

Adaptive surrogate models for the reliable lightweight design of automotive body structures

Maliki Moustapha

► To cite this version:

Maliki Moustapha. Adaptive surrogate models for the reliable lightweight design of automotive body structures. Human health and pathology. Université Blaise Pascal - Clermont-Ferrand II, 2016. English. NNT: 2016CLF22670. tel-01343383

HAL Id: tel-01343383 https://theses.hal.science/tel-01343383

Submitted on 8 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés. N° d'ordre : D. U. : 2670 E D S P I C : 761

> Université BLAISE PASCAL - Clermont II École Doctorale Sciences pour l'Ingénieur de Clermont-Ferrand

Thèse

présentée par

Maliki MOUSTAPHA Ingénieur IFMA

en vue d'obtenir le grade de

Docteur d'Université (Spécialité: Génie Mécanique)

Métamodèles adaptatifs pour l'optimisation fiable multi-prestations de la masse de véhicules

soutenue publiquement le 27 janvier 2016 devant un jury composé de MM.

Bruno SUDRET	ETH Zurich, Zurich	Directeur de thèse
Jean-Marc BOURINET	Sigma Clermont, Clermont-Ferrand	Co-encadrant
Benoît GUILLAUME	PSA Peugeot Citroën, Vélizy-Villacoublay	Encadrant indusriel
Pierre-Alain BOUCARD	ENS Cachan, Cachan	Rapporteur
Bertrand IOOSS	EDF R&D, Chatou	Rapporteur
Alaa CHATEAUNEUF	Université Blaise Pascal, Clermont-Ferrand	Examinateur
Rodolphe LE RICHE	ENSM-SE, Saint-Etienne	Président du jury
Bertrand IOOSS Alaa CHATEAUNEUF Rodolphe LE RICHE	EDF R&D, Chatou Université Blaise Pascal, Clermont-Ferrand ENSM-SE, Saint-Etienne	Rapporteur Examinateur Président du jury

Adaptive surrogate models for the reliable lightweight design of automotive body structures

A thesis submitted by **Maliki MOUSTAPHA** in partial fulfillment of the requirements for the degree of doctor of philosophy

BLAISE PASCAL UNIVERSITY – CLERMONT II

Clermont-Ferrand, France

defended publicly on January 27, 2016 in front of a jury made up of:

Bruno SUDRET	ETH Zurich, Zurich	Thesis advisor
Jean-Marc BOURINET	Sigma Clermont, Clermont-Ferrand	Thesis co-advisor
Benoît GUILLAUME	PSA Peugeot Citroën, Vélizy-Villacoublay	Industrial advisor
Pierre-Alain BOUCARD	ENS Cachan, Cachan	Reviewer
Bertrand IOOSS	EDF R&D, Chatou	Reviewer
Alaa CHATEAUNEUF	Université Blaise Pascal, Clermont-Ferrand	Examiner
Rodolphe LE RICHE	ENSM-SE, Saint-Etienne	President of the jury

À Agna et Baaba

ACKNOWLEDGEMENTS / REMERCIEMENTS

Ce travail n'aurait été possible sans le concours d'un nombre important de personnes. Je tiens tout d'abord à adresser mes remerciements à mes encadrants: Bruno Sudret, Jean-Marc Bourinet et Benoît Guillaume. Je ne voudrais pas insister ici sur vos compétences scientifiques, reconnus par vos pairs, qui ont permis d'assurer un encadrement de qualité. Je souhaiterais surtout souligner vos qualités humaines qui m'ont permis de travailler sereinement durant ces trois années. Bruno, merci donc pour tes nombreuses et rigoureuses relectures de mes écrits ainsi que de ta patience. Merci également de m'offrir l'opportunité de continuer dans la recherche en m'invitant dans ta chaire à Zurich. Merci Jean-Marc pour tes nombreux conseils, pour le temps que tu m'as accordé à chacun de mes passages à Clermont-Ferrand. Je voudrais surtout te faire part de ma reconnaissance pour ton soutien durant la quasi-totalité de mon cursus universitaire. Si je me suis intéressé au monde de la recherche c'est en très grande partie grâce à toi. Benoît, merci d'abord de m'avoir fait confiance pour mener ces travaux de recherche au sein de PSA. Ta grande connaissance de l'automobile dans les détails les plus techniques m'a été fort utile. Enfin merci d'avoir veiller à ce que nos travaux aient une application industrielle concrète.

Je souhaiterais ensuite remercier le jury de la thèse : les rapporteurs Prof. Pierre-Alain Boucard et Dr. Bertrand Iooss, Prof. Alaa Chateauneuf en tant qu'examinateur et Dr. Rodolphe Le Riche en sa qualité de président. La bienveillance dont vous avez fait preuve à mon égard m'a sans doute permis d'exprimer clairement mes idées.

Mes remerciements vont ensuite à l'ensemble des collaborateurs que j'ai pu côtoyer au sein de PSA. Tout d'abord Mihai Socoliuc pour ton accueil et ton abnégation à valoriser en interne les travaux des doctorants. Bonne continuation dans l'animation de la communauté des doctorants de PSA. Ensuite, Laurent Di Valentin et Emmanuelle Vallée qui ont participé à mon entretien d'embauche. Merci de votre confiance et particulièrement à Emmanuelle pour ton soutien lorsque j'ai eu à vivre des moments difficiles. Je souhaiterais également remercier les membres du pôle optim pour votre aide : Babacar D. Mendy, Cyril Montamat, Fabien Figueres et Alexandre Bonnet. Enfin merci à l'ensemble du plateau VDE avec qui j'ai partagé des repas ou les petits-déjeuners du Vendredi, entre autres, Jean-Christophe mon voisin lors des derniers mois, Sylvain, Nicolas, Nicola, Yan, les Benoît, David, JC, Julien, Damien, Francis, Jacques, Zora, Pascal, Charles, Arnaud...

Je voudrais également adresser mes remerciements à Florian Canderatz qui a su ré-écrire sur Matlab mes développements afin de pérenniser les travaux de thèse au sein de PSA ainsi qu'à Miguel Dinis pour ses nombreux conseils sur Matlab. Je n'oublie pas l'ensemble des doctorants que j'ai pu rencontrer à l'ETH et à l'IFMA et qui m'ont certainement permis à un moment ou à un autre d'avancer dans mes travaux. Merci particulièrement à Mathieu et Djily de m'avoir aidé dans l'organisation de mon pot de thèse. Merci également à Marion qui s'est toujours assurée que mes déplacements se passent dans les meilleures conditions.

Ce travail n'aurait pu aboutir sans le soutien permanent de mes proches et amis particulièrement Mouktar, Yanick, Papi, Ahmed(s) et Pat. Un grand merci à mes soeurs Adda, Djami, Yas, et Nai pour vos encouragements et toute votre affection. Au moment d'écrire ces mots je suis triste que mes parents, *Agna* et *Baaba*, ne puissent les lire. Les mots pour leur exprimer ma reconnaissance et mon affection ne seraient de toutes façons pas adéquats.

Maliki Moustapha, le 06 Mars 2016.

ABSTRACT

One of the most challenging tasks in modern engineering is that of keeping the cost of manufactured goods small. With the advent of computational design, prototyping for instance, a major source of expenses, is reduced to its bare essentials. In fact, through the use of high-fidelity models, engineers can predict the behaviors of the systems they design quite faithfully. To be fully realistic, such models must embed uncertainties that may affect the physical properties or operating conditions of the system.

This PhD thesis deals with the constrained optimization of structures under uncertainties in the context of automotive design. The constraints are assessed through expensive finite element models. For practical purposes, such models are conveniently substituted by so-called *surrogate models* which stand as cheap and easy-to-evaluate proxies. In this PhD thesis, Gaussian process modeling and support vector machines are considered.

Upon reviewing state-of-the-art techniques for optimization under uncertainties, we propose a novel formulation for reliability-based design optimization which relies on quantiles. The formal equivalence of this formulation with the traditional ones is proved. This approach is then coupled to surrogate modeling. Kriging is considered thanks to its built-in error estimate which makes it convenient to adaptive sampling strategies. Such an approach allows us to reduce the computational budget by running the true model only in regions that are of interest to optimization. We therefore propose a two-stage enrichment scheme. The first stage is aimed at globally reducing the Kriging epistemic uncertainty in the vicinity of the limit-state surface. The second one is performed within iterations of optimization so as to locally improve the quantile accuracy. The efficiency of this approach is demonstrated through comparison with benchmark results.

An industrial application featuring a car under frontal impact is considered. The crash behavior of a car is indeed particularly affected by uncertainties. The proposed approach therefore allows us to find a reliable solution within a reduced number of calls to the true finite element model. For the extreme case where uncertainties trigger various crash scenarios of the car, it is proposed to rely on support vector machines for classification so as to predict the possible scenarios before metamodeling each of them separately.

Keywords: Surrogate modeling, Adaptive design, Uncertainties, Reliability-based design optimization, Quantiles, Frontal impact.

Résumé

Cette thèse s'inscrit dans le cadre des travaux menés par PSA Peugeot Citroën pour l'allègement de ses véhicules. Les optimisations masse multi-prestations réalisées sur le périmètre de la structure contribuent directement à cette démarche en recherchant une allocation d'épaisseurs de tôles à masse minimale qui respectent des spécifications physiques relatives à différentes prestations (choc, vibro-acoustique, etc.). Ces spécifications sont généralement évaluées à travers des modèles numériques à très haute-fidélité qui présentent des temps de restitution particulièrement élevés. Le recours à des fonctions de substitution, connues sous le nom de métamodèles, reste alors la seule alternative pour mener une étude d'optimisation tout en respectant les délais projet. Cependant la prestation qui nous intéresse, à savoir le choc frontal, présente quelques particularités (grande dimensionnalité, fortes non-linéarités, dispersions physique et numérique) qui rendent sa métamodélisation difficile.

L'objectif de la thèse est alors de proposer une approche d'optimisation basée sur des métamodèles adaptatifs afin de dégager de nouveaux gains de masse. Cela passe par la prise en compte du choc frontal dont le caractère chaotique est exacerbé par la présence d'incertitudes. Nous proposons ainsi une méthode d'optimisation fiabiliste avec l'introduction de quantiles comme mesure de conservatisme. L'approche est basée sur des modèles de krigeage avec enrichissement adaptatif afin de réduire au mieux le nombre d'appels aux modèles éléments finis. Une application sur un véhicule complet permet de valider la méthode.

Mots clés : Métamodèles, Plan d'expériences adaptatifs, Incertitudes, Optimisation robuste et fiable, Quantiles, Choc frontal.

CONTENTS

Contents vii				
1	Introduction			1
2	Stru	ctural o	lesign optimization	9
	2.1	Introd	uction to design optimization	10
		2.1.1	Deterministic design optimization	10
		2.1.2	An overview of optimization algorithms	12
			2.1.2.1 Gradient-based methods	12
			2.1.2.2 Gradient-free methods	13
		2.1.3	Beyond the deterministic framework	18
			2.1.3.1 Sources of uncertainties	18
			2.1.3.2 Probability theory	19
			2.1.3.3 Summary	23
	2.2	Optim	ization under uncertainties	24
		2.2.1	Robust design optimization	24
			2.2.1.1 The concept of robust design	24
			2.2.1.2 A glimpse on Taguchi's pioneering work	25
			2.2.1.3 Broader formulations for robust design optimization	26
		2.2.2	Reliability-based design optimization	29
			2.2.2.1 From deterministic to probabilistic approach of safety	29
			2.2.2.2 Problem formulation	31
			2.2.2.3 Structural reliability analysis	33
			2.2.2.4 Solution of the RBDO problem	39
		2.2.3	Concluding remarks	43
	2.3	Quant	ile-based RBDO	44
		2.3.1	Motivation	44
		2.3.2	Formulation	44
		2.3.3	Analytical examples	46
			2.3.3.1 Mathematical models	46
			2.3.3.2 Mechanical models	49
			2.3.3.3 Summary	53
	2.4	Conclu	asion	54
3	Mot	amodel	ling techniques	55
5	31	Introd	uction	56
	3.2	An int	roduction to computer experiments	56
	0.2	3.2.1	Concepts and notations	56
			1	20

		3.2.2	Quality of a metamodel
		3.2.3	A brief overview of existing methods
	3.3	Suppo	ort vector machines
		3.3.1	An introduction to the structural learning theory
			3.3.1.1 From empirical to structural risk minimization 60
			3.3.1.2 The structural risk minimization principle
		3.3.2	Support vector machines for classification
			3.3.2.1 Classification of linearly separable data
			3.3.2.2 Classification of non-linearly separable data
		3.3.3	Support vector machines for regression
			3.3.3.1 Formulation of the SVR problem
			3.3.3.2 Implementation of the regression problem
			3.3.3.3 Other formulations of support vector regression
		3.3.4	Model training in SVR
			3.3.4.1 Overview of training techniques
			3.3.4.2 Approximation of the leave-one-out error
		3.3.5	An illustration of support vector regression
			3.3.5.1 Manual selection
			3.3.5.2 Automatic selection
	3.4	Krigin	g or Gaussian process modeling
		3.4.1	Introduction to Gaussian random functions
		3.4.2	The two-stage model83
			3.4.2.1 The different types of Kriging 83
			3.4.2.2 Auto-correlation functions for Kriging
			3.4.2.3 Sample path properties
		3.4.3	Formulation of the Kriging model
			3.4.3.1 The generalized least-square estimate of the parameters 88
			3.4.3.2 Kriging as the best linear unbiased predictor
		3.4.4	Model training in Kriging
			3.4.4.1 Maximum likelihood estimation
			3.4.4.2 Cross-validation
		3.4.5	An illustration of Kriging
			3.4.5.1 Some properties of Kriging predictor
			3.4.5.2 The nugget effect
	3.5	Conclu	usion
1	C	logato 1	accidentimization 07
4		Ugate-	viction 97
	4.1	Cloba	l optimization using surrogate models
	4.4	G100a	Problem formulation 00
		4.2.1	The sugmented reliability space 100
		4.2.2	4.2.2.1 A brief literature review 100
			4.2.2.1 The proposed suggested space 101
		422	Analytical validation example
		T. 2.0	4 2 3 1 Presentation of the problem 102
			4232 Results 103
	43	Adapt	ive design of experiments
	т.Ј	4 3 1	A word on initial design of experiments
		1.0.1	The of the of the matter of the of th

		4.3.2 Using a surrogate model for the objective function	08
		4.3.2.1 Sampling at the current minimum	08
		4.3.2.2 Probabilistic framework for surrogate-based optimization 1	09
		4.3.3 Using surrogate models for the constraints	12
		4.3.3.1 Adaptations of the expected improvement	13
		4.3.3.2 Expected feasibility function	15
		4.3.3.3 Deviation number	15
		4.3.3.4 Margin probability function	16
		4.3.3.5 Illustration examples	16
		4.3.4 Multiple points enrichment	21
		4.3.4.1 Sampling from the learning function	21
		4.3.4.2 Weighted K-means clustering	23
		4.3.5 Multi-constraints handling	24
	4.4	RBDO with surrogate model and enrichment	27
		4.4.1 Introduction	27
		4.4.2 Contour refinement with quantile estimation	27
	4.5	RBDO with two stages of enrichment	31
		4.5.1 Motivations and first stage of enrichment	31
		4.5.2 The second stage of enrichment	31
		4.5.2.1 Enrichment for quantile accuracy	32
		4.5.2.2 A summary of the proposed methodology	36
		4.5.3 Validation example	38
	16	Conductor	10
	4.0		40
_	4.0		40
5	4.0 Aca	demic and industrial applications	40
5	4.0 Aca 5.1	demic and industrial applications 1 Introduction 1	40 41 42
5	4.6 Aca 5.1 5.2	demic and industrial applications 1 Introduction 1 Academic validation examples 1	41 42 42
5	4.0 Aca 5.1 5.2	demic and industrial applications 1 Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1	40 41 42 42 42
5	4.0 Aca 5.1 5.2	demic and industrial applications 14 Introduction 14 Academic validation examples 14 5.2.1 Column under compression 14 5.2.2 Choi problem 14	40 41 42 42 42 42 44
5	4.0 Aca 5.1 5.2	Introduction Introduction Academic validation examples Introduction 5.2.1 Column under compression Introduction 5.2.2 Choi problem Introduction 5.2.3 Bracket structure Introduction	40 41 42 42 42 42 44 47
5	4.0 Aca 5.1 5.2	Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1 5.2.2 Choi problem 1 5.2.3 Bracket structure 1 5.2.4 Divide the structure 1 5.2.5 Choi problem 1 5.2.4 Divide the structure 1 5.2.5 Divide the structure 1 5.2.4 Divide the structure 1 5.2.5 Divide the structure 1 5.2.4 Divide the structure 1 5.2.5 Divide the structure 1 5.2.6 Divide the structure 1 5.2.7 Divide the structure 1 5.2.8 Divide the structure 1 5.2.9 Divide the structure 1 5.2.9 Divide the structure 1 5.2.1 Divide the structure 1 5.2.3 Divide the structure 1 5.2.4 Divide the structure 1 5.2.5 Divide the structure 1 5.2.6 Divid	40 41 42 42 42 42 44 47 50
5	4.0 Aca 5.1 5.2 5.3	ConclusionImage: Image: Im	40 41 42 42 42 42 44 47 50 50
5	4.0 Acad 5.1 5.2 5.3	ConclusionIntroductionIntroduction14Academic validation examples145.2.1Column under compression145.2.2Choi problem145.2.3Bracket structure145.2.3Bracket structure14Toward an industrial application145.3.1Elements of non-linear dynamic analysis145.3.2Numerical noise in frontal impact14	40 41 42 42 42 42 44 47 50 50 52
5	4.0 Acad 5.1 5.2 5.3	Image: ConclusionImage: Image: Im	40 41 42 42 42 42 44 47 50 50 52 58
5	4.0 Aca 5.1 5.2 5.3	Image: Conclusion Image: Conclusion Introduction Image: Conclusion Introduction Image: Conclusion Academic validation examples Image: Conclusion 5.2.1 Column under compression Image: Conclusion 5.2.2 Choi problem Image: Conclusion 5.2.3 Bracket structure Image: Conclusion 5.2.3 Bracket structure Image: Conclusion Toward an industrial application Image: Conclusion Image: Conclusion 5.3.1 Elements of non-linear dynamic analysis Image: Conclusion Image: Conclusion 5.3.3 Frontal impact crash scenarios Image: Conclusion Image: Conclusion Image: Conclusion 5.3.3.1 Application with available data (Initial experimental design) Image: Conclusion Image: Conclusion	40 41 42 42 42 42 44 47 50 50 52 58 61
5	4.0 Aca 5.1 5.2 5.3	Image: Conclusion Image: Conclusion Introduction Image: Conclusion Introduction Image: Conclusion Academic validation examples Image: Conclusion 5.2.1 Column under compression Image: Conclusion 5.2.2 Choi problem Image: Conclusion 5.2.3 Bracket structure Image: Conclusion 5.2.3 Bracket structure Image: Conclusion Toward an industrial application Image: Conclusion Image: Conclusion 5.3.1 Elements of non-linear dynamic analysis Image: Conclusion Image: Conclusion 5.3.3 Frontal impact crash scenarios Image: Conclusion Image: Conclusion Image: Conclusion 5.3.3.1 Application with available data (Initial experimental design) Image: Conclusion Image: Conclusion	40 41 42 42 42 44 47 50 50 52 58 61 62
5	4.0 Aca 5.1 5.2 5.3	demic and industrial applications 1 Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1 5.2.2 Choi problem 1 5.2.3 Bracket structure 1 7 Toward an industrial application 1 5.3.1 Elements of non-linear dynamic analysis 1 5.3.2 Numerical noise in frontal impact 1 5.3.3 Frontal impact crash scenarios 1 5.3.3.1 Application with available data (Initial experimental design) 1 5.3.3.2 Application with enrichment of the initial dataset 1	40 41 42 42 42 42 42 47 50 50 52 58 61 62 68
5	4.0 Aca 5.1 5.2 5.3	demic and industrial applications 1 Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1 5.2.2 Choi problem 1 5.2.3 Bracket structure 1 Toward an industrial application 1 5.3.1 Elements of non-linear dynamic analysis 1 5.3.2 Numerical noise in frontal impact 1 5.3.3 Frontal impact crash scenarios 1 5.3.3.1 Application with available data (Initial experimental design) 1 5.3.3.1 Application with enrichment of the initial dataset 1 5.3.3.3 Concluding remarks 1 5.3.4 Application on a vehicle frontal impact 1	40 41 42 42 42 42 42 47 50 50 52 58 61 62 68 69
5	4.0 Aca 5.1 5.2 5.3 5.3	demic and industrial applications 1 Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1 5.2.2 Choi problem 1 5.2.3 Bracket structure 1 Toward an industrial application 1 5.3.1 Elements of non-linear dynamic analysis 1 5.3.2 Numerical noise in frontal impact 1 5.3.3 Frontal impact crash scenarios 1 5.3.3.1 Application with available data (Initial experimental design) 1 5.3.3.2 Application with enrichment of the initial dataset 1 5.3.3.3 Concluding remarks 1 7.4.1 Application on the sidemember subsystem 1	40 41 42 42 42 44 47 50 50 52 58 61 62 68 69 70
5	4.0 Aca 5.1 5.2 5.3 5.3	Introduction Introduction Academic validation examples Introduction 5.2.1 Column under compression Introduction 5.2.2 Choi problem Introduction 5.2.3 Bracket structure Introduction 7 Toward an industrial application Introduction 5.3.1 Elements of non-linear dynamic analysis Introduction 5.3.3 Frontal impact Introduction 5.3.3 Frontal impact crash scenarios Introduction 5.3.3.1 Application with available data (Initial experimental design) Introduction 5.3.3.2 Application with enrichment of the initial dataset Introduction 5.3.3.3 Concluding remarks Introduction Introduction 7.4.1 Problem formulation Introduction Introduction	40 41 42 42 42 42 42 47 50 50 52 58 61 62 68 69 70 70
5	4.0 Aca 5.1 5.2 5.3 5.3	Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1 5.2.2 Choi problem 1 5.2.3 Bracket structure 1 Toward an industrial application 1 5.3.1 Elements of non-linear dynamic analysis 1 5.3.2 Numerical noise in frontal impact 1 5.3.3 Frontal impact crash scenarios 1 5.3.3.1 Application with available data (Initial experimental design) 1 5.3.3.2 Application with enrichment of the initial dataset 1 5.3.3.3 Concluding remarks 1 7.4.1 Application on the sidemember subsystem 1 5.4.1.1 Problem formulation 1 5.4.1.2 Results and discussion 1	40 41 42 42 42 44 47 50 50 52 58 61 62 68 69 70 70 70
5	4.0 Aca 5.1 5.2 5.3 5.3	Introduction Introduction Academic validation examples Introduction 5.2.1 Column under compression Introduction 5.2.2 Choi problem Introduction 5.2.3 Bracket structure Introduction 5.2.3 Bracket structure Introduction 5.2.4 Choi problem Introduction 5.2.5 Bracket structure Introduction 5.2.6 Numerical application Introduction 5.3.1 Elements of non-linear dynamic analysis Introduction 5.3.2 Numerical noise in frontal impact Introduction 5.3.3 Frontal impact crash scenarios Introduction 5.3.3.1 Application with available data (Initial experimental design) Introduction 5.3.3.2 Application with enrichment of the initial dataset Introduction 5.3.3.3 Concluding remarks Introduction Introduction 5.4.1 Application on the sidemember subsystem Introduction Introduction 5.4.1.2 Results and discussion Introduction Introduction Introduction 5.4.2 Application on a full automotive bod	40 41 42 42 42 42 42 42 42 42 50 50 50 50 50 50 50 50 50 50 50 50 50
5	4.0 Aca 5.1 5.2 5.3 5.3	demic and industrial applications 1 Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1 5.2.2 Choi problem 1 5.2.3 Bracket structure 1 Toward an industrial application 1 5.3.1 Elements of non-linear dynamic analysis 1 5.3.2 Numerical noise in frontal impact 1 5.3.3 Frontal impact crash scenarios 1 5.3.3.1 Application with available data (Initial experimental design) 1 5.3.3.2 Application with enrichment of the initial dataset 1 5.3.3.3 Concluding remarks 1 5.4.1 Application on the sidemember subsystem 1 5.4.1.1 Problem formulation 1 5.4.1.2 Results and discussion 1 5.4.2 Application on a full automotive body structure 1 5.4.2.1 Presentation of the finite element model 1	40 41 42 42 42 42 42 42 47 50 50 52 58 61 62 68 69 70 70 70 70 74 74
5	4.0 Aca 5.1 5.2 5.3 5.3	demic and industrial applications 1 Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1 5.2.2 Choi problem 1 5.2.3 Bracket structure 1 Toward an industrial application 1 5.3.1 Elements of non-linear dynamic analysis 1 5.3.2 Numerical noise in frontal impact 1 5.3.3 Frontal impact crash scenarios 1 5.3.3.1 Application with available data (Initial experimental design) 1 5.3.3.2 Application with enrichment of the initial dataset 1 5.3.3.3 Concluding remarks 1 5.4.1 Application on a vehicle frontal impact 1 5.4.1.1 Problem formulation 1 5.4.1.2 Results and discussion 1 5.4.2.1 Presentation of the finite element model 1 5.4.2.2 Problem formulation 1	40 41 42 42 42 42 42 42 42 42 50 50 52 58 61 62 68 69 70 70 70 70 74 74 74
5	4.0 Aca 5.1 5.2 5.3 5.3	demic and industrial applications 1 Introduction 1 Academic validation examples 1 5.2.1 Column under compression 1 5.2.2 Choi problem 1 5.2.3 Bracket structure 1 5.2.3 Bracket structure 1 Toward an industrial application 1 5.3.1 Elements of non-linear dynamic analysis 1 5.3.2 Numerical noise in frontal impact 1 5.3.3 Frontal impact crash scenarios 1 5.3.3.1 Application with available data (Initial experimental design) 1 5.3.3.2 Application with enrichment of the initial dataset 1 5.3.3.3 Concluding remarks 1 5.4.1 Application on a vehicle frontal impact 1 5.4.1.1 Problem formulation 1 5.4.1.2 Results and discussion 1 5.4.2 Problem formulation 1 5.4.2.1 Presentation of the finite element model 1 5.4.2.2 Problem formulation 1 5.4.2.3 Results and discussions <td< td=""><td>40 41 42 42 42 42 44 47 50 50 52 58 61 62 68 69 70 70 70 70 70 74 74 74 78</td></td<>	40 41 42 42 42 42 44 47 50 50 52 58 61 62 68 69 70 70 70 70 70 74 74 74 78

6 Conclusion

Α	Initi	al desi	gn of experiments: A short state-of-the-art	191
	A.1	Detern	ninistic design of experiments	191
		A.1.1	Factorial designs	191
		A.1.2	General linear specific designs	192
		A.1.3	Bayesian designs	192
		A.1.4	Geometric criteria based designs	193
	A.2	Rando	om designs	194
		A.2.1	Latin hypercube sampling	194
		A.2.2	Orthogonal arrays	196
	A.3	Quasi	-random designs	196
		A.3.1	A short introduction to discrepancy	197
		A.3.2	Some low-discrepancy sequences	199
			A.3.2.1 van der Corput sequence	199
			A.3.2.2 Hammersley sequence	199
			A.3.2.3 Halton sequence	200
			A.3.2.4 Faure sequence	202
		A.3.3	(t, m, s)-nets and (t, s) -sequences	203

B Implementation of the (1 + 1)-CMA-ES algorithm for constrained optimization problems 207

References

211



INTRODUCTION

Context of the study

Man-made global warming has been constantly increasing since the industrial revolution and the need to burn fuel to power machines, to heat buildings or to transport goods and people. The latter is a major cause of carbon dioxide (CO_2) emissions, the main greenhouse gas. Globally speaking, passenger cars are responsible for 10% of CO_2 emissions from human sources, with an average of 124 g of CO_2/km per car in Europe (PSA Peugeot Citroën, 2012). In an attempt to reduce our ecological footprint, the European Union has adopted a legislation in 2009 mandating car manufacturers to lower their car average CO_2 emissions to 95 g of CO_2/km per car by 2020.

As a leading car manufacturer, PSA Peugeot Citroën has been permanently working towards meeting this binding requirement. Recently the company has taken back the lead in terms of emissions reductions with an average of 110 g of CO_2/km per car. Various strategies have been adopted at different levels to reduce the gas emissions of PSA Peugeot Citroën cars. Among them, improvement of engines performances, lightweight design, improvement of rolling resistance, aerodynamic efficiency and energy management. For lightweight design, beside the use of light materials (*e.g.* composite or high-strength steel), the optimization of the thicknesses of parts belonging to the body-in-white is a promising approach for weight saving. In general, saving 10 kg approximately leads to reducing the CO_2 emissions by 1 g.

We are interested here in the latter strategy of weight savings. The body-in-white or automotive body structure is a collection of metal sheets that are welded together. A main concern for engineers is to optimally distribute tens of metal sheets thicknesses while satisfying the safety and comfort requirements (*e.g.* fatigue, vibro-acoustic behavior, crashworthiness, etc.). These requirements are assessed by high-fidelity models whose simulations may be extremely time-consuming, *i.e.* up to 24 hours per simulation. For optimization study, where several allocations of thicknesses are evaluated, a classical approach through finite element (FE) simulations is not an option given the project lead times. So far, the approach considered is the substitution of these FE simulations by so-called *metamodels* or *surrogate models*. In a nutshell, a metamodel is an analytical function which aims at approximating a black-box function (*i.e.* whose inputs/outputs are only known pointwise) by learning over a limited set of experiments. This has shown to be effective for optimization against particular specifications (*e.g.* lateral impact and vibro-acoustic analysis), thus leading to fair weight savings. However, for some other specifications such as the frontal impact, the outputs of the simulation models are particularly non-linear and noisy. That, in conjunction with the high dimensionality of the problem (*i.e.* the number of parts to optimize), makes the accuracy of the built metamodels insufficient for a proper optimization. The main objective of this PhD thesis is to propose a methodology for high-fidelity adaptive metamodeling, so as to open a path to further weight savings. The physical specifications which are to be approximated by surrogate models in a general framework are introduced in the following.

Physical specifications for car design

Cars are designed and manufactured to meet customers' needs. These include comfort, safety and price. To ascertain that these requirements are fulfilled, car manufacturers may physically test their products using prototypes. However, with the growing demand, an exhaustive test approach is neither cost-effective nor time-efficient. So, engineers rely on numerical methods, more specifically finite element analyses, to simulate the vehicles behaviors.

In a design cycle as given by the so-called *systems engineering*, the customer needs are translated into physical specifications that are simulated. The lightweight design through optimization mainly investigates two specification families. The first one is related to the ride comfort of passengers and is analyzed under the framework of *Noise Vibration & Harshness* (NVH). The associated analyses are essential to the car manufacturer since they are viewed by potential customers as a guarantee of the car quality. They are henceforth considered in the early design stages and especially during weight optimization. The second one, which concerns us here, relates to the occupant safety and is analyzed through finite element explicit non-linear dynamic analysis.

Safety requirements

With the ever-increasing road fatalities, safety has become a concern of utmost importance to car manufacturers. PSA Peugeot Citroën works on three complementary aspects of safety:

- Primary safety which aims at preventing accidents;
- Secondary safety which aims at protecting the occupants in the event of an accident;
- Tertiary safety which deals with emergency.

As a consequence of the increase in road fatalities, European legislation requires that all vehicles pass some minimal crash-tests before any commercialization. The *European New Car Assessment Program* (Euro NCAP) goes further with more constraining requirements in order

to promote safety among car manufacturers. Hence, they provide customers with independent safety assessment of new vehicles. The main crash-tests are:

• **Frontal impact:** The scenario involves a car hitting a deformable barrier at the speed of 64 km/h. Since the most frequent type of roads crashes involves only a part of the vehicles fronts, the barrier is offset with an overlap of 40%. The aim of the designer is to control the crash scenario by absorbing as much as possible the vehicle kinetic energy through plastic deformation of the front end metal sheets. The idea is also to lower the crash deceleration pulse below the limit of the human body tolerance.



Figure 1.1: Frontal impact test (Courtesy of Euro NCAP)

• **Side impact:** The side impact resistance of a vehicle is assessed through the car-to-car and pole side impacts. The former is simulated by propelling a mobile deformable barrier at the speed of 50 km/h into the driver's door. The latter consists of propelling the car sideways into a fixed narrow pole at the speed of 29 km/h. They represent the second most frequent types of cars accidents. The designer objective is to reduce as much as possible the car body parts intrusions into the occupant compartments.



Figure 1.2: Car-to-car side impact test (Courtesy of Euro NCAP)

• **Pedestrian impact:** This test replicates accidents involving child and adult pedestrians. Three zones are investigated: the lower leg, the upper leg and the head. The first assesses that the bumper is pedestrian-friendly, *i.e.* it deforms at the contact of the leg and spreads the crush force along the entire leg to prevent serious injuries. The upper leg and head tests assess that the bonnet (the hinged cover of the engine compartment) could be sufficiently bent to absorb the energy at the impact. To achieve this, it is necessary to provide enough space between the bonnet and the stiff components beneath it. Some cars are even provided with pop-up bonnets which lift when an accident involving a pedestrian is detected.



(a) Lower leg

(b) Upper leg

(c) Head

Figure 1.3: Pedestrian impact tests (Courtesy of Euro NCAP)

The limitations with frontal impact

The simulation of frontal impact is realized at PSA with the RADIOSS software, a finite element solver developed by Altair Engineering. It is a challenging non-linear dynamic problem which is handled by an explicit formulation. The differential equations that govern the physical be-

havior of the car and barrier during the impact are solved in a weak form (Du Bois et al., 2004). This involves both spatial and time discretization of the problem. With an appropriate choice of the parameters for the time discretization, the problem may be solved using an iterative scheme where only the mass matrix is inverted. This is known as the *explicit formulation* in contrast to the *implicit* one and only requires the solution of a system of uncoupled equations. At each iteration, nodal displacements are calculated as a function of the previous step. For the short time impacts in the automotive industry (usually around 120 ms), this explicit formulation has shown to be the most robust.

However, it also comes with some drawbacks for there is *numerical scatter* in the finite element simulations. In fact, the code by itself is deterministic, *i.e.* running two identical models on the same computer gives the same results. In contrast, any change in the input, even insignificant, leads to dramatic variations of the outputs. For instance, moving infinitesimally the position of one node of the mesh may produce significantly different results. The reasons for this lie in the chaotic behavior of crash. Schematically, crash can be seen as a succession of events in a short time-lapse featuring bifurcations triggered by input variability which ends up in different solution paths or crash scenarios. The typical example is the *on-off contact, i.e.* whether a contact between two parts occurs or not or any inversion in the chronology of the events. In the same way, buckling of parts such as the sidemember which may bend in a given direction or crush axially can alter the crash scenario. On top of that, there are non-physical parameters, such as spatial and time discretization, contact detections, round-off errors which add to the variability of the outputs. Many studies have been carried out in order to identify the origins of the instability in crash simulations (Wauquiez and Zeitouni, 2008; Thole and Mei, 2003). They can be mainly grouped into physical and numerical causes (Roux et al., 2006).

Furthermore, crash is known to be a chaotic phenomenon. This means that the slightest variation in the initial conditions leads to dramatic changes in the crash outcome. The explanations are twofold. First is the *physical variability* of the initial conditions. Uncertainties are indeed inherent to real life systems. As such, two crash tests of the same car are always different, despite the scrutiny in the test protocol. The difference may come from the manufacturing process, *e.g.* parts dimensional tolerances or spot welding. They may also result from the crash protocol itself, *e.g.* variability in the initial speed, the position or stiffness of the barrier.

All these causes put together make the frontal impact simulation noisy. Engineers should however deliver stable models for a proper utilization. At PSA Peugeot Citroën, the stability of a frontal impact simulation is assessed through Monte Carlo sampling. In the case of physical scatter, parameters that are identified to be the most relevant are considered random with known distributions. For numerical scatter, the Monte Carlo sampling is set up by infinitesimally perturbing the model mesh nodes. Figure 1.4 illustrates results from a physical scatter Monte Carlo sampling, which consists of a sample set of 100 points: In the left panel, timecurves of one output of interest (herein a force) and in the right one a histogram of the distribution of the specific required value (here the maximum force). In the same fashion, results of a numerical scatter Monte Carlo sampling are shown in Figure 1.5. Now an acceleration in a given point is considered.

The idea with these simulations is to assess the scatter in the outputs. More specifically, the numerical scatter analysis evaluates the stability and thereby the quality of the finite element models whereas the physical scatter analysis checks their robustness. The variability must be kept below a given threshold, otherwise misconception is assumed. As for the present PhD thesis, this observation motivates the methodology we propose for robust optimization of body-

1. INTRODUCTION

in-white under crashworthiness constraints.



Figure 1.4: Example of a Monte Carlo simulation with different models considering the physical parameters variability.



Figure 1.5: Example of a Monte Carlo simulation with different models considering the numerical parameters variability.

Objective of the thesis and outlines

Beside the noisy nature of crash simulation mentioned above, other difficulties arise when building surrogate models to substitute them. In fact, the outputs are highly non-linear and the dimensionality of the problem (*i.e.* the number of parts to optimize) is relatively large. For these reasons, frontal impact is not directly considered in current lightweight designs. In fact, all other specifications but frontal impact are considered during optimization. Among the solutions, the ones which satisfy frontal impact-related constraints are kept.

The aim of this thesis is therefore to propose a methodology to overcome these difficulties and thereby include frontal impact in the optimization process. This will allow the designers to increase the weight savings through body-in-white lightweight design. This manuscript is organized as follows.

Chapter 2 is devoted to structural design optimization. After a brief introduction of design optimization, the limitations of a deterministic approach are pointed out. Then, a probabilistic viewpoint is introduced with its associated approaches. They involve robust and reliability-based design optimization (RBDO). The state-of-the art of structural reliability methods are reviewed. In the last section, we propose an alternative formulation of RBDO which is based on quantiles as a measure of conservatism.

Chapter 3 is mainly focused on a literature review of two types of surrogate modeling techniques, namely support vector machines and Kriging. They are selected as they are, to the author's knowledge, the most appropriate ones to account for noisy data. A specific attention is paid to the calibration of their hyperparameters.

Chapter 4, which is the core of this manuscript, is about surrogate-based optimization. Kriging is considered due to its local accuracy measure. Adaptive techniques, which consist in iteratively updating Kriging models so as to ensure fitting quality only in regions that matter for optimization, are investigated. A substantial part of this chapter is dedicated to a literature review of the methods allowing to achieve this. The surrogate and quantile-based RBDO problem is formulated. A methodology with two stages of enrichment is proposed. The first is global and the second, which is local, is coupled with optimization. In this work, a global search algorithm is considered, in particular the covariance matrix adaptation evolution strategy (CMA-ES).

Chapter 5 eventually validates the proposed methodology. In the first part, analytical models are considered with different configurations for the probabilistic models. The results provided by our methodology are compared to those found in the literature on benchmark examples. A detailed study on two car models under frontal impacts is then presented. The effects of numerical and physical scatter are investigated. Eventually, we apply the proposed methodology to the lightweight design of a car while considering uncertainties coming from the crash test protocol.



STRUCTURAL DESIGN OPTIMIZATION

Contents

2.1 Introduction to design optimization		uction to design optimization
	2.1.1	Deterministic design optimization
	2.1.2	An overview of optimization algorithms 12
	2.1.3	Beyond the deterministic framework
2.2	Optim	ization under uncertainties 24
	2.2.1	Robust design optimization
	2.2.2	Reliability-based design optimization
	2.2.3	Concluding remarks
2.3	Quant	ile-based RBDO
	2.3.1	Motivation
	2.3.2	Formulation
	2.3.3	Analytical examples
2.4	Conclu	ision

Introduction

The need to find the most effective and efficient solution among various alternatives has always been a concern for researchers in different disciplines such as economy, biology or engineering. Newton and Leibniz's works on calculus during the 17th century provided a unified mathematical framework for solving optimization problems in these disciplines. In mechanical engineering, the concern is put on *structural design optimization* which is the task of finding the best performance of load-carrying structures while preserving their integrity. Examples of such structures are aircrafts, automobiles or buildings. The early contributions to this field can be traced back to the middle of the 20th century (Schoofs, 1993). First, Prager and Taylor (1968) and Venkayya et al. (1968) introduced the so-called *optimality criteria* for the optimization of simple structures such as beams, plates and trusses using respectively analytical and numerical forms. About the same time, Schmit (1960) laid the foundations of the coupling between finite element analysis and optimization for structural design in an iterative scheme.

With the advent of computer-aided engineering, the designers have acquired the flexibility to explore alternate concepts or designs using virtual prototypes. In this context, optimization does not only serve as a mean to reduce a cost (*e.g.* by reducing a weight) but also allows the designers to better understand their products, *i.e.* their strengths and weaknesses. All this ultimately lead to a drastic reduction of the cost of goods manufactured.

In general, the optimization problems can be classified into three categories: *size, shape* and *topology optimization*. This classification is defined according to the types of design variables. In size optimization, material properties or geometrical dimensions such as thicknesses and cross-sectional areas are considered. Most of the applications fall under this category. Shape optimization leverages over parameters describing the shape of some structural parts. Finally, topology optimization focuses on allocating matter and voids in a structure so as to optimally distribute the load paths. It is the most general case and may serve to identify a starting point for size or shape optimization.

In this thesis, we are focusing on size optimization. Our ultimate aim is the lightweight design of an automotive body structure where the sizing variables are the thicknesses of some set of metal sheets. For this application, numerous sources of uncertainties can be identified. For a safe design, it is necessary to consider them in the process of optimization. This is done here through a probabilistic approach. In this chapter, we first describe the general framework for optimization in a deterministic context. Then, we discuss the means for optimization under uncertainties, while specifically focusing on robust and reliability-based design optimization (respectively RDO and RBDO). Finally, we introduce a special case where the probabilistic content of RBDO is handled through quantiles. An associated formulation of quantile-based RBDO is henceforth proposed.

2.1 Introduction to design optimization

2.1.1 Deterministic design optimization

For design optimization, we seek to minimize a cost, generally expressed in a mathematical form by a merit function. This is usually done by iterating on alternate admissible designs, the space of which is delimited by some constraint functions. A general formulation of a determin-

istic design optimization problem may read as follows:

$$d^* = \arg\min_{d\in\mathbb{D}} \mathfrak{c}(d) \quad \text{subject to:} \begin{cases} \mathfrak{f}_j(d) \le 0, & \{j=1,\ldots,n_s\}\\ \mathfrak{g}_k(d,z) \ge 0, & \{k=1,\ldots,n_h\}. \end{cases}$$
(2.1)

where a *cost* function c is minimized with respect to *design* variables $d \in \mathbb{D} \subset \mathbb{R}^{s_d}$ while satisfying to a collection of n_s soft and n_h hard constraints.

The hard constraints, $\mathfrak{g} = {\mathfrak{g}_k, k = 1, ..., n_h}$, denote the requirements the structure needs to meet to perform adequately. In structural design, they ensure the mechanical integrity of the structure and are usually assessed through finite element methods. This might be for instance the maximal value of a nodal displacement or that of an element stress. In this case, they can be cast as $\mathfrak{g}_k(d, z) = \overline{\mathfrak{g}}_k - \mathcal{M}_k(d, z)$, where $\overline{\mathfrak{g}}_k$ is a threshold not to be exceeded and \mathcal{M}_k the mathematical or black-box function representing the response of the structure. Beside the design variables d, the limit state function is explicitly written here with respect to another type of parameter, the so-called *environmental* variables z. In DDO, their values actually remain constant. We however introduce them here anticipating the probabilistic case (next section), where they can vary around their nominal values. Likewise, the soft constraints are introduced separately from the hard ones despite they are both treated in the same way in DDO. In fact, the soft constraints are simple analytical functions bounding the design space D.

In Eq. (2.1), equality constraints have been omitted without loss of generality. Moreover, we have considered a single objective function. However, in a wide number of applications, many objectives might be of interest. This is known as *multi-criteria* optimization as opposed to *monocriterion* optimization introduced here. In such a case, the definition of the optimal design is ambiguous. One seldom finds a solution with the best merit for all the objective functions. Compromising becomes necessary. Two alternatives are possible: a priori choice and a posteriori choice (Baudoui, 2012). In the approaches qualified as a priori, one has to rank the objective functions with respect to their relative importances. This way, the multi-objective problem is transformed into a mono-objective one. The simplest way to do this is to consider a weighted combination of the initial objective functions as the new one to minimize. The difficulty here is of course the allocation of the weights. Alternatively, lexicographic ordering methods can be used so as to solve a sequence of mono-objective optimization problems in order of priority. That is, the optimization is initialized by solving a sub-problem where only the first objective function is considered. The others are set as inequalities. Once a solution is found, the first is set as equality, the second optimized and the others set as inequalities. The procedure is repeated until all objective functions have been optimized. In any of these approaches, the result is highly correlated with the preference settings. In a posteriori choices, the solution is rather based on the notion of *Pareto set*, which consists of a subset of feasible solutions which do not dominate each other. A solution is said to dominate another one in the sense of Pareto if it is not worse in all the criteria and has a better merit in at least one of them. A method based on Pareto-optimality is more interesting from the designer's viewpoint since it allows him to have an overview of the solutions, a valuable information for decision making.

To be exhaustive, we shall also mention *multidisciplinary design optimization* (MDO) concerned with complex structures such as aircrafts or automobiles. In MDO two or more disciplines are considered. There may exist coupling between them through shared design variables. This thesis is concerned with the lightweight design of automotive body structures under multiple constraints. These constraints come from several disciplines such as crashworthiness or vibro-acoustic analyses. However, there is no coupling at all between the disciplines. This allows us

to treat them separately. As a matter of fact, we will only focus on the crash-related constraints for the application of this thesis.

2.1.2 An overview of optimization algorithms

The solution of an optimization problem such as in Eq. (2.1) generally resorts to numerical methods and search algorithms. An exhaustive list of these algorithms is hardly possible because of their diversity and quantity. In this section, we review some of the widely used ones and briefly introduce those used throughout this thesis. The choice of a given algorithm mostly depends on the class of problem at hand. Whether the objective function is linear, quadratic, smooth, multi-modal, differentiable or not highly affects the performance of a given algorithm. This performance is generally measured by the number of functions evaluations required before convergence to a solution. Wolpert and Macready (1997) show, through the *no free lunch theorem* for optimization, that there is not such case where a single algorithm performs well on all class of problems. There is indeed a strong connection between an algorithm effectiveness and the problem it solves, which makes it difficult to a priori decide which one to use.

Beside, when solving an optimization problem, one is often concerned with finding the global minimum. Most algorithms only ensure the convergence to a local minimum despite some have a greater likelihood of not being trapped in local minima. However, in the case where the problem is convex (*i.e.* the objective function and the feasible set are convex), the local minimum is necessarily a global one. But in general, such assumption is not possible. It is just empirically accepted that gradient-based approaches are likely to be trapped in local minima while gradient-free approaches offer greater chances to find a global optimum.

In this section, we present some general-purpose algorithms for solving optimization problems. Despite most of the problems we are concerned with are constrained, the algorithms we introduce focus on unconstrained cases without loss of generality. The consideration of constraints may be implemented by various techniques with only minor adaptations of the algorithms such as penalty methods or accept-reject approaches.

2.1.2.1 Gradient-based methods

Gradient-based methods make use of the derivatives of the objective function to locally search for better designs. Using them requires that the first-order and often the second-order derivatives exist. One of the classical gradient-based approaches is the family of Newton methods that we briefly describe below.

Newton method

The Newton method assumes that the cost can be locally approximated by a quadratic function, more specifically by its second-order Taylor expansion. This expansion around a given point d writes:

$$\mathfrak{c}(d + \Delta d) \approx \mathfrak{c}(d) + \Delta d^{T} \nabla \mathfrak{c}(d) + \frac{1}{2} \Delta d^{T} H(d) \Delta d, \qquad (2.2)$$

where Δd represents a small increment around d and $H \equiv \nabla^2 \mathfrak{c}$ denotes the Hessian matrix or second-order derivatives of \mathfrak{c} . ∇ stands for the gradient operator.

The idea here is to find the optimal value of the increment that leads to a stationary point of \mathfrak{c} . This is achieved by finding the root of the gradient $\nabla \mathfrak{c}$ with respect to Δd , which can be

computed from Eq. (2.2). In mathematical terms, this leads to:

$$\Delta d = -H^{-1}(d) \nabla \mathfrak{c}(d).$$
(2.3)

Starting from an initial guess $d^{(0)}$, the method proceeds by iteratively updating the design following Eq. (2.3). At iteration *k*, the updating formula reads:

$$d^{(k+1)} = d^{(k)} - H^{-1}(d) \nabla c(d).$$
(2.4)

This converges in only one iteration if the objective function is quadratic. However in general, the convergence to a local minimum is not sure. Starting from a guess point which is too far from a local minimum may lead the algorithm to diverge. In fact the Hessian matrix is required to be positive definite. But far from the minimum, this condition might not be fulfilled. Beside, each iteration requires the inversion of the Hessian of size $s_d \times s_d$, where s_d is the number of design variables. This operation can be expensive for high dimensional problems. Often a simple approximation of the Hessian and its inverse is enough for the algorithm to converge. Methods which rely on these approximations are known as *quasi-Newton* methods.

Quasi-Newton methods

The class of quasi-Newton methods aims at bypassing the difficulty associated to the inversion of the matrix in Eq. (2.4). One of the simplest approach is to replace the Hessian by the identity matrix *I*. This actually means that the objective is locally approximated by a linear function. The convergence can subsequently be slower. This approach is also known as the *steepest descent* method. The associated updating formula reads:

$$d^{(k+1)} = d^{(k)} - \nabla c(d).$$
(2.5)

Another approach consists in approximating the Hessian by rank-one updates of the objective function's gradients. The update is made so as to gradually converge to the true Hessian as the number of iterations grows. One of the most popular approach is the so-called *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) algorithm (Shanno, 1970) which updates the Hessian approximation as follows:

$$\boldsymbol{B}^{(k)} = \boldsymbol{B}^{(k-1)} + \frac{\left(\boldsymbol{y}^{(k-1)} - \boldsymbol{B}^{(k-1)}\Delta \boldsymbol{d}^{(k-1)}\right) \left(\Delta \boldsymbol{d}^{(k-1)}\right)^{T}}{\left(\Delta \boldsymbol{d}^{(k-1)}\right)^{T}\Delta \boldsymbol{d}^{(k-1)}},$$
(2.6)

where $\Delta d^{(k-1)} = d^{(k)} - d^{(k-1)}$, $y^{(k-1)} = \nabla \mathfrak{c} \left(d^{(k)} \right) - \nabla \mathfrak{c} \left(d^{(k-1)} \right)$ and $B^{(k-1)}$ is the approximation of the Hessian at the point $d^{(k-1)}$. Its initial value is set at $B^{(0)} = I$.

Other methods such as the conjugate gradient can also be used for unconstrained gradientbased optimization. In this thesis, the BFGS algorithm is used to find the optimal hyperparameters of the Kriging model introduced in the next chapter.

2.1.2.2 Gradient-free methods

Local search

Gradients by construction are well fitted to local search. When they are not available, say because the objective function is discontinuous, local search is still possible. The family of so-called *pattern search* algorithms explores the design space without any derivative information

(Dennis et al., 1994). In essence, pattern search methods rely on moving a pattern through the design space so as to iteratively find better designs. In fact, the objective function is sampled on some points around the current design following a predefined pattern. If a point with a better merit is found, this step is considered successful. If not, the pattern size is reduced and this exploratory move is performed again. After a successful step, the pattern is moved so that its base corresponds to the newly found best point. The process is eventually restarted until convergence is achieved. This is the crudest approach. In most case, many tricks are used to make the search robust, leading to several variants of pattern search.

Global search

In contrast to the previous methods, global search algorithms do not rely on local sensitivities of the cost function. Instead, they proceed by evaluating the cost function in the entire design space in order to find regions with best merit. Most of them mimic natural selection or biological mechanisms *e.g.* genetic algorithm, particle swarm or ant colony optimization. Due to their wide variety, an exhaustive enumeration would not be possible. We focus here on those which are used throughout this thesis for various purposes such as model calibration.

Simulated annealing

Simulated annealing implements a very elegant algorithm which allows the optimizer to escape from local minima. It is inspired from annealing in metallurgy, a physical process by which a material is initially heated and then slowly cooled so that at any given temperature, thermodynamical equilibrium is reached. By this heat treatment, the particles progressively arrange themselves eventually reaching a state of minimum energy (van Laarhoven and Aarts, 1987). This increases the material ductility, making further treatment, such as stamping or forming, easier. Metropolis et al. (1953) proposed a Monte Carlo algorithm to simulate the cooling of the material and its successive thermodynamical equilibrium states. The Metropolis algorithm proceeds by generating a new state from the current one thanks to a predefined transition mechanism. The energy of the resulting state is then compared with the previous one. If it is lower, then this state is accepted as the new one. If in contrast it is higher, the new state is only accepted with probability $p = \exp\left(-\frac{\Delta E}{T}\right)$, where ΔE is the difference in energy and T the temperature. Kirkpatrick et al. (1983) adopted this technique to propose a combinatorial optimization algorithm. The analogy is the following: the states correspond to different configurations or designs of the system, the energy is the cost function and the temperature is a free parameter that controls the algorithm convergence to a global minimum. Higher temperature allows the algorithm to adopt uphill moves, hence preventing premature convergence to a local minimum. One of the greatest issues here is the setting of the annealing schedule, in other words, the scheme by which the temperature is decreased. Usually, the algorithm starts with a high value of T, then the temperature is decreased according to a predefined scheme. In the early iterations, the algorithm explores the design space by allowing uphill moves and in the latter ones it focuses on small regions to decrease the cost function. The convergence is achieved when temperature is equal to 0 (no possibility of accepting any new state as the acceptance probability is 0) or when the maximum budget for evaluating the cost function has been reached.

In this thesis, simulated annealing is used as an optimization algorithm to construct optimal Latin hypercubes as described in Appendix A.

• Genetic algorithm

Genetic algorithm (Goldberg, 1989) was developed by Holland (1975) who was inspired by the biological mechanisms involved in natural selection. The basic idea is to evolve a population of candidate solutions to a problem by using variations-inducing operators so as to select those with expected better performance. The terminology from genetic science is also adapted. The cost function is therefore translated into a fitness function that is to be maximized. Any candidate solution is called a chromosome, most often coded by bit strings. The algorithm proceeds by first generating a set of chromosomes and evaluates their fitness. Afterwards, genetics-inspired operators are used for evolution to the next generation (Mitchell, 1995). The first one is selection where some chromosomes are selected to be parents. The higher their fitness, the more likely they are to be selected. The second operator is *crossover*, a recombination technique used to generate offsprings with the genetic material of the fittest chromosomes. Two offsprings are actually created by randomly combining subsequences of the chromosomes of two selected parents. The final operator is *mutation* which consists in randomly flipping the bits of the offsprings. It occurs with some probability, usually small. This operator allows the algorithm to spring out from a local minimum, so its probability of occurrence can be set high in the early iterations and low in the latter ones. By the end of these operations, a new generation of candidates emerges and replaces the old one. The process is repeated until convergence.

According to the type of problem or for the matter of improving performance, a wide variety of genetic algorithms have been developed. Usual instances are multi-island genetic algorithms or the non-dominated sorting genetic algorithm (NSGA-II). In the former, the population is split into many islands which are handled separately. The latter is well suited to multi-criteria problems. In this thesis, genetic algorithms are used as one of the methods to calibrate the Kriging parameters (see section 3.4.4.1).

• Cross-entropy method for optimization

The cross-entropy method was initially developed by Rubinstein (1997) as a simulation technique for the estimation of probabilities of rare events. Rubinstein and Davidson (1999) later adapted it for the optimization of multi-extremal problems. Let us restate the optimization problem in terms of maximization of an objective instead of minimization. In other words we search:

$$\gamma^* = \arg\max_{d \in \mathbb{D}} \mathfrak{c}\left(d\right). \tag{2.7}$$

The idea is to associate this optimization problem with a rare event estimation problem which is tackled by an adaptive approach (Kroese et al., 2006). For Eq. (2.7), this so-called *associated stochastic problem* reads:

$$\ell(\gamma) = \mathbb{P}_{v}\left(\mathfrak{c}\left(\mathbf{D}\right) \geq \gamma\right) = \mathbb{E}_{v}I_{\{\mathfrak{c}(\mathbf{D}) \geq \gamma\}},\tag{2.8}$$

where **D** is a random variable drawn following a distribution $f(\bullet, v), v \in \mathcal{V}$. \mathbb{P}_v and \mathbb{E}_v are respectively probability measures and expectation over v.

If γ is close to γ^* , $f(\bullet, v^*)$ assigns most of its probability mass in the area around d^* (Boer et al., 2004). In this case, $f(\bullet, v^*)$ may be used to sample an approximate solution to Eq. (2.7). On the other hand, estimating ℓ for a value of γ close to γ^* makes { $\mathfrak{c}(D) \ge \gamma$ } a rare event. Resorting to simple Monte Carlo simulation would require a huge amount of samples. The cross-entropy method suggests to solve this through a *multi-level* approach. A sequence of tuples { (γ_k, v_k) } converging to (γ^*, v^*) is therefore generated. More specifically, the algorithm proceeds as follows. Let us define a sufficiently small quantile ϱ and an initial set of parameters

 v_0 . At iteration k, γ_k is defined such that:

$$\mathbb{P}_{\nu_{t-1}}\left(\mathfrak{c}\left(\boldsymbol{D}\right) \geq \gamma_{k}\right) = \varrho,\tag{2.9}$$

where $D \sim f(\bullet, v_{k-1})$. The second stage consists in updating v_k , which can be done as follows:

$$\boldsymbol{\nu}_{k} = \arg \max_{\boldsymbol{\nu}} \mathbb{E}_{\boldsymbol{\nu}_{k-1}} I_{\{\boldsymbol{\mathfrak{c}}(\boldsymbol{D}) \geq \gamma\}} \ln f\left(\boldsymbol{D}, \boldsymbol{\nu}\right).$$
(2.10)

For the developments getting to this final equation, refer to Boer et al. (2004) (p. 9 - 12). This updating formula can be obtained analytically for a certain type of PDFs such as the natural exponential family (normal, Gamma, Poisson, etc.). In the case of normal distribution, *v* stands for the two parameters of the distribution *i.e.* mean and standard deviation.

The algorithm therefore consists in the following steps:

- 1. Set the parameters of the initial distribution: $v_0 = \{\mu_0, \sigma_0\}$ and k = 1;
- 2. Sample a population $\{d_1, d_2, ..., d_N\}$ from the multivariate normal distribution: $d_i \sim \mathcal{N}(\mu_{k-1}, \sigma_{k-1}^2)$ for any $i = \{1, ..., N\}$.
- 3. Evaluate the objective function on these samples: $c_i = c(d_i)$
- 4. Compute the (1ϱ) -quantile of the performance and define the set of indices $\mathcal{I} = \{i \in \{1, ..., N\} | \mathfrak{c}_i \ge \gamma_k\}$
- 5. Compute the statistics of the elite samples

$$\widetilde{\mu}_{k} = \frac{1}{N_{\varrho}} \sum_{i \in \mathcal{I}} d_{i},$$

$$\widetilde{\sigma}_{k}^{2} = \frac{1}{N_{\varrho}} \sum_{i \in \mathcal{I}} (d - \widetilde{\mu}_{k})^{2},$$
(2.11)

where N_{ρ} is the length of \mathcal{I} .

6. Update the distribution parameters:

$$\mu_{k} = \alpha \widetilde{\mu}_{k} + (1 - \alpha) \mu_{k-1}$$

$$\sigma_{k} = \beta \widetilde{\sigma}_{k} + (1 - \beta) \sigma_{k-1}$$
(2.12)

where α and β are weighting parameters introduced to smooth out the convergence.

7. If not converged, set $k \leftarrow k + 1$ and go to step 2.

In this thesis, cross-entropy is used as a search algorithm to set the hyperparameters of the support vector machine models. Its parameters are tuned so as to fit the problem at hand as described in Section 3.3.5.2.

• Covariance marix adaptation - Evolution strategy (CMA-ES)

In a nutshell, CMA-ES is an evolution strategy which relies on multivariate normal distributions to iteratively sample solutions in the descent direction of the objective function (Hansen, 2005). At iteration k, a sample population of size λ is generated as follows:

$$\boldsymbol{d}_{i}^{(k+1)} \sim \boldsymbol{m}^{(k)} + \boldsymbol{\sigma}^{(k)} \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{C}^{(k)}\right), \quad \text{for} \quad i = 1, \dots, \lambda,$$
(2.13)

where $m^{(k)}$ is the mean value of the search distribution, $\sigma^{(k)}$ is the size-step and $C^{(k)}$ is the scaled covariance matrix of the search distribution at iteration *k*.

The parameters for the next generation are generated by simultaneously moving the mean and adapting the covariance matrix. In general, the new mean is obtained as a weighted average of μ selected points among λ generated offsprings:

$$\boldsymbol{m}^{(k+1)} = \sum_{i=1}^{\mu} w_i \boldsymbol{d}_{(i)}^{(k)}$$
(2.14)

where w_i are positive weight coefficients defined such that $\sum_i^{\mu} w_i = 1$ and $w_1 \ge w_2 \ge ... \ge w_{\mu}$. In general one takes $w_i = 1/\mu$ for any $i = \{1, ..., \mu\}$. $d_{(i)}^{(k)}$ is the *i*-th best individual with respect to the objective function *i.e.* $\mathfrak{c}\left(d_{(1)}^{(k)}\right) \le \mathfrak{c}\left(d_{(2)}^{(k)}\right) \le ... \le \mathfrak{c}\left(d_{(\mu)}^{(k)}\right)$.

On the other hand, the covariance matrix can be adapted from scratch as follows:

$$\boldsymbol{C}^{(k+1)} = \sum_{i=1}^{\mu} w_i \left(\boldsymbol{d}_{(i)}^{(k)} - \boldsymbol{m}^{(k)} \right) \left(\boldsymbol{d}_{(i)}^{(k)} - \boldsymbol{m}^{(k)} \right)^T$$
(2.15)

In general, for this covariance matrix to be reliable, the number of samples must be high enough. When the population size λ is small, CMA-ES achieves fast convergence by implementing the so-called *rank-µ-update*. This technique consists in considering also information from previous iterations by weighting the current covariance matrix with the previous ones. Numerous benchmarks have shown efficiency of CMA-ES in solving various optimization problems (Hansen and Kern, 2004; Auger and Hansen, 2005; Arnold and Hansen, 2010). The main assets of CMA-ES over other similar evolution strategies is that it relies on a relatively small population size and number of tunable parameters.

A limiting case, known as (1 + 1)-CMA-ES, has been developed by Igel et al. (2006). In (1 + 1)-CMA-ES, one parent generates one offspring in each iteration. Let us consider that at iteration k, a mutation $v^{(k)}$ has considerably increased the fitness of the objective function. The basic idea behind the (1 + 1)-scheme is to reproduce such a successful step by shifting the mutation distribution toward the direction that produced it. In the family of normal distributions with zero mean, $\mathcal{N}\left(\mathbf{0}, v^{(k)}v^{(k)^{T}}\right)$ is the one with the highest probability of generating $v^{(k)}$ (Hansen and Ostermeier, 2001). This naturally leads to the following adaptation of the covariance matrix:

$$\boldsymbol{C}^{(k+1)} = (1-c)\boldsymbol{C}^{(k)} + c\boldsymbol{\nu}^{(k)}\boldsymbol{\nu}^{(k)^{T}}, \qquad (2.16)$$

where $c \in [0, 1]$ is a weighting coefficient.

