorithm may fail to find the most feasible box for the given set of samples. The selection of the infeasible sample to remove at each iteration affects the algorithm speed, but not its accuracy. The selection of the dimension along which the box is shrunk at each iteration does affect, instead, the search for the optimum. In fact, the algorithm uses a heuristic to approximate the best decision to take at each iteration. Nevertheless, the heuristic may sometimes be wrong. This error can be mitigated by allowing the algorithm to *correct* the decision taken at each iteration.

Noteworthy, Zimmermann and von Hoessle [141] use Algorithm 5.1 to find the largest feasible box inside a candidate one, not directly inside the design space. To find that inside the design space, they repeatedly enlarge, sample and shrink a candidate box until it does not change significantly. At each iteration, they generate a set of 100 samples, which, already starting from few dimensions, leaves lots of regions unexplored. For this reason, they require a large number of iterations to precisely identify the box's boundaries. Furthermore, this optimization strategy does not allow to look for different boxes. Therefore, I use a *one-shot* approach, where the minimum bounding box is sampled only once, yet with a relatively large amount of samples, and a revised version of Algorithm 5.1 seeks the boxes with highest flexibility.

5.2.4 *Depth-Bounded Discrepancy Search*

Taking a decision means choosing one of several options. These can be considered *branches* starting from the same *node*. If decisions are taken more than once consecutively –as it happens, for instance, in an iterative loop– a tree structure is obtained. This is composed of nodes, branches and *leaves*, i. e., the termination points of any possible path starting from the root node.

Algorithm 5.1, at each iteration, decides the dimension along which the box is shrunk so as to remove a chosen infeasible sample and to retain as many feasible samples as possible. The decision is based on a heuristic: the number of retained feasible samples. There are as many options as dimensions, yet only one is explored. This is repeated for each infeasible sample still present in the box, giving rise to a tree structure. One tree is obtained for each feasible sample. Ultimately, the number of iterations, also called *depth levels*, of the tree are related to the steps necessary to remove all infeasible samples, while the number of trees to the amount of feasible samples.

When a heuristic is available, it is convention to rank the branches in descending heuristic order from left to right. Accordingly, Algorithm 5.1 always chooses the left-most branch, and so explores only the *left-most path*, as displayed in Figure 5.13a and Figure 5.14a. Several other paths exist as well. A precise estimation of their number is difficult, as the tree is rather unbalanced. In fact, it is reasonable to expect some paths to reach a leaf sooner –i. e., with less iterations– than others, depending on how many infeasible samples are removed at each iteration. Nonetheless, a balanced tree with branching factor b and depth d has b^d leaves. Therefore, to get a sense of the problem size, assuming a rather small, 5-dimensional problem and an average of 8 iterations, the tree would have $5^8 = 390{,}625$ leaves, and so paths.[17]

In order to increase the chances of finding the largest feasible box containing a particular feasible sample, the algorithm should explore more than just one path. In turn, this would increase the chances of finding the box with highest flexibility –or the p

17 There is only one path between the root node and each leaf.

ones– for the given set of samples. Obviously, the more paths the algorithm explores, the longer it takes. Speed and accuracy often do not like each other and a tradeoff is legitimate.

There are two fundamental ways to traverse a tree and several variations of them [19]:

DEPTH-FIRST SEARCH explores all the depth levels of a branch before moving to the next branch,

BREADTH-FIRST SEARCH explores all the branches of a depth level before moving to the next level.

Both of them visit each node only once. Breadth-First Search (BFS) requires an amount of space exponentially growing with the depth levels, while Depth-First Search (DFS) only linearly [72]. BFS is not suitable to applications, such as the one in this thesis, where a leaf must be reached in order to get a solution. In fact, it may take a vast amount of time before all the branches of every depth level are explored and even a single solution is reached. A DFS approach, instead, attains a promising solution fast. In fact, it starts from the left-most path, explored in Algorithm 5.1, and reaches this solution with the same speed. Any further path exploration implies further possibilities to find a better solution. By exploring all possibilities, the best solution is guaranteed. Nevertheless, a complete tree-traversal may be impractical for problems with high branching factor or high number of depth levels –or both.

Pruning can be a powerful technique to reduce the complexity of the tree. Shrinking the box along different dimensions may retain the same set of feasible samples. This means that those branches hold the same information, and so will lead to the same solutions. Therefore, only one of those branches can be further explored and the remaining ones pruned.

Furthermore, since a promising solution is obtained fast, some branches may be pruned if the box they will lead to cannot be larger than the best one found so far. Nevertheless, pruning is most effective when it is possible to stop branches early in the tree, namely in the first depth levels; otherwise, the tree will *explode* anyway. If the branch can only be pruned in the second-to-last depth level, only few paths can be saved. Additionally,

computational effort is spent to compute the volume and perform the check.

In case pruning is not enough to completely traverse the tree, incomplete strategies must be considered. A thorough overview of them is given in [6].

The most simple incomplete strategy is the timeout: the search is interrupted after a user-defined time and the best solution found so far returned. This is rather useful, as it sets an upper limit on the computational time, and is often combined with strategies that try to improve the order according to which paths are explored, so that to reach the optimal solution sooner. They frequently rely on iterative procedures that first expand the portion of the tree that is most likely to lead to the optimal solution and then progressively increase the complexity of the tree. Iterative searches, though, re-visit some nodes multiple times, as opposed to DFS and BFS. The underlying assumption is that the better order makes up for the inefficiency due to node re-visiting. It is clear, thus, that iterative algorithms take longer than DFS or BFS to search the complete tree. They are, indeed, not meant for this, as the denomination *incomplete* strategies suggest.

Among the basic iterative strategies:

ITERATIVE DEEPENING performs, at iteration i, DFS up to depth i –hence, it progressively increases the depth limit of the tree to search [71];

ITERATIVE BROADENING performs, at iteration i, BFS up to branch i –hence, it progressively increases the branching factor of the tree to search.

Harvey and Ginsberg [48] introduced the concept of discrepancy to guide tree-search with the Limited Discrepancy Search (LDS) algorithm. A discrepancy is a decision taken against the heuristic. Accordingly, the left-most branch has discrepancy $b = 0$, the second from left $b = 1$, the third one $b = 2$ and so on. Discrepancy-based algorithms rely on the assumption that if the heuristic fails, the number of mistakes made along the path is rather low. They try to correct those mistakes at little cost. Figure 5.12 show the discrepancy values of the nodes of a tree with branching factor

3 and depth 3. LDS explores, at iteration i, the nodes whose cumulative discrepancy, *discrep*, is lower or equal to i, $discrep \leq i$ with $i \geq 0$ [48]. Hence, the 0-th iteration follows the left-most path, the 1-th one expands both the left-most path and those where only one discrepancy takes place. It does not matter whether the mistake takes place at the top or at the bottom of the tree, LDS explores paths with the same amount of mistakes at the same iteration. On the contrary, DFS would need to explore a great portion of the tree before correcting a mistake at the top of the tree. Improved Limited Discrepancy Search (ILDS) avoids re-exploring previously explored paths, yet requires to specify an a-priori depth limit [72]. Figure 5.13 shows the first four iterations of ILDS on a tree with branching factor 3, depth 3 and depth limit 3.

It is reasonable to expect that the heuristic –i. e., the number of feasible samples retained inside the box– is more likely to make a mistake at the beginning of the search, when still several infeasible samples must be removed, rather than at the end, when only few infeasible samples are left. It would be beneficial, then, to correct an early mistake at less cost than a late one. Depth-bounded Discrepancy Search (DDS), developed by Walsh [128], does that: it iteratively increases the depth limit at which discrepancies may take place. This is obtained by combination of LDS and iterative deepening.

Let *depth* be the depth level of the current node, *discrep* the discrepancy accumulated until the current node and b the discrepancy of a branch extending from the current node. At iteration *iter*, DDS expands:

- if $depth < iter - 1$, branches with $b \leq iter - discrep$;
- if $depth = iter - 1 \wedge iter \neq discrep$, branches with $1 \leq b \leq iter - discrep$;
- else, the branch with $b = 0$, i. e., the left-most one.

Figure 5.14 shows the first four iterations of DDS on a tree with branching factor 3 and depth 3. The complexity of the search increases more gradually with DDS than with ILDS: the first iterations are faster, yet the last ones slower.

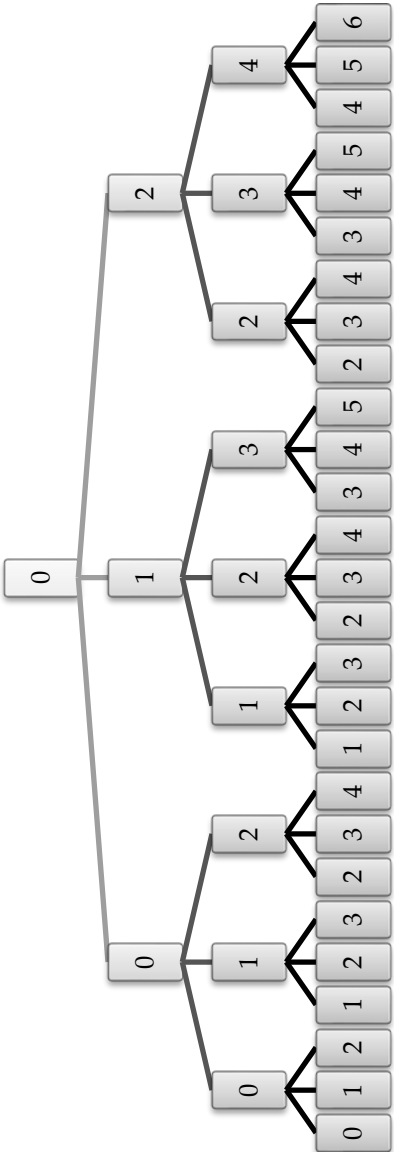
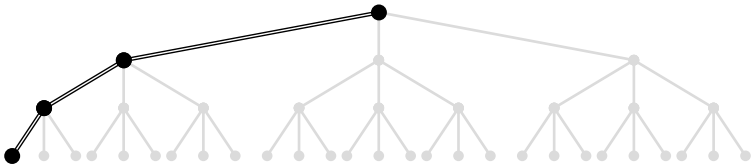
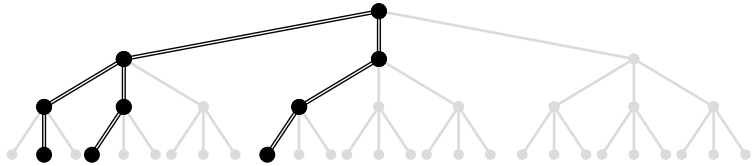


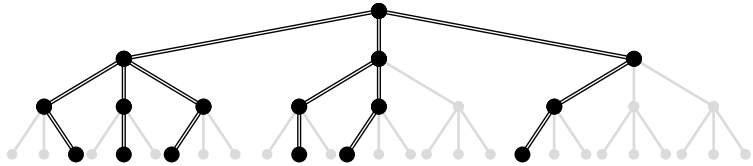
Figure 5.12: Discrepancies in a tree with branching factor 3 and depth 3.



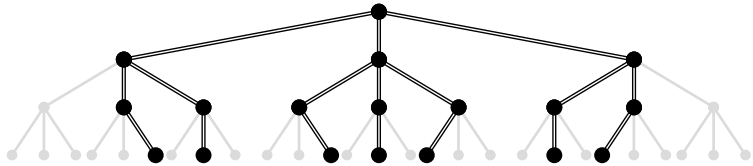
(a) $iter = 0$



(b) $iter = 1$

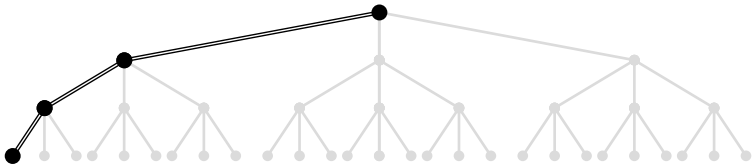


(c) $iter = 2$

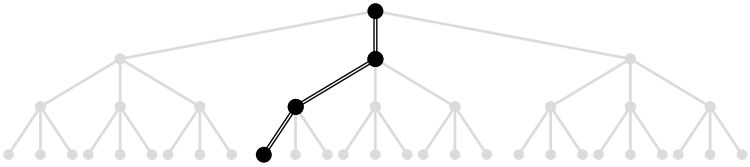


(d) $iter = 3$

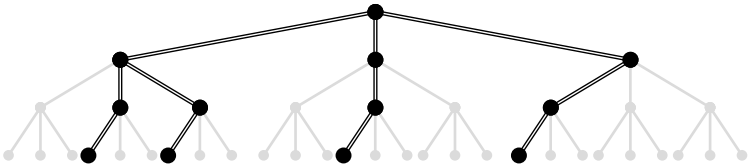
Figure 5.13: Improved Limited Discrepancy Search on a tree with branching factor 3, depth 3 and depth limit 3.



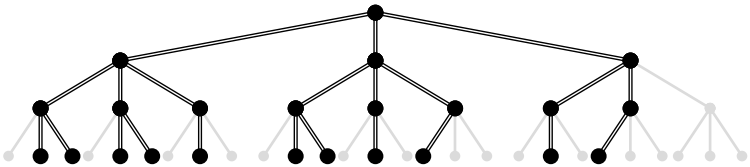
(a) $iter = 0$



(b) $iter = 1$



(c) $iter = 2$



(d) $iter = 3$

Figure 5.14: Depth-bounded Discrepancy Search on a tree with branching factor 3 and depth 3.

5.2.5 *Optimization Algorithm*

I tackle the optimization problem in Equation 5.37 with the following steps:

1. list all maximal feasible boxes,
2. sort them according to their feasibility measure,
3. select first p ones whose relative overlap is lower than or equal to q_{max}.

The second and third steps are computationally simple and do not interfere with the validity of the optimization process. The first step is, instead, from a theoretical point of view, impossible: the number of maximal feasible boxes –if the feasible region is non-point-like– is infinite. Yet, because I use a sampling-based approach to approximate feasible boxes, a maximal feasible box becomes one for which no other available sample can be added to it.[18] In other words, finding all maximal feasible boxes *for the given set of samples* means finding, for each feasible sample, the largest feasible box in which it is contained. Algorithm 5.1 tries already to find the largest feasible box for each feasible sample. In order to increase the chances of succeeding in the task, I extend its search strategy with a DDS approach.

Algorithm 5.2 reports a pseudo-code of the optimization algorithm:

`shrink_dds` is a recursive implementation of the shrinking procedure in Algorithm 5.1: at each function call, the input box is shrunk so as to remove the selected infeasible sample and retain the input feasible one. Unlike Algorithm 5.1, the dimensions along which the box is shrunk –that is, the branches to expand at each function call– are selected according to the DDS strategy. The zeroth-iteration explores, then, the left-most path, as in Algorithm 5.1. The number of feasible samples retained in the box and the number of removed infeasible ones are used as heuristics to sort the

18 Feasible samples in a box can be thought as adjacent nodes in a graph; in this way, maximal feasible boxes are the equivalent of maximal cliques.

branches. If a leaf is reached, namely if the box of a branch contains no more infeasible samples, the resulting feasible box is yielded. The search does not stop at this point, but continues until all the leaves planned for the DDS iteration are reached. Branches that contain the same set of feasible samples are pruned to avoid reaching the same feasible boxes more than once. Instead, I do not prune branches that cannot lead to a feasible box larger than the largest one obtained so far for the considered feasible sample. In fact, this check is computationally expensive, as it should be performed for each branch at each function call, and becomes effective only towards the end of tree.

`main` controls the calls to `shrink_dds`. I implement both a maximum time and iteration limit: the former because, when the number of feasible samples increases, the computational time for even few iterations may become prohibitive; the latter because, when iterations are very fast, I assume a maximum number of discrepancies to may happen along the path. The iterations could be also stopped if no larger box is found for a certain number of consecutive iterations. The largest feasible box for each feasible sample is stored in the set of maximal feasible boxes, $\mathbb{B}$. This set is then sorted on the basis of the flexibility measure and the first p boxes, whose overlap, calculated according to Equation 5.35, satisfies the constraint, are returned as output.

5.2.6 *Workflow*

The algorithm presented in the previous subsection solves, with a stochastic approach, a mathematical optimization problem. Alone, though, it does not guarantee accurate results. Its effectiveness lies in the quality of its inputs. In fact, Algorithm 5.2 is part of a larger framework that, from the results of the Bayesian feasibility analysis –described in Section 5.1– leads to design guidelines for flexibility. This framework is outlined in the following:

Function `shrink_dds`($\mathbb{X}_{OK}$, $\mathbb{X}_{NOK}$, $\mathbf{C}$, $\mathbf{x_{OK}}$, d, *iter*, *depth*, *discrep*)**:**

Input: Set of feasible samples $\mathbb{X}_{OK}$, set of infeasible samples $\mathbb{X}_{NOK}$, input box $\mathbf{C}$, feasible sample $\mathbf{x_{OK}}$, dimensionality d, iteration *iter*, depth level *depth*, accumulated discrepancy *discrep*.

Output: Any feasible box $\mathbf{B}$ found during the search.

select $\mathbf{x_{NOK}}$ in $\mathbb{X}_{NOK}$ closest to $\mathbf{x_{OK}}$;
for i in $\{1, \ldots, d\}$**:**
  $\mathbf{T} := \mathbf{C}$;
  if $x_{NOK_i} \geq x_{OK_i}$**:**
    $x^T_{i_U} := x_{NOK_i}$;
  else:
    $x^T_{i_L} := x_{NOK_i}$;
  $n^T_{OK} :=$ number of feasible samples in $\mathbf{T}$;
  $n^T_{NOK} :=$ number of infeasible samples in $\mathbf{T}$;
  if $n^T_{NOK} = 0$**:**
    $\mathbf{B} :=$ minimum bounding box around $\mathbb{X}^T_{OK}$;
    yield $\mathbf{B}$
  save $\mathbf{T}$ to $\mathbb{T}$;
remove duplicates in $\mathbb{T}$ containing same feasible samples;
sort $\mathbb{T}$ in descending order of n_{OK} and ascending order of n_{NOK};
if $depth < iter - 1$**:**
  $branches := \{0, \ldots, iter - discrep\}$;
elif $depth = iter - 1$ **and** $iter \neq discrep$**:**
  $branches := \{1, \ldots, iter - discrep\}$;
else:
  $branches := \{0\}$;
for b in $branches$**:**
  if $n^{T_b}_{NOK} > 0$**:**
    `shrink_dds`($\mathbb{X}^{T_b}_{OK}$, $\mathbb{X}^{T_b}_{NOK}$, $\mathbf{T_b}$, $\mathbf{x_{OK}}$, d, *iter*, $depth + 1$, $discrep + b$);

Algorithm 5.2: Pseudocode of a stochastic algorithm for optimization of flexibility boxes, according to Equation 5.37 – Part 1.

Function `main`($\mathbb{X}_{OK}$, $\mathbb{X}_{NOK}$, d, p, q_{max}, t_{max}, i_{max})**:**

Input: Set of feasible samples $\mathbb{X}_{OK}$, set of infeasible samples $\mathbb{X}_{NOK}$, dimensionality d, number of output boxes p, maximum allowed overlap q_{max}, time limit t_{max} and iteration limit i_{max}.

Output: Set of p feasible boxes with highest flexibility, $\mathbb{B}^*$, whose relative overlap $\leq q_{max}$.

$\mathbf{C_{init}}$:= minimum bounding box around $\mathbb{X}_{OK}$;
$time := 0$;
$iter := 0$;
while $time \leq t_{max}$ **and** $iter \leq i_{max}$**:**
  for $\mathbf{x_{OK}}$ in $\mathbb{X}_{OK}$**:**
    $\mathbf{B}_{\mathbf{x_{OK}}}$:= `shrink_dds`($\mathbb{X}_{OK}$, $\mathbb{X}_{NOK}$, $\mathbf{C_{init}}$, $\mathbf{x_{OK}}$, d, $iter$, 0, 0);
    compute volume of $\mathbf{B}_{\mathbf{x_{OK}}}$;
    if $\mathbf{B}_{\mathbf{x_{OK}}}$ *is larger than* $\mathbf{B}^*_{\mathbf{x_{OK}}}$**:**
      $\mathbf{B}^*_{\mathbf{x_{OK}}} := \mathbf{B}_{\mathbf{x_{OK}}}$;
      update $\mathbf{B}^*_{\mathbf{x_{OK}}}$ in $\mathbb{B}$;
  update $time$;
  $iter := iter + 1$;
compute flexibility measure of boxes in $\mathbb{B}$;
sort $\mathbb{B}$ in descending order of flexibility measure;
$\mathbb{B}^*$:= first p boxes in $\mathbb{B}$ whose relative overlap $\leq q_{max}$;
return $\mathbb{B}^*$

Algorithm 5.2: Pseudocode of a stochastic algorithm for optimization of flexibility boxes, according to Equation 5.37 – Part 2.

1. INPUT: Import design space, trained surrogate models, training data.

   The design space is the same used as input for the Bayesian feasibility analysis procedure in Section 5.1.5, while the trained surrogate models and training data are the outputs of that procedure. Accordingly, the surrogate models approximate the normalized constraints that delimit the feasible region.

2. PROCESS: Generate uniform sampling inside design space.

   I use the standard LHS, developed by McKay et al. [85] and implemented in the `python` library `diversipy` [134]. I use this version over the improved one, used for training, due to faster run time: linear versus quadratic with the number of samples. Thanks to the surrogate model computational efficiency, I can define a sampling budget significantly larger than the one used for Bayesian feasibility analysis: 10 times the number of training samples, namely $n = 10 \cdot (5d)^2$.

3. PROCESS: Predict samples' feasibility function with surrogate models.

   In case of multiple surrogate models, the feasibility function is computed as the maximum of the predictions of each surrogate model, according to Equation 5.2.

4. PROCESS: Compute minimum bounding box around feasible region.

   The area of the design space of interest is the feasible region. Therefore, I zoom into this by extracting the minimum axis-aligned bounding box around it and concentrating here the samples.

5. PROCESS: Generate uniform sampling inside bounding box.

   I use the standard LHS, as in step 2. The initial sampling budget in the bounding box is $n = 7 \cdot (5d)^2$. Sampling uniformity is crucial, as it is exploited by the heuristics used in Algorithm 5.2.

6. PROCESS: Repeat step 3.

7. DECISION: Check number of feasible samples.

   Sampling of the feasible region should be as dense as possible to compute accurate flexibility boxes. Otherwise, these may include unsampled infeasible regions or exclude unsampled feasible ones. However, as often stated in the current chapter, dense sampling in high dimensional spaces is utopia. Moreover, the larger the number of feasible samples n_{OK}, the more loops Algorithm 5.2 runs, the longer it takes. Therefore, a compromise is required.
   If $n_{OK} > 7 \cdot (5d)^2$, move on; otherwise, increase sampling budget in bounding box and repeat from step 5.

8. PROCESS: Add training samples to uniform sampling.

   The training samples carry *true* information, i. e., not affected by surrogate model error. Furthermore, those collected during the adaptive phase lie mostly on the boundary of the feasible region and help to delimit it. It is, therefore, valuable to add them to the set of predictions.[19]

9. PROCESS: Compute flexibility boxes.

   I use Algorithm 5.2 to perform the task. In general, I set a time limit of one hour for the last iteration to start.

10. PROCESS: Generate uniform sampling inside selected flexibility box.

    Out of the resulting boxes, I select one –e. g., the one with highest flexibility– and perform a validation run. Goal is to prove that the percentage of feasible samples in the box is larger than a given threshold. To do so, I must define how many samples I need to simulate, n, and the percentage of them that has to result feasible, p_{meas}. I follow the probabilistic approach proposed by Lehar and Zimmermann [78].

19 Adding feasible samples may slightly compromise sampling uniformity inside the feasible region. However, this effect is considered negligible, due to the large number of feasible samples, n_{OK}.

By means of Bayes' theorem, they demonstrate that, given an uniform set of samples of which n_{OK} are feasible and n_{NOK} are infeasible, the probability density function, f, of the percentage, p, of feasible samples in the box is given by:

$$f(p \mid n_{OK}, n_{NOK}) = \frac{p^{n_{OK}} (1-p)^{n_{NOK}}}{\int_0^1 s^{n_{OK}} (1-s)^{n_{NOK}} \, ds} \tag{5.40}$$

which corresponds to the probability density function of a beta distribution of the first kind, whose parameters are $\alpha = n_{OK} + 1$ and $\beta = n_{NOK} + 1$ [57]. The cumulative distribution, F, for values of p larger than the threshold p_{low} is computed as:

$$F(p \geq p_{low}) = \int_{p_{low}}^1 f \, dp = \frac{\int_{p_{low}}^1 p^{n_{OK}} (1-p)^{n_{NOK}} \, dp}{\int_0^1 s^{n_{OK}} (1-s)^{n_{NOK}} \, ds} \tag{5.41}$$

and represents the confidence level in the statement that the percentage of feasible samples in the box is larger than p_{low}. p can also be interpreted as a *probability of feasibility* inside the box. Noteworthy, as long as the samples are randomly drawn from an uniform distribution, Equation 5.40 and Equation 5.41 do not depend on the problem dimensionality. n_{OK} and n_{NOK} can be expressed in terms of the total number of samples, $n = n_{OK} + n_{NOK}$, and measured percentage of feasible samples, $p_{meas} = n_{OK}/n$:

$$F(p \geq p_{low}) = \frac{\int_{p_{low}}^1 p^{p_{meas} n} (1-p)^{(1-p_{meas})n} \, dp}{\int_0^1 s^{p_{meas} n} (1-s)^{(1-p_{meas})n} \, ds}. \tag{5.42}$$

Equation 5.42 presents four variables: the confidence level F, the percentage lower limit p_{low}, the number of samples n and the measured percentage p_{meas}. I set, first, the confidence level to 97.5%: $F(p \geq p_{low}) = 97.5\%$[20]. I select, then, the remaining quantities with the help of Figure 5.15. This shows the dependency of p_{meas} on n for different values of p_{low} at

20 p_{low} corresponds, accordingly, to the lower limit of a two-sided distribution at 95% confidence level.

a 97.5% confidence level. 12 samples –provided that they are all feasible– are enough to state that the percentage of feasible samples in the box is greater than or equal to 75%; 22 for $p \geq 85\%$ and 71 for $p \geq 95\%$. Zimmermann and von Hoessle [141] require 100 samples to be all feasible to estimate a 96% probability of feasibility inside the box. I opt for a more significant sampling, $n = 200$, and less strict percentage requirement, $p \geq 85\%$. I consider 85% enough for my purpose because:

- some error is acceptable in early phases of the development process,
- the few infeasible samples will not be, in general, far away from feasibility.

According to Figure 5.15, then, the condition that needs to be satisfied in order to validate the feasible box is $p_{meas} \geq 90\%$ –i. e., $n_{OK} \geq 180$.
To sum up, I randomly draw 200 samples from a uniform distribution.

11. PROCESS: Run simulations and collect output quantities.

    Here, *real* FE simulations, and not surrogate model's predictions, are used. I employ the LFM in Chapter 4.

12. DECISION: Check flexibility box's feasibility.

    I evaluate the percentage of feasible samples among the simulated ones, p_{meas}, to validate box's feasibility. If the condition is not satisfied, a more extensive sampling should improve the flexibility boxes' accuracy. Some error may be also related to the surrogate model inaccuracy. However, this has already been validated in Section 5.1.5 and, therefore, should not play the major role.
    If $p_{meas} \geq 90\%$ –i. e., $n_{OK} \geq 180$– move on; otherwise, increase sampling budget in bounding box and repeat from step 5.

13. OUTPUT: Store flexibility boxes.

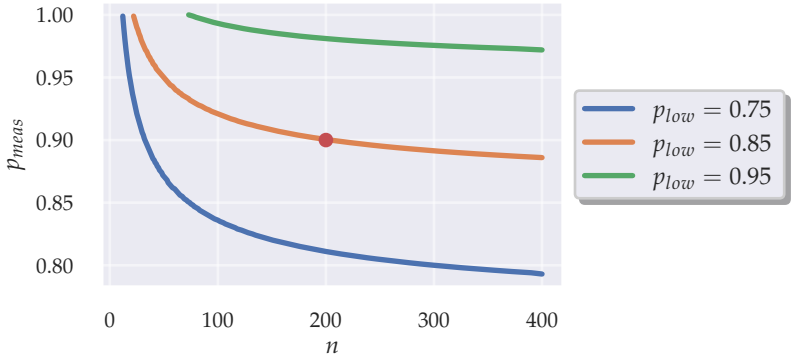


Figure 5.15: Bayesian inference applied to probability of feasibility. Depending on the measured percentage of feasible samples p_{meas} and the total number of samples n, I can predict at 97.5% confidence level different lower limit values p_{low} of the actual percentage of feasible samples, according to Equation 5.42. The red dot indicates the chosen requirement for the validation run.

Design guidelines for flexibility are ready to be delivered to concept engineers.

The described steps are condensed in Figure 5.16.

5.3 NOVELTY OF THE PROPOSED METHOD

Feasible boxes, also called solution boxes, have been the target of several previous works [29, 39, 141]; yet, to the best of my knowledge, the approach I follow differs in many ways. First off, the *workflow* is different: instead of adopting an iterative procedure that enlarges, samples and shrinks a candidate box, I first train a surrogate model for feasibility analysis, that I use to gather a relatively large number of samples over the entire design space, and then apply *once* the stochastic optimization algorithm in Algorithm 5.2 –unless validation fails.

Bayesian feasibility analysis, also called adaptive sampling for feasibility analysis, has become popular in recent times. With respect to the methods described in [10, 101, 131], the main aspects in which my implementation differs are:

MULTIPLE SURROGATE MODELS: instead of training directly one surrogate model with the feasibility function, I train several ones with as many constraints as computationally reasonable and compute the feasibility function afterwards. This is done in order to reduce the number of points of non-differentiability, where the GP approximation would be too rough. Such a simple trick can improve significantly the quality of the approximation, as it will be shown in Figure 6.13.

MULTI-POINTS EI_{feas}: instead of the classical single-point optimization algorithms, I employ a multi-modal one, NMMSO, to find the samples to add to the training set at each iteration, taking advantage of parallel computing capabilities.

VALIDATION METRICS: I separate the validation of the surrogate model training process after uniform and adaptive sampling phases by using a different metric for each, respectively R^2 and $RMSE_{0.1}$.

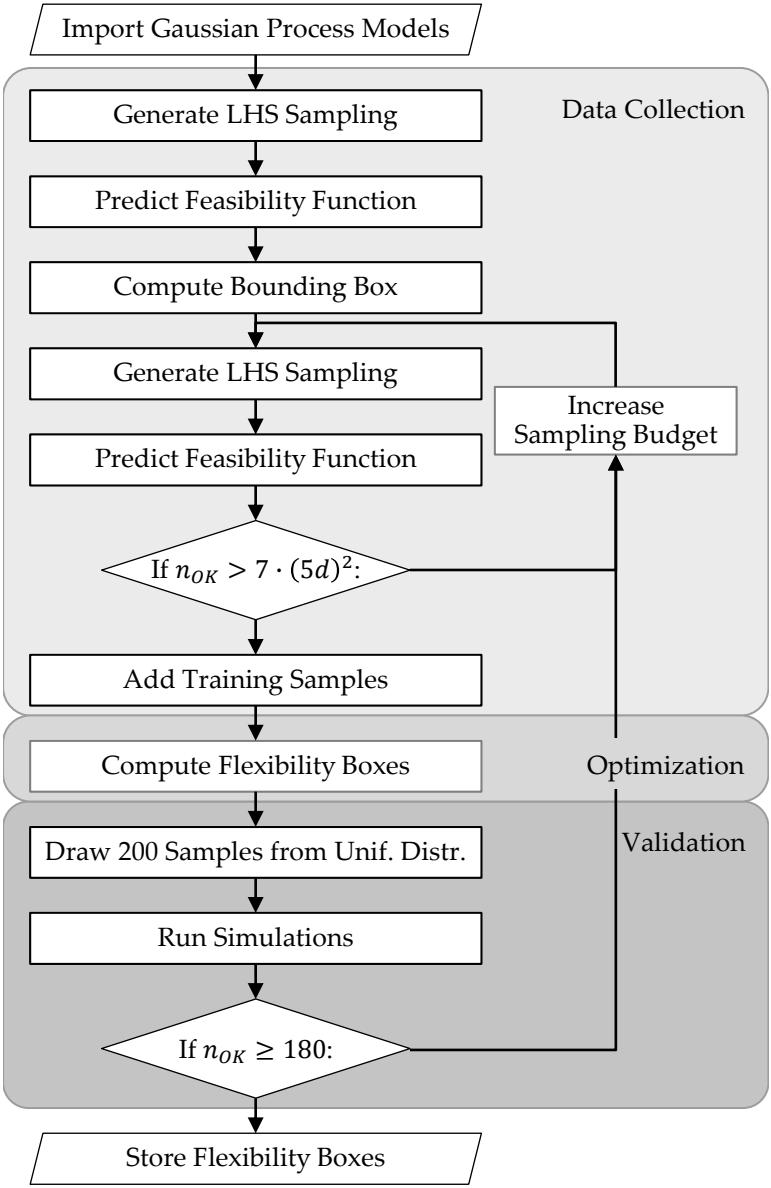


Figure 5.16: Workflow to compute flexibility boxes. A detailed description can be found in Section 5.2.6.

Novel aspects are also to be found in the flexibility optimization:

OPTIMIZATION FORMULATION: instead of only one, I look for different boxes to provide the designers with different solutions and to improve the know-how on the feasible region. The diversity among the boxes is ensured by the maximum relative overlap.

OPTIMIZATION ALGORITHM: instead of an iterative sampling approach, the algorithm uses one set of samples to find the optimal boxes. The shrinking algorithm in [141] is enhanced in Algorithm 5.2 with a DDS strategy in order to extend the search for maximal feasible boxes.

Further minor aspects are the choice of the closest sample as the one to be removed in Algorithm 5.1, which improves algorithm speed, and the introduction of a required interval size –defined by the user– which helps weighting the different parameters in the flexibility measure computation based on minimum-interval width.

6 VALIDATION

Il meglio è l'inimico del bene.[1]

— Italian proverb[2]

The goal of the current chapter is to answer the question whether the proposed methods are *good enough* for the intended use. In other words, these methods need *validation*. According to the Cambridge Dictionary, validation is the "*proof that something is correct*", namely, "*in agreement with true facts or with what is generally accepted*". Taking into account that

- the methods mainly address the early stages of the vehicle development process, where many uncertainties are present,
- several simplifications, either mechanical or mathematical, are done to improve the numerical efficiency,
- the computational budget –i. e., sample size– at disposal is limited,

I propose to accept, in general, a 10% error. Nevertheless, depending on the nature of the problem and of the used error measure, the validation condition may vary.