To make the search robust, the update is rather based on a so-called *search path* which is an exponentially fading record of previous successful mutations. Besides, from a practical point of view, new candidates are sampled according to:

$$d^{(k+1)} = d_s^{(k)} + \sigma^{(k)} A^{(k)} z^{(k)}, \qquad (2.17)$$

where $z^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\sigma^{(k)}$ is the global step size, $A^{(k)}$ is the Cholesky decomposition of the covariance matrix $C^{(k)}$ and $d_s^{(k)}$ is the current best sample.

Igel et al. (2006) proposed an updating formula directly based on $A^{(k)}$ to avoid the costly successive Cholesky decomposition of $C^{(k)}$. Furthermore, in the case of constrained problems, Arnold and Hansen (012a) proposed an alternative updating scheme which additionally decreases the probability of sampling in the direction of the unfeasible region. This latter algorithm is used throughout this thesis for the application on lightweight design. Appendix B provides further details on its implementation.

2.1.3 Beyond the deterministic framework

In the previous two subsections, we have formulated a constrained optimization problem and briefly introduced some of the means to solve it. Usually, the optimal design is found at the boundary of the feasible space. The slightest change in the initial or operating conditions may turn this optimal design into an unfeasible one. It is therefore crucial to account for any circumstance that might lead to such a change. Uncertainties, on top of modeling error, are acknowledged to be the sources of observed discrepancy between an optimal design and its real manufactured counterpart. The designer should hence pay a special care to them, especially when safety is of concern.

2.1.3.1 Sources of uncertainties

There are many factors that can explain the departure between a system behavior and its real life counterpart. Following the taxonomy proposed by Oberkampf et al. (2002), we consider the model *error* and *uncertainty*.

Model error

Model error refers to the bias due to the modeling of the physical behavior of interest. In fact, and especially in structural design, the natural phenomena are modeled through mathematical equations. As sophisticated as they can get, they remain only abstract representation of reality and therefore cannot always model its complex behavior. Besides, simplifying assumptions and approximations, that further depart the simulation from reality, are often made. Let us consider for instance the frontal impact simulation which is of interest in this thesis. The typical example is the control of the time discretization scheme. Recall that we are running a fast dynamics problem in an explicit formulation. The explicit scheme is conditionally stable *i.e.* for the shock wave to propagate properly, the time step in each cycle should not be higher than a critical value. This threshold is often very low and for scheduling constraints, the engineers cannot afford such a small time step. Instead, its value is artificially increased. This artifact induces an error in the model which is acknowledged and monitored. In general, these modeling errors can be dealt with by considering higher fidelity models. But the expected benefit might not worth the induced cost. In this work, we consider that the finite element simulation is the best model representation of reality we can afford given the available technology and at a reasonable cost. We rather focus on the second source of errors *i.e.* uncertainties.

Uncertainty

The ubiquity of uncertainties in engineering has been largely mentioned in the literature, especially in the risk assessment field (Rowe, 1994; Hora, 1996). Given the diversity of their origins, uncertainties can be classified into two groups: *aleatory* and *epistemic* uncertainty. This distinction provides helpful insight into the nature of uncertainty and allows for an easier treatment.

Aleatory uncertainty represents inherent variation of a process (Oberkampf et al., 2002). It is also known as *irreducible* or *stochastic* uncertainty. It is associated with natural variability either in the initial conditions of a process or in its defining parameters. Let us for instance consider the thickness of a metal sheet in the body-in-white. Due to manufacturing conditions, two parts resulting from the same design specifications may not end up having identical measured thicknesses. The same can be said on material properties. The treatment of aleatory uncertainty resorts to probability theory. Under this framework, any identified source of uncertainty is parametrized and modeled by random variables which are characterized by probability and cumulative density distributions. These distributions can be interpreted as relative frequency of occurrence and their full elicitation requires the availability of data in a relatively large quantity.

Epistemic uncertainty, on the other hand, is rather linked to a lack of knowledge. As a matter of fact, epistemic comes from the Greek word *episteme* which means knowledge. It is also known as *reducible* uncertainty. This is to emphasize that one can reduce or eliminate it by increasing his knowledge of the process under study through *e.g.* the collection of more data. Its treatment mostly resorts to non-probabilistic approaches. Among them, the Dempster-Shafer theory of evidence (Dempster, 1967; Shafer, 1976) which uses *belief* and *plausibility* as measures of uncertainty. The two quantities somehow describe a lower and upper bound of the event without any additional information upon which value is more likely to occur (Argawal, 2004). Likewise, *possibility* (Dubois and Prade, 1988, 2011; Zadeh, 1978) or *fuzzy set* theory (Klir and Folger, 1987; Zimmermann, 1996) can be used. They rely on incomplete or vague information from, say, experimental data or expert judgment.

Globally speaking, a sharp distinction between epistemic and aleatory uncertainty may not be possible. In some cases, the two co-exist. As to which uncertainty representation model should be used mostly resort to the amount of data available. However, probability theory is the prevailing model for uncertainty propagation. This is because it provides a mathematical framework which is convenient for uncertainty propagation and related studies such as robust or reliability-based design optimization. We therefore introduce the associated notations in the next section before moving to optimization under uncertainty.

2.1.3.2 Probability theory

Probability theory finds its origin in the games of chance during the sixteenth century. Prominent thinkers such as Pierre de Fermat and Blaise Pascal were among the first to apply mathematics to gambling issues (Grinstead and Snell, 2006). The modern mathematical formalism was laid by Andrei Kolmogorov's *Grundbegriffe der Wahrscheinlichtkeitsrechnung* in 1933 (Shafer and Vovk, 2006). Today, probability represents an entire branch of mathematics which is concerned with the analysis of random events. In this work, probability is used as a mean to model uncertainty. We therefore succinctly introduce the main concepts and notations useful for the sequel. A comprehensive treatment can be found for instance in Ash (1970) or Durrett (2011), should the reader be interested.

Probability space

Let us consider a random experiment. The set of all possible outcomes is called *sample space* and is denoted by Ω . For instance, we may take $\Omega = \{1, 2, 3, 4, 5, 6\}$ in dice rolling. From this experiment, we may also introduce the concept of *event* as any question that can be answered by a simple "yes" or "no" after the experiment is run. To the question of whether the outcome
of the experiment is an odd number corresponds the event $A = \{1,3,5\}$. From the same experiment, multiple events can be defined. For example, one may also look for outcomes with values lower than 3: $B = \{1,2\}$. The set of all possible events is denoted by \mathcal{F} and is assumed to be a σ -algebra *i.e.* a collection of non-empty subsets of Ω which is closed under countable operators (finite union and complement). In other words, \mathcal{F} is a σ -algebra if the following two statements are true:

- (*i*) $A \in \mathcal{F}$ implies that its complement $\overline{A} \in \mathcal{F}$ and;
- (*ii*) For a countable sequence of sets, $A_i \in \mathcal{F}$ implies that their union $\cup_i A_i \in \mathcal{F}$

These two elements form a *measurable space* (Ω, \mathcal{F}) . It is therefore possible to put a measure on the events of this space. This is achieved through the *probability measure*, defined as the application $\mathcal{P} \rightarrow [0, 1]$ which follows the Kolmogorov axioms:

$$\mathcal{P}(A) \ge 0 \qquad \forall A \in \mathcal{F},$$

$$\mathcal{P}(\Omega) = 1,$$

$$\mathcal{P}(A \cup B) = \mathcal{P}(A) + \mathcal{P}(B) \quad A, B \in \mathcal{F}; \quad A \cap B = \emptyset,$$
(2.18)

where \emptyset denotes the empty set. The triplet $(\Omega, \mathcal{F}, \mathcal{P})$ defines the *probability space*.

The probability measure assigns numbers to events which somehow reflect how likely they are to occur. From general agreement, the interpretation of probability endows a frequentist connotation. That is, if we consider an event $A \in \mathcal{F}$, then $\mathcal{P}(A)$ should converge towards the ratio between the number of outcomes favorable to A and the total number of outcomes as the experiment is repeated over and over again. This classical view of probability is quite convenient but is not the only one (O'Hagan and Oakley, 2004; Cooke, 2002; Ramsey, 1926). Subjective interpretation may also be regarded. In this sense, the probability reflects one's degree of belief (or knowledge for epistemic uncertainty). The more plausible the event, the higher the assigned probability. This is also known as *Bayesian* interpretation and proves convenient when it comes to *Bayesian inference* where the probability can be sequentially updated as evidence (data) is acquired. This is achieved through the well-known Bayes formula which reads:

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(B|A)\mathcal{P}(A)}{\mathcal{P}(B)},$$
(2.19)

where $\mathcal{P}(A|B)$ is the conditional probability of *A* given *B*, also defined by:

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(A \cap B)}{\mathcal{P}(B)}.$$
(2.20)

Random variables and probability distribution

A random variable X is a real-valued function on the sample space:

$$\begin{array}{ll} X: & \Omega \to \mathcal{D}_X \subset \mathbb{R} \\ & \omega \mapsto X\left(\omega\right) = x, \end{array} \tag{2.21}$$

where D_X is the support of *X*. It may be discrete or continuous in which case the random variable is said to be respectively *discrete* or *continuous*. Usually the capital letter denotes the random variable and its *realization* is written with the corresponding lower case letter.

With this definition, it is possible to distribute probability mass to events according to their relative likelihood of occurrence. We may for instance look for the probability that the realization of a random variable is within a given interval. The *cumulative distribution function* gives such an information and reads:

$$F_{X}(x) = \mathcal{P}(X \le x), \qquad (2.22)$$

with $F_X(x) \in \mathbb{R}$.

Similarly, the probability density function (PDF) indicates how distributed is the weight assigned to any point of the support D_X :

$$f_X(x) = \lim_{h \to 0, h > 0} \mathcal{P}\left(x \le X \le x + h\right) / h,$$
(2.23)

otherwise put:

$$f_X(x) = \frac{dF_X(x)}{dx}.$$
(2.24)

Eventually, the probability that *X* belongs to an interval, say]a, b], may therefore be computed by:

$$\mathcal{P}(a < X \le b) = F_X(b) - F_X(a) = \int_a^b f_X(x) \, dx.$$
 (2.25)

Statistical moments of random variables

Beside the CDF which completely defines a random variable, other quantities can be derived to give an idea of how the random variable is distributed over its support. The first one is the mathematical expectation, which gives a probability-weighted average of the random variable over its support:

$$\mathbb{E}\left[X\right] = \mu_X = \int_{\mathcal{D}_X} x f_X\left(x\right) dx.$$
(2.26)

The expectation corresponds to the first-order statistical moment of the random variable. Its second-order moment is known as the variance and informs on how the random variable is spread out around its expected value:

Var
$$[X] = \int_{\mathcal{D}_X} (x - \mu_X)^2 f_X(x) \, dx = \mathbb{E}\left[(X - \mu_X)^2 \right],$$
 (2.27)

which is a non-negative number. A small value means that most of the samples of *X* with respect to f_X are close to the expectation μ_X and conversely a high value means that realizations of *X* are scattered. The variance is commonly associated to the *standard deviation* σ , which is simply its square-root:

$$\sigma_X = \sqrt{\operatorname{Var}\left[X\right]}.\tag{2.28}$$

Another quantity that gives a good account of the dispersion of a random variable with respect to its expected value is the *coefficient of variation*. It is expressed as a ratio between the standard deviation and the mean:

$$\delta_X = \frac{\sigma_X}{|\mu_X|}.\tag{2.29}$$

Eventually, we should point out that additional information can be given on the distribution of the random variable by higher order moments. The most common being *skewness* (third-order moment) which indicates how asymmetric is the distribution and *kurtosis* which indicates how

flat or peaked is the distribution (both with respect to normal distribution). They respectively read:

$$\delta_{3_{X}} = \frac{1}{\sigma_{X}^{3}} \int_{\mathcal{D}_{X}} (x - \mu_{X})^{3} f_{X}(x) dx,$$

$$\delta_{4_{X}} = \frac{1}{\sigma_{X}^{4}} \int_{\mathcal{D}_{X}} (x - \mu_{X})^{4} f_{X}(x) dx.$$
(2.30)

Example of well-known distributions

Normal or *Gaussian* distribution is probably the most used distribution in statistics. It has been widely used to model the distribution of physical quantities when no further information is available. A random variable X with mean μ_X and standard deviation σ_X is said to follow a normal distribution $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$ if its PDF writes as follows:

$$f_X(\mathbf{x}) = \frac{1}{\sigma_X \sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\frac{x - \mu_X}{\sigma_X}\right)^2\right].$$
 (2.31)

The associated CDF is the integral of φ over the support of *X*. It cannot be computed analytically. One rather resorts to the so-called *error function* (erf) thus leading to:

$$F_{X}(x) = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x - \mu_{X}}{\sigma_{x}\sqrt{2}}\right) \right].$$
(2.32)

The special case where $\mu_X = 0$ and $\sigma_X = 1$ corresponds to the *standard* Gaussian distribution. Illustrations are given in Figure 2.1.



Figure 2.1: Standard Gaussian distribution.

The wide use of Gaussian distributions can be attributed to the *central limit theorem* (CLT). Indeed, the CLT states that, under mild conditions, the mean of randomly sampled quantities is approximately normally distributed. In the case where the variables are known to be positive, the CLT theorem may be considered in the log domain. This has given rise to *log-normal* distribution. A random variable is said to be log-normally distributed if its logarithm is normally distributed. It is characterized by the scale and location parameters which respectively read:

$$\zeta_X = \sqrt{\operatorname{Var}\left[\ln X\right]} = \sqrt{\ln\left(1 + \delta_X^2\right)}$$

$$\lambda_X = \mathbb{E}\left[\ln X\right] = \ln\left(\mu_X\right) - \frac{1}{2}\zeta_X^2.$$
(2.33)

The random variable *X* is therefore said to follow a lognormal distribution: $X \sim \mathcal{LN}(\lambda_X, \zeta_X)$. The associated probability density function then reads:

$$f_X(x) = \frac{1}{x\zeta_X\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{-\ln x - \lambda_X}{\zeta_X}\right)^2\right].$$
 (2.34)

The PDF and CDF of a log-normal distribution are given in Figure 2.2. The PDF is asymmetric with a higher weight at the left of the mean.



Figure 2.2: Example of a lognormal distribution.

Another widely-used family of distribution is the *uniform* one under which the random variable is equiprobably distributed within an interval. It is completely defined by the parameters *a* and *b* which are respectively the minimum and maximum values and is denoted by $X \sim U(a, b)$.

2.1.3.3 Summary

In this section, we introduced structural optimization in a general context. A deterministic formulation was presented followed by some algorithms to solve it. The limitations of a deterministic view of optimization was highlighted. Model error and the presence of uncertainties make it difficult to rely on a deterministic approach. Probabilistic approaches provide an appealing alternative. For this purpose, the basic concepts of probability theory were introduced. In the next section, we will see how this knowledge can be used for an effective optimization under uncertainties.

2.2 Optimization under uncertainties

Because of model inadequacy and uncertainty, one can never be certain that a structure will perform exactly as planned by the designer. The performance might indeed decline because of variations of the design due to the manufacturing process or environmental conditions. A better approach would be to eliminate or reduce the causes of the variability in the outputs. This implies additional cost. As a matter of fact, a higher fidelity model will potentially delay the timeto-market and tighter design tolerances would require to invest in expensive high-precision tooling. Alternatively, one may simply account for these uncertainties in the model, thus producing a design immune to small variations of its inputs. This may be achieved through robust or reliability-based design optimization according to how the task is addressed. The two approaches are briefly reviewed in this section.

2.2.1 Robust design optimization

2.2.1.1 The concept of robust design

A robust design is one for which the sensitivity with respect to system variations at the optimal point is kept as small as possible. The concept of robustness is inextricably rooted in operational research and design engineering (Beyer and Sendhoff, 2007; Mulvey et al., 1995). Beside the effect of uncertainty mentioned earlier, a robust concept is also meant to integrate the life-cycle cost of the design, by anticipating maintenance and recycling capability. To this end, a robust design should remain close to optimal and feasible as its environmental conditions change during its entire life-cycle span.

An illustration of the concept of robustness is given in Figure 2.3 (Kang, 2005). Two feasible solutions of a problem are shown. The horizontal axis represents the cost function and the vertical one its distribution under random variation of the inputs in the neighborhood of the given solution. μ_c^* is the optimal solution found with respect to a deterministic approach. A nearly optimal solution $\mu_c^{(*)}$ is also provided. The two are compared according to their distributions. It turns out that the optimal design is highly sensitive to random perturbation of its nominal input. In contrast, the nearly optimal solution is associated with much less variability. From the point of view of robustness, $\mu_c^{(*)}$ is preferable to μ_c^* despite the latter bears a smaller cost.

In principle, robust design implements methodologies that are more likely to yield solutions like $\mu_c^{(*)}$ rather than μ_c^* . The first step consists in identifying the sources of uncertainty that may cause the variability in the output. For an appropriate treatment, they are classified into different categories. An alternative classification to the epistemological one presented above is usually considered in the related literature (Beyer and Sendhoff, 2007). This classification offers a more convenient way to deal with uncertainty in robust design optimization. They are the following:

- Type A which refers to changing environmental conditions. They are not under the control of the designer. In this thesis, they correspond to the environmental variables *z* (Eq. (2.1)). When a probabilistic approach is considered to account for the related uncertainties, the random variable *Z* ~ *f*_Z is introduced.
- Type B which corresponds to production tolerances on the design due to the manufacturing process. They are modeled here by the random variable $X \sim f_{X|d}$. The probability distribution of X is indexed by the nominal value of the design variable d.



Figure 2.3: Concept of robustness: Two solutions are shown with their associated variability. The scattered solution is favored for robust design.

• Type C which is associated to model inadequacy, also coined model error. It includes measuring errors as well.

To these three mentioned, we may also add the type D also known as feasibility uncertainty, though it is not an additional source of uncertainty. In fact, it is the same as types A, B and C, except that it is considered to act on the constraints rather than on the cost function.

According to the type that is of concern, different concepts of robustness can be found in the literature. In the sequel, we briefly review the most important ones.

2.2.1.2 A glimpse on Taguchi's pioneering work

The name of the Japanese quality expert Dr. Genichi Taguchi is profoundly linked to robust design as he is the one who instilled the design philosophy underlying the concept of robustness (Taguchi and Phadke, 1989). He contributed to generalizing the robust optimization idea as a mean to control the quality of manufactured goods. The Taguchi's methods involve some concepts from statistics.

First is the introduction of a *loss function* which measures the discrepancy between an actual outcome and the desired one. It is expressed in terms of a *mean-square deviation* which can take three different forms according to the designer's aim:

• The-smaller-the-better: the design goal is to have the cost as close as possible to zero. The associated mean-square deviation therefore reads:

$$MSD_{S}(d) = \frac{1}{N} \sum_{i=1}^{N} c(x_{i}(d), z_{i})^{2}, \qquad (2.35)$$

where *N* is the size of the population required to estimate the mean-square deviation.

• The-larger-the-better: here the aim is to maximize a cost.

$$MSD_{L}(d) = \frac{1}{N} \sum_{i=1}^{N} \mathfrak{c}(\mathbf{x}_{i}(d), \mathbf{z}_{i})^{-2}.$$
 (2.36)

• The-nominal-the-better: the design goal here is to be as close as possible to a nominal value

$$MSD_{N}(d) = \frac{1}{N} \sum_{i=1}^{N} (\mathfrak{c}(x_{i}(d), z_{i}) - \bar{c})^{2}, \qquad (2.37)$$

where \bar{c} is the target cost. This situation may occur for instance when one seeks to machine a part at a given thickness.

These loss functions are computed for a fixed value of the design d and are thereafter used to compute so-called *signal to noise ratios* (S/N) defined as follows:

$$S/N(d) = -10\log_{10}(MSD_{\bullet}(d))$$
 (2.38)

where • stands for the subscripts S, L or N. Eq. (2.38) is to be maximized with respect to d in order to find a robust solution.

The question that arises now is how to select the set of points $\{(x_i (d), z_i), i = 1, ..., N\}$ that is to be used to compute the mean-square deviation in Eqs.(2.35), (2.36) and (2.37). Taguchi answers this question through the use of *design of experiments* with two layers. More specifically, an orthogonal array is used. The *inner array* is constructed on the control parameters (design variables *d*). For each point of this lattice, a sequence of noise parameters is considered resulting in the set $\{(x_i (d), z_i), i = 1, ..., N\}$. This sequence is referred to as the *outer array*.

With this approach, Taguchi translates the optimization of Eq. (2.38) into a discrete one by considering only the points sampled in the orthogonal array. Despite it is a pragmatic approach, it has proven to be inefficient when the cost has highly non-linear effects. Moreover, to apply this, one has to carefully set the design of experiments. The number of selected control and noise parameters should be kept as small as possible in order to avoid conducting too many experiments. Other limitations were pointed out by Box et al. (1988) or Nair et al. (1992), specifically regarding the use of the S/N ratio. Globally speaking, Taguchi's method has been largely praised for its philosophy of robustness, making quality search mainstream, but its practical implementation through statistical approaches was proved to be either inefficient or unnecessarily complicated (Trosset, 1997).

2.2.1.3 Broader formulations for robust design optimization

Let us now review the main formulations of robust design from a broader perspective. Beyer and Sendhoff (2007) give an overview over which this section is based.

Worst-case scenario

A usual approach for robust design consists in implementing worst-case scenarios. Let us consider the objective function c(d). A robust counterpart of the optimization problem considering the local variability in the design and environmental parameters x(d) and z may write as follows:

$$d^{*} = \arg\min_{d \in \mathbb{D}} \sup_{(x,z) \in \mathbb{X} \times \mathbb{Z}} \mathfrak{c}(x(d), z), \qquad (2.39)$$

where X and Z respectively denote the finite supports of *x* and *z*.

In this formulation, the cost is replaced by a robustness measure which ensures that for each design (or control) parameter, the worst possible case is accounted for. To define such a case, the probability distribution of random variables modeling the uncertainty is not necessary. Only their bounded support is needed. The worst-case scenario is actually considered when one does not have enough information on the uncertain parameters except for a vague knowledge about their range of variations. In general this is an extremely conservative measure, which is hardly applicable in industrial problems. Beside, this robustness measure might turn out to be discontinuous because of the sup operator.

Moments-based measures

Beyer and Sendhoff (2007) presented another robustness measure based on a utility function which is defined by:

$$\mathcal{U}(\mathfrak{c}) = \operatorname{sign}(\mathfrak{c}) |\mathfrak{c}|^k, \tag{2.40}$$

where the positive power coefficient k can be tuned so as to put more or less emphasis on extreme values of the cost function. The smaller the value of k, the less emphasis is put on extreme values of c.

The associated robustness measure is constructed by taking the expectation of the utility function. This gives the following robust counterpart of c:

$$\boldsymbol{d}^{*} = \arg\min_{\boldsymbol{d}\in\mathbb{D}}\mathbb{E}\left[\boldsymbol{U}\left(\boldsymbol{\mathfrak{c}}\right)|\boldsymbol{d}\right],\tag{2.41}$$

where the expectation is computed with respect to $X \sim f_{X|d}$ and $Z \sim f_Z$.

The special case k = 1 corresponds to the expectation of c as described in Trosset (1997):

$$d^{*} = \arg\min_{d\in\mathbb{D}}\mathbb{E}\left[\mathfrak{c}|d\right] = \arg\min_{d\in\mathbb{D}}\int_{\mathbb{R}^{s}}\mathfrak{c}\left(x,z\right)f_{X|d}\left(x\right)f_{Z}\left(z\right)dxdz,$$
(2.42)

where *s* is the number of design and environmental variables.

By Eq. (2.42), the cost function is averaged around the nominal value of the control parameter. As a consequence, a smoother version is actually minimized thus reducing the likelihood of finding an optimum located in regions where the gradient of the cost function is considerably high. A more straightforward approach to get the same result is to consider directly a *dispersion measure* of the cost function. One such measure is given by the variance, yielding to the following optimization problem:

$$d^{*} = \arg\min_{d\in\mathbb{D}} \operatorname{Var}\left[\mathfrak{c}|d\right] = \arg\min_{d\in\mathbb{D}} \int_{\mathbb{R}^{s}} \left(\mathfrak{c}\left(x,z\right) - \mathbb{E}\left[\mathfrak{c}|d\right]\right)^{2} f_{X|d}\left(x\right) f_{Z}\left(z\right) dxdz.$$
(2.43)

This measure focuses too much on the dispersion of the objective function around a nominal value and does not take into account any information on its merit (*i.e.* whether the objective function is actually minimized or not). In fact, plateau-like areas are somehow favored.

To illustrate this, let us consider the one-dimensional cost function presented in Lee et al. (2005) which writes:

$$\mathfrak{c}(d) = \sum_{i=1}^{10} a_i d^i, \qquad (2.44)$$

where $\{a_i, i = 1, ..., 10\}$ are weighting coefficients. Their exact values are not necessary here so we omit them for the sake of clarity (if needed, see Lee et al. (2005), p. 783).

We then consider a setting with no environmental variable but with randomness in the design parameters, modeled by $X \sim \mathcal{N}(d, 0.2^2)$ for any $d \in [0, 10]$. The blue line in Figure 2.4 illustrates the deterministic cost and its minimum value which corresponds to $d^* = 9.70$. For its robust counterpart, we consider the expectation and variance measures which are respectively represented by the green and red lines. The integrals in Eq. (2.42) and Eq. (2.43) are computed through Monte Carlo sampling. Their optimizations respectively yield $d_E = 0.74$ and $d_V = 2.84$.



Figure 2.4: Robust design optimization: Comparison of different robustness measures.

This simple example shows how different the optimal solution of the deterministic problem can be from that of its robust counterpart. In fact, the expectation measure has smoothed the function thus avoiding the region of large gradient where the deterministic design lies. As for the variance measure, the optimum lies in a plateau-like area. Here it happens that it corresponds to a local minimum of the function, but it need not always be the case. It might have as well been on a local maximum of the function. To avoid this latter situation, a combination of the two robustness measures is often used (Doltsinis and Kang, 2004; Medina and Taflanidis, 2014). This results in a multi-objective optimization problem. One way to solve it is to aggregate the two measures as follows:

$$d^* = \arg\min_{d\in\mathbb{D}} (1-\alpha) \frac{\mu_{\mathfrak{c}}(d)}{\mu_u} + \alpha \frac{\sigma_{\mathfrak{c}}(d)}{\sigma_u}$$
(2.45)

where $\mu_{\mathfrak{c}}(d) = \mathbb{E}[\mathfrak{c}|d]$, $\sigma_{\mathfrak{c}}(d) = \sqrt{\operatorname{Var}[\mathfrak{c}|d]}$, and $\alpha \in [0, 1]$ is a weighting coefficient. μ_u and σ_u are normalizing coefficients. They are actually coordinates of the so-called *utopia point* corresponding to the optimal solutions of each optimization problem taken separately.

With the formulation above, the remaining task is that of finding an appropriate weight parameter. Alternatively, one may directly search for a Pareto set. Then the designer has to decide which solution is the most suitable. This decision can be made by considering some preferences or objectives that were not stated in the initial problem. In the probabilistic framework, Medina and Taflanidis (2014) introduced another robustness measure called *probability of dominance* to help choosing one design from the Pareto-optimal solutions.

Probabilistic threshold measure

The third family of robustness measures can be derived by defining a probabilistic threshold (Beyer and Sendhoff, 2007). This may be formulated through quantiles defined on the distribution of the cost function. The idea is to be conservative with respect to small scatter of the cost that may result from the inputs variability. Following Baudoui (2012), the robust optimization problem may read:

$$d^* = \arg\min_{d\in\mathbb{D}} Q_{\alpha}\left(\mathfrak{c}|d\right) = \arg\min_{d\in\mathbb{D}} \inf\left\{q\in\mathbb{R}: \mathcal{P}\left(\mathfrak{c}\left(d\right)\leq q\right)\geq\alpha\right\}$$
(2.46)

The degree of conservatism can be controlled by the parameter α . The higher it is, the more conservative is the solution. The case $\alpha = 1$ corresponds to the worst-case approach introduced above, provided that the supports of the random variables are finite.

2.2.2 Reliability-based design optimization

2.2.2.1 From deterministic to probabilistic approach of safety

The previous section on robustness was concerned with the effect of randomness in the cost function. The present one rather focuses on the safe performance of a structure during its life time. The assessment of safety is usually carried out in the space of input parameters of the mathematical model that governs the behavior of the structure. Let $w = \{x, z\}^T \in \mathbb{R}^s$ be the vector gathering the input parameters of the model. The input space is conventionally divided into two sets, namely:

- *Failure* domain: $\Omega_f = \{ \boldsymbol{w} \in \mathbb{R}^s | \mathfrak{g}(\boldsymbol{w}) \leq 0 \};$
- Safety domain: $\Omega_s = \{ w \in \mathbb{R}^s | \mathfrak{g}(w) > 0 \}.$

This partition is made by the so-called *limit state function*: $w \mapsto g(w)$ and the set defined by $\{w \in \mathbb{R}^s | g(w) = 0\}$ is known as the *limit state surface*.

In deterministic design optimization, the solution most often lies on this boundary. As argued earlier, this might not be truly safe because of uncertainties that can invalidate the solution. The usual approach to cope with this situation within the deterministic philosophy is to introduce so-called *safety factors*. To illustrate this, let us consider the widely-used capacity-and-demand example where the limit state function reads:

$$\mathfrak{g}\left(d\right)=r-s,\tag{2.47}$$

where $d = \{r, s\}^T$ with *r* being the capacity or resistance and *s* the demand or source. In a deterministic context, the input parameters *w* reduce to the design variables *d*.

For safety factors, the limit state function is modified and recast as follows:

$$\mathfrak{g}\left(d\right) = \frac{r}{\gamma_r} - \gamma_s s, \qquad (2.48)$$

where the coefficients $\gamma_r \ge 1$ and $\gamma_s \ge 1$ are the specified safety factors.

This formulation leads to moving the solution away from the boundary of the failure domain. The coefficients have to be set once and for all according to the degree of confidence the designer has in his model. However, as pointed out in Ditlevsen and Madsen (1996), the expression of the limit state function is not unique. The limit state in Eq. (2.47) could have been equally expressed as g(r,s) = r/s - 1. In this case, the value specified in Eq. (2.48) would not bear the same results. A model invariant setting of the safety factors is therefore necessary. A mean to achieve this is to directly set the safety factors on the input parameters of the model. This approach is known as the *partial safety factors* method. The procedure is the following: Let $d = \{d_1, d_2, \dots, d_s\}$ be a point in the safe domain. If the hyperrectangular volume $\prod_{i=1}^{s} \left[d_i / \gamma_i^l , \gamma_i^u d_i \right]$ is a subset of the safe domain, only then the solution is considered to be acceptable. In practice, the verification is made only on the vertices of the hyperrectangle. This involves 2^s additional evaluations of the model, which might be problematic in high dimensional problems. This number can be reduced by considering simplifying assumptions on the limit state surface such as monotonicity. That is, for parameters of resistance type, the limit state function increases as the parameter becomes large. In contrast for source or demand types, the limit state function decreases as the parameter grows. There is however some other issues that are raised by the approach. In fact, for all the hyperrectangular volume delimited by the vertices to be feasible, the limit state surface has to be convex. This assumption may not always hold. Eventually, the value given to the partial safety factors should reflect the uncertainty in each input parameter. However, their calibration does not fully acknowledge the random distribution of the parameters when such information is available. This often results in over-designed structures.



Figure 2.5: Illustration of the partial safety factors concept. The vertices of the rectangular neighborhood domain of the optimal solution are plotted as black bullets. Two cases are considered: convex (g_1) and non convex (g_2) limit state surfaces.

To overcome the shortcomings discussed above, a probabilistic approach has been extensively used. It consists in modeling the variability of the input parameters by random variables whose PDFs are well defined. The associated uncertainty is therefore propagated to the model outputs through a so-called *reliability analysis*. The associated optimization problem, referred to as *reliability-based design optimization* (RBDO) is introduced below.

2.2.2.2 Problem formulation

In reliability-based design optimization, the uncertainty in the inputs is modeled by random variables and the hard constraints are assessed from a probabilistic viewpoint. This can be formulated as follows (Dubourg, 2011):

$$d^* = \arg\min_{d\in\mathbb{D}} \mathfrak{c}(d) \quad \text{subject to:} \begin{cases} \mathfrak{f}_j(d) \leq 0, & \{j=1,\ldots,n_s\}\\ \mathcal{P}\left(\mathfrak{g}_k\left(\boldsymbol{X}\left(d\right),\boldsymbol{Z}\right) \leq 0\right) \leq \bar{P}_{f_k}, & \{k=1,\ldots,n_h\} \end{cases}.$$
(2.49)

The notations are the same as those introduced in Eq. (2.1). As a reminder c is the cost function that is to be minimized with respect to design variables $d \in \mathbb{D} \subset \mathbb{R}^{s_d}$. This optimization is to be carried out under some constraints. The soft constraints $\mathfrak{f} = \{\mathfrak{f}_j, j = 1, \ldots, n_s\}$ are simple analytical functions that may for instance bound the design space. In contrast, the hard constraints $\mathfrak{g} = \{\mathfrak{g}_k, k = 1, \ldots, n_h\}$ are the performance functions underlying the behavior of the structure. As already pointed out, they most often result from a finite element model and may be expressed as $\mathfrak{g}_k = \overline{\mathfrak{g}}_k - \mathcal{M}_k$, where \mathcal{M}_k is an output from the FE model and $\overline{\mathfrak{g}}_k$ is a threshold which is not to be exceeded. The difference with the DDO formulation lies in the introduction of the random variables $X \sim f_{X|d}$ and $Z \sim f_Z$ for a given design d. $X \in \mathbb{R}^{s_d}$ is indexed on the design variables and $Z \in \mathbb{R}^{s_2}$ stands for the environmental variables. With this probabilistic model, a probability of failure is computed for each constraint and is required to be lower than a given threshold denoted by \overline{P}_{f_k} . Let us gather the random parameters into a single vector: $W = \{X, Z\}$ which follows the joint multivariate distribution f_W . The probability of failure is therefore defined by:

$$P_{f_{k}}(\boldsymbol{d}) = \mathcal{P}\left(\mathfrak{g}_{k}\left(\boldsymbol{W}\right) \leq 0\right) = \int_{\mathfrak{g}_{k}(\boldsymbol{W}) \leq 0} f_{\boldsymbol{W}}\left(\boldsymbol{w}\right) d\boldsymbol{w}.$$
(2.50)

The computation of this integral is not analytically tractable. One rather resorts to approximate methods as shown in the next section. In the following, we will drop the subscript k for the sake of clarity, thus considering only one constraint. The multiple constraints case can be handled by specific reliability techniques that we will enumerate later on.

In this formulation, the cost function is supposed to be deterministic. This setting corresponds to the application in this thesis. Many formulations in the literature of RBDO also assume such deterministic cost functions. As argued in Dubourg (2011), a fully probabilistic formulation is possible. The cost function would therefore be cast as an expectation: $c(d) \equiv \mathbb{E} [c(X(d), Z)]$. In our case, the computational cost required to evaluate this expectation is not high as the cost function is usually a simple, often analytical, model. Accounting for uncertainties in the cost function, it may also be possible to formulate a problem which seeks for a design that is both robust and reliable. For robustness the cost function is simply replaced by one of the robustness measures introduced in the previous section. Mourelatos and Liang (2006) formulate such a problem where the variance of the cost function is considered as merit.

We illustrate the difference between a DDO and an RBDO solution in Figure 2.6. The deterministic example is formulated as follows:

$$d^* = \arg\min_{d \in [0,10]^2} 10 - d_1 - d_2 \quad \text{subject to: } \mathfrak{g}(d) \le 80 / (x_1^2 + 8x_2 + 5) + 1.$$
(2.51)

For RBDO, the probabilistic model simply consists of random design variables $X_i \sim \mathcal{N}(d_i, 0.5^2)$, $i = \{1, 2\}$. The probability of failure should be lower than $\bar{P}_f = 0.05$.

The deterministic optimal solution is found at $d_{\text{DDO}}^* = \{4.00, 7.37\}^T$ and lies on the limit state surface. For RBDO, the solution is found at $d_{\text{RBDO}}^* = \{3.45, 6.74\}^T$. Hence, the final cost for the optimal solution is higher in RBDO (-0.19 against -1.38). This is to be expected with RBDO as the reliability of the solution is traded against the cost. When the problem is well expressed, this results in moving away the solution from the boundary of the failure domain thus resulting in an increased cost. In fact, in RBDO one not only considers the nominal design but possible realizations in its neighborhood. Figure 2.6 illustrates the point. A Monte Carlo simulation using 1,000 points is drawn according to the distribution of the design variables at the optimal solution. A small portion, approximately 5%, of these points is not in the safe domain. They are represented in red while those in the safe domain are green. This proportion actually corresponds to the target probability of failure \overline{P}_f set to 0.05. This explains why the optimal solution does not lie on the limit state surface. In general, the smaller the probability of failure, the further the solution lies from the limit state surface.



Figure 2.6: Illustration of deterministic vs. reliability-based design optimization on a twodimensional problem.

In this example, the probability of failure was computed by a simple Monte Carlo simulation. There exists many other approaches to estimate failure probabilities as will be explained in the next subsection.

2.2.2.3 Structural reliability analysis

The methods used to estimate probabilities of failure in structural reliability analyses can be classified into *approximation* and *simulation methods* (Madsen et al., 1986). We very briefly introduce some of the most widely-used in this section.

Approximation methods

The first-order reliability method (FORM) is known as one of the most practical approaches to approximate a probability of failure. The integration in Eq. (2.50) is computed in the standard space. To this end, a mapping from the space of original random variables W to the standard Gaussian space is performed:

$$u = \mathbb{T}(w),$$

$$\mathfrak{g}_{u}(u) = \mathfrak{g}\left(\mathbb{T}^{-1}(u)\right),$$
(2.52)

where \mathbb{T} is the mapping application which varies according to the method used. We will come back to the mapping later. For now, let us assume that we have independent normally distributed variables U.

FORM approximation proceeds by first identifying the point on the limit state-surface which is closest to the origin. Since the standard Gaussian joint probability distribution $\varphi(u)$ decreases exponentially with the radial distance to the origin and is rotationally symmetric, this point is the one with maximum likelihood over the failure domain. It is actually known as the *design point* or *most probable failure point* (MPFP). Finding the MPFP resorts to a constrained optimization problem:

$$u^* = \arg\min_{u} ||u||$$
 subject to: $\mathfrak{g}_u(u) = 0.$ (2.53)

Many algorithms were developed to specifically solve this equation. One reference algorithm is the Hasofer-Lind-Rackwitz-Fiessler (HLRF) (Hasofer and Lind, 1974; Rackwitz and Fiessler, 1978). Since then, improved approaches have been developed (Liu and Der Kiureghian, 1992; Santos et al., 2012) based on the following recursive formula:

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \lambda_k \boldsymbol{\delta}_k \tag{2.54}$$

where the descent direction is given by:

$$\delta_{k} = \frac{1}{\|\nabla \mathfrak{g}_{u}\left(\boldsymbol{u}_{k}\right)\|^{2}} \left(\nabla \mathfrak{g}_{u}\left(\boldsymbol{u}_{k}\right)\boldsymbol{u}_{k} - \mathfrak{g}_{u}\left(\boldsymbol{u}_{k}\right)\right) \nabla \mathfrak{g}_{u}^{T}\left(\boldsymbol{u}_{k}\right) - \boldsymbol{u}_{k}.$$
(2.55)

 λ_k is the step size. Many of the available algorithms only differ in how this size step is calibrated. Eventually, one may also consider general-purpose algorithms as those presented in section 2.1.2 to solve the problem.

Once the problem in Eq. (2.53) is solved, the limit state function is linearized around the MPFP following its first-order Taylor expansion:

$$\mathfrak{g}_{u}\left(\boldsymbol{u}\right) \approx \mathfrak{g}_{u}\left(\boldsymbol{u}^{*}\right) + \nabla^{T}\mathfrak{g}_{u}\left(\boldsymbol{u}^{*}\right)\left(\boldsymbol{u}-\boldsymbol{u}^{*}\right). \tag{2.56}$$

Note that the first summand of the right-hand side of this equation is zero since u^* belongs to the limit state surface. Upon normalizing the remaining part of the equation, the following normalized approximation of the limit state function can be derived:

$$\widetilde{\mathfrak{g}}_{u}\left(\boldsymbol{u}\right) = \frac{\nabla^{T}\mathfrak{g}_{u}\left(\boldsymbol{u}^{*}\right)}{\left\|\nabla^{T}\mathfrak{g}_{u}\left(\boldsymbol{u}^{*}\right)\right\|}\left(\boldsymbol{u}-\boldsymbol{u}^{*}\right) = \beta - \boldsymbol{\alpha}^{T}\boldsymbol{u},$$
(2.57)

33

where $\alpha = -\nabla^T \mathfrak{g}_u(u^*) / \|\nabla^T \mathfrak{g}_u(u^*)\|$ and $\beta = \alpha^T u^*$ is the so-called *reliability index*. This latter quantity represents the shortest distance between the origin of the standard space and the hyperplane that approximates the limit state surface. This geometrical interpretation is due to Hasofer and Lind (1974) and it is therefore referred to as *Hasofer-Lind reliability index*.

The final step in FORM is to replace the probability of failure by the one computed with respect to its above-approximation:

$$\mathcal{P}\left(\mathfrak{g}_{u}\left(\boldsymbol{u}\right)\leq0\right)\approx\mathcal{P}\left(\widetilde{\mathfrak{g}}_{u}\left(\boldsymbol{u}\right)\leq0\right)=\mathcal{P}\left(-\boldsymbol{\alpha}^{T}\boldsymbol{u}\leq-\beta\right)$$
(2.58)

Since the quantity $-\alpha^T u$ turns out to be a standard Gaussian random vector, the probability of failure can be analytically evaluated and reads:

$$P_{f_{FORM}} = \mathcal{P}\left(-\boldsymbol{\alpha}^{T}\boldsymbol{u} \leq -\boldsymbol{\beta}\right) = \Phi\left(-\boldsymbol{\beta}\right), \qquad (2.59)$$

where Φ is the CDF of the standard Gaussian distribution.

This estimation is actually exact when the limit state surface is linear. An extension of FORM which uses second-order polynomial approximation has been proposed by Breitung (1989). *Second-order reliability method* (SORM) requires to compute the Hessian of the limit state function to construct a quadratic approximating surface at the design point. The probability of failure obtained by FORM is hence corrected by including curvature terms derived from this surface. This results in an asymptotic approximation which becomes exact as $\beta \rightarrow \infty$:

$$P_{f_{SORM}} = \Phi(-\beta) \prod_{i=1}^{s} (1 - \beta \kappa_i)^{-1/2}, \qquad (2.60)$$

provided that $1 - \beta \kappa_i > 0$ with κ_i being the main curvatures of the limit state surface.

These two methods have been extensively used in the literature. They generally provide fairly good results at minimum cost for engineering problems. However, they assume the existence of only one most probable failure point. Otherwise, it has been shown that they provide inaccurate results (Der Kiureghian and Dakessian, 1998).

An illustration of the first- and second-order reliability methods is given in Figure 2.7. We consider a two-dimensional problem. The left panel shows the limit state surface in the physical space as well as the PDF of the joint distribution of the random variables $W = \{W_1, W_2\}$. The mapping to the standard Gaussian space is illustrated in Figure 2.7b. The three limit state surfaces are shown: the original, the FORM and the SORM approximations.

The developments above assumed that the random variables were independent and normally distributed. However, this is often not the case and a mapping is then necessary. According to the nature of the original random variables, one of the following mapping may be used.

• Normally distributed but correlated random variables. It is assumed in this case that $W \sim \mathcal{N}(M, C)$ where $M = (\mu_1, \dots, \mu_s)^T$. The covariance matrix reads C = DRD where $D = [r_{ij}]_{s \times s}$ is the linear correlation matrix and $D = \text{diag}(\sigma_1, \dots, \sigma_s)$ is the diagonal matrix of standard deviations. The mapping therefore reads:

$$u = L^{-1}D^{-1}(w - M),$$

$$w = DL(u - M),$$
(2.61)

where *L* is the lower triangular matrix of the Cholesky decomposition of $R = LL^T$



Figure 2.7: Schematic representation of FORM and SORM approximations with a mapping from the physical to the standard Gaussian space.

• Non-normally distributed and independent random variables. In this case, an isoprobabilistic transformation can be performed. Let F_{W_i} be the marginal CDF of the original random variables. By requiring that $\Phi(u_i) = F_{W_i}(w_i)$, the following mapping can be derived:

$$u_{i} = \Phi^{-1} \left[F_{W_{i}} \left(w_{i} \right) \right],$$

$$w_{i} = F_{W_{i}}^{-1} \left[\Phi \left(u_{i} \right) \right]$$
(2.62)

• General case: the Nataf and Rosenblatt transformations. For the most general case, the Nataf transformation (Nataf, 1962; Liu and Der Kiureghian, 1986, 1992) proceeds in two steps, combining the two previous transformations. The mapping may be defined as the composition of two functions (Lebrun and Dutfoy, 2009a):

$$\mathbb{T}^{N} = \mathbb{T}_{2}^{N} \circ \mathbb{T}_{1}^{N} \quad \text{such that:} \quad \mathbb{T}_{1}^{N} : \boldsymbol{W} \mapsto \boldsymbol{Y} = \left(\Phi^{-1}\left[F_{W_{1}}\left(w_{1}\right)\right], \dots, \Phi^{-1}\left[F_{W_{s}}\left(w_{s}\right)\right]\right)^{T} \quad (2.63)$$
$$\mathbb{T}_{2}^{N} : \boldsymbol{Y} \mapsto \boldsymbol{U} = \boldsymbol{\Gamma}\boldsymbol{Y},$$

where Γ is any square-root of R_0 , the correlation matrix computed between components of Y. Similarly, one may use the Rosenblatt transformation (Rosenblatt, 1952; Lebrun and Dutfoy, 2009b) which involves a recursive formula defined by:

$$\mathbb{T}^{R} = \mathbb{T}_{2}^{R} \circ \mathbb{T}_{1}^{R} \quad \text{such that:} \quad \mathbb{T}_{1}^{R} : \mathbf{W} \mapsto \mathbf{Y} = \begin{pmatrix} F_{W_{1}}(w_{1}) \\ F_{W_{2}|W_{1}}(w_{2}) \\ \vdots \\ F_{W_{s}|W_{1},\dots,W_{s-1}}(w_{s}) \end{pmatrix}$$
(2.64)
$$\mathbb{T}_{2}^{R} : \mathbf{Y} \mapsto \mathbf{U} = \left(\Phi^{-1}(Y_{1}), \Phi^{-1}(Y_{2}), \dots \Phi^{-1}(Y_{s})\right)^{T}$$

35

where $F_{W_s|W_1,...,W_{s-1}}$ is the cumulative distribution function of the conditional random variable $W_s|W_1,...,W_{s-1}$.

Crude Monte-Carlo Sampling

Simulation techniques have been used as well for estimation of failure probabilities. Their principle relies on rewriting the integral in Eq. (2.50) as:

$$P_{f} = \int_{\mathfrak{g}(W) \leq 0} f_{W}(w) \, dw = \int_{\mathcal{D}_{W}} \mathbb{I}\left(\mathfrak{g}\left(w\right)\right) f_{W}(w) \, dw \equiv \mathbb{E}_{f_{W}}\left[\mathbb{I}\left(\mathfrak{g}\left(W\right)\right)\right], \tag{2.65}$$

where \mathcal{D}_W is the support of the random vector W and \mathbb{I} is the indicator function which is equal to 1 if $\mathfrak{g}(w) \leq 0$ and to 0 otherwise. This way, the hardly tractable integration in Eq. (2.50) becomes the expectation of the indicator function which can be estimated more easily. In fact, crude Monte Carlo simulation consists in empirically estimating the probability of the failure by:

$$P_{f_{MCS}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\mathfrak{g}\left(\boldsymbol{w}_{i}\right)\right] = \frac{n_{f}}{N}$$
(2.66)

where $\{w_i, i = 1, ..., N\}$ is a sequence on *N* independent realizations of the random vector *W* and n_f is the number of failed samples, *i.e.* such that $\mathfrak{g}(w_i) \leq 0$.

It can be shown that $P_{f_{MCS}}$ is an unbiased estimator of the failure probability P_f . We are indeed assured by the law of large numbers that $P_{f_{MCS}}$ will eventually converge to P_f as N grows. The remaining question is how fast. The answer lies in the variance of the estimator which reads:

$$\operatorname{Var}\left[P_{f_{MCS}}\right] = \frac{1}{N} P_f\left(1 - P_f\right).$$
(2.67)

The exact value of P_f is actually not known. Anyway, considering a target coefficient of variation, we can estimate how many samples are needed to compute a given probability of failure within some prescribed accuracy. Indeed the coefficient of variation may be cast as:

$$\delta_{MCS} = \frac{\sqrt{\frac{1}{N}P_f(1-P_f)}}{P_f} = \sqrt{\frac{1-P_f}{NP_f}}.$$
(2.68)

From this relationship, one can conversely find the number of samples required for a given probability of failure and coefficient of variance:

$$N = \frac{1 - P_f}{\delta_{MCS}^2 P_f}.$$
(2.69)

For instance, if we look for a probability of failure of 5% with a coefficient of variation of 5%, we would need approximately 7,600 sampling points, which is quite reasonable. However, looking for a failure probability of 10^{-6} with the same level of accuracy would require 4×10^8 samples. This is hardly tractable even if g is cheap to evaluate. Other simulation techniques have therefore been developed specifically for such purposes. They are known as *variance reduction* techniques and implement various strategies to produce an equally accurate estimator as Monte Carlo but with fewer samples. Well-known examples are importance sampling, stratification (such as Latin hypercube sampling), splitting, control variates or conditional Monte Carlo as described in Asmussen and Glynn (2007) (chapter 5). Most of them are general-purpose algorithms but we present them here in the context of reliability analysis.

Importance sampling

Importance sampling, applied to reliability analysis, is aimed at by-passing the difficulty associated to drawing samples in the failure domain when concerned with rare events. By definition, such points are scarce in a crude Monte Carlo simulation. The idea is henceforth to draw instead from another distribution which puts higher weight to samples in the failure domain and then to correct the bias introduced by replacing the original distribution.

Let us consider the integral in Eq. (2.65). Importance sampling proceeds by recasting it as:

$$P_{f} = \int_{\mathcal{D}_{W}} \mathbb{I}\left[\mathfrak{g}\left(w\right)\right] \frac{f_{W}\left(w\right)}{h_{W}\left(w\right)} h_{W}\left(w\right) dw \equiv \mathbb{E}_{h}\left[\mathbb{I}\left[\mathfrak{g}\left(w\right)\right] \frac{f_{W}\left(w\right)}{h_{W}\left(w\right)}\right]$$
(2.70)

where $h_W(w)$ is a so-called *instrumental* PDF easy to sample from. It must be chosen such that there is an algorithm to draw $w_i \sim h_W$ and that it is possible to evaluate the ratio f_W/h_W at any w_i (Owen and Zhou, 2000).

In practice, the probability of failure is obtained by:

$$P_{f_{IS}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(\boldsymbol{w}_{i}\right) \frac{f_{W}\left(\boldsymbol{w}_{i}\right)}{h_{W}\left(\boldsymbol{w}_{i}\right)}$$
(2.71)

where $\{w_i, i = 1, ..., N\}$ are independent and identically distributed samples simulated according to the PDF h_W . If the instrumental PDF is chosen appropriately, by the law of large numbers, the estimated probability of failure will converge to the true one. Its variance can be derived as follows:

$$\operatorname{Var}\left[P_{f_{IS}}\right] = \operatorname{Var}\left[\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left(w_{i}\right)\frac{f_{W}\left(w_{i}\right)}{h_{W}\left(w_{i}\right)}\right] = \frac{1}{N^{2}}\sum_{i=1}^{N}\operatorname{Var}\left[\mathbb{I}\left(w_{i}\right)\frac{f_{W}\left(w_{i}\right)}{h_{W}\left(w_{i}\right)}\right]$$
$$= \frac{1}{N^{2}}N\operatorname{Var}\left[\mathbb{I}\left(W\right)\frac{f_{W}\left(W\right)}{h_{W}\left(W\right)}\right]$$
$$= \frac{1}{N}\left(\mathbb{E}_{h}\left[\left(\mathbb{I}\left[\mathfrak{g}\left(w\right)\right]\frac{f_{W}\left(w\right)}{h_{W}\left(w\right)}\right)^{2}\right] - \left(\mathbb{E}_{h}\left[\mathbb{I}\left[\mathfrak{g}\left(w\right)\right]\frac{f_{W}\left(w\right)}{h_{W}\left(w\right)}\right]\right)^{2}\right)$$
$$= \frac{1}{N}\left(\mathbb{E}_{h}\left[\mathbb{I}\left[\mathfrak{g}\left(w\right)\right]\left(\frac{f_{W}\left(w\right)}{h_{W}\left(w\right)}\right)^{2}\right] - P_{f}^{2}\right)$$
$$(2.72)$$

The choice of the instrumental PDF is crucial in order to achieve a variance reduction. Examples of appropriate choices can be found in Asmussen and Glynn (2007). In the case when FORM is performed, a wise choice would be a multivariate normal distribution with mean centered on the design point as proposed in Schuëller and Stix (1987). This assumes that the MPFP has been found in a first stage. The method hence inherits the shortcomings of FORM *i.e.* those related to finding a unique, in other words global, design point. In general, when the dimension of the problem is high, finding an appropriate proposal distribution is not trivial. Bucher (1988) has introduced a parametrization of h_W , where the parameter is tuned adaptively during the simulation. Au and Beck (1999) also implement an adaptive importance sampling using Monte Carlo pre-samples.

We now illustrate importance sampling with the capacity-and-demand example. We consider that the two parameters follow a normal distribution: $R \sim \mathcal{N}(\mu_r, \sigma_r)$ and $S \sim \mathcal{N}(\mu_s, \sigma_s)$. For

this particular case, the probability of failure can be calculated analytically. In fact, the limit state surface in the standard space reads:

$$\mathfrak{g}_{u}\left(\boldsymbol{u}\right) = \sigma_{r}u_{1} - \sigma_{s}u_{2} + \left(\mu_{r} - \mu_{s}\right), \qquad (2.73)$$

where $u_1 = (r - \mu_r) / \sigma_s$ and $u_2 = (s - \mu_s) / \sigma_s$ are the random variables expressed in the standard space. The distance from this hyperplane to the origin is the reliability index and reads:

$$\beta = \frac{\mu_r - \mu_s}{\sqrt{\sigma_r^2 + \sigma_s^2}}.$$
(2.74)

Let $R \sim \mathcal{N}(100, 10)$ and $S \sim \mathcal{N}(60, 6)$. The exact probability of failure in this case is $P_f = 3.0182 \times 10^{-4}$. Using a set of 5,000 samples, we estimate this probability using both crude Monte Carlo and importance sampling. By repeating, the simulation 10,000 times, we obtain the following means $P_{f_{MCS}} = 2.9938 \times 10^{-4}$ and $P_{f_{IS}} = 3.0170 \times 10^{-4}$. The great difference lies in the coefficient of variations which are respectively $\delta_{MCS} = 81.18\%$ whereas $\delta_{IS} = 2.76\%$. From this result, it is obvious how crude Monde Carlo is inappropriate for such a low probability of failure given the limited number of samples.

Subset simulation

Subset simulation (Au and Beck, 2001), also known as multi-level splitting, is another simulation technique which starts from the premise that it is not trivial to obtain samples in the failure domain under crude Monte Carlo sampling when it comes to rare events. The method then suggests to compute small failure probabilities as the product of larger conditional ones. In order to achieve this, the space is split into a sequence of nested domains $F_1 \supset F_2 \supset ... \supset F_m$ where $F_i = \{W : g(W) \le y_i, i = 1, ..., n\}$ is an intermediate event. The thresholds y_i are chosen such that $y_1 > y_2 > ... > y_m = 0$. The probability of failure is henceforth computed as follows:

$$P_{f} = \mathcal{P}(F_{m}) = \mathcal{P}(F_{m}) \mathcal{P}(F_{m}|F_{m-1}) = \dots$$

$$= \mathcal{P}(F_{m}) \mathcal{P}(F_{m}|F_{m-1}) \dots \mathcal{P}(F_{2}|F_{1}) \mathcal{P}(F_{1})$$

$$= \mathcal{P}(F_{1}) \prod_{i=1}^{m-1} \mathcal{P}(F_{i+1}|F_{i}).$$

(2.75)

The events F_i are set such that each term of the product is computed at a low cost. This begs the question of how to choose the intermediate thresholds. One approach is to set a priori the values of y_i . However by doing so, one does not control the intermediate probabilities of failure and may therefore end up with another too small probability to estimate. The optimality of the procedure is hence not achieved. Au and Beck (2001) proposes to rather set a fixed value of the conditional probabilities, *i.e.* $\mathcal{P}(F_i|F_{i-1}) = p_0 \in [0, 1]$. This is achieved by adaptively setting each threshold as the $(1 - p_0)$ -quantile of the samples at each iteration. Cérou et al. (2012) proves that the variance of the estimated probability of failure is optimal if the levels are evenly spaced in terms of conditional probabilities. However, such a sequence is not always easy to achieve in practice.

Another issue is the simulation of the conditional probabilities. A naive Monte Carlo, although technically possible, would not be efficient. Instead, subset simulation implementations rather resort to the so-called *Markov Chain Monte Carlo* (MCMC) algorithms (Andrieu et al., 2003;

Metropolis et al., 1953). Basically MCMC is used when one can evaluate a PDF up to a normalizing constant but cannot sample from it. Using a Markov Chain mechanism, MCMC allows one to draw a sequence of samples whose stationary distribution is the target PDF. In the case of subset simulation, Au and Beck (2001) used a modified Metropolis algorithm to generate samples with density distribution conditional to a given event F_i . In practice, the initial probability $p_1 = \mathcal{P}(F_1)$ is obtained through a direct Monte Carlo simulation. Then the conditional probabilities are sequentially computed though MCMC. At the *i*-th conditional level, let $\{w_k^{(i)}, k = 1, ..., N\}$ be *N* samples generated by MCMC conditionally to F_i . The conditional probability therefore reads:

$$\mathcal{P}\left(F_{i+1}|F_{i}\right) \approx p_{i+1} = \frac{1}{N} \mathbb{I}\left[\boldsymbol{w}_{k}^{(i)} \in F_{i+1}\right].$$
(2.76)

The sequential procedure is stopped when a level $y_i < 0$ is reached. This last threshold is then replaced by $y_m = 0$. The estimated probability of failure eventually reads:

$$P_{f_{SS}} = p_1 \prod_{i=2}^{m} p_i.$$
(2.77)

This estimator can be accurate when the thresholds are appropriately set. However, it is biased due to the intermediate quantile estimations (Walter and Defaux, 2015).

The techniques presented above can be used to compute probabilities of failure. The choice of the appropriate one depends on the context and the characteristics of the problem at hand *e.g.* linearity of the limit state, number of samples allocated by the computational budget, target probability of failure, etc. In the next section, these methods are embedded in optimization techniques so as to solve the RBDO problem.

2.2.2.4 Solution of the RBDO problem

Over the past few decades, various methods combining reliability analysis and optimization have been proposed to efficiently solve RBDO problems. The present section reviews some of them following Tsompanakis et al. (2008), more specifically the chapter by Chateauneuf and Aoues (2008). A benchmark study is also performed in Aoues and Chateauneuf (2010). There, the authors propose a general classification of RBDO methods into three groups: *two-level, mono-level* and *decoupled* approaches. We adopt this classification in this thesis.

Two-level approach

Two-level approaches are among the most direct and probably natural ways to solve the RBDO problem. They consist of two loops. The outer loop naturally explores the design space by iterating on different values of *d* while the inner loop performs a reliability analysis. The latter is usually also an optimization problem carried out in the random variables space when MPFP-based methods are used, thus making the two-level approach a nested optimization problem. However, simulation techniques can also be used instead while keeping the same global strategy. Two main formulations have been developed in this category, namely the *reliability index approach* (RIA) and the *performance measure approach* (PMA).

The reliability index approach (Enevoldsen and Sorensen, 1994) consists in using the FORM analysis in the inner loop. For the sake of mathematical convenience, the probability of failure

is equivalently replaced by the reliability index. The optimization problem therefore reads:

$$d^* = \arg\min_{\boldsymbol{d}\in\mathbb{D}}\mathfrak{c}(\boldsymbol{d}) \quad \text{subject to:} \begin{cases} f_j(\boldsymbol{d}) \leq 0, & \{j=1,\ldots,n_s\}\\ \bar{\beta}_k - \beta_k \leq 0, & \{k=1,\ldots,n_h\} \end{cases}.$$
 (2.78)

where $\beta_k = -\Phi^{-1}(\mathcal{P}(\mathfrak{g}_k(X(d), \mathbb{Z}) \leq 0))$ is the generalized reliability index already introduced and $\bar{\beta}_k = -\Phi^{-1}(\bar{P}_{f_k})$ is the corresponding threshold. From now on we consider that $n_h = 1$ for the sake of clarity.

The reliability index is computed after the MPFP has been found in the standard Gaussian space following Eq. (2.53). As a reminder it writes:

$$u_{MPFP}^{*} = \arg\min \|u\|$$
 subject to: $\mathfrak{g}_{u}(u) \leq 0$,

and the reliability index is hence given by $\beta = \alpha^T u_{MPFP}^*$.

This approach requires to compute the sensitivities of the reliability index with respect to the random variables. As argued in Enevoldsen and Sorensen (1994), they must be accurate for stability of the optimization algorithm used.

An alternative formulation, known as the performance measure approach has been introduced by Tu and Choi (1997). In PMA, the reliability problem is taken from another perspective. Instead of searching to minimize the reliability index under non-linear (possibly complex) constraints, one rather sets a target reliability index. That is, the reliability index is set at its target value and the limit state function is minimized. The formulation is as follows:

$$\boldsymbol{u}_{MPTP}^{*} = \arg\min_{\boldsymbol{u}} \mathfrak{g}_{\boldsymbol{u}}\left(\boldsymbol{u}\right) \quad \text{subject to:} \quad \|\boldsymbol{u}\| = \bar{\beta},$$
 (2.79)

where the subscript MPTP stands for *minimum performance target point*.

The introduced spherical constraint is easier to handle than the constraint of the RIA. PMA only requires sensitivity of the limit state function with respect to design variables. The two points make that PMA has generally a better convergence behavior than RIA and requires fewer iterations (Lee et al., 2002). Tu et al. (1999) also show that the RIA can yield to singularity issues when evaluating the probabilistic constraint. This occurs when the failure probability for a given design is zero. In such a case, the reliability index numerically approaches infinity and the RIA can fail to converge. On the other hand, thanks to the spherical constraint, PMA can benefit from specific algorithms which are fitted for such constraints *e.g.* advanced mean value (Wu et al., 1990) or more recently enhanced hybrid mean-value (Youn et al., 2005).

As pointed out in Tu et al. (1999), RIA and PMA are identical when all the constraints are active. They however differ when some constraints are inactive or violated. The numerical instability of RIA occurs in the presence of inactive constraints. Lee et al. (2002) compare the two approaches on three different problems and argue that PMA is numerically more advantageous than RIA. Conceptually, the two approaches are illustrated in Figure 2.8. While RIA searches for the smallest reliability index under the subset $\{u \in \mathbb{R}^s | \mathfrak{g}_u (u) = 0\}$, PMA rather searches the minimum performance point under the subset $\{u \in \mathbb{R}^s | \|u\| = \bar{\beta}\}$.