The first section focuses on the capability of the LFM to reproduce the impactor's kinematics of the HFM, both in reference conditions and when modifications are applied. The second one focuses on the capability to find axis-aligned boxes in the design space, where feasibility is guaranteed and flexibility optimized.

1 Literally, the best is the enemy of the good.
2 In: Voltaire. *Dictionnaire philosophique*. 1770.

6.1 LOW-FIDELITY MODEL

In Chapter 4, I describe a procedure to generate a simplified vehicle front-end model, LFM, for the pedestrian lower leg impact from a detailed one, HFM. This section focuses on the LFM requirements related to run time and accuracy.

An LFM simulation runs in approximately 20 min with 48 processors on a HPC cluster, about 17 times faster than the corresponding HFM simulation. These values are obtained by average of 8 different vehicle models. Nearly all the computational effort is associated with the impactor, being the vehicle model in the LFM reduced only to rigid bodies and few translational springs. The impactor, instead, still consists of over 250 thousand elements. A further speed up would be, therefore, obtained by simplifying the impactor as well. Nevertheless, this is outside the scope of the current work. The topic is discussed further in Chapter 8.

Central Shooting Position

First, the accuracy is evaluated in reference conditions –i. e., those used to generate the LFM– at the central shooting position, $y = 0$ mm. Figure 6.1 shows the comparison of the injury criteria curves over time for three models: the HFM in Figure 4.3a, the corresponding LFM with 80 load levels of 10 mm equal height –of which only 53 have non-zero energy absorption– and the LFM with 20 clustered load levels. In general, both simplified models are able to replicate faithfully all the reference curves. More in detail, all injury criteria of both LFMs slightly deviate from the HFM towards the end of the simulation. This is reasonable, as the error tends to accumulate during the simulation and the unloading phase of the force-deformation characteristics is usually represented less accurately than the loading one.

The femur bending moments are the first criteria to show a permanent –although slight– deviation between LFMs and HFM. This is associated with the re-loading of the first part of the bonnet, which takes place during the rotation phase of the UBM

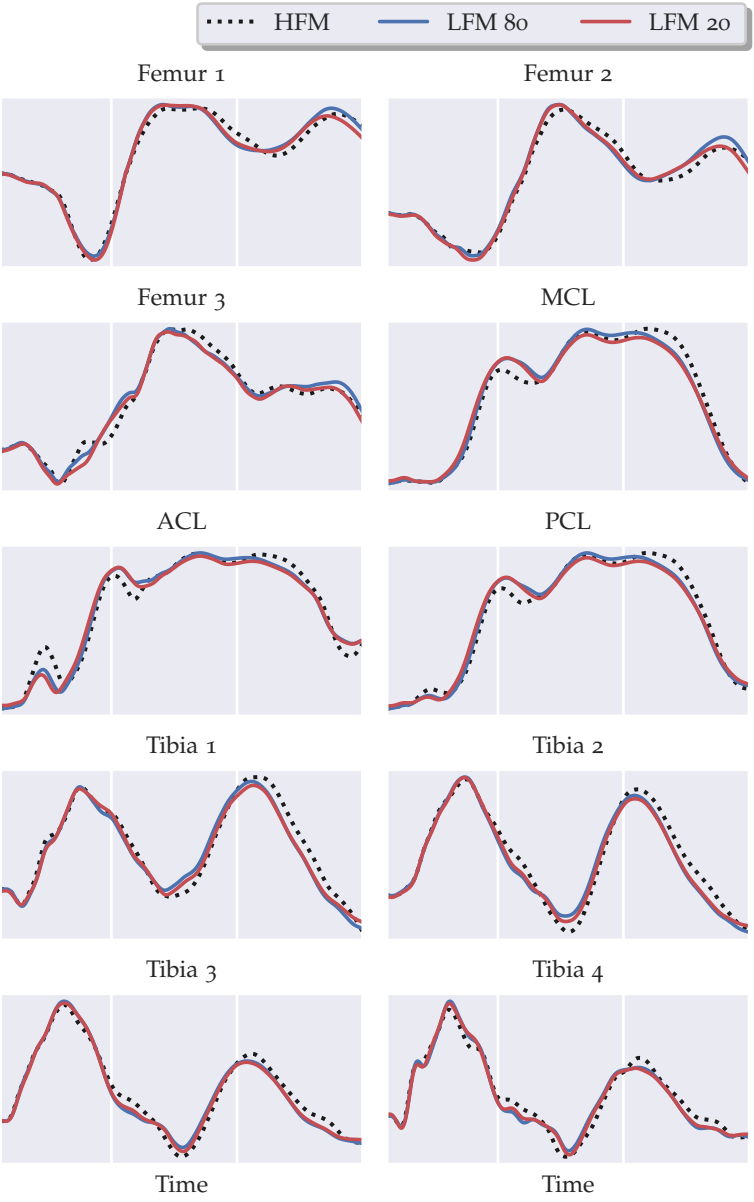


Figure 6.1: Comparison of the aPLI injury criteria among the HFM shown in Figure 4.3a, its LFM with 80 equally high load levels and its LFM with 20 clustered load levels at shooting position $y = 0\,\text{mm}$.

over the bonnet.[3] Furthermore, the femur curves show some difference between LFM 80 and LFM 20 at the very end of the simulation. This hints that the rougher discretization of the bonnet slightly affects energy absorption and deformation in this area. Nevertheless, the peak values of the curves are not affected, as the deviation starts later in time.

The bending moments measured on the tibial shaft do not show, instead, particular differences between LFM 80 and LFM 20 for most of the simulation time. Both tend to underestimate the second peak of the HFM curves, which is related to the contact of the UBM with the bonnet. The same effect is seen in the ligaments curves –namely, ACL, MCL and PCL.

The MCL and PCL curves look alike. It would not be the case with the FlexPLI and it is the direct consequence of the change in the positioning of the cruciate ligaments in the aPLI [55]. Both LFMs:

1. slightly overestimate the first local peak of the MCL curve, due to a steeper unloading of the mid-up load region,

2. reconnect to the HFM MCL curve, when –before the second peak– the UBM starts the rotation over the bonnet, thanks to a faithful rebound of the lower load region,

3. deviate a little in presence of the third peak, when the UBM hits the bonnet. This effect is slightly larger in the LFM 20.

The fact that the only meaningful deviations between LFM 80 and LFM 20 seem to come from the bonnet area suggests that the discretization obtained with clustering in the lower and mid load regions can be considered accurate.

Small oscillations in the curves, such as those appearing at the beginning of the simulation in ACL and Femur 3, are not reproduced by any LFMs. This suggests that the simplified model is capable of faithfully representing the main global trend, though *filters out* small non-linearities of the loading phase of the HFM. This effect is stronger, when cubic polynomials are used to fit the

3 The re-loading behavior of the force-deformation characteristics is explained in Section 4.1.4 and shown in Figure 4.7b.

	RELATIVE ERROR		CORA ERROR	
	80	20	80	20
TIBIA MAX	3.3%	1.6%	7.1%	5.2%
ACL	0.4%	-1.7%	7.0%	9.6%
MCL	-0.4%	-4.0%	6.8%	7.9%
PCL	0%	-3.2%	6.2%	7.5%
FEMUR MAX	4.9%	4.5%	7.1%	6.7%

Table 6.1: Relative error $E_{r_{@max}}$, as defined in Equation 4.7, and CORA error E_{CORA}, as defined in Equation 4.8, of the injury criteria for LFM 80 and LFM 20 at the shooting position $y = 0\,\text{mm}$.

original loading force-deformation characteristics. Yet, the impact on the injury criteria, overall, is small and, hence, not reported in this thesis.

The relative error $E_{r_{@max}}$, as defined in Equation 4.7, is reported for each injury criterion in Table 6.1. For sake of conciseness, I consider the maximum bending moment among the four tibia sections and that among the three femur sections. The relative errors are comparable between LFM 80 and LFM 20; only MCL and PCL suffer a significant worsening, of about 3.5%. Nevertheless, all injury criteria are remarkably precise, all below 5% relative error. Therefore, the LFM shows correct performance under the simulated loading condition.

Table 6.1 reports also the CORA error, as defined in Equation 4.8. This error is significantly higher than the relative one and lies around 7% for both LFMs. The largest difference between LFM 80 and LFM 20 is in the ACL, which worsens of 2.6%. The bending moments, instead, show a slight improvement.

Outwards Shooting Position

The previous comparison was carried out for a central shooting position. When the impactor is moved towards the side of the car, the tapering of the front-end generates a rotation of the leg

around the z-axis. A y-component is introduced in the LFM right to improve the representation of this situation. The injury criteria curves over time at the shooting position $y = 450\,\text{mm}$ are shown in Figure 6.2 for the same three models previously considered: the reference HFM, the corresponding LFM with load levels of equal 10 mm height and the LFM with 20 clustered load levels. The second model consists here of 97 load levels, of which 63 have non-zero energy absorption. LFM 97 and LFM 20 show only marginal differences between them in all injury criteria. This indicates that the clustering can be considered accurate over the entire model in this shooting position.

Both LFMs follow the HFM curves with remarkable precision. More in detail, the bending moments on the femur are more accurately reproduced at $y = 450\,\text{mm}$ than at $y = 0\,\text{mm}$, due to a smaller influence of the re-loading of the bonnet. MCL and PCL show an analogous behavior as in the central shooting position, with the overestimation of the first peak and underestimation of the third one. Differently from the $y = 0\,\text{mm}$ position, the tibia bending moments experience an overestimation of the first peak in the sections closer to the knee. This is related to a steeper unloading of the middle load region, which causes also an increase in the first MCL peak. The ACL curve shows, again, some filtering effect in both simplified models. Also for this shooting position, cubic polynomial fits enhance the filtering effect, without though major impact on the injury criteria overall.

The relative errors, as defined in Equation 4.7, are reported in Table 6.2. The ligaments show excellent accuracy, whereas the bending moments lie around 7%. Nonetheless, this is still acceptable. Also in the relative error, LFM 97 and LFM 20 show only marginal differences. This confirms the quality of the clustering procedure.

The CORA errors, reported in the same table, show, instead, larger values for the ligaments than for Tibia and Femur Max. Now the bending moments lie below 5% and the ligaments above. Since this error measure takes into account the entire curve in the calculation, this suggests that the LFMs represent very well the ligaments in the peak value, yet experience some deviations in the rest of the simulation. The bending moments, instead, suffer

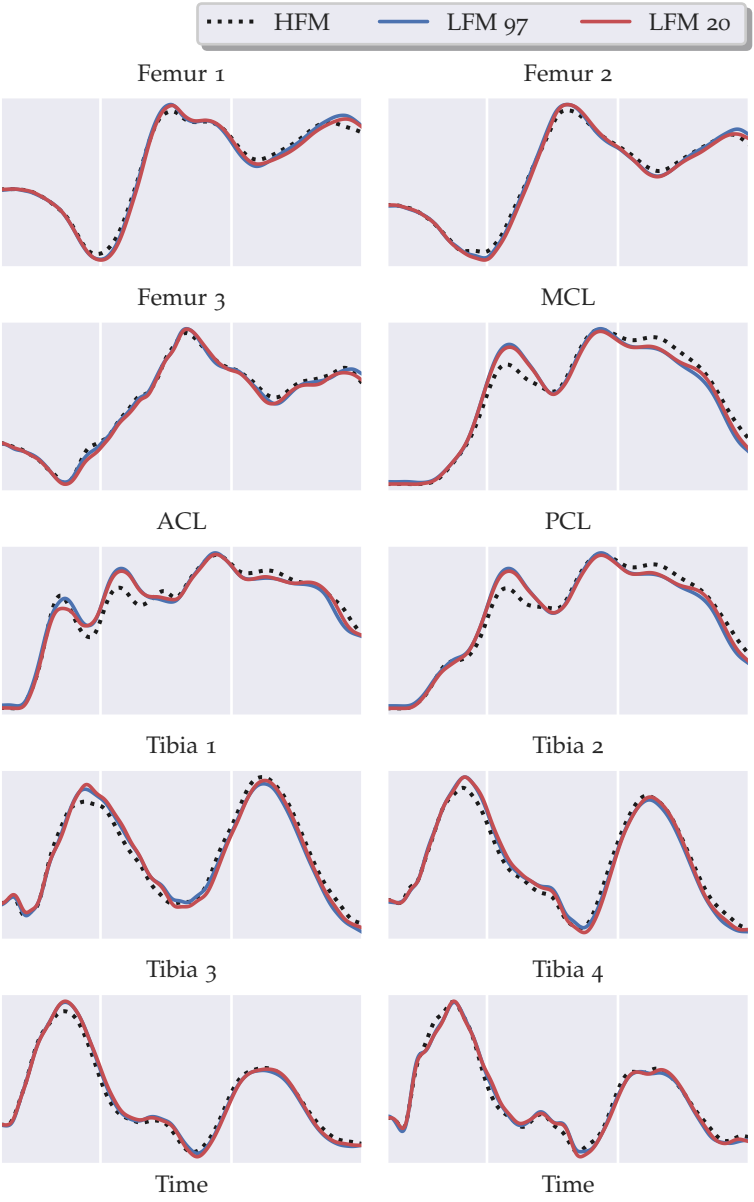


Figure 6.2: Comparison of the aPLI injury criteria among the HFM shown in Figure 4.3a, its LFM with 97 equally high load levels, shown in Figure 4.3c, and its LFM with 20 clustered load levels at shooting position $y = 450\,\text{mm}$.

	RELATIVE ERROR		CORA ERROR	
	97	20	97	20
TIBIA MAX	7.9%	8.8%	3.4%	4.1%
ACL	1.4%	0.3%	6.3%	5.9%
MCL	0.7%	-1.1%	7.3%	5.5%
PCL	1.0%	-0.5%	7.1%	5.7%
FEMUR MAX	6.1%	7.4%	3.0%	4.6%

Table 6.2: Relative error $E_{r_{@max}}$, as defined in Equation 4.7, and CORA error E_{CORA}, as defined in Equation 4.8, of the injury criteria for LFM 97 and LFM 20 at the shooting position $y = 450\,\text{mm}$.

no deviation other than the peak value, as confirmed by the curve trend in Figure 6.2.

The CORA error is certainly more suitable to objectively evaluate the LFM performance during the entire simulation. Nevertheless, as the peak values of the injury criteria are the quantities that matter in the assessment of the pedestrian leg impact, only this relative error will be reported in the following investigations.

Clustering

The comparisons at different shooting positions in Table 6.1 and Table 6.2 report, in general, no significant increase in the relative error when the number of load levels is reduced to 20. Nevertheless, they do not reveal how far clustering can go without compromising the LFM accuracy. Figure 6.3 shows the trend of the absolute value of the relative error $E_{r_{@max}}$ for different numbers of load levels, with the model in Figure 4.3b, at the shooting position $y = 0\,\text{mm}$.

Taking into account that the LFM 80 has only 53 non-zero load levels, it is of little surprise that there is minimal variation when the number of load levels is reduced to 40. Between 40 and 20 load levels, we can spot an improving trend for Tibia Max, while the other criteria slightly worsen. 20 load levels is the minimum

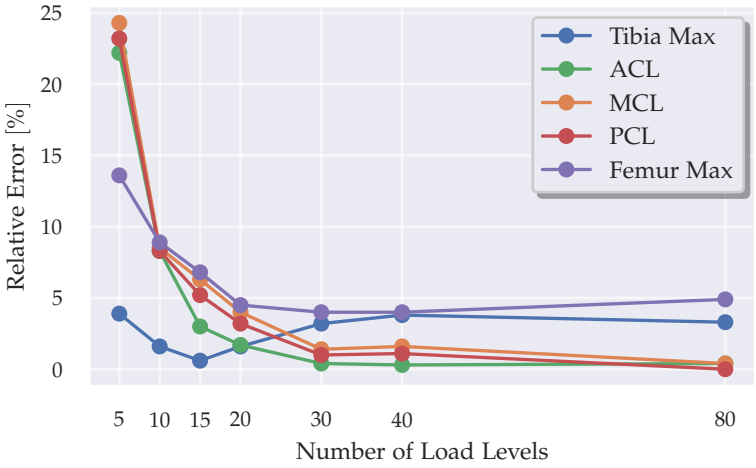


Figure 6.3: Evolution of the absolute value of the relative error $E_{r_{@max}}$, as defined in Equation 4.7, for the aPLI injury criteria for different numbers of load levels in central position. The model in Figure 4.3b is used.

number of load levels –for this loading condition– where all injury criteria stay below 5% relative error. With 15 load levels, Femur Max, MCL and PCL lie between 5 and 7% relative error, while Tibia Max decreases to 0.6%. With 10 load levels, the relative error of Tibia Max becomes negative and slightly increases in absolute value. The relative error of the ligaments, negative until here, jumps above 8% in the positive side. With 5 load levels, the ligaments rise above 20% relative error, Femur Max at 14%, while Tibia Max remains below 5%. However, the section of the tibia shaft where the maximum occurs changes, with the old one falling at about -15%. Noteworthy, with 5 load levels most of the curves score a CORA error close to 30%.

The strong inaccuracy of LFM 5 is due to the over-sized load levels, as the outer surface of LFMs is modeled as rigid body. If some levels were simply deleted –instead of added up to the clusters– results would, most likely, improve.

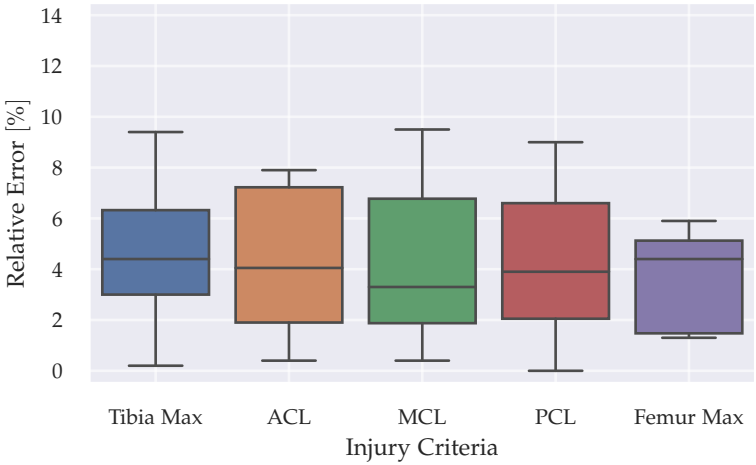
To conclude, I consider 20 a good number of load levels for the vehicle under investigation.

Vehicle Type

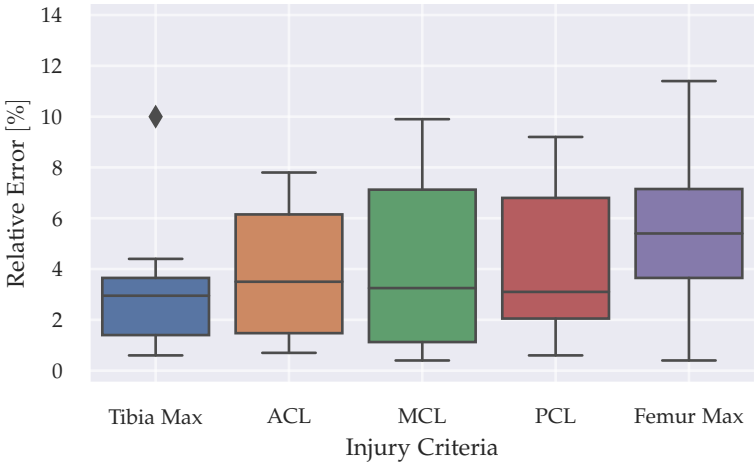
In the previous subsections, the accuracy of the LFM was assessed for one specific vehicle model, the medium-sized sedan in Figure 4.3a. Nonetheless, different vehicles deform differently. Furthermore, the impactor's kinematics varies greatly depending on the front-end outer surface. Hence, in order to analyze the LFM performance for different vehicle types, 8 models are compared. They range from a compact car to a large SUV.

Figure 6.4a shows the box plot of the absolute value of the relative error $E_{r_{@max}}$ of 8 LFMs with 10 mm high load levels for each injury criterion. The analysis is performed in central shooting position. All the medians remarkably lie around 3 to 4%. Femur Max has the smallest variation in the relative error: from 1.5% to 6%. This hints that the responses on the bonnet are well reproduced with any kinematics. Tibia Max and MCL have the largest variation, from 0% to 9%, but Tibia Max has the lowest inter-quartile range, about 3%, and MCL the lowest median, at 3%. This suggests that there exists very few vehicle models where the deformation mode of the lower or middle load region is not extremely well reproduced. PCL is strictly related to MCL, in confirmation of the results in previous subsections. In general, the lower and upper quartiles are closer for the bending moments than for the ligaments elongation. This means that several models commit a similar error in the bending moments.

In order to examine the effect of clustering for different vehicle models, the same box plot is repeated with 20 load levels in Figure 6.4b. The medians are still remarkably low: 3% relative error for the ligaments and Tibia Max and 5% for Femur Max. The range of Femur Max moves from 1 to 6% to 0 to 11%. This suggests that the rougher discretization is not beneficial for the bonnet representation; yet, the inter-quartile range is rather low, about 3%. Tibia Max's median and upper quartile decrease, as one simulation is recognized as outlier. The rougher discretization in the lower and middle region, in general, benefits the approximation, as it



(a) 10 mm equally-high load levels



(b) 20 load levels

Figure 6.4: Box plot of the absolute value of the relative error $Er_{@max}$, as defined in Equation 4.7, for the aPLI injury criteria with 8 different vehicle LFMs in central position, with: (a) 10 mm equally-high load levels and (b) 20 load levels.

tends to stabilize the deformation mode and avoid large errors. The MCL distribution looks analogous to that in Figure 6.4a, with the whisker and inter-quartile ranges slightly extending. This means that the MCL is not greatly affected by the clustering of the load levels to 20. Similar considerations apply to ACL and PCL.

To sum up, the LFM performs well for all 8 considered vehicle models: the median relative errors lie below 6% for all injury criteria and the largest error in the distribution hardly overcomes 10%. A number of 20 load levels appears to be a good choice for all models.

Geometry Variation

Recalling from Section 4.2, the usefulness of the LFM lies in its capability to be modified while preserving the validity of its output. It is, therefore, fundamental to analyze the LFM performance in out-of-reference conditions –i. e., different from the one used to generate the model. The modifications defined for the LFM affect the position and the force-deformation characteristics of the load levels. As regards the first one, load levels are rigidly shifted in the xz-plane. It is rather demanding to move the load levels in the HFM as it is done in the LFM. Morphing is a great tool. However, its capability to make changes that are comparable with the rigid translations of the LFM parametrization is questionable. Therefore, I investigate the effect of the variation in relative position *between the load levels and the impactor and not among the load levels themselves*. In other words, I do not move the levels individually, but the vehicle as a whole. This analysis is, first of all, directly comparable. Furthermore, it provides information about the behavior of the load levels with a different impactor's kinematics –thus, in out-of-reference conditions– similarly to as if I moved the load levels separately.

Figure 6.5 shows the trend of the absolute value of the relative error for each injury criterion for variations of the impactor shooting height in the interval $[-50\,\text{mm}, 50\,\text{mm}]$ in central position. The LFM is generated at the reference shooting height. It is immediately evident that the trend is not symmetrical. Raising the impactor –or lowering the car– generates more troubles than

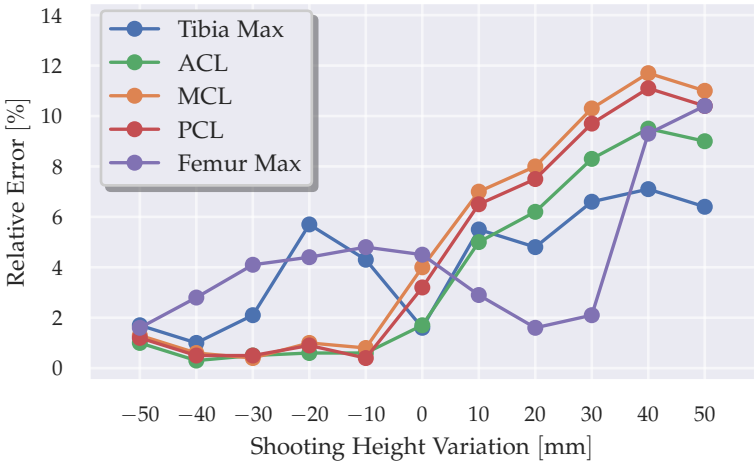


Figure 6.5: Evolution of the absolute value of the relative error $Er_{@max}$, as defined in Equation 4.7, for each injury criterion, for different aPLI shooting height variations, in central position. The model in Figure 4.3b, obtained with no shooting height variation, is used.

the opposite. The relative errors remain marginal in the negative range: below 1% for the ligaments, below 5% for Femur Max and below 6% for Tibia Max. Instead, already at 10 mm variation, the ligaments lie above 5% error; they grow steadily until 40 mm, with 10% relative error being crossed at 30 mm.

When the impactor is raised –i. e., for positive shooting height variations– the maximum elongation of each ligament is underestimated. This is caused by the inaccurate extrapolation of the force-deformation characteristics in the first portion of the bonnet. The original curve has a decreasing slope in the last quarter of the deformation, as shown in Figure 4.7b. Therefore, according to the procedure described in Section 4.1.5, the extrapolation follows a constant line starting from the point of maximum intrusion. However, this underestimates the reality. Furthermore, the MCL elongation decreases with larger intrusion in the upper load region, as it will be shown in the next chapter in Figure 7.2.

Therefore, the higher the impactor, the larger the intrusion in the first part of the bonnet, the larger the error.

The extrapolation in the first part of the bonnet is the only significant source of error for impactor positions higher than the reference one. It notably affects also Femur Max from 30 mm onward. Tibia Max is, instead, less influenced.

Noteworthy, for positive shooting height variations, the CORA errors of the ligaments are smaller than the relative ones. Starting from 40 mm, this holds also for the bending moments. In general, the CORA errors lie below 10% in the whole range of variation. This means that the LFM always replicates accurately the kinematics of the impactor, though not always the peak value of the injury criteria.

To sum up, the LFM performs well also in out-of-reference conditions, except for the extrapolation of the first part of the bonnet. Nevertheless, when the impactor is vertically shifted in the range $[-50\,\text{mm}, 30\,\text{mm}]$, the relative error lies approximately below 10%.

Stiffness Variation

The second type of modifications that are implemented in the LFM parametrization concerns the stiffness of the load levels. I define four load regions –each one consisting of multiple levels– whose force-deformation characteristics is modified. In loading, two scaling factors, respectively at the beginning and at the end of the curve, can be varied; in unloading, the dissipation factor. It is not straightforward to apply these changes to the HFM, especially in an automated way. As discussed in Section 4.2, this is outside the scope of the current work. Therefore, I do not prove whether the LFM changes can be accurately reproduced by the HFM, rather *whether HFM changes can be accurately reproduced by the LFM*.

I vary one load region at a time and compute the relative error between HFM and LFM.[4] More in detail, I scale the thickness –in

4 I do not perform a DOE, where I vary all load regions together, because it would become, then, more difficult to understand where the error comes from.

case of shells– or the density –in case of foam– of the most important structural components of each load region in the HFM in the range 0.5 to 1.5. I apply the newly recorded force-deformation characteristics of the levels in the analyzed load region to the corresponding LFM levels. I leave, though, the inclination of their springs unchanged, as in the LFM parametrization. Furthermore, the levels not present in the modified load region remain also unvaried. In this way, I define a new force-deformation curve for some load levels –as if I used some scaling factors– and test the accuracy of both their inclination and the other levels' response in out-of-reference conditions.

The error of the LFM relative to the HFM is shown in Figure 6.6 for each load region and each injury criterion, with the vehicle model in Figure 4.3a, at the central shooting position.

Starting from the *lower* load region, two large errors are experienced: in Tibia Max, about 9%, when the scaling factor is lowered to 0.75 and in the ligaments, about 10%, with 0.5. These errors, actually, come from the same source: when the stiffness is lowered, the intrusion in this region becomes larger; if the principal spring direction is not aligned with the new intrusion, the second spring in the xz-plane opposes to the movement. As a result, the LFM intrudes less, with a significant impact on the tibial bending moment and ligaments elongation. At 0.5, Tibia Max is not recorded at the section in contact with the lower load region anymore and, thus, does not suffer from this effect.

In the *middle* load region, modifications do not seem to significantly alter the LFM accuracy. This is helped by the fact that the deformation of the levels in this region is almost-horizontal. Therefore, the relative error is rather stable and does not experience large variations. This is a very good indication. In fact, the LFM is useful to investigate trends and sensitivities. If the relative error in the investigated range of stiffness remains constant, then trends and sensitivities are identical between LFM and HFM.

In the *mid-up* load region, stiffness changes significantly affect the relative error of the ligaments: the softer this load region, the larger the error and vice versa. The cause is, actually, to be found in the upper load region: the softer mid-up, the larger the intrusion in the first part of the bonnet, the larger the effects of

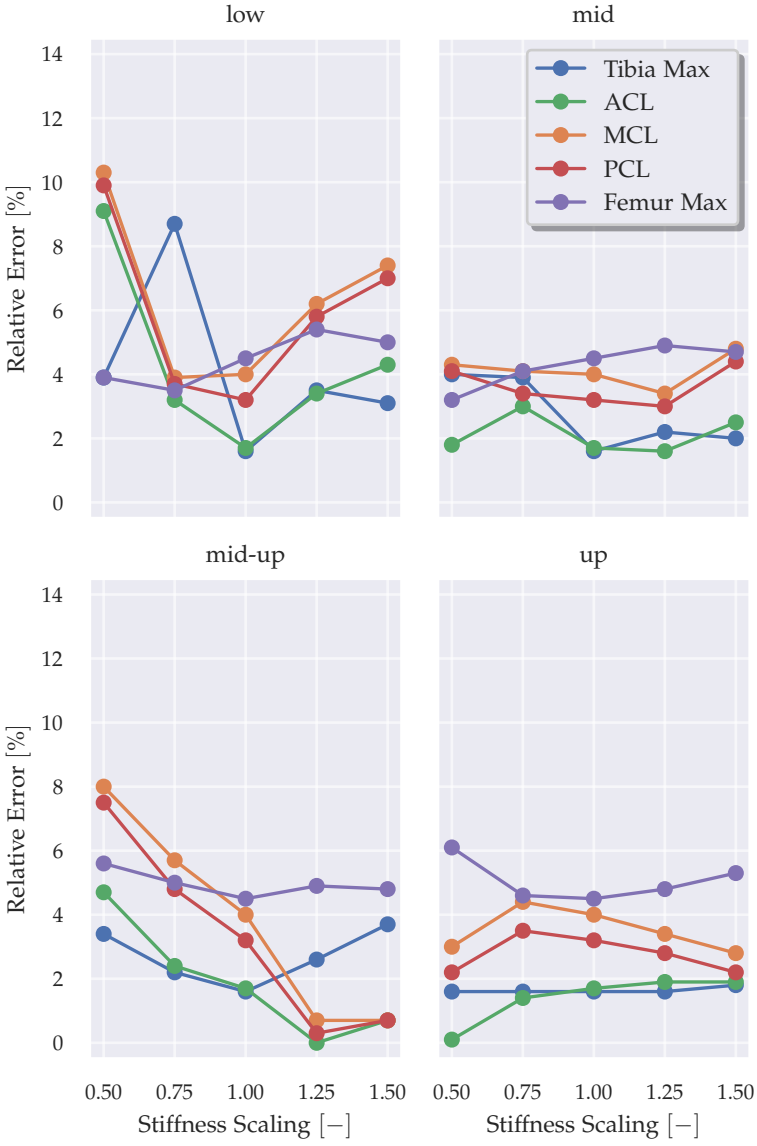


Figure 6.6: Evolution of the absolute value of the relative error $E_{r@max}$, as defined in Equation 4.7, for the aPLI injury criteria for different shooting height variations in central position. The model in Figure 4.3b, obtained at the reference shooting height, i. e., with zero variation, is used.

extrapolation and re-loading already discussed in the previous subsection –when addressing positive impactor's shooting height variations. Vice versa, the stiffer mid-up, the lower those effects.

Finally, the *upper* load region –analogously to the middle one– does not generate significant accuracy variations. Interestingly, increasing its stiffness –and thus, reducing the intrusion in this region– benefits less in terms of accuracy than increasing the stiffness of the region mid-up. The variation is, however, marginal.

To sum up, two significant errors seem to occur by lowering the stiffness of the levels:

- in the lower load region, due to the inclination of the main spring of a level in this region;[5]
- in the mid-up load region, due to the extrapolation of the first part of the bonnet –similarly to the error generated by raising the impactor.

Therefore, an intrusion larger than that in reference conditions –induced by a lower stiffness– highlights some error due to extrapolation or stiffness in non-principal directions. However, the error remains below 10%. Furthermore, for most of the stiffness changes, the error remains stable, suggesting that trends and sensitivities investigated with the LFM should be mostly accurate.

6.2 DESIGN FLEXIBILITY OPTIMIZATION

In Chapter 5, I describe a method to solve a design flexibility optimization problem for computationally expensive black-box functions. The method consists of, first, adaptively training a GP regression model for feasibility analysis and, then, using a stochastic algorithm to find axis-aligned flexibility boxes in the design space. In this section, I validate this method via mathematical functions, with a special focus on the determination of the effect of dimensionality. In order to account for the influence of random factors, I repeat the runs 10 times and provide mean and standard deviation for each computed metric.

5 In the actual LFM parametrization, when the force-deformation characteristics of a load level is scaled down, all its springs become more compliant and this effect should be less significant.

6.2.1 *Bayesian Feasibility Analysis*

Bayesian feasibility analysis comprises two phases:

1. a uniform sampling, where a global model accuracy is sought;
2. an adaptive sampling, where the model accuracy is refined around the feasible region boundary, $\psi = 0$.

The detailed procedure is described in Section 5.1.5 and summarized in Figure 5.8.

In the following, I compare three strategies that make different use of the sampling budget in Equation 5.31:

U100: the budget is entirely used for uniform sampling;

U60/A40: the budget is split in 60% uniform sampling and 40% adaptive one;

U80/A20: the budget is split in 80% uniform sampling and 20% adaptive one.