Mono-level approach

Because of the computational burden associated to the nested analyses above and scaling issues, Madsen and Hansen (1992) have proposed a single-loop approach. This method is



Figure 2.8: Comparison of the reliability index and the performance measure approaches.

based on Karush-Kuhn-Tucker optimality conditions of the design point. In fact, if u^* is the MPFP resulting from the reliability analysis, the necessary KKT conditions of Eq. (2.53) lead to:

$$\boldsymbol{u}^{*T} \nabla_{\boldsymbol{u}} \mathfrak{g}_{\boldsymbol{u}} \left(\boldsymbol{u} \right) + \beta \| \nabla_{\boldsymbol{u}} \mathfrak{g}_{\boldsymbol{u}} \left(\boldsymbol{u} \right) \| = 0 \tag{2.80}$$

Following the idea of Madsen and Hansen (1992), Kuschel and Rackwitz (1997) have proposed a mono-level formulation which can be stated as follows:

$$d^* = \arg\min_{d\in\mathbb{D}} \mathfrak{c}(d) \quad \text{subject to:} \begin{cases} \mathfrak{f}_j(d) \leq 0, & \{j=1,\ldots,n_s\},\\ \mathfrak{g}_u(u) = 0, \\ u^T \nabla_u \mathfrak{g}_u(u) + \|u\| \| \nabla_u \mathfrak{g}_u(u) \| = 0,\\ \bar{\beta} - \beta \leq 0. \end{cases}$$
(2.81)

It is worth noting that Kuschel and Rackwitz (1997) originally formulated the problem with respect to a total cost. That is, the initial cost considered here plus an expected cost of failure. We do not consider such a formulation in this thesis.

The advantage of this formulation is that any general-purpose non-linear optimization algorithm can be used. However, to this end it is necessary to provide explicit mapping from the physical to the standard space (and vice-versa). Both papers show that convergence is not straightforward as often tricks are needed to avoid numerical instability (such as monotonic transformation of the cost or constraints). More importantly, the reformulation is based on RIA which has been shown to be numerically unstable. Agarwal et al. (2007) have proposed a similar mono-level approach but which is rather based on the PMA formulation. Contrary to the previous work on RIA, they also provide conditions under which the two-level and monolevel formulations are equivalent (this involves convexity assumptions on the constraints in the standard space). They also restate the inverse FORM in Eq. (2.79) as:

$$u^* = \arg\min_{u} \mathfrak{g}_u(u) \quad \text{subject to:} \quad \|u\| \le \overline{\beta},$$
 (2.82)

following the recommendation in Polak et al. (2000) (and other similar contributions) which argue that inequality constraints are handled easier than equality in this case. The mono-level approach can henceforth be stated as:

$$d^{*} = \arg\min_{d\in\mathbb{D}}\mathfrak{c}(d) \quad \text{subject to:} \begin{cases} \mathfrak{f}_{j}(d) \leq 0, & \{j=1,\ldots,n_{s}\},\\ \mathfrak{g}_{u}(u) \geq 0, \\ \nabla_{u}\mathfrak{g}_{u}(u) + \lambda \frac{u}{\|u\|} = 0, \\ \beta - \bar{\beta} \leq 0, \\ \lambda (\beta - \bar{\beta}) = 0. \end{cases}$$
(2.83)

where $\lambda \ge 0$ is a Lagrange multiplier. This formulation hence increases the dimension of the optimization problem. It is however shown to be numerically more stable than that of Eq. (2.81).

From another perspective, other approaches have been developed that convert the double-loop into a single-loop approach. Chen et al. (1997) formulates a problem where it is assumed that all random variables are associated to design parameters (*i.e.* no environmental variables). Under this assumption, the probabilistic constraint is transformed into an equivalent deterministic one. The idea is to evaluate the constraint at an approximated most probable failure point. Thanks to the Gaussian assumption, the relation between the most probable failure point $x^{(i)}$ and the design variable $d^{(i)}$ is straightforward. The equivalent deterministic formulation therefore reads:

$$d^* = \arg\min_{d \in \mathbb{D}} \mathfrak{c}(d) \quad \text{subject to:} \begin{cases} \mathfrak{f}_j(d) \leq 0, & \{j = 1, \dots, n_s\} \\ \mathfrak{g}\left(d^{(i)}, x^{(i)}\right) \geq 0, \end{cases}$$
(2.84)

where $\mathbf{x}^{(i)}$ is the approximated MPFP at iteration *i* and $\mathbf{d}^{(i)} = \boldsymbol{\mu}^{(i)}$ is the corresponding design variable. The MPFP is approximated at each iteration as follows:

$$\boldsymbol{x}^{(i)} = \boldsymbol{\mu}_{\boldsymbol{X}}^{(i)} - \bar{\boldsymbol{\beta}} \sigma_{\boldsymbol{X}} \boldsymbol{\alpha}^{(i-1)},$$

$$\boldsymbol{\alpha}^{(i)} = \frac{\nabla_{\boldsymbol{x}} \boldsymbol{\mathfrak{g}} \left(\boldsymbol{d}^{(i)}, \boldsymbol{x}^{(i)} \right)}{\|\nabla_{\boldsymbol{x}} \boldsymbol{\mathfrak{g}} \left(\boldsymbol{d}^{(i)}, \boldsymbol{x}^{(i)} \right)\|}.$$
(2.85)

This method is known as the *single loop single vector* approach. Liang et al. (2007) have proposed a similar method known as the *single loop approach* as an extension to series system. Wang and Kodiyalam (2002) have extended the approach to the case of random variables which are not normally distributed. From another perspective Kharmanda et al. (2002) proposed to solve the problem in a *hybrid design space*.

Decoupled approach

While the mono-level approach seeks to avoid reliability analysis, the decoupled approach rather seeks to solve an equivalent deterministic problem followed (or preceded) by a reliability analysis. This sequential strategy is aimed at reducing the overall cost with respect to a standard double-loop approach. One of the most popular methods that fall into this category

is the *sequential optimization and reliability assessment* (SORA) developed by Du and Chen (2004). In SORA, successive cycles of deterministic optimization and reliability analysis are performed. The probabilistic constraints are equivalently expressed by deterministic ones thanks to the inverse FORM method. In fact, the following equivalence is used:

$$\mathcal{P}\left(\mathfrak{g}\left(\boldsymbol{X},\boldsymbol{Z}\right)\leq 0\right)\leq\bar{P}_{f}\Leftrightarrow\mathfrak{g}\left(\boldsymbol{x}_{MPTP},\boldsymbol{z}_{MPTP}\right)\leq0\tag{2.86}$$

where $w_{MPTP} = \{x_{MPTP}, z_{MPTP}\}$ is the minimum performance target point corresponding to the target reliability $1 - \bar{P}_f$.

At each cycle, SORA solves the deterministic optimization by shifting the design variables according to the MPTP found on the previous cycle. At cycle *i*, the problem reads:

$$d^* = \arg\min_{d \in \mathbb{D}} \mathfrak{c}(d) \quad \text{subject to:} \begin{cases} \mathfrak{f}_j(d) \leq 0 & \{j = 1, \dots, n_s\} \\ \mathfrak{g}\left(d - s^{(i)}, z^{(i-1)}\right) \geq 0 \end{cases}$$
 (2.87)

where $s^{(i)} = d^{(i-1)} - x^{(i-1)}_{MPTP}$. At the first iteration, the MPTP is taken equal to the mean values of the design and environmental variables.

By repeating these cycles, SORA eventually converges to the sought optimum. The number of iterations may however be important. Cho and Lee (2011) have proposed an enhanced alternative which combines SORA and convex linearization, thus reducing the number of calls to the constraint functions.

Another family of decoupled approach has been developed based on the approximation of the probabilistic constraints *e.g.* Taylor series expansion of the performance functions as proposed by Chandu and Grandhi (1995) or the *sequential approximate programming* proposed by Cheng et al. (2006).

2.2.3 Concluding remarks

This section was concerned with optimization under uncertainties. Two main concepts were introduced: robustness and reliability. Robust design optimization seeks to produce a design that has little or no sensitivity to uncertainties. Various formulations, relying on different robustness measures, exist. In contrast, reliability-based design optimization seeks to produce a design that can still perform adequately despite the presence of uncertainties. This translates into the specification of a probability of failure that is set arbitrarily low. The main difference between RDO and RBDO is that the former focuses on the objective function whereas the latter is concerned only with the constraints.

For the applications in this thesis, the effect of uncertainties on the objective function is negligible. For the lightweight design of the automotive body structure for instance, the objective function is the weight of some parts of the body-in-white. The possible scatter in the design variables, *i.e.* the thicknesses of the parts, may only marginally affect the final weight. This is therefore not a concern for the designers. In contrast, the constraints are highly sensitive to uncertainties because of the chaotic nature of crash, as already argued. We therefore adopt the reliability-based design optimization approach in order to integrate these uncertainties. Furthermore, we introduce another formulation based on quantiles and Monte Carlo for reasons that will become clear in the next section.

2.3 Quantile-based RBDO

In this section, we propose another formulation of RBDO which relies on the quantile of the constraints. We first motivate this choice and then reformulate the problem in Eq. (2.49) accordingly.

2.3.1 Motivation

The methods presented above for RBDO have been successfully used in practice. The success of each method relies upon the conditions in which it is used. For instance, it is widely accepted that FORM is more likely to perform poorly when the limit state surface is highly non-linear. Moreover, its implementation requires a few steps which might be hard to implement within a general-purpose optimization framework. Double-loop approaches with Monte-Carlo sampling on the other hand provide more accurate results. However, MCS becomes prohibitively expensive when the target failure probability is low. Variance reduction techniques may serve as substitute in this case. They however also require intermediate steps which are hard to integrate in a general-purpose optimization algorithm.

In this thesis, we aim at proposing an RBDO formulation in an industrial context, this while remaining within the prevailing culture and practice. The proposed methodology must remain as close as possible to the current deterministic optimization so as to be easily implemented by designers and this as straightforward as possible. With this, we start by emphasizing a particular aspect of the problem we aim to solve. The DDO has proven to be inefficient because of the scatter that can be observed in frontal impact experiments and simulations. RBDO is therefore considered as a means to account for this scatter for a safe design within a probabilistic context. To this end, the uncertainties in the design variables and environmental conditions are modeled by random variables. These uncertainties are propagated to the outputs of the constraints. Target probabilities of failure are set as a means to be conservative. Their sole role is conservatism: we are not concerned with rare events so the failure probabilities thresholds are set relatively high. Their order of magnitude is indeed 1% – 10%. In such a context we do not need any sophisticated variance reduction technique. Crude Monte Carlo sampling allows us to accurately estimate the failure probabilities within a reasonable computational budget.

Moreover, instead of explicitly expressing the probabilistic constraint in terms of a probability of failure, we rather consider quantiles of the constraints distribution. In this way, the computed quantile can be directly plugged in the workflow of the already existing deterministic design process without difficulty. This is therefore a double-loop approach. While the outer loop explores the design space, the inner loop simply reduces to the computation of the quantile of the constraints.

2.3.2 Formulation

To formulate the quantile-based RBDO method, let us first restate the probabilistic constraint. Considering the mathematical model M_k which represents the system output , the following equivalence holds:

$$\mathcal{P}\left(\mathfrak{g}_{k}\left(\boldsymbol{X}\left(\boldsymbol{d}\right),\boldsymbol{Z}\right)\leq0\right)\leq\bar{P}_{f_{k}}\Leftrightarrow\mathcal{P}\left(\mathcal{M}_{k}\left(\boldsymbol{X}\left(\boldsymbol{d}\right),\boldsymbol{Z}\right)\geq\bar{\mathfrak{g}_{k}}\right)\leq\bar{P}_{f_{k}},$$
(2.88)

where $\bar{\mathfrak{g}}_k$ is the threshold not to be exceeded.

Let us now define the quantile as an alternative to express the probability of failure:

$$Q_{\alpha_{k}}\left(d;\mathcal{M}_{k}\left(X\left(d\right),Z\right)\right) = \inf\left\{\mathfrak{q}_{\alpha_{k}} \in \mathbb{R} \middle| \mathcal{P}\left(\mathcal{M}_{k}\left(X\left(d\right),Z\right) \leq \mathfrak{q}_{\alpha_{k}}\right) \geq \alpha_{k}\right\},\tag{2.89}$$

where $\alpha_k = 1 - \bar{P}_{f_k}$.

For a given design $d^{(i)}$ during the optimization process, this quantile is computed by sampling a Monte Carlo population of size N following the distribution of the random variables $X \sim f_{X|d}$ and $Z \sim f_Z$:

$$\mathfrak{C}_q\left(\boldsymbol{d}^{(i)}\right) = \left\{ \left(\boldsymbol{x}^{(j)}, \boldsymbol{z}^{(j)}\right), j = 1, \dots, N \right\}.$$
(2.90)

We emphasize that $x^{(j)}$ depends on the design variable $d^{(i)}$. It is assumed that the input parameters are independent. For numerical stability, this Monte Carlo simulation is generated using *common random numbers*. That is, the same seed is used to generate the samples within iterations of the optimization algorithm. As a consequence, the same realizations of Z are used throughout the optimization. This has shown to improve convergence of the algorithm.

To estimate the quantile, the mechanical model is evaluated on these samples resulting in a set of points $\{y^{(j)} = \mathcal{M}_k(x^{(j)}, z^{(j)}), j = 1, ..., N\}$ and eventually giving a distribution of the mechanical response. These points are ranked in ascending order such that $y_{(1)} \leq y_{(2)} \leq ... \leq y_{(N)}$. Eventually, the quantile is estimated as:

$$Q_{\alpha_k}\left(\boldsymbol{d}^{(i)}; \mathcal{M}_k\left(\boldsymbol{X}\left(\boldsymbol{d}^{(i)}\right), \boldsymbol{Z}\right)\right) \equiv \mathfrak{q}_{\alpha_k}\left(\boldsymbol{d}^{(i)}\right) = y_{(\lfloor N\alpha_k \rfloor)},$$
(2.91)

where $|\bullet|$ denotes the floor function. In general, we consider $N = 10^4$.

Using the above notation, the target probability of failure can be equivalently expressed by a constraint on quantiles as follows:

$$\mathcal{P}\left(\mathcal{M}_{k}\left(\boldsymbol{X}\left(\boldsymbol{d}^{(i)}\right),\boldsymbol{Z}\right) \geq \bar{\mathfrak{g}}_{k}\right) = \bar{P}_{f_{k}} \Leftrightarrow q_{\alpha_{k}}\left(\boldsymbol{d}^{(i)}\right) = \bar{\mathfrak{g}}_{k}.$$
(2.92)

This equivalence is illustrated in Figure 2.9 where the distributions of the response are plotted for two different design variables. The quantiles corresponding to target failure probabilities are thereafter compared with the threshold \bar{g}_k resulting in two configurations which are safe and unsafe. The median value $q_{0.5}$ is also illustrated. It may for instance correspond to the value checked in deterministic design optimization, although this is not systematic. In this situation, both configurations would have been considered safe.

Following the above assessment of safety, the RBDO problem in Eq. (2.49) can eventually be reformulated as:

$$d^{*} = \arg\min_{\boldsymbol{d}\in\mathbb{D}}\mathfrak{c}(\boldsymbol{d}) \quad \text{subject to:} \begin{cases} \mathfrak{f}_{j}(\boldsymbol{d}) \leq 0, & \{j=1,\ldots,n_{s}\}\\ Q_{\alpha_{k}}(\boldsymbol{d};\mathcal{M}(\boldsymbol{X}(\boldsymbol{d}),\boldsymbol{Z})) \leq \tilde{\mathfrak{g}}_{k}, & \{k=1,\ldots,n_{h}\} \end{cases}$$
(2.93)

We finally consider that the failure modes are independent. The system fails when at least one constraint is not satisfied. This corresponds to a *series system* as opposed to *parallel systems* where all constraints need to be violated for the failure system to occur.



Figure 2.9: Definition of a quantile of the response with two configurations corresponding to safe and unsafe designs.

2.3.3 Analytical examples

We now illustrate the RBDO methodology on some validation examples. Four analytical cases are considered: two with mathematical functions and two with mechanical models. The first mathematical example only serves for illustration purpose. The remaining examples are used to validate the approach thanks to analytical solutions or by comparison with results available in the related literature.

2.3.3.1 Mathematical models

Two-dimensional non-linear limit state function

This example is meant to illustrate the methodology, especially the computation of the quantile. The function will be a running example for the developments in this thesis. The mathematical model comes from Janusevskis and Le Riche (2013) and reads:

$$\mathcal{M}(d,z) = \left(\frac{1}{3}z^4 - 2.1z^2 + 4\right)z^2 + dz + 4d^2\left(d^2 - 1\right),$$
(2.94)

where $d \in [-1, 1]$ and z is an environmental variable whose deterministic mean value is 0.5. In this reference, the authors used this function for robust optimization. Here we rather consider RBDO, so the optimization problem is defined differently and reads:

$$d^{*} = \arg\min_{d \in [-1,1]} (d+1)^{2} \quad \text{subject to:} \quad \mathcal{P}(0.5 - \mathcal{M}(X(d), Z) \le 0) \le 0.05$$
(2.95)

where $X \sim \mathcal{N}(d, 0.05^2)$ and $Z \sim \mathcal{N}(0.5, 0.05^2)$.

This probability of failure corresponds to a quantile value of 0.95. The equivalent quantilebased RBDO formulation therefore reads:

$$d^* = \arg\min_{d \in [-1,1]} (d+1)^2$$
 subject to: $Q_{\alpha}(d; \mathcal{M}(X(d), Z)) \le 0.5$ (2.96)

Constrained (1+1)-CMA-ES is used for optimization and is followed by a gradient-based approach to fine-tune the solution (MATLAB's fmincon function). Starting from an initial point

 $d^{(0)} = 0.6$, convergence is achieved after eight iterations of CMA-ES. The optimal solution is $d^* = -0.9322$.

Figure 2.10a shows the function $\mathcal{M}(x,z)$ in its space of definition. Two sets of Monte Carlo samples $\mathfrak{C}(d^{(0)})$ and $\mathfrak{C}(d^*)$ corresponding respectively to the initial and optimal points are plotted as blue and green crosses. The distributions used to generate them are also illustrated. The simulated quantile over the domain $\mathbb{D} = [-1,1]$ is shown in Figure 2.10b. The special cases of $\mathfrak{q}_{\alpha}(d^{(0)})$ and $\mathfrak{q}_{\alpha}(d^*)$ are also highlighted. If the estimation of the quantile is noiseless, the RBDO actually behaves as if we were optimizing with respect to d under the constraint defined by the function $\mathfrak{q}_{\alpha}(d)$. The solution d^* actually lies at the boundary of the feasible space delimited by the red line.



Figure 2.10: Two-dimensional mathematical illustration example.

Three non-linear limit state functions

This example involves a problem with three non-linear constraints. The optimization problem reads:

$$d^{*} = \arg\min_{d \in [0,10]^{2}} 10 - d_{1} + d_{2} \quad \text{s.t.:} \begin{cases} \mathfrak{g}_{1}(d) = \frac{d_{1}^{2}d_{2}}{20} - 1 \leq 0\\ \mathfrak{g}_{2}(d) = \frac{(d_{1} + d_{2} - 5)^{2}}{30} + \frac{(d_{1} - d_{2} - 12)^{2}}{120} - 1 \leq 0\\ \mathfrak{g}_{3}(d) = \frac{80}{(d_{1}^{2} + 8d_{2} + 5) - 1} \leq 0 \end{cases}$$

$$(2.97)$$

We consider the setting in Lee and Jung (2008) where an associated RBDO problem is solved.

The probabilistic model consists of the random design variables $X_i \sim \mathcal{N}(d_i, 0.6^2)$, $i = \{1, 2\}$. The authors consider a reliability index of $\beta = 2$ for all constraints which corresponds to a failure probability $P_{f_k} = 0.0228$, $k = \{1, 2, 3\}$. The corresponding quantile level is $\alpha_k = 1 - 0.0228 = 0.9772$, $k = \{1, 2, 3\}$. We may formulate the quantile-based RBDO as follows:

$$d^{*} = \arg\min_{d \in [0,10]^{2}} 10 - d_{1} + d_{2} \quad \text{s.t.:} \begin{cases} Q_{\alpha_{1}}(d; \mathcal{M}_{1}(X(d))) \leq 0, \\ Q_{\alpha_{2}}(d; \mathcal{M}_{2}(X(d))) \leq 0, \\ Q_{\alpha_{3}}(d; \mathcal{M}_{3}(X(d))) \leq 0. \end{cases}$$
(2.98)

Lee and Jung (2008) solve the problem using the performance mean approach and find an optimal design $d^* = \{5.28, 3.79\}^T$. We apply the quantile-based RBDO combined with constrained (1+1)-CMA-ES to solve the same problem. Figure 2.11 shows the history of the optimization. The left panel shows the cost function and the three limit state surfaces respectively in blue, red and yellow. The points sampled during the optimization by CMA-ES are also represented. The red color implies that the design is not feasible. The feasible samples with improved fitness are shown in green. The right panel simply shows the value of the cost function with respect to the iteration number. Due to its random nature, CMA-ES explores the design space by sampling points which do not improve the value of the cost function. The history of the feasible samples with improved fitness is also represented in green. Due to the convexity of the problem, a naive gradient-based algorithm would have been sufficient and more efficient in terms of function evaluations.



Figure 2.11: Three non-linear limit state functions

The algorithm converges to $d^* = \{5.29, 3.76\}^T$ which is quite close to the solution found in Lee and Jung (2008). Figure 2.12 illustrates the similarity between the quantile-based and the performance measure approaches used in Lee and Jung (2008). In this figure, the sets $\{d \in \mathbb{D} | \mathfrak{q}_{\alpha_k}(d) = 0, k = \{1, 2, 3\}\}$ are plotted in dashed lines. The optimum found is shown as a green diamond. The associated Monte Carlo, used to compute the quantile at this point is

represented as well. The blue color corresponds to points falling in the safe domain whereas the red one to those falling in the failure domain. As explained earlier, PMA searches the point of minimum performance on the circle of radius β in the standard Gaussian space. In the physical space, this circle is centered on the design point and has a radius of $0.6\beta = 1.2$ (0.6 being the standard deviation of the random design variables). At the optimal solution, it is shown as the black curve in Figure 2.12. This circle is, as expected, tangent to the two active constraints showing good convergence of the algorithm. In contrast, for the quantile-based approach, a MC centered on the optimum produces a given proportion of points that fall in the failure domain. This proportion actually corresponds to the target failure probability $\bar{P}_f = \Phi(-\bar{\beta})$.



Figure 2.12: Conceptual comparison of PMA and quantile-based approach on the three nonlinear mathematical model.)

2.3.3.2 Mechanical models

Column under compression

In this section we consider the optimal design of a column under compression as shown in Figure 2.13, also studied in Dubourg (2011). The column is of rectangular cross-section $b \times h$ and is submitted to a service load F_{ser} . We aim at minimizing this section while avoiding buckling. For such a column, buckling can occur if the service load is higher than a so-called *Euler critical force*. The later may be obtained by:

$$F_{cr} = \frac{\pi^2 EI}{L^2},\tag{2.99}$$

where *L* is the length of the column, *E* is the Young's modulus of its constitutive material and $I = bh^3/12$ (b > h) is the column area moment of inertia.



Figure 2.13: Column under compression

Table 2.1: Probabilistic model for the column under compression example.

Parameter	Distribution	Mean (μ)	COV (δ%)
k	Lognormal	0.6	10
E (MPA)	Lognormal	10000	5
L (mm)	Lognormal	3000	1
F _{ser} (kN)	_	$1.4622 imes 10^6$	_

To account for noise that may affect the Euler force, an additional multiplicative parameter k is considered. The deterministic optimization problem therefore reads:

$$d^{*} = \arg\min_{d \in [150,350]^{2}} bh \quad \text{subject to:} \begin{cases} f(d) = h - b \le 0, \\ g(d,z) = F_{ser} - k \frac{\pi^{2} E b h^{3}}{12L^{2}}, \end{cases}$$
(2.100)

where $z = \{k, E, L\}^T$ is the vector of environmental variables.

For the RBDO application, we consider that there is randomness only in the environmental variables. The probabilistic model is summarized in Table 2.1.

All distributions are lognormal. In this case, it is possible to analytically compute the solution of both the DDO and RBDO problems (Dubourg, 2011). As a matter of fact, they respectively read:

DDO solution

$$b^* = h^* = \left(\frac{12\mu_L^2 F_{ser}}{\pi^2 \mu_k \mu_E^2}\right)^{1/4}$$
(2.101)

50

RBDO solution

$$b^{*} = h^{*} = \frac{12F_{ser}}{\pi^{2}\exp\left(\lambda_{k} + \lambda_{E} - 2\lambda_{L} + \Phi^{-1}\left(P_{f}\right)\sqrt{\zeta_{k}^{2} + \zeta_{E}^{2} + 4\zeta_{L}^{2}}\right)},$$
(2.102)

where $\{\lambda_{\bullet}, \zeta_{\bullet}\}$, $\bullet = \{k, E, L\}$ are the parameters of the lognormal distributions. They can be retrieved by the transformation described in Eq. (2.33).

Considering a probability of failure $\bar{P}_f = 0.05$, the numerical application yields $b^* = h^* = 227.25$ mm for DDO and $b^* = h^* = 238.45$ mm for RBDO.

Alternatively, we may solve the problem using the quantile-based approach. Figure 2.14 shows the history plot of samples during the optimization. The red area denotes the failure domain due to the buckling while the blue one corresponds to the soft constraint b > h. The DDO solution lies at the boundary of the two areas. For RBDO, the solution remains on the boundary of the soft constraint. However, it is moved away from the hard constraint boundary. The found value $b^* = h^* = 238.59$ mm is close to the analytical one.



Figure 2.14: Convergence of the Euler problem using CMA-ES

Bracket structure

This last application example involves a two-member bracket structure proposed by Aoues and Chateauneuf (2010) which is represented in Figure 2.15. The member CD is horizontal and AB is inclined by 60° . The two are pin-joined at the point *B*. They support a vertical load *P* which is applied on D at a distance *L* of the hinge. The aim is to minimize the structure's weight under two constraints:



Figure 2.15: Illustration of the bracket structure (from Dubourg (2011))

• The first one is related to the bending stress in the member CD. Its maximal value σ_b , located at the point B, should be lower than the yield stress f_y . The maximum bending stress reads:

$$\sigma_b = \frac{6M_B}{w_{CD}}t^2,\tag{2.103}$$

where $M_B = PL/3 + \rho g w_{CD} t L^2/18$. In this equation, ρ is the unit mass of the bracket's constitutive material and *L* the length of the member CD as illustrated in Figure 2.15.

• The second constraint concerns the compression force *F*_{AB} in the member AB which must be lower than the Euler critical force of buckling *F*_b. The compression force reads:

$$F_{AB} = \frac{1}{\cos\theta} \left(\frac{3P}{2} + \frac{3\rho g w_{CD} t L}{4} \right), \qquad (2.104)$$

where θ is the inclination angle of the member AB which is equal to 60°. On the other hand, the critical load is given by:

$$F_b = \frac{\pi EI}{L_{AB}^2} = \frac{\pi^2 Et w_{AB}^3}{12 \left(2L/3 \sin\theta\right)^2},$$
(2.105)

where L_{AB} is the length of the member AB.

The optimization problem eventually reads:

$$d^{*} = \arg\min_{d \in [5,30]^{3}} \rho t L \left(\frac{4\sqrt{3}}{9} w_{AB} + w_{CD} \right) \quad \text{subject to:} \quad \begin{cases} f(d) = w_{AB} - t \le 0\\ \mathfrak{g}_{1}(d) = f_{y} - \sigma_{b} \ge 0\\ \mathfrak{g}_{2}(d) = F_{b} - F_{AB} \ge 0 \end{cases} \quad (2.106)$$

The RBDO application is constructed by a probabilistic model whose parameters are given in Table 2.2. The three random design parameters $d = \{w_{AB}, w_{CD}, t\}^T$ follow a normal distribution with a coefficient of variation of 5%. Five environmental variables are considered random. Their distributions and parameters are given in Table 2.2. The target reliability index is set at $\bar{\beta}_1 = \bar{\beta}_2 = 2$ corresponding to a failure probability of 0.023.

Parameter	Distribution	Mean	$\text{COV}(\delta\%)$	
Width of AB (w_{AB} in m)	Normal	w_{AB}	0.05	
Width of CD (w_{CD} in m)	Normal	w_{CD}	0.05	
Thickness (t in m)	Normal	t	0.05	
Applied load (<i>P</i> in kN)	Gumbel	100	0.15	
Young's modulus (E in GPa)	Gumbel	200	0.08	
Yield stress (f_{y} in MPa)	lognormal	225	0.08	
Unit mass (ρ in kg/m ³)	Weibull	7860	0.10	
Length (<i>L</i> in m)	Normal	5	0.05	

Table 2.2: Parameters of the variables defining the probabilistic model for the bracket structure problem.

We solve this problem using the quantile measure for reliability. Constrained (1+1)-CMA-ES is used as optimization algorithm. We compare the results with two reference solutions from the literature respectively that of Dubourg (2011) and Aoues and Chateauneuf (2010). The former author uses importance-sampling based on a global approximation of the limit state surface. The latter proposes several approaches among which we select the PMA. The solutions are given in Table 2.3. While PMA and Meta-RBDO give relatively close results, the quantile-based approach yields to a fundamentally different design. The resulting weight is lower than that of the two others. This may be explained by the fact that we use a global search algorithm, namely CMA-ES, in contrast to the two other reference solutions that were found with gradient-based approaches.

Table 2.3: Comparative results for the bracket structure. The PMA result comes from Aoues and Chateauneuf (2010) and Meta-RBDO from Dubourg (2011).

Design method	Weight (kg)	w_{AB} (cm)	w_{CD} (cm)	<i>t</i> (cm)	eta_1	β_2
PMA	1673	6.08	15.68	20.91	1.99	2.01
Meta-RBDO	1584	5.80	12.80	23.30	1.98	1.94
Quantile-based RBDO	1357	5.35	7.40	30.00	2.00	2.00

2.3.3.3 Summary

In this section, we have presented an alternative approach for solving RBDO problems which is based on quantiles. Some application examples were used to validate the approach. Three different cases for the probabilistic model were covered by the examples: randomness in design variables only, randomness in environmental variables only and randomness in both design and environmental variables. The first mathematical problem was used only for illustration purpose. The second one aims at validating the proposed approach by comparison with a reference solution found in the literature. The column under compression problem was introduced in order to validate the approach with respect to the exact analytical solution. Finally the bracket structure was included as an additional item in an existing benchmark.

Despite the approach is shown here to effectively be able to solve RBDO problems, it lacks of one crucial quality. In fact, we did not mention the computational cost of the method. Crude Monte Carlo sampling is indeed greedy in terms of functions evaluations. When the limit state functions are expensive to evaluate, this issue becomes prohibitive for the approach. It is the object of the next chapter to introduce techniques that will allow us to reduce the computational cost of the method.

2.4 Conclusion

This chapter has set the general framework for structural design optimization. The traditional deterministic viewpoint was first introduced. It was then argued why a deterministic design suffers from serious drawbacks. This argumentation relies upon the notion of model error and uncertainties which are inherent to any real-world problem. This naturally lead to the introduction of different approaches to account for uncertainty in optimization *i.e.* robust and reliability-based design optimization.

Despite robust design may serve as an interesting tool to reduce the sensitivity of a given design to randomness, it was argued that it is not the most suited to the problem we seek to solve. In fact, for the lightweight design of automotive body structures, reliability-based design optimization is the most appropriate approach since it allows us to account for the scatter in frontal impact-related constraints. RBDO problems may in general be solved by various techniques. The most widely-used ones were presented here.

In this thesis, we additionally introduce an approach which combines crude Monte Carlo sampling and quantiles of the constraints. The methodology was validated on a few test cases. However, its application to non-academic cases is seriously jeopardized by the high computational budget it requires. In general, this drawback is common to all simulation-based approaches. In fact, most of the applications in the literature rely on MPFP-based methods such as FORM. However, it has been shown how inaccurate they may be. They might even fail to converge for real industrial problems which are complex because of their dimensionality and non-linearity, *i.e.* a set of features that is met in frontal impact simulations. To by-pass the cost of simulation techniques, one method has been shown to be useful, namely surrogate modeling. Surrogate models are actually mathematical functions which can substitute the mechanical model response and are cheaper to evaluate. In this thesis, we will consider combining surrogate models and the quantile-based approach for the lightweight design of automotive body structures. The next chapter introduces the surrogate models that will be used to that end.



METAMODELING TECHNIQUES

Contents

3.1	Introd	Introduction	
3.2	An int	roduction to computer experiments	
	3.2.1	Concepts and notations	
	3.2.2	Quality of a metamodel	
	3.2.3	A brief overview of existing methods	
3.3	Suppo	ort vector machines	
	3.3.1	An introduction to the structural learning theory 60	
	3.3.2	Support vector machines for classification	
	3.3.3	Support vector machines for regression	
	3.3.4	Model training in SVR	
	3.3.5	An illustration of support vector regression	
3.4	Krigin	g or Gaussian process modeling	
	3.4.1	Introduction to Gaussian random functions	
	3.4.2	The two-stage model	
	3.4.3	Formulation of the Kriging model	
	3.4.4	Model training in Kriging	
	3.4.5	An illustration of Kriging 92	
3.5	Concl	usion	
3.1 Introduction

Historically there are two ways of thinking which are *deductive* (from the general to the particular) and *inductive* (from the particular to the general). The deductive approach is used in modern science to derive *first-principle* models to explain natural phenomena or a physical system behavior. Such models stem from idealized mathematical formulations which are validated afterwards through experimental observations. In opposition, the inductive approach makes use of already available observations to infer general assumptions about a system behavior. In the past few decades, the interest in *learning from data* has been growing because of the lack of sound deductive models. This lack is mostly due to the complexity of the systems that are to be described (*e.g.* meteorological phenomena). But most often, the growing interest in learning from data can be attributed to the availability of data in increasingly large amounts thanks to the fast development of recording tools and computational power. Identifying key features in these data and exploiting them is now an economical asset.

The fields of application are various and range for example from financial engineering (predicting future prices in a stock market from past data), to business intelligence (using big data to feed marketing strategies) and medical diagnosis. Notwithstanding this wide variety, the developed learning techniques share the same conceptual framework (Cherkassky and Mulier, 2007).

This manuscript is concerned with the field of mechanical engineering, in a slightly different setting. We consider that we have a model \mathcal{M} describing the physical behavior of a given system. The model consists of a set of partial differential equations whose solutions are numerically approximated by finite element codes. These codes have reached a high level of sophistication allowing them to be extremely accurate. However this increase of accuracy comes at the expense of time efficiency. Computationally intensive studies like optimization, reliability or sensitivity analyses are non affordable with such time-consuming models. In the framework of learning from data, they may alternatively be substituted by easy-to-evaluate mathematical functions known as *surrogate models* or *metamodels*. In the fashion of physical experiments, socalled *computer experiments* are performed to gather the data necessary to build such surrogate models.

In this chapter, we start by introducing the general concepts and notations associated to surrogate modeling. Then we introduce in two different sections the two metamodels types that will be used throughout this manuscript, namely support vector machines and Kriging.

3.2 An introduction to computer experiments

3.2.1 Concepts and notations

In the computer experiments settings, we consider that the physical system is described by a mapping $x \mapsto y = \mathcal{M}(x)$, where $x \in \mathbb{X} \subset \mathbb{R}^s$ is an *s*-dimensional *input* vector and *y* is the system *output*. *y* may either span a continuous space or a set of discrete values. In the statistical learning paradigm, the two cases respectively correspond to *regression* and *classification* tasks, often grouped into *supervised learning* techniques. In contrast, only inputs are provided in *unsupervised learning*. The goal is then to identify patterns such as clusters of points in the input space or more generally density distributions (Hastie et al., 2001). In this thesis, both

techniques will be used in various degrees but we are primarily concerned with the regression problem.

Restricting our study to the case of scalar outputs, let us consider the following set of observations:

$$\mathcal{D} = \{ (\mathbf{x}_i, y_i), i \in \{1, \dots, n\}, \mathbf{x}_i \in \mathbb{X} \subset \mathbb{R}^s, y_i \in \mathbb{Y} \subset \mathbb{R} \}.$$

$$(3.1)$$

The outputs result from a *deterministic* mapping \mathcal{M} (black-box function) with a fixed but unknown density p(x, y). We are interested in estimating the dependency between the inputs and outputs by learning over the limited set of observations \mathcal{D} , also known as the *design of experiments*. The idea is then to exploit this dependency to build up the metamodel, denoted by $\widehat{\mathcal{M}}$, which allows us to predict the response of the system at any new input point.

Formally, we search for a function f(x) in the hypothesis space of continuous functions S such that the discrepancy between the prediction and the output of the true model is minimum. Its quality is measured by a so-called *loss* function which is basically the penalty incurred for predicting f(x) instead of y. More specifically, the expectation of the loss also known as *risk* functional is minimized as follows:

$$f^* = \arg\min_{f \in \mathcal{S}} R\left(f\right) = \int_{\mathbb{X} \times \mathbb{Y}} \mathcal{L}\left(y, f\left(x\right)\right) p\left(x, y\right) dx dy,$$
(3.2)

where \mathcal{L} denotes a given loss function.

However, the probability density function p(x, y) is not known and the risk functional may only be approximated over the discrete set of observations D. This leads to the so-called *empirical* risk:

$$R_{\text{emp}}(f) = \sum_{i=1}^{n} \mathcal{L}(y_i, f(\mathbf{x}_i)).$$
(3.3)

Fitting a surrogate model by minimizing Eq. (3.3) is referred to as *empirical risk minimization* (ERM) principle. It is an ill-posed problem. In fact, if the hypothesis space S is sufficiently rich, one can find an infinite number of functions which perfectly fit the data, *i.e.* with $R_{emp}(f) = 0$. Such functions are not necessarily good at predicting over a new set of points. This pitfall is known in the related literature as *overfitting*. According to the type of metamodels, various strategies have been developed to bypass this difficulty *e.g.* regularization or complexity control as will be seen in the sequel.

3.2.2 Quality of a metamodel

Before reviewing different types of metamodels, let us introduce some metrics that will be used throughout this manuscript to assess the quality of a given surrogate model.

Derived from the quadratic loss function for regression, one of the most widely-used error measures for surrogate models is the *mean-square error* (*MSE*) and its normalized version (*NMSE*):

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (f(\boldsymbol{\chi}_{i}) - \widehat{y}_{i})^{2},$$

$$NMSE = \frac{\sum_{i=1}^{N} (f(\boldsymbol{\chi}_{i}) - \widehat{y}_{i})^{2}}{\sum_{i=1}^{N} (\overline{y} - \widehat{y}_{i})^{2}},$$
(3.4)

where $\chi_i \in X$ are input points from a validation set of size N, \hat{y}_i are the corresponding predictions by the metamodels and $\bar{y} = 1/N \sum_{i=1}^{N} f(\chi_i)$ is the mean of the true outputs.

Obviously for this error to be truly representative of the generalization ability of the metamodel, the points in this validation set should not belong to the design of experiments. Usually one creates a *training* set to fit the metamodel and a disjoint *validation* set to assess its quality.

In some cases, the cost of evaluating the true model is too high and one might not be willing to create a separate validation set beside the design of experiments. It is possible to use resampling techniques instead. That is, the design of experiments is randomly partitioned into non-overlapping training and validation sets. There are many ways to re-sample, the most exhaustive one requiring to consider all $\binom{k}{n}$ combinations for $k = \{1, ..., n\}$. This is generally known as *cross-validation* and is non-affordable, especially when *n* is high. A more practical approach, the so-called *k-fold cross validation*, consists in splitting the dataset \mathcal{D} into *K* subsets $\mathcal{D}^{(i)}$ of roughly equal sizes such that:

$$\bigcup_{i=1}^{K} \mathcal{D}^{(i)} = \mathcal{D} \quad \text{and} \quad \mathcal{D}^{(i)} \cap \mathcal{D}^{(j)} = \emptyset \quad \forall i, j = \{1, \dots, K\}, i \neq j.$$
(3.5)

With this partitioning, one can train on $\mathcal{D}^{(-i)} = \bigcup_{j \neq i} \mathcal{D}^{(j)}$ and test on $\mathcal{D}^{(i)}$ for all $i = \{1, \dots, K\}$. This random partitioning may be repeated and the error averaged over different partitions.

Eventually, the *leave-one-out* (LOO) procedure is a special case of *K*-fold cross-validation where K = n. The leave-one-out error is then defined by:

$$e_{\text{loo}} = \frac{1}{n} \sum_{i=1}^{n} \left(f^{(-i)} \left(\mathbf{x}_{i} \right) - y_{i} \right)^{2}, \qquad (3.6)$$

where $f^{(-i)}$ denotes the metamodel built by removing the pair (x_i, y_i) from the training set. Note that the LOO error is defined here with respect to the L_2 -norm. One may also consider an LOO error with respect to the L_1 -norm.

There have been empirical studies aiming at choosing the best value of *K* for model validation. Following the idea of Efron and Tibshirani (1993), Meckesheimer et al. (2002) have tried many configurations with different metamodels. They conclude that the optimal choice depends of the type of metamodel. They suggest $K \approx 0.1n$ or $K \approx \sqrt{N}$ for Kriging and K = n for polynomials and radial basis functions. Breiman and Spector (1992) suggest that for *K*-fold cross-validation, K = 5 gives a less biased error measure than K = n. In fact, there is no consensus for the best choice of *K*. Anyway, in this thesis the cross-validation procedure will be used for model calibration rather than model validation. With this respect, the leave-one-out procedure is the most appropriate choice because, for many metamodel types, there exits an approximation of the LOO error as a byproduct of the model construction. As a matter of fact, we will use an estimate of the LOO to calibrate the support vector machines in this manuscript.

3.2.3 A brief overview of existing methods

Before focusing on the two metamodels that are object of this work, let us review briefly the existing ones.

The most widely known is probably the *polynomial surface response* (Box and Draper, 1986; Box and Wilson, 1951; Kleijnen, 2007), where the approximation is viewed as a linear combination of monomials. The coefficients of this expansion are fitted based on observed data so as to

minimize the generalization error. They were first developed to fit physical experiments before an extension to numerical experiments. The applications are mostly geared toward establishing causes and effects relationships. Studies, such as *screening* which aims at identifying the more important variables of a system, are largely based on polynomial surface responses (Goupy and Creighton, 2006). However, the main drawback of such a model is that the produced function is non local. Capturing local non-linearities requires higher order polynomials which are expensive to train. Other more flexible metamodels are used for such applications.

Another widely used class of metamodels is *artificial neural networks* (ANN), first developed by the neurologists McCulloch and Pitts (1988) after their preliminary work on frogs eyes (Lettvin et al., 1959). The idea is to mimic the way a brain processes information to learn complex problems and predict new situations. That is, an ANN is a set of so-called neurons (non-linear functions) which process information from inputs to output when inter-connected in a given architecture. Training a neural network involves two tasks. The first one consists in choosing an appropriate architecture *i.e.* the number of hidden layers and nodes and the connection scheme. This may be done either thanks to the user's experience or by trial and error. Then one has to adjust the weights parameters through minimization of a risk. The pitfall to avoid here is the convergence to a local minimum or, most often overfitting. Besides, the behavior of the ANN might sometimes be unpredictable because of its complexity, leading to erroneous interpretations. Radial basis functions (RBF) networks are a particular and simpler case of ANN introduced by Broomhead and Lowe (1988). As explicitly suggested by the name, they consist of a linear combination of radially symmetric functions, mostly based on Euclidean distance $(\tilde{\phi}(x) = \phi(||x||))$. RBF networks are easier to train than ANNs. Thanks to their simplicity, they are quite popular in industrial applications.

Coming from the stochastic mechanics field, *Polynomial Chaos* expansions have been used in a probabilistic context as a metamodel (Ghanem and Spanos, 2003; Sudret, 2007). The approximation relies upon considering the output of a system as a random response which is a linear combination of *multivariate orthonormal polynomials* in the input variables. The inputs here are considered as random variables and there are families of polynomial corresponding to classical probabilities distribution functions (*e.g.* Hermite polynomials for the Gaussian PDF or Legendre polynomials for uniform PDF, etc). In the *non-intrusive* approach, which supported the fast development of the method, many techniques can be applied to identify the deterministic coefficients of the expansion (spectral projection, least-square regression or stochastic collocations methods, see Blatman (2009) for further details). Recently, a metamodel, called PC-Kriging, combining both Kriging and Polynomial chaos expansion has been proposed by Schöbi et al. (2015); Schöbi and Sudret (2014). They have shown on a few examples, that its performance is at least equal to the best of the two approaches.

In the present work, we are interested in metamodels which, beside generalization ability, will allow us to consider the noise in the outputs. This means to somehow take into account uncertainties in the data. Among the metamodels briefly introduced above, the PC expansions are the only one dealing with uncertainties. However, despite they are particularly suited to uncertainties propagation, they do not feature a tool to handle noisy outputs. Some other models, which are more suitable to the task, exist. More specifically, we will focus in this thesis on *support vector machines* and *Kriging*. The former is particularly adapted to noisy observations thanks to its *insensitive zone* as we will explain shortly. The latter, with its *nugget effect* may also account for noisy outputs. Moreover, Kriging has a probabilistic framework which provides a local error estimator very useful in our applications. This will be the object of the next chapter.

3.3 Support vector machines

Support vector machines have been developed by Vapnik and its associate at AT&T Bells laboratories in the late 1980s. Based on a sound theoretical framework, they were first developed for classification problems. Later, a regression extension was proposed by the same authors. Support vector machines have been used in many engineering applications. For instance, Osuna et al. (1997) used support vector classification (SVC) for face detection. For the automotive industry, Guo et al. (2012) proposed an algorithm for pedestrian detection based on SVC. As for regression, applications on structural optimization under crashworthiness exist in the literature (Pan et al., 2010; Zhu et al., 2012).

3.3.1 An introduction to the structural learning theory

3.3.1.1 From empirical to structural risk minimization

In the previous section, we explained that the empirical risk may be the practical substitute for the unknown risk functional in model selection. For this to be possible, other conditions have to be fulfilled (Vapnik, 1998). A crucial one is *asymptotic consistency*, meaning that the empirical risk should converge toward the true and possibly minimal risk function when the sample size grows. Besides, the rate of convergence of the empirical risk to the true one should be fast and distribution-independent. These are quite abstract conditions and cannot be implemented for learning problems. For practical purpose, the conditions must be explicitly expressed in terms of properties of the set of approximating functions. This is done by introducing bounds on the true risk which are functions of the empirical risk. To derive such bounds, Vapnik and Chervonenkis (1971) introduced an index describing the complexity of a family of functions named *VC dimension*.

In the sequel, we will define the VC dimension and the resulting bounds for classification problems. Similar concepts and results exist for regression but to the author's opinion, they are conceptually easier to understand with classification.

In a nutshell, the VC dimension of a set of functions reflects its capacity to separate data into two sets. More formally speaking, the VC dimension of a set of functions is the maximum number of vectors *h* that can be separated in all 2^h configurations by this set of functions. The function is then said to *shatter* the data. As an example, let us consider the family of linear indicators in a two-dimensional space with labels $y \in \{-1,1\}$. Up to three points can be shattered by linear functions in all $2^3 = 8$ possibilities as shown in Figure 3.1. This cannot be said of four points. For this example the VC dimension is hence equal to three. In general, it has been shown that the VC dimension of hyperplanes in a *s*-dimensional space is s + 1.

This notion of VC dimension is used to derive bounds of the difference between the unknown true risk and the empirical risk. The finiteness of the VC dimension for a set of functions actually ensures its ability for generalization. As a matter of fact, Vapnik (1995) has shown that with probability $1 - \eta$ ($\eta << 1$), the risk for the function f^* which minimizes the empirical risk satisfies the inequality:

$$R(f^*) < R_{\text{emp}}(f^*) + \frac{\epsilon(n)}{2} \left(1 + \sqrt{\frac{1 + 4R_{\text{emp}}(f^*)}{\epsilon(n)}} \right),$$
(3.7)

where *n* is the number of training points, *h* the VC dimension, and $\epsilon(n) = \frac{4h \ln(\frac{2n}{h} + 1) - \ln(\frac{\eta}{4})}{n}$.



Figure 3.1: VC dimension: How three points can be shattered in all eight configurations by a set of lines in a two-dimensional space.

The second part of the right-hand side of Eq. (3.7) is called the *confidence interval*. With all other parameters fixed, it monotonically decreases as the ratio n/h increases. That is, for relatively large sample sizes, the empirical risk converges to the true risk and one can apply with enough confidence the empirical risk minimization principle. However, when the number of training points is small with respect to the VC dimension of the set of learning functions (*i.e.* n/h < 20), the confidence interval cannot be ignored, otherwise overfitting may occur. The structural risk minimization principle, developed by Vapnik, is in this respect a good alternative.

3.3.1.2 The structural risk minimization principle

In contrast to the ERM principle where the learning problem is addressed by minimizing the empirical risk, the *structural risk minimization* (SRM) principle rather seeks to minimize an upper bound of the true risk, more specifically the right-hand side of Eq. (3.7). The aim is to find a trade-off between the empirical risk minimization and the complexity of the set of functions used for learning. Recall that if the hypothesis space is rich, that is, the set of functions has a high degree of complexity or capacity, the minimizer of the empirical risk is not necessarily the one of the true risk. On the other hand, if the hypothesis space is too restricted, one might not find a function which truly approximates the system's behavior. This is known as *underfitting* as opposed to overfitting. The VC dimension has been developed to essentially capture this notion of complexity. The idea of the SRM principle is to balance the empirical risk minimization and the complexity of the set of functions. Its scheme is described as follows.

Let us introduce a structure of nested hypothesis spaces S_{h_i} with increasing VC dimension h_i , *i.e.* $S_{h_1} \subset S_{h_2} \subset ... \subset S_{h_k}$. The strategy provided by the SRM principle is to a) select in each hypothesis space the function that minimizes the empirical risk, b) find the confidence interval of the true risk for the selected functions c) estimate the bounds of the true risk and d) select the function from the structure with the lowest bound on the true risk. The idea is illustrated in Figure 3.2.



Figure 3.2: Illustration of the SRM principle: the best model is chosen as a trade-off between the empirical risk and the optimal complexity of nested structures (Vapnik, 1998).

As an example, consider the family of polynomials of degree p. A nested structure as described above can be constructed with such a family. Restricting the candidates to polynomial of low orders, say $p = \{0, 1\}$, one might not find any polynom that approximates accurately the actual model. On the contrary, if p is chosen too high, the resulting approximation may be too complex and despite explaining perfectly the data, it will not be good at generalization. The SRM principle suggests to find an optimal trade-off between the two aspects.

This principle has been a solid foundation for the implementation of new types of learning machines, among which the *support vector machines*.

3.3.2 Support vector machines for classification

Support vector machines implement the structural risk minimization principle in a slightly different scheme (Cherkassky and Mulier, 2007). The search is restricted to hyperplanes (not necessarily in the input space as will be seen later). Within families of hyperplanes with equivalent empirical risks, the optimal one is chosen as the one having the smallest VC dimension. This is done by introducing so-called *margin-based* loss functions.

3.3.2.1 Classification of linearly separable data

In the context of classification, we are given a set of experiments \mathcal{D} where the outputs can take two values or labels, say $y_i \in \{-1, 1\}$ (for the less general case of *binary classification*). From this finite set of data, we would like to define an indicator function or classifier which will allow us to determine the class of a new input vector.

We first consider the case of linearly separable data (this assumption will be relaxed later). We seek for the hyperplane that perfectly separates the data into the two classes with labels -1 and 1, respectively. There are infinitely many such hyperplanes. However, there is only one for which the distance between the closest points to each class is maximal. This separator is called

the *optimal separating hyperplane* and the distance between the two classes is the *margin*. This idea is illustrated in Figure 3.3 for a two-dimensional case.



Figure 3.3: A simple linear classification example. Left: The classes of data are separated by various lines. Right: The unique and optimal separating hyperplane is shown, together with its margin.

Let us write the separating hyperplane as follows:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b, \tag{3.8}$$

where the coefficient w and the bias parameter b are to be set and $\langle \bullet, \bullet \rangle$ denotes the inner product in X, *i.e.* the classical scalar product in Euclidean spaces.

For a separating hyperplane with a margin of width 2ρ , the following classification constraints must be satisfied:

or in a more compact form,

$$y_i\left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b\right) \ge \rho. \tag{3.10}$$

On the other hand, the distance of a point x_i to an hyperplane defined by Eq. (3.8) reads:

$$d(\mathbf{x}_i, f) = \frac{|\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b|}{\|\boldsymbol{w}\|}.$$
(3.11)

From Eq. (3.11), it turns out that maximizing the margin is equivalent to minimizing ||w||. By rescaling the coefficients w and b so that Eq. (3.10) is represented in a canonical form, the SVM

problem has the following form:

$$\min_{w} \quad \frac{1}{2} \|w\|^2,$$
subject to $y_i \left(\langle w, x_i \rangle + b \right) - 1 \ge 0, \quad i = \{1, \dots, n\}.$

$$(3.12)$$

The solution of Eq. (3.12) with respect to the unknown parameters w and b allows one to define the optimal separating hyperplane as a function of the training data only. As intended by the support vector machines, the complexity of the separator is controlled by the margin. That is, bigger margins reduce the flexibility of the separator and thus the possibility of correctly shattering the points. For a rather formal insight, Vapnik (1995) has shown that the VC dimension of a set of hyperplanes with a margin ρ is bounded:

$$h \le \min(R^2/\rho, s) + 1,$$
 (3.13)

where R is the radius of the smallest hypersphere enclosing all the training points. This relationship shows that the margin is inversely proportional to the VC dimension. Once again maximizing the margin leads to decreasing the VC dimension and therefore the confidence interval on the true risk.

Another important feature here is that the margin is independent of the dimensionality of the problem. One can thus expect good generalization ability in high dimension (Bennett and Campbell, 2000). However, solving Eq. (3.12) for high dimensional problems is hardly tractable. Since it is a quadratic convex optimization problem with linear constraints, one can rather solve its *dual formulation* by introducing *Lagrange multipliers*.

The Lagrangian therefore reads:

$$L(\boldsymbol{w}, b; \boldsymbol{\alpha}) = \frac{1}{2} \|\boldsymbol{w}\|^2 - \sum_{i=1}^n \alpha_i \left(y_i \left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \right) - 1 \right),$$
(3.14)

where $\alpha = \{\alpha_i \ge 0, i = \{1, ..., n\}\}$ are the Lagrange multipliers.

The solution of the Lagrangian is given by its saddle point, *i.e.* the one that minimizes L with respect to w and b and maximizes it with respect to α . Applying the optimality conditions of Karush-Kuhn-Tucker translates into:

$$\begin{cases} \frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0, \\ \frac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0. \end{cases}$$
(3.15)

To get the saddle point, the Lagrangian may be expressed in terms of α only by introducing Eq. (3.15) in Eq. (3.14):

$$L(\boldsymbol{w},b;\boldsymbol{\alpha}) = \frac{1}{2} \langle \boldsymbol{w}, \boldsymbol{w} \rangle - \sum_{i=1}^{n} \alpha_{i} y_{i} \boldsymbol{x}_{i} \boldsymbol{w} - b \sum_{i=1}^{n} \alpha_{i} y_{i} + \sum_{i=1}^{n} \alpha_{i},$$

which ultimately leads to:

$$L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \sum_{i=1}^{n} \alpha_i.$$
(3.16)

Eventually, the formulation of the dual optimization problem reads:

$$\max_{\alpha} \quad -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{i} \alpha_{j} y_{i} y_{j} \langle \mathbf{x}_{i}, \mathbf{x}_{j} \rangle + \sum_{i=1}^{n} \alpha_{i},$$

subject to
$$\sum_{i=1}^{n} \alpha_{i} y_{i} = 0, \quad \alpha_{i} \ge 0, \quad i = \{1, \dots, n\}.$$
 (3.17)

Solving this equation gives the coefficients α of the SVM expansion. Besides, the Karush-Kuhn-Tucker conditions state that any solution corresponding to non-zero Lagrange multipliers satisfies the inequality of the primal problem with equality. That is,

$$y_s\left(\langle \boldsymbol{w}, \boldsymbol{x}_s \rangle + b - 1\right) = 0, \quad \forall \left\{s = 1, \dots, n : \alpha_s \neq 0\right\}.$$
(3.18)

The value of *b* is henceforth given by:

$$b = y_s - \sum_{i=1}^n \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x}_s \rangle, \qquad (3.19)$$

for any pair (x_s, y_s) .

The optimal separating hyperplane may finally be recast as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b.$$
(3.20)

An important feature of support vector machines is carried by the solution of the Lagrangian problem. In fact, the data points (called "vectors" here) whose Lagrange multipliers are nonzeros lie on the margin as expressed in Eq. (3.18). Beside, they are the only vectors which are actually meaningful in expressing Eq. (3.20). They are henceforth called *support vectors*. In Figure 3.4, the support vectors corresponding to the example introduced in Figure 3.3 are circled in black. Additionally, we have stressed another geometrical interpretation of the dual problem. That is, the optimal separating hyperplane is the one that bisects the minimal distance between the convex hulls of the two classes of data (Cherkassky and Mulier, 2007).

3.3.2.2 Classification of non-linearly separable data

Hard margin versus soft margin

In the section above, we have explained how to build the SVC model. We were in the simplest case of linearly separable data. Such a model is referred to as *hard margin classifier*. However, it may exist situations where this assumption no longer holds. It is henceforth necessary to resort to the so-called *soft margin classifier*: one should consider points lying inside the margin or in the wrong class. This is achieved by introducing the so-called *slack variables* and a penalty function. The slack variables simply measure the deviation of a sample point from its class (margin included). The optimization problem in Eq. (3.12) then becomes:

$$\min_{\boldsymbol{w}} \quad \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^n \xi_i,$$

subject to: $y_i \left(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b \right) \ge 1 - \xi_i, \quad i = \{1, \dots, n\},$ (3.21)



Figure 3.4: Geometrical interpretation of the dual problem: The optimal separating hyperplane bisects the shortest line between the two convex hulls of the data. The support vectors are circled in black.

where ξ_i is the slack variable associated to the data pair (x_i, y_i) and *C* a penalty term trading the margin maximization with the accepted misclassification error. Cortes and Vapnik (1995) shows that the solution to this problem also exists and is unique for any training set. In fact, the dual problem is similar to the one in Eq. (3.17) except that here the coefficients α_i are upper-bounded by *C*.

The feature space

So far we were interested in linear separators. Support vector machines also allow one to specify non-linear boundaries. This is achieved by mapping the training points into a high dimensional *feature space* where the data could be separated linearly, following the idea by Aizerman et al. (1964). Figure 3.5 shows a naive yet representative illustration of the idea behind the mapping into a higher dimensional space. In this example, we consider data separable by a circle in a two-dimensional space. The mapping $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$, $(x_1, x_2) \mapsto (x_1, x_2, \sqrt{x_1^2 + x_2^2})$ allows the data to be separated in the resulting three-dimensional space by a plane.

A drawback from a higher dimensional space is that the construction of the optimal separating hyperplane would be much more complicated. The so-called *kernel trick* is used to bypass this difficulty. Recall that the expansion of the SVM relies on the inner product in the input space. Consequently, if x_i is replaced by its image in the feature space $\Phi(x_i)$, Eq. (3.20) shows that the only information needed to construct the model is the inner product in the feature space $\langle \Phi(x_i), \Phi(x) \rangle$. It turns out that this computation may also be equivalently carried out by a so-called *kernel* function. Thus, there is no need to carry out any direct operation in the feature space or even to know an explicit form of the mapping.

Let us denote the kernel function as follows:

$$k(\mathbf{x},\mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle. \tag{3.22}$$



Figure 3.5: Feature space: The data are separated by a circle in the input space and by the gray hyperplane in the feature space

For a given kernel, the so-called *Mercer's conditions* need to be fulfilled for the underlying mapping to exist (Cherkassky and Mulier, 2007). In fact, k(x, x') is any function such that:

$$\iint k(\mathbf{x}, \mathbf{x}') \varphi(\mathbf{x}) \varphi(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \ge 0, \quad \forall \varphi \neq 0, \quad \int \varphi^2(\mathbf{x}) d\mathbf{x} < \infty.$$
(3.23)

Indeed any function satisfying Mercer's conditions corresponds to an inner product in some feature space. As it happens, many widely used functions fall into this category. Examples are:

- Polynomials: $(\langle x, x' \rangle + d)^p$, where $d \in \mathbb{R}$ and $p \in \mathbb{N}^*$;
- Radial basis functions: $\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2l^2}\right)$ where l > 0 is a characteristic length scale;
- Neural networks: $\tanh(\nu \| \mathbf{x} \mathbf{x}' \| + a)$ where $\nu > 0$ and $a \in \mathbb{R}$, etc.

With all the above mentioned topics, the general formulation of the SVM classifier for a new point *x* reads:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b.$$
(3.24)

The sign of f(x) gives the predicted class for x. Besides, if $-1 \le f(x) \le 1$, then x is inside the margin.

3.3.3 Support vector machines for regression

For real-valued outputs, support vector regression implements the same concept beneath SVM for classification. Repeating the same arguments, Vapnik came up with the so-called *ɛ-insensitive*

loss function. Tuning the parameters of this loss function is actually equivalent to maximizing the margin. In this section, we will go straight to the formulation of the soft-margin regression problem since the derivations are exactly the same as for classification.

3.3.3.1 Formulation of the SVR problem

In the SVR problem, a regression is performed on the data based on some penalization. We first start by describing the problem in the simplified case of linear regression. Generalization to non-linear cases will be dealt with later. The concept of ε -insensitive loss is illustrated in Figure 3.6 and states the following: As long as the difference between the actual and predicted values for a given point is less than ε , the loss function is zero. Beyond this limit, the point is penalized linearly. This is formulated as follows (Vapnik et al., 1997; Smola and Schölkopf, 2004):

$$\mathcal{L}_{\varepsilon} = \begin{cases} 0 & \text{if } |f(x) - y| < \varepsilon, \\ |f(x) - y| - \varepsilon & \text{otherwise.} \end{cases}$$
(3.25)



Figure 3.6: Concept of support vector regression: Only the vectors outside the ε -insensitive tube (gray-shaded area) are penalized (Smola and Schölkopf, 2004).

By similarity with classification, the SVR problem is formulated as follows:

$$\min_{w} \quad \frac{1}{2} \|w\|^{2} + C \sum_{i=1}^{n} \left(\xi_{i} + \xi_{i}^{*}\right),$$
subject to
$$y_{i} - \langle w, x_{i} \rangle - b \leq \varepsilon + \xi_{i},$$

$$\langle w, x_{i} \rangle + b - y_{i} \leq \varepsilon + \xi_{i}^{*},$$

$$\xi_{i}, \xi_{i}^{*} \geq 0.$$
(3.26)

Adopting the margin-based structures as for classification, the capacity or complexity is equivalently controlled by the insensitive tube width ε . From another perspective, the objective function of Eq. (3.26) may be regarded as a regularized risk functional where C is the regularization constant (Evgeniou et al., 2000; Smola and Schölkopf, 2004).

Once again, the solution of this problem is carried out in its dual form by introducing the Lagrangian:

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \frac{1}{2} \|\boldsymbol{w}\|^{2} + C \sum_{i=1}^{n} (\xi_{i} + \xi_{i}^{*}) - \sum_{i=1}^{n} (\eta_{i}\xi_{i} + \eta_{i}^{*}\xi_{i}^{*}) - \sum_{i=1}^{n} \alpha_{i} (\varepsilon + \xi_{i} - y_{i} + \langle \boldsymbol{w}, \boldsymbol{x}_{i} \rangle + b) - \sum_{i=1}^{n} \alpha_{i}^{*} (\varepsilon + \xi_{i}^{*} + y_{i} - \langle \boldsymbol{w}, \boldsymbol{x}_{i} \rangle - b),$$
(3.27)

where $\alpha_i \ge 0$, $\alpha_i^* \ge 0$, $\eta_i \ge 0$ and $\eta_i^* \ge 0$ are Lagrange multipliers.