In order to evaluate the effect of the adaptive phase, the initial uniform one must be evaluated as well. Therefore, I consider also U60 and U80 in the analysis.

I validate the proposed method for Bayesian feasibility analysis with the help of three questions:

QUESTION 1. Are the samples added during the adaptive phase located close to the feasibility boundary?

QUESTION 2. How much does the adaptive phase improve the prediction accuracy of the surrogate model in feasibility analysis?

QUESTION 3. Is the surrogate model at the end of the adaptive phase accurate enough for my purposes?

The last question would be, actually, enough to validate the surrogate model for feasibility analysis. Yet, the first two help me to evaluate the added value of the Bayesian approach.

In order to quantify the accuracy of the surrogate model, I compare two approaches:

- 50,000 new test samples,
- cross-validation of the training samples.

Branin

In Section 5.1.3, a modified version of the Branin function –defined in Equation 5.20– was introduced for exemplary purposes and used as feasibility function, $\psi(x_1, x_2) = f(x_1, x_2)$. I use this 2D function to run the first validation. In accordance with Table 5.2, I set the budget to 100 samples. The Branin function is used for validation of feasibility analysis also in [130, 131].

QUESTION 1. Figure 6.7 shows the location of the training samples over the design space every five iterations of the adaptive phase for the U60/A40 strategy. Two samples are added at each iteration. The newly added samples lie extremely close to the feasible region boundary. This is highlighted by Figure 6.8a, which shows the PDF distribution of the feasibility function ψ of the training samples added during, respectively, uniform and adaptive phases. In order to obtain a more representative outcome, I take into account all the training samples of the 10 runs. Only a glimpse of the uniform distribution can be caught, as the spike of the adaptive one at $\psi = 0$ predominates.
It is not only important to consider the feasibility function value of the training samples. Their location matters as well. In fact, the samples added during the adaptive phase should be spatially spread around the whole feasible region boundary and not concentrated in one specific area. In 2D, it is rather easy to visualize them: from Figure 6.7, we can see that the samples completely cover the feasible region boundary. In higher dimensions, visualization is not possible. Therefore, it is convenient to analyze the Euclidean distance between each sample and its closest one. This indicates the samples' degree of spatial uniformity. Figure 6.8b shows the PDF distribution of the minimum distance for the samples added during uniform and adaptive phases. For the latter, the closest sample is searched among all training ones –i. e., uniform and adaptive. An ideal uniform sampling would present a single spike at the largest minimum distance possible –meaning that

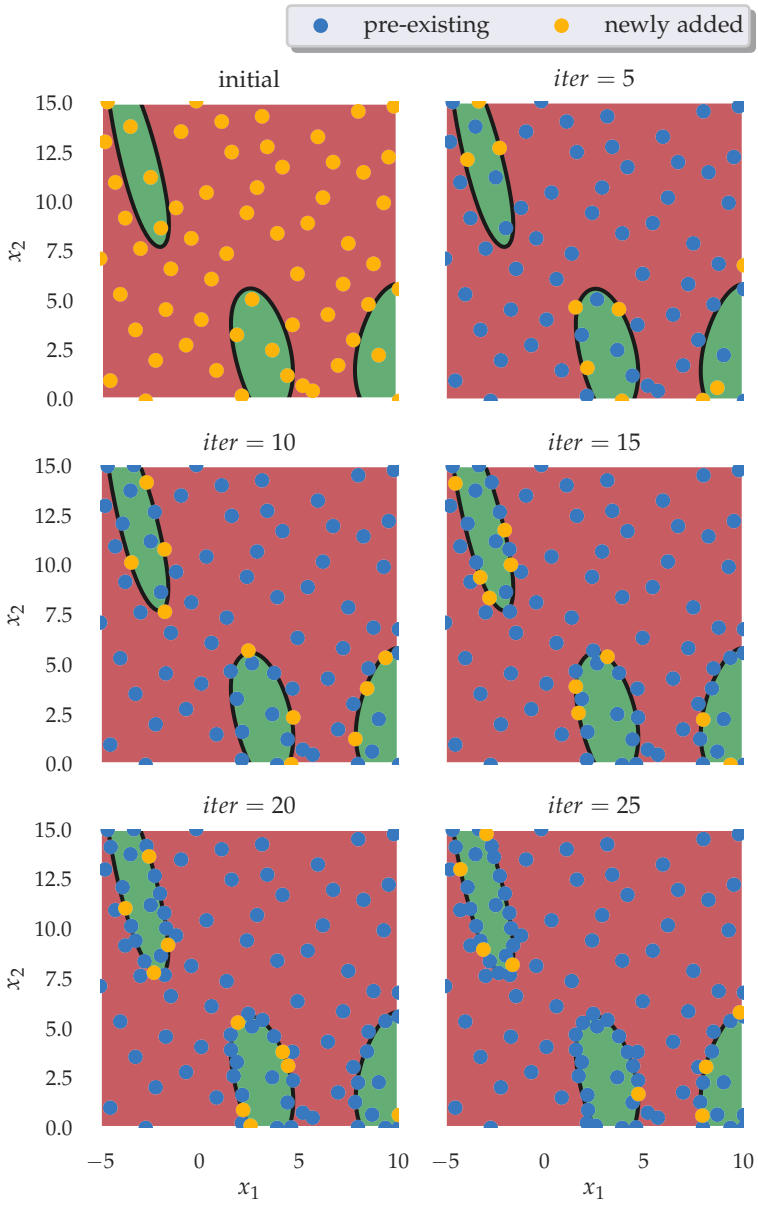
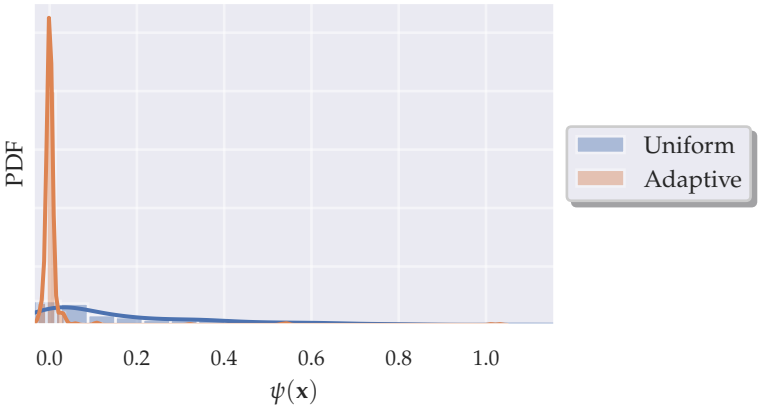
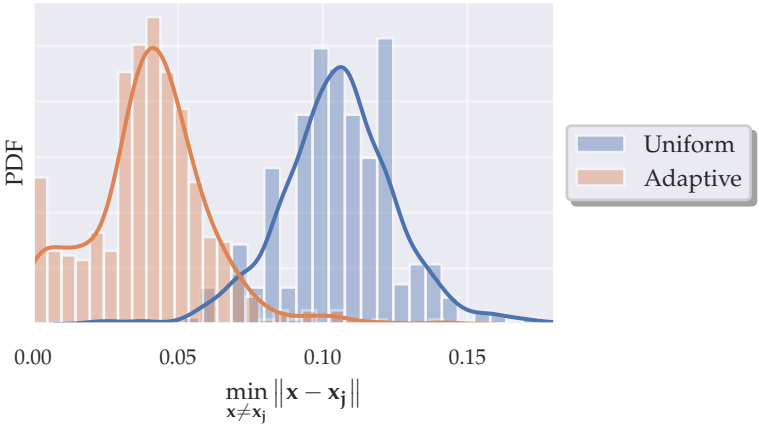


Figure 6.7: Training samples over the design space for different iterations of the adaptive phase of the U60/A40 strategy with the modified Branin function in Equation 5.20.



(a)



(b)

Figure 6.8: Probability Density Function (PDF) of:
(a) the feasibility function ψ,
(b) the distance from the closest sample
of the training samples added during, respectively, uniform and adaptive phases of the U60/A40 strategy with the modified Branin function in Equation 5.20.

all samples are at the same distance and maximally spread. The distribution obtained with an improved LHS presents, instead, some non-uniformities and extends over a range of distances. The sampling obtained during the adaptive phase is clearly denser. Its distribution moves towards lower values, yet it remains at reasonable distance. In fact, the peak lies at about half the one from the uniform sampling[6] and only few samples almost coincide.
On the basis of the information coming from both Figure 6.8a and Figure 6.8b, we can state that the adaptive phase of the U60/A40 strategy successfully explores the area immediate around the feasible region boundary. Specifically, the plots suggest that, in 2D, both the Expected Improvement for feasibility analysis, EI_{feas}, and the multi-modal optimizer, NMMSO, work properly. This leads us to the question how much additional value the adaptive phase actually brings to feasibility analysis.

QUESTION 2. The used GP is a regression model, yet feasibility analysis is a classification task. Therefore, it is reasonable to assume that most of the error in predicting feasibility lies in a range of values around $\psi = 0$ –say $\psi \in [-0.1, 0.1]$. Figure 6.9 scatters the predicted feasibility function $\hat{\psi}$ against the observed one ψ for new test data in the interval $\psi \in [-0.1, 0.1]$ with strategies U60, U100 and U60/A40. The bisector represents the ideal prediction. In blue, True Positive and True Negative predictions; in red, False Positive and False Negative ones. A great improvement is obtained increasing the number of uniform samples from U60 to U100, yet an outstanding tiny error is achieved with U60/A40. Noteworthy, the distribution of predictions for U60/A40 is significantly narrower than for U100 only in the very proximity of $\psi = 0$.
Quantitatively, I validate the adaptive phase with the metric $RMSE_{0.1}$ –i. e., the *RMSE* for observed values $\psi \in [-0.1, 0.1]$. Figure 6.10a plots mean and standard deviation of $RMSE_{0.1}$ for all the sampling strategies with new test data, while Figure 6.10b those with cross-validation data. Taking into account, first, the results with new test data, the lowest $RMSE_{0.1}$ is achieved with

6 Note that 67% of the uniform sampling budget is added during the U60/A40 adaptive phase.

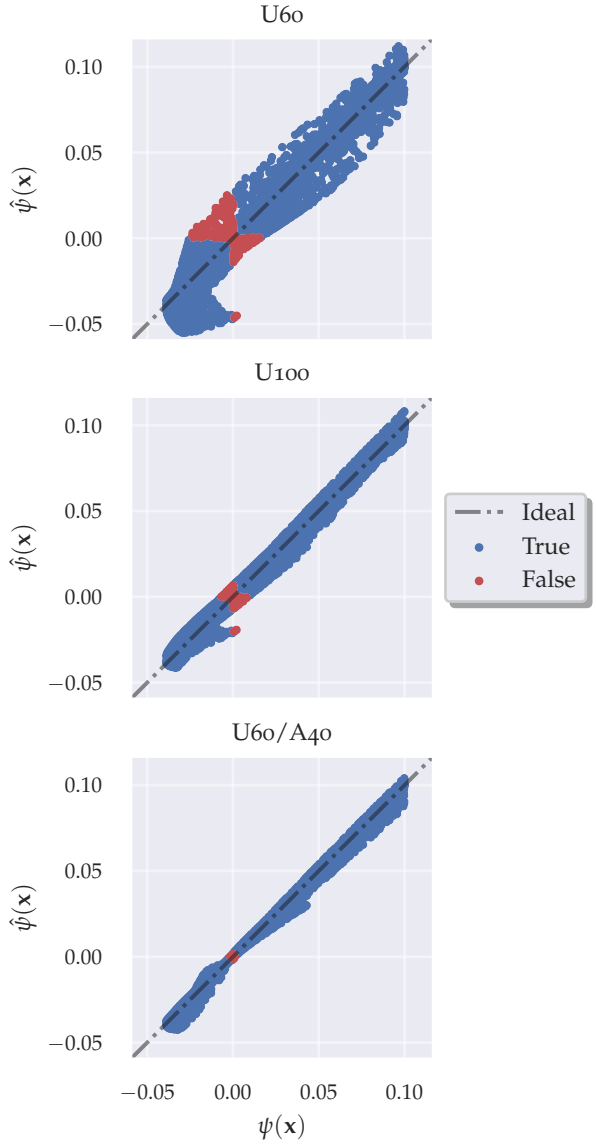


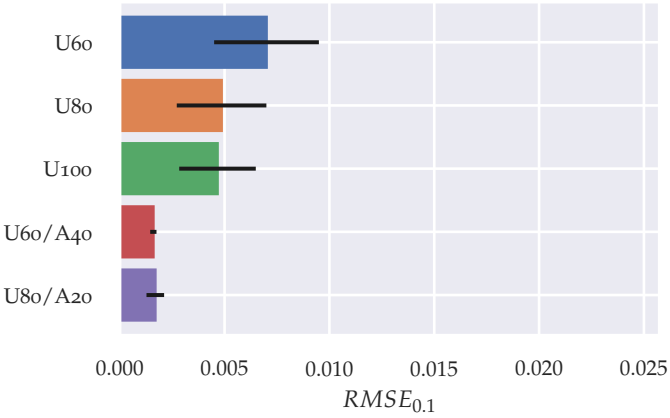
Figure 6.9: Predicted feasibility function $\hat{\psi}$ against observed one ψ for new test data in the interval $\psi \in [-0.1, 0.1]$ for U60, U100 and U60/A40 strategies with the modified Branin function in Equation 5.20.

U60/A40. While the mean is similar to that of U80/A20, the standard deviation is significantly lower, hinting that it is a more robust strategy. In general, both adaptive strategies show outstanding low standard deviation values with respect to the uniform ones. This suggests that the adaptive phase compensates for the randomness of the initial uniform sampling. During the adaptive phase of U60/A40, $RMSE_{0.1}$ mean decreases of about 75%, while its standard deviation of 95%. As U80 offers a better accuracy than U60, $RMSE_{0.1}$ mean and standard deviation decrease of about, respectively, 65% and 80% during the adaptive phase of U80/A20. Noteworthy, U100 does not seem to improve significantly the $RMSE_{0.1}$ over U80: an adaptive strategy is considerably more effective. To sum up, the two-dimensional Branin function shows that the adaptive strategies reduce the $RMSE_{0.1}$ on average of 65% with respect to a uniform strategy with the same sampling budget.
The difference in $RMSE_{0.1}$ between cross-validation and test data is remarkable, with the former being regularly more than double the latter one. This is related to two factors:

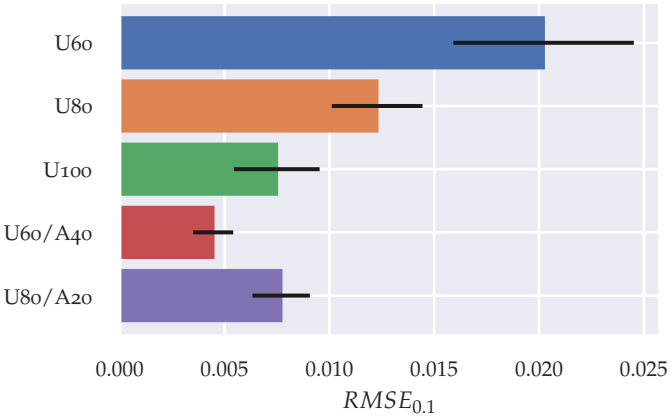
1. cross-validation repeatedly uses only a *subset* of the available samples as training samples –in this case, 90%– with the rest being used as test samples;

2. as direct consequence of the previous point, in cross-validation the test samples are located where training samples lack, thus the surrogate model is evaluated at some of the *weakest* spots.

Therefore, cross-validation generally *underestimates* the surrogate model accuracy. The smallest effect seems to affect the U100 strategy: $RMSE_{0.1}$ with cross-validation data is only 1.5 times that with new test data. This is due to the fact that, the more training samples are used, the less effect it has using only a subset of them. The strategy that is underestimated the most with cross-validation is, instead, U80/A20, with $RMSE_{0.1}$ differing of a factor four and being even higher than for U100. This may find an explanation in the purpose of the adaptive phase: to add samples that improve the accuracy of the GP around the feasible region



(a) New test data



(b) Cross-validation data

Figure 6.10: Bar plot of mean and standard deviation of *RMSE* for $\psi \in [-0.1, 0.1]$, for different sampling strategies, with the modified Branin function in Equation 5.20, with: (a) new test and (b) cross-validation data.

boundary. Using only a subset of these samples undermines its accuracy. The U60/A40 strategy, having more points around $\psi = 0$ than U80/A20, is less affected.

The *RMSE* is an absolute measure of the inaccuracy of the surrogate model. Relative metrics, such as the R^2 or the *MCC* coefficients, more intuitively express its accuracy. Nevertheless, as anticipated in Section 5.1.4, both R^2 and *MCC* do not work well with cross-validation and adaptive sampling for feasibility analysis. This is made clear by Table 6.3. Here, R^2, $R^2_{0.1}$ and $MCC_{0.1}$ are reported with both new test data and cross-validation. The metrics with subscript $_{0.1}$ are computed for $\psi \in [-0.1, 0.1]$. I do not report the global *MCC*, as I am mostly interested in the classification ability close to the feasible region boundary. R^2, instead, is used for the validation of the initial uniform phase.

Taking into account, first, the results with new test data, R^2 shows that the adaptive strategies do not improve significantly the global surrogate model accuracy –which is already remarkably high. Yet, $R^2_{0.1}$ and $MCC_{0.1}$ show that they succeed in ensuring global accuracy also on local level, both in terms of mean and standard deviation. U60/A40 reaches consistently 99.8% accuracy in all metrics with almost zero standard deviation. U80/A20 performs similarly well: slightly better globally, slightly worse in feasibility analysis. The adaptive phase of U60/A40 improves $R^2_{0.1}$ of about 4% and $MCC_{0.1}$ of about 8% and zeroes both standard deviations. All the relative metrics suggest that U80 and U100 do not differ much in terms of neither global nor local accuracy.

Very different –and misleading– numbers are obtained with cross-validation for $\psi \in [-0.1, 0.1]$. As for the absolute metric $RMSE_{0.1}$, U100 is the least underestimated by cross-validation. The adaptive strategies suffer further underestimation with relative metrics[7]: besides the standard downsides of cross-validation, discussed previously, since many samples lie close to the feasibility boundary $\psi = 0$, on the one hand, the average of the observations tends to zero –influencing the $R^2_{0.1}$– on the other hand, it is more likely to wrongly predict the feasibility class of the observations –influencing the $MCC_{0.1}$. This results, e. g., in an $MCC_{0.1}$ of

7 See Section 5.1.4 for details.

U80/A20 16.5% lower than that of U100, instead of 3.4% greater. The global R^2 score, although underestimated, is the only one to remain comparable to the one with new test data and is, therefore, used for validation of the uniform phase: both U60 and U80 fulfill the condition $R^2 \geq 0.8$.

QUESTION 3. I assess the accuracy of the surrogate model in feasibility analysis with the condition $RMSE_{0.1} \leq 0.05$ on cross-validation data. The condition –independent on the dimensionality of the problem– is, in 2D, abundantly satisfied even by the U60 strategy. U60/A40 achieves a score lower than 0.002 and, thus, can be considered remarkably accurate.

To sum up, Bayesian approaches for feasibility analysis look very promising on the two-dimensional Branin function. They are more computationally intensive than uniform sampling –as they require successive iterations of GP training and acquisition function optimization– yet they provide a substantial boost in accuracy. U60/A40 performs slightly better than U80/A20. Cross-validation generally underestimates both regression and classification accuracy scores with respect to brand new test data. This is aggravated by Bayesian approaches for feasibility analysis, as samples are mostly added in critical locations. However, due to the fact that the proposed method mostly addresses problems with computationally expensive black-box functions, it is inconvenient to generate a large set of test data. Therefore, I do use cross-validation. Cross-validated $RMSE_{0.1}$ better approximates the *true* score than relative metrics.

Rosenbrock

Several indications on Bayesian feasibility analysis can be obtained with the Branin function. However, this is only defined in 2D. The effect of dimensionality on the proposed method remains an open point. A popular function in global optimization is the Rosenbrock function, first formulated in 2D by Rosenbrock [103]. Because of its multi-modal behavior and multi-dimensional generalization, it is a good choice to further test the method.

STRATEGY	R^2		$R^2_{0.1}$		$MCC_{0.1}$	
	Mean	Std	Mean	Std	Mean	Std
U60	0.997	0.002	0.959	0.032	0.917	0.034
U80	0.999	0.001	0.980	0.021	0.960	0.017
U100	0.999	0.001	0.982	0.014	0.960	0.022
U60/A40	0.998	0.001	0.998	0	0.998	0.001
U80/A20	0.999	0	0.998	0.001	0.994	0.001

(a) New test data

STRATEGY	R^2		$R^2_{0.1}$		$MCC_{0.1}$	
	Mean	Std	Mean	Std	Mean	Std
U60	0.969	0.013	0.670	0.140	0.713	0.179
U80	0.992	0.003	0.889	0.039	0.842	0.099
U100	0.998	0.001	0.956	0.022	0.917	0.054
U60/A40	0.986	0.010	0.971	0.010	0.829	0.041
U80/A20	0.994	0.003	0.943	0.023	0.752	0.088

(b) Cross-validation data

Table 6.3: Regression and classification metrics for different sampling strategies with the modified Branin function in Equation 5.20, with: (a) new test and (b) cross-validation data.
The subscript $_{0.1}$ indicates that the metric is computed for $\psi \in [-0.1, 0.1]$.

As I aim to apply Bayesian feasibility analysis on a problem with multiple constraints –i. e., pedestrian leg impact's injury criteria– I define two modified versions of the generalized Rosenbrock function as constraints:

$$
\begin{aligned}
g_1(\mathbf{x}) &= \frac{\sum_{i=1}^{d-1} \left(100 \left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right)}{12d} - 1 \\
g_2(\mathbf{x}) &= -\left(\frac{\sum_{i=1}^{d-1} \left(100 \left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right)}{6d} - 1\right) \\
\psi(\mathbf{x}) &= \max\left(g_1(\mathbf{x}), g_2(\mathbf{x})\right)
\end{aligned}
\tag{6.1}
$$

where $\mathbf{x} \coloneqq (x_1, \ldots, x_d) \in \prod_{i=1}^{d} [-0.4, 0.7]$ and d is the number of dimensions, with $d \geq 2$. With respect to the standard definition of the generalized Rosenbrock function in [56], I shift and normalize g_1 and g_2 so as to, first, ensure that both constraints are active and, then, create feasible regions of reasonable size over all dimensions under investigation –namely 2, 5, 8, 12, 15.

Figure 6.11 shows the contour plot of the feasibility function ψ for the 2D variant. Here, the combination of two constraints generates two separate feasible valleys, which cover approximately 22% of the design space. Table 6.4 reports the approximate size of the feasible region with respect to the one of the design space for all investigated dimensions. In high-dimensions, the portion of feasible region in the design space gets larger and reaches about half the total volume.

As done for the Branin function, I validate the Rosenbrock-based problem by answering the three questions posed at the beginning of the section. The sampling budget used in each dimension is reported in Table 5.2.

QUESTION 1. Figure 6.12 shows the feasibility function and minimum distance PDF distributions for the training samples added during uniform and adaptive phases of the U60/A40 strategy for all investigated dimensions.
In 2D, the distributions look very similar to those obtained with the Branin function: most of the samples added during the adaptive phase are located in the very close proximity of $\psi = 0$; they

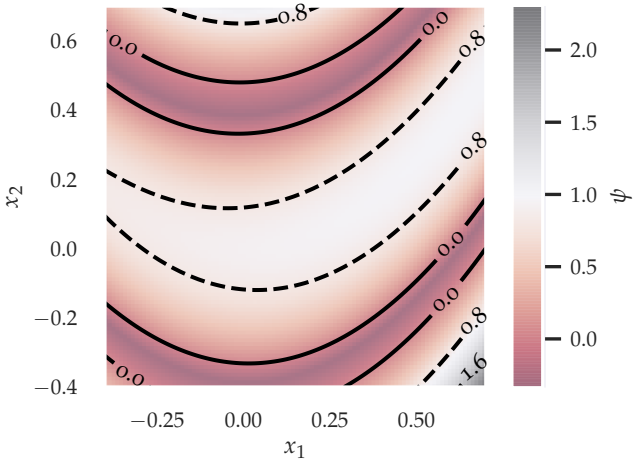


Figure 6.11: Contour plot of the 2D variant of the Rosenbrock-based feasibility function defined in Equation 6.1.

DIMENSION	FEASIBLE REGION VOLUME
2	22%
5	47%
8	53%
12	54%
15	53%

Table 6.4: Approximate size of the feasible region with respect to that of the design space for different dimensions according to the Rosenbrock-based feasibility function defined in Equation 6.1.

increase the density of samples inside the design space, yet remain at a reasonable distance.
As the problem dimensionality increases, the curves of both feasibility function and minimum distance flatten. Following considerations apply:

FEASIBILITY FUNCTION: the larger the problem dimensionality, the more spread the adaptive samples are around the feasible region boundary. This is due to the fact that, on the one hand, the GP prediction becomes less accurate, since the initial training samples become sparser –as discussed below. On the other hand, the multi-modal NMMSO algorithm may become less effective in the optimization of the expected improvement. Noteworthy, a strong worsening in sharpness of the adaptive sampling distribution seems to happen between 2D and 5D and between 8D and 12D.

MINIMUM DISTANCE: the larger the problem dimensionality, the larger the minimum distances –i. e., the sparser the points– and the wider the *range* of minimum distances -–i. e., the less uniform the degree of sparseness inside the design space. This is especially relevant for the uniform sampling distribution, whose sharpness in 2D is rapidly lost increasing the number of dimensions.[8] This suggests that the performance of the improved LHS, used to generate the uniform sampling, becomes less effective in high-dimensions –i. e., the samples become less uniformly distributed. The minimum distance of the adaptive samples becomes also more spread, yet moves towards larger distances, comparable to those obtained from the uniform sampling. This indicates that the samples added during the adaptive phase always maintain an *exploratory* characteristic.

QUESTION 2. Figure 6.13a plots mean and standard deviation of $RMSE_{0.1}$ for all investigated sampling strategies and dimensions with 50,000 new test samples.

8 NB: the most uniform sampling distribution would have a single spike at the maximum possible minimum distance.

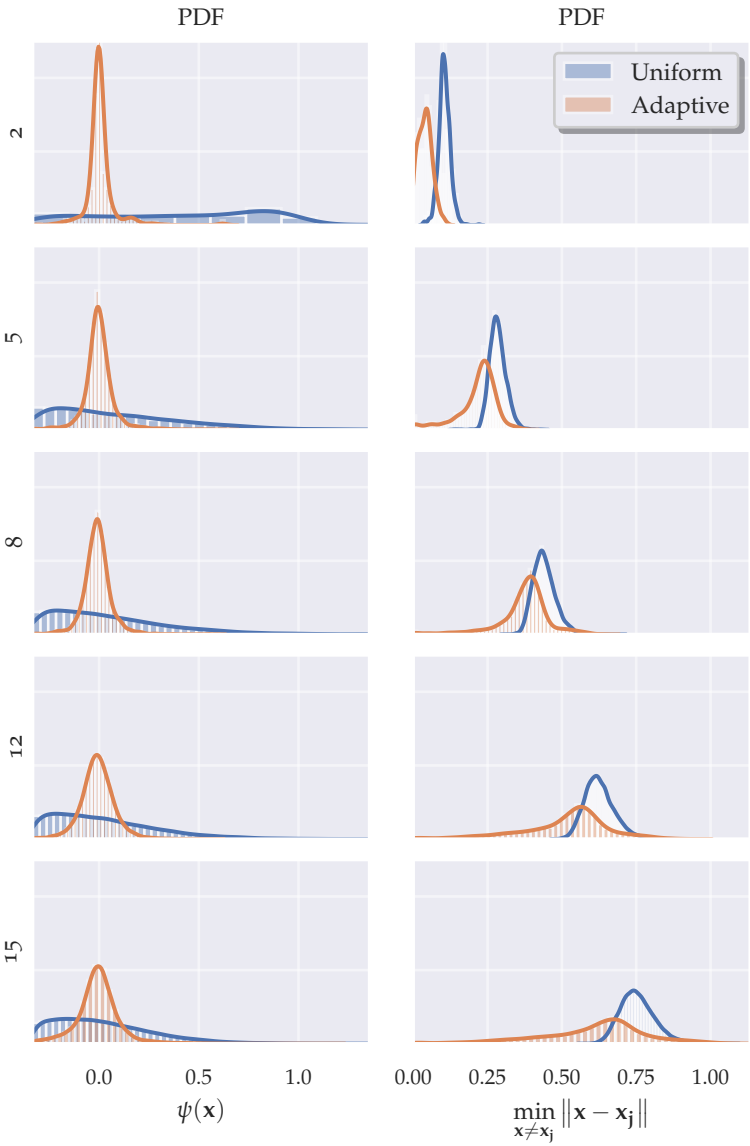


Figure 6.12: Probability Density Function (PDF) of the feasibility function ψ and of the minimum distance for each training sample added during uniform and adaptive phases of the U60/A40 strategy, for different dimensions, with the Rosenbrock-based problem in Equation 6.1.
Both x- and y-axis are shared column-wise.

In 2D, the relative performance of the different sampling strategies is similar to that in the Branin case. From an absolute perspective, though, mean and standard deviation values are larger here –circa double. This suggests that the Rosenbrock-based problem –with the combination of two constraints– is more challenging than the Branin function. U60/A40 is the best performing strategy and reduces the mean error of: about 80% with respect to U60 –standard deviation -65%– 55% with respect to U100 –standard deviation -50%– and 25% with respect to U80/A20 –standard deviation -20%. The standard deviation values with adaptive strategies are significantly higher than with the Branin function. This indicates that, for the Rosenbrock-based problem, the adaptive phase is not enough to compensate for random effects intrinsic to the initial sampling.

The difference in mean $RMSE_{0.1}$ error between U60 and U80 remains approximately constant over the investigated dimensions: increasing the initial sampling by 20% brings about 15 to 20% improvement. Such a large improvement suggests that the surrogate model accuracy obtained with the U60 strategy is often inadequate to proceed with the adaptive phase: if in low dimensions the Bayesian approach compensates for the inaccuracy of the initial sampling, in high dimensions this is no more the case. The difference in mean error between U80 and U100 is very large in 2D –i. e., 40%– then settles around 7 to 9% for dimensions 5 to 15. This indicates that, even in large dimensions, the GP accuracy can be significantly enhanced by decreasing the degree of sparseness of the samples.

In 5D, the sampling strategies U60/A40 and U80/A20 perform equally, both in terms of mean and standard deviation. The improvement with respect to U100 is 20% in mean value and 33% in standard deviation.

From 8D upwards, U80/A20 performs better than U60/A40, suggesting that a better starting point –namely, a larger initial sampling– prevails over an extended adaptive phase. In 8-, 12- and 15-D, U80/A20 reduces the mean error with respect to U100 of, respectively, 6%, 3%, 1%. The standard deviation is, generally, more significantly improved. In 15D, the adaptive strat-

egy U60/A40 performs slightly worse than the uniform strategy U100.

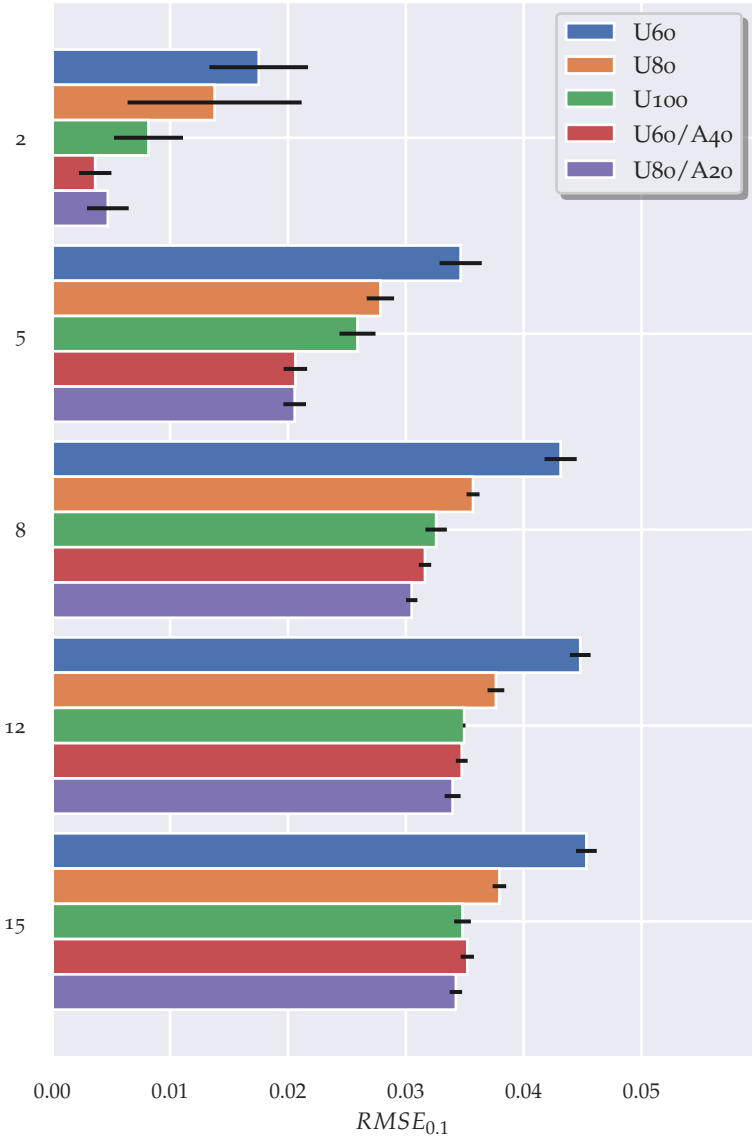
Summing up, as dimension increases, two effects are notable: first, the $RMSE_{0.1}$ error increases; second, the beneficial effect of adaptive strategies decreases. The two effects are, actually, related: the larger the $RMSE_{0.1}$ error, the less accurate the surrogate model, the larger the uncertainty of the surrogate model prediction, the more EI_{feas} tends to *explore* rather than *exploit*, the less effective EI_{feas} is, the more spread the points are around the feasibility threshold $\psi = 0$. The inaccuracy of the surrogate model comes at a large extent from the increasing degree of sparseness of the samples in high dimensions, displayed by the minimum distance distribution in Figure 6.12. In other words, it comes from the insufficient sampling budget. Furthermore, the performance of the multi-modal optimizer –namely, the NMMSO algorithm– may worsen in high-dimensions.

Figure 6.13b shows the cross-validated data: as discussed for the Branin function, also here we can note that cross-validation underestimates the accuracy. Yet, the effect reduces as the problem dimensionality increases. This may be related to the increase in sample size. Analogously to the Branin case, adaptive strategies are more underestimated than uniform ones: from dimension 5 upwards, the U100 strategy achieves the best cross-validated score.

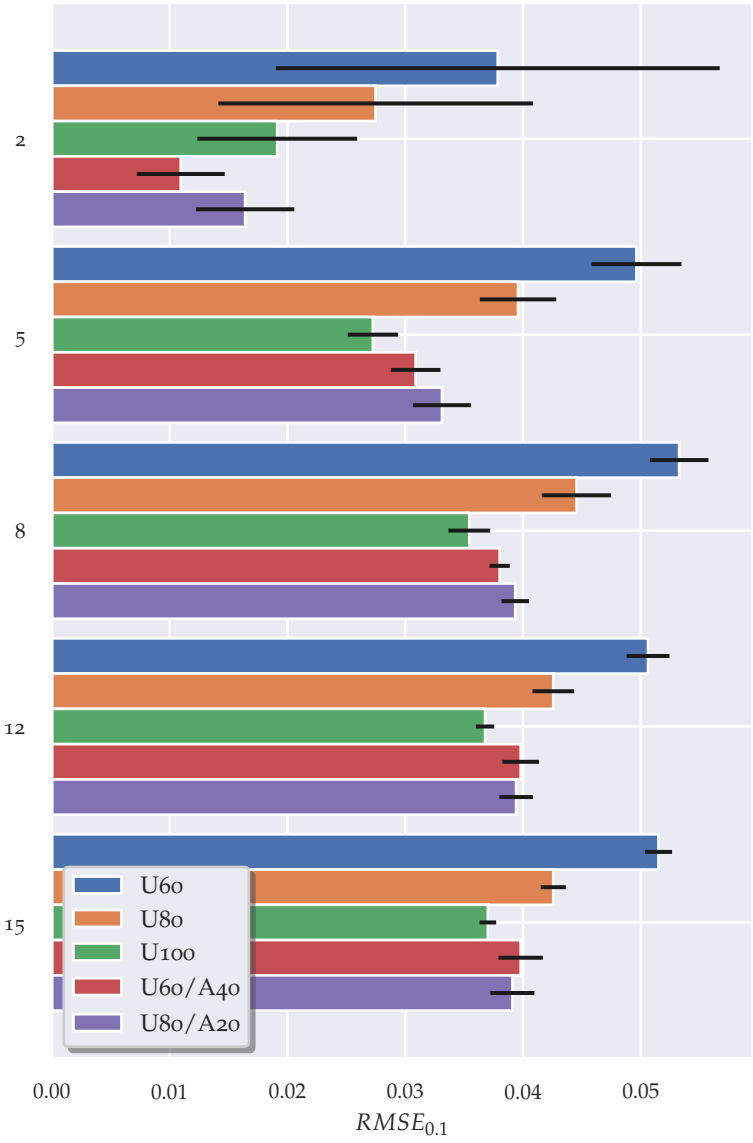
Table 6.5 reports relative regression and classification metrics for all investigated dimensions and sampling strategies with new test data. Globally, the R^2 score does not undergo substantial drops: even in 15D, U60 scores 0.955 and U100 0.974 with very low standard deviation.[9] Locally, though, the accuracy decreases more significantly: $R^2_{0.1}$ passes from 0.996 of U60/A40 in 2D to 0.643 of U80/A20 in 15D.[10] Therefore, if in 2D the global accuracy can be achieved –with adaptive strategies– also locally, in high dimensions that is no more the case. Furthermore, in high dimensions, even the standard deviation is little improved.

9 The cross-validated R^2 performs similarly. Therefore, the condition $R^2 \geq 0.8$ in Figure 5.8 is satisfied for any uniform strategy in any dimension.

10 The $R^2_{0.1}$ of the worst-performing strategy, U60, passes from 0.903 in 2D to 0.373 in 15D.



(a) New test data



(b) Cross-validation data

Figure 6.12: Bar plot of mean and standard deviation of *RMSE* for $\psi \in [-0.1, 0.1]$, for different dimensions and sampling strategies, with the Rosenbrock-based problem in Equation 6.1, with: (a) new test and (b) cross-validation data.

The classification metric $MCC_{0.1}$ steadily decreases as dimension increases; less dramatically, though, than $R^2_{0.1}$. The difference between best- and worst-performing sampling strategy remains approximately constant over the dimensions, about 8%. Up to 8D, the feasibility analysis capabilities of the surrogate model look improved with an adaptive strategy. From 12D upwards, those capabilities do not seem to differ significantly between U80/A20 and U100 strategies, certainly less than the regression capabilities.

Noteworthy, with any metric used, the accuracy of the surrogate model for $\psi \in [-0.1, 0.1]$ decreases rapidly across low-dimensions –say from 2D to 8D– yet only slightly across high-dimensions –from 8D to 15D. Furthermore, the difference in accuracy among sampling strategies that make use of the whole sampling budget is large in low-dimensions and little in high-dimensions. Therefore, it seems that the surrogate model accuracy with the Rosenbrock-based problem converges as the number of dimensions increases and that to improve this accuracy becomes more difficult without an increase in sampling budget.

QUESTION 3. Only the strategy U60 does not satisfy, on average, the validation condition $RMSE_{0.1} \leq 0.05$ with cross-validation data from 8D upwards. All strategies that make use of the whole sampling budget do fulfill the validation condition. However, the margin with respect to the threshold becomes smaller as the number of dimensions increases. A stricter condition, such as $RMSE_{0.1} \leq 0.03$, may require a much larger sampling budget for any sampling strategy. Based on the results, I choose to adopt the U60/A40 sampling strategy up to 7D and U80/A20 from 8D upwards.

To conclude, the Rosenbrock-based test problem confirms that Bayesian approaches are great in low-dimensions. Here, they significantly boost the accuracy of the surrogate model in feasibility analysis. It is questionable whether Bayesian approaches, *with the current sampling budget*, should be applied in high-dimensional spaces: they are very computationally demanding and seem to bring little benefit. This discussion will be examined more in depth in Chapter 8.

DIM	STRATEGY	R^2		$R^2_{0.1}$		$MCC_{0.1}$	
		Mean	Std	Mean	Std	Mean	Std
2	U60	0.997	0.001	0.903	0.047	0.905	0.024
	U80	0.999	0.001	0.927	0.081	0.942	0.016
	U100	0.999	0.001	0.955	0.060	0.955	0.018
	U60/A40	0.999	0.001	0.996	0.003	0.984	0.006
	U80/A20	1	0	0.992	0.006	0.979	0.004
5	U60	0.984	0.001	0.638	0.038	0.763	0.009
	U80	0.990	0.001	0.766	0.020	0.814	0.008
	U100	0.992	0.001	0.792	0.028	0.828	0.009
	U60/A40	0.990	0.001	0.872	0.012	0.848	0.007
	U80/A20	0.993	0.001	0.872	0.012	0.849	0.006
8	U60	0.971	0.001	0.435	0.034	0.684	0.009
	U80	0.979	0.001	0.614	0.012	0.735	0.007
	U100	0.983	0.001	0.673	0.018	0.758	0.008
	U60/A40	0.979	0.001	0.696	0.009	0.752	0.006
	U80/A20	0.983	0.001	0.718	0.009	0.765	0.006
12	U60	0.961	0.002	0.389	0.025	0.656	0.008
	U80	0.972	0.001	0.567	0.015	0.709	0.008
	U100	0.977	0.001	0.624	0.011	0.734	0.006
	U60/A40	0.972	0.001	0.632	0.012	0.724	0.008
	U80/A20	0.975	0.001	0.647	0.013	0.730	0.007
15	U60	0.955	0.001	0.373	0.025	0.637	0.006
	U80	0.969	0.001	0.562	0.014	0.703	0.004
	U100	0.974	0.001	0.632	0.014	0.727	0.011
	U60/A40	0.969	0.001	0.621	0.013	0.714	0.007
	U80/A20	0.972	0.001	0.643	0.011	0.727	0.005

Table 6.5: Regression and classification metrics with new test data, for different dimensions and sampling strategies, for the Rosenbrock-based problem in Equation 6.1.
The subscript $_{0.1}$ indicates that the metric is computed for $\psi \in [-0.1, 0.1]$.

Influence of Multiple Surrogate Models

Being the Rosenbrock-based feasibility function composed of two constraints, I can evaluate the effect of training one surrogate model for each constraint, instead of only a single one directly for the feasibility function. Figure 6.13 compares the $RMSE_{0.1}$ with a single surrogate and with multiple surrogates for the U60/A40 strategy, for different dimensions, with new test data. The difference is extremely large. Training multiple surrogate models boosts the accuracy, especially in low-dimensions: the mean error sinks of about 90% in 2D and of 60% in 15D. Also the standard deviation is, regularly, at least 70% less.

Hence, points of non-differentiability do undermine the accuracy of GP models and each constraint should be individually trained, as long as the computational budget allows it.

6.2.2 *Stochastic Optimization of Flexibility Boxes*

After training of the GP regression model for feasibility analysis, the proposed workflow for design flexibility optimization continues with the stochastic optimization of flexibility boxes. The design space is, first, reduced to the minimum axis-aligned box that bounds the feasible region; then, I sample this box and run an algorithm to *find the p maximal feasible boxes with highest flexibility level, whose relative overlap is lower than or equal to* q_{max}. The detailed procedure is described in Section 5.2.6 and summarized in Figure 5.16.

The flexibility boxes will not be *perfectly feasible*. I accept their feasibility level if the measured percentage of feasible samples among 200 ones randomly drawn from a uniform distribution is larger than 90%. With this percentage, it is possible to estimate a probability of feasibility inside the box larger than 85% at 97.5% confidence level.[11]

Besides the evaluation of the validation condition, in the following investigations I will focus on the effect of two factors on the flexibility boxes:

11 See Section 5.2.6 for details.

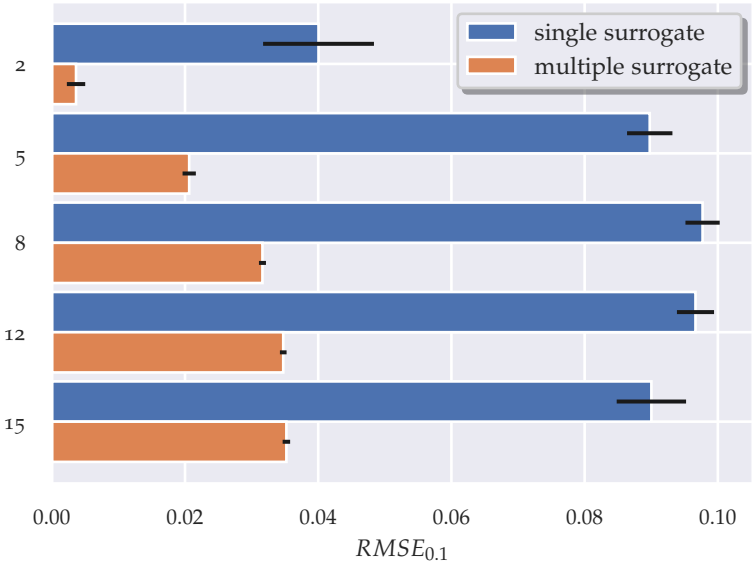


Figure 6.13: Bar plot of mean and standard deviation of *RMSE* for $\psi \in [-0.1, 0.1]$ with a single surrogate model, approximating the feasibility function, or multiple ones, approximating each constraint of the Rosenbrock-based problem in Equation 6.1, with the U60/A40 strategy, for different dimensions, with new test data.

- the curse of dimensionality –that is, in this case, the effect of samples' sparseness in high-dimensions,
- the accuracy of the input surrogate model –trained and validated in the previous sub-section.

Feasible Hyper-Sphere

The first test problem is a feasible hyper-sphere, investigated in 2-, 5-, 8-, 12 and 15-dimensions.[12] The peculiarities of this shape are the non-linearity, the multi-dimensional generalization and the possibility to compute analytically its largest inscribable hyper-cube as measure of comparison.

No underlying feasibility function is defined: the samples are classified as feasible or infeasible depending on the Euclidean distance from the sphere center and its radius. Therefore, Bayesian feasibility analysis does not apply to this test problem: samples' feasibility is not predicted via a surrogate model, rather based on *true* evidence. Hence, the results are not affected by the input surrogate model accuracy, instead –at a first approximation– by the amount of samples and Algorithm 5.2 only.[13]

I define the design space as $\mathbf{X} = \prod_{i=1}^{d} [0, 1]$ and set the hyper-sphere center at $\mathbf{x_c} = (0.5, \dots, 0.5)$. The hyper-sphere radius R remains to be set. The largest hyper-cube inscribable in a hyper-sphere is the one whose diagonal D equals the diameter of the hyper-sphere, $D = 2R$. Being the diagonal of a hyper-cube $D = \sqrt{d}\, l$ –where d is the number of dimensions and l the length of a hyper-cube edge– the radius of the hyper-sphere can be expressed as:

$$R = \frac{\sqrt{d}}{2} l. \tag{6.2}$$

12 A feasible hyper-*cube* that covers a quarter of the design space can be successfully solved with more than 99% accuracy in any dimension in the range 2 to 15. This is simply obtained by computing the minimum axis-aligned bounding box. Due to the not-so-interesting results, I do not include the hyper-cube as test problem in this work.

13 The position of the samples, chosen according to an LHS scheme, also plays a role and causes the variance of results among the 10 repetitions.

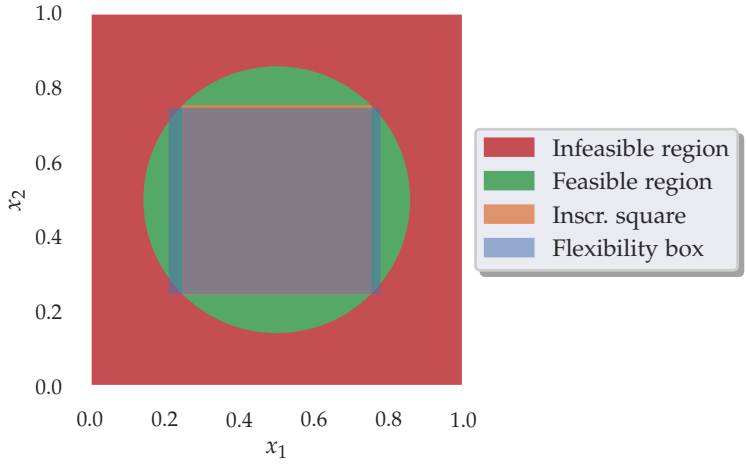


Figure 6.14: Largest flexibility box for the 2D variant of the hyper-sphere test problem in Equation 6.4. The largest square inscribable in the feasible circle is highlighted for sake of comparison.

I set $l = 0.5$, i. e., half the length of a design space edge. Hence:

$$R = \frac{\sqrt{d}}{2} 0.5. \tag{6.3}$$

The feasibility condition is, therefore:

$$\|\mathbf{x} - \mathbf{x_c}\| \leq \frac{\sqrt{d}}{2} 0.5 \tag{6.4}$$

where $\|\cdot\|$ indicates the Euclidean norm and $\mathbf{x} \coloneqq (x_1, \ldots, x_d) \in \mathbf{X}$.

In this test problem, I look for the largest flexibility box only. Figure 6.14 shows design space, feasible circle, largest inscribable square and an exemplary output flexibility box in 2D. The flexibility box is very close to the maximum inscribable square, yet slightly larger. In fact, all four vertices lie in the infeasible region. However, this is acceptable as –with a 99.2% percentage of feasible samples– the probability of feasibility inside the box is still larger than 96.8% with 97.5% confidence level.

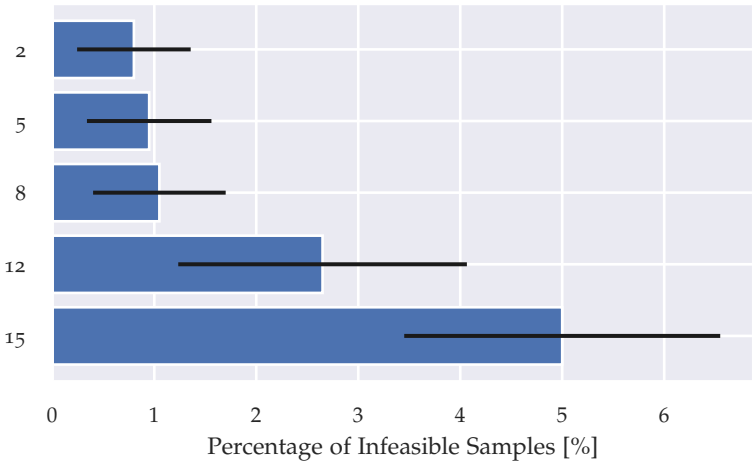


Figure 6.15: Bar plot of percentage of infeasible samples in the largest flexibility box out of 200 ones randomly drawn from a uniform distribution, for different dimensions, for the hyper-sphere test problem in Equation 6.4. Noteworthy, the almost-exponential behavior.

Figure 6.15 plots the percentage of infeasible samples in the largest flexibility box for all investigated dimensions. The validation condition is always satisfied. Up to 8D, the infeasible samples cover, on average, less than 1% of the total, which means a probability of feasibility larger than 96.4%. In 12- and 15-D, the infeasibility exponentially increases to, respectively, 2.8 and 5%. The growing sparseness of the samples in high-dimensions leads to larger voids where feasibility is not proved and, thus, lowers the method accuracy. Nevertheless, the probability of feasibility, even in 15D, is larger than 91% and, therefore, satisfactory.

The feasible hyper-sphere problem, besides the assessment of the feasibility of flexibility boxes in multi-dimensions, offers the possibility to compare their volume with that of the largest inscribable –hence, 100% feasible– hyper-cube, denoted as reference box in the following. The volume of this box can be computed as:

$$V = l^d = 0.5^d. \tag{6.5}$$

DIM	REFERENCE BOX VOL	FLEXIB. BOX VOL		RATIO FLEX/REF	
		Mean	Std	Mean	Std
2	0.25	0.28	5.9E-3	1.1	0.024
5	0.031	0.081	1.8E-3	2.6	0.059
8	3.9E-3	0.029	9.4E-4	7.5	0.24
12	2.4E-4	0.011	3.6E-4	45	1.46
15	3.1E-5	6.2E-3	2.2E-4	204	7.27

Table 6.6: Volume of reference and flexibility boxes and relative ratio for different dimensions, for the hyper-sphere problem in Equation 6.4.

Table 6.6 reports the volume of reference and flexibility boxes and their ratio for all investigated dimensions.[14] In 2D, the optimum flexibility box is about 10% larger than the reference one. In 5D and 8D, it is, respectively, 2.6 and 7.5 times as large as the reference one. The difference in volume is remarkable. The difference in feasibility, though, is not as significant: almost negligible –as depicted in Figure 6.15. As for the 2D case shown in Figure 6.14, also in higher dimensions *all* the vertices are infeasible. The number of vertices of a box is 2^d: 4 in 2D, 256 in 8D, 32768 in 15D. By compromising the vertices, then, a great volume can be included in the flexibility box, which is excluded by the largest inscribable cube. Besides, due to the shape of the hyper-sphere, most of this volume belongs to the feasible region, as only the area around the vertices is infeasible. Therefore, as both more feasible and more infeasible regions are added, the probability of feasibility remains approximately constant, yet the volume profits from both. The higher the number of dimensions, the larger this effect. It comes at no surprise, now, that in 12D the flexibility box is 45 times larger than the reference one and in 15D 204 times. Here, though, the percentage of infeasible samples increases, as a result of the larger samples' sparseness.

14 Noteworthy, the effect of dimensionality on the reference box: whereas in 2D, a cube with 0.5 edge length covers 25% of the unitary cube, in 15D only 0.003%.

It is worth to highlight that compromising the vertices is not a deliberate choice, rather a side-effect of both the sample-based approach used to compute boxes and the choice of ignoring the feasibility of the vertices.

To sum up, on the basis of the results of the feasible hyper-sphere test problem we can state that as the dimensionality increases, the volume of the flexibility box becomes significantly larger –in other words, it offers more flexibility– than that of the largest 100% feasible box. Remarkably, up to 8D, this happens without notably diminishing the probability of feasibility. In higher dimensional spaces, instead, the curse of dimensionality increases not only the volume, but also the probability of infeasibility. Nevertheless, the validation condition is satisfied for any investigated dimension.

Rosenbrock

Both the effect of dimensionality and that of the input surrogate model accuracy can be assessed with the Rosenbrock-based test problem, introduced in Section 6.2.1 and defined in Equation 6.1.

I run the flexibility boxes optimization workflow in Figure 5.16 both with and without inputs from the Bayesian feasibility analysis workflow, validated in Section 6.2.1. The first case is how I intend to use the proposed method, while the second one is how I validated the feasible hyper-sphere test problem –that is, with samples based on true observations, rather than on predictions. In the following, the two cases will be denoted as, respectively, prediction-based and observation-based.

First, I look for the three largest flexibility boxes, whose relative overlap does not exceed 50%. Figure 6.16 shows design space, feasible region and resulting boxes in 2D with prediction-based samples. Similarly to the feasible hyper-sphere test problem, most of the vertices lie in the infeasible region. Due to the shape of the feasible region, not all vertices actually lie in the infeasible one. On average, the percentage of feasibility measured in the boxes lies around 96%, which ensures a probability of feasibility larger than 92% with 97.5% confidence level. Therefore, according to the

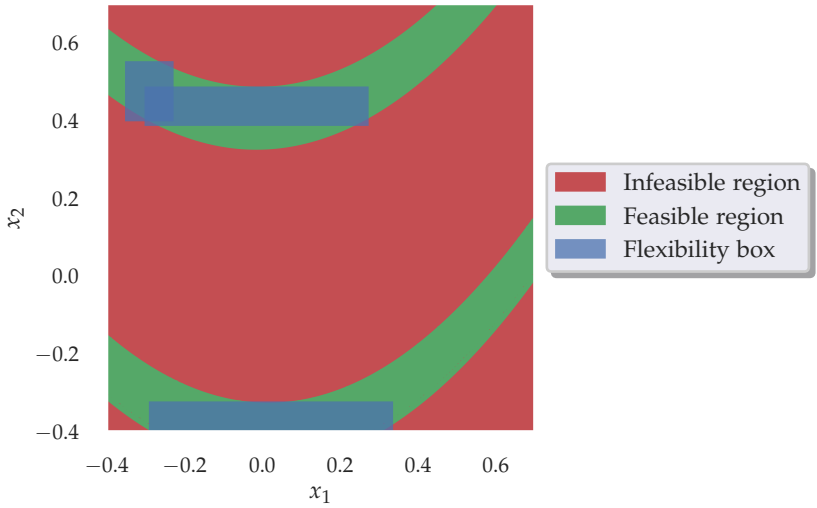


Figure 6.16: Three largest flexibility boxes, whose relative overlap is lower than or equal to 0.5, with prediction-based samples, for the 2D variant of the Rosenbrock-based problem in Equation 6.1.

validation condition, the level of feasibility inside the boxes in 2D is acceptable.

Looking for three boxes –rather than one– offers, first, alternative solutions to satisfy the constraints and, then, valuable information on the feasible region. Figure 6.17 shows the parallel coordinates plot with lower and upper limits of the flexibility boxes in Figure 6.16. The two largest boxes are similar in volume and in the range of flexibility of x_1 –say [-0.3, 0.3]– but very different in x_2 values: box 1 lies around 0.4, while box 2 around -0.4. If this was a real design problem, we would have to decide how to satisfy the requirements between two opposite designs. Moreover, as box 3 is significantly smaller, we would realize that we would lose a lot of design flexibility if we were not able to stay in box 1 or box 2. Then, as no flexibility box has $x_1 \geq 0.4$, we would also learn that, although other feasible solutions may lie in this range, their flexibility would be very low. For a 2D problem, the previous

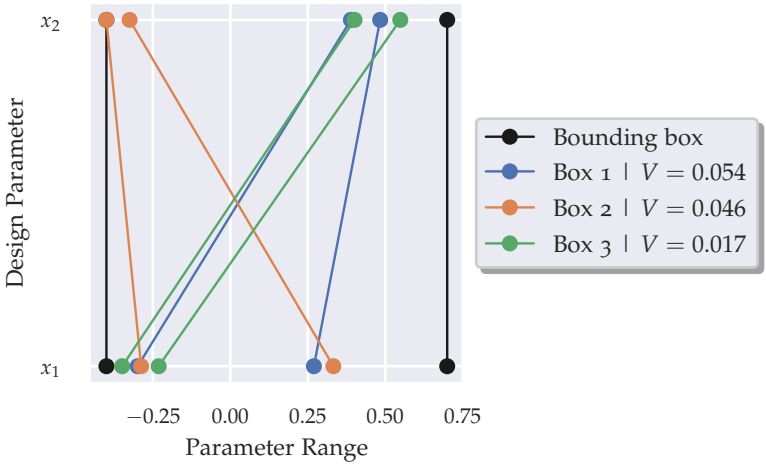


Figure 6.17: Parallel coordinates plot of the three largest flexibility boxes, whose relative overlap is lower than or equal to 0.5, with prediction-based samples, for the 2D variant of the Rosenbrock-based problem in Equation 6.1.

considerations can be easily made by looking at the plot of the design space, such as Figure 6.16. However, in higher dimensions, a parallel coordinates plot of the flexibility boxes is one of the best ways to understand how the feasible region extends.

Figure 6.18 shows mean and standard deviation of the average percentage of infeasible samples in the resulting boxes, with both prediction- and observation-based samples, for all investigated dimensions. First of all, contrary to what we might expect, the boxes computed with samples predicted by the GP regression model contain, on average, less infeasible samples than those computed with true observations; this consistently happens in any dimension. The difference is only slight –1.5% at most– yet an explanation may come from the average volume of the boxes. Figure 6.19 shows the average volume of the boxes considered in Figure 6.18.[15] The boxes computed with observation-based sam-

15 It is worth to compare the fraction of feasible region inside the design space, reported in Table 6.4, with the average volume of the largest three flexibility boxes.

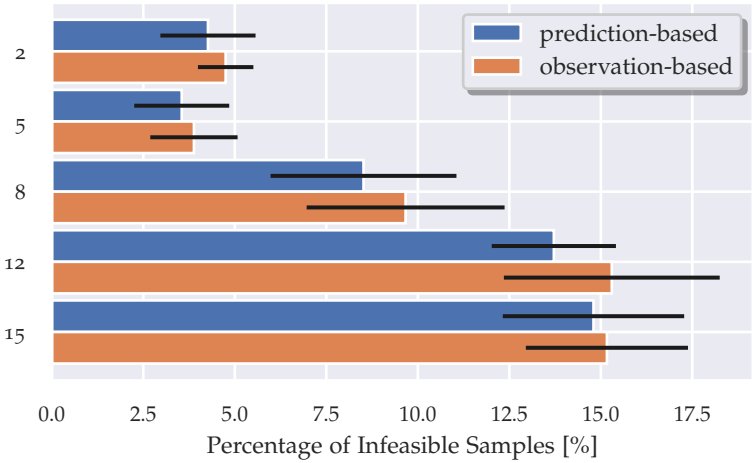


Figure 6.18: Bar plot of average of percentage of infeasible samples in the three optimized flexibility boxes, out of 200 ones randomly drawn from a uniform distribution, for different dimensions, for the Rosenbrock-based problem in Equation 6.1.

ples are consistently larger than the prediction-based counterparts. This suggests that the surrogate model, trained in the previous subsection, may slightly underestimate the size of the feasible region. Accordingly, the boxes computed with predictions would be, first of all, smaller than the observation-based ones and, then, more likely to be inside the feasible region.

The validation condition for the Rosenbrock-based test problem is only satisfied in 2-, 5 and 8-dimensions. In 12- and 15-D, the measured percentage of feasible samples lies below 90% and

In 2D, the feasible region covers about 22% of the design space, while one box 4%. In 15D, the feasible region covers about 53% of the design space and the flexibility box 1.5%. This means that, in 2D, approximately 82% of the feasible region is excluded by the flexibility box and, in 15D, 97%. In other words, the flexibility box neglects 97% of the feasible solutions. This is a direct consequence of the choice to look for axis-aligned boxes.

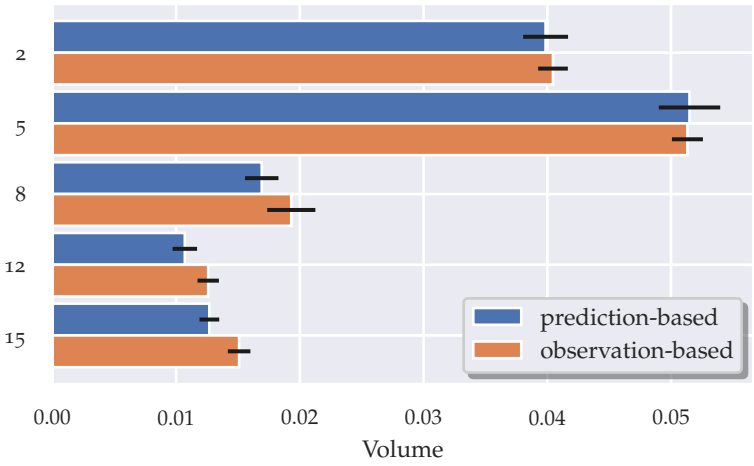


Figure 6.19: Bar plot of average of volume of the three optimized flexibility boxes, for different dimensions, for the Rosenbrock-based problem in Equation 6.1.

approaches 85%.[16] This means that, in these dimensions, the probability of feasibility inside the flexibility box is, on average, larger than 80% with 97.5% confidence level.

It is worth looking at the feasibility function value of the infeasible samples in the flexibility boxes in 15D. The PDF distribution of these samples' ψ is shown in Figure 6.20, together with the distribution of infeasible samples inside the design space as measure of comparison. Although the probability of infeasibility can be as high as 20% in the flexibility boxes, the infeasible samples mostly lie close to the feasible region boundary –say, up to 0.1. Hence, feasibility is not too far away: if this was a real design problem, these samples would have potential to be turned in feasible designs. Yet, due to the fast growing shape of the Rosenbrock function, there are as well samples inside the flexibility boxes with $\psi > 0.1$.

16 According to the workflow in Figure 5.16, I should increase at this point the amount of samples used to compute the boxes, in order to enhance their feasibility level. I do not report, though, this investigation in the present work.

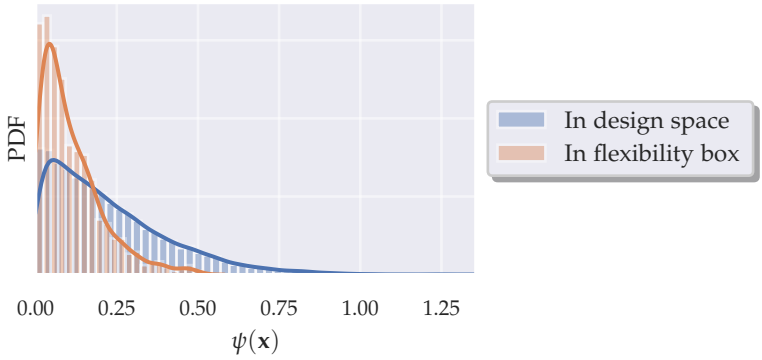


Figure 6.20: Probability Density Function (PDF) of the feasibility function ψ of the infeasible samples inside the flexibility boxes and inside the design space, in 15-dimensions, for the Rosenbrock-based problem in Equation 6.1.

Finally, I investigate the benefit of the DDS tree-search approach in Algorithm 5.2 to find the maximal feasible boxes. Extending the search further than the left-most path, as in Algorithm 5.1, increases the chances, not only to find the largest box, but all maximal –i. e., inextensible– boxes. This is especially important both when looking for different boxes and when not using volume as flexibility measure.

The increase in boxes' maximality thanks to the tree-search can be evaluated in many ways. I measure the increase in volume, from the 0-th iteration –i. e., the left-most path– to the 5-th iteration, of the 50 largest feasible boxes whose overlap is lower than or equal to 90%. I stop the search at the 5-th iteration to limit the computational effort. Since the longer the tree-search goes, the more boxes it returns, I have to filter them so as to find a reasonable set of boxes for comparison. Therefore, each box from the 5-th iteration may have only *one* box among those from the 0-th iteration with an overlap larger than 90%. This ensures that each box from the upper iteration finds at most one equivalent in the lower iteration.

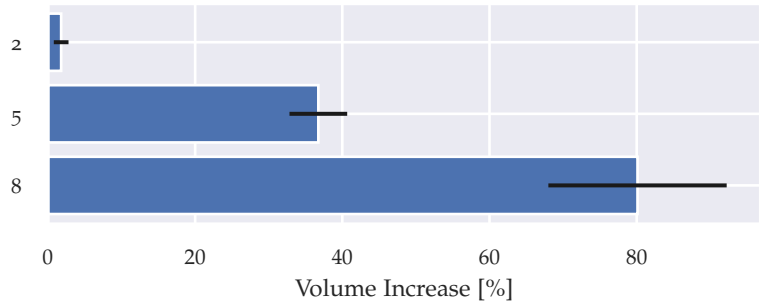


Figure 6.21: Increase, from 0-th iteration to 5-th iteration of Algorithm 5.2, in sum of volumes of largest 50 flexibility boxes whose maximum overlap does not exceed 0.9, for 2-, 5- and 8-dimensions, for the Rosenbrock-based problem in Equation 6.1.

I run the investigation for 2-, 5- and 8-dimensions. In 12- and 15-dimensions, the search until the 5-th iteration is too computationally intensive.[17] I do not stop the search earlier for these dimensions, as it would compromise the comparison of results among dimensions. Figure 6.21 shows the increase in the sum of the volumes of the resulting boxes. In 2D, the increase is negligible: 1.8%. Being the branching factor only 2, the heuristic regularly predicts the correct branch to expand. Nevertheless, the computational effort involved in running the tree-search up to the 5-th iteration is very low. In other words, little effort, little benefit. In 5D, both effort and benefit grow: the flexibility boxes are, on average, about 37% larger after the 5-th iteration, yet the computational time –on 32 Central Processing Units (CPUs)– passes approximately from 10 seconds to 10 minutes. In 8D, the trend amplifies: the volume gain is about 80% and the run time changes from 3 minutes to 3 hours.