The KKT conditions lead to the following relations:

$$\begin{cases} \frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_{i} x_{i} + \sum_{i=1}^{n} \alpha_{i}^{*} x_{i} = 0 \quad \Rightarrow w = \sum_{i=1}^{n} (\alpha_{i} - \alpha_{i}^{*}) x_{i}, \\ \frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_{i} + \sum_{i=1}^{n} \alpha_{i}^{*} = 0 \qquad \Rightarrow \sum_{i=1}^{n} (\alpha_{i} - \alpha_{i}^{*}) = 0, \\ \frac{\partial L}{\partial \xi_{i}} = C - \alpha_{i} - \eta_{i} = 0 \qquad \Rightarrow \eta_{i} = C - \alpha_{i}, \\ \frac{\partial L}{\partial \xi_{i}^{*}} = C - \alpha_{i}^{*} - \eta_{i}^{*} = 0 \qquad \Rightarrow \eta_{i}^{*} = C - \alpha_{i}^{*}, \end{cases}$$
(3.28)

By appropriately grouping the terms in Eq. (3.27) such that:

$$L(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \frac{1}{2} \langle \boldsymbol{w}, \boldsymbol{w} \rangle - \sum_{i=1}^{n} (\alpha_{i} - \alpha_{i}^{*}) \langle \boldsymbol{w}, \boldsymbol{x}_{i} \rangle + \sum_{i=1}^{n} (C - \alpha_{i} - \eta_{i}) \boldsymbol{\xi}_{i}$$
$$+ \sum_{i=1}^{n} (C - \alpha_{i}^{*} - \eta_{i}^{*}) \boldsymbol{\xi}_{i}^{*} - \sum_{i=1}^{n} (\alpha_{i} + \alpha_{i}^{*}) \boldsymbol{\varepsilon} + \sum_{i=1}^{n} (\alpha_{i} - \alpha_{i}^{*}) \boldsymbol{y}_{i} - \boldsymbol{b} \sum_{i=1}^{n} (\alpha_{i} - \alpha_{i}^{*}),$$

and introducing the relations of Eq. (3.28), the Lagrangian can be recast as a function of α and α^* only:

$$L(\boldsymbol{\alpha},\boldsymbol{\alpha}^*) = -\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \left(\alpha_i - \alpha_i^*\right) \left(\alpha_j - \alpha_j^*\right) \left\langle \boldsymbol{x}_i, \boldsymbol{x}_j \right\rangle - \sum_{i=1}^n \left(\alpha_i + \alpha_i^*\right)\varepsilon + \sum_{i=1}^n \left(\alpha_i - \alpha_i^*\right)y_i.$$
(3.29)

The solution to this optimization problem is given by the saddle point of the Lagrangian:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} L\left(\boldsymbol{\alpha},\boldsymbol{\alpha}^*\right), \quad \text{subject to:} \quad \sum_{i=1}^n \left(\alpha_i - \alpha_i^*\right) = 0 \quad \text{and} \quad 0 \le \alpha_i, \alpha_i^* \le C. \tag{3.30}$$

After Eq. (3.30) has been solved, the following expression gives the prediction for a new point:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \langle \mathbf{x}_i, \mathbf{x} \rangle + b.$$
(3.31)

The expression of *b* will be given in the sequel.

As a final step, we introduce the kernel function as the inner product in the high-dimensional feature space to allow for non-linear regression. The prediction then reads:

$$f(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x}) + b = \sum_{i=1}^n \left(\alpha_i - \alpha_i^* \right) k(\mathbf{x}_i, \mathbf{x}) + b.$$
(3.32)

Some comments have to be made on the different configurations resulting from the solution of Eq. (3.30) (Smola and Schölkopf, 2004). First, α_i and α_i^* cannot be simultaneously nonzero as there is no point x_i which could lie on both sides of the ε -tube. Second, the solution depends only on points for which $\alpha_i - \alpha_i^* \neq 0$. Here again such points are called *support vectors*. Finally, the two last relationships stemming from the KKT conditions in Eq. (3.28) imply that $0 \le \alpha_i^{(*)} \le C$ since $\eta_i^{(*)} \ge 0$, where $\bullet_i^{(*)}$ refers to both \bullet_i and \bullet_i^* . The support vectors for which $\alpha_i^{(*)} = C$ are called *bounded support vectors* and those for which $0 < \alpha_i^{(*)} < C$ are called *unbounded support vectors* and the points lying strictly outside the ε -insensitive tube (defined by the gray-shaded area in Figure 3.6) and the latter to those lying at the boundary of the ε -insensitive tube.

The unbounded SVs are useful when it comes to computing the bias (Gunn, 1998). In fact for those SVs, the slack variables vanish and the inequalities in the constraints of Eq. (3.26) become equalities, leading to the following expression of *b*:

$$b = \operatorname{sign}\left(\alpha_{\rm usv} - \alpha_{\rm usv}^*\right)\varepsilon + y_{\rm usv} - \sum_{i=1}^{n} \left(\alpha_i - \alpha_i^*\right)k\left(x_i, x_{\rm usv}\right),$$
(3.33)

for any couple (x_{usv} , y_{usv}) corresponding to an unbounded support vector with coefficients $\alpha_{usv}^{(*)}$. For a more robust value of b, this expression may be averaged over the entire set of unbounded support vectors.

3.3.3.2 Implementation of the regression problem

The problem in Eq. (3.30) is a linearly constrained quadratic optimization problem. Let us recast it so that it fits the nominal form of a classical quadratic programming (QP) problem:

$$\min_{\alpha_i,\alpha_i^*} \quad \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \left(\alpha_i - \alpha_i^* \right) \left(\alpha_j - \alpha_j^* \right) k\left(\mathbf{x}_i, \mathbf{x}_j \right) + \sum_{i=1}^n \left(\varepsilon - y_i \right) \alpha_i + \sum_{i=1}^n \left(\varepsilon + y_i \right) \alpha_i^*$$
subject to:
$$\sum_{i=1}^n \left(\alpha_i - \alpha_i^* \right) = 0,$$
(3.34)

which in matricial form reads:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^{*}} \quad \frac{1}{2} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^{*} \end{bmatrix}^{T} \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{K} \\ -\boldsymbol{K} & \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^{*} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon} - \boldsymbol{y} \\ \boldsymbol{\varepsilon} + \boldsymbol{y} \end{bmatrix}^{T} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^{*} \end{bmatrix}$$
subject to:
$$\begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix}^{T} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^{*} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \qquad 0 \le \boldsymbol{\alpha}, \boldsymbol{\alpha}^{*} \le C,$$
(3.35)

where **K** is an $n \times n$ matrix, known as the *Gram* matrix, whose components are $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = \{1, ..., n\}$.

As it is a quadratic convex optimization problem, the solution always exists. It is unique and global. Besides, in this generic formulation many available algorithms are readily usable to solve it. Smola and Schölkopf (2004) briefly review some known optimization packages and algorithms. The major issue in the related literature is the size of the training set as the Gram matrix *K* becomes too large to be computed and stored when *n* is large. This aspect is actually rooted in the nature of SVM for classification where the problems often correspond to extremely

large sets of data. To overcome this difficulty, decomposition methods were introduced (Chang and Lin, 2011; Osuna et al., 1997) and have been also used for regression problems. These methods mainly rely on the sparsity property of SVM (among the training data, only the SVs are needed to construct the model). For the SVs are not known beforehand, one could rather select a subset of points as the working set and solve optimization sub-problems. Then proceed iteratively by adding or replacing new points in the working set so as to converge toward a solution where the KKT conditions are met. An efficient algorithm which falls in this category is the sequential minimal optimization (SMO) developed by Platt (1999). Here, the optimization problem is reduced to only two variables at each step and the solution of each sub-problem is analytically computed.

In this thesis, we are interested in building SVR as surrogate models for crashworthiness design. In this respect, the size of the training set is relatively small (since the computation of a single training point is quite expensive). The sampling size is thus not a concern for us (up to a certain point where the concerns shifts to the time required for model training). We will thus favor more straightforward approaches. We therefore use an interior point algorithm (Vanderbei, 1994) as Bompard (2011) did. Another asset of using such an algorithm is that the slack variables $\xi_i^{(*)}$ and the bias *b* are by-products of the algorithm.

3.3.3.3 Other formulations of support vector regression

In the previous section, we presented the traditional formulation of SVR with the ε -insensitive loss function. The penalization was linear with respect to the slack variables. Other SVR formulations derived from different loss functions also exist. Two of them are now presented.

L_2 -SVR with the ε -insensitive loss function

 L_2 -SVR implements the quadratic penalization of the ε -insensitive loss function. Accordingly, the primal problem reads:

$$\min_{w} \quad \frac{1}{2} \|w\|^{2} + C \sum_{i=1}^{n} \left(\xi_{i}^{2} + \xi_{i}^{*2}\right),$$
subject to
$$y_{i} - \langle w, x_{i} \rangle - b \leq \varepsilon + \xi_{i},$$

$$\langle w, x_{i} \rangle + b - y_{i} \leq \varepsilon + \xi_{i}^{*},$$

$$\xi_{i}, \xi_{i}^{*} \geq 0.$$
(3.36)

Following the same developments as for L_1 -SVR, the dual formulation of the problem is then cast as:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^{*}} \quad \frac{1}{2} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{*}\right)^{T} \widetilde{K} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{*}\right) + \sum_{i=1}^{n} \left(\varepsilon - y_{i}\right) \alpha_{i} + \sum_{i=1}^{n} \left(\varepsilon + y_{i}\right) \alpha_{i}^{*},$$
subject to:
$$\sum_{i=1}^{n} \left(\alpha_{i} - \alpha_{i}^{*}\right) = 0, \quad \alpha_{i}, \alpha_{i}^{*} \geq 0, \quad i = \{1, \dots, n\},$$
(3.37)

where $\widetilde{K} = K + (1/C) I$ and I the identity matrix of size $n \times n$.

The L_2 -SVR problem differs from L_1 -SVR in two aspects related to the difference in the derivatives of the Lagrangians with respect to the slack variables. First, the diagonal of Gram matrix is modified by adding the inverse of the penalty term *C*. For high values of *C*, this operation actually serves as a regularization of the Gram matrix making the QP problem easier to solve. Second, the Lagrange multipliers α and α^* are no longer upper-bounded by *C*.

Quadratic loss function

The quadratic loss function simply writes:

$$\mathcal{L}_{\text{quad}} = (f(\mathbf{x}) - y)^2. \tag{3.38}$$

Following Gunn (1998), the associated dual optimization problems reads:

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2} \boldsymbol{\alpha}^{T} \boldsymbol{K} \boldsymbol{\alpha} - \sum_{i=1}^{n} y_{i} \alpha_{i} + \frac{1}{2C} \sum_{i=1}^{n} \alpha_{i}^{2},$$

subject to:
$$\sum_{i=1}^{n} \alpha_{i} = 0, \quad \alpha_{i} \ge 0, \quad i = \{1, \dots, n\}.$$
 (3.39)

The interest of this formulation is that the optimization problem reduces to n unknown (instead of 2n for the ε -insensitive loss function). In contrast there is no insensitive zone. All vectors are therefore support vectors. The regression thus loses in robustness and sparsity, which might be problematic in cases when the training data are noisy and of large size.

To be nearly exhaustive, we should also mention *least square*-support vector machines introduced by Suykens and Vandewalle (1999) for classification and Saunders et al. (1998) for regression. The idea is here to replace the inequalities constraints of the primal formulation by target equalities. This results in a linear programming problem which is easier to solve. Some applications of these ideas are given in de Brabanter (2011).

3.3.4 Model training in SVR

3.3.4.1 Overview of training techniques

It is widely accepted that the performance of a SVM largely relies on a good choice of the model hyperparameters. For regression, these hyperparameters are:

- The regularization or penalty term *C*: It represents the trade-off between the complexity of the model and the amount up to which errors are tolerated. When *C* is small, the model is smooth with a wide acceptance of errors whereas a too large *C* results in focusing on the minimization of the empirical risk;
- The insensitivity parameter ε : It represents the amount up to which deviations from the target value are tolerated. It scales with the number of support vectors (the wider the less support vectors) and can be seen in this sense as a parameter affecting the complexity of the model;
- Kernel parameter: The choice of an appropriate kernel depends on the practitioner experience and the type of problem considered. In this thesis, we will essentially consider radial basis kernels as they have proven to be efficient. In this case, the parameter to calibrate is the *characteristic length scale l* which somehow represents the radius of influence of one vector to its neighborhood.

There are a number of heuristics suggested by various authors that may be used to wisely select the parameters a priori given a set of observations. We will refer to them as *manual selection*. On the other hand, *automatic selection* may be used to thoroughly search for the optimal values of the parameters in their entire definition space.

Manual selection

For the selection of the penalty term, Mattera and Haykin (1999) suggests, under the assumption that ε and l are somehow known, that a good choice of C can be the range of the outputs. This value is however sensitive to outliers in the data and may hence not truly represent the range of the outputs in the true model. Cherkassky and Ma (2004) came up with the following more robust representation:

$$C = \max\left(\left|\bar{y} + 3s_y^*\right|, \left|\bar{y} - 3s_y^*\right|\right), \qquad (3.40)$$

where \bar{y} and s_y^* are respectively the mean and standard deviation of the training data outputs.

As for the insensitive tube width selection, Kwok (2001) first considers that the outputs are actually a sum of a mean value and an independent noise ζ . He then shows that there is a linear dependency between ε and the level of noise in the output. He consequently sets a relationship between ε and the noise variance σ_{ζ}^2 for different distributions, namely Gaussian,

Laplacian and uniform. For instance, with a Gaussian noise defined by $\zeta \sim \mathcal{N}(0, \sigma_{\zeta}^2)$, the optimal value of ε said to be $1.0043\sigma_{\zeta}$. However the distribution of noise in usual applications is not known. Taking a fully different perspective, Smola and Schölkopf (2004) argue that there is anyway no need to use the ε -insensitive loss function if the distribution of the noise is known: the maximum likelihood loss function is likely to be more efficient.

Automatic selection

For an automatic selection, we look for the best model w.r.t. some generalization error. For reasons explained in section 3.2.2, the search is usually based on the leave-one-out error. That is, we search for the model parameters which minimize the LOO error. To automatically tune the parameters in this perspective, some techniques may be used:

- A simple yet practical method is the grid search. In grid searching, the parameter space is bounded and discretized. The model is then built with all possible combinations from the resulting grid. The one with the lowest error is selected as the final model. It leads to rather good results but suffers of two main drawbacks. First, the user should know a priori some bounds of the parameters to avoid costly search in useless regions. Second, the solution depends too much on the fineness of the discretization: the finer the grid, the more accurate the solution. However, too fine a grid is time-consuming.
- Another approach is to use global optimization algorithms to find the best model. As shown in Bompard (2011), the testing error is multimodal in the parameters space. Local optimization algorithms are therefore inefficient and one may only use global ones. Chen and Wang (2007) implemented SVR for tourism demand forecasting using a genetic algorithm for parameters selection. Momma and Bennett (2002) implemented a method based on the pattern search algorithm to tune the parameters for their application in drug design. In this manuscript we rely on the cross-entropy method, a random search technique well fitted to solve multi-extremal optimization problems (Hu and Hu, 2009; Kroese et al., 2006; Boer et al., 2004).

The techniques as presented above rely on the actual computation of the LOO error. This means that for a given set of parameters, the number of SVR trainings is equal to the number of training points. Considered in an iterative scheme, the computation of the LOO becomes too expensive. In the past few years, many researchers have developed bounds on the LOO error which are cheaper to compute. We will focus on this approach in the next section.

3.3.4.2 Approximation of the leave-one-out error

As mentioned earlier, the development of support vector machines is based on sound theoretical analysis of the generalization error. This includes several works on defining approximations and bounds on the generalization errors. More specifically the leave-one-out error, which is well suited to problems where the training set size is relatively small, gives an almost unbiased estimation of the generalization error. In classification, such bounds include for example a very loose one based on the count of support vectors (Vapnik, 1995) or the *radius-margin* bound based on the radius of the smallest hypersphere enclosing all the training vectors in the feature space. More recently, Vapnik and Chapelle (2000) have developed a tighter bound based on the notion of *span of support vectors*. Chang and Lin (2005) have extended these bounds for regression problems. In this manuscript, we will adopt the latter span bound estimate for both regression and classification.

Let us first introduce the following set of points in the feature space, for any $p = \{1, ..., n\}$:

$$\Lambda_{p} = \left\{ \sum_{\mathcal{I}_{usv}^{-p}} \lambda_{i} \Phi\left(\mathbf{x}_{i}\right) : \lambda_{i} \in \mathbb{R} \quad \forall i \in \mathcal{I}_{usv}^{-p} \quad \sum_{i \in \mathcal{I}_{usv}^{-p}} \lambda_{i} = 1 \right\},$$
(3.41)

where $\mathcal{I}_{usv} = \{i \in \{1, ..., n\} : 0 < \alpha_i + \alpha_i^* < C\}$ denotes the set of unbounded support vectors and $\mathcal{I}_{usv}^{-p} = \mathcal{I}_{usv} \setminus \{p\} \quad \forall p \in \{1, ..., n\}.$

The span of the support vector x_p is defined as its distance to Λ_p and reads:

$$S_p^2 = d^2 \left(\Phi \left(\boldsymbol{x}_p \right), \Lambda_p \right) = \min_{\lambda_i \in \mathbb{R}, i \in \mathcal{I}_{usv}^{-p}} \| \Phi \left(\boldsymbol{x}_p \right) - \sum_{i \in \mathcal{I}_{usv}^{-p}} \lambda_i \Phi \left(\boldsymbol{x}_i \right) \|^2, \quad \text{s.t.:} \quad \sum_{i \in \mathcal{I}_{usv}^{-p}} \lambda_i = 1.$$
(3.42)

Based on this span, Chang and Lin (2005) have derived approximations of the leave-one-out error. That is, under the assumption that the set of support vectors remains the same during the leave-one-out procedure, the following equation gives an accurate approximation of the LOO error for L_1 -SVR:

$$\widehat{e}_{\text{loo}} = \frac{1}{n} \left(\sum_{p=1}^{n} \left(\alpha_p + \alpha_p^* \right) S_p^2 + \sum_{p=1}^{n} \left(\xi_p + \xi_p^* \right) \right) + \varepsilon, \qquad (3.43)$$

where $\alpha^{(*)}$ and $\xi^{(*)}$ are solutions of Eq. (3.35).

With Eq. (3.43), one is able to estimate the leave-one-out error without proceeding to its actual computation. One would however need to solve the optimization problem in Eq. (3.42) for any $p = \{1, ..., n\}$. Hopefully, this task can be further simplified (Chapelle, 2002). For doing so, let us first re-order the support vectors so that the unbounded ones are indexed by $\{1, ..., n_{usv}\}$ and introduce a Lagrange multiplier μ to enforce the constraint $\sum_{i \in \mathcal{I}_{usv}^{-p}} \lambda_i = 1$. In the feature space, the optimization problem in Eq. (3.42) may be recast as:

$$S_{p}^{2} = \min_{\lambda} \max_{\mu} \left[\left(\Phi\left(\boldsymbol{x}_{p}\right) - \sum_{i \in \mathcal{I}_{usv}^{-p}} \lambda_{i} \Phi\left(\boldsymbol{x}_{i}\right) \right)^{2} + 2\mu \left(\sum_{i \in \mathcal{I}_{usv}^{-p}} \lambda_{i} - 1 \right) \right]$$

which after expansion gives:

$$S_{p}^{2} = \min_{\lambda} \max_{\mu} \left[\langle \Phi(\mathbf{x}_{p}), \Phi(\mathbf{x}_{p}) \rangle + \sum_{i \in \mathcal{I}_{usv}^{-p}} (\lambda_{i} \Phi(\mathbf{x}_{i}))^{2} - 2 \sum_{i \in \mathcal{I}_{usv}^{-p}} \lambda_{i} \left(\langle \Phi(\mathbf{x}_{p}), \Phi(\mathbf{x}_{i}) \rangle - \mu \right) - 2\mu \right].$$
(3.44)

Let now K_{usv} be the $n_{usv} \times n_{usv}$ matrix of the dot products in the feature space between the unbounded support vectors, *i.e.*

 $K_{\text{usv}} = \begin{bmatrix} k (\mathbf{x}_i, \mathbf{x}_j) \end{bmatrix}_{i, j \in \mathcal{I}_{\text{usv}}} \text{ and } n_{\text{usv}} = \text{Card} (\mathcal{I}_{\text{usv}}).$

The following matricial notation is introduced:

$$\widetilde{K}_{\rm usv} = \begin{bmatrix} K_{\rm usv} & \mathbf{1} \\ \mathbf{1}^T & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \widetilde{\lambda} = \begin{bmatrix} \lambda \\ \mu \end{bmatrix}, \tag{3.45}$$

where $\lambda = (\lambda_i)_{i \in \mathcal{I}_{usv}}$. Eq. (3.44) then rewrites:

$$S_{p}^{2} = \min_{\lambda} \max_{\mu} \left[k \left(x_{p}, x_{p} \right) + \widetilde{\lambda}^{T} H_{p} \widetilde{\lambda} - 2 \widetilde{v}_{p}^{T} \widetilde{\lambda} \right], \qquad (3.46)$$

where \widetilde{v}_p is the *p*-th column of K_{usv} and $H_p = \begin{bmatrix} K_{usv}^{-p} & \mathbf{1} \\ \mathbf{1}^T & \mathbf{0} \end{bmatrix}$, with $K_{usv} = \begin{bmatrix} k (x_i, x_j) \end{bmatrix}_{i,j \in \mathcal{I}_{usv}^{-p}}$.

By requiring the gradient of Eq. (3.46) with respect to λ to be zero, one can get the expression of $\tilde{\lambda}$ as follows:

$$rac{\partial S_p^2}{\partial oldsymbol{\lambda}} = 2 H_p \widetilde{oldsymbol{\lambda}} - 2 \widetilde{v}_p,$$

which implies:

$$\widetilde{\lambda} = H_p^{-1} \widetilde{v}_p. \tag{3.47}$$

By injecting Eq. (3.47) into Eq. (3.46), the expression of S_p^2 boils down to:

$$S_p^2 = k\left(\boldsymbol{x}_p, \boldsymbol{x}_p\right) - \widetilde{\boldsymbol{v}}_p^T \boldsymbol{H}_p^{-1} \widetilde{\boldsymbol{v}}_p.$$
(3.48)

At this point, it is possible to further simplify this expression. However it depends of the category to which the vector x_p belongs:

• x_p is not a support vector $(\alpha_p = 0)$

In this case, removing the point from the training set would not change the SVR model. There is hence no need to even compute the span. The LOO error is zero.

x_p is an unbounded support vector $(0 < \alpha_p < C)$

In this case, a further simplification is possible by introducing the so-called Woodbury formula (Lütkepohl, 1996) which reads:

$$\begin{bmatrix} A_1 & A^T \\ A & A_2 \end{bmatrix}^{-1} = \begin{bmatrix} B_1 & B^T \\ B & B_2 \end{bmatrix},$$
 (3.49)

where $B_1 = (A_1 - AA_2^{-1}A^T)^{-1}$, *B* and *B*₂ are matrices of appropriate sizes.

This formula, applied to Eq. (3.48) with $A_1 = k(\mathbf{x}_p, \mathbf{x}_p)$, $\mathbf{A} = \tilde{\mathbf{v}}_p$ and $A_2 = \mathbf{H}_p$, leads to the following expression of the span:

$$S_p^2 = \frac{1}{\left(\tilde{K}_{\rm usv}^{-1}\right)_{pp}}.$$
(3.50)

The last equality is due to the fact that the left-hand side of Eq. (3.49) is equal to K_{usv} with its *p*-th row and column reordered.

• x_p is a bounded support vector $(\alpha_p = C)$

In this case the simplification brought by the Woodbury formula is no more possible. However, we can point out that $H_p = \tilde{K}_{usv}$ since $\tilde{K}_{usv} = K_{usv}$. The expression of the span is therefore the following:

$$S_p^2 = k\left(\boldsymbol{x}_p, \boldsymbol{x}_p\right) - \widetilde{\boldsymbol{v}}_p^T \boldsymbol{K}_{\text{usv}}^{-1} \widetilde{\boldsymbol{v}}_p.$$
(3.51)

To sum this up, an approximation of the leave-one-out error can be computed as a by-product of the model construction following Eq. (3.43). The only additional computation required is the inversion of the matrix \tilde{K}_{usv} of size $(n_{usv} + 1) \times (n_{usv} + 1)$ while computing the span estimates from Eq. (3.50) and (3.51).

To be exhaustive, we should also point out that the same developments have been made by Chang and Lin (2005) for L_2 -SVR as well. The resulting approximation reads:

$$\widehat{e}_{\text{loo}} = \frac{1}{n} \left(\sum_{p=1}^{n} \left(\alpha_p + \alpha_p^* \right) S_p^2 \right) + \varepsilon, \qquad (3.52)$$

where $\alpha^{(*)}$ is solution of Eq. (3.37) and S_p^2 is computed this time by considering the span of all support vectors (*i.e.* { $i : \alpha_i + \alpha_i^* > 0$ }), not the only unbounded ones.

Finally, for classification, the LOO errors are also computed in the same way, see for example Chapelle et al. (2002).

3.3.5 An illustration of support vector regression

In this section, we illustrate the outcomes of parameters identification in different settings for support vector regression. The target function is one-dimensional and reads:

$$\mathcal{M}(x) = 4x^2 \left(x^2 - 1\right) - 0.5x^2 + 1, \tag{3.53}$$

where *x* is defined in X = [-1, 1].

For now, we disregard the matter of selecting the most adequate design of experiments. The dataset therefore consists of eight points evenly distributed over X. Gaussian kernel is considered by default.

3.3.5.1 Manual selection

Let us start by a manual choice of the parameters. Four cases are investigated as described in Table 3.1. Case #1 is the closest to the heuristics introduced above. That is, *C* is set equal to the range of the outputs and *l* the average distance between the input points. As for ε , we could not use the suggestion of Kwok (2001) as our data are generated noise-free. We therefore arbitrarily set it as roughly 1% of the output range. In cases #2, #3 and #4, the three parameters are dramatically altered one-by-one. The idea is to highlight the effect of each parameter in the model.

Parameter	Case #1	Case #2	Case #3	Case #4
С	1.2	0.12	1.2	1.2
1	0.2	0.2	0.02	0.2
ε	0.05	0.05	0.05	0.25

Table 3.1: Random choice of the SVR parameters

Figure 3.7 illustrates the resulting models. The setting of case #1 results in a rather good model. All the points are unbounded support vectors as they belong to the boundary of the insensitive tube. In case #2, *C* is divided by ten. This produces a "flatter" model than the one in case #1. This makes sense as the emphasis is shifted from the empirical risk to the regularization term. In case #3, *l* is made ten times smaller than in case #1 whereas *C* takes on its original value. The approximation is now almost interpolating (with respect to the ε -insensitive tube) with high variations near the sampling point. The regression considers each point independently as their radius of influence is significantly reduced. Finally in case #4, the ε -tube is made wider. The number of SVs is thus halved. This parameter actually controls the amount of support vectors among the training points.

3.3.5.2 Automatic selection

Comparison of the true and estimated leave-one-out errors

We have explained earlier how to compute an estimate of the leave-one-out error. In this section, we compare it with its true value, *i.e.* to the one obtained from a real leave-one-out resampling.

For this purpose, we start by exploring the dependence of the LOO errors to the parameters *C* and *l* while $\varepsilon = 0.05$ is held constant. The ranges of variation of *C* and *l* are set wide and shown in a log₁₀ scale. Figure 3.8 shows the true and estimated values of the LOO error for both L_1 - and L_2 -SVR. As expected, the estimated error is quite close the true one. We have the same trend when considering the two other pairs of parameters ((C, ε) and (l, ε)). There are interesting properties that can be extracted from these figures. Despite a good accuracy, the estimated error exhibits some noise. The magnitude of the noise depends on many factors. First, the noise exists only in some regions of the space. Often, but not always, the approximation remains smooth for high values of *C* and small values of ε . Coincidentally, it is the region where the LOO error is minimal. Secondly, the noise is more important for L_1 -SVR than for L_2 -SVR. The explanation for this fact lies in the very computation of the span estimate. Recall that, the computation of the span of a vector depends on the category it belongs to. The difficulty arises in the discrimination of the bounded and unbounded SVs. In fact, we never have $\alpha^{(*)} = C$ in



Figure 3.7: SVR models with different set of hyperparameters. The blue triangles are the training points. Among them, the support vectors are circled in cyan. The plain and dotted lines are respectively the true function and SVR approximations.

a numerical implementation but rather $|\alpha^{(*)} - C|$ very small for bounded SVs. Chapelle (2002) decides that a SV is bounded when $|\alpha^{(*)} - C| \leq 10^{-6} \max \alpha^{(*)}$. In our experience the inversion of *K* is less difficult when we have few bounded SVs. We then set an arbitrarily very low threshold (in the magnitude of 10^{-12}). For L_2 -SVR, all support vectors are considered. There is no discrimination between the two kinds of SVs. This results in an easily invertible matrix *K* and therefore to a smoother LOO error for L_2 -SVRs.

Minimizing the span estimate of leave-one-out error

As illustrated above, the estimate of the LOO error is a noisy function. In order to optimize it, Vapnik and Chapelle (2000) smoothed it by introducing a regularization term in the expression of S_p^2 . This allowed them to consider gradient-based algorithms to find the optimal value of



Figure 3.8: Comparison of the true and estimated LOO errors.

the model parameters. Chang and Lin (2005) used the same trick in LIBSVM (Chang and Lin, 2011). In this thesis, we have opted for a first-order optimization method. Many authors have considered such algorithms, *e.g.* simulated annealing as in Pai and Hong (2006); Lin et al. (2008) or particle swarm optimization as in Lin et al. (2008). Here, we specifically use the cross-entropy-based optimizer, a stochastic global optimization algorithm well suited for noisy multi-extremal problems (Bourinet, 2015). It is hence not crucial to smooth the error estimates to be minimized.

The cross-entropy method is a random search technique developed by Rubinstein (1997). It copes with multi-extremal or non-convex objective functions. It is also robust with noisy functions and therefore well suited for the span estimate of the LOO. It proceeds by generating a random sequence of solutions which are steered toward the optimal or near-optimal solution. Typically, the samples are generated by a random mechanism, say a probability distribution function of given parameters. General details of the method can be found in Hu and Hu (2009), Kroese et al. (2006) or Boer et al. (2004).

In our case, we explore the parameter space by sampling candidates through a truncated normal distribution. At each iteration, a percentage of the best candidates are selected and the parameters of the normal PDF are updated accordingly: the mean and standard deviation of the best samples are used to generate the next candidates. Practically, the following steps are implemented:

- 1. Define the initial PDF by its mean μ_0 and standard deviation σ_0 . Set i = 1;
- 2. Generate a sample of *N* points following $\mathcal{N}(\mu_{i-1}, \sigma_{i-1}^2)$;
- 3. Generate *N* SVR models with parameters corresponding to the *N* sampled candidates and evaluate theirs span estimates of the LOO error.
- 4. Select the best $N_{\rm el}$ candidates with respect to the LOO error minimization;
- 5. Compute the mean $\tilde{\mu}_i$ and standard deviation $\tilde{\sigma}_i$ of these points;
- 6. Update the PDF parameters according to the following scheme:

$$\mu_{i} = \alpha \widetilde{\mu}_{i} + (1 - \alpha) p_{\text{best}},$$

$$\sigma_{i} = \beta_{i} \widetilde{\sigma}_{i} + (1 - \beta_{i}) \sigma_{i-1},$$
(3.54)

where α and $\beta_i = \beta + \beta (1 - 1/i)^q$ are smoothing parameters and p_{best} is the vector gathering the parameters that produced the best model so far;

7. Stop when max $\sigma \leq \delta_{CE}$, otherwise update *i* and restart the steps 2 to 6.

There are a few parameters to tune. The range of the search space is set quite large. The starting point is chosen as its center. Its associated standard deviation is equal to twice the range in each direction. The smoothing parameters are chosen dynamic for σ . This is intended to help us avoid premature convergence to a local minimum. We set $\alpha = \beta = 0.4$ and q = 10. The number of points sampled in each iteration depends on the difficulty of the optimization problem. The higher it is the more robust the algorithm. However, when the design size is large, the cost of constructing the model is relatively important. So we capped *N* to 300 for high-dimensional problems. Of course, for smooth functions, we can afford a smaller size of *N*. Finally for the next iteration, we usually take $N_{\rm el} = 0.05N$.

As an application, we use this procedure to select the parameters of the model to approximate the function used throughout this section. Figure 3.9 illustrates the results for L_2 -SVR. The surrogate model perfectly matches the true function. The insensitive tube is almost zero and all data points are support vectors. In Table 3.2, we compare the results produced by the manual and the cross-entropy based selections. As it will be the case in all the applications in this manuscript the value of *C* is extremely high, much higher than the suggestion of various heuristics proposed. Conversely ε tends to zero. This will also be the case anytime the space is not densely sampled. The value ε does scale with the level of noise in the data, yet for this noise to be detectable by the model, a high density of points in the design space is required.



Figure 3.9: L_2 -SVR model with parameters set up by cross-entropy search

Table 3.2: Heuristics versus cross-entropy based selection of the parameter	eters
---	-------

Selection	Model	С	1	ε	NMSE	\widehat{e}_{loo}
Heuristics	L_1 -SVR	1.2445	0.2857	0.01	0.0386	0.4543
	L_2 -SVR	1.2445	0.2857	0.01	0.0490	0.4910
Cross-entropy method	L_1 -SVR	$1.67 \ 10^7$	1.9729	$1.07 \ 10^{-7}$	$8.63 \ 10^{-8}$	0.0041
	L_2 -SVR	$3.03 \ 10^{13}$	1.9719	$5.53\ 10^{-8}$	$8.70 \ 10^{-8}$	0.0042

3.4 Kriging or Gaussian process modeling

The early ideas on Gaussian process modeling have been developed in the 1950s by the South-African mining engineer Danie G. Krige in his master thesis (Krige, 1951). His work was motivated by the need to map ores concentrations in the Witwatersrand soils. The resulting prediction methodology has later been formalized by the French Georges Matheron (Matheron, 1962) who named it *Kriging* as a tribute to Krige.

Let us recall the set of observations introduced in the previous section $\mathcal{D} = \{(x_i, y_i), i \in \mathcal{D}\}$

 $\{1, ..., n\}, \forall x_i \in \mathbb{X} \subset \mathbb{R}^s, y_i \in \mathbb{Y} \subset \mathbb{R}\}$. In geostatistics, the inputs x_i are considered to be spatial coordinates and the outputs y_i are for example concentration values. The main idea is then to assume spatial correlations between the outputs. Indeed for ore mining, near locations are more likely to have similar concentration values. This spatial dependency is essentially characterized by the so-called *variogram*. Another main concept is to consider the output as a realization of an unknown stochastic process *Y* as will be explained in the remainder of this section.

This approach was limited to two or three dimensional problems. It has later been adapted to computer experiments by Sacks et al. (1989). In this context, the inputs are no longer restricted to spatial coordinates but may be high-dimensional parameters.

3.4.1 Introduction to Gaussian random functions

We have introduced in Chapter 2 some elements of probability theory. We now extend the list to an additional element which is the root for inference in Kriging, namely a stochastic process. It is defined as a set $\{Y(x) \mid x \in X\}$ of random variables. Formally speaking, it writes as the following mapping (Dubourg, 2011):

$$\begin{array}{c} Y\left(\boldsymbol{x}\right): \mathbb{X} \times \Omega \to \mathbb{Y} \\ \left(\boldsymbol{x}, \omega\right) \mapsto y\left(\boldsymbol{x}\right), \end{array} \tag{3.55}$$

where Ω is the sample space introduced in section 2.1.3.2, $y(x) = Y(x, \omega)$ is a realization or *sample path* of the random process for a particular value ω , say $\omega = \omega_0$. Conversely, for a particular value of x, say $x = x_0$, $Y(x_0, \omega)$ is a random variable over the probability space $(\Omega, \mathcal{F}, \mathcal{P})$.

The above definition is quite general. For the purpose of Kriging, we are particularly interested in Gaussian stochastic processes and this, for two reasons (Cressie, 1993). The first one is pragmatic as inference and prediction are clearly easier to develop analytically using the Gaussian assumption. The second is related to the central limit theorem which allows us to expect that the combination of small non-Gaussian effects will asymptotically tend to a Gaussian distribution. In this light, a stochastic process is said Gaussian if for any subset $\{x_1, \ldots, x_n\}$, the vector $\{Y(x_1), \ldots, Y(x_n)\}$ has a multivariate normal distribution (Santner et al., 2003). Such a process is entirely determined by its mean and covariance functions:

$$\mu(\mathbf{x}) \equiv \mathbb{E}\left[Y\left(\mathbf{x}\right)\right] \quad \text{for } \mathbf{x} \in \mathbb{X}$$

$$C(\mathbf{x}, \mathbf{x}') \equiv \text{Cov}\left[Y\left(\mathbf{x}\right), Y\left(\mathbf{x}'\right)\right] \quad \text{for } \mathbf{x}, \mathbf{x}' \in \mathbb{X},$$
(3.56)

There are many restricting assumptions that have to be made for inference using Gaussian random processes to be possible. Defining such assumptions is beyond the scope of this

manuscript (see *e.g.* Adler (1981); Cressie (1993)). We will however comment some desired relevant properties.

Without getting too much into details, one property of prime importance required for inference from a single stochastic process to be possible is *ergodicity*. In fact, recall that we are only given a set of observations $\mathcal{Y} = \{y_1 = y(x_1), \dots, y_n = y(x_n)\}$ which are considered to be realizations for a particular value of $\omega_0 \in \Omega$. We thus have only partial information on the process *Y*. For inference to be possible despite this limited knowledge of *Y*, the function needs to exhibit some regularity over X. It turns out that this characteristics is guaranteed by the *ergodicity* property. Furthermore an ergodic random function is a subset of a class of widely used processes, namely the stationary one. A process is said to be stationary if its properties are invariant by translation. That is, its mean is constant and its covariance only depends on the shift $\tau = x - x'$:

$$\mu(\mathbf{x}) = \mu_0, \quad \mathbf{x} \in \mathbb{X}$$

Cov $[Y(\mathbf{x}), Y(\mathbf{x}')] = C(\mathbf{x} - \mathbf{x}') = C(\tau), \quad \mathbf{x}, \mathbf{x}' \in \mathbb{X}.$ (3.57)

One of the advantages of using Gaussian random functions is that it allows for an easily verifiable sufficient condition for ergodicity. In fact any second-order stationary process is ergodic if $C(\tau) = 0$ as $\tau \to \infty$ (Cressie (1993) following Blum and Reich (1982)). This crucial property makes inference from a single sample path valid. Therefore, expectations over Ω can be estimated by spatial averages.

3.4.2 The two-stage model

3.4.2.1 The different types of Kriging

To learn from a limited set of samples, Kriging catches two levels of variability, namely global and local. The underlying model is considered to be a realization of a stochastic process which reads:

$$Y(x) = \mu(x) + Z(x),$$
 (3.58)

where μ (\mathbf{x}) is a deterministic function approximating the mean trend of the model and Z (\mathbf{x}) is a second-order stationary Gaussian process with zero mean and autocovariance Cov [Z (\mathbf{x}), Z (\mathbf{x}')] = $\sigma^2 R$ ($\mathbf{x} - \mathbf{x}'$). Here we have introduced the process variance $\sigma^2 = C(0)$ and the so-called *autocorrelation function* R that provides the dependency properties.

This representation is known as a two-stage Gaussian prior model. The first stage gives the large scale variability of the model and may consist of linear combinations of a functional basis. The second one accounts for short scale variability. This allows a greater flexibility in the class of possible learning functions while keeping the advantages, in term of theoretical simplifications provided by stationarity. In fact, *Y* is not stationary but *Z* is.

The nature of the deterministic part gives rise to various types of Kriging:

- Simple Kriging: The deterministic part is supposed to be known and constant;
- Ordinary Kriging: The deterministic part is supposed to be constant but is unknown. It is usually considered as the mean value of the observations;
- Universal Kriging: The deterministic part depends on *x* and is cast as a linear combination of basis functions:

$$\mu\left(\mathbf{x}\right) = \sum_{j=1}^{P} \beta_j f_j\left(\mathbf{x}\right),\tag{3.59}$$

where $\beta = {\beta_j, j = 1, ..., p}$ is a weight vector and $f = {f_j, j = 1, ..., p}$ is a collection of regression functions.

In the sequel, we make all the developments with the universal Kriging as it is the most general case. But in practice, we mostly use ordinary Kriging. Two main reasons justify this choice. First, the choice of the functional basis is not obvious. One has to make assumptions about the underlying model. We will readily use such assumptions when available, say from an expert knowledge. Yet in general, we do not have any information about the trend and it is preferable to stick to ordinary Kriging. Second, the number of functions *p* in the regression basis is quite important as it conditions the number of points required in the training set. Indeed to estimate the regression weights, the number of observations *n* should be at least equal to *p*. For example, a second-order polynomial (with interactions) in an *s*-dimensional space would require $n \ge p = \frac{1}{2} (s+1) (s+2)$. We are therefore limited in the choice of the functional basis due to limits imposed on the sample size.

On the other hand, the departure from the mean trend is caught by the zero-mean process *Z*, which is entirely determined by its covariance function. Alternatively, it may be more conveniently defined by the auto-correlation function *R*:

$$R(\tau) = \frac{1}{\sigma^2} C(\tau), \quad \text{for} \quad \tau = x - x' \quad \text{and} \quad x, x' \in \mathbb{X},$$
(3.60)

where $\sigma^2 = C(\mathbf{0}) > 0$ is the variance of the process.

In this respect, the first step in learning with Kriging is the choice of the deterministic trend and the autocorrelation function. We now review some widely used autocorrelation functions. Before this, let us start by briefly describing which properties are needed for a given autocorrelation to be valid.

3.4.2.2 Auto-correlation functions for Kriging

Admissible correlation functions

The first requirement for a covariance or correlation function is *symmetry*. Indeed , for a stationary Gaussian stochastic process (GSP), we have Cov [Y(x), Y(x')] = Cov [Y(x'), Y(x)] and henceforth the symmetry about the origin:

$$C(\tau) = C(-\tau)$$
 and $R(\tau) = R(-\tau)$. (3.61)

Besides, admissible auto-covariance functions must be non-strictly *positive definite*. That is, for any subset $\mathcal{X} = \{x_1, \ldots, x_n\}$ and any vector $w = \{w_1, \ldots, w_n\} \in \mathbb{R}^n$, the following inequality holds:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j C(\mathbf{x}_i, \mathbf{x}_j) \ge 0.$$
(3.62)

Eventually, we introduce the properties relating to the average behavior of the sample paths, more specifically the *mean-square continuity*. Following Santner et al. (2003), a stationary process Z with finite second moments is said to be mean-square continuous on $x_0 \in X$ if

$$\lim_{x \to x_0} \mathbb{E}\left[(Z(x) - Z(x_0))^2 \right] = 0.$$
(3.63)

This expectation may be written in terms of the covariance function:

$$\mathbb{E}\left[\left(Z\left(\boldsymbol{x}\right)-Z\left(\boldsymbol{x}_{0}\right)\right)^{2}\right]=2\left(C\left(\boldsymbol{0}\right)-C\left(\boldsymbol{x}-\boldsymbol{x}_{0}\right)\right).$$
(3.64)

So *Z* is mean-square continuous on X if $C(h) \rightarrow C(0) = \sigma^2$ as $h \rightarrow 0$. In terms of autocorrelation function, the MS continuity condition reads:

$$R(\boldsymbol{\tau}) = \frac{1}{\sigma^2} C(\boldsymbol{\tau}) \to 1 \quad \text{as} \quad \boldsymbol{\tau} \to \boldsymbol{0}.$$
 (3.65)

This property provides a way of generating valid covariance functions using *spectral distributions* (see Santner et al. (2003) for some examples). Next we review some widely used ones as described in Rasmussen and Williams (2006) for instance.

3.4.2.3 Sample path properties

Examples of auto-correlation functions

The sample paths generated for a given process are highly affected by the choice of the auto-correlation function. These properties can be controlled by some parameters of the auto-correlation functions. Let us gather them in the vector $\boldsymbol{\theta}$ so that $R(\boldsymbol{x}, \boldsymbol{x}') \equiv R(\boldsymbol{x} - \boldsymbol{x}'; \boldsymbol{\theta})$.

For mathematical convenience, the study is usually restricted to the case of auto-correlation functions which write as a product of univariate ones:

$$R(\mathbf{x},\mathbf{x}';\boldsymbol{\theta}) = R(\mathbf{x}-\mathbf{x}';\boldsymbol{\theta}) = \prod_{i=1}^{s} R(x_i - x_i';\theta_i).$$
(3.66)

Another important property of an auto-correlation function is its differentiability at the origin (*i.e.* when $x \to x'$). This property actually controls the smoothness of the generated process. Hence, this information should be taken into account while choosing the appropriate family of auto-correlation functions, according to the designer experience of the function to approximate.

Let us now introduce examples of widely used auto-correlation functions (Rasmussen and Williams, 2006; Koehler and Owen, 1996). For the sake of simplicity, we consider one-dimensional cases knowing that the multi-dimensional extension can be formulated following Eq. (3.66).

• Linear

$$R(x, x'; l) = \max\left(0, 1 - \frac{|x - x'|}{l}\right),$$
 (3.67)

where l > 0 is the so-called *characteristic length-scale*.

• Generalized exponential

$$R\left(x, x'; \boldsymbol{\theta}\right) = \exp\left(-\left(\frac{x-x'}{l}\right)^{\gamma}\right) \quad \text{for} \quad 0 < \gamma \le 2 \quad \text{and} \quad l > 0, \tag{3.68}$$

where $\theta = \{l, \gamma\}$. There the parameter γ directly controls the degree of smoothness of the associated process. The higher the value, the smoother the sample path. There are two widely used cases. When $\gamma = 1$, we have the exponential auto-correlation function producing the so-called *Orstein-Uhlenbeck* process. The other remarkable case is when $\gamma = 2$, which corresponds to the Gaussian auto-correlation function introduced below. It is actually the only value of γ for which the generated process is mean-square differentiable.

3. METAMODELING TECHNIQUES

• Gaussian

$$R(x, x'; l) = \exp\left(-\frac{1}{2}\left(\frac{x-x'}{2l}\right)^2\right),$$
(3.69)

It is probably the most used auto-correlation function for learning problems. The generated processes are infinitely differentiable and thus very smooth.

• Matérn

$$R\left(x, x'; \boldsymbol{\theta}\right) = \frac{1}{2^{\nu-1} \Gamma\left(\nu\right)} \left(\sqrt{2\nu} \frac{|x-x'|}{l}\right)^{\nu} \mathcal{K}_{\nu}\left(\sqrt{2\nu} \frac{|x-x'|}{l}\right), \qquad (3.70)$$

where $\theta = \{l, v\}, v \ge 1/2$ is the so-called *shape* parameter, Γ the Euler Gamma function and \mathcal{K}_v the modified Bessel function of the second kind. This auto-correlation, just as the generalized exponential, has a control parameter for the smoothness, namely v. It is even more flexible. In fact, it is up to *k*-times differentiable for any positive integer *k* such that v > k. More interesting, if v is chosen as a half-integer, that is v = k +1/2, an analytical expression of the auto-correlation can be derived. The most practical cases correspond to v = 3/2 and v = 5/2 which are respectively once and two-times differentiable. Rasmussen and Williams (2006) argue that, in the absence of any other prior knowledge, there is no use for higher-order differentiability (*i.e.*, from v = 7/2 and so on) because one cannot make any difference in their smoothness. So the Matérn 3/2and Matérn 5/2 are mostly used and respectively write:

$$R\left(x, x'; l\right) = \left(1 + \sqrt{3} \frac{|x - x'|}{l}\right) \exp\left(-\sqrt{3} \frac{|x - x'|}{l}\right), \quad \text{when} \quad \nu = 3/2, \qquad (3.71)$$

$$R(x, x'; l) = \left(1 + \sqrt{5}\frac{|x - x'|}{l} + \frac{5}{3}\frac{(x - x')^2}{l^2}\right) \exp\left(-\sqrt{5}\frac{|x - x'|}{l}\right), \text{ when } \nu = 5/2,$$
(3.72)

Some illustrations

To end this section, some trajectories of Gaussian processes are illustrated by focusing on the general exponential auto-correlation function. The trajectories are generated based on a Cholesky decomposition method. That is, consider a Gaussian random vector $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C})$ of size *n*. It may be simulated following:

$$\mathbf{Z} = L\boldsymbol{\Xi} + \boldsymbol{\mu},\tag{3.73}$$

where *L*, defined such that $C = LL^T$, is the lower triangular matrix of the Cholesky decomposition of *C* and $\Xi = \{\Xi_i \sim \mathcal{N}(0, 1), i = 1..., n\}$.

In the subsequent examples, we consider a zero-mean process with unit variance. The parameters of the auto-correlation function are varied one-by-one to highlight their effects on the generated trajectories. The same realizations of Ξ are used for all the cases.

Some examples with different values of the shape parameter: $\gamma = \{1, 1.5, 2\}$ are plotted in Figure 3.10. As expected, the trajectories are smoother when increasing the value of γ and appear differentiable for $\gamma = 2$.

On the other hand, the characteristic length-scale values are varied within $l = \{0.01, 0.1, 1\}$ in Figure 3.11 where the Gaussian auto-correlation function is considered. This parameter describes how far a given point influences its neighborhood. As shown in Figure 3.11a, the shorter



Figure 3.10: Illustration of the smoothness of Gaussian process trajectories with the generalized exponential autocorrelation function with different values of the power parameter.



Figure 3.11: Illustration of the speed of variation of Gaussian process trajectories considering the Gaussian autocorrelation function with different characteristic length-scales.

the characteristic length-scale, the faster $R(\tau, l)$ tends to zero. The corresponding trajectories become less and less correlated as *l* becomes large.

In summary, the shape parameter controls the regularity and the length-scale the speed of variation of the process. We will see in the sequel how these two information are used when building a Kriging model. Usually, the type of auto-correlation function is chosen a priori. This includes a given value of the shape parameter, when available. However, the length-scales are selected while training the model.

3.4.3 Formulation of the Kriging model

3.4.3.1 The generalized least-square estimate of the parameters

As said earlier, we will focus on the most general formulation, namely universal Kriging. Recall that the function to approximate is considered as a realization of a Gaussian stochastic process which reads:

$$Y(\mathbf{x}) = \sum_{j=1}^{p} \beta_j f_j(\mathbf{x}) + Z(\mathbf{x}), \qquad (3.74)$$

where $\{f_j, j = 1, ..., p\}$ are preselected deterministic functions and $\{\beta_j, j = 1, ..., p\}$ are coefficients to be determined.

The first step is then to estimate the parameters β and σ of the regression model. For that purpose, a frequentist approach is considered. In this setting, the departure of the linear regression from the observation is assumed to be a Gaussian random noise:

$$\mathbf{Z} = \mathbf{Y} - \mathbf{F}\boldsymbol{\beta} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{R}\right), \qquad (3.75)$$

where $Y = \{Y_i = Y(x_i), i = 1, ..., n\}$ and *F* is an $n \times p$ matrix defined by $F = \{F_{ij} = f_j(x_i), i = 1, ..., n, j = 1, ..., p\}.$

By requiring this noise to be minimal, one can derive the *least-square estimates* of the parameters β and σ . Another point of view would be to search for the parameters so that the noise is most likely to be Gaussian as assumed *a priori*. This results in the so-called *maximum likelihood esti-mates*. As argued by Myung (2003), these two approaches are equivalent since the least-square and maximum likelihood methods result in the same estimates when the data are independent and normally distributed.

In this case, the likelihood function is obtained by inverting the role of the observations and the parameters in the multivariate normal probability function:

$$L\left(\boldsymbol{\beta},\sigma^{2}|\boldsymbol{y}\right) = \frac{1}{\left(\left(2\pi\sigma^{2}\right)^{n}\det\boldsymbol{R}\right)^{\frac{1}{2}}}\exp\left(-\frac{1}{2\sigma^{2}}\left(\boldsymbol{y}-\boldsymbol{F}\boldsymbol{\beta}\right)^{T}\boldsymbol{R}^{-1}\left(\boldsymbol{y}-\boldsymbol{F}\boldsymbol{\beta}\right)\right).$$
(3.76)

Maximizing Eq. (3.76) is equivalent to minimizing its opposite logarithm. For convenience, the latter approach is preferred to the former. The estimates are henceforth sought according to the following optimization problem:

$$\left(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\sigma}}\right) = \arg \min_{(\boldsymbol{\beta}, \boldsymbol{\sigma}) \in \mathbb{R}^{p} \times \mathbb{R}^{*+}} -\log L\left(\boldsymbol{\beta}, \boldsymbol{\sigma} | \boldsymbol{y}\right)$$

$$= \arg \min_{(\boldsymbol{\beta}, \boldsymbol{\sigma}) \in \mathbb{R}^{p} \times \mathbb{R}^{*+}} -\log \frac{1}{\left(\left(2\pi\right)^{n} \det \boldsymbol{R}\right)^{\frac{1}{2}}} + n\log \boldsymbol{\sigma} + \frac{1}{2\sigma^{2}}\left(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}\right)^{T} \boldsymbol{R}^{-1}\left(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}\right).$$

$$(3.77)$$

The first-order optimality conditions then read:

$$\begin{cases} \nabla_{\boldsymbol{\beta}} \left(-\log L\left(\boldsymbol{\beta}, \sigma | \boldsymbol{y} \right) \right) &= 0\\ \frac{\partial \left(-\log L\left(\boldsymbol{\beta}, \sigma | \boldsymbol{y} \right) \right)}{\partial \sigma} &= 0 \end{cases}$$
(3.78)

By expanding the first condition, we have:

$$\nabla_{\boldsymbol{\beta}} \left(-\log L\left(\boldsymbol{\beta}, \sigma | \boldsymbol{y}\right) \right) = \frac{1}{\sigma^2} \boldsymbol{F}^T \boldsymbol{R}^{-1} \left(\boldsymbol{y} - \boldsymbol{F} \boldsymbol{\beta} \right) = 0$$

$$\Leftrightarrow \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{y} - \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F} \boldsymbol{\beta} = 0,$$

This leads to the following least-square estimate of β :

$$\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F}\right)^{-1} \boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{y}.$$
(3.79)

As for the second condition, the following expansion may be written:

$$\frac{\partial \left(-\log L\left(\boldsymbol{\beta}, \sigma | \boldsymbol{y}\right)\right)}{\partial \sigma} = \frac{n}{\sigma} - \frac{1}{\sigma^3} \left(\boldsymbol{y} - \boldsymbol{F} \boldsymbol{\beta}\right)^T \boldsymbol{R}^{-1} \left(\boldsymbol{y} - \boldsymbol{F} \boldsymbol{y}\right),$$

which leads to the following least-square estimate of σ^2 :

$$\widehat{\sigma}^2 = \frac{1}{n} \left(\boldsymbol{y} - \boldsymbol{F} \boldsymbol{\beta} \right)^T \boldsymbol{R}^{-1} \left(\boldsymbol{y} - \boldsymbol{F} \boldsymbol{\beta} \right).$$
(3.80)

Once *R* is known, Eq. (3.79) and (3.80) will provide the estimators $(\hat{\beta}, \hat{\sigma}^2)$ necessary in the expression of the Kriging predictor.

3.4.3.2 Kriging as the best linear unbiased predictor

The universal Kriging formulation is obtained by requiring the prediction for a new point $Y_0 = Y(x_0)$ to respect three conditions:

• Linearity with respect to the observations

$$\widehat{Y}_0 = \sum_{i=1}^n a_i Y_i = \boldsymbol{a}^T \boldsymbol{Y}, \qquad (3.81)$$

where $a \equiv a(x_0) = \{a_i, i = 1, ..., n\}$ are coefficients to be determined.

Non-biasedness

$$\mathbb{E}\left[\widehat{Y}_0 - Y_0\right] = 0, \tag{3.82}$$

• Minimal variance *i.e.* the best predictor among all

$$\widehat{Y}_0^* = \arg\min_{\widehat{Y}_0} \mathbb{E}\left[\left(\widehat{Y}_0 - Y_0\right)^2\right].$$
(3.83)

Following the development in Santner et al. (2003), the conditions in Eq. (3.81) and (3.82) lead to:

$$\mathbb{E}\left[\widehat{Y}_{0}-Y_{0}\right] = \mathbb{E}\left[a^{T}Y-Y_{0}\right]$$
$$= \mathbb{E}\left[a^{T}\left(F\beta+Z\right)-f\left(x_{0}\right)^{T}\beta-Z_{0}\right]$$
$$= \mathbb{E}\left[a^{T}Z-Z_{0}\right]+\left(a^{T}F-f\left(x_{0}\right)^{T}\right)\beta=0.$$

The first summand of the right-hand side of the above equality is zero since the quantity $a^T Z - Z_0$ is a linear combination of Gaussian random variables with zero mean. The non-biasedness condition reduces then to:

$$a^{T}F - f(x_{0})^{T} = 0. (3.84)$$

In consequence, the minimal variance condition of Eq. (3.83) can be expanded using Eq. (3.84):

$$\mathbb{E}\left[\left(\widehat{Y}_{0}-Y_{0}\right)^{2}\right] = \mathbb{E}\left[\left(a^{T}Z-Z_{0}+\left(a^{T}F-f\left(x_{0}\right)^{T}\right)\beta\right)^{2}\right]$$
$$= \mathbb{E}\left[\left(a^{T}Z-Z_{0}\right)^{2}\right]$$
$$= \mathbb{E}\left[a^{T}ZZ^{T}a-2a^{T}ZZ_{0}+Z_{0}Z_{0}^{T}\right]$$
$$= a^{T}\mathbb{E}\left[ZZ^{T}\right]a-2a^{T}\mathbb{E}\left[ZZ_{0}\right]+\mathbb{E}\left[Z_{0}Z_{0}^{T}\right]$$
$$= a^{T}\sigma^{2}Ra-2a^{T}\sigma^{2}r_{0}+\sigma^{2},$$

where $\mathbf{r}_0 = \{R(\mathbf{x}_i, \mathbf{x}_0), i = 1, ..., n\}$ is a vector gathering the value of the autocorrelation function computed between \mathbf{x}_0 and each point of \mathcal{X} , such that $\mathbb{E}[\mathbf{Z}, \mathbf{Z}_0] = \sigma^2 \mathbf{r}_0$.

So the minimal variance condition may be recast as a quadratic optimization problem with respect to the coefficients *a* and with linear equality constraints:

$$a^{*} = \arg\min_{a \in \mathbb{R}^{n}} a^{T} \sigma^{2} R a - 2a^{T} \sigma^{2} r_{0} + \sigma^{2}$$

subject to:
$$a^{T} F - f(x_{0})^{T} = 0.$$
 (3.85)

Naturally, it is solved in its dual form. By introducing the Lagrangian, it is equivalent to solving:

$$(\boldsymbol{a}^{*},\boldsymbol{\lambda}^{*}) = \arg\max_{\boldsymbol{\lambda}\in\mathbb{R}^{n+}}\arg\min_{\boldsymbol{a}\in\mathbb{R}^{n}}L\left(\boldsymbol{a},\boldsymbol{\lambda}\right) = \sigma^{2}\left(1 + \boldsymbol{a}^{T}\left(\boldsymbol{R}\boldsymbol{a}-2\boldsymbol{r}_{0}\right)\right) + \left(\boldsymbol{a}^{T}\boldsymbol{F}-\boldsymbol{f}\left(\boldsymbol{x}_{0}\right)^{T}\right)\boldsymbol{\lambda}, \quad (3.86)$$

where λ is a vector of Lagrange multipliers. The first optimality conditions eventually give:

$$\begin{cases} \nabla_{a}L(a,\lambda) = 2\sigma^{2} (Ra - r_{0}) + F\lambda = 0\\ \nabla_{\lambda}L(a,\lambda) = a^{T}F - f(x_{0})^{T} = 0 \end{cases}$$
(3.87)

By pre-multiplying the first equality by $-\frac{1}{2\sigma^2} F^T R^{-1}$, we have:

$$-F^{T}R^{-1}Ra + F^{T}R^{-1}r_{0} - \frac{1}{2\sigma^{2}}F^{T}R^{-1}F\lambda = 0$$

$$-f(x_{0}) + F^{T}R^{-1}r_{0} - \frac{1}{2\sigma^{2}}F^{T}R^{-1}F\lambda = 0,$$

since $F^T R^{-1} R a = F^T a = f(x_0)$ according to Eq. (3.84). So the Lagrange vector reads:

$$\lambda^* = 2\sigma^2 \left(F^T R^{-1} F \right)^{-1} \left(F R^{-1} r_0 - f(x_0) \right).$$

Re-introducing this in the expression of $\nabla_a L$ eventually gives the coefficients *a*:

$$\boldsymbol{a}^{*} = \boldsymbol{R}^{-1} \left(\boldsymbol{r}_{0} - \boldsymbol{F} \left(\boldsymbol{F}^{T} \boldsymbol{R}^{-1} \boldsymbol{F} \right)^{-1} \left(\boldsymbol{F}^{T} \boldsymbol{R}^{-1} \boldsymbol{r}_{0} - \boldsymbol{f} \left(\boldsymbol{x}_{0} \right) \right) \right).$$
(3.88)

The last step is then to inject this expression of a^* in Eq. (3.81) to get the mean and variance of the best linear unbiased predictor (BLUP) \hat{Y}_0 :

$$\mu_{\widehat{Y}_{0}} = f_{0}^{T}\widehat{\beta} + \mathbf{r}_{0}^{T}\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{F}^{T}\boldsymbol{\beta}\right),$$

$$\widehat{\sigma}_{\widehat{Y}_{0}}^{2} = \sigma^{2}\left(1 - \mathbf{r}_{0}^{T}\mathbf{R}^{-1}\mathbf{r}_{0} + \mathbf{u}^{T}\left(\mathbf{F}^{T}\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{u}\right)$$
(3.89)

where $\hat{\beta}$ is the MLE from Eq. (3.79) and $u = F^T R^{-1} r_0 - f(x^{(0)})$ has been introduced for the sake of brevity, see *e.g.* Dubourg (2011) for the details of the calculations.

This Kriging variance acts as a local estimator of the model error. In fact, it stands for the *epistemic* uncertainty due to the lack of knowledge, in this case, the limited (finite) set of observations. This is not to be confused with *aleatory* uncertainty. The former can be reduced by improving our knowledge, more specifically by increasing the size of the design of experiments. For a practical accounting of this uncertainty, confidence intervals of the predictor are usually derived following the Gaussian assumption. That is,

$$\widehat{Y}_{0} \in \left[\mu_{\widehat{Y}_{0}} - \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)\widehat{\sigma}_{\widehat{Y}_{0}}, \mu_{\widehat{Y}_{0}} + \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)\widehat{\sigma}_{\widehat{Y}_{0}}\right] \quad \text{with probability} \quad 1 - \alpha, \qquad (3.90)$$

where $\Phi(\bullet)$ denotes the Gaussian cumulative distribution function and $(1 - \alpha)$ is the confidence level.

We will see in the next chapter how this information is used to improve the quality of the Kriging predictor in regions that matter. For now, let us explain how the Kriging model can be trained.

3.4.4 Model training in Kriging

At this point, we know the general formulation of a Kriging predictor. The dependency structure has been chosen beforehand through the specification of a family of parametric autocorrelation functions. To fully define the model, we still need to specify the parameters of the auto-correlation function. This is achieved by various techniques, the most widely-used one being *cross-validation* and *maximum likelihood*.

3.4.4.1 Maximum likelihood estimation

In the maximum likelihood approach, the hyper-parameters are sought such that the observed data are consistent with the Gaussian assumption.