To sum up, the Rosenbrock-based test problem suggests that the inaccuracy of the *trained* surrogate model has only a minor im-

17 The higher the dimension, the more samples are used. Therefore, both the size of each tree and the number of trees significantly increase.

pact on the probability of feasibility inside the flexibility boxes.[18] The major impact is given by the sparseness of samples in high-dimensions. The validation condition is not satisfied for 12- and 15-dimensions with the standard amount of feasible samples inside the bounding box. This amount should be increased to reach the satisfactory level. Nevertheless, the infeasible samples lie close to the feasibility threshold. Few iterations of tree-search may significantly improve the chances of finding the maximal boxes, though the involved computational cost is not negligible.

18 N.B.: this does not mean that the input surrogate model accuracy is not relevant. It means that the accuracy of the *trained* surrogate model is *high enough* to not play a major role in the determination of the probability of feasibility inside the flexibility box. It is, indeed, an excellent confirmation of the validity of the proposed method for Bayesian feasibility analysis.

Part III

THE SIGNIFICANCE.

7

DESIGN GUIDELINES FOR THE ADVANCED PEDESTRIAN LEGFORM IMPACTOR

God help us, we're in the hands of engineers.

— Ian Malcolm[1]

In the very early phase of the development of a new vehicle, after the definition of its requirements at system-level, concept engineers start to investigate what is necessary, at sub-system level, to satisfy the requirements. Normally, a useful orientation comes from the analysis of a similar vehicle at the end of the development process. Numerical simulation is essential in this phase. Often, a preliminary, rough Computer-Aided Design (CAD) model is constructed, through which first, rudimentary indications are collected.

For pedestrian leg impact's requirements, the method proposed in this thesis offers a tool that can be used right before the availability of –or even together with– early CAD models. This tool provides the means to obtain geometrical and structural guidelines on the most relevant load regions. This information affects, in turn, the next construction cycle of Class A surface and CAD model for the new vehicle.

In the present chapter, I describe an exemplary investigation of design guidelines for the pedestrian leg impact in the initial phase of the development process of a medium-sized sedan. The investigation consists of three steps: first the design parameters and their respective ranges are defined, then a surrogate model is trained for feasibility analysis and, finally, flexibility boxes are identified in the design space where the requirements are, for the most part, satisfied. The investigation is run with, both, aPLI and FlexPLI to highlight the differences in guidelines for front-end design between the two.

1 Fictional character. From the movie: *Jurassic Park*. 1993.

7.1 FINITE ELEMENT LOW-FIDELITY MODEL

First, I choose model, loading condition and design space for the investigation. I use, as a reference, the model shown in Figure 4.3b, and validated in Section 6.1, with the advanced Pedestrian Legform Impactor (aPLI). To increase the generality of the design guidelines, I employ the cubic fit of the original loading force-deformation characteristics of the load levels.

Due to the initial phase of the development process, the investigation should be comprehensive. Therefore, I consider all four load regions depicted in Figure 4.10 and I vary both their structural behavior and position, as described in Section 4.2. I define a scaling factor for the entire loading force-deformation curve, a change in dissipation factor for the unloading, a change in deformation space in front of the bumper beam and translations in the xz-plane. Specifically, I define for the load region:

LOW: scaling factor $s_{1_{low}}$, dissipation factor difference ΔD_{low}, z-position difference Δz_{low};

MID: scaling factor $s_{1_{mid}}$, dissipation factor difference ΔD_{mid}, x-position difference Δx_{mid}, deformation space difference $\Delta u_{DS_{mid}}$;

MID-UP: scaling factor $s_{1_{mid\text{-}up}}$, dissipation factor difference $\Delta D_{mid\text{-}up}$, x- and z-position differences $\Delta x_{mid\text{-}up}$ and $\Delta z_{mid\text{-}up}$;

UP: scaling factor $s_{1_{up}}$, dissipation factor difference ΔD_{up}, x- and z-position differences Δx_{up} and Δz_{up}.

In total, I use 15 parameters. All parameters are expressed relatively to the reference –i. e., original– values of the model. Their range of variation is rather broad and general, as befits initial investigations: I set +/-50% for the scaling factor, +/-20% for the dissipation factor difference, +/-20 mm for the rigid translations –similarly to what is done in [32]– and +/-10 mm for the deformation space difference.[2] Table 7.1 reports the list of 15 design

2 HFM variations, such as those performed in the validation Section 6.1, do not correspond, one-to-one, to LFM variations. Therefore, the defined LFM parameter ranges cannot be *exactly*, but only *indicatively* validated.

PARAMETER	DESIGN SPACE		UNIT
	Lower Lim	Upper Lim	
$s1_{low}$	0.5	1.5	–
ΔD_{low}	-0.2	0.2	–
Δz_{low}	-20	20	mm
$s1_{mid}$	0.5	1.5	–
ΔD_{mid}	-0.2	0.2	–
Δx_{mid}	-20	20	mm
$\Delta u_{DS_{mid}}$	-10	10	mm
$s1_{mid\text{-}up}$	0.5	1.5	–
$\Delta D_{mid\text{-}up}$	-0.2	0.2	–
$\Delta x_{mid\text{-}up}$	-20	20	mm
$\Delta z_{mid\text{-}up}$	-20	20	mm
$s1_{up}$	0.5	1.5	–
ΔD_{up}	-0.2	0.2	–
Δx_{up}	-20	20	mm
Δz_{up}	-20	20	mm

Table 7.1: Name, range of variation inside the design space and unit of measure of the 15 design parameters used for the pedestrian leg impact's design guidelines investigation.

parameters with their respective range of variation. For investigations that take place later in the development process, both ranges and choice of parameters can be fine-tuned for specific purposes.

The constraints that delimit the feasible region are the 8 injury criteria measured with aPLI. I normalize them according to their requirement, as in Equation 5.1. The value of the feasibility function equals, then, for each sample, that of the largest normalized constraint, as in Equation 5.2. For the GP regression model, I

arrange the constraints in three groups and use each group as single output response of the surrogate model:[3]

TIBIA MAX: the maximum bending moment among the four sections on the tibia shaft,

MCL: the maximum Medial Collateral Ligament elongation,

FEMUR MAX: the maximum bending moment among the three sections on the femur shaft.[4]

7.1.1 *Understanding the advanced Pedestrian Legform Impactor*

In high-dimensions, the relevance of each parameter to an output response can be assessed by means of sensitivity analysis [115]. The information that can be acquired with scatter plots is, generally, limited. In fact, these show a projection of the multi-dimensional design space along one single dimension[5]. It follows that the dependency of the output response on the *combination* of two or more parameters cannot be appreciated. In other words, scatter plots only show the global, first-order effect of each parameter, independent on the interaction with the other ones. In some cases, however, this information may be still useful.

TIBIA MAX Figure 7.1 shows the scatter plot of Tibia Max for each design parameter with uniform sampling. Looking at the cubic polynomial fit, 6 parameters, out of 15, show a non-negligible global influence on Tibia Max in the investigated design space:

$s1_{low}$: the larger the scaling factor of the lower load region, the higher the maximum tibial bending moment; the trend soars for values greater than 1. Hence, the more the lower load region presses with force, the more the tibia bends.

ΔD_{low}: the larger the dissipation factor of the lower load region, the slightly lower the maximum tibial bending moment. A

3 See Section 5.1.5 for details.

4 Here, MAX refers to the fact that the maximum is computed among different injury curves. It is implied that, for each curve, the maximum is considered.

5 The second axis of the plot reports the output response.

weaker rebound of the lower load region helps lowering tibia's bending.

Δz_{low}: the higher the position of the lower load region, the higher the maximum tibial bending moment; the trend is linear and is the *most significant* one among the chosen design parameters. The larger the portion of tibia below the lower load region, the larger the bending of this portion due to inertia.

$s1_{mid}$: similar trend to $s1_{low}$, yet less strong, as the middle load region acts around the knee.

Δx_{mid}: for negative values, the more the middle load region moves forward, the higher the maximum tibial bending moment. We have a slight opposite trend for positive values. Positions around the current one seems positive for tibia.

$s1_{mid\text{-}up}$: the larger the scaling factor of the mid-up load region, the lower the maximum tibial bending moment. Supporting the top extremity of tibia, reduces its banana-like deformation shape.

Noteworthy, the upper load region seems to not contribute significantly –in the investigated design ranges– to the global trends of Tibia Max. The scatter plots for Tibia Max with FlexPLI are shown in Figure B.1.

MCL Figure 7.2 shows the scatter plot of MCL for each design parameter with uniform sampling. Looking at the cubic polynomial fit, 11 parameters, out of 15, show a non-negligible global influence on MCL in the investigated design space:

$s1_{low}$: the larger the scaling factor of the lower load region, the lower the maximum MCL elongation; the trend strengthens for values below 1 and almost zeroes for values around 0.5. Limiting tibia's intrusion in the lower load region helps reducing the misalignment with femur and starting earlier its rebound.

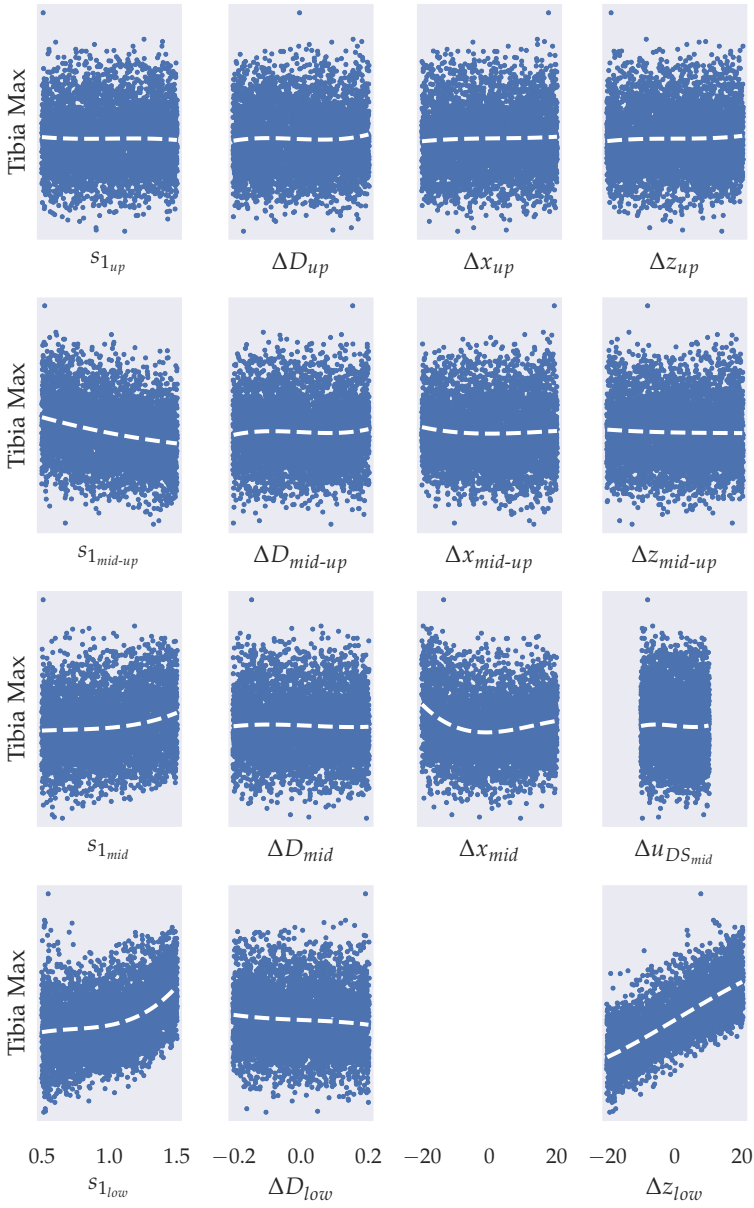


Figure 7.1: Scatter plot of Tibia Max and cubic polynomial fit –dashed– for each design parameter for the medium-sized sedan in Figure 4.3b, with the aPLI, inside the design space in Table 7.1.

ΔD_{low}: the larger the dissipation factor of the lower load region, the higher the maximum MCL elongation. The weaker the tibia rebound is, the slower tibia realigns with femur.

Δz_{low}: the higher the position of the lower load region, the higher the maximum MCL elongation. The larger the distance of the lower load region from the knee, the larger the lever arm, the less tibia intrudes.

$s1_{mid}$: the larger the scaling factor of the middle load region, the higher the maximum MCL elongation. Opposite trend as for $s1_{low}$, as the position of the middle load region hits around the knee. The more the knee intrudes, the more aligned the leg stays –in other words, the less the leg forms a banana-like shape.

ΔD_{mid}: the larger the dissipation factor of the middle load region, the slightly lower the maximum MCL elongation. The less the rebound on knee is, the less the leg forms a banana-like shape.

Δx_{mid}: the more the middle load region moves backward, the lower the maximum MCL elongation. Same explanation as for $s1_{mid}$.

$s1_{mid\text{-}up}$: the larger the scaling factor of the mid-up load region, the higher the maximum MCL elongation. The trend is the *most significant* one among the design parameters. The mid-up load region hits the top part of the knee and acts as a center of rotation for the UBM. The more this part intrudes, the less the leg forms a banana-like shape –as for the middle load region– *and* the later the UBM starts the rotation.

$\Delta D_{mid\text{-}up}$: the larger the dissipation factor of the mid-up load region, the lower the maximum MCL elongation. Same explanation as for ΔD_{mid}.

$\Delta x_{mid\text{-}up}$: the more the mid-up load region moves backward, the significantly lower the maximum MCL elongation. The more the top part of the knee freely intrudes, the less the

leg forms a banana-like shape and the later the UBM starts the rotation.

$\Delta z_{mid\text{-}up}$: the higher the position of the mid-up load region, the slightly higher the maximum MCL elongation. The more the mid-up load region moves towards the bottom part of the knee, the less it acts as center of rotation for the UBM.

Δz_{up}: the higher the position of the upper load region, the slightly lower the maximum MCL elongation. As the whole bonnet is shifted upwards, the higher it is, the less the UBM rotates before hitting it. However, the shift is little.

Noteworthy, as for Tibia Max, the upper load region contributes only slightly –in the investigated design ranges– to the global trends of MCL. The scatter plots for the maximum normalized ligament elongation with FlexPLI are shown in Figure B.2.

FEMUR MAX Figure 7.3 shows the scatter plot of Femur Max for each design parameter with uniform sampling. Looking at the cubic polynomial fit, 9 parameters, out of 15, show a non-negligible global influence on Femur Max in the investigated design space:[6]

$s1_{low}$: the larger the scaling factor of the lower load region, the higher the maximum femoral bending moment. The more the tibia bends –see effect on Tibia Max– the more the knee rotates around the y-axis, the larger the negative curvature of the femur around the y-axis.

Δz_{low}: the higher the position of the lower load region, the higher the maximum femoral bending moment. Same explanation as for $s1_{low}$.

$s1_{mid}$: the larger the scaling factor of the middle load region, the higher the maximum femoral bending moment. The more

6 N.B.: Initially, as the UBM is in free-flight, while tibia is stopped by the vehicle lower and middle load regions, the femur experiences a negative curvature around the y-axis. Then, as the UBM rotation around the y-axis progresses, the femoral curvature changes sign.

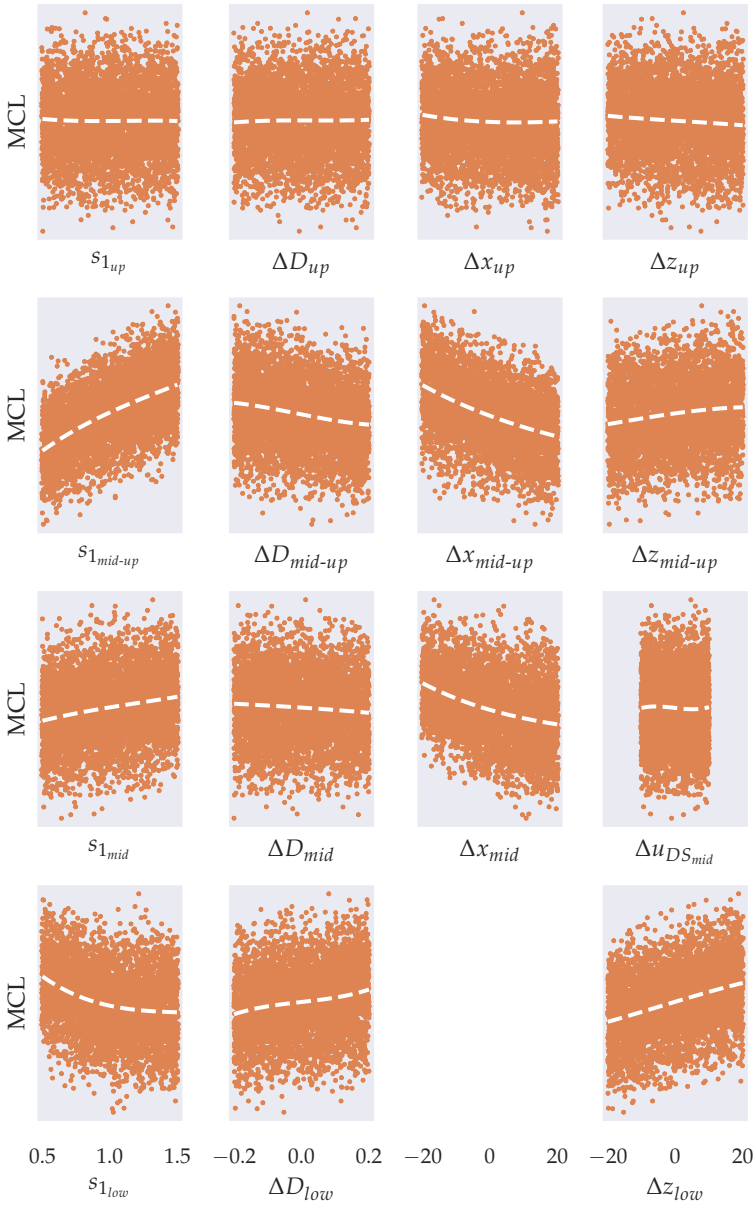


Figure 7.2: Scatter plot of MCL and cubic polynomial fit –dashed– for each design parameter for the medium-sized sedan in Figure 4.3b, with the aPLI, inside the design space in Table 7.1.

the knee intrudes, the less the femur bends in the negative direction around the y-axis.

Δx_{mid}: the more the middle load region moves backward, the lower the maximum femoral bending moment. The trend is the *most significant* one among the design parameters. Same explanation as for $s_{1_{mid}}$.

$\Delta u_{DS_{mid}}$: the more deformation space the middle load region has, the lower the maximum femoral bending moment. Same explanation as for $s_{1_{mid}}$.[7]

$s_{1_{mid\text{-}up}}$: the larger the scaling factor of the mid-up load region, the significantly lower the maximum femoral bending moment. The less the bottom part of the femur intrudes, the sooner the femur bends in the positive direction around the y-axis.

$\Delta x_{mid\text{-}up}$: the more the mid-up load region moves backward, the significantly higher the maximum femoral bending moment. Same explanation as for $s_{1_{mid\text{-}up}}$.

$\Delta z_{mid\text{-}up}$: the higher the position of the mid-up load region, the lower the maximum femoral bending moment. The larger the portion of femur below the mid-up load region, the less the femur bends in the negative direction around the y-axis.

Δx_{up}: for negative values, the more the upper load region moves forward, the lower the maximum femoral bending moment. Same explanation as for $s_{1_{mid\text{-}up}}$. Even moving the upper load region backward seems to have a slight beneficial effect on Femur Max.

Noteworthy, as for Tibia Max and MCL, the upper load region contributes only slightly –in the investigated design ranges– to the global trends of Femur Max.

7 N.B.: If the actual deformation does not make use of the whole permissible space, this parameter is *not active*. Therefore, if it shows no or little tendency, it is not possible to distinguish whether it is active and does not play a role or it is not active. However, this distinction is not relevant for the design guidelines.

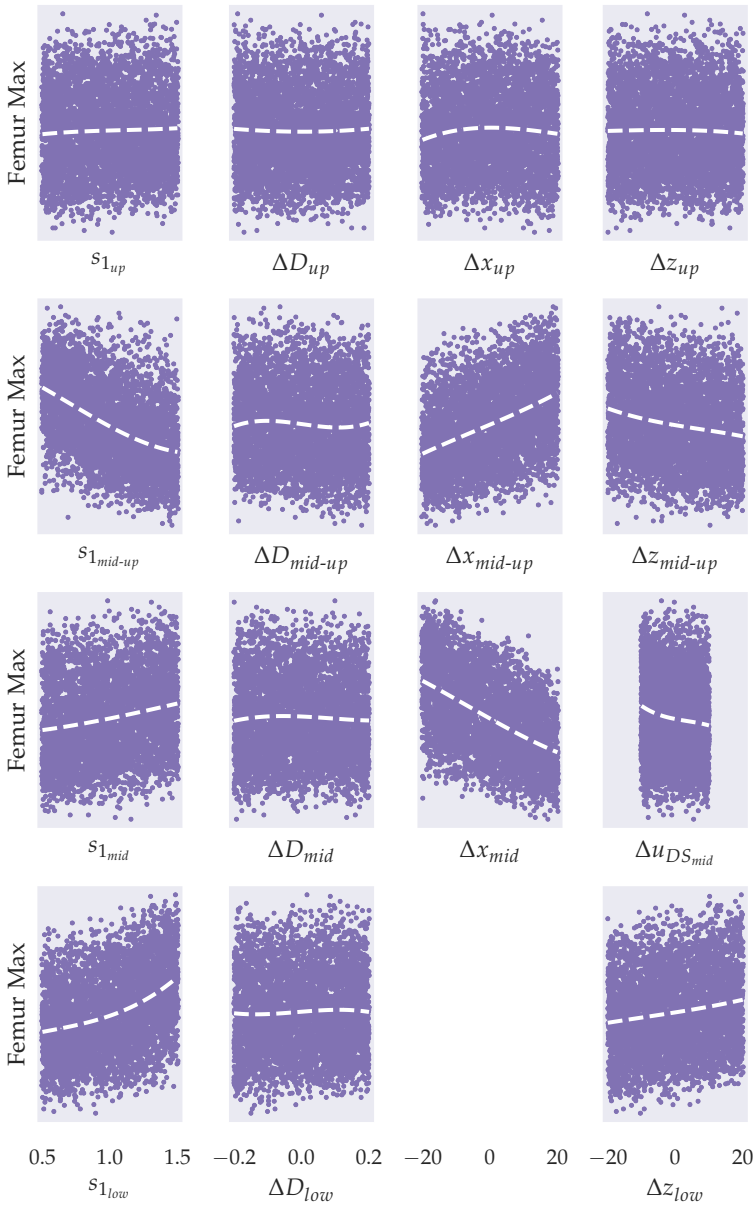


Figure 7.3: Scatter plot of Femur Max and cubic polynomial fit –dashed– for each design parameter for the medium-sized sedan in Figure 4.3b, with the aPLI, inside the design space in Table 7.1.

The above analysis takes into consideration each group of injury criteria *separately*. However, the significance of each parameter on feasibility depends on the actual maximum among the normalized injury criteria. In other words, a parameter that may be very relevant for *one* injury criterion, may be totally irrelevant for the overall fulfillment of requirements, if that criterion, normalized, is always lower than the other ones.

7.2 BAYESIAN FEASIBILITY ANALYSIS

The second step in order to compute the design guidelines is to train a surrogate model for feasibility analysis. I use a Bayesian model to approximate the feasibility function and, iteratively, add samples where they are most needed. According to Table 5.2, the budget defined for this investigation is 5625 samples; as stated in Section 5.1.5, it is split between 80% LHS and 20% adaptive sampling, distributed over 10 iterations.

Table 7.2 reports the R^2 coefficient and $RMSE_{0.1}$ error of the feasibility function ψ and of the single trained GP models after uniform and adaptive phases. The R^2 coefficient of ψ after the uniform phase scores 0.95, which fulfills the validation condition. Tibia Max seems to be the least accurately approximated injury criterion. One explanation may be that the maximum bending moment here occurs at different sections over the design space, making Tibia Max C^0 continuous.

The scores do not improve after the adaptive phase. Rather, some slightly worsen. This is, most likely, due to the use of cross-validation data to compute the scores. As stated in Section 5.1.4 and demonstrated in Figure 6.10b and Figure 6.13b, cross-validation is not reliable when combined with adaptive sampling for feasibility analysis. Furthermore, for both Branin- and Rosenbrock-based problems in Section 6.2.1, U80/A20 was the most underestimating strategy.

To assess more comprehensively the adaptive phase, it is useful to look at both the feasibility function value and the location of the added samples. Figure 7.4a shows the PDF of ψ for the samples added uniform and adaptive phases. The latter consistently adds observations close to the feasible region boundary, rarely

RESPONSE	R^2		$RMSE_{0.1}$	
	U80	U80/A20	U80	U80/A20
ψ	0.95	0.95	0.018	0.019
Tibia Max	0.95	0.94	0.019	0.020
MCL	0.99	0.99	0.010	0.014
Femur Max	0.98	0.97	0.018	0.020

Table 7.2: R^2 coefficient and *RMSE* in the range $[-0.1, 0.1]$, $RMSE_{0.1}$, after uniform, U80, and adaptive, U80/A20, phase of Bayesian feasibility analysis, with cross-validation data, for the design problem in Section 7.1.

exceeding the range $[-0.05, 0.05]$. Hence, the samples from the adaptive phase successfully provide information on the area of interest.

Figure 7.4b shows the PDF of the minimum distance for the same sets of samples. As expected, the standard deviation of the adaptive distribution is larger than for the uniform one, almost double. Surprisingly, however, the samples added during the adaptive phase are located, in general, further from their closest one compared to those added during the uniform phase. This was not the case with the mathematical functions used in Section 6.2.1. An explanation may be found looking at the scatter plots of the feasibility function against the design parameters for the samples added during the adaptive phase, shown in Figure 7.5. For most of the design parameters, samples are added at the extremities of their range of variation. This means that they are located either in vertices or in outer hyper-faces of the design space, where the LHS may fail in placing them. Due to the large minimum distance, these locations may have been characterized by a greater-than-average uncertainty, exploited by the EI_{feas}.

For few parameters, such as $s_{1_{low}}$, Δz_{low} and Δx_{mid}, instead, a significant portion of samples is added inside their range of variation. This suggests that these parameters, opposed to the rest, more actively shape the feasible region boundary.

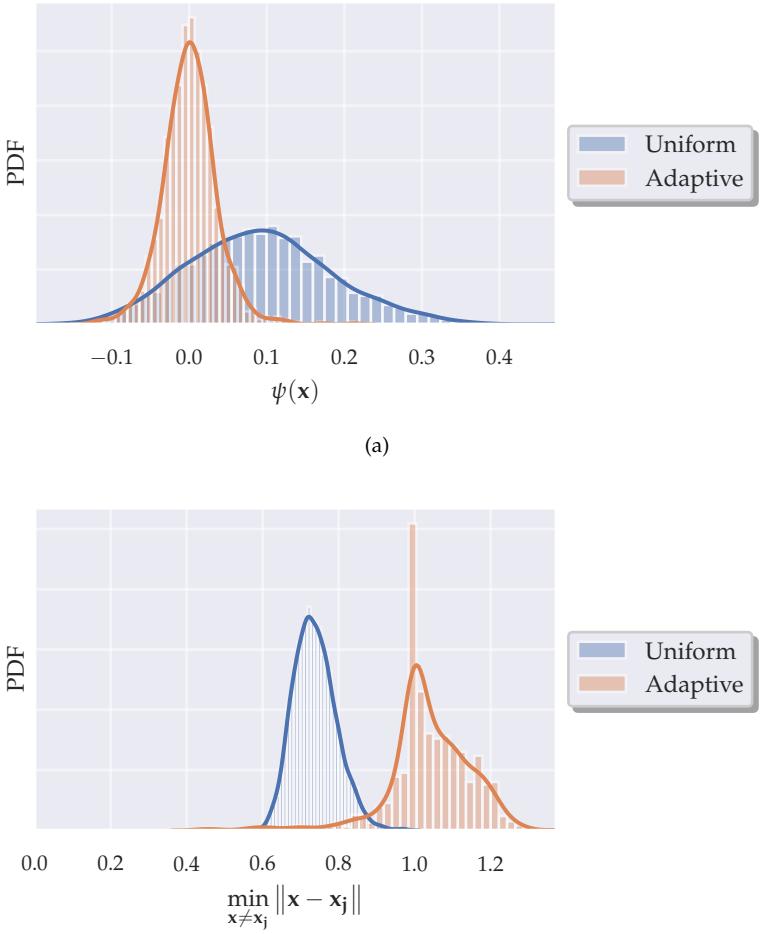


Figure 7.4: Probability Density Function (PDF) of:
(a) the feasibility function ψ,
(b) the distance from the closest sample
for uniform and adaptive phase of Bayesian feasibility analysis applied to the design problem in Section 7.1.

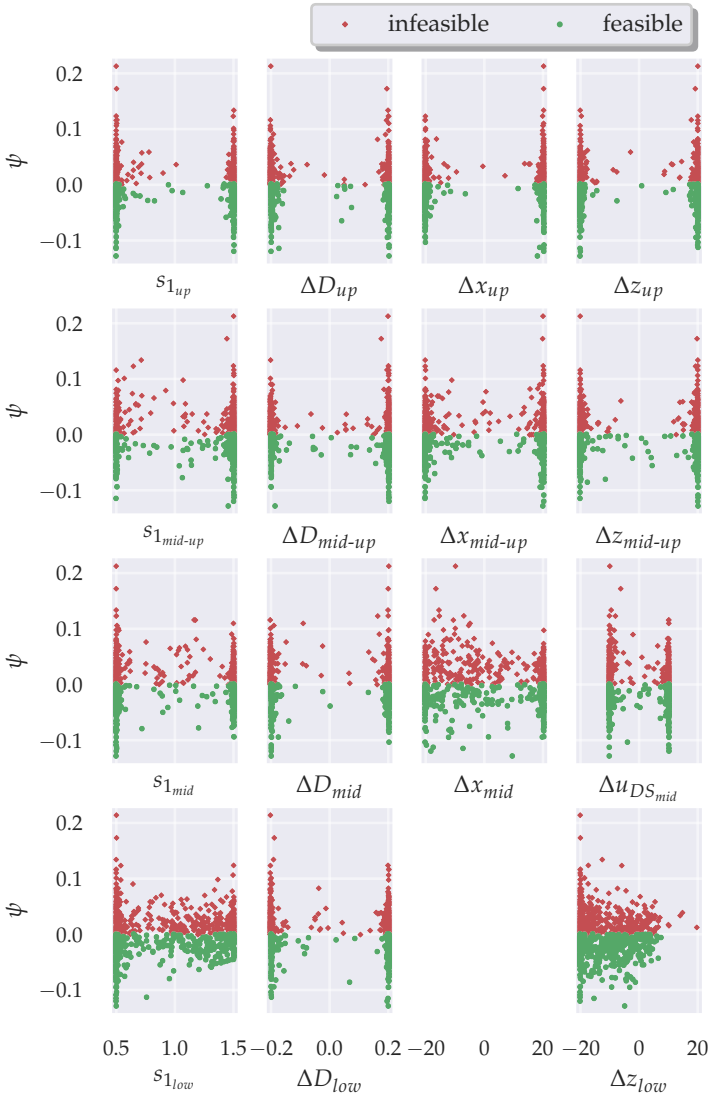


Figure 7.5: Scatter plot of the feasibility function ψ during the adaptive sampling phase for each design parameter for the medium-sized sedan in Figure 4.3b, with the aPLI, inside the design space in Table 7.1.

To conclude, the larger minimum distance of the samples from the adaptive phase may provide a further explanation for the non-improving cross-validation scores. In fact, being located at further distances, the new samples are more critical for cross-validation than the uniform ones. The new samples are, though, not used to test the surrogate model after the uniform phase.

7.3 FLEXIBILITY BOXES

Finally, I compute the flexibility boxes with the procedure described in Section 5.2.6. I look for the three maximal feasible boxes with highest flexibility level whose relative, pairwise overlap does not exceed 25% of their volume. This seems a small portion of volume that is allowed to overlap, yet in high-dimensions it becomes relatively significant: in 15D, as for the current design problem, each parameter is allowed to overlap, *on average*, for 91% of its range of variation. Still, overall three quarter of their volumes is different. I consider this sufficient to, on one hand, understand how the feasible region extends inside the design space, and on the other hand, obtain useful design guidelines –namely, that offer great design flexibility.

I compute the boxes with both volume-based and minimum-interval-based flexibility measures –introduced in Section 5.2.2– and highlight the differences between the two approaches.

7.3.1 *Volume-based*

Figure 7.6 shows the parallel coordinates plot of the volume-based flexibility boxes. In the same plot, the minimum axis-aligned box that bounds the feasible region is also depicted. The bounding box extends for the full range of variation of each parameter, except for Δz_{low}: no design is feasible when the position of the lower load region is too high, namely in the upper 20% of its range of variation.[8] Δz_{low} is, also, the design parameter whose range of flexibility is reduced the most in all flexibility boxes: the

8 This means that about 20% of the uniform samples were wasted, as they provided no information on the feasible region. Therefore, in order to improve the surrogate

	FRACTION OF DESIGN SPACE'S VOLUME
Minimum Bounding Box	78%
Feasible Region	19%
Volume-based Flexibility Box	1.5%
Min-Interval-based Flexibility Box	0.08%

Table 7.3: Approximate volume size relative to the design space's one.

smaller its range, the larger the ranges of the other parameters. Recalling the scatter plots in Section 7.1.1, Δz_{low} significantly influences all injury criteria with a same positive trend: the smaller, the better. We can state, then, that Δz_{low} is the most relevant parameter affecting the design flexibility of the medium-sized sedan in Figure 4.3b for the aPLI.

Box 1, the largest feasible box in the design space, extends for as little as 1.5% of the design space; as reported in Table 7.3, the feasible region extends, instead, for 19% of it.[9] This reduction by a factor 10 is due to the shape –hyper-rectangular– and orientation –axis-aligned– of the flexibility box, which enable the parameter values to be designed *independently* in their given ranges.

Box 1 restricts the interval of Δz_{low} to the bottom 15% of its total range of variation, that of $s1_{low}$ to its bottom 35% and that of Δx_{mid} to its top 60%. Hence, it prescribes a lower and more compliant lower load region and limits the forward shift of the middle load region. Other restrictions, such as those on $s1_{mid\text{-}up}$, $\Delta D_{mid\text{-}up}$, $\Delta x_{mid\text{-}up}$ and $\Delta z_{mid\text{-}up}$, are less severe. The rest of the design parameters can take, instead, almost any value inside their original range of variation.

model accuracy inside the feasible region, the investigation should be run again with a reduced Δz_{low} range of variation.

9 N.B.: The minimum bounding box corresponds to the minimum *circumscribable* axis-aligned box around the feasible region. On the contrary, the maximum feasible box corresponds to the maximum *inscribable* axis-aligned box inside the feasible region. The two volumes differ of a factor 50.

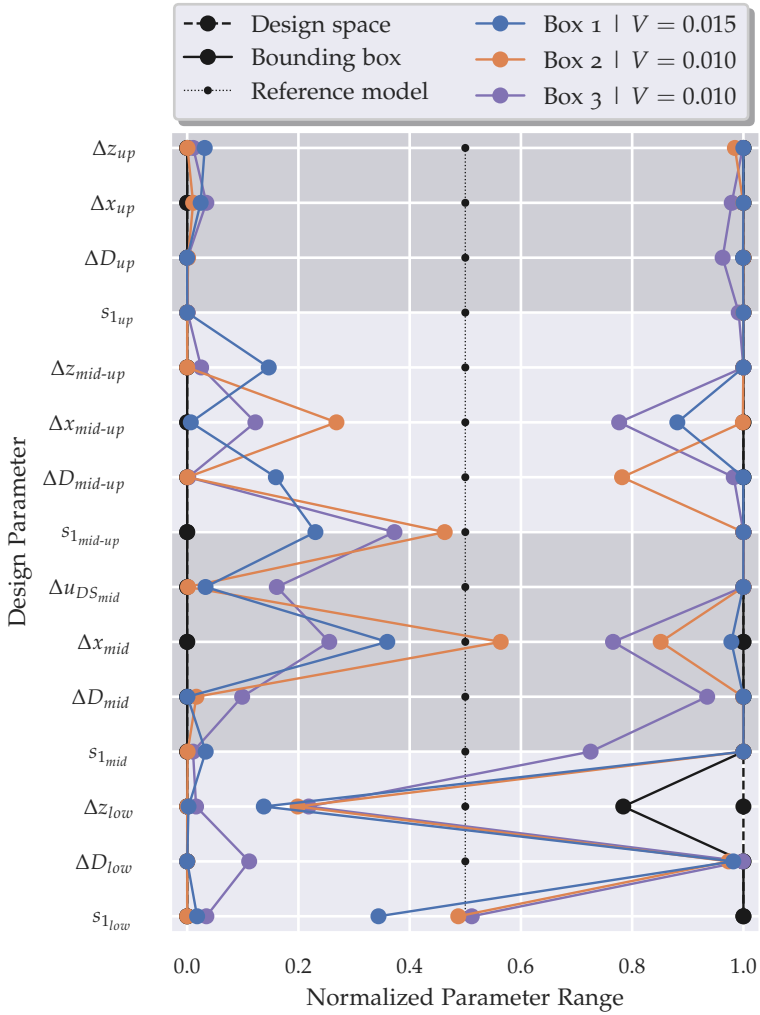


Figure 7.6: Parallel coordinates plot of volume-based flexibility boxes, for the medium-sized sedan in Figure 4.3b, with the aPLI, inside the design space in Table 7.1. The normalized volume is reported in the legend for each flexibility box.

Box 2 allows slightly more flexibility on Δz_{low} and $s_{1_{low}}$ –on both, about one third more– yet compromises the total box volume of one third with respect to Box 1. Strong interval restrictions concern Δx_{mid} –limited between 55 and 85% of its range– $s_{1_{mid\text{-}up}}$ – limited to the top 50%– $\Delta D_{mid\text{-}up}$ –limited to the bottom 80%– and $\Delta x_{mid\text{-}up}$ –limited to the top 75%. Hence, a higher and stiffer lower load region can be compensated by adjusting the longitudinal position of the middle load region and stiffness, dissipation and longitudinal position of the mid-up load region.

Box 3 allows about as much flexibility on Δz_{low} and $s_{1_{low}}$ and covers as much volume as Box 2. Yet, it compromises differently the other parameters. Noteworthy, $s_{1_{mid}}$ is limited to the bottom 70%, Δx_{mid} to the range 25/75% of variation, $\Delta u_{DS_{mid}}$ to its top 80% and $\Delta x_{mid\text{-}up}$ to the range 10/80%. Hence, a higher and stiffer lower load region can be also compensated by adjusting stiffness, longitudinal position and intrusion of the middle load region and the longitudinal position of the mid-up region.

In conclusion, volume-based flexibility boxes effectively indicate which parameters are the most-significant ones and should be carefully considered in the early phases of the development process. Not designing these parameters inside their recommended range may still lead to a feasible design, yet would significantly reduce the flexibility of other parameters. On the other hand, though, the parameter ranges prescribed by volume-based boxes may be very imbalanced, as in the present case: the flexibility of few parameters may be strongly restricted, while that of the majority almost unaffected. In fact, the volume-based flexibility measure does not allow to *weight* the parameters' flexibility, so as to reach a more uniform compromise.

7.3.2 *Minimum-Interval-based*

Maximizing the boxes' minimum interval, defined in Equation 5.39, leads to more cubically-shaped boxes. This metric requests as input a required minimum interval size: I choose, for each parameter, half its total range of variation. Besides providing a considerable degree of flexibility to each parameter, this size would also include

the reference model. In fact, this lies in the present investigation exactly at the middle of the range of variation.

Figure 7.7 shows the parallel coordinates plot of the minimum-interval-based flexibility boxes. None of them satisfies the required minimum interval size: the one with highest flexibility level reaches 96% of the required size. Similarly to the volume-based boxes, Δz_{low} is the most restricted parameter. Nevertheless, *all* parameters are somehow restricted here. This suggests that, although some parameters may not show a strong global tendency inside the design space or may be indeed less significant than others, they are still relevant in order to reach an adequate degree of design flexibility for all the parameters.

The reference model is not included in any boxes. It is, in fact, slightly infeasible.[10,11] However, as Box 1 and 2 suggest, it would be enough to lower the lower load region of few millimeters to obtain a feasible design. Box 3, instead, also slightly increases the stiffness of the mid-up load region.

All boxes roughly agree on the ranges of $s1_{low}$, Δz_{low}, $s1_{mid}$, Δx_{mid}, $s1_{mid\text{-}up}$, $\Delta x_{mid\text{-}up}$, $\Delta z_{mid\text{-}up}$ and Δz_{up}: lower and more compliant lower load region, middle load region with limited stiffness and limited longitudinal shift, stiffer mid-up load region with limited longitudinal and downward shift and upper load region with limited upward shift. It is crucial, therefore, to design these parameters inside the recommended ranges in the early stages of the development. For the remaining parameters, instead, the union of the interval of each box covers the whole range of variation. Hence, any value inside their total range of variation is recommended by at least one box. For these parameters, the design can be adjusted even at later stages, without significantly compromising the flexibility of the other ones.

Box 1 allows the highest flexibility level for $\Delta u_{DS_{mid}}$ among the three boxes. Box 2 restricts the most Δx_{up}, yet provides the largest range for $s1_{up}$, hinting that limiting both forward and backward shifts of the upper load region allows it to take a larger range of loading force-deformation characteristics. Box 3 offers the highest

10 It was, hence, predictable that the minimum interval could not satisfy the required size.

11 The reference model was not developed to fulfill aPLI's requirements.

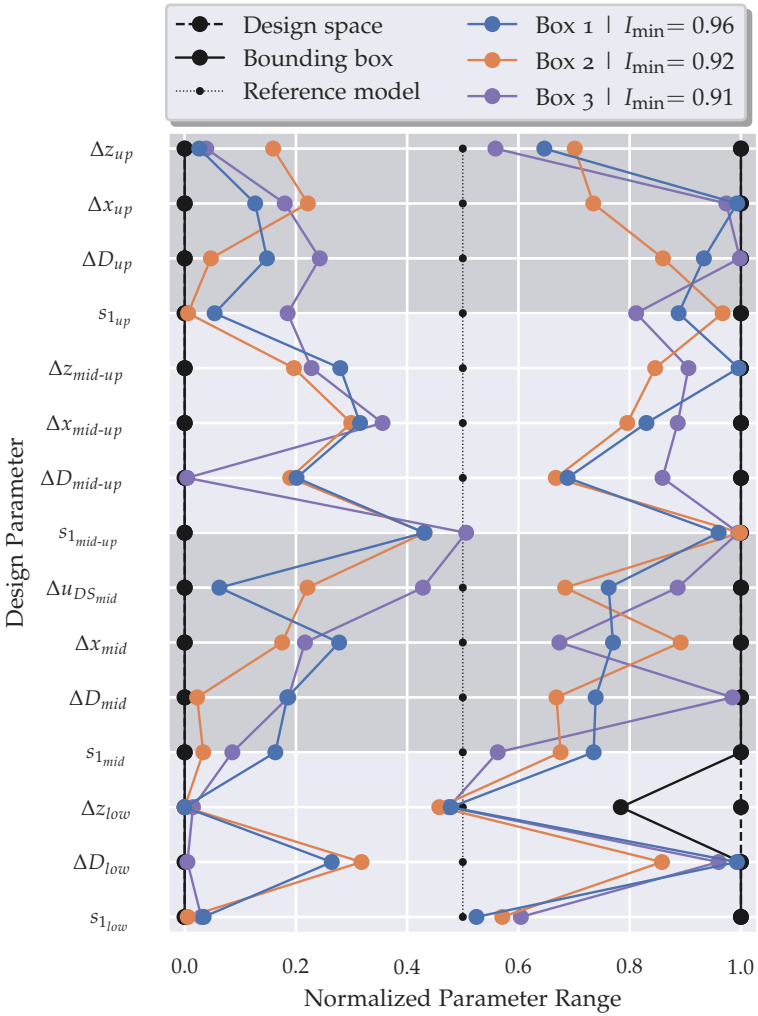


Figure 7.7: Parallel coordinates plot of minimum-interval-based flexibility boxes, for the medium-sized sedan in Figure 4.3b, with the aPLI, inside the design space in Table 7.1. The normalized minimum interval is reported in the legend for each flexibility box.

flexibility level for ΔD_{low}, ΔD_{mid} and $\Delta D_{mid\text{-}up}$, yet limits $s1_{mid}$, $s1_{mid\text{-}up}$ and $s1_{up}$ the most. Each box, hence, has its peculiarity and enables a different compromise among the parameters.

Figure 7.8 shows the parallel coordinates plot of the volume-based optimum flexibility box and the minimum-interval-based one. The difference in volume is substantial: the volume-based one is about 20 times larger than the minimum-interval-based one. Yet, the latter offers a great increase in flexibility of Δz_{low}: from 15% to 48% of its total range of variation. Therefore, as long as the vehicle's styling will not be fixed, the minimum-interval-based box may be a preferable solution. Also the range for $s1_{low}$ becomes larger, while all other intervals get significantly restricted. Noteworthy, the mid-up load region undergoes strong limitations, proving to be particularly crucial for aPLI. Furthermore, the design parameters of the upper load region are restricted in the minimum-interval-based flexibility box, but not in the volume-based one; especially, the reduced height of this load region seems important to counterbalance the increased height of the lower one.

The two boxes have also different probability of feasibility. Out of 200 samples drawn from an uniform distribution, 99% of them were feasible in the volume-based box, while only about 90% in the minimum-interval-based one. The first corresponds to a probability of feasibility greater than 96%, the second one to a probability greater than 85%, still acceptable. Furthermore, the infeasible designs lie here close to the feasible region and can, thus, be potentially modified during the development to become feasible. Figure 7.9 shows the PDF of the feasibility function for the infeasible samples inside the optimum minimum-interval-based flexibility box. Most samples lie below $\psi \leq 0.05$ and violate, hence, one requirement by at most 5%.

In conclusion, maximizing the minimum-interval measure leads to more uniform flexibility among the parameters, although this may implicate a smaller total flexibility –expressed by the volume. Uniformity is reached by compromising more parameters at the same time –in the present investigation, all of them– instead of significantly restricting only the most important ones. Changing the required minimum interval size allows to reach more targeted compromises. Therefore, this quantity can act as a flexibility

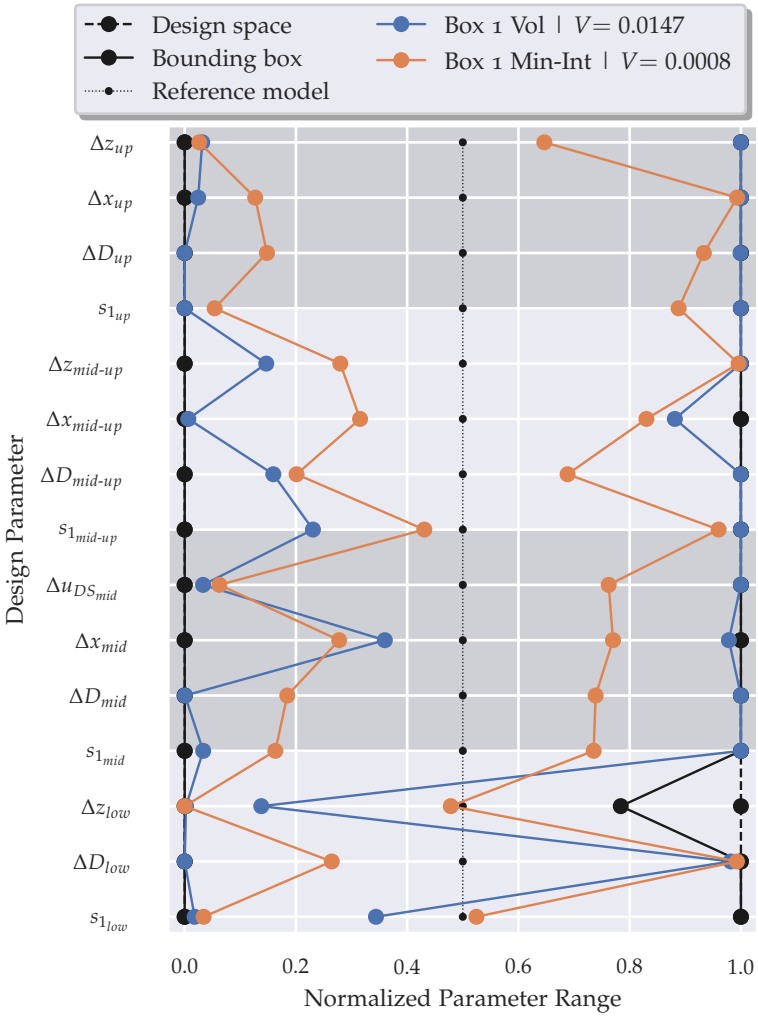


Figure 7.8: Parallel coordinates plot of optimum volume-based and minimum-interval-based flexibility boxes, for the medium-sized sedan in Figure 4.3b, with the aPLI, inside the design space in Table 7.1. The normalized volume is reported in the legend for each flexibility box.

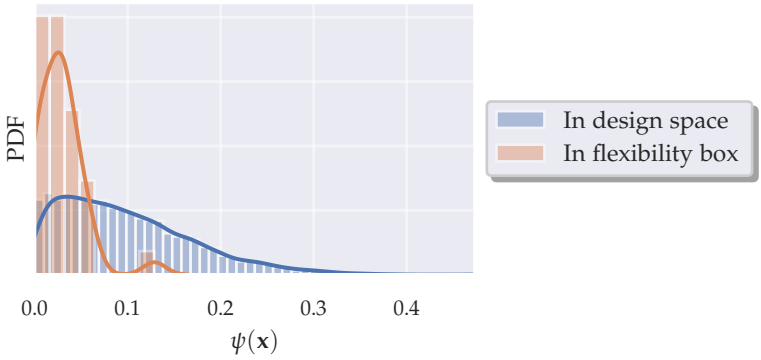


Figure 7.9: Probability Density Function (PDF) of the feasibility function ψ of the infeasible samples inside the optimum minimum-interval-based flexibility box and inside the design space in Table 7.1, for the medium-sized sedan in Figure 4.3b, with the aPLI.

weighting factor. As more parameters are restricted, the difference among the boxes –measured by the overlap– spreads over more ranges and becomes less evident. I recommend, in general, to use a lower allowed overlap with the minimum-interval measure than with the volume one in order to enforce a more tangible difference among the boxes.

7.4 COMPARISON APLI-FLEXPLI GUIDELINES

In Section 1.1, I introduced the aPLI as the new legform impactor that will replace the FlexPLI in C-NCAP consumer rating. However, the FlexPLI will remain –in the near future– the prescribed legform impactor for type approval. Therefore, the requirements for both impactors need to be satisfied: the number of injury criteria to be taken into account almost doubles and reaches a total of 15.[12] Satisfying the requirements of two different impactors raises further the complexity of the design process. In such circum-

12 FlexPLI's injury criteria include tibial bending moments, MCL, ACL and PCL.

stances, design guidelines that indicate feasible sets of parameter ranges become even more valuable.

If I was to compute the flexibility boxes fulfilling the requirements of both aPLI and FlexPLI during a real development process, I would consider the constraints all together and define a single feasibility function.[13] In the present work, though, I aim for a more academical investigation: to identify the differences in vehicle design between the two impactors. Therefore, I compare the resulting optimum flexibility box after applying the exact same procedure to both impactors. I use the Low-Fidelity Model (LFM) for the medium-sized sedan in Figure 4.3b[14] and the design space in Table 7.1.

Figure 7.10 shows the parallel coordinates plot of the optimum minimum-interval-based flexibility box with both aPLI and FlexPLI. The required minimum interval is set to half the total parameter range. As opposed to aPLI's box, this requirement is amply fulfilled by the box for FlexPLI, as its minimum interval is 30% larger than the required one. It comes at no surprise that the reference model is included in the box: it was actually developed to satisfy FlexPLI's requirements, hence the design is feasible.

Looking at the recommended ranges, the two boxes provide *contrasting guidelines* for the following parameters:

$s1_{low}$: FlexPLI benefits from a stiffening of the lower load region, while aPLI from its softening[15]; a light softening, up to 20%, of the current stiffness characteristics would fulfill both impactors' requirements.

ΔD_{low}: FlexPLI benefits from a more energetic rebound of the lower load region, while aPLI from a less energetic one; a light variation, up to +/- 10%, of the current fraction

13 Due to the increased number of constraints, I would also increase the number of trained surrogate models accordingly, in order to ensure a proper degree of accuracy.

14 The validation of a vehicle LFM –similar to the one proposed in this thesis– with the FlexPLI can be found in [89].

15 The guideline, clearly, depends on which injury criterion is most critical for the respective impactor in the reference model and *should not* be considered generally valid.

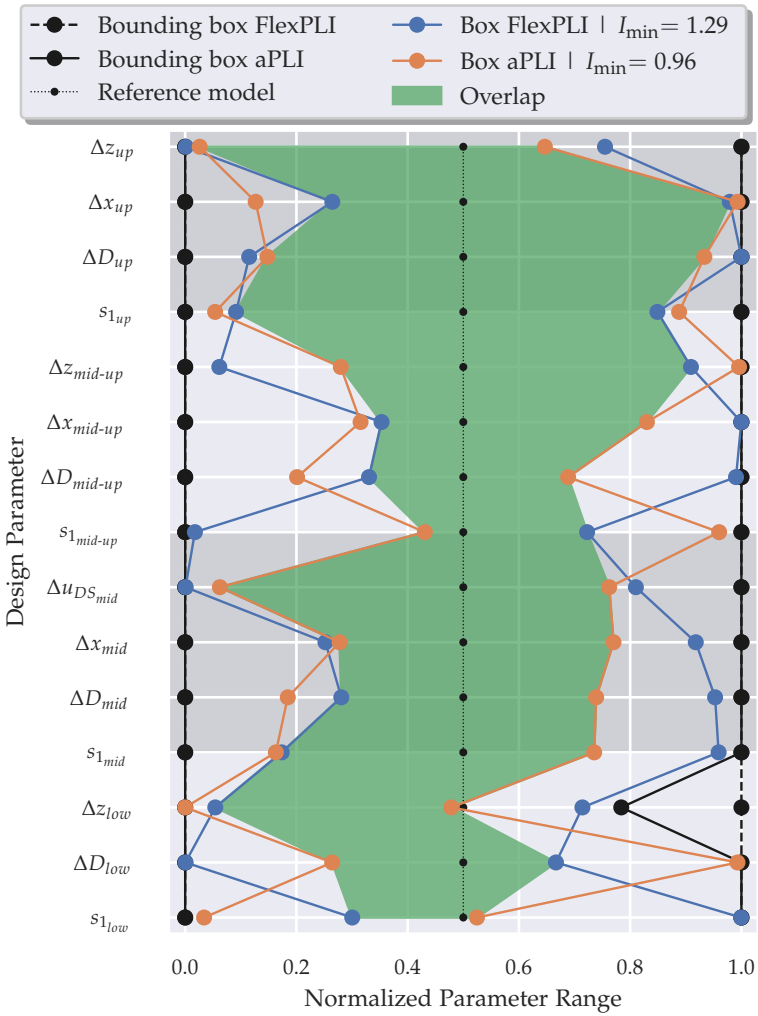


Figure 7.10: Parallel coordinates plot of the optimum minimum-interval-based flexibility box with aPLI and with FlexPLI, for the medium-sized sedan in Figure 4.3b, inside the design space in Table 7.1. The overlap between the two flexibility boxes is highlighted. The normalized minimum interval is reported in the legend for each flexibility box.

of energy recovered during unloading would fulfill both impactors' requirements.

$s1_{mid\text{-}up}$: FlexPLI benefits from a softening of the mid-up load region, while aPLI from its stiffening; a light stiffening, up to 20%, of the current stiffness characteristics would fulfill both impactors' requirements.

$\Delta D_{mid\text{-}up}$: FlexPLI benefits from a less energetic rebound of the mid-up load region, while aPLI from a rebound similar to the current one; a light variation, up to +/- 10%, of the current energy recovered during unloading would fulfill both impactors' requirements.

The guidelines for FlexPLI firmly favor its MCL elongation –as shown in Appendix B– while those for aPLI its tibial and femoral bending moments –as discussed in Section 7.1.1.

For the above parameter ranges, the overlap between the two boxes is lowest. For other parameter ranges, instead, the overlap is more significant and covers almost the whole recommended range for one box, but not for the other one. This means that the two boxes provide *similar guidelines* for these parameters, yet their flexibility is larger for one impactor than for the other. These parameters are: Δz_{low}, $s1_{mid}$, ΔD_{mid}, Δx_{mid}, $\Delta x_{mid\text{-}up}$, $\Delta z_{mid\text{-}up}$ and Δx_{up}. For all of them, the FlexPLI box offers more flexibility, except for Δx_{up}. Here, FlexPLI limits the forward translation of the bonnet more than aPLI. A rearward position of the bonnet leading edge with respect to the lower load region –so-called *soft-nose*– is a typical styling characteristic of modern cars, led by the fulfillment of FlexPLI's requirements. The flexibility boxes indicate that, for the investigated vehicle, a soft-nose is more relevant for FlexPLI than for aPLI.

Finally, for the remaining design parameters, the recommended ranges of variation are analogous for both boxes and display the largest overlap. Such parameters are: $\Delta u_{DS_{mid}}$, $s1_{up}$, ΔD_{up} and Δz_{up}. Hence, these parameters do not show particular differences in the influence they have on both impactors and the boxes provide quite *identical guidelines*.

In conclusion, the guidelines on the vehicle's front-end geometry are similar for both impactors with a medium-sized sedan.

Those on the stiffness of the lower and mid-up load regions are, instead, contrasting. This indicates that finding a good compromise for these parameters is crucial to satisfy both impactors' requirements. The guidelines for the bonnet are mostly identical for both impactors; yet, a soft-nose seems to be less relevant for aPLI than for FlexPLI. Not only the overlap in the upper load region is larger than in the other ones, the design flexibility of the parameters here is larger as well. This indicates that lower, mid and mid-up load regions are, in general, more critical than the upper one for both aPLI and FlexPLI. Lastly, aPLI offers, for the investigated vehicle, less overall front-end design flexibility than FlexPLI.

8 CRITICAL REFLECTION

An education isn't how much you have committed to memory, or even how much you know. It's being able to differentiate between what you do know and what you don't.

— William Feather[1]

With Chapter 7, the work presented in this thesis can be considered concluded: I proposed, validated and applied the method. I take now a step back and reflect on the work done, discussing the achievement of the single objectives set and advantages and disadvantages of the proposed approaches.

8.1 LOW-FIDELITY MODEL

Which low-fidelity modeling technique of the vehicle front-end is appropriate for the impact with the aPLI?

I link the vehicle outer surface to non-linear translational springs. The outer surface is split in several sections, each one modeled as a rigid body and attached to three orthogonal springs. The principal spring is oriented according to the recorded deformation.

The model is accurate enough for the intended use in this thesis: it shows generally an error below 10% in all investigated conditions, both in- and out-of-reference. Nevertheless, the modeling technique has advantages and disadvantages and I discuss them in the following.

ADVANTAGES

- The model consists of only few elements –less than 100 against HFM's 4 millions– and is, thus, very computationally efficient.

1 From: William Feather. *Telephony*. Volume 150. 1956.

- The model reproduces and condenses the macro-level of information exchanged by the vehicle during contact with the pedestrian leg impactor. This is achieved by the high degree of abstraction, as structural members are replaced by translational springs. The implementation of changes becomes, therefore, very easy and *precise*: it does not involve anymore complex geometrical and material properties of the front-end components, but rather *directly* the position or the force-deformation curve of the load levels.
- The model is automatically generated and sufficiently accurate with any vehicle type.
- The model is valid for both FlexPLI and aPLI.
- By means of clustering, the number of load levels can be chosen: 20 seems a reasonable choice.
- The use of a cubic interpolation of the force-deformation curve has little effect on the injury criteria. This is in agreement with Takahashi et al. [121], who, using an HBM, found that maximum force, maximum deformation and energy absorbed by a load level are important in order to ensure a good correlation with the HFM, while the actual shape of the curve plays a minor role.
- Completely new force-deformation curves may be defined, enabling great freedom for conceptual investigations.

DISADVANTAGES

- Replacing millions of DOFs with a bunch of them comes at a price: the model is tailored to *one specific load case* –pedestrian leg impact– and *one specific impactor kinematics*. Constructing the model on the basis of the *recorded* forces and deformations means losing information on the model performance in loading conditions different from the recorded one.
- As direct consequence of the point above:
  - No force-deformation characteristics is recorded for load levels that do not go in contact with the impactor.

  - The force-deformation curve must be extrapolated for deformations larger than the recorded one. I assume a linear positive extrapolation.

  These issues did not appear to affect significantly the model accuracy *in the investigated range* –except for the bonnet, that I will discuss separately. Surely, those issues contribute to limit the range of validity of the model. Possible solutions are the increase in kinetic energy or change of shooting position of the impactor. However, it is questionable whether the advantages of these solutions overcome the disadvantage of changing the reference loading condition. In fact, at the end, *one* force-deformation curve and one inclination must be defined for each spring. A different approach would be to obtain the characteristics individually for each load level with dedicated sub-impactors, as done in [51, 94]. However, this does not appear practical when the number of load levels is high. In fact, it would be difficult to isolate each single load level and avoid the interaction with the other ones. Furthermore, the loading condition of sub-impactors may not be representative of the actual one with aPLI.

- Out of the recorded data, a further simplification takes place in the mapping to one-dimensional elements –i. e., the translational springs. The force may be very non-linear[2], the deformation not. In fact, the model is only valid for small non-linear deformations. The set of three orthogonal springs does not build one three-dimensional system, $\mathbf{F} = \mathbf{F}(\mathbf{u_x}, \mathbf{u_y}, \mathbf{u_z})$, but three one-dimensional ones, $\mathbf{F_x} = \mathbf{F_x}(\mathbf{u_x})$, $\mathbf{F_y} = \mathbf{F_y}(\mathbf{u_y})$, $\mathbf{F_z} = \mathbf{F_z}(\mathbf{u_z})$, whose resultant is three-dimensional. This is based on the assumption that the deformation occurs mostly along one dimension and the other two prevent a locking effect.

- Most of the model inaccuracies were found to be related to the bonnet. Here, two issues arise:

  - The first load levels of the bonnet undergo re-loading during the unloading phase due to the rotation of the fe-

2 In the recorded range.

mur. In the HFM, the unloading, the re-loading and the re-unloading follow very different force-deformation curves. In the LFM, instead, the curve is the same. Furthermore, in the HFM, also the direction along which the deformation occurs, during loading and re-loading, is different. In the LFM, the deformation occurs mostly along the level's first principal component.

– Depending on the impactor's kinematics, the UBM hits the bonnet at a specific location. Hence, most of the forces are recorded here. If the UBM hits the bonnet at a different location in the LFM, the stiffness of the bonnet will most likely be underestimated.[3] Therefore, besides large variations of the model in general, which may cause significant changes in the impactor's kinematics, especially large geometrical variations of the bonnet should be avoided.

The issues on the bonnet are not trivial to be solved with a spring-based LFM. Other techniques for model's complexity reduction may be investigated. Nevertheless, due to the complex bonnet deformation, a sensible alternative would be to use directly the original, shell-based bonnet, without simplifying it.[4] The model would become a hybrid low-/high-fidelity one: low-fidelity lower, middle and mid-up load regions with a high-fidelity upper one. By slicing out the lateral parts of the bonnet, where no deformation takes place, I reckon the computational time of the hybrid model would remain reasonable and not significantly higher than the LFM's one. On the other hand, the accuracy –especially in out-of-reference conditions– could benefit significantly, and so, the range of validity of the model. Geometrical parameters for the upper load region could remain the same: rigid shifts of the entire bonnet in the xz-plane. Stiffness parameters should be reviewed.

3 There were no or little forces recorded here.

4 Former colleague Mößner [88] gave me once this wise suggestion: *"Do not spend too much time on the simplified model. It is a* simplified *model. The more time you will spend on it trying to gain 1-2% accuracy, the more complex you will make it."*

- Essentially, all the computational effort in an LFM simulation is associated with the impactor. In this work, I did not simplify the impactor, because I did not have to implement changes to it –and so, neither to facilitate their implementation. This thesis, in fact, focuses on front-end design, not design of the impactor. However, a great number of simulations is required by the proposed flexibility optimization work-flow, yet that number is limited by the sampling budget at disposal. Simplifying the aPLI could substantially increase the sampling budget. On the other hand, the accuracy of the impactor's response may deteriorate. Model Order Reduction (MOR) techniques, such as the one proposed in [4], seems promising for this task.

So, all in all, "*is the model illuminating and useful*?"[5] Definitely useful. Moderately illuminating, as it allows to run investigations that with the HFM would not be trivial to run; yet, the *illumination* may not always be reliable.

8.2 PARAMETRIZATION

> *Which parametrization approach is recommended for the representation of design variability and handling of complexity?*

I identify four load regions in the vehicle front-end –lower, middle, mid-up and upper– and define geometrical and structural parameters for each of them. Geometrical variations include rigid shifts of the load regions in the xz-plane and the deformation space in front of the bumper beam. From the structural side, I use scaling factors to vary the reference force-deformation curve of each level in the region during loading and dissipation factors during unloading.

ADVANTAGES

- The parameters are easy-to-understand and -to-communicate in the design process. Expressing the parameters in the form

5 See first paragraph of Section 4.2 for contextualization.

of relative variations to existing structures provides, in fact, the designers with a basis for orientation.

- The use of one or two scaling factors for the force-deformation characteristics offers still adequate flexibility: both the force level and the curve shape vary significantly. Additionally, the variations remain related to a real structure, where requirements from other disciplines are also taken into consideration. Finally, scaling factors allow to keep the number of parameters low, reducing the complexity of the parametrization. Defining, e. g., a brand-new cubic curve would require three points[6] and offer superfluous flexibility: the force-deformation characteristics of a bonnet cannot take any shape.

- Moving the load regions independent on another offers a simple solution to the problem of their positioning: the guidelines are meant for concept engineers, not stylists.

DISADVANTAGES

- The range of variation of the parameters should be limited in order to preserve the trustworthiness of the outcome. This is related to the LFM trustworthiness.[7] Being the impactor the *true* one –and not a simplification– the outcome of the simulation is accurate for any parameter choice. The question is, whether the LFM with that parameter choice can be realistically reproduced so in the HFM. In other words, the outcome of the simulation is correct; the *interpretation* of it may not be correct anymore.

- The validation of the parametrization is not straightforward. The parameters defined for the LFM cannot be reproduced so trivially in an automated way in the HFM. Structural members are replaced by non-linear springs in the LFM:

6 The fourth one being the origin.

7 Here, I use the word *accuracy* to refer to investigations for which there is a HFM model to be compared with, while the word *trustworthiness* when there is no corresponding HFM the model can be validated with. Hence, the LFM can only be *trusted*.

force-deformation characteristics are not easily correlated to actual dimensions of the structure they replace. Therefore, I validated the opposite problem: reproducing, with the LFM, changes happening in the HFM.

- Related to the point above, I did not investigate the possibility of an inverse parametrization: i. e., from the LFM to the HFM. The aim of this work was to compute design guidelines, not to implement them back in the detailed model. Nevertheless, that would be a logical further development of the proposed method: topology optimization may help finding the structure that, loaded under pedestrian leg impact, exhibits a force-deformation characteristics inside the prescribed corridor –such as the one in Figure 8.1.

8.3 FEASIBILITY ANALYSIS

Which method for the evaluation of design feasibility is appropriate with computationally-expensive black-box functions?

I exploit an established infill-criterion –the Expected Improvement for feasibility analysis– to improve the classification capability of a surrogate model between feasible and infeasible samples. I define a training work-flow that, first, uniformly samples the design space and, then, iteratively adds samples in the neediest locations. On the basis of the problem dimensionality, I set the available sampling budget. In all investigated test problems, the method satisfied the accuracy requirements.

ADVANTAGES

- The use of a surrogate model enables the evaluation of a very large number of samples in a reasonable period of time. This would not be possible with computationally-expensive functions, such as an LFM simulation, which takes minutes to run: a surrogate model prediction lies in the range of milliseconds.