Let us re-introduce the log-likelihood function from Eq. (3.77) while highlighting its dependency in θ and plugging the MLE estimates $\hat{\beta}$ and $\hat{\sigma}^2$ from Eq. (3.79) and (3.80):

$$-\log L\left(\boldsymbol{\beta}, \sigma^{2}; \boldsymbol{\theta} | \boldsymbol{y}\right) = -\log \frac{1}{\left(\left(2\pi\right)^{n} \det \boldsymbol{R}\left(\boldsymbol{\theta}\right)\right)^{\frac{1}{2}}} + n\log\widehat{\sigma}\left(\boldsymbol{\theta}\right) + \frac{1}{2\widehat{\sigma}^{2}\left(\boldsymbol{\theta}\right)}\left(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}\right)^{T} \boldsymbol{R}^{-1}\left(\boldsymbol{\theta}\right)\left(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}\right)$$
$$= \log\left(2\pi\right)^{\frac{n}{2}} + \log\det \boldsymbol{R}\left(\{\boldsymbol{\theta}\}^{\frac{1}{2}} + \frac{n}{2}\log\widehat{\sigma}^{2}\left(\boldsymbol{\theta}\right) + \frac{n}{2}\log\widehat{\sigma}^{2}\left(\boldsymbol{\theta}\right)$$
The last summand directly comes from Eq. (3.80). This equation can be further written in the following form:

$$-\log L\left(\boldsymbol{\beta}, \sigma^{2}; \boldsymbol{\theta} | \boldsymbol{y}\right) = \frac{n}{2} \left(1 + \log\left(2\pi\right)\right) + \log\left(\widehat{\sigma^{2}}\left(\boldsymbol{\theta}\right) \det \boldsymbol{R}\left(\boldsymbol{\theta}\right)^{\frac{1}{n}}\right).$$
(3.91)

The maximum likelihood (ML) of θ is equivalently found by minimizing Eq. (3.91) which ultimately boils down to the following optimization problem:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \psi\left(\boldsymbol{\theta}\right) = \widehat{\sigma^2}\left(\boldsymbol{\theta}\right) \det \boldsymbol{R}\left(\boldsymbol{\theta}\right)^{\frac{1}{n}}, \qquad (3.92)$$

where *d* is the number of parameters in θ and $\psi(\theta)$ is the so-called *reduced likelihood* function.

This reduced likelihood function is not easy to minimize for various reasons. Lophaven et al. (2002a) showed how badly conditioned is the auto-correlation function for some values of θ , leading to inaccuracy propagation in the optimization process. Beside, $\psi(\theta)$ may also have many local minima. Lophaven et al. (2002a) used a pattern search algorithm to tackle these problems (Lophaven et al., 2002b) as implemented in their Matlab toolbox. In the author experience, this toolbox is somehow limited for providing accurate estimates. Remaining in Matlab, the UQLAB toolbox (Lataniotis and Sudret, 2015) implements more efficient algorithms. In this manuscript, most of the Kriging applications are run using the R package DiceKriging (Roustant et al., 2012). As in UQLAB, many algorithms are available. In our case we use an hybrid approach. The solution of a genetic algorithm is used as a starting-point for a local gradient-based search through a second-order BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm.

3.4.4.2 Cross-validation

The cross-validation (CV) procedure is as explained in Section 3.2.2. It consists in searching for the values of the parameters that minimize the generalization error through re-sampling. As explained earlier, applying this brute force approach is expensive as one has to build *n* Kriging models for a DOE of size *n*, in the specific case of a true leave-one-out. This means solving a *n* systems of $n \times n$ linear equations for a single value of the hyper-parameters. Fortunately, just as for support vector machines, Dubrule (1983) has proposed a reliable estimate of the leave-one-out error which is obtained solely by inverting a matrix of size $n \times n$. He even generalized it to leave-K-out procedure. Another similar technique of LOO estimate for Kriging is used in Bompard (2011) following the work of Rippa (1999) with radial basis functions.

For the record, the two latter toolboxes also implement the cross-validation method for model calibration. Bachoc (2013) compares the two approaches. They conclude that ML gives better results when the covariance structure is well specified, that is it coincides with the one from which the data have been generated. In the case of misspecification, CV is better. On the ground of similar arguments Bompard (2011) uses CV for model calibration, arguing that CV is not influenced by the statistical framework, most specifically the Gaussian assumption . In this thesis, we focus on ML because the tools we use (DiceKriging) is more robust with ML than with CV.

3.4.5 An illustration of Kriging

3.4.5.1 Some properties of Kriging predictor

We now illustrate the Kriging prediction using the same one-dimensional function as for SVR, see Eq. (3.53). The design of experiments consists of eight points randomly drawn in X =

[-1,1].

Ordinary Kriging with exponential and Gaussian auto-correlation functions are used for prediction in X. Figure 3.12 shows the resulting approximation together with its 95% confidence interval. The first difference between the two cases is that as expected the exponential autocorrelation function produces a predictor which is C^0 at the training points, *i.e.* continuous but non-differentiable. On the other hand, the predictor for the Gaussian auto-correlation function is differentiable at the training points.



Figure 3.12: Kriging approximation with two different auto-correlation functions: exponential and Gaussian.

In the two cases, the prediction goes exactly through the training points and the associated variance is zero. Interpolation is indeed one of the main properties of Kriging and can be proved easily using the expression of the mean in Eq. (3.89). Another important property is *asymptotic consistency*. That is, the variance reduces to zero everywhere as the design space is sampled denser. A proof is given in Vazquez (2005) with the condition that the covariance $\sigma^2 R$ is regular (*i.e.* R(0) = 1). An example is shown in Figure 3.13 where the dataset size goes from 8 to 12 and 20. The corresponding confidence interval volume decreases accordingly and it is almost zero for the largest dataset. We should stress here that this margin shrinking is hardly achieved for high-dimensional problems.

3.4.5.2 The nugget effect

We have mentioned in the previous section that the Kriging prediction interpolates the training points. In the case of noisy data, we shall not require the prediction to be interpolating. For this purpose, the common practice is the introduction of the so-called *nugget effect*. This technique comes from geostatistics as well where the properties of the auto-correlation function (or more specifically the variogram) are suddenly modified at the origin due to the presence of a nugget. In fact, when there are, say gold nuggets in a region, the gold concentration abruptly changes at the boundaries of the nugget. This phenomenon is taken into account in geostatistics by



Figure 3.13: Illustration of asymptotic consistency: The variance decreases almost everywhere as the design size increases.

introducing a discontinuity at the origin of the sought variogram.

In practice, an independent noise U(x) is considered as an additional term of the stochastic process in Eq. (3.58):

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}) + U(\mathbf{x}), \qquad (3.93)$$

where U(x) is a zero-mean Gaussian variable with variance σ_{7}^{2} .

This actually corresponds to adding a Dirac auto-correlation function. The corresponding auto-covariance is the unit matrix, leading to the following modified expression of the auto-covariance function of Y(x)

$$\boldsymbol{C} = \sigma^2 \boldsymbol{R} + \sigma_{\zeta}^2 \boldsymbol{I}_n \tag{3.94}$$

where *C* is the covariance matrix defined by $C_{ij} = \text{Cov} \left[\mathbf{x}_i, \mathbf{x}^{(j)} \right]$ and I_n is the identity matrix of size $n \times n$.

Since the Dirac distribution has all its mass concentrated at the origin, the nugget effect translates by the discontinuity at the origin of the auto-covariance.

An application example is given in Figure 3.14. A Gaussian noise of variance $\sigma_{\zeta}^2 = 0.1^2$ is added to the outputs of the training points. The nugget estimation is then enabled while building the Kriging model. As expected, the approximation is no longer interpolating. Besides, the Kriging variance does not go to zero at the training points. Instead, it is equal to σ_{ζ}^2 . The estimated nugget effect in this example is close to 0.01.

However, this needs not be the case all the time. In fact, for the nugget to be well estimated, the design of experiments must be dense. In the applications we are interested in for this thesis, we will not face such settings. The design of experiments will most likely be scarce. So our use of the nugget will be for a different purpose. More specifically, the introduction of the nugget generally smooths the optimization problem of the MLE and thus enables a better prediction. In this case, the nugget is extremely small and behaves more like a ridge.



Figure 3.14: Kriging with nugget estimation for the prediction with noisy observations.

3.5 Conclusion

In this chapter, we have introduced two types of surrogate models for supervised learning, namely support vector machines and Kriging. SVM has been developed around the structural learning theory. It offers good generalization ability, especially in high dimension. Kriging has been developed earlier in a different framework assuming that the true model is a realization of a Gaussian stochastic process.

In this thesis, we are concerned primarily with regression problems. However, classification is also of interest to us. An application is given in Chapter 5. Typically, we seek to classify different crash scenarios. For this purpose, we use support vector machines for classification.

On the other hand, the constrained lightweight design application, which is the core of this thesis, will rely upon Kriging. The reasons for this choice are two-fold. First, Kriging provides us with a local measure of its own accuracy thanks to the Gaussian assumption. This error is due to the epistemic uncertainty raised by the substitution of the true model with a metamodel trained over a limited set of samples. It can be locally reduced by adding points in the design space. As a matter of fact, a wide number of techniques have been developed in the literature to take advantage of this feature by adaptively updating the design of experiments. The methodology for optimization through surrogate models, proposed in the next chapter, is based on such a technique. Second, we have carried out a benchmark analysis of Kriging and support vector regression, with application on crashworthiness design (Moustapha et al., 2014a). It is shown that isotropic Kriging and support vector regression approximately give the same level of accuracy. However, the introduction of anisotropy in Kriging dramatically improves the accuracy of the surrogate model leading to an overall better prediction with Kriging.

These two reasons advocate for the choice of Kriging over support vector regression. We should

however note that there are several works which aim at deriving a probabilistic interpretation for SVR, see for example Gao et al. (2002); Lin and Weng (2004). So the methodology proposed in the next chapter may also be applicable with SVR, provided that its probabilistic version behaves as well as Kriging.



SURROGATE-BASED OPTIMIZATION

Contents

4.1	Introduction	
4.2	2 Global optimization using surrogate models	
	4.2.1	Problem formulation
	4.2.2	The augmented reliability space
	4.2.3	Analytical validation example
4.3	Adaptive design of experiments	
	4.3.1	A word on initial design of experiments
	4.3.2	Using a surrogate model for the objective function
	4.3.3	Using surrogate models for the constraints
	4.3.4	Multiple points enrichment 121
	4.3.5	Multi-constraints handling 124
4.4	RBDO with surrogate model and enrichment	
	4.4.1	Introduction
	4.4.2	Contour refinement with quantile estimation
4.5	RBDO with two stages of enrichment	
	4.5.1	Motivations and first stage of enrichment
	4.5.2	The second stage of enrichment
	4.5.3	Validation example
4.6	Conclusion	

4.1 Introduction

In the first part of this thesis we have introduced on the one hand, a methodology for optimization under uncertainty and, on the other hand the concept of surrogate modeling. In this chapter, the two items are combined for an efficient reliability-based design optimization. The motivations are the following. First, the models used to evaluate the constraints of mechanical systems are usually expensive. Second, optimization techniques require repeated evaluations of these models. This becomes even worse for reliability-based design optimization as the reliability assessment relies as well on numerous model evaluations. The overall number of calls to the model may in this case reach unreasonably high values (*i.e.* in the range of thousands to hundreds of thousands). For instance, let us consider the lightweight design of automotive structures which concerns us here. The validity of a design depends on some performance criteria in frontal impact. The corresponding constraints are assessed by a fast dynamic simulation through finite element methods. The resulting analysis may last up to 24 hours in distributed CPUs. Direct optimization is in this case simply not affordable.

In the past few years, surrogate modeling has emerged as an interesting tool to lessen the computational burden of analyses such as optimization and reliability. Indeed, by offering to replace expensive models by easy-to-evaluate mathematical functions, the use of metamodels took the potential applications of RBDO to the realm of industrial problems. Several such applications exist in the literature. Glaz et al. (2008) used Kriging as a surrogate to reduce the vibrations induced by an helicopter rotor blades during forward flight. In Bellary et al. (2015), computational fluid dynamics were considered for the optimization of turbo-machinery systems. This high fidelity model, which costs 13 hours of CPU, was replaced by blind Kriging for efficient optimization. In a context close to our application, *i.e.* the design of automotive body structures under safety constraints, several applications can be found as well. Su et al. (2011) considered polynomial response surfaces and genetic algorithms to optimize a bus frame under rollover constraints. Side-impact was considered in Xu et al. (2013). In most of these applications, the authors first performed a benchmark of several metamodels and then selected the best one to perform optimization. In contrast, Gu et al. (2015) considered an ensemble of metamodels for the design of occupant protection systems. In this work, the authors consider simultaneously Kriging, support vector regression, radial basis functions and polynomial response surfaces. The four are weighted following a heuristic based on cross-validation error. Frontal impact has also been investigated. Gu et al. (2013) compared multi-objective deterministic, robust and reliability-based design optimizations. In this respect, they used polynomial response surfaces for the objective function and radial basis functions for the constraints.

In this chapter, the frontal impact simulations are replaced by surrogate models for safe design of automotive body structures. The first section formulates the problem combining surrogate models and quantile-based RBDO proposed in section 2.3. In the afore-mentioned references, optimization with surrogate models was one-shot *i.e.* the metamodel is built once for all and optimization is performed on it. However, it has been shown that the computational cost can be further reduced by adopting *adaptive design of experiments* (Jones et al., 1998; Bichon et al., 2008). The second section reviews such methods. In the last section, we present a methodology for RBDO based on surrogate models and enrichment for the estimation of quantiles.

4.2 Global optimization using surrogate models

4.2.1 Problem formulation

In principle, the substitution of a high-fidelity model by a metamodel is straightforward to implement. Let us consider a surrogate model denoted $\widehat{\mathcal{M}}_k$ which approximates the mechanical model \mathcal{M}_k . A global approximation of the limit state surface simply reads:

$$\widehat{\mathfrak{g}}_{k}\left(\boldsymbol{x}\right) = \overline{\mathfrak{g}}_{k} - \widehat{\mathcal{M}}_{k}\left(\boldsymbol{x}\right). \tag{4.1}$$

By this simple trick, computing the failure probability with respect to the approximate limit state surface becomes cheap. The use of surrogate models to compute failure probabilities has received much attention in the past few decades. Artificial neural networks have been successfully used by Chapman and Crossland (1995), Papadrakakis and Lagaros (2002) or Hurtado and Alvarez (2003). Gayton et al. (2003) considered quadratic polynomial response surfaces for the same task. Sparse polynomial chaos have also been used (Blatman, 2009; Blatman and Sudret, 2008, 2010; Hu and Youn, 2011). Hurtado and Alvarez (2001) proposed to rely on classifiers rather than regression models by introducing support vector machines to structural reliability analysis. Basudhar and Missoum (2009) also use SVM to approximate random fields in the context of reliability analysis. In the aforementioned references, the failure probability is computed by Monte Carlo sampling. To achieve variance reduction while using SVM, subset sampling is considered in Bourinet et al. (2011) or Deheeger (2008) and importance sampling in Hurtado (2007). Finally the three simulations methods have been combined with Kriging such as in Echard et al. (2011); Picheny et al. (2010) for MCS, Bichon et al. (2008); Dubourg et al. (2011, 2013); Balesdent et al. (2013); Cadini et al. (2015) for IS.

In this thesis, we focus on Monte Carlo sampling to estimate quantiles of the constraints. The RBDO formulation in Eq. (2.93) with respect to the surrogate model therefore reads:

$$\boldsymbol{d}^{*} = \arg\min_{\boldsymbol{d}\in\mathbb{D}}\mathfrak{c}(\boldsymbol{d}) \quad \text{subject to:} \quad \left\{ \begin{array}{l} \mathfrak{f}_{j}(\boldsymbol{d}) \leq 0, & \{j=1,\ldots,n_{s}\}\\ \widehat{Q}_{\alpha_{k}}\left(\boldsymbol{d};\widehat{\mathcal{M}}_{k}\left(\boldsymbol{X}\left(\boldsymbol{d}\right),\boldsymbol{Z}\right)\right) \leq \tilde{\mathfrak{g}}_{k}, \quad \{k=1,\ldots,n_{h}\} \end{array} \right., \tag{4.2}$$

where \hat{Q}_{α_k} is the function which computes the quantile according to the metamodel.

As a reminder, the quantile for the model $\widehat{\mathcal{M}}_k$ is computed by first sampling the following Monte Carlo population of size *N*:

$$\mathfrak{C}_{q}\left(\boldsymbol{d}\right) = \left\{\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \dots, \boldsymbol{w}^{(N)}\right\},\tag{4.3}$$

where $\boldsymbol{w}^{(i)} = \left\{ \boldsymbol{x}^{(i)}, \boldsymbol{z}^{(i)} \right\}^{T}$. $\boldsymbol{x}^{(i)}$ and $\boldsymbol{z}^{(i)}$ are respectively the *i*-th realizations of $\boldsymbol{X} \sim f_{\boldsymbol{X}|\boldsymbol{d}}$ and $\boldsymbol{Z} \sim f_{\boldsymbol{Z}}$.

Second, the response of the surrogate models at these points is computed, thus giving the set:

$$\widehat{\mathcal{Y}} = \left\{ \widehat{y}^{(1)}, \widehat{y}^{(2)}, \dots, \widehat{y}^{(N)} \right\},$$
(4.4)

where by definition $\widehat{y}^{(i)} = \widehat{\mathcal{M}}_k(w^{(i)})$. The estimated quantile is eventually given by:

$$\widehat{q}_{\alpha_k} = \widehat{y}_{\lfloor (N\alpha_k) \rfloor}, \tag{4.5}$$

99

where $\widehat{y}_{(i)} \in \widehat{\mathcal{Y}}$ is defined such that $\widehat{y}_{(1)} \leq \widehat{y}_{(2)} \leq \ldots \leq \widehat{y}_{(N)}$ and $\lfloor \bullet \rfloor$ denotes the floor function.

The efficiency of the RBDO is enhanced here by introducing the so-called *common random numbers* (CRN) technique to build the set \mathfrak{C}_q for different designs. That is, the same seed is used to generate all random variables throughout the optimization. In particular, the same realizations of the environmental variables are considered in all iterations. This results in more stability for optimization and also reduces the variance of the estimated quantiles since the induced error is consistent within different designs. The bias in the solution is however kept small as the size of the Monte Carlo population is relatively large.

Eventually, to achieve accuracy in estimating quantiles through surrogate models, it is necessary that any point from the Monte Carlo population \mathfrak{C}_q falls within the definition space of the surrogate model. When considering optimization in a deterministic setting, this space of definition may obviously be set as the design space. However for optimization under uncertainties, we need to build the surrogate model in an augmented space to account for random variables that come in addition to the design parameters.

4.2.2 The augmented reliability space

4.2.2.1 A brief literature review

In this section, we define how to build the surrogate model for application in RBDO. As a premise, let us consider a double-loop approach coupled to the FORM/SORM philosophy. In FORM/SORM, for each design, a local approximation of the limit state surface is built in the standard Gaussian space after an appropriate mapping from the physical space. Applying the same strategy for surrogate-based RBDO consists in building a new surrogate model in the standard Gaussian space at each iteration of the optimization. This would require a large number of evaluations of the expensive true model. The overall added value of using the surrogate model would therefore be lost. More efficient approaches have been defined in the literature where a unique space which accounts for both the deterministic and the random variables is considered. Such a space is referred to as the *augmented reliability space*, but endows a variety of interpretations as reviewed in the sequel.

Kharmanda et al. (2002) defines a hybrid formulation as an alternative to the nested optimization problem in classical RBDO. In this hybrid formulation, a new cost function is defined as the product of the initial cost and the image of the reliability index in the physical space. This cost is minimized under the deterministic soft and hard constraints. As defined, the new cost is directly expressed with respect to the deterministic design variables (original cost) and random variables (reliability index). The hybrid design space therefore consists of the tensor product of the design and random variables $\{d, w\}^T$, where it is assumed that $W = \{W_1, W_2, \dots, W_s\}$ is a set of uncertain parameters whose joint PDF is $f_{W|d}$. This allows the authors to carry out simultaneously the optimization task and reliability analysis, thus reducing the computational effort. However, a major drawback is that the dimension of this hybrid problem is higher and this ultimately increases the complexity of the problem to solve. This affects the efficiency of the optimization algorithm and hampers the building of an accurate surrogate model. Eventually, as argued in Dubourg (2011), there is no bijectivity between the space of the random variables and that of the design variables. That is, a particular sample $w^{(i)}$ may have been equally generated by multiple values of design variables *d*. It is not clear how this affects the results provided by the approach.

Alternatively, Au (2005) has introduced an *augmented reliability problem* to efficiently compute the failure probability for a given design. The basic idea of the proposed approach is to artificially consider the design parameters as random. By means of conditional sampling and Bayes' theorem, the failure probability given a realization of design variable can be computed. The authors rely on subset sampling to achieve this. Taflanidis and Beck (2008) constructs a stochastic optimization algorithm based on this definition of the augmented reliability problem. In this way, RBDO does not suffer from increased dimension.

Adopting the same philosophy as in Taflanidis (2007) and Taflanidis and Beck (2008, 2009), Dubourg (2011) also suggests to solve the RBDO problem in an augmented space. In this work, the design and environmental variables are treated separately resulting in an augmented random vector $W = \{X, Z\}^T$. For each type of parameters, a *confidence region* is defined such that samples with extreme values (*i.e.* design variables at the boundary of the design space or realization of environmental variables away from their mean values) fall within this confidence region. For design variables, this region corresponds to a hyperrectangle whose bounds are defined by extending the design space with multi-dimensional confidence intervals according to $f_{X|d}$. The confidence region for the environmental variables covers a hypersphere in the standard Gaussian space whose radius corresponds to a high reliability index, say $\beta = 8$. An inverse isoprobabilistic transform allows one to map this confidence region in the physical space. In the general case, the associated volume is not regular. This implies that the points should be sampled uniformly in the standard Gaussian space in order to build the surrogate model. The augmented reliability space is eventually given by the tensor product between the confidence regions of the two types of parameters.

4.2.2.2 The proposed augmented space

The approach we adopt in this thesis is similar to that of Dubourg (2011) or Dubourg et al. (2011) in the sense that we also consider the augmented space as a tensor product between two confidence regions defined on the design and environmental variables. However, we do not rely on a hypersphere for the environmental variables. This is essentially because we do not work in the standard Gaussian space. The augmented space is thus hyperrectangular. Another important point here is that the normalization often needed to build surrogate models is simply achieved by a linear mapping. There is no need of non-linear isoprobabilistic transform which may add complexity to the surrogate model surface.

The basic idea underlying the use of confidence region is that we need to make sure that the probability of sampling a point outside the space over which the surrogate model has been built remains small. By defining a threshold on this probability, the confidence region may be obtained by quantiles of the parameters distributions. By additionally assuming that the parameters are mutually independent, the augmented space may be written as a tensor product of unidimensional intervals.

The confidence regions associated to the design and environmental variables are respectively denoted by X and Z. The former defines a hyperrectangular volume obtained by the following tensor product:

$$\mathbb{X} = \prod_{i=1}^{s_d} \left[q_{d_i}^-, q_{d_i}^+ \right],$$
(4.6)

where s_d is the number of design parameters and $q_{d_i}^-$ and $q_{d_i}^+$ are quantiles defined respectively

on the lower and upper bounds of the design variable d_i . These quantiles read as follows:

$$\begin{aligned} q_{d_i}^- &= F_{X_i|d_i}^{-1} \left(\alpha_{as}/2 \right) \\ q_{d_i}^+ &= F_{X_i|d_i}^{-1} \left(1 - \alpha_{as}/2 \right), \end{aligned} \tag{4.7}$$

where X_i follows the marginal distribution $f_{X_i|d_i}$ whose associated joint PDF $f_{X|d}$ has been already defined, $F_{X_i|d_i}^{-1}$ is the associated inverse CDF, d_i^- and d_i^+ are respectively the lower and upper bounds of the design variable d_i , and α_{as} is the probability of sampling outside the augmented space.

Similarly, the hyperrectangular volume associated to the environmental variables is defined by:

$$\mathbb{Z} = \prod_{i=1}^{s_z} \left[q_{z_i}^-, q_{z_i}^+ \right], \tag{4.8}$$

where s_z is the number of environmental variables and the bounds are defined by:

$$q_{z_i}^{-} = F_{Z_i}^{-1} \left(\alpha_{as}/2 \right),$$

$$q_{z_i}^{+} = F_{Z_i}^{-1} \left(1 - \alpha_{as}/2 \right).$$
(4.9)

Here Z_i follows the marginal distribution f_{Z_i} as well.

These quantiles may be obtained analytically for usual PDFs. In this thesis, we consider a confidence interval equivalent to 6σ for normal distributions which corresponds to setting $\alpha_{as} = 0.0027$. For distributions other than the Gaussian, the quantiles may be computed analytically or by an iso-probabilistic transform. In the most general case, where no analytical expression is available, the quantiles may be estimated by a large Monte Carlo simulation.

For illustration, let us consider the example introduced in section 2.3.3.1 where a two-dimensional problem consisting of a random design variable $X \sim \mathcal{N}(d, 0.05^2)$ and a random environmental variable $Z \sim \mathcal{N}(0.5, 0.05^2)$ is considered. The design space is defined by $\mathbb{D} = [-1, 1]$. The bounds therefore read:

$$\begin{aligned} q_d^- &= \mu_{X|d^-} - 3\sigma_{X|d^-} = -1 - 3 \times 0.05 = -1.15, \\ q_d^+ &= \mu_{X|d^+} + 3\sigma_{X|d^+} = 1 + 3 \times 0.05 = 1.15, \\ q_z^- &= \mu_Z - 3\sigma_Z = 0.5 - 3 \times 0.05 = 0.35, \\ q_z^+ &= \mu_Z + 3\sigma_Z = 0.5 + 3 \times 0.05 = 0.65. \end{aligned}$$

Figure 4.1 illustrates this example. The augmented space corresponds to the gray-shaded area. The design space over which the optimization problem is defined is simply the line whose bounds are d^- and d^+ , highlighted in blue. It should be stressed here that $f_{X|d^-}$ and $f_{X|d^+}$ may have different standard deviations. This situation typically occurs when the distribution of the random variables is expressed in terms of a constant coefficient of variation.

In general, the augmented space is constructed according to the probabilistic model. The example above consists of both random design and environmental variables. Alternatively, it may happen that the randomness is considered only for the design variables. In this case the augmented space is simply X since the environmental variables have fixed values. Conversely, we may have randomness in environmental variables only. In such a case, the augmented space is $\mathbb{D} \times \mathbb{Z}$. This corresponds to a particular setting where the design variables are considered random but with standard deviations equal to 0, thus leading to $q_{d_i}^{\pm} = d_i^{\pm}$.



Figure 4.1: Illustration of the augmented space for a two-dimensional problem with both random design and environmental variables. Directions x and z do not have the same scale for the sake of clarity. Distances are somehow stretched along z.

4.2.3 Analytical validation example

4.2.3.1 Presentation of the problem

To illustrate the quantile-based RBDO with surrogate models, we consider a two-dimensional highly-non linear limit state surface, which we refer to as the Haupt function (Dubourg, 2011). The deterministic design optimization problem reads:

$$d^* = \arg\min_{d \in \mathbb{D}} (d_1 - 3.7)^2 + (d_2 - 4)^2 \quad \text{s.t.} \quad \mathfrak{g}(d) = -d_1 \sin(4d_1) - 1.1 \sin(2d_2) \le 0.$$
(4.10)

The RBDO problem is formulated by considering that the design variables follow independent normal distributions: $X_i \sim \mathcal{N}(d_i, 0.1^2)$. A reliability index $\bar{\beta} = 2$ corresponding to a target failure probability $\bar{P}_f = \Phi(-\bar{\beta}) = 0.0228$ is considered.

Figure 4.2 illustrates the optimization problem. The left panel shows the highly-non linear mathematical model. In the right one, the limit state surface is shown in red and contours of the cost are shown in a gray-scale plot. The design space $\mathbb{D} = [0, 3.7] \times [0, 4]$ is bounded by the blue rectangle. The augmented space in this example only consists of an extension of \mathbb{D} and corresponds to $\mathbb{X} = [-0.3, 4] \times [-0.3, 4.3]$ as highlighted by the gray rectangle.

To solve this problem we consider first a brute-force approach *i.e.* directly relying on the true model. Then we replace the model by both support vector regression and Kriging approximations. More specifically, L_1 -SVR and anisotropic Kriging are considered with respectively Gaussian and Matérn 5/2 auto-correlation functions.

4.2.3.2 Results

The brute-force approach leads to an optimum $d_{ref}^* = \{2.83, 3.24\}^T$ which is considered as the reference solution. Computing the quantile for each design requires $N = 10^4$ samples. Embedded in an optimization scheme, the total number of calls to the true model is in the order



Figure 4.2: Illustration of the Haupt problem

of $10^6 - 10^7$ (*i.e.* considering 100 to 1,000 iterations of the (1 + 1)-CMA-ES optimization algorithm). For the surrogate-based approach, we only need a few sample points that will be used to build the metamodels. The optimization will be performed on these approximations afterwards. In this example, we consider five cases where the size of the design of experiments is increased from 20 to 100 by steps of 20. Since, these points are sampled randomly, the experiment is repeated ten times for each case. Results are shown as box-plots in Figure 4.3 where the reference solution is represented by the dashed black line. The surrogate models built with less than 60 points for SVR and 40 points for Kriging clearly lead to spurious optima. This is because they fail to capture the high non-linearity of the mathematical model. Kriging is more accurate on this example thanks to the anisotropic assumption. For the remaining cases, the found optima match the reference solution found through the brute-force approach. However, the number of calls to the true model is drastically reduced, *i.e.* only 80 and 100 calls respectively. This is in strong contrast with the $10^6 - 10^7$ calls required by the brute-force approach and thus motivates the need to recourse to surrogate models.

To illustrate the differences between the converged and failed solutions, we plot the true and surrogate models with 20 and 60 points in Figure 4.4. On the right panel, the limit state surface is almost accurately approximated by Kriging. For SVR, it turns out that it is not enough accurate in the region where the optimum lies. From 80 points and more, both metamodels perfectly match the limit state surface. However, in the left panel with only 20 points, the surrogate models are far from representing the limit state surface. This means that the substitution of the true model works fine only if the surrogate model is capable of truly approximating the limit state surface. This is not systematic as shown in this example. Thanks to the asymptotic consistency of both Kriging and SVR, it was expected that they become more accurate as the number of samples increases. However, one does not know in advance how many points are needed to achieve the required accuracy. Anyway, this number depends on how the experiments are chosen to fill the input space of the surrogate model. Many techniques exist to construct such design of experiments. We review some of them in the first part of the next section. It is also possible to go further by requiring the surrogate model to be accurate only in regions that matter as it was the case with the SVR model of 60 points in Figure 4.4b. The second part of the



Figure 4.3: Solution of the Haupt problem with different sizes of the DOE.

next section focuses on some techniques that allow one to achieve this objective.

4.3 Adaptive design of experiments

As shown in the example above, the validity of the optimum found with surrogate-based optimization completely relies on the accuracy of the surrogate model. In a naive approach, one would fill the space as densely as possible to achieve the required accuracy. However as already mentioned, the evaluation of the true model may be expensive for industrial problems (recall our finite element run that lasts 20 hours). In such a case, each point in the design of experiments should be carefully chosen. So-called *space-filling* designs allow one to cover the space as much as possible with the smallest number of points. In this section, we review some of them. Moreover, other techniques allow one to further reduce the number of calls to the true model. These techniques start from the premise that for optimization there is no need to



Figure 4.4: Approximation of the true limit state surface by SVR and Kriging (KRG): Design of experiments of 20 and 60 points.

be accurate in the entire design space since there are only a few regions that are of interest to the designer (*i.e.* those where the constraints are closed to be violated and where the cost function decreases). These techniques iteratively update an initial scarce design of experiments to improve the metamodels accuracy in regions of interest. They are known as *adaptive design of experiments*. The remaining part of this section is devoted to such techniques.

In this section, the input parameters are denoted by x. For the sake of consistency, there is no reference to the random design variables introduced above. We consider in fact that we are in a deterministic setting.

4.3.1 A word on initial design of experiments

The first step in building surrogate models for optimization is to generate a design of experiments (DOE). In the literature, many techniques exist to build DOEs. They can be classified into three groups:

- *Deterministic designs*: The design sites are chosen according to predefined schemes. Wellknown examples include full factorial or geometric criteria based-designs (*i.e.* maximin and minimax).
- *Random designs*: They come from Monte Carlo techniques where points are randomly drawn in the unit hypercube. Latin hypercube sampling, a stratified method, is one of the most widely-used among this category. To improve their ability to uniformly fill the space, initial sampled random designs can be optimized with respect to some predefined criteria.
- *Quasi-random designs*: They consist of well-chosen deterministic sequences whose discrepancy (*i.e.* departure from uniform sampling) is small. Well-known examples are Sobol' or (t, m, s)-nets.

Appendix A presents in details a wide variety of techniques used to build designs of experiments. They perform almost equally in a low-dimensional setting when one can afford a relatively large number of samples with respect to the dimension. In contrast, when one is concerned with high-dimensional problems, some of these techniques are less relevant than the others. This might be for instance due to the high number of points required to truly fill the space *e.g.* factorial designs or Monte Carlo sampling. Another issue is the computational effort such as for those relying on the optimization of some metrics (e.g. minimax, Bayesian designs or OLH). The question of computational time is solved by quasi-random designs. Lowdiscrepancy sequences can indeed be easily built thanks to pre-defined deterministic sequences. However, the sequences fill the space in cycles whose length depends on the basis of the *b*-adic expansion, henceforth conditioning the size of the design. Most of the applications in this thesis are based either on Sobol' designs which are straightforward to build or on optimized Latin hypercube with respect to L₂-discrepancy. The generation of the latter may take time for highdimensional problems. But we argue that in our case, where one simulation takes as long as 24 hours, the relative time used to build the design, which amounts in minutes only, is not a concern.

For the sake of completeness, let us now mention briefly some specificities of the problem we aim to solve. The characteristics of a high-dimensional space are actually extremely different from its low-dimensional counterpart (Lee and Verleysen, 2007; Verleysen, 2003). Its properties are counter-intuitive and the term curse of dimensionality, first coined by Bellman (1961) expresses well what is happening. The main property is the empty space phenomenon (Scott and Thompson, 1983) which refers to the fact that high-dimensional data are inherently sparse. For instance, consider filling an s-dimensional space with a density similar to that of filling a onedimensional space with 10 points. One would need actually 10^s points to reach such a density, which is clearly not affordable. Furthermore, most of the volume of high-dimensional hypercubes (*i.e.* as those in which we sample and build surrogate models here) are concentrated around their vertices. This can be seen by considering the ratio between a hypersphere and its corresponding circumscripted hypercube. It can be shown that this ratio tends to zero as the dimension increases. This means that all the volume is located away from the center. As pointed out in Verleysen (2003), hypercubes in high-dimension look like sea urchins where the spikes occupy all the volume letting only a negligible part to the spherical body. Another property is the concentration of norms and distances. It can be easily shown that when sampling in high dimensions all the points are equally distant. All these properties seem unexpected because of the intuitions we have made over two- and three-dimensional representations of data. Nonetheless they make the task of learning from high-dimensional data difficult. Introducing simplifying assumptions such as smoothness of the model may attenuate these effects. Beside, in physical systems, only a few number of parameters are relevant. One can therefore use methods such as feature selection (e.g. screening) or feature extraction (e.g. principal component analysis).

We assume for the lightweight design of automotive body structures that an empirical screening study has been made thanks to expert knowledge, so that the remaining parameters we use for optimization are the most relevant. Furthermore, instead of trying to fill the space, we rather rely on adaptive techniques which allow us to incorporate further knowledge of the problem by directing the sampling to restricted regions of interest.

4.3.2 Using a surrogate model for the objective function

This section is concerned with the adaptive enrichment of an initial design. As mentioned earlier, we aim at developing metamodels as surrogates of expensive-to-evaluate functions. These metamodels are used for constrained optimization problems. The previous section presented some space-filling designs which have the property of fairly exploring the entire design space. However, the optimization problem we would like to address is high dimensional and filling the design space would require a non-affordable number of simulations of the true expensive models. The question that arises then is whether it is really necessary to evenly span the data in the entire design space. There are certainly regions of the space which are of more interest than the others, say those where the objective function is optimal. Many researchers have lingered on this question leading to several *infill sampling criteria* (Sasena, 2002). The idea is to start with a fair initial design, build the surrogate model and then iteratively update the design according to various criteria so as to improve the metamodel accuracy. The sampling criteria are derived according to the designers aim *i.e.* whether it is the objective or the constraint functions that are approximated. We are concerned with the latter case. However, the enrichment criteria have been originally developed for the former. For the sake of clarity, we first introduce sampling criteria for the minimization w.r.t. a surrogate model.

4.3.2.1 Sampling at the current minimum

The most obvious approach is to consider checking the validity of the solution a posteriori, *i.e.* after an optimal solution has been found. The basic idea is to evaluate the model on the current point found by the optimizer and if it is not accurate enough, then add this point so as to locally improve the metamodel accuracy. The following steps achieve this in an iterative scheme:

- 1. Sample an initial design;
- 2. From the current design, build a surrogate model;
- 3. Find the minimum of this surrogate model;
- 4. Compute the true function at the location of this minimum;
- 5. Add this new point at the current design and go to step 2, unless a predefined convergence criterion is met.

This simple approach may probably work for some functions. However, if the initial metamodel is too inaccurate, it may converge in a local optimum.

Figure 4.5 illustrates the point. The function $\mathcal{M}(x)$ defined by equation Eq. (2.94) is minimized in $\mathbb{X} = [-1, 1]$. An initial sample of six points is randomly drawn in \mathbb{X} . This figure shows that updating the dataset with the current solution found by the metamodel leads to a local minimum. This is due to the poor quality of the initial response surface which mislead the overall optimization.

This shortcoming may be avoided by considering the epistemic uncertainty associated to the metamodel. Using a Kriging model, we show in Figure 4.6 the initial mean prediction of the Kriging approximation and its associated 95% confidence interval (CI). It can be observed on the left panel that beside the neighborhood of the local minimum, the leftmost region is associated with high uncertainty due to the sparsity of samples there. Considering this uncertainty, this region, which actually contains the global minimum, would be a good searching candidate. However by the above-strategy, the successive enrichments have helped reduce the uncertainty



Figure 4.5: Minimization of a function through a metamodel. The updating of the design by the minimum of the metamodel leads to a local minimum.

around the local minimum without exploring the region with the global minimum (Figure 4.6b).



Figure 4.6: Kriging approximation with its 95% confidence intervals (CI) before (left) and after enrichment (right).

This shows the necessity of considering a probabilistic framework while optimizing through a response surface. The criteria presented in the sequel make use of it.

4.3.2.2 Probabilistic framework for surrogate-based optimization

As shown in the previous example, simply relying on a model approximation is not enough when one cannot fully trust the prediction given by the surrogate model. One must therefore account for potential discrepancies between the true and surrogate models. Such information is not readily available at the time of optimization. However, using metamodels such as Kriging, one may rely on the prediction variance which gives an idea on where the model might lack of accuracy. Jones et al. (1998) have introduced the so-called *expected global optimization* (EGO) which is based on this concept. EGO seeks to solve the surrogate-based optimization problem by balancing the search strategy to regions where the surrogate model is minimized (exploitation) and those where its variance is high (exploration). This strategy allows one to achieve a global search and avoid the pitfall illustrated in the previous example. Many algorithms that exploit the same idea can be found in the literature. We present some of them below.

Lower confidence bound

The most natural way to include model epistemic uncertainty is to consider the lower bound of the confidence interval (Cox and John, 1997), though it is not the historical first approach (Kushner (1964) had earlier introduced a criterion based on the so-called *probability of improve-ment*). Considering a Kriging predictor of mean $\mu_{\widehat{\mathcal{M}}}(x)$ and variance $\sigma_{\widehat{\mathcal{M}}}^2(x)$, the updating formula is given as follows:

$$\mathbf{x}_{\text{next}} = \arg\min_{\mathbf{x}\in\mathcal{K}}\mu_{\widehat{\mathcal{M}}}(\mathbf{x}) - \kappa\sigma_{\widehat{\mathcal{M}}}(\mathbf{x}), \tag{4.11}$$

where κ is a coefficient to set.

This way, regions with current minimum and high uncertainty are balanced. A high value of κ puts more emphasis on the region with high uncertainty whereas a low value concentrates the search toward the local minimum. In the previous example of Figure 4.6, taking $\kappa = 1.96$ would result in searching lower bound of the 95% confidence interval. It is clear that the algorithm would add samples in the regions of the global minimum and would converge. On the contrary, it is not sure that a too small value of κ would allow to spring out of the local minimum (*e.g.* $\kappa = 0.25$). Thus this approach shares the same drawback with Kuhsner's probability of improvement, *i.e.* somehow an improvement objective is fixed. To avoid choosing a parameter, Jones et al. (1998) came up with the so-called *Expected Improvement* function.

Expected improvement

An efficient way to take into account the uncertainty of the prediction is presented in Jones et al. (1998) and earlier in Mockus (1974). Consider an initial DOE $\mathcal{D} = \{(x_i, y_i), i = 1, ..., n\}$ and a Gaussian process predictor $\widehat{\mathcal{M}}$, the so-called *improvement* is defined as:

$$I(\mathbf{x}) = \max_{\mathbf{x} \in \mathbb{X}} \left(y_{min} - \widehat{\mathcal{M}}(\mathbf{x}), 0 \right), \qquad (4.12)$$

where $y_{min} = \min(y_i, i = 1, ..., n)$.

This improvement is a truncated positive Gaussian variable. The *expected improvement* is defined as its expectation:

$$EI(\mathbf{x}) = \mathbb{E}\left[I\left(\mathbf{x}\right)\right] = \int_{-\infty}^{y_{min}} (y_{min} - y)\varphi\left(\frac{y - \mu_{\widehat{\mathcal{M}}}(\mathbf{x})}{\sigma_{\widehat{\mathcal{M}}}(\mathbf{x})}\right) dy.$$
(4.13)

After integration by parts of Eq. (4.13), a closed form of the expected improvement can be derived (a proof is given in Ginsbourger (2009), p.109):

$$EI(\mathbf{x}) = \left(y_{min} - \mu_{\widehat{\mathcal{M}}}(\mathbf{x})\right) \Phi\left(\frac{y_{min} - \mu_{\widehat{\mathcal{M}}}(\mathbf{x})}{\sigma_{\widehat{\mathcal{M}}}(\mathbf{x})}\right) + \sigma_{\widehat{\mathcal{M}}}(\mathbf{x}) \varphi\left(\frac{y_{min} - \mu_{\widehat{\mathcal{M}}}(\mathbf{x})}{\sigma_{\widehat{\mathcal{M}}}(\mathbf{x})}\right).$$
(4.14)

110

The next point in the sample is thus defined as the one which maximizes EI(x). This function has two complementary parts. The first half is a product between the improvement and the probability that the point is lower than the current minimum. The second part rather grows when the Kriging variance $\sigma_{\widehat{M}}^2$ at a given point is high. Both regions are henceforth likely to be explored by the algorithm. The example in Figure 4.7 is plotted with the previously introduced initial design. The lower panels show the expected improvement as a function of x. The argument that maximizes them is chosen as the next point to add in the DOE. After a first update of the design near the local minimum, the algorithm explores the region of the space with high uncertainty and eventually converges towards the global minimum.



(c) Mean prediction together with its 95% CI: final

-0.5

(d) Mean prediction together with its 95% CI: final

x

Figure 4.7: Kriging approximation with its 95% confidence intervals (CI) before (left) and after enrichment (right).

Balance between minimum and variance

Instead of searching the minimum of the metamodel, it might be needed to rather seek to reduce the variance of the predictor. This can be accomplished by putting more emphasis on the variance of the predictor than on the current minimum improvement. In this way, one could direct the algorithm to local or global search. The so-called *Generalized Expected Improve*-

ment (*GEI*) provides an additional parameter to direct the optimization. It was introduced by Schonlau (1997), considering a positive integer *g* and reads:

$$I^{g}(\mathbf{x}) = \max_{\mathbf{x}\in\mathbb{X}} \left(\left(y_{min} - \widehat{\mathcal{M}}(\mathbf{x}) \right)^{g}, 0 \right).$$
(4.15)

Taking its expectation, the closed-form of the generalized expected improvement is obtained with the following recursive formula:

$$EI_{g}(\mathbf{x}) = \sigma_{\widehat{\mathcal{M}}}^{g} \sum_{k=0}^{g} (-1)^{k} \left(\frac{g!}{k!(g-k)!}\right) \left(\frac{y_{min} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right)^{g-k} T_{k}$$
with $T_{0} = \Phi\left(\frac{y_{min} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) \quad T_{1} = -\varphi\left(\frac{y_{min} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right)$
(4.16)
and for $k > 1$, $T_{k} = -\varphi\left(\frac{y_{min} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) \left(\frac{y_{min} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right)^{k-1} + (k-1) T_{k-2}.$

The dependence of $\mu_{\widehat{\mathcal{M}}}$ and $\sigma_{\widehat{\mathcal{M}}}$ on *x* has been left aside for the sake of clarity.

The parameter g is used to switch from local to global search and vice-versa. The case g = 1 corresponds to the standard expected improvement. The higher the integer g, the more emphasis is put on the variance. The challenging task is that of finding the value of g which switches the search strategy.

Many authors proposed heuristics to alternate the optimization between local and global search. Sasena (2002) presented in his PhD thesis a *cool criterion*. He started from the observation that too high a value of *g* would converge too slowly, whereas if *g* is chosen too small, the algorithm would overlook the global minimum. He then proposed a simulated annealing-like algorithm, *i.e.* the algorithm starts with a high value of *g* and reduces as the optimization progresses. Hence the early iterations globally seek to reduce the overall variance of the metamodel while the latter search more accurately around the identified local minima.

Jones (2001) used the probability of improvement criterion. Since the search strategy depends on the value of the target improvement, he proposed to compute the probability of improvement for many values of the target so as to find their maximum and then cluster them. This way, it is possible to add simultaneously many points in the design for both local and global searches.

This is actually a good way to make full use of computational capabilities. Indeed, in an industrial context (though not only), clusters of computers allow to launch in parallel many computations. It is therefore appealing to use this capacity to accelerate the optimization by adding simultaneously many points at each iteration. Ginsbourger et al. (2008) extended EI to multipoints by introducing the *q*-*EI*, where *q* points are simultaneously added to the design at each iteration. The authors developed a closed form for *q* = 2 using the uni- and bi-variate standard Gaussian CDF. For *q* \geq 3, the general expression is too complex for analytical solution so they resort to numerical strategies (Monte Carlo Simulation). An implementation is available in the R package DiceOptim.

4.3.3 Using surrogate models for the constraints

We have introduced above some sampling criteria originally used in EGO for optimization. The problem to solve is however slightly different in our case. In fact, our problematic is lightweight

design of vehicles under crashworthiness constraints. So the objective function is the weight of some parts of the body-in-white. With the inputs being parts thicknesses, the weight can be easily computed by simple algebra. Thus there is no need to substitute the objective function by a metamodel. On the other hand, the constraints are finite-element-based expensive functions and therefore require to be approximated by metamodels. Hence the EGO algorithm introduced above is not appropriate, since the response surfaces are not being optimized. The adaptive enrichment of the design would rather serve to improve the quality of the metamodel, and preferably on regions where the constraints are likely to be violated.

4.3.3.1 Adaptations of the expected improvement

Some authors have adapted the widely-used expected improvement to fit the problem of constraints handling. We review some of them below.

Adjusted expected improvement

Schonlau et al. (1998) introduced a method to deal with non-linear constraints in optimization. It consists of multiplying the expected improvement by the probability that a constraint is respected. The problem may be cast as:

$$EI_{a}(\boldsymbol{x}) = EI(\boldsymbol{x}) \prod_{k=1}^{n_{h}} \mathbb{P}\left(\widehat{\mathcal{M}}_{k}(\boldsymbol{x}) \leq \bar{\mathfrak{g}}\right), \qquad (4.17)$$

where \mathbb{P} is a probability measure associated to the epistemic uncertainty of the surrogate model, herein Kriging. It must not be confused with the probability measure \mathcal{P} introduced earlier in this manuscript and which rather accounts for the randomness in input parameters of the model.

The next sampling point is defined as the one that maximizes $EI_a(\mathbf{x})$. Hence, when a sample is likely to violate the constraint, the multiplication of $EI(\mathbf{x})$ by the low probability $\mathbb{P}\left(\widehat{\mathcal{M}}_k(\mathbf{x}) \leq \overline{\mathfrak{g}}\right)$ makes $EI_a(\mathbf{x})$ small. This method does not really eliminate the possibility of sampling an unfeasible point, it just reduces its likelihood. Since it is involved in an iterative scheme, the method would eventually converge to a feasible global optimum. As for our problem, keeping the expected improvement on a perfectly known function is a waste even if the constraints are handled.

Expected violation

Audet et al. (2000) proposed a method for constraints handling which relies on the so-called *expected violation* (EV). It is an adaptation of the expected improvement to constraints violations and is defined as:

$$EV_{k}(\mathbf{x}) = \begin{cases} (\bar{\mathbf{g}} - \mu_{\widehat{\mathcal{M}}_{k}}(\mathbf{x}))\Phi\left(\frac{\bar{\mathbf{g}} - \mu_{\widehat{\mathcal{M}}_{k}}(\mathbf{x})}{\sigma_{\widehat{\mathcal{M}}_{k}}(\mathbf{x})}\right) + \sigma_{\widehat{\mathcal{M}}_{k}}(\mathbf{x})\varphi\left(\frac{\bar{\mathbf{g}} - \mu_{\widehat{\mathcal{M}}_{k}}(\mathbf{x})}{\sigma_{\widehat{\mathcal{M}}_{k}}(\mathbf{x})}\right) & \text{if } \sigma_{\widehat{\mathcal{M}}_{k}}(\mathbf{x}) > 0, \\ 0 & \text{if } \sigma_{\widehat{\mathcal{M}}_{k}}(\mathbf{x}) = 0. \end{cases}$$
(4.18)

where $\mu_{\widehat{\mathcal{M}}_k}(\mathbf{x})$ and $\sigma_{\widehat{\mathcal{M}}_k}(\mathbf{x})$ are respectively the mean and standard deviation of the prediction for the constraint k ($k = \{1, ..., n_h\}$).

By analogy with the expected improvement, EV is made of two contributions. The first part of the expression is low when the constraint $\mu_{\widehat{\mathcal{M}}_k}(\mathbf{x}) \geq \overline{\mathfrak{g}}$, *i.e.* when the constraint is likely to be violated. The second part is high when the uncertainty in the prediction grows.

Audet and his co-authors use this technique together with the traditional EI. They actually sample a set of points for which they compute the expected violations. Those points which respect predefined criteria on the EV are then selected and their corresponding EI's are calculated. Eventually, the *q* best points with respect to EI are selected as additional samples for the design update.

Constrained EGO formulation

Using the similarities with methods based on merit functions to solve constrained optimization problems, Bichon et al. (2009) introduced the so-called *constrained EGO formulation*. They actually use the augmented Lagrangian principle to transform the original optimization problem into a unique non-constrained problem:

$$\arg\min_{\mathbf{x}\in\mathbb{N}}\mathfrak{c}\left(\mathbf{x}\right)+\lambda\mathfrak{g}\left(\mathbf{x}\right)+r_{p}\left(\mathfrak{g}\left(\mathbf{x}\right)\right)^{2},\tag{4.19}$$

where $\lambda \ge 0$ is a Lagrange multiplier and $r_p \ge 0$ is a penalty coefficient.

Considering that $\mathfrak{c}(x)$ and $\mathfrak{g}(x)$ can be approximated by Gaussian processes, the idea is to introduce them in an EGO process with computation of the expected improvement. However, recall that the simple analytical formulation of the expected improvement has been derived only for Gaussian processes. In Eq. (4.19), the combination will not be a Gaussian process because of the square in the penalty term. So instead, Bichon et al. (2009) suggested the use of the expected violation function as constraints. Eq. (4.19) then becomes:

$$\arg\min_{\mathbf{x}\in\mathbb{N}}\mathfrak{c}\left(\mathbf{x}\right)+\lambda EV\left(\mathbf{x}\right)+r_{p}EV\left(\mathbf{x}\right)^{2},\tag{4.20}$$

where EV(x) is the expected violation function introduced in Eq. (4.18).

As EV(x) is deterministic, the objective function of Eq. (4.20) remains a Gaussian process. It is henceforth readily usable for the computation of the expected improvement.

Expected improvement for contour estimation

Ranjan et al. (2008) adopted an approach where priority is given to the accurate estimation of the contour $\{x \in \mathbb{X} : \widehat{\mathcal{M}}(x) = \overline{\mathfrak{g}}\}$. For this purpose, they introduced an improvement function which aims at selecting trials where the surrogate is in the vicinity of the level $\overline{\mathfrak{g}}$. It reads:

$$I_{\bar{\mathfrak{g}}}(\boldsymbol{x}) = \epsilon^{2}(\boldsymbol{x}) - \min\left(\left(\widehat{\mathcal{M}}(\boldsymbol{x}) - \bar{\mathfrak{g}}\right)^{2}, \epsilon^{2}(\boldsymbol{x})\right), \qquad (4.21)$$

where $\epsilon(\mathbf{x}) = \alpha_{\epsilon} \sigma_{\widehat{\mathcal{M}}}(\mathbf{x})$, with α_{ϵ} being a positive multiplier coefficient.

This improvement function allows one to sample points where the prediction is in the neighborhood of \bar{g} . To allow one to consider the uncertainty of the prediction as well, its expectation is computed and reads:

$$\mathbb{E}\left[I_{\bar{\mathfrak{g}}}\left(\boldsymbol{x}\right)\right] = \left[\alpha_{\epsilon}^{2}\sigma_{\widehat{\mathcal{M}}}^{2}\left(\boldsymbol{x}\right) - \left(\mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right) - \boldsymbol{z}\right)^{2}\right] \left[\Phi\left(\frac{\bar{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right)}{\sigma_{\widehat{\mathcal{M}}}^{2}\left(\boldsymbol{x}\right)} + \alpha_{\epsilon}\right) - \Phi\left(\frac{\bar{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right)}{\sigma_{\widehat{\mathcal{M}}}^{2}\left(\boldsymbol{x}\right)} - \alpha_{\epsilon}\right)\right] \\ + 2\left(\mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right) - \bar{\mathfrak{g}}\right)\sigma_{\widehat{\mathcal{M}}}^{2}\left(\boldsymbol{x}\right) \left[\varphi\left(\frac{\bar{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right)}{\sigma_{\widehat{\mathcal{M}}}^{2}\left(\boldsymbol{x}\right)} + \alpha_{\epsilon}\right) - \varphi\left(\frac{\bar{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right)}{\sigma_{\widehat{\mathcal{M}}}^{2}\left(\boldsymbol{x}\right)} - \alpha_{\epsilon}\right)\right] \\ - \int_{\bar{\mathfrak{g}} - \alpha_{\epsilon}\sigma_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right)}^{\bar{\mathfrak{g}} + \alpha_{\epsilon}\sigma_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right)}\left(\boldsymbol{y} - \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right)\right)^{2}\varphi\left(\frac{\boldsymbol{y} - \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}\right)}{\sigma_{\widehat{\mathcal{M}}}}\right)d\boldsymbol{y}.$$

$$(4.22)$$

114

The properties of this function are similar to the traditional expected improvement. The three terms of Eq. (4.22) have complementary aspects. The first term dominates when the prediction is close to the level $\bar{\mathfrak{g}}$. The second is rather high when the prediction is far from the ϵ -band defined by $I_{\bar{\mathfrak{g}}}(x)$ and the variance of the prediction is high. Finally the third term tends to dominate when the prediction is close to $\bar{\mathfrak{g}}$ and the uncertainty is high. This means that samples are also added to sparse regions of the space near the sought contour.

4.3.3.2 Expected feasibility function

Bichon et al. (2008) have introduced a new optimization technique similar to EGO, the so-called *Efficient Global Reliability Analysis* (EGRA). Applied in a context of reliability and inspired by the work of Ranjan et al. (2008) previously introduced, they came up with the so-called *expected feasibility function*. It gives an insight on how well the true value of the surrogate function is expected to satisfy the constraint $\widehat{\mathcal{M}}(x) = \overline{\mathfrak{g}}$ in the vicinity of $\overline{\mathfrak{g}} \pm \epsilon$ and reads:

$$EF(\mathbf{x}) = \left(\mu_{\widehat{\mathcal{M}}} - \bar{\mathfrak{g}}\right) \left[2\Phi\left(\frac{\bar{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) - \Phi\left(\frac{\bar{\mathfrak{g}}^{-} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) - \Phi\left(\frac{\bar{\mathfrak{g}}^{+} - \mu_{\widehat{G}}}{\sigma_{\widehat{\mathcal{M}}}}\right) \right] - \sigma_{\widehat{\mathcal{M}}} \left[2\phi\left(\frac{\bar{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) - \phi\left(\frac{\bar{\mathfrak{g}}^{-} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) - \phi\left(\frac{\bar{\mathfrak{g}}^{+} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) \right] + \epsilon \left[\Phi\left(\frac{\bar{\mathfrak{g}}^{+} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) - \Phi\left(\frac{\bar{\mathfrak{g}}^{-} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) \right],$$
(4.23)

where $\bar{\mathfrak{g}}^{\pm} = \bar{\mathfrak{g}} \pm \epsilon$.

Here as well, ϵ is chosen so as to be proportional to $\sigma_{\widehat{\mathcal{M}}}$. This function behaves as the previously introduced improvement functions. It is likely to be high when the prediction is close to the contour level $\overline{\mathfrak{g}}$ and when the variance of the prediction is high.

4.3.3.3 Deviation number

In a reliability analysis context, Echard et al. (2011) proposed a new sequential enrichment technique called *AK-MCS* (for active Kriging combined with Monte Carlo sampling). In this technique, candidates for enrichment are selected among the samples of an initial MCS which minimize the so-called *U-function*, defined by:

$$\mathcal{U}(\mathbf{x}) = \frac{\left|\bar{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}}(\mathbf{x})\right|}{\sigma_{\widehat{\mathcal{M}}}(\mathbf{x})}.$$
(4.24)

The behavior of the function is quite simple as it is minimum when either the prediction is near the contour level $(\mu_{\widehat{\mathcal{M}}} \to \overline{\mathfrak{g}})$, when its associated variance is high or both. The AK-MCS algorithm then consists in selecting among an initial large Monte Carlo sample $\mathfrak{C} = \{x^{(i)} \in \mathbb{X}, i = \{1, \ldots, m\}\}$, the point that minimizes $\mathcal{U}(x)$ and to iteratively process until $\mathcal{U}(x)$ is small enough. The stopping criterion is usually taken to be:

$$\min_{i \in \{1,\dots,m\}} \mathcal{U}\left(\boldsymbol{x}^{(i)}\right) \ge 2.$$
(4.25)

In other words, this criterion means that the probability of misclassifying any point (*i.e.* considering it feasible whereas it is not and vice-versa) in the MC candidates for enrichment is lower than $\Phi(-2) \approx 0.05$.

4.3.3.4 Margin probability function

This criterion was used by Dubourg (2011) in his PhD thesis. It relies on the *margin of uncertainty* of the Gaussian Process predictor. Such a margin is defined as a confidence region with respect to the Gaussian process epistemic uncertainty and writes:

$$\mathbb{M}_{1-\alpha} = \left\{ x \in \mathbb{X} : \bar{\mathfrak{g}} - k_{1-\alpha}\sigma_{\widehat{\mathcal{M}}} \le \mu_{\widehat{\mathcal{M}}} \le \bar{\mathfrak{g}} + k_{1-\alpha}\sigma_{\widehat{\mathcal{M}}} \right\},\tag{4.26}$$

where $k_{1-\alpha}$ is a coefficient associated to the confidence level. We consider throughout this thesis a 95%-confidence margin for which $k_{0.95} \approx 2$.

The margin probability function (MPF) is then defined as the probability that $\widehat{\mathcal{M}}(x)$ belongs to $\mathbb{M}_{1-\alpha}$ and reads:

$$MPF(\mathbf{x}) = \mathbb{P}\left[\widehat{\mathcal{M}}(\mathbf{x}) \leq \bar{\mathfrak{g}} + k_{1-\alpha}\sigma_{\widehat{\mathcal{M}}}\right] - \mathbb{P}\left[\widehat{\mathcal{M}}(\mathbf{x}) \leq \bar{\mathfrak{g}} - k_{1-\alpha}\sigma_{\widehat{\mathcal{M}}}\right]$$
$$= \Phi\left(\frac{\bar{\mathfrak{g}} + k_{1-\alpha}\sigma_{\widehat{\mathcal{M}}} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right) - \Phi\left(\frac{\bar{\mathfrak{g}} - k_{1-\alpha}\sigma_{\widehat{\mathcal{M}}} - \mu_{\widehat{\mathcal{M}}}}{\sigma_{\widehat{\mathcal{M}}}}\right).$$
(4.27)

The next sampling point is then defined as the one that maximizes MPF(x).

4.3.3.5 Illustration examples

Many criteria were presented above. Most of those which are based on the adaptation of EI have been developed while considering that the objective function is also to be approximated by a surrogate model. Therefore there may not be optimal for our case. We thus focus on the remaining three *i.e.* expected feasibility (EFF), margin probability (MPF) and deviation number (DNF) functions. In this section, we perform a comparative analysis of these criteria based on two analytical functions. The first is the one presented earlier in this chapter (see section 4.2.3.1) in Eq. (4.10). The second is the so-called three-hump camel function which has the interesting property of featuring three isolated feasible regions when used as a constraint. It reads:

$$\mathcal{M}(\mathbf{x}) = x_1^6 + 2x_1^2 - 1.05x_1^4 + x_1x_2 + x_2^2.$$
(4.28)

The associated constraint is $\mathcal{M}(x) \leq \bar{\mathfrak{g}}$ over the domain $\mathbb{X} = [-2, 2]^2$, where $\bar{\mathfrak{g}} = 0.5$. Figure 4.8 illustrates the function.

For all the following examples, we consider an anisotropic Kriging with a constant trend and an automatic calibration of the nugget. The model is set up using DiceKriging in R with an hybrid algorithm for the maximum likelihood estimation of the hyperparameters: a global search by genetic algorithm followed by a local BFGS method. In this section, the initial design of experiments consists of a L_2 -discrepancy based optimized Latin hypercube sample of ten points.

Figure 4.9 illustrates the initial settings. The DOE then consists of the ten blue triangles. The true contour is plotted in a thick blue line whereas the Kriging mean prediction is red. The limits of the margin confidence are plotted in black (the dashed line being the lower limit). In the subsequent examples, the next best points to sample will be presented as black diamonds whereas the points already added in previous iterations will be red squares. It should be noted that in the following plots, the input parameters have been mapped to the unit square.



Figure 4.8: Illustration of the three-hump camel function.



Figure 4.9: Initial set up before adaptive sampling (in the reduced space)

Highly non-linear limit state function

In this example only four iterations are presented: 7, 15, 22 and 30. The first observation is that the deviation number and the margin probability function behave the same way. Up to iteration #15, exactly the same points are added with either one of the criteria. There is some divergence starting from that iteration. But this is merely due to the Kriging model. In fact for some reason and despite the DOE is the same, the two MLE optimizations fail to converge to the same hyperparameters. The one obtained during MPF iterations seem erroneous as the hyperparameter in the direction of x_2 is too high, producing a slow-varying process in this direction (Figure (4.10e).

On the other hand, the behavior of the expected feasibility is quite different from the two others. It actually tends to add a lot of points at the boundary of the design space in the early iterations. For this reason, the spread of the margin at the last iteration is higher than those of MPF and

DNF. However, the contour is accurate enough and difference cannot be made between the three (at least visually).