- The proposed Bayesian feasibility analysis strategy is very effective in low-dimensions: adding samples in progressively critical locations greatly benefits feasibility analysis. The adaptive approach, in low-dimensions, consistently outperformed the completely-uniform one in terms of accuracy, both mean and standard deviation. This means that, over the iterations, the adaptive sampling phase neutralizes the randomness of the uniform sampling and regularly reaches the attainable degree of accuracy with the available sampling budget.
- The proposed strategy is also relatively efficient in low-dimensions: due to the moderate amount of samples, the computational effort to train the Gaussian process model remains acceptable.
- Training multiple surrogate models with the normalized constraints, instead of approximating directly the feasibility function, has a great, positive impact on the prediction accuracy. This, clearly, comes at a computational cost; yet, its value, at least for the Rosenbrock-based problem, proved to be much more significant than the adaptive strategy itself.
- Gaussian process model and EI for feasibility analysis perform well, as the selected samples at each iteration lie very close to the feasible region boundary.
- NMMSO algorithm performs well, as the selected samples –besides having high EI_{feas} values– lie far enough from one another.
- The accuracy of the trained Gaussian process model proved, in the Rosenbrock-based test problem, to be high enough to not significantly affect the accuracy of the flexibility boxes. Therefore, the accuracy requirement on the surrogate model seems satisfactory.

DISADVANTAGES

- The proposed strategy is not very effective in high-dimensions: for the Rosenbrock-based problem, the benefit in terms of ac-

curacy decreases as dimension increases and almost nullifies in 15D. This is not primarily related to the adaptive sampling approach itself, but rather to the proposed work-flow, and more particularly, to the *available sampling budget*[8]. The problem is not adding samples iteratively; rather, starting the iterations from a bad, inaccurate spot. As the strategies U60, U80, U100 show, accuracy does improve with higher sample size. In high-dimensions, though, the adaptive phase is not able to compensate for the smaller initial uniform sample size. This is confirmed by the better accuracy of U80/A20 over U60/A40 in high dimensions, despite the fewer iterations. Hence, a good starting point –i. e., an accurate surrogate model– is *crucial* to get the most out of an adaptive strategy. In high-dimensions, for the Rosenbrock problem, neither U60 nor U80 deliver a surrogate model accurate enough for an effective adaptive phase. In the work-flow proposed in Figure 5.8, I check the *global* surrogate model accuracy, $R^2 \geq 0.8$ after the uniform phase: I reckon now that that is not appropriate. The adaptive phase relies on the expected improvement for feasibility analysis, which values samples close to the feasible region boundary. *Here*, therefore locally, the surrogate model must have a minimum accuracy; globally is not relevant. Hence, I recommend to prove, after the uniform sampling phase, its local accuracy with cross-validation: e. g., $R^2_{0.1} \geq 0.5$.

- The proposed strategy is not efficient in high-dimensions either. High-dimensional spaces, in fact, require a large sampling scheme. This raises two issues:
  - Function evaluations, namely simulations, are computationally-expensive.
  - Training Gaussian processes becomes computationally-intensive when a large data set is used, e. g., 5000 samples.

8 This hypothesis is supported also by the increasing average distance among samples with the available sampling budget, as dimension increases.

Function evaluations are treated in Bayesian feasibility analysis as black-box functions. So, little can be done to improve the first issue. The second issue, instead, is related to the adaptive approach: the number of training iterations should be, hence, limited.

- As direct consequence of the above two points, the proposed strategy does not offer the best value for money –time– with dimensions larger that or equal to 12. It is worth noting that, the accuracy with an adaptive approach is still high and satisfies the requirement. However, about the same accuracy can be obtained with a completely-uniform strategy. The time saved on iterations could, actually, be invested to run more function evaluations, and so, to significantly improve accuracy.

- I did not precisely investigate:
  - how large the sampling budget should be for a completely-uniform strategy to achieve the same accuracy of the U60/A40 one in low-dimensions and compare the computational effort of both approaches;
  - how high the accuracy would be if the computational effort of the U80/A20 strategy would be used for a completely-uniform sampling in high-dimensions.

  Both studies would be interesting to analyze the value for money/time of adaptive approaches. However, they should be carried out with expensive-to-evaluate functions and not mathematical ones.

- Cross-validation does not return trustworthy results with adaptive sampling for feasibility analysis and relative metrics. The situation improves with absolute metrics. Still, function evaluations are computationally expensive and all samples should be used to train the model. Therefore, cross-validation should be preferred. However, it would be worth to investigate a more reliable cross-validation strategy in order to more precisely quantify the actual model accuracy.

- The proposed strategy uses a regression model for a classification task: feasibility analysis. It would be interesting to analyze the benefits of using directly a classifier for the task, at least at the end of the adaptive phase, for the final classification.
- The proposed work-flow does not make use of methods to filter out non-relevant parameters, such as feature selection. However, this would solve, quite pragmatically, the limits of the current strategy in high-dimensions: simply not working in high-dimensions in the first place. The curse of dimensionality is a renowned problem and should be avoided: *dura lex, sed lex.*

8.4 TYPE OF GUIDELINES

Which type of guideline is recommended to support the design process?

I formulate a problem that aims to find axis-aligned feasible boxes in the design space, which offer, on the one hand, a high degree of flexibility and, on the other hand, different solutions to the feasibility problem.

ADVANTAGES

- Being axis-aligned, the boxes prescribe independent intervals of values for each design parameter. This means that feasibility is ensured as long as each parameter is designed inside its prescribed interval, without consideration of its exact value.
- Maximizing the flexibility level, the boxes offer room for uncertainties in component design, manufacturing tolerances, requirements from other disciplines, etc. that may occur during the development process.

- Limiting the maximum overlap, the boxes propose different, equally-valid[9] solutions and provide valuable information on the extension of the feasible region.
- The volume-based flexibility measure requires no normalization of the design parameters –hence, no user input– and points out which parameters are most relevant for feasibility analysis.
- The minimum-interval-based flexibility measure allows to define the compromise among design parameters that the user seeks. In this way, engineers' experience can directly flow into the problem formulation.
- Searching for maximal feasible boxes allows to obtain still the largest box under the same minimum interval.

DISADVANTAGES

- The maximum overlap between boxes is affected by the problem dimensionality: boxes, where each interval overlaps for 70%, overlap overall for about 50% in 2D and for about 0.5% in 15D. I recommend, hence, to scale the maximum allowed overlap on the basis of the dimension in order to obtain really diverse solutions across most dimensions: e. g., $q_{max} = 0.8^d$.
- The volume-based flexibility measure, not imposing any aspect ratio, delivers extremely unbalanced results in case the relevance of the design parameters to the feasibility problem is unbalanced.
- The minimum-interval-based flexibility measure, favoring a precise aspect ratio, may exclude a big portion of the feasible region to reach the desired compromise.

8.5 OPTIMIZATION OF FLEXIBILITY BOXES

Which optimization procedure is appropriate to maximize the flexibility level of the guidelines?

9 Apart from the level of design flexibility.

I propose a stochastic optimization algorithm that, based on an extensive uniform sampling of the box bounding the feasible region, implements an incomplete, depth-first search strategy to collect all maximal feasible boxes and choose the desired ones.

Out of the investigated test problems, the strategy satisfied the probability of feasibility requirement in all dimensions of the hyper-sphere problem and in the real pedestrian leg impact example in 15D. Yet, for the Rosenbrock-based problem, that presents two distinct feasible regions, accuracy was slightly below the threshold in 12D and 15D with the default sample size.

ADVANTAGES

- Sampling the bounding box only *once* and running the optimization algorithm on that allows to find more than one flexibility box at the same time. Furthermore, it is less likely to fall into a local optimum than by zooming into a specific area with an iterative sampling approach.
- The tree-search enhances the optimization of the flexibility boxes. When the heuristics for the shrinking direction fails, a tree-search strategy gives the chance to correct the mistake. In 2D the error of the left-most path is negligible; for higher problem dimensionalities, instead, the error becomes remarkable. In 8D, correcting up to 5 mistakes along the tree-search path, the sum of volumes of the largest 50 maximal boxes increases of about 80%.

DISADVANTAGES

- The probability of feasibility inside the flexibility boxes decreases significantly in high-dimensions. The surrogate model inaccuracy proved here not to play a role. Hence, the inaccuracy is solely due to the limited amount of samples in high-dimensions: no information on feasibility lies in non-sampled areas. Therefore, in general, in order to improve the probability of feasibility, the sampling budget must be increased. In cases where two or more feasible regions are present, as in the Rosenbrock-based test problem, a significant boost in accuracy would come from the generation of

one bounding box per each one of them. Each bounding box would, then, be individually optimized. Since the number and location of separate feasible regions cannot be known a-priori, cluster analysis may be used to solve this task, as done for instance in [37].

- The tree-search strategy becomes very computationally intensive with large sample size and branching factor. With the proposed work-flow, this occurs in 12/15D. A time limit can effectively stop the exploration, ultimately exploring only the state-of-the-art left-most path.
- *One-shot* sampling approaches tend not to be suitable when the feasible region is small with respect to the design space [141]. Because of that, I find first the box bounding the feasible region. In this way, the samples are as concentrated as possible in the feasible region. The design space should not be, anyway, much larger than the bounding box, since, otherwise, also the surrogate model accuracy around the feasible region may be too low.
- As side-effect of the optimization procedure, vertices of the flexibility boxes are, often, infeasible. On the one hand, this generates much larger flexibility boxes. On the other hand, this compromises the accuracy. The test problems show that, in low-dimensions, the first effect predominates. In high-dimensions, instead, larger portions of the flexibility boxes lie in the infeasible region and the probability of feasibility drops accordingly. Infeasible samples here, usually, do not experience large feasibility function values and can, thus, potentially be turned feasible during the design process. Nevertheless, concept engineers –the recipients of the guidelines– should be made aware that the limits of the intervals are, in general, troubling.

All in all, the proposed one is a satisfactory *bonization* procedure[10]: the algorithm does not return the *true* feasible boxes with highest flexibility level, still the boxes present a reliable degree of flexibility and feasibility.

10 See introduction of Chapter 5 for contextualization.

8.6 RECOMMENDED RANGES

Which range of values is recommended for front-end parameters to fulfill the pedestrian leg impact requirements?

I applied the entire proposed method to a real design problem: I computed design guidelines for the front-end of a medium-sized sedan under consideration of the pedestrian leg impact requirements with both aPLI and FlexPLI. Furthermore, I compared both the volume-based and the minimum-interval-based flexibility boxes with the aPLI.

ADVANTAGES

- The volume-based boxes, clearly, point out which the most relevant parameters are for the fulfillment of the aPLI requirements, for the investigated vehicle: the stiffness and vertical position of the lower load level and the longitudinal position of the middle one. Although not reported in this thesis, the rank of the parameters that are restricted the most in the largest flexibility box, interestingly, matches the rank obtained with Sobol's sensitivity indices [115].[11]

- The minimum-interval-based boxes make a more uniform compromise among the parameters: more flexibility is given to stiffness and vertical position of the lower load level, less flexibility to all other parameters. By defining the required minimum interval, the flexibility measure indicates also whether the required flexibility has been achieved.

- Some parameters show equal ranges among all flexibility boxes. These ranges, therefore, represent those that should be carefully considered in the early phases, as flexibility may drop significantly outside. For the remaining parameters, instead, the boxes show that different solutions over the

11 N.B.: Although the interesting analogy, the two ranks intrinsically measure different things: the maximum feasible box reveals the most relevant parameters for feasibility analysis, the Sobol's sensitivity indices the most relevant parameters globally.

entire original range of variation are possible, without major consequences on the flexibility level.

- Out of the prescribed ranges of stiffness variation, a force-deformation corridor for each load region can be identified. Figure 8.1 shows the corridors obtained with the optimum minimum-interval-based box for aPLI in Section 7.3.2, together with the reference curve. This plot represents a clear prescription for the designers, where the design should lie.

- The boxes are very useful to identify the differences in design guideline between aPLI and FlexPLI: contrasting guidelines are, for the vehicle under study, stiffness and dissipation factor of the lower and mid-up load regions.

- The boxes are very useful also in later development stages, when the design is almost finalized, to understand how much tolerance to uncertainties each load region has and which the most critical one is.

DISADVANTAGES

- The design guidelines are specific to the reference HFM used to generate the LFM and do not have general validity. This is an advantage in the development of a vehicle project –because the indications are precisely tailored to that project– and a disadvantage if general guidelines are sought.

- The guidelines are only limited to the original range of variation. This should be kept relatively small to not compromise the model trustworthiness. Large geometrical variations should be, hence, avoided: an LFM based on a sedan should not be used to investigate an SUV.

- In the minimum-interval-based boxes, some parameters are slightly restricted although showing no or little relevance in any scatter plot. It is questionable, whether the restriction comes from the actual shape of the feasible region or from the limited amount of samples, since the boxes are only tenths of percent the size of the bounding box. Excluding

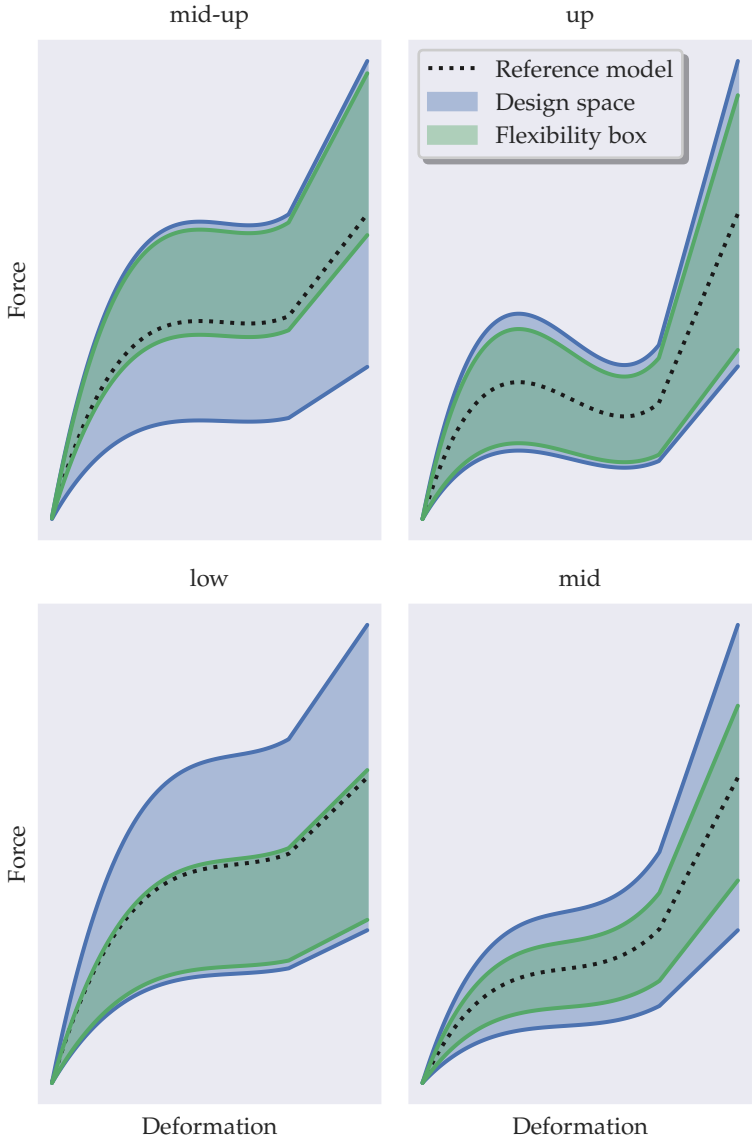


Figure 8.1: Force-deformation corridor for each load region of the medium-sized sedan in Figure 4.3b, according to the optimum minimum-interval-based flexibility box with aPLI, reported in Chapter 7.

non-significant parameters from the investigation would be very advantageous to reduce the curse of dimensionality and, in turn, improve this issue.

- About 99.6% of the feasible region is not included in the optimum minimum-interval-based box for aPLI. In other words, only the *0.4%* of the designs fulfilling the requirements are contained in the flexibility box.[12] It is clear, then, that the guideline communicates only a tiny, tiny portion of the actual possible solutions. This finds three explanations:
  - the box-like shape,
  - the axis-parallelism of the box,
  - the aspect ratio of the box, promoted by the defined minimum intervals.

  Without any restriction on the aspect ratio, the largest feasible box includes about 7.9% of the feasible designs: a significant increase from 0.4%, yet still a small portion. Therefore, most of the feasible designs are excluded because of the box-like shape and/or axis-parallelism. Dropping axis-parallelism completely would couple all parameters. The approach proposed in [24] seems more promising: coupling pairs of parameters. This could be done by either finding a tilted rectangle for each pair [46] or, more general, a polygon [47]: the *box*[13] would become a product of polygons, rather than of rectangles. The current parametrization would also be suitable to a pair-wise coupling: for each load region, the scaling factor could be coupled with the dissipation factor and the longitudinal shift with the vertical one. The intra-pair dependencies could be, easily, displayed on a 2D Cartesian plot, while the inter-pair independencies still with a parallel coordinates plot.

12 Actually, not all samples inside this flexibility box are feasible. The probability of feasibility is only greater than or equal to 85% at 97.5% confidence level.

13 It would be, actually, a *polytope*, not a box anymore.

9 CONCLUSION

A work is never completed
–a word which for them is meaningless–
but abandoned.

— Paul Valéry[1]

9.1 SUMMARY OF FINDINGS

In this thesis, I propose an answer to the following question:

> *Which method is appropriate to compute design guidelines for pedestrian leg impact on the basis of a low-fidelity vehicle front-end model?*

LOW-FIDELITY MODEL The method starts with the generation of a Low-Fidelity Model (LFM) of the vehicle front-end. I introduce a very computationally-efficient model for the impact with the advanced Pedestrian Legform Impactor (aPLI) and validate it for different vehicle types and shooting positions. The structural properties of the model are obtained from a reference High-Fidelity Model (HFM). I use the original front-end outer surface for contact representation with the impactor and nonlinear translational springs for stiffness representation. The outer surface is split in several load levels, which I then cluster down to about 20 in order to be representative of the real components they substitute. Out of these load levels, I identify four load regions to apply structural changes to. I define a wide range of geometry- and stiffness-related parameters, which rigidly translate the load regions, limit the deformation space in front of the bumper beam,

1 In: *La Nouvelle Revue française*. 1933.

scale the original loading force-deformation curve and vary the energy dissipated during the unloading phase. The model is about 17 times faster than the correspondent HFM and exhibits less than 10% error in all investigated conditions. Nevertheless, the modeling of the bonnet raises some challenges. Furthermore, the range of the parameters should be limited in order to preserve the model trustworthiness.

GUIDELINES The LFM is used to generate guidelines for designers of the vehicle front-end on how to fulfill the pedestrian leg impact requirements. Guidelines are expressed in the form of an independent range of values for each design parameter. In the design space, the guidelines become axis-parallel hyper-rectangles, or boxes, inscribed in the feasible region. The computational efficiency of the LFM is still not high enough to collect all samples that are necessary to compute those boxes. Hence, first, I train a surrogate model for feasibility analysis.

BAYESIAN FEASIBILITY ANALYSIS I adopt a Bayesian approach in the sense that I use a Gaussian Process (GP) to build a prior distribution and derive the posterior one as observations are collected. The available sampling budget is split in two phases: an initial uniform sampling is followed by an iterative process, where batches of samples, maximizing the Expected Improvement for feasibility analysis, are added. A multi-modal optimization algorithm, the Niching Migratory Multi-Swarm Optimiser (NMMSO), is used to select the samples for the next iteration. I use cross-validation to quantify the model accuracy, although it proved to underestimate significantly the accuracy of adaptive strategies. I define a multi-dimensional test function that combines together two Rosenbrock functions. The proposed Bayesian feasibility analysis satisfies the accuracy requirements in every investigated dimension. Especially remarkable is the improvement on the standard deviation. Nevertheless, a tangible boost of accuracy is only noticeable up to 8D. Afterwards, completely-uniform and adaptive strategies are comparable, mostly because of the limited available budget in high-dimensions. Furthermore, GP training becomes very

computationally-expensive with large data sets, making adaptive strategies less suitable in high-dimensions.

DESIGN FLEXIBILITY OPTIMIZATION The final phase of the proposed method involves the design flexibility optimization of black-box functions. I formulate a problem to find multiple boxes with highest flexibility level and limited relative overlap. The constraint on the relative overlap forces the boxes to explore different areas of the feasible region. I introduce a stochastic optimization procedure to solve this problem: it, first, extensively samples the bounding box and, then, runs a discrepancy-based tree-search strategy to shrink the bounding box to the maximal feasible box for each feasible sample. The tree-traversal strategy allows to correct early mistakes along the box-finding path and proves to be very beneficial as the number of branches increases. Nevertheless, the computational effort becomes prohibitive in high-dimensions. Hence, I stop the iterative search as a time limit is reached. I recommend the use of a minimum-interval-based flexibility measure over a volume-based one in case the parameters' significance is very unbalanced. The introduction of the required minimum intervals allows to favor a particular aspect ratio during the box optimization, based on engineering experience. I verify the boxes' feasibility with a state-of-the-art probabilistic approach. The validation shows negligible influence of the surrogate model inaccuracy on the boxes' probability of feasibility. This is, instead, mostly related to the sparseness of samples and becomes especially troubling in high-dimensional spaces. Boxes' vertices lie often in the infeasible region and should be avoided.

PEDESTRIAN LEG IMPACT I applied the method to a real-world industrial problem of vehicle front-end design for pedestrian leg impact. The stiffness and vertical position of the lower load region and the relative longitudinal position between lower and middle load regions were identified as the most significant parameters for the investigated vehicle and the aPLI. Defining a required minimum-interval for each parameter allowed to obtain a more uniform allocation of design flexibility among the parameters.

Stiffness and dissipation factor of the mid-up load region, relative longitudinal position between lower and mid-up regions and height of the bonnet turned out to also play a major role in the design. Furthermore, I compared the design guidelines with aPLI and with FlexPLI: for the investigated vehicle, contrasting guidelines were obtained for the loading and unloading characteristics of lower and mid-up load regions. Still, a compromise is possible and is obtained by intersection of the respective flexibility boxes.

9.2 SO WHAT?

So...

- the LFM provides the designer with a tool that facilitates extremely the implementation of structural changes and accelerates their evaluation;
- the parameters are easy-to-understand and define general, conceptual modifications, as they act on the macro-structural behavior of the load regions[2];
- adaptive strategies are very effective to improve the accuracy of the approximation in targeted regions in low-dimensions;
- the method generates guidelines for the designers on the structural properties that the front-end should have in order to fulfill the pedestrian leg impact requirements;
- flexibility boxes prescribe the exact range of values that each parameter may assume so that its design does not depend on any other parameters;
- understanding the flexibility of each parameter in the early phases allows to timely recognize eventual bottle-necks in the development process;
- providing the concept engineers with multiple compromises on how to satisfy the functional requirements support them with know-how, which can, in turn, guide future trial-and-error, ultimately minimizing it;

2 As opposed to micro-structural properties, such as the thickness of a stiffening rib.

- the LFM generation and the computation of guidelines, being mostly automatized, can become a standard procedure during the development process, to be run regularly;
- flexibility-oriented design guidelines are suitable to both concept and series development phases:

  CONCEPT PHASE: to identify the parameters to focus on and how to design them,

  SERIES PHASE: to quantify the parameters' tolerance to uncertainties, and so, the robustness of the design;
- the method has successfully been validated on industrial applications.

9.3 OUTLOOK

This work is finished, yet several questions remain unanswered:[3]

- Which is the most appropriate low-fidelity modeling technique for the bonnet under pedestrian leg impact loading? Or would it be an option to use directly the high-fidelity model of the bonnet, giving rise to a hybrid low-/high-fidelity model of the front-end?
- Which is the most appropriate procedure to automatically generate components whose force-deformation characteristics under pedestrian leg impact loading lies in the prescribed corridor? Would topology optimization be an option?
- Which is the most efficient method for feasibility analysis in high-dimensions?
- Which is the most reliable cross-validation strategy when a large portion of samples has similar function values?
- Which is the most appropriate technique to identify the least relevant parameters for feasibility analysis, that may

3 For further contextualization, see Chapter 8.

assume quite any value in the design space and may be, hence, excluded by the flexibility optimization procedure?

- Which is the most appropriate technique to identify separate feasible region and construct a bounding box for each of them?
- How larger would the flexibility polytopes be, if pairs of parameters would be coupled in tilted rectangles? And how larger, if coupled in polygons?

I am convinced that revealing the answer to these questions will push the method significantly further. In the worst case, it will help to discover and to define better questions.

I think solutions come through evolution.
It comes through asking the right question,
because the answer pre-exists.
But it's the question that we have [...] to discover and to define.
[...] If you think of David and Michelangelo:
it was in the stone. But it had to be unveiled and revealed.
You don't invent the answer. You reveal the answer.

— Jonas Salk[4]

4 During an interview in Bill Moyers's television series *A World of Ideas*, broadcast on January 17, 1990.
Dr. Jonas Salk is best known for having developed the first successful polio vaccine, announced on April 12, 1955.
I report a larger excerpt of the interview:

BILL MOYERS I read the other day [. . .] that by the year 2000, which is not very far from now, there will be some 20 million people in the world carrying the AIDS virus. Is that a comparable challenge to what you faced with polio 50 years ago?

DR. JONAS SALK Well, it's an even more difficult challenge, [. . .]. The virus, if it prevails, then we will lose. But if we are able to reduce the damage caused by the virus and, at the same time, try to enhance the immune response to the virus and establish a more favorable balance between the two, then we will be doing in relation to that problem what we want to do in relation to the world and that is to reduce the negative and enhance the positive at one and the same time.

MOYERS The good news would be that there is a vaccine that protects us and immunizes us, against the AIDS virus. Are we going to have that good news, do you think, in your time and mine?

SALK My expectation is that we will solve the problem. It's just a matter of time and just a matter of strategy. Now, why do I say that this is the case? It's because I think solutions come through evolution. It comes through asking the right question, because the answer pre-exists. But it's the question that we have to define and discover, to discover and to define.

MOYERS You mean, when you asked the question about how to defeat polio, the answer was already there?

SALK Mm-hmm, in a way. If you think of David and Michelangelo: it was in the stone. But it had to be unveiled and revealed. You don't invent the answer. You reveal the answer.

MOYERS From nature.

SALK From nature.

Part IV

THE APPENDIX.

A

DERIVATION OF THE EXPECTED IMPROVEMENT

A.1 EXPECTED IMPROVEMENT FOR GLOBAL OPTIMIZATION

Let I be the improvement for global optimization, as defined in Equation 5.12.

Let z be a random variable with a standard normal distribution, i. e., zero mean and unitary standard deviation such that:

$$z = \frac{y - \hat{y}}{\sigma} \qquad f'_{\min} = \frac{f_{\min} - \hat{y}}{\sigma} \tag{A.1}$$

where $f'_{\min}$ is the best observed value of the variable z.

I can be, therefore, rewritten as:

$$I = \begin{cases} f'_{\min} - z & \text{if } z < f'_{\min} \\ 0 & \text{otherwise} \end{cases} = \max\left(0, f'_{\min} - z\right) \tag{A.2}$$

Accordingly, EI for global optimization can be computed as:

$$\begin{aligned} EI &= \mathbb{E}\left[\max\left(0, f'_{\min} - z\right)\right] = \\ &= \int_{-\infty}^{f'_{\min}} \sigma\left(f'_{\min} - z\right) \varphi\left(z\right) dz = \\ &= \sigma f'_{\min} \int_{-\infty}^{f'_{\min}} \varphi\left(z\right) dz - \sigma \int_{-\infty}^{f'_{\min}} z\varphi\left(z\right) dz = \\ &= \sigma f'_{\min} \Phi\left(f'_{\min}\right) - \sigma \int_{-\infty}^{f'_{\min}} z \frac{\exp\left(-\left(z\right)^2 \Big/ 2\right)}{\sqrt{2\pi}} dz = \\ &= \sigma f'_{\min} \Phi\left(f'_{\min}\right) + \sigma \left[\frac{\exp\left(-\left(z\right)^2 \Big/ 2\right)}{\sqrt{2\pi}}\right]_{-\infty}^{f'_{\min}} = \\ &= \sigma f'_{\min} \Phi\left(f'_{\min}\right) + \sigma \frac{\exp\left(-\left(f'_{\min}\right)^2 \Big/ 2\right)}{\sqrt{2\pi}} = \end{aligned} \tag{A.3}$$

$$= (f_{\min} - \hat{y})\,\Phi\left(\frac{f_{\min} - \hat{y}}{\sigma}\right) + \sigma\varphi\left(\frac{f_{\min} - \hat{y}}{\sigma}\right)$$

A.2 EXPECTED IMPROVEMENT FOR FEASIBILITY ANALYSIS

Let I_{feas} be the improvement for feasibility analysis, as defined in Equation 5.14.

Let z be a random variable with a standard normal distribution, i. e., zero mean and unitary standard deviation such that:

$$z = \frac{y - \hat{y}}{\sigma} \qquad z_{0'} = \frac{0 - \hat{y}}{\sigma} = -\frac{\hat{y}}{\sigma} \tag{A.4}$$

where $z_{0'}$ is the feasibility threshold of the variable z.

I_{feas} can be, therefore, rewritten as:

$$I_{feas} = \begin{cases} -\sigma z & \text{if } z \leq z_{0'} < 0 \\ \sigma z & \text{if } 0 \leq z_{0'} \leq z \\ 0 & \text{otherwise} \end{cases} \tag{A.5}$$

Accordingly, for the case z is predicted as infeasible, $z_{0'} < 0$, equivalent to $0 < \hat{y}$ in Equation 5.14, EI for feasibility analysis can be computed as:

$$\begin{aligned} EI_{feas} &= \mathbb{E}\left[-\sigma z\right] = \\ &= \int_{-\infty}^{z_{0'}} -\sigma z \varphi\left(z\right)\, dz = \\ &= -\sigma \int_{-\infty}^{z_{0'}} z \frac{\exp\left(-\left(z\right)^2 \Big/ 2\right)}{\sqrt{2\pi}}\, dz = \\ &= \sigma \left[\frac{\exp\left(-\left(z\right)^2 \Big/ 2\right)}{\sqrt{2\pi}}\right]_{-\infty}^{z_{0'}} = \\ &= \sigma \frac{\exp\left(-\left(z_{0'}\right)^2 \Big/ 2\right)}{\sqrt{2\pi}} = \\ &= \sigma \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\hat{y}}{\sigma}\right)^2\right) = \end{aligned} \tag{A.6}$$

$$= \sigma\varphi\left(\frac{\hat{y}}{\sigma}\right)$$

Similar steps can be taken for the case z is predicted as feasible, $0 \leq z_{0'}$, equivalent to $\hat{y} \leq 0$ in Equation 5.14, ending with the same result.

B

UNDERSTANDING THE FLEXIBLE PEDESTRIAN LEGFORM IMPACTOR

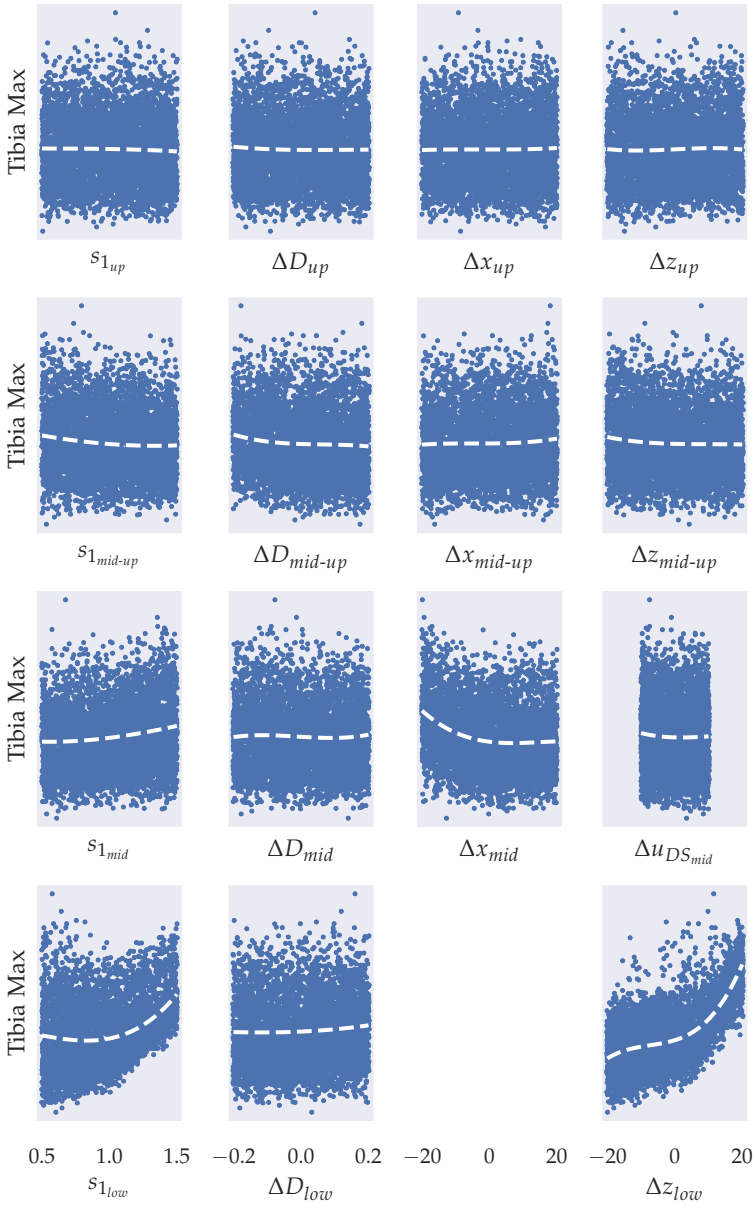


Figure B.1: Scatter plot of Tibia Max and cubic polynomial fit –dashed– for each design parameter for the medium-sized sedan in Figure 4.3b, with the FlexPLI, inside the design space in Table 7.1.