Figure 4.10: Adaptive design for the estimation of contour with $\bar{g} = 0$ of the Haupt function

Three-hump camel function

We proceed in the same way for this second example. The idea is now to be able to detect the three distinct contours leading to local minima of the function (as shown in Figure 4.8b). Once

again, MPF and DNF behave exactly in the same way. The functions $\mathcal{U}(x)$ and MPF(x) are represented respectively in Figures 4.12a and 4.12b. One can see that in the vicinity of the limit states, their landscapes are quite similar (considering that we minimize $\mathcal{U}(x)$ and maximize MPF(x)). However, in general they put too much emphasis on exploitation at the detriment of exploration. The second contour at the lower right corner is found only at the iteration #25 and then all subsequent points are added there. At the last iteration the contour at the upper left corner is still not identified. On the contrary EFF, thanks to its tendency to add points at the boundaries of the design space, catch up quite early the local minima and hence produces a much more accurate contour.



Figure 4.11: Adaptive design for the estimation of contour with $\bar{\mathfrak{g}} = 0.5$ of the three-hump camel function.



Figure 4.12: $\mathcal{U}(\mathbf{x})$ and $MPF(\mathbf{x})$

4.3.4 Multiple points enrichment

To make full use of the available computational power, parallel computation is often used for crash simulations. In such a context, it is appealing to add many points per iteration. The number of points to add is then to match with the number of available CPUs.

Beside these pragmatical considerations, there are other situations where adding many points at once becomes a necessity. As can be seen in the previous examples, the updating criteria are multi-modal. In terms of improvement of the contour estimation, there might not be one single best point but many equally important candidates. Let us take the previous three hump-camel example with MPF. The third contour level could not be identified after 30 iterations. Looking closely at the refinement criteria, one can realize that after a few iterations, it gets trapped by high maxima concentrated on extremely small areas in the vicinity of the middle contour. Thus, large regions of uncertainties are not explored (the two corners) because their local maxima remain slightly lower than the global one located around the middle contour. Figure 4.13 illustrates the point. Contours of the MPF function at iterations 15 and 22 and the associated best next point to add in the learning set are illustrated. During the intermediate iterations and beyond, the optimizer focuses on high peaks in the middle region largely overlooking the local maxima. In this sense, a better strategy would be to seek for the volume under the criteria in small subregions of the design space. The same result can be achieved by multiple points enrichment. In fact, adding many points per iteration will weight the value of the sampling criterion with the size of the region in the input space where uncertainty lies and hence may avoid being trapped in only one part of the contour.

4.3.4.1 Sampling from the learning function

This approach was introduced in Dubourg (2011); Dubourg et al. (2011). They proposed to consider the sampling criterion as a probability density function (up to a normalizing constant) of the best point to add in the learning dataset. The idea is then to sample a large number of candidates according to this distribution. Regions with high expected improvement of the contour will naturally have a higher concentration of points. A statistical reduction technique, herein *K-means clustering*, is then used to summarize the provided information by identifying



Figure 4.13: Contours of MPF(x) at two different iterations of enrichment.

only *K* clusters centers. As a last step, the points for enrichment are selected as those in the Monte Carlo set og candidates which are the closest to the *K* clusters centers.

Any sampling criterion might be used as a probability density function, provided that it fulfills some requirements. More specifically, it should be positive (or zero) in \mathbb{R}^s and maximum at the best point. These two requirements can be met by simple analytical transformations on any function. Besides, its integral should be finite on \mathbb{R}^s , which is not the case here since the Kriging variance increases far from the sampling points. However, since the design space in bounded in X, it was proposed to multiply it by a pseudo-PDF so as to avoid sampling in regions far from the design space. The simplest one, an *indicator function*, was assumed. Hence, the next best point is considered as a random variable *p* whose PDF is

$$f_{P}(\boldsymbol{p}) \propto \mathcal{E}(\boldsymbol{p}) \mathbb{1}_{\mathbb{X}}(\boldsymbol{p}), \qquad (4.29)$$

where \mathcal{E} is any of the above enrichment function and $\mathbb{1}_{\mathbb{X}}$ is an indicator function equals to 1 if $p \in \mathbb{X}$ and 0 otherwise. MPF and EFF can be taken as such but not DNF. In fact, the criterion should be maximized. Despite $-\mathcal{U}$ fulfills this condition, $\varphi(-\mathcal{U})$ is preferred because it is smoother.

Sampling according to $f_P(p)$ resorts to *Markov Chain Monte Carlo* (MCMC) techniques. A wide variety of MCMC algorithms exist (Andrieu et al., 2003). For the following example, we consider the so-called *slice sampler* (Neal, 2003; Damien et al., 1999). In a nutshell, slice sampling proceeds by sampling uniformly in the region that lies under $f_P(p)$. This is achieved by introducing an auxiliary variable u, sampling jointly (u, p) and eventually ignoring u. To the authors experience for the purpose of contour refinement, slice sampling works better than the more general Gibbs sampler or the traditional Metropolis-Hastings algorithm. The main asset is that it does not feature tunable parameters as the latter two. As argued in Dubourg (2011), it is more adapted to the task because of the multi-modal and highly skewed form of $\mathcal{E}(p)$ (for instance, see the middle panels of Figures 4.14 and 4.15).

Figure 4.14 shows iterations 1, 4, 7 and 10 of the MCMC approach for adaptively sampling three points per iteration. The MCMC samples are shown as green crosses in the right end of each row of the figure. One can see that the three contours are identified quite early. Of course the choice of the three points is convenient as we have three disjoint contours. However, examples with two points per iteration also proved to work equally fine.



Figure 4.14: Multiple points enrichment based on the MCMC approach. Three points are added per iteration.

4.3.4.2 Weighted *K*-means clustering

Schöbi and Sudret (2014) proposed a framework for adding multiple points using AK-MCS. Starting from an initial MC sampling, they define the set of points belonging to the margin of uncertainty:

$$\mathbb{M} = \left\{ x \in \mathbb{X} : \bar{\mathfrak{g}} - 2\sigma_{\widehat{\mathcal{M}}}\left(x\right) \le \mu_{\widehat{\mathcal{M}}}\left(x\right) \le \bar{\mathfrak{g}} + 2\sigma_{\mathcal{M}}\left(x\right) \right\}.$$
(4.30)

123

Next, they select *K* points as the centroids of *K* clusters derived from a weighted *K*-means clustering with weight $w(x) = \varphi(-\mathcal{U}(x))$, where φ is the standard Gaussian PDF. We can extend this here to other refinement criteria. As with the MCMC-based approach, when a cluster centroid is not within the predefined margin of uncertainty, its closest point in this margin is selected instead.

Figure 4.15 shows some history plots of the enrichment procedure. This technique has the advantage of simplicity and is straightforward to implement. Especially, it is not as computationally intensive as the MCMC-based approach. We will therefore rely on it for the applications in this thesis.

4.3.5 Multi-constraints handling

In the previous section, we have presented some enrichment functions that can be used for constrained surrogate-based optimization. The enrichment criteria apply to single limit state surfaces. However, most of the applications consist of multi-constrained problems. Additional measures may be taken to account for them. Dubourg et al. (2011) consider to enrich the limit state surfaces sequentially. This is the simplest approach but may not be optimal. In fact, some constraints may be more important than others and focusing on the least important ones may lead to wasting computational budget. Beside, one looses the advantage brought by parallel computing as each time the DOE is updated, one has to re-build all the surrogate models. Finally, it does not assume that all the limit state surfaces can be obtained by a single call to a mechanical model *i.e.* a finite element model herein.

Alternatively, Fauriat and Gayton (2014) have proposed a global criterion for AK-MCS in the case of system reliability. For series systems, *i.e.* those whose failure probability may be obtained as follows:

$$P_f = \mathcal{P}\left(\bigcup_{k=1}^{n_h} \mathfrak{g}_k\left(\boldsymbol{X}, \boldsymbol{Z}\right) \le 0\right).$$
(4.31)

They define a composite limit state surface which reads:

$$\mathfrak{g}_{comp} \equiv \min_{k} \mathfrak{g}_{k}. \tag{4.32}$$

Running a component reliability analysis on this single limit state surface is equivalent to running a system reliability analysis. One may therefore approximate this unique composite function. However, due to its highly irregular shape, the task of metamodeling is made unnecessarily harder. Fauriat and Gayton (2014) propose to rather apply the composite criterion directly on the enrichment function. For the deviation number, this criterion reads:

$$\mathcal{U}_{comp} = \frac{\left|\bar{\mathfrak{g}}_{k_0(\boldsymbol{x})} - \mu_{\widehat{\mathcal{M}}_{k_0(\boldsymbol{x})}}(\boldsymbol{x})\right|}{\sigma_{\widehat{\mathcal{M}}_{k_0(\boldsymbol{x})}}(\boldsymbol{x})},\tag{4.33}$$

where $\{k_0(\mathbf{x}) \in \{1, ..., n_h\} | \hat{\mathfrak{g}}_{k_0(\mathbf{x})}(\mathbf{x}) = \min_{k=1}^{n_h} \hat{\mathfrak{g}}_k(\mathbf{x})\}$ denotes the "most violated" constraint. Recall that the limit state functions are defined by $\hat{\mathfrak{g}}_k(\mathbf{x}) = \bar{\mathfrak{g}}_k - \widehat{\mathcal{M}}_k(\mathbf{x}), k = \{1, ..., n_h\}.$

This criterion may be generalized to the other learning functions by simply writing:

$$\mathcal{E}_{comp}\left(\mathbf{x}\right) = \mathcal{E}\left(\mu_{\widehat{\mathcal{M}}_{k_{0}(\mathbf{x})}}\left(\mathbf{x}\right), \sigma_{\widehat{\mathcal{M}}_{k_{0}(\mathbf{x})}}\left(\mathbf{x}\right)\right), \tag{4.34}$$



Figure 4.15: Multiple points AK-MCS. Three points are added per iteration.

where the index $k_0(\mathbf{x})$ is selected as above. The model $\widehat{\mathcal{M}}_{k_0(\mathbf{x})}$ is therefore the one associated to the most violated constraint.

This approach might be flawed in cases where the different constraints have values not expressed in the same scale. The ranking of the constraints in terms of the degree of violation would require a normalization. If one constraint dominates the other in terms of magnitude, then the enrichment will focus on this constraint. This effect was shown in Moustapha et al.

(2015) where the approach was applied to the lightweight design of an automotive body structure sub-system. A turn-around would be to consider directly ranking the constraints with respect to their \mathcal{U} -values. The function is actually normalized so one may write instead:

$$\mathcal{U}_{comp} = \min_{k \in \{1, \dots, n_h\}} \mathcal{U}_k\left(\mathbf{x}\right) = \frac{\left|\bar{\mathbf{g}}_k - \mu_{\widehat{\mathcal{M}}_k}\left(\mathbf{x}\right)\right|}{\sigma_{\widehat{\mathcal{M}}_k}\left(\mathbf{x}\right)}.$$
(4.35)

The two approaches are compared in Figure 4.16. The upper panel shows the approximated limit states surfaces. The dots are points in \mathfrak{C}_2 which are used to compute the criterion. The colors refer to the constraint that is used for the computation of \mathcal{U}_{comp} in either of the approaches. The lower panel shows contours of the resulting criterion and the next best points (black diamonds).



Figure 4.16: Comparison of the two approaches to account for multiple constraints: the composite criterion proposed by Fauriat and Gayton (2014) and the one we proposed simply based on the minimum of the deviation number.

4.4 **RBDO** with surrogate model and enrichment

4.4.1 Introduction

The above items are now combined so as to propose a methodology for reliability-based design optimization with surrogate models and quantiles measures of conservatism in the design. Three steps can be clearly identified.

- 1. Start an initial design over which a surrogate model is built. This initial DOE can be as scarce as possible so that the computational budget is saved for refinement of the contour in regions where the constraints are likely to be violated;
- 2. Iteratively update the initial design so that the surrogate models are accurate enough on the limit state surfaces;
- 3. Proceed to optimization on the surrogate model.

The first step may be achieved by one of the designs introduced in Appendix A. In this thesis, we mainly consider a L_2 -discrepancy optimized Latin hypercube. The second step resorts to any of the criteria for contour refinement *i.e.* deviation number, margin probability or expected feasibility functions. The three approaches, despite some minor differences, roughly lead to the same results. We consider the deviation number for its simplicity. However, the criteria were introduced in the case of deterministic analysis. Therefore, we propose below to adapt it to quantile estimation. Eventually, the third step can be solved by any general-purpose optimization algorithm. CMA-ES has been selected as the default optimizer in this thesis.

4.4.2 Contour refinement with quantile estimation

The criteria introduced for contour refinement are readily applicable for deterministic design optimization. In the case of quantile-based RBDO, some adaptations must be made. In fact, we seek to refine the surrogate model so that the quantiles are accurately estimated in regions where the constraints are likely to be violated. The constraints are expressed in the design space \mathbb{D} but the surrogate model is built in the augmented space $\mathbb{X} \times \mathbb{Z}$. The trick is then to find a point in $\mathbb{X} \times \mathbb{Z}$ which will most likely improve the metamodel so that, *in fine*, the quantile is more accurate in the regions of interest. To achieve this, we devise the following algorithm (summarized in Figure 4.19), adapting the deviation number:

- 1. Sample a set of candidates for enrichment $\mathfrak{C} = \left\{ d^{(i)} \in \mathbb{D}, i = 1, ..., n \right\};$
- 2. For any design $d^{(i)} \in \mathfrak{C}$:
 - a) Build the Monte Carlo set $\mathfrak{C}_q^{(i)} = \left\{ \left(x^{(j)}, z^{(j)} \right), j = 1, \dots, N \right\}$ needed to compute the quantile, see Eq. (2.90);
 - b) Compute the quantile $\widehat{q}_{\alpha}\left(d^{(i)}\right) = \widehat{y}_{\lfloor N\alpha \rfloor}$ as defined in Eq. (2.91)
 - c) Identify the following point:

$$\left(\boldsymbol{x}_{\alpha}^{(i)}, \boldsymbol{z}_{\alpha}^{(i)}\right) = \left\{ (\boldsymbol{x}, \boldsymbol{z}) \in \mathfrak{C}_{q}^{(i)} | \widehat{q}_{\alpha} \left(\boldsymbol{d}^{(i)}\right) = \mu_{\widehat{\mathcal{M}}} \left(\boldsymbol{x}, \boldsymbol{z}\right) \right\};$$
(4.36)

127
d) Compute the criterion on this point:

$$\widetilde{\mathcal{U}}\left(\boldsymbol{d}^{(i)}\right) \equiv \mathcal{U}\left(\boldsymbol{x}_{\alpha}^{(i)}, \boldsymbol{z}_{\alpha}^{(i)}\right) = \frac{\left|\tilde{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}_{\alpha}^{(i)}, \boldsymbol{z}_{\alpha}^{(i)}\right)\right|}{\sigma_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}_{\alpha}^{(i)}, \boldsymbol{z}_{\alpha}^{(i)}\right)};$$
(4.37)

3. Select the next best point as the one which minimizes the deviation number:

$$(x_{\text{next}}, z_{\text{next}}) = \arg\min_{(x_{\alpha}, z_{\alpha}) \in \mathfrak{C}_{\alpha}} \widetilde{\mathcal{U}}(d),$$
 (4.38)

where \mathfrak{C}_{α} is the set of all points defined in Eq. (4.36).

Figure 4.17 illustrates how the point are chosen for the computation of the deviation number in Eq. (4.37). In the left panel, four points are selected in the design space. They are shown as blue crosses. For each of them, the MC population $\mathfrak{C}_q^{(i)}$ used to compute the quantile is shown as colored dots. The corresponding points $(\mathbf{x}_{\alpha}^{(i)}, \mathbf{z}_{\alpha}^{(i)})$ are highlighted by the red crosses. These points are the one used to compute the \mathcal{U} criterion.



(a) Monte Carlo sample sets \mathfrak{C}_q and associated candidates for enrichment (b) $\mathcal{M}(\mathbf{x}, \mathbf{z})$ in the augmented space and resulting quantile $q_{\alpha}(d)$

Figure 4.17: Illustration of enrichment in the augmented space while the constraint is defined in the design space.

The entire methodology is also illustrated in Figure 4.18. We consider the two-dimensional function introduced in Eq. (2.94). The left column shows contours of the enrichment function in the augmented space. The contour $\{(x, z) \in X \times \mathbb{Z} | \mu_{\widehat{\mathcal{M}}}(x, z) = 0\}$ is also shown as the black dashed line. The small crosses define the set \mathfrak{C}_{α} which are the input points for computation of \mathcal{U} in Eq. (4.37). In the right column, the true and approximated quantiles are plotted respectively in blue and black. The initial DOE consists of six points (blue triangles). The figure illustrates four iterations of the enrichment. The convergence occurs after 11 iterations when the confidence interval of the quantile estimation has shrunk considerably. This confidence interval $[q_{\alpha}^-, q_{\alpha}^+]$ is computed here by evaluating the quantiles with respect to $\mu_{\widehat{\mathcal{M}}} \pm 2\sigma_{\widehat{\mathcal{M}}}$.



(a) Augmented space (left) and design space (right): iteration #1



(b) Augmented space (left) and design space (right): iteration #4



(c) Augmented space (left) and design space (right): iteration #7



(d) Augmented space (left) and design space (right): iteration #10

Figure 4.18: Enrichment with the function $(d, z) \mapsto \mathcal{M}(d, z) = (1/3z^4 - 2.1z^2 + 4)z^2 + dz + 4d^2(d^2 - 1)$. In the left panel, the augmented space with contour of the enrichment functions and the set \mathfrak{C}_{α} shown as small crosses. In the right panel, the quantities q_{α} , q_{α}^- and q_{α}^+ respectively in black, red and cyan. Triangles and squares respectively stand for initial and enrichment points.



Figure 4.19: Flowchart of the methodology with enrichment for quantile estimation

4.5 **RBDO** with two stages of enrichment

4.5.1 Motivations and first stage of enrichment

The stopping criterion proposed here is quite conservative. Requiring $\min U \ge 2$ leads to focusing in the vicinity of the limit state surface in the entire design space. This would definitely lead to a needlessly high number of points. Besides, there are regions of the space where the objective function increases and which consequently are not of interest to the designer. EGRA approach does not indeed account for information about the fitness of the objective function as EGO does.

For these reasons, we propose a two-stages enrichment methodology. The first stage consists in globally updating the design of experiments as proposed in the previous section. However, the conservative stopping criterion is modified. Instead of requiring $\tilde{\mathcal{U}}$ to be higher than 2 for all points of \mathfrak{C} , we relax the condition to only a portion of the points. Let us consider the following subset of \mathfrak{C}

$$\mathfrak{C}_{2} = \left\{ d \in \mathfrak{C} | \widetilde{\mathcal{U}} \left(d \right) \leq 2 \right\}.$$
(4.39)

The relaxed stopping criterion for this first stage of enrichment is considered achieved when:

$$\eta = \operatorname{Card}\left(\mathfrak{C}_{2}\right) / \operatorname{Card}\left(\mathfrak{C}\right) \leq \bar{\eta}. \tag{4.40}$$

 $\eta \in [0,1]$ is hence the percentage of points for which $\tilde{\mathcal{U}} \leq 2$. This threshold can be tuned to control the global accuracy of the surrogate model in the vicinity of the limit state surface. The early stopping criterion min $\tilde{\mathcal{U}} \geq 2$ corresponds to $\bar{\eta} = 0$. For our applications, we set $\bar{\eta} = 0.15$. Figure 4.20 shows two quantile estimates for two different iterations of enrichment in the example introduced above. The right panel shows the last iteration for $\bar{\eta} = 0$ whereas the left one corresponds to $\bar{\eta} = 0.15$. The latter does not match perfectly the true quantile in the entire design space. But achieving this in the former case had required 6 additional iterations thus doubling the computational budget for enrichment. We argue here that such an accuracy is not really necessary. In fact, let us consider the optimization problem introduced in Eq. (2.96) whose objective function is min $(d+1)^2$. The rightmost region of the space corresponds to increased values of the objective function, and we do not need to be extremely accurate there. Instead, we henceforth advocate to stop this first stage of enrichment with respect to the relaxed criterion. The residual model uncertainty can be further reduced during the optimization process as explained in the sequel. A pseudo-algorithm for this global enrichment strategy is given in Algorithm 1.

4.5.2 The second stage of enrichment

The idea for this second stage of enrichment is to bring an additional information to the enrichment in the fashion of EGO. The optimization algorithm starts with a surrogate model which is known to be globally accurate *i.e.* roughly all regions of interest have been located but there remains a residual epistemic uncertainty in these regions. During the optimization, each time the surrogate model is called, its accuracy is checked. In the case it is not deemed satisfactory, a local enrichment is performed so as to improve the accuracy of the quantile estimate.



Figure 4.20: Global enrichment of the function $(d, z) \mapsto \mathcal{M}(d, z) = (1/3z^4 - 2.1z^2 + 4)z^2 + dz + 4d^2(d^2 - 1)$ with two values of the thresholds for enrichment. In the left panel, the criterion we propose. In the right panel, the original criterion min $\tilde{\mathcal{U}} \ge 2$

4.5.2.1 Enrichment for quantile accuracy

To assess the accuracy of the quantiles, let us first introduce upper and lower bounds on the quantiles, respectively denoted q_{α}^+ and q_{α}^- . These two quantities are defined with respect to the upper and lower margins of uncertainty as defined by the Kriging model with a 95% confidence level. In other words, the values of q_{α}^+ and q_{α}^- are defined by computing the quantile with respect to $\mu_{\widehat{M}} + 2\sigma_{\widehat{M}}$ and $\mu_{\widehat{M}} - 2\sigma_{\widehat{M}}$. The following relationship holds for any $d \in \mathbb{D}$:

$$\widehat{q}_{\alpha}^{-}\left(d\right) \leq \widehat{q}_{\alpha}\left(d\right) \leq \widehat{q}_{\alpha}^{+}\left(d\right), \tag{4.41}$$

since $\sigma_{\widehat{M}}(x, z) \ge 0$ for any point $(x, z) \in \mathbb{X} \times \mathbb{Z}$.

The quantities q_{α}^{-} and q_{α}^{+} therefore bound the quantile estimate. The width of the corresponding margin decreases with $\sigma_{\widehat{\mathcal{M}}}$, that is as the Kriging epistemic uncertainty is reduced. This bound can therefore be used to quantify the error brought by replacing the true model by surrogate models. Dubourg et al. (2013) and earlier Deheeger and Lemaire (2007) used similar bounds to assess accuracy of failure probabilities estimated with respect to Kriging and support vector machines. Here we consider the relative width of this quantile margin with respect to the threshold. This leads to the following accuracy criterion:

$$\eta_{q}\left(d\right) = \frac{\widehat{q}_{\alpha}^{+}\left(d\right) - \widehat{q}_{\alpha}^{-}\left(d\right)}{\bar{\mathfrak{g}}} \leq \bar{\eta}_{q},\tag{4.42}$$

if $\bar{\mathfrak{g}} \neq 0$. Otherwise, only the numerator is considered and the threshold $\bar{\eta}_q$ is modified to account for it.

If at the *i*-th iteration of optimization $\eta_q(d^{(i)}) > \bar{\eta}_q$, then it is considered that the accuracy of the quantile with respect to the threshold is not enough. Since the margin width is proportional to the Kriging epistemic uncertainty, the means to reduce it boil down to adding points to the design of experiments. In this respect, the candidates for enrichment are taken among the

Algorithm 1 Global enrichment strategy wi	th quantiles as constraints	
Initialization:		
Initial DOE ${\cal D}$	▷ Here optimal Latin h	ypercube
Initial metamodel $\widehat{\mathcal{M}}$		
Global accuracy criterion threshold $\bar{\eta}$	⊳ By default here	$\bar{\eta} = 0.15$
Size of the Monte Carlo candidates for e	enrichment m > By default here m	= 5,000
Quantile value α	▷ By default here	$\alpha = 0.95$
Size of the Monte Carlo set for quantile	computation $N \triangleright$ By default here $N =$	= 10,000
1: repeat		
2: Sample uniformly $\mathfrak{C} = \left\{ d^{(1)}, \dots, d^{(t)} \right\}$	$^{n)} \}$ in the design space \mathbb{D}	
3: for $i = 1$ to m do	j	
4: Draw samples $(x_1, z_1), \ldots, (x_N, z_N)$	_N) in the augmented space where $X \sim f$	$\mathbf{x}_{\mathbf{X} \mathbf{d}(i)}$ and
$\mathbf{Z} \sim f_{\mathbf{Z}}$		$\mathbf{A} \mathbf{u}^{(\prime)}$
5: for $j = 1$ to <i>N</i> do		
6: $\widehat{y}_{i} = \mu_{\widehat{\mathcal{M}}}(x_{i}, z_{i})$		
7: end for	`	
8: $q_{\alpha}\left(\boldsymbol{d}^{(i)}\right) = \text{quantile}\left(\left\{\widehat{y}_{j}\right\}_{j=1}^{N}, \boldsymbol{u}_{j}^{N}\right)$	x)	
9: Identify the point $(\mathbf{x}_{\alpha}^{(j)}, \mathbf{z}_{\alpha}^{(j)})$	⊳ As in]	Eq. (4.36)
10: $\widetilde{\mathcal{U}}_{j} = \left \bar{\mathfrak{g}} - \mu_{\widehat{\mathcal{M}}} \left(\boldsymbol{x}_{\alpha}^{(j)}, \boldsymbol{z}_{\alpha}^{(j)} \right) \right / \sigma_{\widehat{\mathcal{M}}} \left(\boldsymbol{x}_{\alpha}^{(j)}, \boldsymbol{z}_{\alpha}^{(j)} \right) \right $	$\mathbf{x}_{lpha}^{(j)}$, $\mathbf{z}_{lpha}^{(j)}$)	
11: end for	,	
12: if $K == 1$ then	\triangleright The point that minimizes $\widetilde{\mathcal{U}}$ i	s chosen
13: $(\mathbf{x}_{\text{next}}, \mathbf{z}_{\text{next}}) = \arg\min\left\{\widetilde{\mathcal{U}}_{j}\right\}_{j=1}^{N}$		
14: else	\triangleright <i>K</i> points are chosen among the <i>N</i> can	ndidates
15: $(x_{\text{next}}, z_{\text{next}}) = \text{Weighted } K\text{-mean}$	is clustering with weight $arphi\left(-\widetilde{\mathcal{U}}_{j} ight)$	
16: end if		
17: Update DOE \mathcal{D} and metamodel \mathcal{M}		
18: Compute accuracy criterion η	⊳ Following E	Eq. (4.40)
19: until $\eta \leq \bar{\eta}$		

points $(x, z) \in \mathfrak{C}_q^{(i)}$ since they are the very points used to compute the quantile. The deviation number can henceforth be adapted so as to find points that enable an improvement of the quantile. In this respect, the following criterion can be derived:

$$\mathfrak{U}(\mathbf{x}, \mathbf{z}) = \frac{\left|\widehat{q}_{\alpha}\left(\boldsymbol{d}^{(i)}\right) - \mu_{\widehat{\mathcal{M}}}\left(\mathbf{x}, \mathbf{z}\right)\right|}{\sigma_{\widehat{\mathcal{M}}}\left(\mathbf{x}, \mathbf{z}\right)},\tag{4.43}$$

where $(x, z) \in \mathfrak{C}_q^{(i)}$.

The arguments that minimize Eq. (4.43) are the one that will improve the quantile estimate. Sequentially adding points in the design of experiments following this scheme will eventually lead to an accurate estimate of the quantile. In the particular case when only one point is added, the argument that minimizes \mathfrak{U} is actually $(\mathbf{x}_{\alpha}^{(i)}, \mathbf{z}_{\alpha}^{(i)})$ defined in Eq. (4.36).

This enrichment scheme is embedded in optimization. We avoid gradient-based approaches since the problems we aim to solve are multi-modal and therefore not well fitted to local search methods. We rather consider a global search technique, more specifically the covariance matrix adaptation evolution strategy method (CMA-ES). Furthermore, we consider the (1 + 1)-scheme where design points are sampled sequentially, *i.e.* one parent generates one offspring. This setting allows us to check for all sampled points that the quantile accuracy is below a given threshold before moving to the next one. However a more general setting where many points are sampled in each generation of the optimization algorithm may also be considered. The enrichment strategy would however need to be slightly modified so as to select the particular design points for which the \mathfrak{U} criterion are calculated. This methodology is summarized in the pseudo-algorithm 2.

Alg	orithm 2 Adaptive Kriging Quantile-based	RBDO
Init	ialization:	
	Initial DOE \mathcal{D}	Here based on optimal Latin hypercube
	Initial metamodel $\hat{\mathcal{M}}$	
	Optimization starting point $d^{(0)}$	
	Number of enrichment points per iteration	• K
	Constraint and quantile accuracy threshold	ds $\bar{\mathfrak{g}}$ and $\bar{\eta}_q$ \triangleright Here by default $\bar{\eta}_q = 0.1$
	Size of the Monte Carlo set for quantile est	imation N \triangleright Here by default $N = 10,000$
1:	$\overline{i=0}$	
2:	while Not converged do	
3:	Draw samples $\mathfrak{C}_q^{(i)} = \{(x_1, z_1), \dots, (x_N,$	$\{m{z}_N)\}$ in the augmented space where $m{X} \sim f_{m{X} m{d}^{(i)}}$
	and $\mathbf{Z} \sim f_{\mathbf{Z}}$	
4:	for $j = 1$ to N do	
5:	$\widehat{y}_{j} = \mu_{\widehat{\mathcal{M}}}\left(\boldsymbol{x}_{j}, \boldsymbol{z}_{j}\right)$	
6:	$y_j^- = \mu_{\widehat{\mathcal{M}}}(x_j, z_j) - 2\sigma_{\widehat{\mathcal{M}}}(x_j, z_j)$	
7:	$\widehat{y}_{j}^{+}=\mu_{\widehat{\mathcal{M}}}\left(x_{j},z_{j} ight)+2\sigma_{\widehat{\mathcal{M}}}\left(x_{j},z_{j} ight)$	
8:	end for $(r^{(i)})$	
9:	$q_{lpha}\left(oldsymbol{d}^{(i)} ight) = ext{quantile}\left(\left\{\widehat{y} ight\}_{j=1}^{N}, lpha ight)$	
10:	$q_{lpha}^{-}\left(oldsymbol{d}^{(i)} ight)= ext{quantile}\left(\{\widehat{y}^{-}\}_{j=1}^{N},lpha ight)$	▷ Lower bound of the quantile
11:	$q_{lpha}^{+}\left(\boldsymbol{d}^{\left(i ight)} ight) = ext{quantile}\left(\left\{ \widehat{y}^{+} ight\}_{j=1}^{N}, lpha ight)$	▷ Upper bound of the quantile
12:	if $(q_{\alpha}^+ - q_{\alpha}^-) / \bar{\mathfrak{g}} > \bar{\eta}_q$ then	
13:	for $k = 1$ to N do	Υ.
14:	$\mathfrak{U}_{k} = \mu_{\widehat{\mathcal{M}}}(\mathbf{x}_{k}, \mathbf{z}_{k}) - q_{\alpha} / \sigma_{\widehat{\mathcal{M}}}(\mathbf{x}_{k}, \mathbf{z}) - q_{\alpha} / \sigma_{\widehat$	(z_k)
15:	if $K = -1$ then	N The point that minimizes 1/ is chosen
10.	$(\mathbf{x} + \mathbf{z} + \mathbf{z}) = \arg\min\{\{\mathbf{x}\}\}^N$	V The point that minimizes of is chosen
17:	$(x_{\text{next}}, z_{\text{next}}) - \arg \min \{x_k\}_{k=1}$	N K points are chosen among the N candidates
10. 19·	$(\mathbf{x}_{novt}, \mathbf{z}_{novt}) = Weighted K-mean$	is clustering with weight $\alpha(-\mathfrak{U}_{k})$
20:	end if	
21:	end if	
22:	Update DOE ${\mathcal D}$ and metamodel $\widehat{{\mathcal M}}$	
23:	$d^{(i)} \leftarrow d^{(i)} + v^{(i)}$ > Explore the r	hext design point (here we use $(1 + 1)$ -CMA-ES)
24:	$i \leftarrow i + 1$	
25:	Check convergence of the optimization	algorithm
26:	end while	

4.5.2.2 A summary of the proposed methodology

The entire methodology for quantile-based RBDO with enrichment of an initial design of experiments is summarized in the flowchart of Figure 4.21. The initialization consists in setting the size of the starting design of experiments n, the number of points that are added per iteration in the first and second stages of enrichment, respectively denoted by K_1 and K_2 and thresholds $\bar{\eta}$ and $\bar{\eta}_q$. The size of the initial design of experiments can be set relatively low. As for $\bar{\eta}$, the lower it is, the more residual uncertainty will be left before the second stage of enrichment is started. The overall optimization time may therefore be affected by the setting of these parameters. In an industrial context, where the project lead times matter a lot, it is necessary to fine-tune these parameters so as to avoid too many iterations of enrichment. One leverage is of course the recourse to parallel computations to add many points per iteration. Besides, from a more practical point of view, one knows in advance what the computational budget approximately is for industrial applications. In principle for a given total number of calls to the model, the surrogate model is more accurate when the enrichment is distributed over more iterations. But this does not favor a reduction of the project lead-time. The trick is then to adapt the choices of n, $\bar{\eta}$ and $\bar{\eta}_q$ so as to trade the optimization time with the number of different iterations of enrichment.

Once these parameters are set, an initial design of experiments is generated and the Kriging models are built. Then the first stage of enrichment starts as described in Section 4.4.2. This may be seen as a *global enrichment stage*. The algorithm first searches to locate all regions of interest. In contrast, the following second stage of enrichment is *local*. Points are added locally around the current design to improve the quantile estimate. The optimization algorithm we use is the (1 + 1)-CMA-ES for constrained problems (Arnold and Hansen, 012a). In this second stage some tricks may be used to fine-tune the efficiency of the enrichment scheme. In fact, it is expected that the optimization algorithm explores the design space in the early iterations and then will focus on exploiting the identified local minima in the latter iterations. In such a case, it is not necessary to be extremely accurate in the early iterations, so the threshold $\bar{\eta}_a$ may be relaxed. Likewise, during optimization, the algorithm may sample points which are either unfeasible or which do not improve the fitness with respect to the current best point. In such cases, it might also be interesting to relax the threshold. These two points can be used to finetune the enrichment scheme. Especially, one can consider a decreasing sequence of thresholds in a simulated-annealing fashion *i.e.* a threshold that is relatively high in the early iterations and that decreases in the latter ones as the optimization algorithm starts exploitation.



Figure 4.21: Flowchart of the methodology with enrichment for quantile estimation

4.5.3 Validation example

To illustrate and validate this methodology, we consider the Haupt problem already introduced in this chapter. The optimization problem reads:

$$d^* = \arg\min_{d \in \mathbb{D}} (d_1 - 3.7)^2 + (d_2 - 4)^2 \quad \text{s.t.} \quad \mathfrak{g}(d) = -d_1 \sin(4d_1) - 1.1 \sin(2d_2) \le 0.$$
(4.44)

The RBDO problem was formulated by considering that the design variables follow independent normal distributions: $X_i \sim \mathcal{N}(d_i, 0.1^2)$. A reliability index $\bar{\beta} = 2$ corresponding to a target failure probability $\bar{P}_f = \Phi(-\bar{\beta}) = 0.0228$ was also considered.

In section 4.2.3.1, the optimization was performed with models built on different DOEs of increasing sizes. The results were accurate only with the designs of 80 and 100 points. In this section, we apply the enrichment strategy introduced here. We start with a scarce initial design of size 10.

For the first stage of enrichment, we consider $\bar{\eta} = 0.15$. This leads to adding 24 points in the design. For the record, letting the algorithm run until the criterion min $\tilde{\mathcal{U}} \ge 2$ is reached leads to adding 47 points. Figure 4.22 shows the resulting DOE and Kriging models after enrichment for each of the two cases. The true and approximated limit state surfaces are respectively plotted in red and blue. In the left panel, the contour of the limit state surface is well identified however the approximation is not completely accurate as residual uncertainty of the Kriging model has been left on purpose for the next stage. In the right panel, the Kriging model completely matches the true model in the entire design space. This enhanced accuracy however comes at the cost of model evaluations which almost double between the two cases.





(b) Final state with the stopping criterion $\bar{\eta}=0$ (min $\mathcal{U}\geq 2$)

Figure 4.22: DOE and resulting Kriging models after the first stage of enrichment. In the left panel, the relaxed threshold we propose and in the right one, the original criterion min $\tilde{U} > 2$.

Considering the setting in Figure 4.22a, we start optimization using constrained (1+1)-CMA-ES. The optimization converges to the reference solution after updating the design ten times. The

overall number of model evaluations in the entire process is 45. This is to compare with the introduction example in section 4.2.3.1 where an accurate solution could not be found with an initial design of size 60. Likewise, there are still less points in the designs as if we were to use the initial criterion of min $\tilde{\mathcal{U}} \geq 2$. The optimization process is illustrated in Figure 4.23. The final Kriging model is shown to be accurate in the region around the minimum. In fact, enrichment points were added only there. The accuracy of region with increased objective function has not been improved, hence avoiding the waste of model evaluations. In this figure, we also plot the points sampled by CMA-ES. The red points are those falling in the unfeasible set. The green pattern show the successive best points sampled during optimization up to the optimal solution shown as green diamond. Finally, the blue points are feasible but did not improve the current best point at the moment they were sampled. These information may be used to further reduce the computational cost. For instance, one may relax the accuracy criterion for blue and red points in the latter iterations as one already knows that they will not be kept by the optimizer.



Figure 4.23: Convergence of CMA-ES: Points sampled during optimization

Eventually, history plots of the enrichment criteria for each stage are shown in Figure 4.24 below. The thresholds are plotted in dashed red lines. For the second stage, the threshold was set dynamically. Three levels of decreasing values were considered. In fact by setting the threshold high in the early iterations and lowering in the latter one, we avoid enriching too much while CMA-ES is exploring. Enrichment is carried out each time the blue plot is above the red line. As the design is enriched and the optimization starts focusing on a region of the space, the quantile accuracy criterion η_q becomes small.



(a) Evolution of the criterion η for the first stage of enrichment (b) Evolution of the criterion η_q of the second stage of enrichment

Figure 4.24: DOE and resulting Kriging models after the first stage of enrichment.

4.6 Conclusion

In this chapter, we have introduced a surrogate modeling approach for reliability-based design optimization. This RBDO problem is solved thanks to a double-loop approach where the inner loop simply consists of the computation of quantiles of the approximated constraints. The surrogate models are built on the augmented space so as to cover a sufficiently large area where the realizations of the random parameters lie. Various techniques for sampling in this space were reviewed. It was shown on a few examples that it is necessary to rely on adaptive designs of experiments in order to reduce the computational cost. Among the numerous techniques, deviation number, whose main virtue is simplicity, was considered for applications.

A methodology for RBDO with two stages of enrichment was eventually proposed. The first stage consists of global enrichment. The aim is to identify all regions where the constraints are likely to be violated. The surrogate models, herein Kriging, are then refined in these regions but only up to a certain point. The second stage combines optimization and enrichment. The idea is to bring additional information to the optimizer by redirecting the computational budget in regions with increased fitness. The approach was illustrated on a problem with a two-dimensional highly non-linear limit state function. Efficiency of the RBDO is shown to be enhanced thanks to this approach. In the next chapter, applications are shown on other analytical problems with a special emphasis on comparison with solutions found in literature. Finally, the lightweight design of an automotive body structure is performed.



ACADEMIC AND INDUSTRIAL APPLICATIONS

Contents

5.1	Introd	uction
5.2	Acade	mic validation examples
	5.2.1	Column under compression
	5.2.2	Choi problem
	5.2.3	Bracket structure
5.3	Towar	d an industrial application
	5.3.1	Elements of non-linear dynamic analysis
	5.3.2	Numerical noise in frontal impact
	5.3.3	Frontal impact crash scenarios
5.4	RBDO	application on a vehicle frontal impact
	5.4.1	Application on the sidemember subsystem
	5.4.2	Application on a full automotive body structure
5.5	Conclu	asion

5.1 Introduction

In the previous chapter, we have introduced a methodology for quantile-based RBDO. This chapter is devoted to an application of this methodology to the lightweight design of an automotive body structure. Prior to this, we validate the approach on analytical examples. The investigated problems were selected so as to cover the three possible cases for the definition of the probabilistic model:

- Case #1: Randomness in environmental variables only;
- Case #2: Randomness in design variables only;
- Case #3: Randomness in both design and environmental variables.

Furthermore, for the two latter cases, the efficiency of the approach is cross-validated thanks to data available in the literature. As to the first one, only the accuracy of the found optimum is checked since an analytical exact solution can be derived.

The second part of this chapter concerns the lightweight design of an automotive body structure under frontal impact-related constraints. As pointed out in the introduction of this thesis, frontal impact is chaotic in nature. Thus as a preliminary stage, we study some effects of this chaotic behavior, *i.e.* numerical scatter of crash simulations and the possibility of various crash scenarios. The latter makes use of support vector machines for classification introduced in Section 3.3.2. Afterwards, we apply the adaptive-Kriging quantile-based RBDO to the design of a vehicle.

5.2 Academic validation examples

In this section, we validate the approach introduced in the previous chapter. The three cases mentioned above are treated. Kriging is selected as default surrogate model in this section. More specifically, we consider anisotropic Kriging with a constant trend (*i.e.* ordinary Kriging). The nugget effect is turned on so as to facilitate the maximum likelihood estimation of the Kriging hyperparameters. Matérn 5/2 kernel is used as an auto-correlation function. The initial model is built with a design consisting of an optimal Latin hypercube sampling. As for optimization, we consider (1 + 1)-CMA-ES for constrained problems completed by an SQP (gradient-based) algorithm through MATLAB's fmincon function.

5.2.1 Column under compression

Problem definition

This example was introduced in Section 2.3.3.2. It consists of a column with rectangular crosssection submitted to compressive loading F_{ser} . The aim is to minimize its section $b \times h$ while avoiding buckling. Buckling is assessed thanks to the critical Euler force which reads:

$$F_{cr} = \frac{\pi^2 EI}{L^2},\tag{5.1}$$

where *L* is the length of the column, *E* is the Young's modulus of its constitutive material and $I = bh^3/12$ (b > h) is the column moment of inertia.

The associated limit state surface therefore reads:

$$\mathfrak{g}(\boldsymbol{d},\boldsymbol{z}) = F_{ser} - k \frac{\pi^2 E b h^3}{12L^2},$$
(5.2)

where *k* is a coefficient introduced to account for noise arising from unidentified sources that may affect the Euler force, $d = \{b, h\}^T$ and $z = \{k, E, L, F_{ser}\}^T$ are respectively vectors gathering the design and environmental variables. The probabilistic model for this example is reminded in Table 5.1. Three of the environmental variables are considered as random. The target probability of failure is $\bar{P}_f = 0.05$. The analytical solution leads to $b^* = h^* = 238.45$ mm, as detailed in page 51.

Table 5.1: Probabilistic model for the column under compression.

Parameter	Distribution	Mean (μ)	COV (δ%)
k	Lognormal	0.6	10
E (MPa)	Lognormal	10,000	5
<i>L</i> (mm)	Lognormal	3,000	1
F_{ser} (kN)	_	$1.4622 imes 10^6$	_

First stage of enrichment

To solve this five-dimensional problem, we consider an initial design of size 10. We set the threshold for global enrichment to $\bar{\eta} = 0.15$. This leads to adding two points in the design. The successive values of the global accuracy criterion are therefore $\eta = \{0.21, 0.15, 0.12\}$. The initial value of η is quite low, despite the design of experiments of size 10 is scarce with respect to the dimension of the problem (five). This means that the constraint is smooth and therefore easy to approximate. Contour plots of the limit state surface confirm this.

Optimization and Second stage of the enrichment

In this second stage, we start the optimization with the following three values of the accuracy thresholds in a simulated annealing fashion. For iterations up to 30 we set $\eta_q^{(1)} = 1$ between iterations 31 and 100, $\eta_q^{(2)} = 0.5$ and beyond 100, $\eta_q^{(3)} = 0.1$. The algorithm converges to the solution $b^* = h^* = 239.12$ mm which corresponds to a relative error of 0.28 % with respect to the reference solution. Figure 5.1 illustrates the convergence of the problem. In Figure 5.1a, the points sampled by CMA-ES are shown in the design space. The red and blue areas delimit the infeasible space and contours of the objective function are shown in shades of gray. The points for which the quantile accuracy was beyond the threshold are circled in cyan. History plots of this criterion is shown Figure 5.1b where η_q and $\bar{\eta}_q$ are plotted respectively in blue and red. Five enrichments were necessary before convergence. The total number of functions calls amounts to 18 which is quite low if one is to consider the dimension of the problem. Beside, when we consider to perform optimization without enrichment with an initial DOE of size 18, convergence to the true solution is most of the time not achieved.



(a) Points sampled during CMA-ES in the design space

(b) Quantile accuracy criterion during optimization

Figure 5.1: Convergence of the column under compression optimization problem. In the left panel, the green points are the successive best points. The blue one are admissible points that did not improve the best point at the moment they were sampled. The red points are unfeasible sampled designs. The points encircled in cyan are those around which enrichment has been done during optimization. In the right panel the cyan circles show iterations where enrichment has been done.

5.2.2 Choi problem

Problem presentation

A similar example was introduced in Section 2.3.3.1. Here, the objective function and probabilistic model are slightly modified so as to fit the cases treated in the literature. The DDO problem reads:

$$d^{*} = \arg\min_{d \in [0,10]^{2}} d_{1} + d_{2} \quad \text{s.t.:} \begin{cases} \mathfrak{g}_{1} \left(d \right) = \frac{d_{1}^{2}d_{2}}{20} - 1 \leq 0 \\ \mathfrak{g}_{2} \left(d \right) = \frac{\left(d_{1} + d_{2} - 5 \right)^{2}}{30} + \frac{\left(d_{1} - d_{2} - 12 \right)^{2}}{120} - 1 \leq 0 \\ \mathfrak{g}_{3} \left(d \right) = \frac{80}{\left(d_{1}^{2} + 8d_{2} + 5 \right) - 1} \leq 0 \end{cases}$$
(5.3)

For the probabilistic model, we consider the settings in Shan and Wang (2008) and Dubourg (2011) for comparison purposes. In these settings, the design variables are considered as random with the following distribution $X_i \sim \mathcal{N}(d_i, 0.3^2)$, $i = \{1, 2\}$. The reliability index is set at $\bar{\beta}_1 = \bar{\beta}_2 = \bar{\beta}_3 = 3$. The corresponding failure probability is $P_f = 0.0013$. The two aforementioned references solve this problem considering different approaches. Some of them are metamodel-free.

Results and comparison

We start with an initial design of 15 points. For the first stage of enrichment, setting the

threshold to $\bar{\eta} = 0.15$ leads to adding 19 points to that design. The evolution of the accuracy criterion is shown in Figure 5.2. It decreases steadily. In fact, when looking at the history of the added points, one can observe that the enrichment focuses too much on one single constraint, *i.e.* \mathfrak{g}_3 . Figure 5.3a illustrates this. The blue triangles represent the initial DOE, the red squares the added points and the black crosses, the points remaining in the set \mathfrak{C}_2 (Eq. (4.40)). The contours $\{d \in \mathbb{D} | \mathfrak{g}_i(d) = 0\}$ are plotted respectively in blue, red and yellow for $i = \{1, 2, 3\}$. The dotted lines represent the associated quantiles q_{α_i} . The constraints \mathfrak{g}_1 and \mathfrak{g}_2 are actually accurately approximated. Despite many points were added around \mathfrak{g}_3 , its accuracy is not sufficient. It is not even active in the optimization problem. So in the second stage there is no enrichment as the surrogate models are accurate enough in the vicinity of the optimal solution. Figure 5.3b shows the points sampled by CMA-ES.



Figure 5.2: Evolution of the global enrichment criterion. The red line shows the threshold $\bar{\eta}$.

The solution found here is compared with those in the aforementioned references as shown in Table 5.2. The brute-force approach consists in optimizing with the true constraints considering Monte Carlo simulation with 10^4 samples for the estimation of the quantile. Except in Dubourg (2011), the other approaches do not rely on surrogate models. PMA and SORA clearly lead to a larger number of calls to the true models. For the remaining cases, the quantile-based approach has a larger number of calls to the true models. This can be explained by the enrichment strategy. As mentioned above, the global enrichment focuses too much on a function which is not even active. For instance, Dubourg (2011) directly considers the enrichment during optimization. Thus, the author only enriches once (adding simultaneously 10 points) on the first constraint. This is enough to converge. It appears that our strategy is not optimally tuned for this specific example. For instance, considering a higher threshold, say $\bar{\eta} = 0.5$, leads to converge with less than 20 calls to the true functions without any loss in the accuracy of the found optimum.



(a) Enrichment points for the Choi problem: The enrich- (b) Convergence of CMA-ES. The green points are the sucthe remaining points in the set \mathfrak{C}_2

ment points are shown in red whereas the blue ones cessive best points. The blue one are admissible points that represent the initial DOE. The small black crosses are did not improve the best point at the moment they were sampled. The red points are unfeasible sampled designs.

Figure 5.3: Illustration of the Choi problem

Method	d_1^*	d_2^*	$\mathfrak{c}\left(oldsymbol{d}^{st} ight)$	g-calls
Brute force	3.45	3.30	6.75	$pprox 10^{6}$
PMA^1	3.43	3.29	6.72	1,551
SORA ²	3.44	3.29	6.73	151
Single loop ³	3.43	3.29	6.72	19
RDS^1	3.44	3.28	6.72	27
Meta-RBDO ⁴	3.46	3.27	6.74	20(20/10/10)
Quantile-RBDO	3.45	3.29	6.74	34

Table 5.2: Results comparison for the Choi problem.

As calculated in Shan and Wang (2008)
 As calculated in Du and Chen (2004)
 As calculated in Liang et al. (2004)

⁴ As calculated in Dubourg (2011)

5.2.3 Bracket structure

Problem definition

A two-member bracket structure described in Section 2.3.3.2 is considered for this last mechanical validation example. The mechanical problem was introduced in Chateauneuf and Aoues (2008) (in the book by Tsompanakis et al. (2008)) and later exploited by Dubourg (2011). A sketch of the problem is shown in Figure 5.4. Beside from their own weight, the two members which are pin-joined at a point *B*, support a vertical load *P* applied at a distance *L* from their left ends.



Figure 5.4: A sketch of the bracket structure (as illustrated in Dubourg (2011))

The aim is to minimize the structure weight. Two failure modes are considered:

• The maximum bending stress in the member *CD* should be lower than the yield strength f_y . The resulting limit state function reads:

$$\mathfrak{g}_1\left(\boldsymbol{d},\boldsymbol{z}\right) = f_y - \sigma_b,\tag{5.4}$$

where $\sigma_b = 6M_B / w_{CD} t^2$ is the maximum bending stress with $M_B = PL/3 + \rho g w_{CD} t L^2 / 18$. In these equations, w_{CD} and t are respectively the width and height of the beam *CD* and ρ is the unit mass of its constitutive material.

• The compression force F_{AB} in the bar AB must be lower than the critical Euler force F_b :

$$\mathfrak{g}_2\left(\boldsymbol{d},\boldsymbol{z}\right) = F_b - F_{AB},\tag{5.5}$$

where

$$F_b = \frac{\pi^2 EI}{L_{AB}^2} = \frac{\pi^2 Et w_{AB}^3}{12 \left(2L/3 \sin\theta\right)^2},$$
(5.6)

and

$$F_{AB} = \frac{1}{\cos\theta} \left(\frac{3P}{2} + \frac{3\rho g w_{CD} t L}{4} \right).$$
(5.7)

147

Here w_{AB} and L_{AB} are respectively the width and length of the bar AB and θ is its inclination angle as shown in Figure 5.4.

Considering these two failure modes, the optimization problem reads:

$$d^* = \arg\min_{d \in [5,30]^3} \rho t L \left(\frac{4\sqrt{3}}{9} w_{AB} + w_{CD}\right) \quad \text{subject to:} \quad \begin{cases} \mathfrak{f}(d) = w_{AB} - t \leq 0\\ \mathfrak{g}_1(d) = \sigma_y - \sigma_b \geq 0\\ \mathfrak{g}_2(d) = F_b - F_{AB} \geq 0 \end{cases}$$
(5.8)

where $d = \{w_{AB}, w_{CD}, t\}^T$ and $z = \{P, E, \sigma_y, \rho, L\}^T$ are the vectors gathering respectively the design and environmental variables.

In this example, we consider the case where both the design and environmental variables are considered random. The probabilistic model is reminded in Table 5.3. The coefficient of variation of the design variables are kept constant throughout the optimization.

Parameter	Distribution	Mean	$\text{COV}\left(\delta\%\right)$
Width of AB (w_{AB} in m)	Normal	w_{AB}	0.05
Width of CD (w_{CD} in m)	Normal	w_{CD}	0.05
Thickness (t in m)	Normal	t	0.05
Applied load (<i>P</i> in kN)	Gumbel	100	0.15
Young's modulus (E in GPa)	Gumbel	200	0.08
Yield stress (f_{y} in MPa)	Lognormal	225	0.08
Unit mass (ρ in kg/m ³)	Weibull	7,860	0.10
Length (<i>L</i> in m)	Normal	5	0.05

Table 5.3: Parameters of the variables defining the probabilistic model for the bracket structure.

Results and comparison

The initial setting consists of an optimal Latin hypercube design of size 50 for this 8-dimensional problem. In the first stage of enrichment, 10 points are added per iteration. The threshold is set to $\bar{\eta} = 0.30$. Its value has been increased to account for the results above where a too low value of $\bar{\eta}$ has lead to refine too much in regions of the space where the objective function is far from optimal. With this threshold, the design of experiments is enriched six times as illustrated in Figure 5.5, corresponding to the final value of η being 0.27. This sums up to 110 points at the end of the first stage of enrichment.

We then start the optimization with the constrained (1 + 1)-CMA-ES whose solution is refined by Matlab fmincon function. During CMA-ES, the Kriging models are refined 15 times. In this example, the thresholds are $\bar{\mathfrak{g}} = 0$, so the criterion $\eta_q = (q_{\alpha}^+ - q_{\alpha}^-) / \bar{\mathfrak{g}}$ introduced in Eq. (4.42) is not applicable. Therefore, we replace the denominator by the value of the quantile so as to quantify the relative width of the margin of uncertainty with respect to the quantile estimate itself. The criterion therefore reads:

$$\widetilde{\eta}_{q}\left(\boldsymbol{d}\right) = \frac{\widehat{q}_{\alpha}^{+}\left(\boldsymbol{d}\right) - \widehat{q}_{\alpha}^{-}\left(\boldsymbol{d}\right)}{q_{\alpha}\left(\boldsymbol{d}\right)}.$$
(5.9)

Figure 5.6 shows this criterion together with the three thresholds. In the first iterations, the criterion is quite high, but after five enrichments, its average value drops. However, in the last



Figure 5.5: Evolution of the convergence criterion for the first stage of enrichment. The red line shows the threshold $\bar{\eta}$.

iterations, some values of $\tilde{\eta}_q$ are extremely high. This is due to the optimizer which is in the vicinity of the limit state surface where the optimum lies. The associated values of q_{α} are therefore close to zero. Despite this noise, the optimization problem converges to a solution close to the brute-force approach (see the reference solution computed in Section 2.3.3.2). However, the number of calls to the true function is small compared to the brute-force approach.



Figure 5.6: Evolution of the convergence criterion for the second stage of enrichment. The red line shows the threshold $\bar{\eta}_q$.

In total, only 125 calls to the true limit state functions have been necessary. This is slightly smaller from the one of Dubourg (2011) and much smaller than that of Chateauneuf and Aoues (2008). It should be however stressed that the latter results did not rely on surrogate models. These results are summarized in Table 5.4. Once again, the design we found is fundamentally different from that of the other two references, but rather close to the brute force solution. Here again this might be due to the fact that we use a global search algorithm (CMA-ES) in contrast to the benchmark solutions which rely on local search methods.

Table 5.4: Comparative results for the bracket structure. The PMA result comes from Chateauneuf and Aoues (2008) and Meta-RBDO from Dubourg (2011).

Design method	Weight (kg)	w_{AB} (cm)	w_{CD} (cm)	<i>t</i> (cm)	g-calls
Brute force	1357	5.35	7.40	30.00	$pprox 10^{6}$
PMA^1	1673	6.08	15.68	20.91	2340
Meta-RBDO ²	1584	5.80	12.80	23.30	160(160/90)
Quantile-based RBDO	1364	5.57	7.28	30.00	125

¹ As computed by Chateauneuf and Aoues (2008)

² As calculated by Dubourg (2011)

5.3 Toward an industrial application

The industrial application of the proposed methodology concerns the lightweight design of an automotive body structure. As a preliminary stage, we first introduce elements of fast dynamic simulations so as to shed light on the specificity of the problem that we are addressing. An insight to numerical and physical scatter is given through case studies of two different models. The first one concerns a so-called *side-member subsystem* and the second one comprises an entire car under frontal impact.

5.3.1 Elements of non-linear dynamic analysis

Before the application, let us introduce some elements of non-linear dynamic analysis. In this PhD work, the finite element simulations are used as black-box functions. Thus, we only present the fundamental equations. More detailed considerations are beyond the scope of the present work. However, we enumerate some points that might explain the difficulties encountered when building surrogate models as approximations of the FE responses.

The crash behavior of a car is governed by general differential equations. The discretized equation of motion without accounting for damping reads (Altair Engineering, 2009; Du Bois et al., 2004):

$$M\ddot{u} + Ku = f_{ext},\tag{5.10}$$

where *M* and *K* are respectively the inertia and stiffness matrices, \ddot{u} is the nodal acceleration vector and f_{ext} is the sum of external forces. The internal nodal forces here are given by $f_{int} = Ku$. To solve this problem, the so-called *Newmark*- β *method* is adopted. With this integration scheme, the nodal displacement and the first-order derivative (velocity) at the time step *n* is

generally given by:

$$u_{n} = u_{n-1} + \delta t \, \dot{u}_{n-1} + \delta t^{2} \left(1/2 \, \ddot{u}_{n-1} + \beta \left(\ddot{u}_{n} - \ddot{u}_{n-1} \right) \right),$$

$$\dot{u}_{n} = \dot{u}_{n-1} + \delta t \left((1-\gamma) \, \ddot{u}_{n-1} + \gamma \ddot{u}_{n} \right),$$
(5.11)

where δt is the time increment, γ and β are some parameters of the integration scheme.

In the context of crash simulation, the so-called *explicit formulation* is considered. In such a technique, the equilibrium is expressed at a time where all nodal displacements are known (Du Bois et al., 2004). In other words, at a given time step, nodal positions are computed as a function of quantities from the previous step. Explicit formulation is achieved when the two parameters of the Newmark scheme take the values $\beta = 0$ and $\gamma = 1/2$, thus leading to:

$$u_{n} = u_{n-1} + \delta t \dot{u}_{n-1} + 1/2 \, \delta t^{2} \ddot{u}_{n-1},$$

$$\dot{u}_{n} = \dot{u}_{n-1} + 1/2 \, \delta t \left(\ddot{u}_{n-1} + \ddot{u}_{n} \right),$$

$$M \ddot{u}_{n} = f_{ext} - f_{int}.$$
(5.12)

This formulation has the advantage of being fast since only a system of uncoupled equations has to be solved. In contrast, the implicit formulation requires to solve a matrix problem. Additionally, a lump-mass approach is used and this means that there is no matrix to invert at all.

However, this approach also comes with some drawbacks. First, the problem as formulated is only conditionally stable contrary to an implicit formulation which is always stable. This conditional stability means that the analysis time steps should not be higher than the so-called *critical time step* whose value is given by the Courant condition and reads:

$$\delta t < \frac{l_c}{c},\tag{5.13}$$

where δt is the analysis time step, l_c is the characteristics element length which depends on the shape and size of the element and c is the sound velocity throughout the material, given by $c = \sqrt{E/\rho}$, with E and ρ being respectively the Young's modulus and density of the material.

The instability occurs when the Courant condition is not respected *i.e.* when a stress wave propagates across more than one element in a given time step. The critical time step is directly related to the spatial discretization of the model (l_c) and the material used (E and c). These three levers may be tuned to avoid too small time steps and consequently time-consuming simulations. Increasing l_c is not a good option as it requires to remesh the model without accurately fitting the shape of the components. Modifying E is not really an option. Hence, the common procedures recommend to artificially increase the density so as to keep the time step at an acceptable level. This however modifies the kinetic energy of the car and therefore, it is required to check locally and globally the weight increase in the model before any post-processing. When a threshold is reached, the computation is considered not valid.

Second, the transmission of forces all over the car through body parts is ensured by the definition of contacts. This is considered in the models by setting up rules to avoid interpenetrations of individual parts. These are the so-called *master-slave conditions* where sets of nodes, edges or surfaces are not allowed to cross predefined contact areas. In the event the distance between a slave and master entity is less than a given threshold, herein called *gap*, a spring between them is activated. This spring is annihilated as soon as the entities get again far enough from each other. The difficulty associated to the contact definition is on the one hand the setting of the spring stiffness. Too flexible a spring may result in large penetrations and kinematic discontinuities while a too stiff spring might lead to numerical instabilities. On the other hand, detecting the contact is not a straightforward task and the algorithms used for that are not always fully efficient.

Beside these numerical difficulties, frontal impact is known to be a chaotic phenomenon. It can be schematically expressed as bifurcations triggered by initial conditions which lead to dramatic variations of the responses for small changes in the inputs. Many studies have been done in order to identify the origins of the instability in crash simulations (Wauquiez and Zeitouni, 2008; Thole and Mei, 2003). They can be mainly grouped into physical and numerical causes (Roux et al., 2006).

The physical scatter is mainly due to the geometric imperfections during the manufacturing process and non-repeatability of the crash tests, *e.g.* variability in the thicknesses of car body parts, position of the barrier or car speed during the crash tests. For robust design, these scatters are taken into account in the simulations by representing the input parameters and initial conditions as random variables.

These physical variabilities when considered in the simulation are amplified by the numerical instabilities. In fact, different bifurcations in the process may be triggered by one or another initial condition and lead to different solution paths. The typical example is the *on-off impact*, *i.e.* whether a contact between two parts occur or not or even a slight change in the chronology of events may lead to noticeable different crash scenarios. In the same way, buckling of parts such as the sidemember which may bend in a given direction or crush axially may alter the crash scenario. On top of that, there are non-physical parameters, such as spatial and time discretization, contact detections, round-off errors which add to the variability of the outputs.

5.3.2 Numerical noise in frontal impact

In this section, we study the numerical noise in a frontal impact. To this end we introduce the so-called *side-member subsystem*. The side-member is actually a part in the front end of the car whose main objective is to absorb the kinetic energy of the crash by proper crushing. Therefore the subsystem introduced here consists of a collection of parts in the front end of the car including the side-member. The parameters of this model are tuned so as to achieve the same level of numerical noise as in a entire car model. Its main asset is that it is less computationally intensive. One model run lasts only 10 to 15 minutes (in distributed CPUs) in contrast to that of an entire car which may last up to 24 hours. This allows us to perform Monte Carlo based studies without worrying much about the reduced computational time.

The finite element model consisting of the sidemember subsystem and a barrier is illustrated in Figure 5.7. For optimization purpose, five parts are considered as annotated in the figure.

The usual approach using the Radioss software to perform a numerical scatter analysis is to randomly move each node of the mesh in all directions with a given amplitude. The amplitude varies from one node to the other. So each displacement follows a uniform distribution $\mathcal{U}(-10^{-6}, 10^{-6})$ mm. We consider for this case study 10,000 such models.

The first question that comes to mind is how distributed are the outputs. A total of 29 outputs is available for each run of the model. We illustrate only a few. Figure 5.8 shows the resulting distributions for four selected outputs: maximum sidemember compression, maximum contact



Figure 5.7: Side-member subsystem problem finite element model

force, maximum force along the left sidemember, maximum force along the right sidemember. At first sight, the distributions seem different, even when looking within groups of similar outputs (forces, intrusions or accelerations).