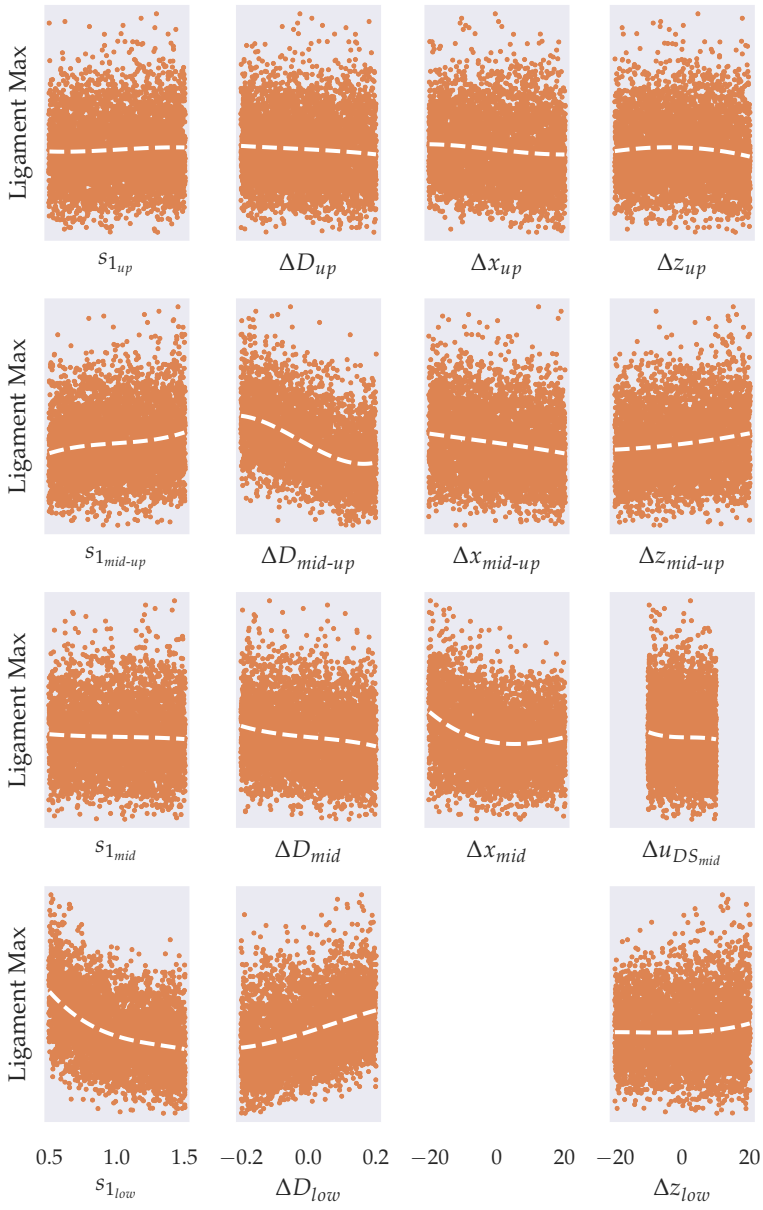


Figure B.2: Scatter plot of maximum normalized ligament and cubic polynomial fit –dashed– for each design parameter for the medium-sized sedan in Figure 4.3b, with the FlexPLI, inside the design space in Table 7.1.

BIBLIOGRAPHY

[1] Akihiko Akiyama, Suguru Yoshida, Tetsuo Matsuhashi, Nagarajan Rangarajan, Tariq Shams, Hirotoshi Ishikawa, and Atsuhiro Konosu. "Development of Simulation Model and Pedestrian Dummy." In: *International Congress & Exposition*. SAE International, 1999 (cit. on p. 12).

[2] Pierre-Jean Arnoux, Dominique Cesari, Michel Behr, Lionel Thollon, Jérome Cardot, and Christian Brunet. "Pedestrian lower limb injury criteria evaluation: A finite element approach." In: *Traffic Injury Prevention* 6.3 (2005), pp. 288–297 (cit. on p. 12).

[3] Hiroyuki Asanuma, Yukou Takahashi, Miwako Ikeda, and Toshiyuki Yanaoka. "Investigation of a Simplified Vehicle Model that Can Reproduce Car-Pedestrian Collisions." In: *SAE World Congress & Exhibition*. SAE International, 2014 (cit. on pp. 16, 17).

[4] Christopher Bach. "Data-driven model order reduction for nonlinear crash and impact simulations." PhD Thesis. Germany: Technical University of Munich, 2019 (cit. on p. 221).

[5] David Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012 (cit. on p. 72).

[6] Roman Barták. "Incomplete Depth-First Search Techniques: A Short Survey." In: *Proceedings of the 6th Workshop on Constraint Programming for Decision and Control*. Ed. by Jarosław Figwer. 2004 (cit. on p. 118).

[7] Brian Beachkofski and Ramana Grandhi. "Improved Distributed Hypercube Sampling." In: *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2002 (cit. on pp. 69, 99).

[8] Richard Bellman. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957 (cit. on pp. 55, 97).

[9] Hans-Georg Beyer and Bernhard Sendhoff. "Robust optimization: a comprehensive survey." In: *Computer Methods in Applied Mechanics and Engineering* 196.33 (2007), pp. 3190–3218 (cit. on p. 20).

[10] Fani Boukouvala and Marianthi G. Ierapetritou. "Feasibility analysis of black-box processes using an adaptive sampling Kriging-based method." In: *Computers & Chemical Engineering* 36 (2012), pp. 358–368 (cit. on pp. 88, 98, 99, 132).

[11] Fani Boukouvala and Marianthi G. Ierapetritou. "Derivative-free optimization for expensive constrained problems using a novel expected improvement objective function." In: *AIChE Journal* 60.7 (2014), pp. 2462–2474 (cit. on pp. 80, 81).

[12] George E. P. Box. "Robustness in the Strategy of Scientific Model Building." In: *Robustness in Statistics*. Ed. by Robert L. Launer and Graham N. Wilkinson. Academic Press, 1979, pp. 201–236 (cit. on p. 55).

[13] Riaan Brits, Andries P. Engelbrecht, and Frans van den Bergh. "A niching Particle Swarm Optimizer." In: *In Proceedings of the Conference on Simulated Evolution And Learning*. 2002, pp. 692–696 (cit. on p. 88).

[14] R. Buchheim, K.-R. Deutenbach, and H.-J. Lückoff. "Necessity and Premises for Reducing the Aerodynamic Drag of Future Passenger Cars." In: *SAE Transactions* 90 (1981), pp. 758–771 (cit. on p. 3).

[15] Dominique Cesari, Cavallero, Claude, Cassan, Françoise, and Charles Moffatt. "Interaction between human leg and car bumper in pedestrian tests." In: *Proceedings of the International Conference on the Biomechanics of Impacts*. Bergisch Gladbach, Germany, 1988 (cit. on p. 7).

[16] Stefano Chiapedi, Andreas Koukal, and Fabian Duddeck. "Sensitivity Analysis for Pedestrian Lower Leg Impact." In: *Proceedings of the 7th GACM Colloquium on Computational Mechanics for Young Scientists from Academia and Industry*. 2017 (cit. on p. 10).

[17] Davide Chicco and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation." In: *BMC genomics* 21.1 (2020), p. 6 (cit. on p. 95).

[18] China Automotive Technology and Research Center Co., Ltd. *C-NCAP Management Regulation: (2021 Edition)*. 2022 (cit. on pp. 9, 50).

[19] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. 3rd ed. The MIT Press, 2009 (cit. on p. 117).

[20] Marco Daub, Fabian Duddeck, and Markus Zimmermann. "Optimizing component solution spaces for systems design." In: *Structural and Multidisciplinary Optimization* 61.5 (2020), pp. 2097–2109 (cit. on p. 23).

[21] Marc Peter Deisenroth, Yicheng Luo, and Mark van der Wilk. *A Practical Guide to Gaussian Processes*. Distill (cit. on pp. 79, 100).

[22] Norman R. Draper and Harry Smith. *Applied Regression Analysis*. 3rd ed. Wiley-Interscience. John Wiley & Sons, Inc., 1998 (cit. on pp. 91, 92).

[23] Fabian Duddeck. "Multidisciplinary optimization of car bodies." In: *Structural and Multidisciplinary Optimization* 35.4 (2008), pp. 375–389 (cit. on p. 5).

[24] Stefan Erschen, Fabian Duddeck, Matthias Gerdts, and Markus Zimmermann. "On the Optimal Decomposition of High-Dimensional Solution Spaces of Complex Systems." In: *ASCE-ASME J. Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 4.2 (2017) (cit. on pp. 23, 234).

[25] ESI Group. *Virtual Performance Solution: Solver Reference Manual: Version: 2018*. Tech. rep. (cit. on pp. 36, 43, 46, 52, 53).

[26] European Experimental Vehicles Committee. *EEVC Working Group 17 Report: Improved Test Methods To Evaluate Pedestrian Protection Afforded By Passenger Cars*. Tech. rep. 1998 (cit. on p. 7).

[27] European New Car Assessment Programme. *Assessment Protocol: Pedestrian Protection. Version 7.0.* 2014 (cit. on p. 9).

[28] European Parliament and Council of the European Union. *Directive 2003/102/EC*. 2003 (cit. on pp. 6, 7).

[29] Johannes Fender. "Solution Spaces for Vehicle Crash Design." PhD Thesis. Germany: Technical University of Munich, 2014 (cit. on pp. 22, 110, 132).

[30] Jonathan E. Fieldsend. "Running Up Those Hills: Multimodal search with the niching migratory multi-swarm optimiser." In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 2593–2600 (cit. on pp. 88, 90).

[31] Rikard Fredriksson. "Priorities and Potential of Pedestrian Protection: Accident data, Experimental tests and Numerical Simulations of Car-to-Pedestrian Impacts." PhD Thesis. Sweden: Karolinska Institutet, 2011 (cit. on p. 6).

[32] Yue Fu, Huijie Xu, Guan Lin, Zhenfei Zhan, Ping Wang, Ruyi Chen, and Huili Yu. "A Design and Optimization Method for Pedestrian Lower Extremity Injury Analysis with the aPLI Model." In: *WCX SAE World Congress Experience*. SAE International, 2020 (cit. on pp. 13, 14, 190).

[33] Christian Gehre, Heinrich Gades, and Philipp Wernicke. "Objective Rating of Signals Using Test and Simulation Responses." In: *Proceedings of the 21st International Technical Conference on the Enhanced Safety of Vehicles Conference (ESV)*. 2009 (cit. on p. 51).

[34] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. "Kriging Is Well-Suited to Parallelize Optimization." In: *Computational Intelligence in Expensive Optimization Problems*. Ed. by Yoel Tenne and Chi-Keong Goh. Adaptation Learning and Optimization. Berlin, Heidelberg: Springer, 2010, pp. 131–162 (cit. on p. 86).

[35] Gene H. Golub and Charles F. van Loan. *Matrix computations*. Fourth edition. Johns Hopkins studies in the mathematical sciences. Baltimore: The Johns Hopkins University Press, 2013 (cit. on p. 41).

[36] Vishal Goyal and Marianthi G. Ierapetritou. "Determination of operability limits using simplicial approximation." In: *AIChE Journal* 48.12 (2002), pp. 2902–2909 (cit. on p. 20).

[37] Wolfgang Graf, Marco Götz, and Michael Kaliske. "Computing permissible design spaces under consideration of functional responses." In: *Advances in Engineering Software* 117 (2018), pp. 95–106 (cit. on pp. 22, 230).

[38] Lavinia Graff. "A stochastic algorithm for the identification of solution spaces in high-dimensional design spaces." PhD Thesis. Switzerland: Universität Basel, 2013 (cit. on p. 108).

[39] Lavinia Graff, Helmut Harbrecht, and Markus Zimmermann. "On the computation of solution spaces in high dimensions." In: *Structural and Multidisciplinary Optimization* 54.4 (2016), pp. 811–829 (cit. on pp. 21, 22, 107, 112, 132).

[40] Robert B. Gramacy and Herbert K. H. Lee. "Bayesian Treed Gaussian Process Models With an Application to Computer Modeling." In: *Journal of the American Statistical Association* 103.483 (2008), pp. 1119–1130 (cit. on pp. 78, 79).

[41] Robert B. Gramacy and Herbert K. H. Lee. "Cases for the nugget in modeling computer experiments." In: *Statistics and Computing* 22.3 (2012), pp. 713–722 (cit. on p. 78).

[42] Ignacio E. Grossmann and Manfred Morari. *Operability, Resiliency, and Flexibility: process design objectives for a changing world*. Tech. rep. DRC-O6-55-83. Design Research Center, Carnegie-Mellon University, 1983 (cit. on pp. 19, 66).

[43] Reinhard Haberfellner, Olivier de Weck, Ernst Fricke, and Siegfried Vössner. *Systems Engineering: Fundamentals and Applications*. Springer International Publishing, 2019 (cit. on p. 4).

[44] Keshava P. Halemane and Ignacio E. Grossmann. "Optimal process design under uncertainty." In: *AIChE Journal* 29.3 (1983), pp. 425–433 (cit. on pp. 19, 67, 68).

[45] Yong Ha Han and Young Woo Lee. "Optimization of Bumper Structure for Pedestrian Lower Leg Impact." In: *SAE World Congress & Exhibition*. SAE International, 2002 (cit. on p. 12).

[46] Helmut Harbrecht, Dennis Tröndle, and Markus Zimmermann. "A sampling-based optimization algorithm for solution spaces with pair-wise-coupled design variables." In: *Structural and Multidisciplinary Optimization* 60.2 (2019), pp. 501–512 (cit. on p. 234).

[47] Helmut Harbrecht, Dennis Tröndle, and Markus Zimmermann. *Approximating solution spaces as a product of polygons*. Preprints Fachbereich Mathematik. University of Basel. 2019 (cit. on p. 234).

[48] William D. Harvey and Matthew L. Ginsberg. "Limited Discrepancy Search." In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. Ed. by Christopher S. Mellish. Morgan-Kaufmann, 1995, pp. 607–615 (cit. on pp. 118, 119).

[49] William T. Hollowell. "Adaptive Time Domain, Constrained System Identification of Nonlinear Structures." PhD Thesis. USA: University of Virginia, 1986 (cit. on p. 15).

[50] Deng Huang, Theodore T. Allen, William I. Notz, and Ning Zeng. "Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models." In: *Journal of Global Optimization* 34.3 (2006), pp. 441–466 (cit. on p. 70).

[51] Jun Huang, Yong Xia, Bingbing Nie, and Qing Zhou. "A bumper model with dynamic contact stiffness for simulations of pedestrian legform impacts." In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 227.6 (2013), pp. 905–913 (cit. on pp. 15, 16, 62, 219).

[52] Hirotoshi Ishikawa, Janusz Kajzer, and Günter Schroeder. "Computer Simulation of Impact Response of the Human Body in Car-Pedestrian Accidents." In: *Stapp Car Crash Conference*. SAE International, 1993 (cit. on p. 11).

[53] Takahiro Isshiki, Jacobo Antona-Makoshi, Atsuhiro Konosu, and Yukou Takahashi. "Simplifying the Structural Design of the Advanced Pedestrian Legform Impactor for Use in Standardized Testing." In: *WCX World Congress Experience*. SAE International, 2018 (cit. on p. 9).

[54] Takahiro Isshiki, Atsuhiro Konosu, and Yukou Takahashi. "Influence of the Upper Body of Pedestrians on Lower Limb Injuries and Effectiveness of the Upper Body Compensation Method of the FlexPLI." In: *SAE International Journal of Transportation Safety* 3.2 (2015), pp. 153–163 (cit. on p. 9).

[55] Takahiro Isshiki, Atsuhiro Konosu, and Yukou Takahashi. "Development and Evaluation of the Advanced Pedestrian Legform Impactor Prototype Which Can be Applicable to All Types of Vehicles Regardless of Bumper Height: Part 1: Finite Element Model." In: *International Research Council on Biomechanics of Injury (IRCOBI)*. Malaga , Spain, 2016 (cit. on pp. 9, 138).

[56] Momin Jamil and Xin-She Yang. "A literature survey of benchmark functions for global optimisation problems." In: *International Journal of Mathematical Modelling and Numerical Optimisation* 4.2 (2013), pp. 150–194 (cit. on p. 163).

[57] Norman L. Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous Univariate Distributions*. 2nd ed. John Wiley & Sons, Inc., 1995 (cit. on p. 129).

[58] Donald R. Jones. "A Taxonomy of Global Optimization Methods Based on Response Surfaces." In: *Journal of Global Optimization* 21.4 (2001), pp. 345–383 (cit. on p. 70).

[59] Donald R. Jones, Matthias Schonlau, and William J. Welch. "Efficient Global Optimization of Expensive Black-Box Functions." In: *Journal of Global Optimization* 13.4 (1998), pp. 455–492 (cit. on pp. 70, 80, 81).

[60] Mounir M. Kamal. "Analysis and Simulation of Vehicle to Barrier Impact." In: *International Automobile Safety Conference*. SAE International, 1970 (cit. on p. 15).

[61] Shaikh Karimullah, Ajit Gokhale, Samir Joshi, and Sujit Chalipat. "Design of "Lower Stiffener" for Controlling Pedestrian Lower Leg Injuries." In: *SIAT*. The Automotive Research Association of India, 2011 (cit. on p. 12).

[62] Venkatason Kausalyah, Sivaguru Shasthri, Kassim A. Abdullah, Moumen M. Idres, Qasim H. Shah, and Shaw V. Wong. "Optimisation of vehicle front-end geometry for adult and pediatric pedestrian protection." In: *International Journal of Crashworthiness* 19.2 (2014), pp. 153–160 (cit. on p. 12).

[63] James Kennedy and Russell C. Eberhart. "Particle swarm optimization." In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. 1995, pp. 1942–1948 (cit. on p. 88).

[64] André I. Khuri and Siuli Mukhopadhyay. "Response surface methodology." In: *WIREs Computational Statistics* 2.2 (2010), pp. 128–149 (cit. on p. 68).

[65] Jack P.C. Kleijnen and Robert G. Sargent. "A methodology for fitting and validating metamodels in simulation." In: *European Journal of Operational Research* 120.1 (2000), pp. 14–29 (cit. on p. 90).

[66] Ron Kohavi. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. 1995 (cit. on p. 96).

[67] Andrey Kolmogoroff. "Interpolation und Extrapolation von stationären zufälligen Folgen." Russian, German. In: *Bulletin de l'Académie des sciences de l'URSS. Série mathématique.* 5 (1941), pp. 3–14 (cit. on p. 71).

[68] Atsuhiro Konosu, Hirotoshi Ishikawa, and Masaaki Tanahashi. "Reconsideration of injury criteria for pedestrian subsystem legform test Problems of rigid legform impactor." In: *International Technical Conference on Enhanced Safety of Vehicles*. National Highway Traffic Safety Administration, 2001 (cit. on p. 7).

[69] Atsuhiro Konosu, Takahiro Issiki, Yukou Takahashi, Hideki Suzuki, Bernard Been, Mark Burleigh, Takahiro Hirasawa, and Hitoshi Kanoshima. "Development Of A Biofidelic Flexible Pedestrian Legform Impactor Type Gtr Prototype: Part 1: Development And Technical Evaluations." In: *Proceedings of the 21st International Technical Conference on the Enhanced Safety of Vehicles Conference (ESV)*. 2009 (cit. on p. 7).

[70] Atsuhiro Konosu, Takahiro Issiki, and Masaaki Tanahashi. "Development of a Biofidelic Flexible Pedestrian Leg-form Impactor (Flex-PLI 2004) and Evaluation of its Biofidelity at the Component Level and at the Assembly Level." In: *SAE Transactions* 114 (2005), pp. 2298–2311 (cit. on p. 7).

[71] Richard E. Korf. "Depth-first iterative-deepening: An optimal admissible tree search." In: *Artificial Intelligence* 27.1 (1985), pp. 97–109 (cit. on p. 118).

[72] Richard E. Korf. "Improved Limited Discrepancy Search." In: *Proceedings of the 13th National Conference on Artificial Intelligence*. 1996, pp. 286–291 (cit. on pp. 117, 119).

[73] Andreas Koukal. "Crash- und Bruchverhalten von Kunststoffen im Fußgängerschutz von Fahrzeugen." PhD thesis. Germany: Technical University of Munich, 2014 (cit. on p. 7).

[74] Daniel G. Krige. "A statistical approach to some mine valuation and allied problems on the Witwatersrand." Master's thesis. Johannesburg, South Africa: University of the Witwatersrand, 1951 (cit. on p. 71).

[75] Matthias Kühn, Robert Fröming, and Volker Schindler. *Fußgängerschutz: Unfallgeschehen, Fahrzeuggestaltung, Testverfahren*. German. Springer, 2006 (cit. on p. 6).

[76] Steffen L. Lauritzen. "Time Series Analysis in 1880: A Discussion of Contributions Made by T.N. Thiele." In: *International Statistical Review / Revue Internationale de Statistique* 49.3 (1981), pp. 319–331 (cit. on p. 71).

[77] Y. H. Lee, Y. J. Joo, J. S. Park, Y. S. Kim, and H. J. Yim. "Robust design optimization of frontal structures for minimizing injury risks of Flex Pedestrian Legform Impactor." In: *International Journal of Automotive Technology* 15.5 (2014), pp. 757–764 (cit. on pp. 13, 14).

[78] Matthew Lehar and Markus Zimmermann. "An inexpensive estimate of failure probability for high-dimensional systems with uncertainty." In: *Structural Safety* 36-37 (2012), pp. 32–38 (cit. on p. 128).

[79] Guibing Li, Jikuang Yang, and Ciaran Simms. "Safer passenger car front shapes for pedestrians: A computational approach to reduce overall pedestrian injury risk in realistic impact scenarios." In: *Accident Analysis & Prevention* 100 (2017), pp. 97–110 (cit. on p. 12).

[80] Xiaodong Li, Andries P. Engelbrecht, and Michael G. Epitropakis. "Results of the 2015 IEEE CEC Competition on Niching Methods for Multimodal Optimization." In: *IEEE Congress on Evolutionary Computation*. 2015 (cit. on p. 88).

[81] Xiaojiang Lv, Xianguang Gu, Liangguo He, Dayong Zhou, and Weiguo Liu. "Reliability design optimization of vehicle front-end structure for pedestrian lower extremity protection under multiple impact cases." In: *Thin-Walled Structures* 94 (2015), pp. 500–511 (cit. on p. 12).

[82] Bertil Matérn. "Spatial Variation: Stochastic models and their application to some problems in forest surveys and other sampling investigations." PhD Thesis. Stockholm: Forestry Research Institute of Sweden, 1960 (cit. on p. 73).

[83] Georges Matheron. "Principles of geostatistics." In: *Economic Geology* 58.8 (1963), pp. 1246–1266 (cit. on p. 71).

[84] Brian W. Matthews. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." In: *Biochimica et Biophysica Acta (BBA) - Protein Structure* 405.2 (1975), pp. 442–451 (cit. on p. 95).

[85] Michael D. McKay, Richard J. Beckman, and William J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." In: *Technometrics* 21.2 (1979), pp. 239–245 (cit. on pp. 69, 127).

[86] Jonas Močkus. "On bayesian methods for seeking the extremum." In: *IFIP Technical Conference on Optimization Techniques Novosibirsk, 1974*. Ed. by Gury I. Marchuk. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer-Verlag, 1975 (cit. on pp. 70, 80).

[87] Jonas Močkus. *Bayesian Approach to Global Optimization: Theory and Applications*. Mathematics and its Applications. Springer Netherlands, 1989 (cit. on p. 71).

[88] Simon Mößner. "Multi-fidelity Structural Design for Pedestrian Safety with particular Reference to the FlexPLI." PhD Thesis. Germany: Technical University of Munich, 2019 (cit. on pp. 17, 43, 58, 62, 220).

[89] Simon Mößner, Tim Rudolph, and Fabian Duddeck. "Surface modelling of vehicle frontends for pedestrian safety with the FlexPLI." In: *International Journal of Crashworthiness* 22.3 (2017), pp. 243–259 (cit. on pp. 16–18, 34, 38, 44, 47, 50, 62, 213).

[90] Daisuke Nakane, Sadayuki Kuwahara, Yasuhito Ozeki, Junichi Taniguchi, and Koji Mizuno. "Analysis of loading of lower extremities based on bending moment in car-to-pedestrian collisions." In: *International Journal of Crashworthiness* 15.5 (2010), pp. 481–490 (cit. on pp. 13, 14).

[91] Alok Nanda, Gopikrishna Surisetty, Subhransu Mohapatra, Shaleena AD, Stephen Shuler, and Frank Mooijman. "Method for Designing and Evaluating Pedestrian Protection Energy Absorbers for Various Car Geometries." In: *SAE World Congress & Exhibition*. SAE International, 2004 (cit. on pp. 13, 14).

[92] Mark O. Neal, ed. *Front Structure Design Procedure for Optimal Pedestrian Leg Impact Performance*. Vol. Volume 1: 30th Design Automation Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 2004 (cit. on pp. 15, 16, 34).

[93] Mark O. Neal, Jian Tu, and Donald R. Jones. "A Response Surface Based Tool for Evaluating Vehicle Performance in the Pedestrian Leg Impact Test." In: *SAE World Congress & Exhibition*. SAE International, 2008 (cit. on pp. 15, 17, 18).

[94] Bingbing Nie, Jun Huang, Yong Xia, Qing Zhou, Bing Deng, and Mark Neal. "Development of a Parametric Vehicle Front Structure Model for Pedestrian Impact Simulations." In: *Proceedings of the FISITA World Automotive Congress*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 295–309 (cit. on pp. 15, 16, 62, 219).

[95] Bingbing Nie, Yong Xia, Qing Zhou, Jun Huang, Bing Deng, and Mark Neal. "A Response-surface-based Tool for Vehicle Front-end Design for Pedestrian Impact Protection using Human Body Model." In: *International Journal of Vehicle Design* 66 (2014), pp. 347–362 (cit. on pp. 17, 18).

[96] Travis E. Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006 (cit. on p. 41).

[97] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on pp. 48, 79).

[98] Rüdiger Rackwitz. "Reliability analysis: a review and some perspectives." In: *Structural Safety* 23.4 (2001), pp. 365–395 (cit. on p. 20).

[99] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006 (cit. on pp. 71–73, 76, 78, 79).

[100] Saman Razavi, Bryan A. Tolson, and Donald H. Burn. "Review of surrogate modeling in water resources." In: *Water Resources Research* 48.7 (2012) (cit. on p. 68).

[101] Amanda Rogers and Marianthi Ierapetritou. "Feasibility and flexibility analysis of black-box processes Part 1: Surrogate-based feasibility analysis." In: *Chemical Engineering Science* 137 (2015), pp. 986–1004 (cit. on pp. 87, 99, 101, 132).

[102] Lior Rokach. "A survey of Clustering Algorithms." In: *Data Mining and Knowledge Discovery Handbook*. Ed. by Oded Maimon and Lior Rokach. Boston, MA: Springer US, 2010, pp. 269–298 (cit. on p. 48).

[103] Howard H. Rosenbrock. "An Automatic Method for Finding the Greatest or Least Value of a Function." In: *Computer Journal* 3 (1960), pp. 175–184 (cit. on p. 161).

[104] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. "Design and Analysis of Computer Experiments." In: *Statistical Science* 4.4 (1989), pp. 409–423 (cit. on p. 69).

[105] Thomas J. Santner, Brian J. Williams, and William I. Notz. *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. Springer-Verlag New York, 2003 (cit. on p. 69).

[106] Michael J. Sasena, Panos Papalambros, and Pierre Goovaerts. "Exploration of Metamodeling Sampling Criteria for Constrained Global Optimization." In: *Engineering Optimization* 34.3 (2002), pp. 263–278 (cit. on p. 87).

[107] Alessandro Scattina, Fuhao Mo, Catherine Masson, Massimiliano Avalle, and Pierre-Jean Arnoux. "Analysis of the influence of passenger vehicles front-end design on pedestrian lower extremity injuries by means of the LLMS model." In: *Traffic Injury Prevention* 19.5 (2018), pp. 535–541 (cit. on p. 12).

[108] Matthias Schonlau. "Computer Experiments and Global Optimization." PhD Thesis. Canada: University of Waterloo, 1997 (cit. on pp. 80, 86, 87).

[109] Burr Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012 (cit. on p. 70).

[110] Thomas A. Severini. *Likelihood Methods in Statistics*. Oxford University Press, 2000 (cit. on p. 79).

[111] Songqing Shan and G. Gary Wang. "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions." In: *Structural and Multidisciplinary Optimization* 41.2 (2010), pp. 219–241 (cit. on p. 97).

[112] Kohei Shintani, Atsuji Abe, and Yasushi Yamamoto. "A Target Cascading Method Using Model Based Simulation in Early Stage of Vehicle Development." In: *SAE Technical Paper*. SAE International, 2019 (cit. on p. 81).

[113] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. "Practical Bayesian Optimization of Machine Learning Algorithms." In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc, 2012, pp. 2951–2959 (cit. on p. 79).

[114] Durward K. Sobek II, Allen C. Ward, and Jeffrey K. Liker. "Toyota's principles of set-based concurrent engineering." In: *MIT Sloan Management Review* 40.2 (1999), p. 67 (cit. on p. 19).

[115] Ilya M. Sobol. "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates." In: *Mathematics and Computers in Simulation* 55 (2001), pp. 271–280 (cit. on pp. 17, 192, 231).

[116] Statistisches Bundesamt. *Verkehrsunfälle*. Tech. rep. 2021 (cit. on p. 6).

[117] Katharina Stielau and David Blauth. "Advanced Pedestrian Legform Impactor (aPLI)." In: *16th International LS-DYNA Users Conference*. 2020 (cit. on p. 8).

[118] Mario Super. "Development of a Generic Vehicle Front-End Model for Pedestrian Lower Leg Impact Simulations (FlexPLI)." Master's Thesis. Germany: Technical University of Munich, 2016 (cit. on p. 52).

[119] Jiri Svoboda and Martin Kuklik. "Influence of bumper design to lower leg impact response." In: *FISITA World Automotive Congress*. 2006 (cit. on p. 12).

[120] Ross E. Swaney and Ignacio E. Grossmann. "An index for operational flexibility in chemical process design. Part I: Formulation and theory." In: *AIChE Journal* 31.4 (1985), pp. 621–630 (cit. on pp. 19, 20).

[121] Yukou Takahashi, Shunji Suzuki, Masayoshi Okamoto, Shinsuke Oda, Rikard Fredriksson, and Bengt Pipkorn. "Effect of stiffness characteristics of vehicle front-end structures on pedestrian pelvis and lower limb injury measures." In: *International Research Council on Biomechanics of Injury (IRCOBI) Conference*. 2011 (cit. on pp. 15, 34, 218).

[122] Alaa Tharwat. "Classification assessment methods." In: *Applied Computing and Informatics* (2018) (cit. on p. 95).

[123] Henri Theil. *Economic forecasts and policy*. 2nd ed. Amsterdam: North-Holland Pub. Co., 1961 (cit. on p. 92).

[124] United Nations. *UN Regulation No 127. 01 series of amendments to the Regulation*. 2015 (cit. on p. 9).

[125] Verein Deutscher Ingenieure. *Design methodology for mechatronic systems (VDI 2206)*. VDI, 2004 (cit. on p. 4).

[126] Marc E. Vogt, Fabian Duddeck, Martin Wahle, and Markus Zimmermann. "Optimizing tolerance to uncertainty in systems design with early- and late-decision variables." In: *IMA Journal of Management Mathematics* 30.3 (2018), pp. 269–280 (cit. on p. 23).

[127] David D. Walden, Garry J. Roedler, Kevin J. Forsberg, R. Douglas Hamelin, and Thomas M. Shortell, eds. *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. 4th ed. Hoboken, NJ: Wiley, 2015 (cit. on p. 4).

[128] Toby Walsh. "Depth-bounded Discrepancy Search." In: *Proceedings of the 15th International Joint Conference on Artificial Intelligence*. Ed. by Martha E. Pollack. Vol. 2. Morgan-Kaufmann, 1997, pp. 1388–1393 (cit. on p. 119).

[129] G. Gary Wang and Songqing Shan. "Review of Metamodeling Techniques in Support of Engineering Design Optimization." In: *Journal of Mechanical Design* 129.4 (2006), pp. 370–380 (cit. on p. 68).

[130] Zilong Wang and Marianthi Ierapetritou. "A novel feasibility analysis method for black-box processes using a radial basis function adaptive sampling approach." In: *AIChE Journal* 63.2 (2017), pp. 532–550 (cit. on pp. 68, 88, 98, 99, 153).

[131] Zilong Wang and Marianthi Ierapetritou. "Surrogate-based feasibility analysis for black-box stochastic simulations with heteroscedastic noise." In: *Journal of Global Optimization* 71.4 (2018), pp. 957–985 (cit. on pp. 81, 98, 132, 153).

[132] Joe H. Ward JR. "Hierarchical Grouping to Optimize an Objective Function." In: *Journal of the American Statistical Association* 58.301 (1963), pp. 236–244 (cit. on p. 48).

[133] Julian Weber. *Automotive Development Processes: Processes for Successful Customer Oriented Vehicle Development*. Springer Berlin Heidelberg, 2009 (cit. on p. 3).

[134] Simon Wessing. "Two-stage Methods for Multimodal Optimization." PhD Thesis. Germany: Technische Universität Dortmund, 2015 (cit. on pp. 99, 127).

[135] Isabelle Wetzstein, Beate Lauterbach, Matthias Erzgräber, and Lothar Harzheim. "Optimization of a Lower Bumper Support regarding Pedestrian Protection Requirements using ANSA and LS-Opt." In: *10th European LS-Dyna Conference*. 2015 (cit. on p. 13).

[136] Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: with engineering applications*. The MIT Press, 1949 (cit. on p. 71).

[137] World Health Organization. *Global status report on road safety*. Tech. rep. 2018 (cit. on p. 6).

[138] Jikuang Yang. "Review of injury biomechanics in car-pedestrian collisions." In: *International journal of vehicle safety* 1.1-3 (2005), pp. 100–117 (cit. on p. 6).

[139] Guanjun Zhang, Libo Cao, Jingwen Hu, and King H. Yang. "A field data analysis of risk factors affecting the injury risks in vehicle-to-pedestrian crashes." eng. In: *Annals of advances in automotive medicine. Association for the Advancement of Automotive Medicine. Annual Scientific Conference* 52 (2008), pp. 199–214 (cit. on p. 6).

[140] Markus Zimmermann, Simon Königs, Constantin Niemeyer, Johannes Fender, Christian Zeherbauer, Renzo Vitale, and Martin Wahle. "On the design of large systems subject to uncertainty." In: *Journal of Engineering Design* 28.4 (2017), pp. 233–254 (cit. on p. 5).

[141] Markus Zimmermann and Johannes E. von Hoessle. "Computing solution spaces for robust design." In: *International Journal for Numerical Methods in Engineering* 94.3 (2013), pp. 290–307 (cit. on pp. 21, 22, 103, 110, 112–115, 130, 132, 134, 230).

DECLARATION

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

Munich, 2021

Stefano Chiapedi

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*".