In general, the distributions look Gaussian. For instance a Kolmogorov-Smirnov test fails to reject the null hypothesis of Gaussian distribution at 5% significance level for the fourth output plotted (Maximum forward right sidemember force). However this hypothesis is rejected for three other outputs. Their empirical cumulative distribution functions (when normalized) are quite close to the standard Gaussian CDF though. This is illustrated in Figure 5.9. In these figures, u_i are the normalized values of y_i , *i.e.* $u_i = (y_i - \mu_{y_i})/\sigma_{y_i}$.

The issue of checking Gaussianity is the following. Kriging allows one to account for noisy observations by including noise variance at each observation (Roustant et al., 2012). If we assume the noise to be Gaussian, we can use this knowledge when building the Kriging models. Despite Kriging may assume heterogeneity of noise at the observations, running a numerical scatter analysis at each DOE point would be cumbersome. The second question that arises is whether the noise is the same in all the design space or not. If it is the case, one would compute this variance once and for all in the early stage where the finite element model is validated, and use this value as input for the Kriging nugget.

To answer this question, we consider two configurations of the model. The five parts highlighted in Figure 5.7 are considered. In the first configuration, the nominal values of the thicknesses are considered. These are the default values in the above Monte Carlo sampling. The second configuration is based on an optimization result presented in Moustapha et al. (2014b). The five parameters are listed in Table 5.5.

The same Monte Carlo analysis is performed with the two configurations. We emphasize that the seeds for the random number generation are the same, thus identical perturbations are made on the mesh nodes in both cases. The size of the MCS is reduced to 500. An empirical analysis showed that this is enough to obtain a reasonable estimate of the noise variance. Distributions of the four selected outputs are shown in Figure 5.10. The left column corresponds



Figure 5.8: Numerical scatter analysis on the sidemember subsystem problem: Histogram of four outputs of interest.

Table 5.5: Parameters values for the two configurations used to perform numerical scatter analysis on the sidemember subsystem.

Configuration	d_1	d_2	d_3	d_4	d_5
Nominal	2.00	1.80	2.00	1.70	0.65
Optimal	2.16	1.50	2.00	1.93	0.60

to the nominal design and the right one to the optimum. In most cases, the distributions look similar. However, the parameter values differ. There is a shift in the mean values which is expected since we have changed the design. However, the coefficients of variation in some cases are significantly different. For extreme values of design parameters, the variation can be even more pronounced. This can be explained by the fact that the quality of the finite element



Figure 5.9: Comparison of empirical and standard Gaussian CDF for the four outputs of interest.

models are fine-tuned for ranges of parameters around the nominal model. For parameters too far from the nominal values, the numerical noise may be considerably larger.

Table 5.6: Parameters values for the two configurations used to perform numerical scatter analysis. The two rows between lines correspond to the same output.

Output	mean	std. dev.	COV (%)
(12, (12, 12, 2))	506.20	4.40	0.87
y_1 (mm)	523.11	1.28	0.25
<i>y</i> ₂ (kN)	166.47	10.66	6.40
	151.90	0.50	0.33
(kN)	14.35	0.01	0.06
<i>y</i> ₃ (KIN)	14.35	0.01	0.04
<i>y</i> ₄ (kN)	121.28	0.06	0.05
	132.00	0.18	0.14



Figure 5.10: Comparison of outputs distributions with two configurations of the model for the numerical scatter analysis of the sidemember subsystem.

In summary, the numerical noise in this model can be assumed to be Gaussian. Furthermore it is *heteroscedastic, i.e.* the dispersion is not the same for all design points. Thus, to account for numerical noise, one cannot consider a Kriging model with homogeneous nugget variance, although this would be a first stage in accounting for this noise. We may also think of learning the noise through automatic calibration of the nugget. However, this requires either to fill a space with such a sample density we cannot afford or to repeat observations (including the numerical noise), something we cannot afford either. Finally, we must stress here that the absolute value of the numerical noise variance is not that high when compared to the variability of the output in the entire design space. Besides, as we show in the sequel, the physical scatter is also to be accounted for. In fact, this one does encompass the numerical noise. The physical scatter can be accounted for through reliability-based design optimization. The quantile-based approach we propose in this thesis is aimed at giving conservative values to these noisy constraints.

5.3.3 Frontal impact crash scenarios

The ultimate aim of this thesis is to add frontal impact in the current optimization so as to increase the potential weight savings. In the current process where all specifications but the frontal impact-related ones are considered, the parts selected for optimization shall have little or no influence on frontal impact results. Thus, parts in the front end of the car are avoided. An example of optimization parts for the current process is given in Figure 5.11a. When integrating frontal impact to the optimization, the increase of weight savings may only be achieved by enlarging the perimeter of optimization parts. To this end, parts belonging to the front end are considered. In the example below, such parts belong in the area encircled in red in Figure 5.11b.



Figure 5.11: Extension of the number of parts before including frontal impact. In the left panel, the initial set of parts belong to the passenger compartment. The set of parts encircled in the right panel are those added to the initial one when frontal impact is included in the multidisciplinary optimization.

The effects of the chaotic nature of frontal impact are amplified when considering such parts as those belonging to the front end. In this section, we study closely these effects with the set up presented above. We consider a model with the initial optimization parts shown in Figure 5.11a

to which we add some of the parts in the circled area of Figure 5.11b. The final problem consists of 37 design parameters, when considering symmetries and linked parts. A 132-point optimal Latin hypercube makes the initial design of experiments.

Let us now consider a particular output quantity which perfectly illustrates the bifurcations we have mentioned. We call this criterion, which is defined in terms of a displacement at the front end of the sidemember, y_B . To illustrate our point, let us consider the time-history plots for two designs of experiments based on the two configurations. They are shown in Figure 5.12. It is clear in these figures that in the first case (Figure 5.12a), where no parts belonging to the front end are considered, only one crash scenario can be identified. However in the second case (Figure 5.12b) which is of interest to us, the behavior is more scattered and one can identify two groups of curves. In general, the designer looks at the output value at 40 ms, *i.e.* $y_B = y_B^{\text{th}}$ (40). In this case, we can distinguish the two sets of curves.



Figure 5.12: Time history plots related to y_B for the two configurations *i.e.* sets of parts for optimization without (left) and with (right) frontal impact.

Figure 5.13 shows the distributions of the runs with respect to each group. Let us call A and B respectively the upper and lower sets. The presence of these distinct groups confirm that, according to the region of the design space, the car response to frontal impact is significantly different. This may explain the difficulties in building surrogate models for frontal impact-related constraints. As pointed out in the benchmark in Moustapha et al. (2014a) numerical noise alone does not explain the difficulties in building accurate metamodels. In the sequel, we introduce a methodology to account for these groups of crash scenarios. In short it consists in three steps:

- 1. Unsupervised classification: The different groups are identified through *K*-means clustering. This algorithm partitions the points with respect to their relative distances (in the output space). For this example, the two groups are shown in Figure 5.13.
- 2. Supervised classification: In order to build a single surrogate model for each group, it is necessary to know a priori to which group a new point belongs. To this end, a support



Figure 5.13: Distribution of y_B with respect to the run number for the set of parts for which frontal impact is also considered. The blue triangles and black squares respectively represent the points belonging to group A (class +1) and group B (class -1).

vector classifier is built with the initial design of experiments.

3. Regression: Eventually, a surrogate model is built for each group. At this point, either Kriging or support vector regression can be used.

5.3.3.1 Application with available data (Initial experimental design)

We now apply the approach presented above with the data available, *i.e.* the 132 initial simulation results. To validate the results, we consider a validation set consisting of 125 points (resulting from an optimal Latin hypercube design used in another study).

On the one hand we build a single and global surrogate model, *i.e.* as in the usual approach. On the other hand, we build one surrogate model for each class. Unfortunately, with the available data, the group B is too poorly populated to build a sound metamodel (27 points in a 37-dimension problem). Thus in a first step, we only consider the group A. The results are shown in Figure 5.14. In the left panel, the true versus the predicted values are plotted. The blue and black colors stand respectively for global and local (*i.e.* on group A) metamodels. The right panel shows histograms of the gap between the two approaches with the dataset of group A only. It is clear from these figures that focusing on one group by building a surrogate model increases the accuracy of this latter. This motivates the approach we have proposed above.



(a) True vs predicted responses by a Kriging model



(b) Histograms of the predicted responses with respect to training points consisting of group A only (blue) and all data points (black).

Figure 5.14: Comparison of metamodeling by considering all training points or only a those belonging to the class +1.

We now proceed to build a support vector classifier on the available basis. The prediction by this classifier on the validation set is shown in Figure 5.15. As mentioned earlier, the squares and triangles stand respectively for groups A and B. The green color shows points that are well classified and red the misclassified ones. Out of 125, 11 are in the latter situation, *i.e.* misclassified. Despite being relatively few (less than 10%), such misclassifications are problematic since the two metamodels are completely different.

To summarize, two limitations have been raised at this stage:

• Unbalanced groups: One behavior occurs much more often than the other. The training set is made of 107 sample points of group A and 25 of group B. In the same way, the



Figure 5.15: Classification of the outputs by support vector machine on a separate validation set. The squares and triangles respectively represent groups A and B. Points drawn in red are misclassified by the SVC model.

validation set consists of 100 points from group A and 25 from group B. In both cases, the datasets were obtained by space-filling sampling. It is therefore difficult, if not impossible, to build accurate surrogate metamodels for the smaller group. In such a case, the idea of local surrogate modeling loses much of its relevance.

• Classification errors: With a limited number of experiments, it is not straightforward to obtain a 100% accurate classifier. This means that during the evaluation of new points, we are prone to errors as we might use the wrong surrogate model for a given point.

5.3.3.2 Application with enrichment of the initial dataset

To by-pass the two limitations highlighted above, we propose to enrich the initial dataset. To this end, two aims are simultaneously sought after:

- Improving the quality of the classifier;
- Adding points specifically in the less populated group so as to reach the critical size necessary to build a sound surrogate model.

The enrichment here is made in two stages as well, so as to fulfill each of the two objectives above. The first stage is aimed at improving the classifier only. To achieve this, we rely on one major property of support vector machines. Let us consider a design of experiments with two identified classes: $\mathcal{D} = \{(x_i, y_i), x_i \in \mathbb{R}^s, y_i = \{-1, 1\}, i = 1, ..., n\}$. The classifier as intro-

duced in Eq. (3.24) for a new point reads:

$$\widehat{y}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i k\left(\mathbf{x}_i, \mathbf{x}\right) + b, \qquad (5.14)$$

where α_i 's are the support vector coefficients, *k* is the kernel function and *b* is the offset parameter.

The classification rules state that if $\hat{y}(x) > 0$, then x belongs to the class labeled +1 and conversely if $\hat{y}(x) < 0$, x belongs to the class labeled -1. The points for which $-1 < \hat{y} < 1$ constitute the margin, somehow a region of uncertainty. The support vectors, which are the only points in the training set that are actually relevant to the classifier, belong to this margin. Thus, to improve the classifier, points for enrichment should be selected within this margin. We consequently sample points for enrichment uniformly in this margin. This can be achieved in three steps:

- 1. Sample a large number of points in the entire space, say $N_{SVC} = 10^4$;
- 2. Evaluate them with the support vector classifier and select only those that are in the margin, *i.e.* the set $\{x \in \mathbb{R}^s | -1 < \hat{y}(x) < 1\}$;
- 3. Reduce this set to *K* points thanks to *K*-means clustering.

The *K* points are used to update the design of experiments. These steps are repeated until the classifier is judged accurate enough. This is considered to be true when the predictions on a large number of points among a given set does not change between two iterations. In practice, we sample a large Monte Carlo set. At each iteration, we keep track of the percentage of points whose predictions signs change (*i.e.* which move from one class to the other according to the classifier). When this percentage is small enough and does not evolve, we consider that adding points will not improve significantly the classifier and stop the iterations.

Once this first stage of enrichment is over, we sample uniformly points in the less populated group to balance the two classes.

Illustration example

To illustrate this methodology, we consider a two-dimensional problem where the space is divided into two classes as follows:

$$\mathbf{x} \in [0,1]^2 : \begin{cases} y(\mathbf{x}) = -1 & \text{if } x_2 < f(x_1) = x_1 \sin(x_1) + 1, \\ y(\mathbf{x}) = 1 & \text{if } x_2 \ge f(x_1) = x_1 \sin(x_1) + 1. \end{cases}$$
(5.15)

Figure 5.16 illustrates the problem in its initial setting. The green line is the separator function $f(x_1) = x_1 \sin(x_1) + 1$. As shown by the figure, the space devoted to one class dominates the other in terms of area. The initial design of experiments consists of an OLH design of 20 points. The built support vector classifier is shown as the dashed black line. The associated margin is bounded by the blue and red dashed lines. The support vectors are those points on the margin encircled in cyan. The gray-shaded mapping shows the prediction of the SVC model on the input-space. The brighter the color the higher the value of $\hat{y}(x)$.

We first proceed by adding iteratively three points per iteration in the margin of the successive models. To assess the convergence of the classifier, we consider a Monte Carlo sample set of 10,000 points. We then compute the relative number of points that change signs w.r.t. two successive classifiers. The evolution of this percentage, denoted pm is shown in Figure 5.17a. This


Figure 5.16: Classification in a two-dimensional space with the function $x_1 \mapsto x_1 \sin(x_1)$. The blue triangles are the training points. The black dashed line is the classifier which should approximate the true boundary shown by the green line. The red and blue lines correspond respectively to the lower and upper SVM margin. Support vectors are encircled in cyan. Outputs of the SVC model is represented by the gray-shaded scale in the background: the lighter the color the higher the value.

first stage of enrichment is stopped at iteration 7 since the classifier does not evolve perceptibly between the 6th and 7th iterations.

By the end of the first stage, the design of experiments consists of 32 points, among which 22 belong to class +1 and merely 10 to class -1. In the second stage, only class -1 is enriched so that its size is big enough for a proper metamodeling by regression or interpolation. To achieve this, we first sample a large Monte Carlo set in the design space. Then we select the subset defined by $\{x \in \mathbb{R}^s | \hat{y}(x) < 0\}$. This subset is statistically reduced to 15 points by *K*-means clustering. Within these points, three turn out not to belong to class -1 as predicted by the classifier. Thus the final design of experiments consists of 47 points, among which 25 belong to class +1 and 22 to class -1. Figure 5.18 shows the final setting. Compared to Figure 5.16, the SVC model has been improved and the distribution of points between the two classes is almost even.



(a) Evolution of the accuracy criterion for the SVC models

(b) State at the end of the first stage of enrichment

Figure 5.17: First stage of enrichment to improve the SVC model. In the right panel: The blue triangles are the training points. The black dashed line is the classifier which should approximate the true boundary shown by the green line. The red and blue lines correspond respectively to the lower and upper SVM margins. Support vectors are encircled in cyan. Outputs of the SVC model is represented by the gray-shaded scale in the background: the lighter the color the higher the value.



Figure 5.18: Final state after two stages of enrichment. The blue triangles are the training points. the black dashed line is the classifier which should approximate the true boundary shown by the green line. The red and blue lines correspond respectively to the lower and upper SVM margins. Support vectors are encircled in cyan. Outputs of the SVC model is represented by the gray-shaded scale in the background: the lighter the color, the higher the value.

Application on a vehicle model

We now return to the case study where we first considered to build surrogate models by group. In the present version, the methodology above is applied before building the surrogate models. As a reminder, the initial DOE consists of 132 points, among which 27 belong to class -1 (group B). In the first stage of enrichment, 10 points are added per iteration. This step is stopped after 120 points have been added to the initial DOE. Convergence, as shown in Figure 5.19, is not strictly achieved since the number of points changing signs is not very stable between iterations. The stopping is rather due to reaching the allocated computational budget.



Figure 5.19: Evolution of the enrichment criterion in the second stage.

After this first stage of enrichment 42 points are added in group B to increase its size. In total, the final DOE consists of 294 points with the following distribution: 172 in group A and 122 in group B. This is shown in Figure 5.20 where the different stages are separated by the vertical lines (*i.e.* before enrichment and after stages 1 and 2 of enrichment). In the first stage of enrichment, the points from the two classes are closer with respect to y_B . This is because they were selected in the margin of the classifier. In the second stage, among the 42 points selected for enrichment of group B, four were actually in group A.

To build the surrogate models by group, we start by predicting the class of each point of the validation set. Figure 5.21 shows the two groups. The squares and triangles respectively stand for group A and B. In green are the points that are well classified by the SVC model. In this example, there is a gain in prediction accuracy. However, not all data are well classified as pointed out by the four red points. This is not surprising given that the boundary between the two classes is not clear-cut. In fact, it is sometimes difficult to decide to which group belongs a simulation, even after analyzing the time-history plot.



Figure 5.20: Classification of the different runs in the validation set after the two stages of enrichment. The black squares and blue triangles correspond respectively to groups A and B.



Figure 5.21: Classification of the outputs by support vector machine on a separate validation set after the two stages of enrichment. The squares and triangles respectively represent groups A and B. Points drawn in red are misclassified by the SVC model.

Once we have predicted groups, we build one Kriging model for each class. Each point in the validation set is evaluated with the corresponding Kriging model. To validate the methodology, we also build a single Kriging model. We therefore have three Kriging models:

- $\widehat{\mathcal{M}}_A$: a model built on group A which consists of 172 training points;
- $\widehat{\mathcal{M}}_B$: a model built on group B which consists of 122 training points;
- $\widehat{\mathcal{M}}_G$: a global model built considering the 172 + 122 = 294 points altogether.

The true model versus predicted values on the validation set χ is shown in Figure 5.22. χ_A and χ_B are the points of the validation set that were predicted to belong respectively to group A and B. It can be seen that local metamodeling globally improves the accuracy of the prediction. The computation of the normalized mean-square error may confirm this. We should emphasize here that the global surrogate model was built with 294 points whereas the local with only 172 and 122 points respectively. Despite this, the local surrogates are more accurate. The four misclassified points are clearly predicted as outliers of their respective groups. To lower the amplitudes of such errors, it might be interesting to consider weighted Kriging models where the weight given to each model is somehow proportional to the confidence associated to the SVC model prediction at a given point.



Figure 5.22: Comparison of the true and predicted responses by the Kriging models with the final set of points.

5.3.3.3 Concluding remarks

In this section, we have studied in details the specificities of frontal impact simulations. Two case studies were concerned. In the first one, numerical scatter on the side-member subsystem

was investigated. It appears that the numerical noise on the outputs is close to Gaussian. Besides, it is heteroscedastic, *i.e.* its amplitude does not remain the same when moving from one design to another. This point invalidates the possibility of feeding the Kriging model with a known homogeneous nugget variance. Furthermore, we cannot learn this noise through automatic calibration of the Kriging nugget because of the limited computational budget. From a broader perspective, it can be shown that the numerical noise is much smaller than the physical one. Therefore, by adopting a probabilistic approach where only variability in the physical parameters is considered, we might include the effects of numerical noise. To this end, the quantile-based approach proposed in this thesis is applied for weight optimization of cars. The quantile is considered here simply as a conservative measure of feasibility which accounts for the noise of different sources that can be encountered when simulating frontal impact. It shall not be seen as a failure probability of designed cars from a frequentist point of view.

The second study was done with the aim of improving accuracy of the surrogate models that are built to approximate frontal impact-related constraints. By experience, it is known that many crash scenarios are possible according to the input parameters. In this example, two groups were identified. They correspond to distinct physical behaviors, *i.e.* the deformations of the parts in the front end follow two different patterns. We show that including this knowledge when building Kriging models allows one to improve the prediction accuracy. As such, this methodology cannot be combined with RBDO for frontal impact as the overall computational budget would be too high. However, there are other disciplines where classification of different behaviors can be used. For instance, we may consider that engineers would like to avoid one of the behavior. In such a case, an additional constraint in optimization may help avoiding unwanted scenarios. The typical example is the lightweight design of a subsystem with respect to seatbelt anchorage. In such a study, some configurations of initial designs lead to the breaking of the seatbelt anchorage. This produces unrealistic efforts which stand as outliers, thus perturbing the optimization. Classification of the two scenarios (*i.e.* break and non-break) may help to avoid this situation.

5.4 RBDO application on a vehicle frontal impact

Safety is one of the main concerns in automotive design. Frontal impact being among the most frequent and dangerous types of accidents, a particular attention is given to it. Various regulations (according to the targeted market) provide some minimal crash tests that must be successfully passed for approval of new vehicles. They aim at ensuring that the driver and passengers are well protected from the steering mechanism in the event of a frontal impact. Besides, there also exists consumerist tests among which the European New Car Assessment Program (EuroNCAP) that aims at stimulating competition between car manufacturers. The idea is to foster continuous efforts toward increasing safety by rating the cars (in a five-star scale). The applications we present below falls into this category.

Two models are considered. First, we apply the methodology to the sidemember subsystem introduced earlier in this chapter. Thanks to the reduced CPU time of the associated model run, we can validate the approach by actually computing quantiles on the found optimum. The second example is a full-scale car model with industrial settings. It should be noted here that the numerical values are somewhat modified for confidentiality reasons.



Figure 5.23: Euro NCAP frontal impact test of a Peugeot 308 (Courtesy of Euro NCAP).

5.4.1 Application on the sidemember subsystem

5.4.1.1 Problem formulation

The finite element model used for this example is shown in Figure 5.24a. Five parts are considered for optimization as shown in this figure. The initial design and the bounds of the design space are illustrated in Figure 5.24b. In this configuration the subsystem weighs 9.67 kg. Two parameters among the crash protocol are considered important enough to be included as environmental variables. These are respectively the initial speed and the lateral position of the barrier. The probabilistic model is defined with the following distributions:

- Initial speed: $V \sim \mathcal{U}(34, 35) \text{ km/h};$
- Lateral barrier position displacement: $P \sim \mathcal{N}(0, 2)$ mm.

Two constraints are considered for the optimization problem: the maximum wall force which must be kept below 170 kN and the maximum sidemember compression whose value must not exceed 525 mm. To be conservative, the quantile values for RBDO are set to $\alpha_1 = \alpha_2 = 0.95$ for each constraint.

For both constraints, we consider anisotropic Kriging with Matérn 5/2 auto-correlation function and constant trend. The initial DOE consists of 70 points OLH for this 7-dimensional problem.

5.4.1.2 Results and discussion

In the first stage of enrichment, we consider a Monte Carlo set of 5,000 points for enrichment in the design space. The quantiles are computed with $N = 10^4$ samples. A total of K = 10points are added per iteration of enrichment. The threshold for global accuracy is set to $\bar{\eta} =$ 0.2 (See Eq. (4.42)). Convergence is achieved after only two iterations. Figure 5.25 shows the evolution of the corresponding accuracy criterion. Thanks to these two enrichments, the design of experiments is now of size 90.



Figure 5.24: 5-part-sidemember subsystem: Finite element model and the initial design together with admissible bounds in the subsystem.



Figure 5.25: 5-part side-member subsystem convergence of the first stage of enrichment after two iterations.

For the second stage of enrichment, we slightly modify the enrichment strategy. We globally set the threshold for quantile accuracy to $\bar{\eta}_q = 0.1$ (that is, we accept an error of 10%). As explained earlier, the constrained (1 + 1)-CMA-ES algorithm we use is a global stochastic approach. In

fact, the algorithm samples points in the entire space in an iterative scheme. The covariance matrix adaptation scheme allows it to direct the search to regions of space with increased fitness. However, this is not straightforward, *i.e.* most of the sampled points are not feasible nor do they increase the current fitness. Although such points are not kept by the optimizer, they give information that help to appropriately adapt the covariance matrix. Figure 5.26 shows the global step size of CMA-ES. This parameter indicates how far the next sampled point is likely to be from the current best point. Given this figure, in the early iterations, the sample step size increases, meaning that the algorithm is exploring properly the space. In the end, it decreases as the algorithm starts to sample in the vicinity of the current best point. Thus, we take into account this information in our enrichment scheme. In the latter iteration, we skip the enrichment if the sample point does not improve the fitness with respect to the current best point. In contrast, for any admissible point that increases the current best point, the accuracy threshold is set to the conservative value $\bar{\eta}_q = 0.01$ (*i.e.* we only accept a relative error lower than 1%).



Figure 5.26: Global size step of CMA-ES for optimization of the 5-part sidemember subsystem problem.

Given these settings, convergence is achieved after 300 iterations. Among these iterations, enrichment was performed 8 times with K = 3 additional points per enrichment. Overall the second stage of enrichment lead to adding 24 points in the DOE. This brings the total number of calls to the true model to 114. The initial and optimal designs are compared in Figure 5.27. Table 5.7 gives the constraints for the found solution and the associated weight. As a reminder, the maximum admissible values for the two constraints are respectively 170 kN and 525 mm. The table shows that the second constraint, *i.e.* maximal sidemember compression, is the sizing one. The weight savings is approximately 1.08 kg, which corresponds to 11.15% of the initial weight. Likewise, the initial solution was only admissible with respect to a deterministic

approach.



Figure 5.27: Comparison of the initial and optimal designs with respect to each parameter for the 5-part sidemember subsystem. The blue color stands for the nominal design and the green color for the optimal one.

Table 5.7: Comparison of the responses and weights associated to the initial and optimal designs for the 5-part sidemember subsystem.

Design	\widehat{q}_{α_1} (kN)	$\widehat{q}_{\alpha_2}(mm)$	c (kg)
d_0	177.27	527.20	9.67
d^*	155.62	523.12	8.59

Let us now validate the accuracy of the surrogate model in estimating the quantile in the vicinity of the optimal solution. In this respect, we randomly sample 100 points in the augmented space with environmental variables realizations following their distributions. The quantiles throughout optimization were computed with N = 10,000 points. We cannot afford to run the true model on such a large number of points. We therefore limit our validation to 100 points but only focus on the discrepancy between the quantiles computed by the metamodel and those computed thanks to the true model. We however consider the mean values given by 500 bootstrap replications to lower the error in the estimation of the quantile due to this reduced set of points. Table 5.8 shows the results in the two settings. The quantiles estimated by the surrogate models are quite close to those computed by the true model. The relative errors are within the threshold set during optimization.

Table 5.8: Results of simulation with respect to the true finite element model computed on 100 points with 500 bootstrap replicates. q_{α_i} represents the quantile computed with the true responses and \hat{q}_{α_i} the one computed with respect to the surrogate models. The quantiles are computed based on the 100 points resulting from the Monte Carlo set.

Model	q_{α_1} (kN)	\widehat{q}_{α_1} (kN)	$q_{\alpha_2} \ (\mathrm{mm})$	$\widehat{q}_{\alpha_2}(mm)$
Bootstrap mean	150.66	148.02	527.81	528.04

In summary, the application of the quantile-based RBDO and the two stages of enrichment proposed here allowed us to propose a solution to the lightweight design on this reduced car model. *In fine*, only 114 calls to the true model were necessary in this 7-dimensional problem. In CMA-ES, the accuracy criterion for quantile was kept less than 1% for all successive current best design. This was validated with a reduced Monte Carlo sampling on the optimal design (considering variability in the environmental designs). Next, we try to apply this methodology to a full car. In terms of numerical noise the two models are somewhat similar. The difference lies in the number of parameters and constraints.

5.4.2 Application on a full automotive body structure

5.4.2.1 Presentation of the finite element model

The model we consider here simulates Euro NCAP frontal impact crash test. The simulation is carried out using the software Radioss developed by Altair Engineering. The model consists of an entire car, without some equipment, and a barrier. Figure 5.28 shows the finite element model of the car we consider. It is mainly made of the body-in-white or automotive body structure. Since the front parts play an essential role in the crash scenario, the bumper and the parts in the engine compartment (*i.e.* the powertrain) are modeled as well. Rigid bodies allow to reach the true equipped car weight as this parameter is important in frontal impact (there is kinetic energy to dissipate). In this model, the car hits a fixed barrier at the speed of 64 km/h. The barrier is slightly offset so as to cover only 40% of the front car on the driver's half side of the car. It is made of an a aluminum honeycomb structure whose stiffness is supposed to be close to that of another car front end. As already pointed out, the solution of the finite element problem resorts to an explicit formulation. One run of the model lasts approximately 20 hours while distributed on 48 CPUs, hence the need of surrogate modeling.

5.4.2.2 Problem formulation

The lightweight design we perform is based on a reduced number of parts which belong to the front end of the car. They result from a first empirical screening (*i.e.* from expert knowledge) of the initial set of parts. As already argued, these selected parts are the most relevant for frontal impact-related specifications. In total, there are 39 distinct parts considered. However, due to symmetry and linked parts, the optimization problem consists of 20 design parameters. Figure 5.30 shows different views of the selected parts for optimization. They are highlighted in a transparent model so as to show their locations. The initial parameters values are shown in Figure 5.29 together with their bounds for optimization. For this application, the search space is defined in the region within ± 0.3 mm around each design. This is actually the maximum thickness variation that would not require adaptation of manufacturing tooling. The initial weight for this set of parts is 44.882 kg.



Figure 5.28: Finite element model of a Peugeot 308 for frontal impact analysis.



Figure 5.29: Thickness values of the nominal design together with their bounds for the 39-part body shell problem.

Eight constraints are considered here. They refer to maximum accelerations at a given point, some intrusions and a particular angle. Table 5.9 gives their maximal admissible values.

Table 5.9: Thresholds for the eight constraints of the 39-part body shell optimization problem.

Constraint	$\bar{\mathfrak{g}}_1(m/s^2)$	$\bar{\mathfrak{g}}_2 \ (mm)$	$\bar{\mathfrak{g}}_3 \ (mm)$	$\bar{\mathfrak{g}}_4~(\mathrm{mm})$	$\bar{\mathfrak{g}}_5 \ (mm)$	$\bar{\mathfrak{g}}_6 \ (mm)$	$\bar{\mathfrak{g}}_7~(\mathrm{mm})$	$\bar{\mathfrak{g}}_8$ (rad)
Thresholds	44	26	25	30	26	28	17	4

To apply the quantile-based RBDO, we set the quantile levels to $\alpha_i = 0.95$, $i = \{1, ..., 8\}$ as a conservative measure for the constraints. The input probabilistic model consists of three environmental variables. They all come from the crash test protocol:

- Initial speed: $V \sim \mathcal{U}$ (63.5, 64.5) km/h;
- Displacement of the lateral position of the barrier $U_{y} \sim \mathcal{U}(-20, 20)$ mm;
- Displacement of the vertical position of the barrier $U_z \sim \mathcal{U}(-20, 20)$ mm.

The initial design of experiments consists of 200 points in the augmented 23-dimensional space. This is actually a composite design made of an OLH of 150 points (among which some runs failed) completed by already available observations. For the Kriging models, we consider anisotropy with Matérn 5/2 auto-correlation function. The mean trend is constant (ordinary Kriging).



Figure 5.30: Representation of the 39 parts that are selected for optimization. They all belong to the front end of the car as highlighted.

5.4.2.3 Results and discussions

Before starting optimization, we proceed to a global enrichment of the design of experiments. Given the time allocated for this study, we have a restricted budget for model runs. Therefore, only two iterations of enrichment are performed with 50 points each. Among these, 20% of the simulations failed to converge. This ratio is quite high and shows that we are running the models with parameters for which the finite element model was not optimally designed. In most of the computations, energy error occurred (*i.e.* the model created energy for unknown reasons). In such cases, the simulations are considered invalid. Thus at the end of the first stage of enrichment, the DOE consists of 281 points. The global enrichment criterion is still close to 1, that is it has almost not been reduced. This is way above the thresholds we have set for the sidemember subsystem (*i.e.* $\bar{\eta} = 0.2$). Yet this is the best we could afford given the project lead-time.

We now start optimization using CMA-ES. We consider the same enrichment scheme that was used for the sidemember subsystem, *i.e.* enrichment is made only for sampled designs that improve the current best solution. This leads to four enrichments of five points each. Once again, more than 25% of the model simulations failed to converge. This leaves us with only 14 additional points to the DOE. By convergence of CMA-ES, the final DOE is of size 295. The final solution given by CMA-ES is refined by local search using Matlab's fmincon function (interior-point algorithm).

Unfortunately, we could not find any admissible solution despite numerous iterations of CMA-ES. It seems that the responses are extremely scattered. The conservative measure given by the quantiles are too penalizing for this car model. Let us however consider the *best* solution that we could find, *i.e.* the one which violates the less the constraints. For a given design, we measure the relative constraint violation as follows:

$$\bar{\mathfrak{g}}_{i}^{\%}\left(\boldsymbol{d}\right) = 100 \frac{q_{\alpha_{i}}\left(\boldsymbol{d}\right) - \bar{\mathfrak{g}}_{i}}{\bar{\mathfrak{g}}_{i}}, \qquad i = \left\{1, \dots, 8\right\},$$
(5.16)

where $q_{\alpha_i}(d)$ is the quantile of the constraint *i*, computed on the design *d*.

Table 5.10 gives the relative percentage of constraints violations for each of them. A positive value means that the constraint is violated. They are emphasized in the table with bold fonts. In this case, five constraints are violated in various degrees. For g_1 and g_3 , it is still acceptable. However, the constraint g_2 is violated by 7.50%. This occurs despite increasing the initial weight. One of the reasons for this actually lies in the numerical noise of the finite element model. As a matter of fact, a prior numerical scatter analysis as introduced in Section 5.3.2 would have pinpointed an abnormally high level of noise. Let us consider a Monte Carlo sampling of the nominal design while infinitesimally perturbing the mesh nodes positions. Recall that this is the method used to assess the quality of the models with respect to a non-linear dynamic analysis.

Figure 5.31 shows the resulting time-history for the four most scattered outputs. They all correspond to displacements. Three of them (y_2 , y_3 and y_4) refer to an intrusion on the same part. The different colors correspond to different runs. The Monte Carlo simulation consists of 50 points. The black curves correspond to the simulation of the nominal model. For the optimization problem, the designer look at the maximum of each curve. These are the scalar values that are meta-modeled. A quick look at these figures show that most of the responses lie above the thresholds defined by the optimization problem. For instance, let us consider the output y_2

Design	$ar{\mathfrak{g}}_1^\%$ (m/s ²)	$ar{\mathfrak{g}}_2^\%$ (%)	$ar{\mathfrak{g}}_3^\%$ (%)	$ar{\mathfrak{g}}_4^\%$ (%)	$ar{\mathfrak{g}}_5^\%$ (%)	$ar{\mathfrak{g}}_6^\%$ (%)	$ar{\mathfrak{g}}_7^\%$ (%)	$ar{\mathfrak{g}}_8^\%$ (%)
$d_{95\%}^{*}$	0.13	7.50	0.85	2.69	3.60	-1.83	-10.70	-13.10
$d_{90\%}^{*}$	-2.06	-3.02	-19.59	-4.04	-1.71	-11.30	-10.55	-15.63
$d^*_{\rm det}$	-2.34	0.00	-15.02	-10.11	0.00	0.00	-9.55	-9.54

Table 5.10: Relative constraint violations for different designs with respect to the 39-part bodyin-white optimization problem.

(Figure 5.31a) for which the threshold is $\bar{\mathfrak{g}}_2 = 26$. All simulations, nominal included, produce maximal values above this threshold. As argued earlier, the quantile-based RBDO we propose here is aimed at taking into account physical variability arising from the crash protocol. In this setting we consider that the numerical noise, supposedly smaller, is implicitly handled. However, according to these figures, the numerical noise alone is quite high and its amplitude does not let room for any design improvement.



Figure 5.31: Time-history plots of four of the most scattered outputs resulting from a numerical noise Monte Carlo sampled runs. The different colors represent the different MC runs. The black bold curve is the nominal design.

The distribution of the outputs with respect to the physical input variability is given in Figure 5.32 for the four most scattered outputs. These histograms are generated using the surrogate models. They may not be completely accurate but give an insight on the variability we are trying to take into account. The red and black lines correspond respectively to the 95% quantile and constraints limits. The distribution are skewed with a higher weight at the left of the median. Thus, a small variation of the quantile level α in the right tail of the distribution leads to a large variation of the associated quantile value. For this reason, we also perform an optimization while considering a quantile level $\alpha = 90\%$. This leads to an admissible design $d_{90\%}^*$ whose relative constraint violation is given in Table 5.10 as well. This corresponds to a weight saving of 1.568 kg. Given the parts selected for optimization, this weight saving is rather good. However, it also implies that reliability has a cost in terms of weight reduction. For comparison, we eventually perform a deterministic design optimization. The resulting thickness distribution

 d_{det}^* is shown in Figure 5.33 together with that of the other designs. The corresponding weight saving is as expected much higher, *i.e.* 5.144 kg. The results for all three designs are gathered in Table 5.11.



Figure 5.32: Physical scatter for four of the most noisy outputs with respect to the unfeasible design $d_{95\%}^*$. The histograms are generated by Monte Carlo sampling with respect to the surrogate models. The black and red curves represent the constraint threshold and the 95% quantile respectively.

It seems important to emphasize two points here. First, the solutions presented above are conditional to the surrogate models accuracy. We could only check the true solution with respect to the deterministic solution. In turned out that roughly all the constraints were satisfied, except for y_2 and y_6 which where violated respectively by only 0.38% and 2.54%. This was considered small enough for the design to be validated. We could not run a Monte Carlo simulation for the RBDO results. However, the responses of the FE models considering the nominal values of the

Table 5.11: Final weight for the different solutions compared to the nominal design d_0 .

Design	d_0	$d_{95\%}^{*}$	$d_{90\%}^{*}$	$d_{ m det}^*$
Weight (kg)	44.882	45.720	43.314	39.738
Weigth variation	_	0.838	-1.568	-5.144
Relative weight saving (%)	_	1.87	-3.49	-11.46



Figure 5.33: Thicknesses of the various parts corresponding to different solutions compared to the nominal one. The black lines bound the search space for each thickness.

environment variables were also shown to be globally acceptable. Second, these weight savings, *i.e.* 1.568 kg and 5.144 kg respectively for RBDO and DDO, are only the contributions of frontal impact related parts to the overall weight reduction of the full body shell. In fact, recall that the weight optimization here is multi-disciplinary. Therefore, thanks to experts knowledge, distinct perimeters of parts are defined for each discipline so that there is no coupling between the different responses. Optimization is therefore carried out sequentially for each subset of parts within each discipline. The overall weight saving is henceforth substantial.

5.5 Conclusion

This chapter has presented numerical analyses with the proposed adaptive Kriging and quantilebased RBDO approach. The chapter is divided into two parts. The first one was aimed at validating the proposed methodology with respect to literature references. Three applications examples were treated. They all differ in the type of probabilistic models considered, *i.e.* randomness in design variables only, randomness in environmental variables and eventually randomness in all types of variables. The first example allowed us to validate the methodology with respect to an analytical solution. An accurate solution could indeed be found in a reduced number of calls to the true model. In the two other examples, comparison was made with other available reference solutions. This allowed us to validate the efficiency of the approach as the number of calls to the true model was shown to be either smaller or roughly equal to these benchmark solutions.

The second part of the chapter is devoted to applying the methodology to the lightweight design of an automotive body shell under frontal impact. Prior to this application, we investigate the specificity of the frontal impact, *i.e.* the presence of noise and its chaotic behavior. In this respect, two case studies were carried out. The first study allows us to have a deeper understanding of how the numerical noise affects the simulations. The important aspect is its heteroscedasticity which disables the possibility of feeding the Kriging model with an already known homogeneous noise variance. Accounting for heterogeneous noise is not possible because of the computational budget it would require. The second study was concerned with the chaotic behavior of cars in frontal impact which leads to different crash scenarios. It was shown how support vector machines for classification may help improve surrogate modeling in presence of multiple physical behaviors.

Eventually, RBDO was applied to the two models, namely the 5-part sidemember subsystem and a 39-part full body shell model. Thanks to a relatively low dimension in the sidemember subsystem (s = 7), the methodology was applied successfully. The metrics for accuracy in the two stages were monitored and were shown to decrease as enrichment was applied. A reliable solution with reduced weight was found. As for the full body-in-white problem, the application was somehow difficult for various reasons. First, the model is highly noisy. This makes any attempt to find conservative solutions extremely difficult. We could however find some designs while slightly decreasing the degree of conservatism. Second, the high dimension of the problem (s = 23) made it hard to train accurate Kriging models for the 8 outputs. Because of time restriction (we perform the optimization in a project-like environment), we could not make many iterations of enrichments. Beside, given the high dimension of the problem, the Kriging margin shrinking is not easy to achieve. Therefore, the accuracy criteria remain quite high despite enrichment. A better strategy in such a case would be to start the second stage of enrichment in the very last iterations of optimization, *i.e.* once the region with optimal solution has been identified.



CONCLUSION

The need to explore new approaches for efficient lightweight design of new cars has motivated the work presented in this PhD thesis. In particular, the automotive body shell, also known as body-in-white, provides an interesting potential for design to the strictly necessary, *i.e.* there is room for further weight saving that might be achieved through constrained multidisciplinary optimization. This work primarily deals with the lightweight design of an automotive body structure under frontal impact. The following section restates the problem and summarizes the approaches explored for a solution.

Summary and main contributions

The lightweight design of an automotive body structure under frontal impact

To achieve weight reduction of the automotive body structure, one searches for an optimal distribution of its part thicknesses while satisfying a collection of constraints resulting from safety and comfort requirements. In this work, we are concerned with frontal impact since it is the one which causes the most difficulties in optimization. This is due to its chaotic nature which makes any finite element simulation noisy. The noise may come either from inherent physical variability of the input parameters or from modeling error. The latter is termed in this thesis *numerical noise*. In this work, it was proposed to handle this problem by introducing a probabilistic framework which allows us to propagate the input variability to the outputs and henceforth to search for a conservative design. This can be achieved through design optimization under uncertainties.

Optimization under uncertainties

Design optimization under uncertainties is an important field of research in which scientists contributions have been constantly increasing. Chapter 2 of this thesis aimed at reviewing the state-of-the-art techniques developed to account for uncertainties in design optimization. The

6. CONCLUSION

limitations of a deterministic design in a context where uncertainty is ubiquitous were first highlighted. Then the various sources of uncertainties were analyzed, followed by the possible ways of handling them. From the point of view of design safety, robust and reliability-based design optimizations (respectively RDO and RBDO) provide the necessary tools. Given that uncertainties in the problems of interest primarily act on the constraints, we argue that RBDO is the most appropriate approach. We therefore review reliability analysis tools and RBDO techniques. The latter involves methods that can be grouped into two-level, mono-level and decoupled approaches.

In this chapter, we propose an alternative approach to solve RBDO problem. This approach is based on the computation of the constraints quantiles. Its formulation is motivated by some characteristics of our applications. First, we are not concerned with the simulation of rare events. In fact, the RBDO is simply meant to account for inputs variability and thereafter to propagate them to the outputs where a conservative solution is sought. In this respect, the quantile is used as an arbitrary measure of conservatism. Its possible values are henceforth relatively high. In this case, there is no need of any of the sophisticated variance-reduction simulation techniques traditionally used in RBDO. A quantile estimated through a basic Monte Carlo sampling is enough for our application. The approach may therefore be seen as double-loop where the outer loop explores the design space and the inner one quantifies the conservatism of the design thanks to a straight Monte Carlo sampling. The second aspect of our problem is that it is meant to be applied in an industrial context where deterministic design optimization is the cultural reference. Providing an approach which can be built upon the current workflow without requiring intrusive developments is essential. As such, the quantile can be directly plugged as constraints into the current optimization algorithms. This will supposedly facilitate the transition from a deterministic design optimization to its reliable counterpart.

The application of such a direct approach to an industrial problem is however not possible because of the high computational budget it would require. Chapter 3 was concerned with methods to alleviate the high cost of implementing an RBDO approach in an industrial context.

Surrogate modeling

An exponential increase of computational power has been observed during the last few decades. This gave designers the possibility to refine their modeling of the physical behaviors they aim to simulate. Thus, the progress made in computers followed the complexity of mechanical models. In the automotive industry, high-fidelity models now allow designers to simulate a frontal impact extremely finely, hence reducing the cost of physical prototyping. However, the associated computational cost in terms of CPU time is quite a hurdle to any optimization analysis as the latter requires repeated evaluations of the time-consuming model. Lately, surrogate modeling has emerged as a tool to overcome these difficulties.

Chapter 3 of this thesis is mainly concerned with two types of surrogate models, namely Kriging and support vector machines. Support vector machines (SVM), whose development has been based on structural risk minimization principles, provides a mathematical framework for learning from data. Regression and classification are possible through SVM. In this thesis, the latter is used for classification of crash scenarios in an attempt to improve predictability. Kriging, a.k.a. Gaussian Process modeling, has been extensively used in the literature for the approximation of black-box functions. In this work, the main asset of Kriging is that it provides a local measure of its own accuracy. Given in a form of a probabilistic measure, this enables the development of adaptive techniques which allows us to improve accuracy only in regions of interest. The main contributions of this work rely on this point.

Quantile-based optimization and enrichment

It was proposed in Chapter 4 to combine the quantile-based RBDO method introduced in Chapter 2 with an adaptive refinement of the Kriging models. The idea is to reduce the overall number of calls to the true time-consuming finite element model by only focusing in regions of the space that are meaningful to the problem at hand. To this end, a two-stage enrichment scheme is proposed. The first stage is global. The idea is to identify all limit-state surfaces and to add new training points in their vicinity so that the associated Kriging margin of uncertainty is made small enough. This means that we only search accuracy in regions of space where the constraints are likely to be violated. A metric that allows us to monitor the convergence of this enrichment has been defined. The second stage of enrichment is local. In fact, it is coupled with optimization. As the algorithm explores the design space, accuracy of the estimated quantiles is estimated. A metric which consists of an upper and a lower bounds on the quantiles has been derived to assess this accuracy. An iteration of enrichment is performed when the relative distance between the bounds is beyond a pre-defined threshold. By coupling optimization and enrichment, we direct the enrichment (and therefore the computational budget) to regions where the objective function is decreasing (assuming we consider a minimization problem). Tricks were also suggested to keep the enrichment to the strict necessary amount.

Perspectives and future research

Applications of this enrichment strategy on analytical and mechanical problems has shown the accuracy and the enhanced efficiency brought by the approach. However, this is only true for problems of moderate dimensions. In the case of high-dimensional problems, some difficulties have been encountered. In this section, we review their possible causes and suggest a few avenues of thought for future work aiming at applying RBDO to lightweight design of automotive body structures.

Application to high-dimensional problems

The approach we have proposed here might be challenged in two ways. The first point is related to its application in an industrial context. The enrichment scheme implies that model evaluations are made sequentially. Despite the overall number of model runs is reduced, this sequencing extends the time necessary to carry out the optimization. When schedules are tight, one might not be able to fully apply the methodology. Beside, for high-dimensional problems, the surrogate models trainings and the numerous computations of quantiles (*i.e.* in the first stage of enrichment) make the approach prohibitively "heavy".

Second, we faced several difficulties, especially due to the curse of dimensionality, in the full body-in-white application. From the viewpoint of Kriging, it is hard to train an accurate model. More importantly, the approach relies on the shrinking of the margin of uncertainty in areas where points are added to the DOE. The metrics to assess improvement of the fitting accuracy were developed with respect to this concept. However, in a high-dimensional setting the refinement criteria stall for a large number of enrichments iterations. Such criteria are certainly not well adapted in this case. Alternatively, one may consider other types of surrogate models.

In this thesis, we have presented support vector machines for regression. They are *theoretically* well adapted to high-dimensional problems. However, in contrast to Kriging, SVR in its native formulation does not feature any tool to assess its local accuracy. Many works have been achieved in order to implement a probabilistic framework to support vector machines (Gao et al., 2002; Lin and Weng, 2004). One may therefore consider applying them to this work in replacement of Kriging. The recourse to bootstrap to derive a measure of local error in SVR may also be investigated. We should however stress here that the calibration of a SVR hyperparameters is quite a challenging task. Gradient-based approaches often fail to produce good models. In this thesis, we used a global search method, namely the cross-entropy based optimizer which is quite expensive. The cost of training therefore grows with both the dimension and the size of the training set.

From another perspective, model reduction techniques should be exploited to reduce the size of the problems. Screening techniques may be used to lower the dimension of the optimization problem and henceforth that of the augmented space in which the surrogate models are built. Likewise, the high number of constraints in optimization increases the number of models trainings. Feature extraction methods such as principal component analysis may also help reduce the number of models trainings. Eventually, it is widely accepted that the metamodeling of outputs which are maximum values of processes is difficult. It may be interesting to develop approaches that would allow us to directly approximate the time-history curves given by the finite element models. This is however acknowledged to be a tough assignment.

Implementation in an industrial context

From a broader perspective, some challenges still need to be faced in order to implement a RBDO approach to body shell lightweight design in an industrial environment. The prime aspect that comes to mind is the quality of the finite element models. Recall the percentage of failed simulations when creating and updating the DOE for the full car application. There was 20% of the simulations that failed. Of course this ratio is not the same for all models and is usually smaller. However, it shows that the finite element model was not optimally designed for parameters which depart too much from the nominal values. Besides, the numerical noise, which is attributed to modeling error and other simplifying assumptions, was quite high in this application. This made the finding of an admissible reliable solution for relatively high values of the quantiles difficult (*e.g.* 95%). There is certainly much to be gained by providing the designers with more robust FE models. Beside, the number of calls to the true model is necessarily increased in RBDO in comparison to a DDO approach. Here again there is much to be gained by increasing the computational power in terms of available CPUs. This would allow the designers to run models in a shorter time-lapse and reduce the restitution time of a job (which is often much higher than that of a single model simulation).

Finally, to close this discussion, we must also mention cultural barriers that refrain engineers from fully considering uncertainties in the early stage of new concepts developments. Despite the role of uncertainties in real-life problems is widely accepted, there is still a long path way to go before uncertainties are systematically accounted for in engineering processes. In this thesis, we have considered the physical variability in the crash protocol as the main source of uncertainties. The difficulty here is how to quantify these uncertainties and propose appropriate distributions for each of them. Furthermore, many other aspects are sources of uncertainties but quantifying them is quite challenging. All these reasons make the implementation of such an RBDO approach difficult. Moreover, one of the objectives of this work was also to handle the

numerical noise alone since this has seriously bothered engineers. This would have been a first milestone toward systematic consideration of uncertainties. Further work is still necessary to achieve this. However, we have proposed an approach which accounts for uncertainties from a broader perspective. This approach may allow the engineers to directly include frontal impact to lightweight design of automotive body structures. An application has shown the potential gain in terms of weight savings and therefore a contribution to the reduction of CO_2 emissions.



INITIAL DESIGN OF EXPERIMENTS: A SHORT STATE-OF-THE-ART

According to the methods that are used to build them, space-filling designs can be classified into three groups: *deterministic, random* and *quasi-random* designs. We briefly review some of the most-widely used here while focusing on quasi-random designs.

A.1 Deterministic design of experiments

By deterministic, it is simply meant that there is no randomness in the choice of the input parameters values. The design sites are chosen thanks to a predefined scheme. Examples of such designs are given below.

A.1.1 Factorial designs

A *full factorial design* is an *s*-dimensional *L*-level grid *i.e.* L > 1 levels are chosen in each dimension and the points of the learning basis are generated considering all possible combinations of these levels. Such designs are known to be well suited to polynomial models as their properties make the interpretation of the model coefficients easy. They are also very easy to build but their size grows dramatically with both the dimension and the number of levels: $n = L^s$. This makes them hardly tractable when either the dimension or the number of levels is high. In such a case, one may consider *fractional factorial designs* which are constructed as a subset of full factorial designs where only a fraction of the fraction of points to keep is based on the *sparsity-of-effect principle* (Goupy and Creighton, 2006). Under this principle, it is assumed that the model is primarily driven by marginal effects and possibly second-order interactions. Higher-order interactions are considered negligible. These two types of designs are mainly used with poly-

nomial response surfaces, often of low orders. They may not be suitable for general-purpose surrogate models such as Kriging.

A.1.2 General linear specific designs

When one knows exactly which type of model is to be built, it is possible to use this knowledge to design an experiment that conveniently suits the model. These types of designs exist for general linear models. Such models, which consists of a linear combination of regression functions, are calibrated by tuning the regression coefficients. The variance of these coefficients is known to be proportional to the so-called *information matrix* $F^T R^{-1}F$, where F and R are respectively the regression and auto-correlation matrix. A family of designs based on criteria minimizing this variance have been developed, among which:

• A-optimality which consists of minimizing the trace of the information matrix:

$$\mathcal{D}_{A} = \arg\min_{\mathcal{D}\in\mathbb{D}} \operatorname{trace}\left(\boldsymbol{F}^{T}\boldsymbol{R}^{-1}\boldsymbol{F}\right)^{-1}; \qquad (A.1)$$

• D-optimality which avoids the inversion above by rather maximizing the determinant of the information matrix

$$\mathcal{D}_D = \arg \max_{\mathcal{D} \in \mathbb{D}} \det \left(\boldsymbol{F}^T \boldsymbol{R}^{-1} \boldsymbol{F} \right).$$
(A.2)

Other designs, such as the G-optimal which minimizes the maximum variance of the estimates, exist. Despite their apparent simplicity, their implementations require the solution of a high-dimensional optimization problem of size $s \times n$. For the D-optimal criterion for example, the commonly used technique relies upon exchanging coordinates, which turns out to be a combinatorial optimization problem (a technique used in Matlab statistical toolbox).

A.1.3 Bayesian designs

Just as for the above general linear models, some designs have been developed specifically for Bayesian predictors. They attempt to exploit a prior knowledge of the autocorrelation function. In this category, Lindley (1956) introduced the *maximum entropy* criterion for statistical theory based upon Shannon's entropy (Shannon, 1948). The work of Shannon in information theory was concerned with the measure of the amount of information received during the process of its transmission. In statistical theory, this idea is equivalent to considering the knowledge we have about a process before and after an experiment is performed. Consider now an initial design $\mathcal{D} \in \mathbb{D}$. We would like to predict the output of a system in the unsampled points of \mathbb{D} : $\overline{\mathcal{D}} = \mathbb{D} \setminus \mathcal{D}$ using Bayesian methods, say Kriging. As shown in the previous chapter, the prediction relies on a stochastic process *Y*. The best design, in the sense of entropy, is the one that maximizes (Lindley, 1956):

$$H(Y) = \mathbb{E}_{Y} \left[\log f_{Y}(Y) \right], \tag{A.3}$$

where \mathbb{E}_Y denotes the expectation w.r.t. *Y* and *f*_{*Y*} is the density of *Y*.

For Bayesian process with Gaussian priors, Currin et al. (1988) have shown that the criterion becomes:

$$\mathcal{D}_{E} = \max_{\mathcal{D} \in \mathbb{D}} \det \left[C \right], \tag{A.4}$$

where *C* is the covariance matrix of the process *Y*. Under the assumption of stationarity, the criterion reduces to maximizing det [*R*], *R* being the autocorrelation matrix. This criterion has been successfully applied to computer experiments by Shewry and Wynn (1987) and Currin et al. (1988), among others. The resulting optimization is however not trivial to solve. Currin et al. (1991) proposed an efficient algorithm to handle it.

Figure A.1 illustrates this criterion for a Gaussian process with auto-correlation length of 0.4. In the left panel, an initial random design is sampled. The determinant of its auto-correlation is $3.51 \ 10^{-5}$. The right panel shows a design obtained by maximizing det [*R*] to a value of 0.21. The optimization was carried out with the R package DiceDesign (Franco et al., 2013). Despite the resulting design clearly fills the space in a better way than the initial one, the optimization problem may be hard to solve, especially for high dimensional cases.



Figure A.1: A two-dimensional maximum entropy design of size 20.

Other well-known Bayesian designs are based on the mean-square error, more specifically the *integrated* and *maximum mean-square error* (Sacks et al., 1989). But just as the entropy criterion, they result from high-dimensional optimization problems. It has also been shown that they have a tendency to gather points inside the design space in the expense of the boundaries (Koehler and Owen, 1996).

A.1.4 Geometric criteria based designs

These designs, introduced by Johnson et al. (1990), are based on the optimization of some Euclidean distances on the design space. Starting from a random design, they developed so-called *maximin* and *minimax* criteria.

Let consider an initial design \mathcal{X} . The prediction at a new point x is likely to be more accurate

when x is close to a design site and is on the contrary less accurate when x is remote from all design sites. Minimax designs are constructed accordingly with this observation. This is achieved by searching a design that makes the maximum distance of all points in the design space with their closest points in \mathcal{X} as small as possible. In a formal way, the following minimax distance criterion is minimized:

$$\Phi_{mM}\left(\mathcal{X}\right) = \min_{\boldsymbol{x}\in\mathcal{D}} \max_{\boldsymbol{x}_i\in\mathcal{X}} \|\boldsymbol{x} - \boldsymbol{x}_i\|.$$
(A.5)

Geometrically, this can be seen as covering the design space with the smallest balls centered on the design sites (Pronzato and Müller, 2011).

Alternatively one might need to spread the design sites as much as possible. This is achieved in *maximin* designs which consist in maximizing the so-called maximin distance criterion:

$$\Phi_{Mm}(\mathcal{X}) = \min_{i \neq j} \|x_i - x_j\|, \quad i, j = \{1, 2, \dots, n\}.$$
(A.6)

By analogy with minimax designs, they seek to maximize the radius of non-intersecting balls centered on the design sites. (Pronzato and Müller, 2011).

A.2 Random designs

The introduction of randomness in design sampling comes from Monte Carlo techniques where an integral of a function f is approximated in the unit hypercube by averaging the values it takes on some randomly drawn points:

$$\int_{[0,1]^s} f(t) dt = \frac{1}{n} \sum_{i=1}^n f(x_i),$$
(A.7)

where *s* is the dimension of the space and $\{x_i, i = \{1, ..., n\}\}$ is a set of independent and identically distributed points in $[0, 1]^s$.

The variance of this estimate is σ_f^2/n , where σ_f^2 denotes the variance of f. A good estimation of this integral requires the unit hypercube to be filled as uniformly as possible. For this purpose, many techniques have been developed. By extrapolation, they have been brought to the field of surrogate modeling where they serve for the design of experiments. The most natural one is Monte Carlo sampling which consists in drawing from a uniform distribution in the unit hypercube:

$$\mathcal{X}_{MCS} = \{x_{ij}, i = \{1, \dots, n\}, j = \{1, \dots, s\}\}.$$
(A.8)

Such a design is quite easy to implement. However its rate of convergence to the uniform distribution, in the order of $1/\sqrt{n}$, is quite slow. It thus fails to truly fill the unit hypercube when its size is not large enough. Alternative designs, known as *stratified sampling* allow to increase this rate of convergence.

A.2.1 Latin hypercube sampling

Latin hypercube sampling has been developed by McKay et al. (1979) as an improvement of MCS. The authors that the associated variance is smaller than that of MCS. This reduction is achieved by ensuring the uniformity in any dimension. Hence, each input variable has all

portion of its distribution represented by a sample point. LHS is quite easy to implement, which makes it a powerful and widely-used design. It takes the form:

$$\mathcal{X}_{LHS} = \left\{ \mathbf{x}_{ij} = \frac{\pi_{ij} - 1 + u_{ij}}{n}, \{i = 1, \dots, n\}, \{j = 1, \dots, s\} \right\},$$
(A.9)

where π_{ij} is the *i*-th element of a random permutation of the sequence $\{1, ..., n\}$ along the dimension *j* and u_{ij} is a realization of a uniform random variable *U* in [0, 1].

In a practical point of view, to construct this design, one divides each unit axis corresponding to a input variable into n bins of equal width. The tensor product of these bins gives rise to a mesh of n^s cells. One then chooses n cells so that each interval of the marginal input variables is represented only once. Eventually, one randomly places a sample point in each of the chosen cells.

Despite LHS reduces MCS asymptotic variance, its rate of convergence is not better than that of MCS, *i.e.* $\propto 1\sqrt{n}$. Furthermore, the uniformity in the marginal input variables does not guarantee uniformity in the entire design space. For instance, consider a 2-dimensional ten points LHS with the following permutations: $\pi_1 = \{1, 2, ..., 10\}$ and $\pi_2 = \{10, 9, ..., 1\}$. The resulting is plot in Figure A.2. The picked cells turn out to be in the diagonal and the distribution is far from being uniform over the unit square.



Figure A.2: Latin hypercube sampling with an ill-chosen permutation sequence

Of course, this example is one of the worst possible cases and its occurrence is not very likely. However, it illustrates the limitation of a pure Latin hypercube sampling. Many authors have attempted to improve LHS by implementing the so-called *optimal Latin hypercube* (OLH) sampling. They proceed by combining the good one-dimensional projection property of LHS with another criterion such as maximin (Morris and Mitchell, 1995), IMSE and entropy (Park, 1993) or discrepancy (Fang et al., 2002). Many authors proposed algorithms to handle the task such as simulated annealing search in Morris and Mitchell (1995), an exchange and Newton-type algorithms (Park, 1993) or a genetic algorithm in Bates et al. (2004). The resulting optimization problems have prohibitive computational costs, especially for high values of *s* and *n*. Near op-

timal Latin hypercube designs are an alternative where a good, but not the best, OLH is sought (*e.g.* in Viana et al. (2010) which also provides ready-to-use Matlab codes).

A.2.2 Orthogonal arrays

Owen (1992) and Tang (1993) independently introduced *orthogonal arrays* for computer experiments, which are generalizations of Latin hypercube designs in dimension *s*. They are defined as follows (Owen, 1992, 1994).

An orthogonal array of strength *r* and symbol *q* is a matrix of *n* rows and *s* columns with elements taken in $\{0, 1, ..., q - 1\}$ such that in every submatrix of size $n \times r$, each of the possible q^r rows occur the same number of times λ (note that $n = \lambda q^r$). They are denoted by OA(n, s, q, r).

Owen (1992) also showed that an orthogonal array remains orthogonal if the symbols $\{0, 1, ..., q - 1\}$ are permuted. He then proposed the following randomized orthogonal array sampling formulation:

$$\mathcal{X}_{OA} = \{ x_0, x_1, \dots, x_{n-1} \},$$
with $x_{ij} = \frac{\pi_j (A_{ij}) + u_{ij}}{q},$
(A.10)

where π_j is the permutation of the sequence $\{0, ..., q-1\}$, A_{ij} are elements of the orthogonal array and u_{ij} is a realization of a uniform random variable in [0, 1].

The main asset of an orthogonal array of strength r is that it allows one to generalize in dimension r the good projection properties in dimension one of Latin hypercubes. However, for projection in dimensions lower than r, there is no guarantee that the projection properties are good. Besides, the cost to build an OA can be quite high especially with increasing values of $\lambda = nq^{-r}$. Given a strength and a symbol, it is thus recommended to build an OA with $\lambda = 1$ (Franco, 2008).

Figure A.3 shows a three-dimensional randomized orthogonal array of strength r = 2 and symbol q = 7 with n = 49 points. It was generated with the R package DiceDesign. The projection in all two sub-dimensions are shown for (x_1, x_2) , (x_1, x_3) , (x_2, x_3) respectively in Figures A.3b, A.3c and A.3d. As expected, all unit axes are divided in q = 7 cells and each cell contains exactly $\lambda = 49 \times 7^{-2} = 1$ point. Thus, the uniformity in all the subspaces of dimension 2 are ensured. Note that, without the random noise u, the projections in the two-dimensional subspaces are nothing but 7-level full factorial designs.

A.3 Quasi-random designs

Quasi-random or Quasi-Monte Carlo designs were introduced in an attempt to improve MCS as well. They actually replace the random sampling in MCS by a well-chosen deterministic sequence. The reasons for their use are two-fold. First, it is not possible to generate in a strict sense random sequences of numbers since randomness in computers spring from deterministic algorithms. They are known as *pseudo-random numbers* (see Niederreiter (1992) for a deeper insight). Then if we could only hope to generate pseudo-random numbers or otherwise said, quasi-random points, we might as well search those which rate of convergence toward uniformity is higher than the one of a pure MCS (Jourdan, 2000) (recall that the more uniform the



Figure A.3: Randomized orthogonal array in three dimensions: OA(49,3,7,2)

design, the better the integral approximation in a Monte Carlo technique). This is the aim of quasi-Monte Carlo designs. In these techniques, the uniformity of a sequence is measured by the so-called *discrepancy*.

A.3.1 A short introduction to discrepancy

The discrepancy is a measure of the deviation of a given sequence from uniformity. Let us consider a sequence of at least *n* points \mathcal{X} in the unit hypercube $I_s = \{0, 1\}^s$. Uniformity ensures that the proportion of points in any subset of I_s is asymptotically equal to its volume.

Mathematically this translates into (Hlawka, 1964):

$$\frac{A(P,\mathcal{X},n)}{n} \stackrel{n \to +\infty}{\to} \lambda(P), \tag{A.11}$$

where *P* is any subset of *I*_s, A(P, X, n) is the number of points of the sequence X belonging to *P* and $\lambda(P)$ is the Lebesgue measure of *P*. The Lebesgue measure of a subset is actually its *s*-dimensional volume and coincides for $s = \{1, 2, 3\}$ with respectively the length, the area and the volume. In the sequel, we will simply call it "volume".

Figure A.4 depicts the 128 points of a random sequence and two subsets P_1 and P_2 , each of which has a volume of 1/16. A necessary condition for this sequence to be uniform is that the number of points they contain equals 128/16 = 8. However, one has $A(P_1, \mathcal{X}, n) = 13$ and $A(P_2, \mathcal{X}, n) = 6$. Both of them are different from 8. These deviations are somehow measured by the discrepancy $(D_n(\mathcal{X}))$. Various definitions of discrepancy exist, among which (Thiemard, 2000; Dick and Pillichshammer, 2010; Franco, 2008):



Figure A.4: A random sample of 128 points in I_2 and two subsets P_1 and P_2

• Extreme discrepancy

$$D_n^{ext}(\mathcal{X}) = \max_{P \subset I_s} \left| \frac{A(P, \mathcal{X}, n)}{n} - \lambda(P) \right|.$$
(A.12)

Here *P* is any subset of the form $\prod_{i=1}^{s} [a_i, b_i]$ such that $0 \le a_i < b_i < 1, \forall i = \{1, \dots, s\}$.

• Star discrepancy is a less conservative definition and writes:

$$D_n^*(\mathcal{X}) = \max_{P \subset I_s^*} \left| \frac{A(P, \mathcal{X}, n)}{n} - \lambda(P) \right|,$$
(A.13)

where I_s^* represents all the subsets of I_s anchored at the origin, *i.e.* in the form $\prod_{i=1}^{s} [0, b_i)$ (*e.g.* $P_1 \subset I_s^*$ in Figure A.4).

• *L*₂-discrepancy

$$D_n^{L_2}(\mathcal{X}) = \left[\int_{P \subset I_s^*} \left(\frac{A(P, \mathcal{X}, n)}{n} - \lambda(P) \right)^2 dP \right]^{1/2}.$$
 (A.14)

A property which follows from these definitions is that $0 \le D_n(\mathcal{X}) \le 1$. A sequence is said to be uniform when $D_n(\mathcal{X}) \xrightarrow{n \to +\infty} 0$. The computation of the discrepancy for any given sequence is however not straightforward and many authors resort to lower and higher bounds (Thiemard, 2000). In dimensions one and two, there exists some simple formulations of discrepancy as derived by Niederreiter (1992) for example. For a Monte-Carlo sampling, the discrepancy is known to be $1/\sqrt{n}$. The discrepancy of a full factorial design is 1/n.

A.3.2 Some low-discrepancy sequences

Many authors have developed sequences for which the discrepancy is lower than the one of Monte-Carlo. They are usually in the order of $\log(n)^s/n$ and are known as *low-discrepancy sequences*. Some important results are reported in the sequel.

A.3.2.1 van der Corput sequence

This sequence, introduced by J.G. van der Corput in 1935 (van der Corput, 1935), features a small discrepancy and is easy to implement. It is based on the so-called *b-adic inverse radical function*. Any integer $i \in \mathbb{N}$ has the following *finite* b-adic expansion: $i = n_0 + n_1 b + n_2 b^2 + \dots$ Its associated radical inverse function $\phi_b(i) : \mathbb{N} \to \{0, 1\}$ is defined as:

$$\phi_b(i) = \frac{n_0}{b} + \frac{n_1}{b^2} + \frac{n_2}{b^3} + \dots$$
(A.15)

The first *n* points of a van der Corput (VDC) sequence are then defined as any sequence $\mathcal{X}_{VDC} = \{x_0, x_1, \dots, x_{n-1}\}$ where $x_i = \phi_b(i)$. It can be shown that a VDC sequence for the prime integer *b* consists of monotonically increasing cycles of length *b*. At each cycle, the gaps between the points of the previous cycles are filled. This allows an asymptotic uniform coverage of the space.

Table A.1 below illustrates the construction of the first ten points of a VDC sequence in base 2. The third column shows the increasing cycles of length 2: (0, 0.5), (0.25, 0.75), etc. In this particular case, the sequence fills alternatively the segments [0, 0.5] and [0.5, 1].

A.3.2.2 Hammersley sequence

Hammersley (1960) developed an *s*-dimensional generalization of the van der Corput sequence. Halton (1960) has demonstrated that the discrepancy of this sequence is in the same order as the one of van der Corput, as postulated by Hammersley. The first n points of the sequence are given by:

$$\mathcal{X}_{HAM} = \{ \boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_{n-1} \},$$

with $\boldsymbol{x}_i = \left(\frac{i}{n}, \phi_{b_1}(i), \dots, \phi_{b_{s-1}}(i) \right),$ (A.16)

where ϕ_{b_j} is the b_j -adic radical inverse function introduced above, and b_j are integers that are prime to each other.
i	i _{base2}	$\phi_2(i)$	Sequence Sample
0	0000	0	•
1	0001	$\frac{1}{2^1} = \frac{1}{2}$	•
2	0010	$\frac{1}{2^2} = \frac{1}{4}$	• • •
3	0011	$\frac{1}{2^1} + \frac{1}{2^2} = \frac{3}{4}$	• • • • • • • • • • • • • • • • • • • •
4	0100	$\frac{1}{2^3} = \frac{1}{8}$	• • • • • • • • • • • • • • • • • • • •
5	0101	$\frac{1}{2^1} + \frac{1}{2^3} = \frac{5}{8}$	• • • • • • •
6	0110	$\frac{1}{2^2} + \frac{1}{2^3} = \frac{3}{8}$	• • • • • • •
7	0111	$\frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} = \frac{7}{8}$	• • • • • • • • •
8	1000	$\frac{1}{2^4} = \frac{1}{16}$	• • • • • • • • • •
9	1001	$\frac{1}{2^1} + \frac{1}{2^4} = \frac{9}{16}$	• • • • • • • • • • • • • • • • • • • •

Table A.1: The first ten points of a VDC sequence.

Faure has developed an upper bound of the discrepancy of a Hammersley sequence (Thiemard, 2000):

$$D_n^*(\mathcal{X}_{HAM}) \le \frac{s}{n} + \frac{1}{n} \prod_{i=1}^{s-1} \left(\frac{b_i - 1}{2\log b_i} \log n + \frac{b_i + 1}{2} \right).$$
(A.17)

This bound exhibits a dominating constant: $\prod_{i=1}^{s-1} \frac{b_i-1}{2 \log b_i}$. It is thus possible to reduce the discrepancy by choosing the s - 1 first prime integers.

For comparison with MCS and LHS, a Hammersley sequence in dimension two is plotted in Figure A.5 together with a Halton sequence (introduced in the next section). They all consist of 100 points. It is clear from these figures that the two quasi-random designs offer a better uniform coverage of the unit square.

The main drawback of the Hammersley sequence is that one has to know beforehand the desired sample size *n* since it is directly used in the computation of the coordinate of the points in the first dimension $(x_{i1} = i/n)$. Besides, for an already built sequence, adding points is not natural and would deteriorate the sequence properties. Halton (1960) proposed an alternative to Hammersley sequences.

A.3.2.3 Halton sequence

Halton sequences are also *s*-dimensional generalizations of van der Corput sequences. Unlike Hammersley, they do not feature the *n*-dependent term in the first dimension and solely consist of radical inverse functions (Halton, 1960):

$$\mathcal{X}_{HAL} = \{ x_0, x_1, \dots, x_{n-1} \},$$

with $x_i = (\phi_{b_1}(i), \phi_{b_2}(i), \dots, \phi_{b_s}(i)).$ (A.18)

200



Figure A.5: Different designs of 100 points

Hence they are easier to update. It is only required to have integers b_j prime to each other and once more, selecting the *s* first prime integers leads to the best results. However in high dimensions, the sequence exhibits some inconvenient correlations resulting in diagonal strips.

Figure A.6a above shows projection of an 8-dimensional Halton sequence on the two last dimensions (corresponding to the bases 17 and 19). The parallel lines are due to the linear correlation between the bases 17 and 19. Franco (2008) points out that this phenomenon is more likely to occur in high dimensions and when the relative difference between the two bases is



Figure A.6: The first 100 points of a Halton and scrambled Halton sequences projected in dimensions 7 (basis 17) and 8 (basis 19)

small. A well-known remedy is to randomize the sequence by applying a permutation π_{b_j} on the sequence $(0, 1, ..., b_j)$ for each prime integer b_j . The so-called *scrambled* Halton sequence is then obtained with the following scrambled radical inverse function:

$$S_{\Phi_{b_j}}(i) = \frac{\pi_{b_j}(n_0)}{b_j} + \frac{\pi_{b_j}(n_1)}{b_j^2} + \frac{\pi_{b_j}(n_2)}{b_j^3} + \dots$$
(A.19)

So far, the best known scrambled Halton sequences have been obtained by Braaten and Weller (1979). They proposed a sequential scrambling method which holds 0 fixed and showed that the resulting discrepancies are much better than the one of a conventional Halton sequence. They also set up a table with the suggested permutations for the prime integers up to 53. In Figure A.6b, we used the following permutation as in Braaten and Weller (1979)

$$\pi_{17} = \begin{pmatrix} 0 & 8 & 13 & 3 & 11 & 5 & 16 & 1 & 10 & 7 & 14 & 4 & 12 & 2 & 15 & 6 & 9 \end{pmatrix},$$

$$\pi_{19} = \begin{pmatrix} 0 & 9 & 14 & 3 & 17 & 6 & 11 & 1 & 15 & 7 & 12 & 4 & 18 & 8 & 2 & 16 & 10 & 5 \end{pmatrix}.$$
(A.20)

A.3.2.4 Faure sequence

Faure (1982) introduced some sequences for which he explicitly formulated the discrepancies. He proved that for these sequences the discrepancy is also in the order of $\log(n)^s/n$. Their definition is based on the radical inverse function and the Pascal matrix $C = (c_{kl})$ defined as:

$$c_{kl} = \begin{pmatrix} l-1\\ k-1 \end{pmatrix} = \begin{cases} \frac{(l-1)!}{(k-1)!(l-k)!} & \text{if } k \le l, \\ 0 & \text{otherwise} \end{cases} \quad \forall k, l \in \mathbb{N}^*.$$
(A.21)

A Faure sequence in base *b* is then defined as follows (Franco, 2008):

$$\mathcal{X}_{FAU} = \left\{ x^{(0)}, x^{(1)}, \dots \right\},$$

with $x_j^{(i)} = \phi_b(C^{j-1}i),$ (A.22)
and $C^{j-1} = {j-1 \choose i-1} (j-1)^{j-i} \mod b.$

where C^{j-1} is the generator matrix of the *j*-th dimension of the sequence in dimension *s* and *b* is a prime number such that $b \ge s$.

It has been proven that better discrepancies are obtained when selecting the smallest prime integer higher or equal to *s*. As pointed out in Morokoff and Caflisch (1995), each dimension of a Faure sequence is nothing but a permutation of a Halton sequence. Henceforth, for any prime *b*, an optimal permutation sequence is prescribed and the problems due to large primes in Halton sequences do not occur quickly. Note that other generalizations of a Halton sequence exist. For instance a Sobol' sequence, which is not presented here, is a generalization in base 2. Figure A.7 illustrates three Faure sequences in dimension two with respectively 30, 50 and 100 points. The prime integer used is 2 as it is the smallest higher or equal to the dimension. The designs were also generated with DiceDesign.



Figure A.7: The 30, 50 and 100 first points of a Faure sequence in dimension two

A.3.3 (t, m, s)-nets and (t, s)-sequences

Niederreiter (1987) introduced the concepts of (t, m, s)-nets or simply nets and (t, s)-sequences as low-discrepancy sequences. Dick and Pillichshammer (2010) point out that their constructions were motivated by the fact that there exists no sequence for which the discrepancy actually equals zero. However, it is possible to define a class of subsets in $I_s = [0, 1]^s$ for which the local discrepancy is always zero. Such a sequence is called a net and a (t,s)-sequence is an infinite sequence in I_s for which finite segments are nets. By construction, they are appropriate and good candidates for low-discrepancy sequences. Niederreiter (1987) even showed that their discrepancy has the same upper bound as those of the previously presented sequences with smaller constants and developed expressions of these bounds for some values of *s* and *b*. Let us start by defining the class of subsets we are referring to. They are the so-called *elementary intervals in base b* and are any interval in the form:

$$\prod_{j=1}^{s} \left[\frac{a_j}{b^{d_j}}, \frac{a_j+1}{b^{d_j}} \right), \tag{A.23}$$

where a_i , $d_i \in \mathbb{N}$ and $a_i < b^{d_j}$ $\forall j = \{1, \ldots, s\}$.

An obvious consequence of this definition is that the volume of such an interval always writes in the form b^{-d} , where $d = \sum_{j=1}^{s} d_j$. Figure A.8 shows three subsets. P_1 and P_2 are actual elementary intervals in base 2 as they respectively write $\left[\frac{0}{2^2}, \frac{1}{2^2}\right) \times \left[\frac{1}{2^1}, \frac{2}{2^1}\right)$ and $\left[\frac{2}{2^2}, \frac{3}{2^2}\right) \times \left[\frac{1}{2^2}, \frac{2}{2^2}\right)$. Their respective volumes are $2^{-3} = 1/8$ and $2^{-4} = 1/16$. The subset P_3 ([0.1, 0.4) × [0.1, 0.3083)), despite its volume is 0.625 = 1/16, is not an elementary interval since it could not be cast as in Eq. (A.23).



Figure A.8: Three subsets in I_2 . P_1 and P_2 are elementary intervals in base 2 whereas P_3 is not.

The construction of a (t,*m*,*s*)-net follows from this definition. Consider two integers *t* and *m* such that $0 \le t \le m$. A (*t*, *m*,*s*)-net in base *b* is a sequence \mathcal{X}_{tms} of b^m points in I_s such that $A(P, \mathcal{X}_{tms}, b^m) = b^t$ for all elementary intervals *P* in base *b* of volume $\lambda(P) = b^{t-m}$.

Let us give an example of a (0, 2, 2)-net in base 3. By definition, the number of points of this sequence is $b^m = 3^2 = 9$. The volume of the elementary intervals for this net is $b^{t-m} = b^{0-2} = 1/9$. For this specific volume and considering the definition of the elementary interval above (Eq. (A.23)) there are three couples (d_1, d_2) satisfying $2 = d_1 + d_2$, which are (2, 0), (1, 1) and (0, 2). Figure A.9 illustrates the three classes of elementary intervals: Figure A.9a for the couple (2, 0), Figure A.9b for (1, 1) and Figure A.9c for (0, 2). It could be easily checked that they all have a volume of 3^{-2} . Since the sequence illustrated in this figure is a (0, 2, 2)-net, there should be exactly $b^t = 3^0 = 1$ point in each elementary interval, which is indeed the case. Note that for this figure, we built this sequence with the following formulation:

$$x_{ij} = \frac{A_{ij} + u_{ij}}{9}, \quad i = \{0, \dots, 8\}, \quad j = \{1, 2\}.$$
 (A.24)

where u_{ij} is a realization of a uniform random variable in [0, 1] and A is a sequence carefully chosen and defined by $A_1 = (1, 6, 3, 2, 7, 4, 0, 5, 8)$ and $A_2 = (2, 3, 7, 5, 1, 0, 8, 4, 6)$.



(a) Elementary intervals with the cou-(b) Elementary intervals with the cou-(c) Elementary intervals with the couple (2,0) ple (1,1) ple (0,2)

Figure A.9: A (0, 2, 2)-net in base 3

A (t, s)-sequence is an infinite generalization of nets and is defined as follows. Consider an integer $t \ge 0$. A (t, s)-sequence in base b is a sequence $\{x_0, x_1, x_2, ...\}$ such that for all couples of integers $k \ge 0$ and $m \ge t$, the sequence $\{x_{(kb^m)}, ..., x_{((k+1)b^m-1)}\}$ is a (t, m, s)-net in base b.

Some of the designs introduced above are particular cases of nets or (t, s)-sequences. Many authors have implemented algorithms or tables of (t, m, s)-nets and (t, s)-sequences. Clayman et al. (1997) present an updated survey of known constructions of these sequences for various parameters. First a Latin hypercube of size b is a (0,1,s)-net in base b. In fact, the only elementary intervals of volume b^{-1} are in the form $[0,1) \times \left[\frac{a}{b}, \frac{a+1}{b}\right]$ and $\left[\frac{a}{b}, \frac{a+1}{b}\right] \times [0,1)$. These intervals should contain $b^0 = 1$ point as in a Latin Hypercube design. The (0,2,2)-net in base 3 shown in Figure A.9 above is also (0,1,2)-net in base 9. It is clear from Figure A.9a and Figure A.9c that it is also a Latin hypercube of size 9. Furthermore, it could also be proven that a Latin hypercube based on an Orthogonal array $OA(b^2, s, b, s)$ is a (0,2,s)-net in base b. Again, the (0,2,2)-net in base 3 of the example in Figure A.9 is also a OA(9,2,3,2) as highlighted in Figure A.9b.

Finally Franco (2008) and Thiemard (2000) point out that a van der Corput sequence in base *b* is a (0, 1) sequence in base *b*, a Sobol' sequence is a (t, s) sequence in base 2 and a Faure sequence is a (0, s) sequence in base $b \ge s$.



IMPLEMENTATION OF THE (1+1)-CMA-ES ALGORITHM FOR CONSTRAINED OPTIMIZATION PROBLEMS

We present here a pseudo-algorithm of the (1 + 1)-CMA-ES optimizer we are using. It is based on the paper by Arnold and Hansen (012a). The reader may refer to Hansen (2005) for a welldetailed explanation of the mechanisms underlying the covariance matrix adaptation scheme. Other references such as Arnold and Hansen (2010); Auger and Hansen (2005); Igel et al. (2006); Hansen and Kern (2004) implement various declinations of the CMA-ES method.

We consider here a general formulation of the constrained optimization as introduced in Eq. (2.1):

$$d^* = \arg\min_{d\in\mathbb{D}} \mathfrak{c}(d) \quad \text{subject to:} \begin{cases} f_j(d) \le 0, & \{j = 1, \dots, n_s\}\\ \mathfrak{g}_k(d, z) \le 0, & \{k = 1, \dots, n_h\}. \end{cases}$$
(B.1)

For the sake of clarity, we consider that the environmental variables z remain constant. We also merge the soft and hard constraints. The optimization problem solved using the pseudo-algorithm in the sequel therefore reads:

$$d^* = \arg\min_{d \in \mathbb{D}} \mathfrak{c}(d) \quad \text{subject to: } g_j(d) \le 0, \quad \{j = 1, \dots, n\},$$
(B.2)

where $n = n_s + n_h$.

Algorithm 3 Constrained (1 + 1)-CMA-ES algorithm (Arnold and Hansen, 012a)

Initialization:

Starting point $d^{(0)}$ $d_{\text{best}} = d^{(0)}$, $\mathfrak{c}_{\text{best}} = \mathfrak{c}\left(d^{(0)}\right)$ ▷ Best admissible solution Bounds of the search space d^{\max} and d^{\min} Initial value of the global step size $\sigma^{(0)} \triangleright$ By default $\sigma^{(0)} = 1/3 \max_{\{k=1,\dots,s\}} \left(d_k^{\max} - d_k^{\min} \right)$ $\triangleright P_s = 2/11$ Initial success probability P_s Initial Cholesky decomposition of the CMA-ES covariance matrix A_0 $\triangleright A_0 = I_{s \times s}$ Initial constraints vectors $v_i = \mathbf{0}_{s \times 1}$ and $w_i = \mathbf{0}_{s \times 1}$, $j = \{1, \dots, n\}$ Initial record of succesful steps $s = \mathbf{0}_{s \times 1}$ Convergence threshold η_d and maximum number of iterations N_{max} Initialize counter i = 1Optimally tuned internal parameters of CMA-ES (Arnold and Hansen, 2010, 012a) m = 1 + s/2, c = 2/(s+2), $c_p = 1/12$, $P_t = 2/11$ \triangleright *s* is the dimension $c_c = 1/(s+2), \quad \beta = 0.1/(s+2)$ $c_c^+ = 2/(s^2 + 6), \quad c_c^- = \min(0.4/(s^{1.6} + 1), 1/(2||z||^2 - 1))$

1: repeat

Sample a new design $d^{(i)} \sim d_{\text{best}} + \sigma^{(i)} A^{(i)} z$ 2: where $z = \{z_k \in \mathbb{R} | z_k \sim \mathcal{N}(0, 1), k = 1, ..., s\}$ 3: for i = 1 to n do 4: if $g_i(d) > 0$ then 5: $\boldsymbol{v}_i = (1 - c_c)\boldsymbol{v}_i + c_c \boldsymbol{A}^{(i)} \boldsymbol{z}$ 6: 7: end if end for 8: if *d*^(*i*) is unfeasible then 9: 10: for j = 1 to n do $\boldsymbol{w}_i = \boldsymbol{A}^{(i)^{-1}} \boldsymbol{v}_i$ 11: end for 12: 13: $oldsymbol{A}^{(i)} \leftarrow oldsymbol{A}^{(i)} - rac{eta}{\sum_{j=1}^n \mathbbm{1}_{g_j>0}} \sum_{j=1}^n \mathbbm{1}_{g_j>0} rac{oldsymbol{v}_j oldsymbol{w}_j^T}{oldsymbol{w}_i^T oldsymbol{w}_i}$ 14: else 15: 16: $P_s \leftarrow (1 - c_p) P_s + c_p \mathbb{1}_{c_i < c_{\text{bost}}}$ ▷ Update the success probability 17: 18: $\sigma^{(i)} \leftarrow \sigma^{(i)} \exp\left(\frac{1}{m} \frac{P_s - P_t}{1 - P_t}\right)$ Update the global step size 19: end if 20: if $c_i \leq c_{\text{best}}$ then 21: $d_{\text{best}} = d^{(i)}$ 22: 23: $\mathfrak{c}_{\text{best}} = \mathfrak{c}_i$ $s \leftarrow (1-c)s + \sqrt{c(2-c)}A^{(i)}z$ 24: $\boldsymbol{u} = \boldsymbol{A}^{(i)^{-1}}\boldsymbol{s}$ 25: $A^{(i)} \leftarrow \sqrt{1 + c_c^+} A^{(i)} + \frac{\sqrt{1 + c_c^+}}{\|u\|^2} \left(\sqrt{1 + \frac{c_c^+ \|u\|^2}{1 - c_c^+}} - 1 \right) s u^T$ 26:

else if $\mathfrak{c}_i > \mathfrak{c}_{\text{best}}$ and $i \ge 5$ and $(\mathfrak{c}_{\text{best}} > \mathfrak{c}_{i-1}$ and $\mathfrak{c}_{\text{best}} > \mathfrak{c}_{i-2}$ and $\mathfrak{c}_{\text{best}} > \mathfrak{c}_{i-3}$ and $\mathfrak{c}_{\text{best}} > \mathfrak{c}_{i-4})$ 27: then

28:

 $\boldsymbol{A}^{(i)} \leftarrow \sqrt{1 + c_c^-} \boldsymbol{A}^{(i)} + \frac{\sqrt{1 + c_c^-}}{\|\boldsymbol{z}\|^2} \left(\sqrt{1 - \frac{c_c^- \|\boldsymbol{z}\|^2}{1 + c_c^-}} - 1 \right) \boldsymbol{A}^{(i)} \boldsymbol{z} \boldsymbol{z}^T$ 29: end if 30:

31:

 $i \leftarrow i + 1$ 32:

33: **until** $||d_i - d_{i-1}|| > \eta_d$ or $i = N_{\max}$

REFERENCES

Adler, R. (1981). The geometry of random fields. J. Wiley, New York.

- Agarwal, H., C. K. Mozumder, J. E. Renaud, and L. T. Watson (2007). An inverse-measure-based unilevel architecture for reliability-based design optimization. *Struct. Multidisc. Optim.* 33(3), 217–227.
- Aizerman, M. A., E. A. Braverman, and L. Rozonoer (1964). Theoretical foundations of the potential function method in pattern recognition learning. In *Autom. Rem. Contr.*, Number 25, pp. 821–837.
- Altair Engineering (2009). Radioss theory manual 10.0 version. Altair Engineering.
- Andrieu, C., N. de Freitas, A. Doucet, and M. I. Jordan (2003). An introduction to MCMC for machine learning. *Mach. Learn.* 50(1-2), 5–43.
- Aoues, Y. and A. Chateauneuf (2010). Benchmark study of numerical methods for reliabilitybased design optimization. *Struct. Multidisc. Optim.* 41(2), 277–294.
- Argawal, H. (2004). *Reliability based design optimization: Formulations and methodologies*. Ph. D. thesis, University of Notre Dame, USA.
- Arnold, D. V. and N. Hansen (2010). Active covariance matrix adaptation for the (1+1)-CMA-ES. In M. Pelikan and J. Branke (Eds.), *GECCO*, pp. 385–392. ACM.
- Arnold, D. V. and N. Hansen (2012a). A (1+1)-CMA-ES for constrained optimisation. In T. Soule and J. H. Moore (Eds.), *Genetic and evolutionary computation conference*, pp. 297–304.
- Ash, R. B. (1970). *Basic probability theory*. New York: John Wiley & Sons.
- Asmussen, S. and P. W. Glynn (2007). *Stochastic simulation : algorithms and analysis*. Stochastic modelling and applied probability. New York: Springer.
- Au, S.-K. (2005). Reliability-based design sensitivity by efficient simulation. *Comput. Struct.* 83(14), 1048–1061.
- Au, S.-K. and J. L. Beck (1999). A new adaptive importance sampling scheme for reliability calculations. *Struct. Saf.* 21(2), 135–158.
- Au, S.-K. and J. L. Beck (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Prob. Eng. Mech 16*, 263–277.

- Audet, C., A. J. Booker, J. Dennis, P. D. Frank, and D. W. Moore (2000). A surrogate-modelbased method for constrained optimization. In *Proc. 8th Symposium on Multidisciplinary Anal*ysis and Optimization, Long Beach, CA, USA.
- Auger, A. and N. Hansen (2005). A restart CMA evolution strategy with increasing population size. In *Proc. IEEE Congress on Evolutionary Computation*, pp. 1769–1776. IEEE.
- Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Comput. Stat. Data. An. 66*, 55–69.
- Balesdent, M., J. Morio, and J. Marzat (2013). Kriging-based adaptive importance sampling algorithms for rare event estimation. *Struct. Saf.* 44, 1–10.
- Basudhar, A. and S. Missoum (2009). A sampling-based approach for probabilistic design with random fields. *Comput. Methods Appl. Mech. Engrg.* 198, 3647 3655.
- Bates, S. J., J. Sienz, and V. V. Toropov (2004). Formulation of the optimal Latin hypercube design of experiments using a permutation genetic algorithm. In 45th AIAA/ASME/ASCE/AH-S/ASC Structures, Structural Dynamics and Materials Conference.
- Baudoui, V. (2012). *Optimisation robuste multiobjectifs par modèles de substitution*. Ph. D. thesis, Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France.
- Bellary, S., A. Samad, I. Couckuyt, and T. Dhaene (2015). A comparative study of Kriging variants for the optimization of a turbomachinery system. *Eng. Comput.*, 1–11.
- Bellman, R. E. (1961). *Adaptive control processes A guided tour*. Princeton, New Jersey, U.S.A.: Princeton University Press.
- Bennett, K. P. and C. Campbell (2000). Support vector machines: Hype or halellujah?
- Beyer, H.-G. and B. Sendhoff (2007). Robust optimization: A comprehensive survey. *Comput. Methods Appl. Mech. Engrg.* 196(33–34), 3190–3218.
- Bichon, B., M. Eldred, L. Swiler, S. Mahadevan, and J. McFarland (2008). Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal* 46(10), 2459–2468.
- Bichon, B. J., S. Mahadevan, and M. S. Eldred (2009). Reliability-basedased design optimization using efficient global reliability analysis. In 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Palm Springs, California, Volume 47.
- Blatman, G. (2009). *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis.* Ph. D. thesis, Université Blaise Pascal, Clermont-Ferrand, France.
- Blatman, G. and B. Sudret (2008). Adaptive sparse polynomial chaos expansions application to structural reliability. In *Proc. 4th Int. ASRANet Colloquium, Athens, Greece*.
- Blatman, G. and B. Sudret (2010). Reliability analysis of a pressurized water reactor vessel using sparse polynomial chaos expansions. In D. Straub, L. Esteva, and M. Faber (Eds.), *Proc. 15th IFIP WG7.5 Conference on Reliability and Optimization of Structural Systems, Munich, Germany*, pp. 9–16. Taylor & Francis.
- Blum, J. R. and J. I. Reich (1982). Pointwise ergodic theorems in l.c.a. groups. *Pacific J. Math.* 103(2), 301–306.

- Boer, P.-T. D., D. P. Kroese, S. Mannor, and R. Y. Rubinstein (2004). A tutorial on the crossentropy method. *Ann. Oper. Res.* 134.
- Bompard, M. (2011). *Modèles de substitution pour l'optimisation globale de forme en aérodynamique et méthode locale sans paramétrisation*. Ph. D. thesis, Université de Nice Sophia-Antipolis, France.
- Bourinet, J.-M. (2015). Reliability assessment with adaptive surrogates based on support vec tor machine regression. In G. S. e. M. Papadrakakis, V. Papadopoulos (Ed.), Proc. 1st ECCO-MAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering, Crete Island, Greece, May 25-27.
- Bourinet, J.-M., F. Deheeger, and M. Lemaire (2011). Assessing small failure probabilities by combined subset simulation and support vector machines. *Struct. Saf.* 33(6), 343–353.
- Box, G., S. Bisgaard, and C. Fung (1988). An explanation and critique of Taguchi's contributions to quality engineering. *Qual. Reliab. Eng. Int.* 4(2), 123–131.
- Box, G. and K. Wilson (1951). On the experimental attainment of optimum conditions. *J. Roy. Stat. Soc. B. Met.* 13(1), 1–45.
- Box, G. E. P. and N. R. Draper (1986). *Empirical model building and response surface*. New York, NY, USA: John Wiley & Sons, Inc.
- Braaten, E. and G. Weller (1979). An improved low-discrepancy sequence for multidimensional quasi-Monte Carlo integration. *J. Comput. Phys.* 33(2), 249–258.
- Breiman, L. and P. Spector (1992). Submodel selection and evaluation in regression. The X-random case. *Int. Stat. Rev.* 60, 291–319.
- Breitung, K. (1989). Asymptotic approximations for probability integrals. *Prob. Eng. Mech* 4(4), 187–190.
- Broomhead, D. and D. Lowe (1988). Multivariable functional interpolation and adaptive networks. *AIP Conf. Proc.* 2, 321–355.
- Bucher, C. G. (1988). Adaptive sampling: An iterative fast Monte Carlo procedure. *Struct. Saf.* 5(2), 119–126.
- Cadini, F., A. Gioletta, and E. Zio (2015). Improved metamodel-based importance sampling for the performance assessment of radioactive waste repositories. *Reliab. Eng. Sys. Safety* 134, 188–197.
- Cérou, F., P. Del Moral, T. Furon, and A. Guyader (2012). Sequential Monte Carlo for rare event estimation. *Stat. Comp.* 22(3), 795–808.
- Chandu, S. V. and R. V. Grandhi (1995). General purpose procedure for reliability-based structural optimization under parametric uncertainties. *Adv. Eng. Softw.* 23(1), 7–14.
- Chang, C.-C. and C.-J. Lin (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27. Software available at http: //www.csie.ntu.edu.tw/~cjlin/libsvm.
- Chang, M.-W. and C.-J. Lin (2005). Leave-one-out bounds for support vector regression model selection. *Neural Comput.* 17(5), 1188–1222.

- Chapelle, O. (2002). *Support Vector Machines : principes d'induction, réglage automatique et connaissances a priori.* Ph. D. thesis, Université Pierre et Marie Curie, France.
- Chapelle, O., V. Vapnik, O. Bousquet, and S. Mukherje (2002). Choosing multiple parameters for support vector machines. In N. Cristianini (Ed.), *Mach. Learn.*, Volume 46, pp. 131 159.
- Chapman, O. and A. Crossland (1995). Neural networks in probabilistic structural mechanics. In C. Sundararajan (Ed.), *Probabilistic Structural Mechanics Handbook*, pp. 317–330. Springer US.
- Chateauneuf, A. and Y. Aoues (2008). *Structural design optimization considering uncertainties*, Chapter 9, pp. 217–246. Taylor & Francis.
- Chen, K.-Y. and C.-H. Wang (2007). Support vector regression with genetic algorithms in forecasting tourism demand. *Tourism Manage*. 28(1), 215–226.
- Chen, X., K. Hasselman, Timothy, and D. J. Neil (1997). Reliability based structural design optimization for practical applications. In *38th Structures, Structural Dynamics, and Materials Conference*.
- Cheng, G., L. Xu, and L. Jiang (2006). A sequential approximate programming strategy for reliability-based structural optimization. *Comput. Struct.* 84(21), 1353–1367.
- Cherkassky, V. and Y. Ma (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks* 17(1), 113–126.
- Cherkassky, V. and F. Mulier (2007). Learning from data: concepts, theory, and methods. Wiley.
- Cho, T. M. and B. C. Lee (2011). Reliability-based design optimization using convex linearization and sequential optimization and reliability assessment method. *Struct. Saf.* 33(1), 42–50.
- Clayman, A. T., K. M. Lawrence, G. L. Mullen, H. Niederreiter, and N. J. A. Sloane (1997). Updated tables of parameters of (t,m,s)-nets. *J. Comb. Des.* 7, 381–393.
- Cooke, R. M. (2002). The anatomy of the squizzel: The role of operational definitions in representing uncertainty. *Reliab. Eng. Sys. Safety 85*(1-3), 313–319.
- Cortes, C. and V. Vapnik (1995). Support vector networks. In Mach. Learn., pp. 273–297.
- Cox, D. D. and S. John (1997). SDO: A statistical method for global optimization. In *SIAM Proc. S.*, pp. 315–329.
- Cressie, N. A. C. (1993). *Statistics for spatial data*. Wiley Series in Probability and Mathematical Statistics. New York, Chichester, Toronto: J. Wiley & Sons.
- Currin, C., T. Mitchell, M. Morris, and D. Ylvisaker (1988). A Bayesian approach to the design and analysis of computer experiments. Technical Report ORNL–6498, Oak Ridge Laboratory.
- Currin, C., T. Mitchell, M. Morris, and D. Ylvisaker (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Am. Stat. Assoc.* 86(416), 953–963.
- Damien, P., J. C. Wakefield, and S. G. Walker (1999). Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *J. Royal Stat. Soc. B. Met.* 61, 331–344.

- de Brabanter, K. (2011). *Least-square support vector regression with applications to large-scale data: a statistical approach*. Ph. D. thesis, Katholieke universiteit Leuven, Beligium.
- Deheeger, F. (2008). *Couplage mécano-fiabiliste : ²SMART méthodologie d'apprentissage stochastique en fiabilité*. Ph. D. thesis, Université Blaise Pascal, Clermont-Ferrand, France.
- Deheeger, F. and M. Lemaire (2007). Support vector machine for efficient subset simulations: ²SMART method. In *Proc. 10th Int. Conf. on Applications of Stat. and Prob. in Civil Engineering* (*ICASP10*), *Tokyo, Japan*.
- Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics 38*, 325–339.
- Dennis, J., Jr., and V. J. Torczon (1994). Derivative-free pattern search methods for multidisciplinary design problems. In Proc. 5th AIAA/USAF/NASA/ISSMO Symposium of multidisciplinary Analysis and Optimization, Houston, Texas, USA, pp. 922–932.
- Der Kiureghian, A. and T. Dakessian (1998). Multiple design points in first and second-order reliability. *Struct. Saf.* 20(1), 37–49.
- Dick, J. and F. Pillichshammer (2010). *Digital nets and sequences: discrepancy theory and quasi-Monte Carlo Integration*. New York, NY, USA: Cambridge University Press.
- Ditlevsen, O. and H. Madsen (1996). Structural reliability methods. John Wiley & Sons Inc.
- Doltsinis, I. and Z. Kang (2004). Robust design of structures using optimization methods. *Comput. Method. Appl. M.* 193, 2221–2237.
- Du, X. and W. Chen (2004). Sequential optimization and reliability assessment method for efficient probabilistic design. *J. Mech. Design* 126(2), 225–233.
- Du Bois, P., C. C. Chou, B. B. Fileta, T. B. Khalil, A. I. King, H. F. Mahmood, H. J. Mertz, J. Wismans, P. Prasad, and J. E. Belwafa (2004). Vehicle crashworthiness and occupant protection. American Iron and Steel Institute, USA.
- Dubois, D. and H. Prade (1988). *Possibility theory: an approach to computerized processing of uncertainty.* New York: Plenum Press.
- Dubois, D. and H. Prade (2011). Possibility theory and its applications: Where do we stand? Université Paul Sabatier, Toulouse, France.
- Dubourg, V. (2011). Adaptive surrogate models for reliability analysis and reliability-based design optimization. Ph. D. thesis, Université Blaise Pascal, Clermont-Ferrand, France.
- Dubourg, V., B. Sudret, and J.-M. Bourinet (2011). Reliability-based design optimization using Kriging and subset simulation. *Struct. Multidisc. Optim.* 44(5), 673–690.
- Dubourg, V., B. Sudret, and F. Deheeger (2013). Metamodel-based importance sampling for structural reliability analysis. *Prob. Eng. Mech* 33, 47–57.
- Dubrule, O. (1983). Cross validation of Kriging in a unique neighborhood. J. Int. Ass. Math. Geol. 15(6), 687–699.
- Durrett, R. (2011). Probability: Theory and examples. Cambridge University Press.

- Echard, B., N. Gayton, and M. Lemaire (2011). AK-MCS: an active learning reliability method combining Kriging and Monte Carlo simulation. *Struct. Saf.* 33(2), 145–154.
- Efron, B. and R. J. Tibshirani (1993). An introduction to the bootstrap. Chapman & Hall.
- Enevoldsen, I. and J. Sorensen (1994). Reliability-based optimization in structural engineering. *Struct. Saf.* 15(3), 169–196.
- Evgeniou, T., M. Pontil, and T. Poggio (2000). Regularization networks and support vector machines. *Adv. Comput. Math.* 13(1), 1–50.
- Fang, K.-T., C.-X. Ma, and P. Winker (2002). Centered L₂-discrepancy of random Sampling and Latin hypercube design, and construction of uniform designs. *Math. Comput.* 71(237), 275–296.
- Faure, H. (1982). Discrépances de suites associées à un système de numération (en dimension *s*). *Acta. Arith.* 41, 337–351.
- Fauriat, W. and N. Gayton (2014). AK-SYS: an adaptation of the AK-MCS method for system reliability. *Rel. Eng. & Sys. Safety* 123, 137–144.
- Franco, J. (2008). *Planification d'expériences numériques en phase exploratoire pour la simulation des phénomènes complexes*. Ph. D. thesis, Ecole Nationale Supérieure des Mines, Saint-Etienne, France.
- Franco, J., D. Dupuy, O. Roustant, G. Damblin, and B. Iooss (2013). Package DiceDesign.
- Gao, J., S. R. Gunn, C. J. Harris, and M. Brown (2002). A probabilistic framework for SVM regression and error bar estimation. *Mach. Learn.* 46(1-3), 71–89.
- Gayton, N., J. Bourinet, and M. Lemaire (2003). CQ2RS: a new statistical approach to the response surface method for reliability analysis . *Struct. Saf.* 25(1), 99–121.
- Ghanem, R. G. and P. D. Spanos (2003). *Stochastic finite elements: a spectral approach*. Courier Dover Publications.
- Ginsbourger, D. (2009). *Multiples métamodèles pour l'approximation et l'optimisation de fonctions numériques multivariables*. Ph. D. thesis, École des Mines de Saint-Étienne, France.
- Ginsbourger, D., R. Le Riche, and L. Carraro (2008). A Multi-points criterion for deterministic parallel global optimization based on Gaussian processes. Technical report, École des Mines de Saint-Étienne, France.
- Glaz, B., P. Friedmann, and L. Liu (2008). Surrogate based optimization of helicopter rotor blades for vibration reduction in forward flight. *Struct. Multidisc. Optim.* 35(4), 341–363.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Goupy, J. and L. Creighton (2006). Introduction aux plans d'expériences. Dunod.
- Grinstead, C. M. and L. J. Snell (2006). *Introduction to probability*. American Mathematical Society.

- Gu, X., J. Lu, and H. Wang (2015). Reliability-based design optimization for vehicle occupant protection system based on ensemble of metamodels. *Struct. Multidisc. Optim.* 51(2), 533–546.
- Gu, X., G. Sun, G. Li, L. Mao, and Q. Li (2013). A comparative study on multiobjective reliable and robust optimization for crashworthiness design of vehicle structure. *Struct. Saf.* 48(3), 669–684.
- Gunn, S. (1998). Support vector machines for classification and regression. Technical Report ISIS-1-98, Dpt. of Electronics and Computer Science, University of Southampton, UK.
- Guo, L., P.-S. Ge, M.-H. Zhang, L. Li, and Y. Zhao (2012). Pedestrian detection for intelligent transportation systems combining AdaBoost algorithm and support vector machine. *Expert Syst. Appl.* 39(4), 4274–4286.
- Halton, J. H. (1960). On the Efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.* 2(1), 84–90.
- Hammersley, J. M. (1960). Monte carlo methods for solving multivariate problems. *Ann. NY Acad. Sci.* 86(3), 844–874.
- Hansen, N. (2005). The CMA Evolution Strategy: A tutorial.
- Hansen, N. and S. Kern (2004). Evaluating the CMA-evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature - PPSN VIII*, Volume 3242 of *Lecture Notes in Computer Science*, pp. 282–291. Springer Berlin Heidelberg.
- Hansen, N. and A. Ostermeier (2001). Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* 9(2), 159–195.
- Hasofer, A. M. and N. C. Lind (1974). Exact and invariant second-moment code format. *J. Eng. Mech. Div.-ASCE* 100(1), 111–121.
- Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer Series in Statistics. Springer London.
- Hlawka, E. (1964). Discrepancy and uniform distribution of sequences. *Composito Mathematica* 16, 98–91.
- Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.
- Hora, C. S. (1996). Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliab. Eng. Syst. Safety* 54, 217–223.
- Hu, C. and B. Youn (2011). Adaptive-sparse polynomial chaos expansion for reliability analysis and design of complex engineering systems. *Struct. Multidisc. Optim.* 43(3), 419–442.
- Hu, J. and P. Hu (2009). On the performance of the cross-entropy method. In B. J. A. D.
 M. D. Rossetti, R. R. Hill and R. G. Ingalls (Eds.), *Proceedings of the 2009 Winter Simulation Conference*, WSC '09, pp. 459–468.
- Hurtado, J. E. (2007). Filtered importance sampling with support vector margin: A powerful method for structural reliability analysis. *Struct. Saf.* 29(1), 2–15.

- Hurtado, J. E. and D. A. Alvarez (2001). Neural-network-based reliability analysis: a comparative study. *Comput. Method. Appl. M.* 191, 113–132.
- Hurtado, J. E. and D. A. Alvarez (2003). Classification approach for reliability analysis with stochastic finite-element modeling. *J. Struct. Eng.* 129(8), 1141–1149.
- Igel, C., T. Suttorp, and N. Hansen (2006). A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In *Proc. 8th Annual Conference on Genetic and Evolutionary Computation GECCO*, pp. 453–460. ACM.
- Janusevskis, J. and R. Le Riche (2013). Simultaneous Krigingiging-based estimation and optimization of mean response. *J. Global Optim.* 55(2), 313–336.
- Johnson, M. E., L. M. Moore, and D. Ylvisaker (1990). Minimax and maximin distance designs. *J. Stat. Plan. Inference* 26(2), 131–148.
- Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *J. Global Optim.* 21(4), 345–383.
- Jones, D. R., M. Schonlau, and W. J. Welch (1998). Efficient global optimization of expensive black-box functions. *J. Global Optim.* 13(4), 455–492.
- Jourdan, A. (2000). *Analyse statistique et échantillonnage d'expériences simulées*. Ph. D. thesis, Université de Pau et des Pays de l'Adour, Pau, France.
- Kang, Z. (2005). *Robust design optimization of structures under uncertainties*. Ph. D. thesis, Fakultät Luft- und Raumfahrttechnik und Geodäsie der Universität Stuttgart, Germany.
- Kharmanda, G., A. Mohamed, and M. Lemaire (2002). Efficient reliability-based design optimization using a hybrid space with application to finite element analysis. *Struct. Multidisc. Optim.* 24(3).
- Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. *Science* 220(4598), 671–680.
- Kleijnen, J. (2007). Kriging metamodeling in simulation: A review. Discussion Paper 2007-13, Tilburg University, Center for Economic Research.
- Klir, G. J. and T. A. Folger (1987). *Fuzzy Sets, Uncertainty, and Information*. Upper Saddle River, NJ, USA: Prentice-Hall.
- Koehler, J. R. and A. Owen (1996). Computer experiments. In S. Ghosh and C. Rao (Eds.), *Handbook of Statistics*, Volume 13, pp. 261–308. North Holland.
- Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *J. Chem., Metal. and Mining Soc. of South Africa* 52 (6), 119–139.
- Kroese, D. P., S. Porotsky, and R. Y. Rubinstein (2006). The cross-entropy method for continuous multi-extremal optimization. *Meth. Comp. Appl. Prob. 8*, 383–407.
- Kuschel, N. and R. Rackwitz (1997). Two basic problems in reliability-based structural optimization. *Math. Method. Oper. Res.* 46(3), 309–333.

- Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *J. Basic Eng. (ASME) 86*, 97–106.
- Kwok, J. T. (2001). Linear dependency between *ε* and the input noise in *ε*-support vector regression. In *Lect. Notes Comput. Sc.*, London, UK, pp. 405–410. Springer-Verlag.
- Lataniotis, C., M. S. and B. Sudret (2015). UQLab user manual Kriging. Technical report, Chair of Risk, Safety & Uncertainty Quantification, ETH Zurich. Report #UQLab-V0.9-105.
- Lebrun, R. and A. Dutfoy (2009a). An innovating analysis of the Nataf transformation from the copula viewpoint. *Prob. Eng. Mech* 24(3), 312–320.
- Lebrun, R. and A. Dutfoy (2009b). Do Rosenblatt and Nataf isoprobabilistic transformations really differ? *Prob. Eng. Mech* 24(4), 577–584.
- Lee, J. A. and M. Verleysen (2007). *Nonlinear dimensionality reduction* (1st ed.). Springer Publishing Company.
- Lee, J.-O., Y.-S. Yang, and W.-S. Ruy (2002). A comparative study on reliability-index and targetperformance-based probabilistic structural design optimization. *Comput. Struct.* 80(3:4), 257–269.
- Lee, K.-H., G.-J. Park, and W.-S. Joo (2005). A global robust optimization using the Kriging based approximation model. *The Japan Society of Mechanical Engineers*, 779–788.
- Lee, T. and J. Jung (2008). A sampling technique enhancing accuracy and efficiency of metamodel-based RBDO: constraint boundary sampling. *Comput. Struct. 86*, 1463 1473.
- Lettvin, J. Y., H. R. Maturana, W. S. McCulloch, and W. Pitts (1959). What the frog's eye tells the frog's brain. In *Proc. Institute of Radio Engineers*.
- Liang, J., Z. Mourelatos, and E. Nikolaidis (2007). A single-loop approach for system reliabilitybased design optimization. *J. Mech. Des.* 129(12), 1215 – 1224.
- Liang, J., Z. Mourelatos, and J. Tu (2004). A single-loop method for reliability-base desgin optimization. In *Proc: DETC'04 ASME 2004 Design engineering technical conferences and computer and information in engineering conference, Sept. 28 - Oct. 2, Salt Lake City, Utah, USA.*
- Lin, C.-j. and R. C. Weng (2004). Simple probabilistic predictions for support vector regression. Technical report, National Chengchi University, Taipei, Taiwan.
- Lin, S.-W., Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing* 8(4), 1505–1512.
- Lin, S.-W., K.-C. Ying, S.-C. Chen, and Z.-J. Lee (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Syst. Appl.* 35(4), 1817–1824.
- Lindley, D. V. (1956). On a measure of the information provided by an experiment. *Ann. Math. Stat.* 27(4), 986–1005.
- Liu, P.-L. and A. Der Kiureghian (1986). Multivariate distribution models with prescribed marginals and covariance. *Probabilistic Engineering Mechanics* 1(2), 105–112.

- Liu, P.-L. and A. Der Kiureghian (1992). Optimization algorithms for structural reliability analysis. *Struct. Saf. 9*, 161–177.
- Lophaven, S. N., H. B. Nielsen, and J. Sondergaard (2002a). Aspects of the Matlab toolbox DACE. Technical University Denmark, Kongens Lyngby, Denmark.
- Lophaven, S. N., H. B. Nielsen, and J. Sondergaard (2002b). DACE A Matlab Kriging toolbox – Version 2.0. Technical University Denmark, Kongens Lyngby, Denmark.
- Lütkepohl, H. (1996). Handbook of Matrices. Wiley & Sons, 1996.
- Madsen, H. and P. Hansen (1992). A comparison of some algorithms for reliability based structural optimization and sensitivity analysis. In R. Rackwitz and P. Thoft-Christensen (Eds.), *Reliability and Optimization of Structural Systems '91*, Volume 76 of *Proceedings of the 4th IFIP WG7.5 Conference, Munich, Germany*, pp. 443–451. Springer Berlin Heidelberg.
- Madsen, H., S. Krenk, and N. Lind (1986). *Methods of structural safety*. Prentice Hal, Inc. Englewood Cliffs.
- Matheron, G. (1962). Traité de géostatistique appliquée. Editions Technip.
- Mattera, D. and S. Haykin (1999). Advances in kernel methods. Chapter Support vector machines for dynamic reconstruction of a chaotic system, pp. 211–241. Cambridge, MA, USA: MIT Press.
- McCulloch, W. S. and W. Pitts (1988). Neurocomputing: foundations of research. Chapter A logical calculus of the ideas immanent in nervous activity, pp. 15–27. Cambridge, MA, USA: MIT Press.
- McKay, M. D., R. J. Beckman, and W. J. Conover (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 42(1), 55–61.
- Meckesheimer, M., A. J. Booker, R. R. Barton, and T. W. Simpson (2002). Computationally inexpensive metamodel assessment strategies. *AIAA Journal* 40, 2053–2060.
- Medina, J. C. and A. Taflanidis (2014). Probabilistic measures for assessing appropriateness of robust design optimization solutions. *Struct. Multidisc. Optim.* 51, 813–834.
- Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21, 1087.
- Mitchell, M. (1995). Genetic algorithm: An overview. Complexity 1, 31–39.
- Mockus, J. (1974). On Bayesian methods for seeking the extremum. In *Proc. of the IFIP Technical Conference*, London, UK, UK, pp. 400–404. Springer-Verlag.
- Momma, M. and K. P. Bennett (2002). A pattern search method for model selection of support vector regression. In *SIAM Proc. S.* SIAM.
- Morokoff, W. J. and R. E. Caflisch (1995). Quasi-Monte Carlo integration. J. Comput. Phys. 122(2), 218–230.

- Morris, M. D. and T. J. Mitchell (1995). Exploratory designs for computational experiments. *J. Stat. Plan. Inference* 43(3), 381–402.
- Mourelatos, Z. and J. Liang (2006). A methodology for trading-off performance and robustness under uncertainty. *Transactions of ASME 128*.
- Moustapha, M., B. Sudret, J.-M. Bourinet, and B. Guillaume (2014a). Metamodeling for crashworthiness design: Comparative study of Kriging and support vector regression. In E. S. de Cursi (Ed.), *Proc. 2nd International Symposium on Uncertainty Quantification and Stochastic Modeling, Rouen France, June 23-25.*
- Moustapha, M., B. Sudret, J.-M. Bourinet, and B. Guillaume (2014b). Reliability-based design optimization of an automotive body structure under crashworthiness constraints. In J. Li. and Y. Zhao (Eds.), *Proc. 17th IFIP WG7.5 Conference on Reliability and Optimization of Structural Systems, Huangshan, China.*
- Moustapha, M., B. Sudret, J.-M. Bourinet, and B. Guillaume (2015). Adaptive Kriging reliabilitybased design optimization of an automotive body structure under crashworhtiness constraints. In T. E. Haukaas (Ed.), *Proc. 12th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP12), Vancouver, Canada, July 12-15.*
- Mulvey, J. M., R. J. Vanderbei, and S. A. Zenios (1995). Robust optimization of large-scale systems. *Oper. Res.* 43(2), 264–281.
- Myung, I. J. (2003). Tutorial on maximum likelihood estimation. J. Math. Psychol. 47(1), 90–100.
- Nair, V. N., B. Abraham, J. MacKay, G. Box, R. N. Kacker, T. J. Lorenzen, J. M. Lucas, R. H. Myers, G. G. Vining, J. A. Nelder, et al. (1992). Taguchi's parameter design: a panel discussion. *Technometrics* 34(2), 127–161.
- Nataf, A. (1962). Détermination des distributions de probabilités dont les marges sont données. Compte Rendu de l'Académie des Sciences 225.
- Neal, R. (2003). Slice sampling. Annals of Statistics 31(3), 705–767.
- Niederreiter, H. (1987). Point sets and sequences with small discrepancy. *Monatshefte für Mathematik* 104, 273–338.
- Niederreiter, H. (1992). *Random number generation and quasi-Monte Carlo methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.
- Oberkampf, W. L., S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin (2002). Error and uncertainty in modeling and simulation. *Rel. Eng. & Sys. Safety* 75(3), 333–357.
- O'Hagan, A. and J. E. Oakley (2004). Probability is perfect, but we can't elicit it perfectly. *Reliab. Eng. Syst. Safe.* 85(1-3), 239–248.
- Osuna, E., R. Freund, and F. Girosi (1997). Training support vector machines: an application to face detection. In *Proc. CPVR'97, June 17-19, 1997, Puerto Rico*, pp. 130–136.
- Owen, A. and Y. Zhou (2000). Safe and effective importance sampling. *Journal of the American Statistical Association* 95(449), 135–143.

- Owen, A. B. (1992). Orthogonal arrays for computer experiments, integration and visualization. *Stat. Sinica* 2, 439–452.
- Owen, A. B. (1994). Randomly permuted (t,m,s)-nets and (t,s)-sequences. Technical report, Department of Statistics, Standford University, Standford, California,USA.
- Pai, P.-F. and W.-C. Hong (2006). Software reliability forecasting by support vector machines with simulated annealing algorithms. *J. Soft. Software* 79(6), 747–755.
- Pan, F., P. Zhu, and Y. Zhang (2010). Metamodel-based lightweight design of B-pillar with TWB structure via support vector regression. *Comput. Struct.* 88, 36–44.
- Papadrakakis, M. and N. D. Lagaros (2002). Reliability-based structural optimization using neural networks and Monte Carlo simulation. *Comput. Method Appl. M.* 191(32), 3491–3507.
- Park, J.-S. (1993). Optimal Latin-Hypercube designs for computer experiments. J. Stat. Plan. Infer. 39(1), 95–111.
- Picheny, V., D. Ginsbourger, O. Roustant, R. T. Haftka, and N. H. Kim (2010). Adaptive designs of experiments for accurate approximation of a target region. *J. Mech.Design* 132(7).
- Platt, J. C. (1999). Advances in kernel methods. Chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208. Cambridge, MA, USA: MIT Press.
- Polak, E., R.-B. Wets, and A. der Kiureghian (2000). On an approach to optimization problems with a probabilistic cost and or constraints. In G. Pillo and F. Giannessi (Eds.), *Nonlinear Optimization and Related Topics*, Volume 36 of *Applied Optimization*, pp. 299–315. Springer US.
- Prager, W. and J. Taylor (1968). Problems of optimal structural design. J. Appl. Mech. 35(1), 102–106.
- Pronzato, L. and W. G. Müller (2011). Design of computer experiments: space filling and beyond. *Stat. Comput.* 22(3), 681–701.
- PSA Peugeot Citroën (2012). Rapport d'activité et de développement durable.
- Rackwitz, R. and B. Fiessler (1978). Structural reliability under combined random load sequences. *Comput. Struct.* 9(5), 489–494.
- Ramsey, F. P. (1926). Truth and probability. In R. Braithwaite (Ed.), *The foundations of mathematics and other logical essays*, Chapter VII, pp. 156–198.
- Ranjan, P., D. Bingham, and G. Michailidis (2008). Sequential experiment design for contour estimation from complex computer codes. *Technometrics* 50(4), 527–541.
- Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian processes for machine learning (adaptive computation and machine learning)*. The MIT Press.
- Rippa, S. (1999). An algorithm for selecting a good value for the parameter *c* in radial basis function interpolation. *Adv. Comput. Math.* 11(2-3), 193–210.
- Rosenblatt, M. (1952). Remarks on a multivariate transformation. *The Annals of Mathematical Statistics* 23(3), 470–472.

- Roustant, O., D. Ginsbourger, and Y. Deville (2012). DiceKriging, DiceOptim: two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *J. Stat. Software* 51(1), 1–55.
- Roux, W., N. Stander, F. Günther, and H. Müllerschön (2006). Stochastic analysis of highly non-linear structures. *Int. J. Numer. Meth. Eng.* 65, 1221–1241.
- Rowe, W. D. (1994). Understanding uncertainty. Risk Analysis 14(5), 743-750.
- Rubinstein, R. and W. Davidson (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability* 1(2), 127–190.
- Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *Eur. J. Oper. Res.* 99, 89–112.
- Sacks, J., W. Welch, T. Mitchell, and H. Wynn (1989). Design and analysis of computer experiments. *Stat. Sci.* 4(4), 409–423.
- Santner, T., B. Williams, and W. Notz (2003). *The design and analysis of computer experiments*. Springer, New York.
- Santos, S., L. Matioli, A. T. Beck, et al. (2012). New optimization algorithms for structural reliability analysis. *Computer Modeling in Engineering & Sciences(CMES)* 83(1), 23–55.
- Sasena, M. J. (2002). *Flexibility and efficiency enhancements for constrained global design optimization with Kriging approximations*. Ph. D. thesis, University of Michigan, USA.
- Saunders, C., A. Gammerman, and V. Vovk (1998). Ridge regression learning algorithm in dual variables. In *Proc. of the 15th International Conference on Machine Learning*, ICML '98, San Francisco, CA, USA, pp. 515–521. Morgan Kaufmann Publishers Inc.
- Schmit, L. A. J. (1960). Structural design by system synthesis. In *Proc. 2nd international conference on eletronic computation, Pittsburgh PA,* pp. 105–122.
- Schöbi, R. and B. Sudret (2014). PC-Kriging: A new meta-modelling method and its applications to quantile estimation. In J. Li and Y. Zhao (Eds.), Proc. 17th IFIP WG7.5 Conference on Reliability and Optimization of Structural Systems, Huangshan, China. Taylor & Francis.
- Schöbi, R., B. Sudret, and S. Marelli (2015). Rare event estimation using polynomial-chaoskriging. *ASCE J. Risk Uncert. Eng. Sys.*. Accepted.
- Schonlau, M. (1997). *Computer experiments and global optimization*. Ph. D. thesis, University of Waterloo, Ontario, Canada.
- Schonlau, M., W. J. Welch, and D. R. Jones (1998). Global versus local search in constrained optimization of computer models. *New Developments and Applications in Experimental Design*.
- Schoofs, A. (1993). Structural optimization history and state-of-the-art. In J. Dijksman and F. Nieuwstadt (Eds.), *Topics in Applied Mechanics*, pp. 339–345. Springer Netherlands.
- Schuëller, G. and R. Stix (1987). A critical appraisal of methods to determine failure probabilities. *Struct. Saf.* 4(4), 293–309.

- Scott, D. and J. Thompson (1983). Probability density estimation in high dimensions. In J. Gentle (Ed.), *Proc. 15th Symposium on the Interface*.
- Shafer, G. (1976). A Mathematical Theory of Evidence. Princeton: Princeton University Press.
- Shafer, G. and V. Vovk (2006, 02). The Sources of Kolmogorov's Grundbegriffe. *Statist. Sci.* 21(1), 70–98.
- Shan, S. and G. G. Wang (2008). Reliable design space and complete single-loop reliabilitybased design optimization. *Rel. Eng. & Sys. Safety* 93(8), 1218–1230.
- Shanno, D. F. (1970). Conditioning of quasi-Newton methods for function minimization. *Math. Comput.* 24(111), 647–656.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.* 27, 379–423, 623–656.
- Shewry, M. C. and H. P. Wynn (1987). Maximum entropy sampling. J. Appl. Stat. 14(2), 165–170.
- Smola, A. J. and B. Schölkopf (2004). A tutorial on support vector regression. *Stat. Comp.* 14(3), 199–222.
- Su, R., L. Gui, and Z. Fan (2011). Multi-objective optimization for bus body with strength and rollover safety constraints based on surrogate models. *Struct. Multidisc. Optim.* 44(3), 431–441.
- Sudret, B. (2007). Uncertainty propagation and sensitivity analysis in mechanical models Contributions to structural reliability and stochastic spectral methods. Université Blaise Pascal, Clermont-Ferrand, France. Habilitation à Diriger des Recherches.
- Suykens, J. A. K. and J. Vandewalle (1999). Least squares support vector machine classifiers. *Neural Process. Lett.* 9(3), 293–300.
- Taflanidis, A. (2007). *Stochastic system design and applications to stochastic robust structural control.* Ph. D. thesis, California Institute of Technology, Pasadena, California, USA.
- Taflanidis, A. and J. L. Beck (2008). Stochastic subset optimization for optimal reliability problems. *Prob. Eng. Mech* 23, 324–338.
- Taflanidis, A. A. and J. L. Beck (2009). Stochastic subset optimization for reliability optimization and sensitivity analysis in system design. *Comput. Struct.* 87, 318–331.
- Taguchi, G. and M. Phadke (1989). Quality engineering through design optimization. In K. Dehnad (Ed.), *Quality Control, Robust Design, and the Taguchi Method*, pp. 77–96. Springer.
- Tang, B. (1993). Orthogonal array-based Latin Hypercubes. J. Am. Stat. Assoc. 88(424).
- Thiemard, E. (2000). *Sur la calcul et la majoration de la discrépance à l'origine*. Ph. D. thesis, Université Polytechnique Fédérale de Lausanne, Switzerland.
- Thole, C.-A. and L. Mei (2003). Reasons for scatter in crash simulations results. In *4th European LS-DYNA Users Conference, Ulm, Germany.* Crash / Automotive Applications III.

- Trosset, M. (1997). Taguchi and robust optimization. Technical report, Rice University, Houston, USA.
- Tsompanakis, Y., N. Lagaros, and M. Papadrakis (2008). *Structural design optimization considering uncertainties.* Taylor & Francis.
- Tu, J. and K. K. Choi (1997). A performance measure approach in reliability-based structural optimization. Technical report, Center for computer-aided design, The University of Iowa, Iowa City, IA.
- Tu, J., K. K. Choi, and Y. H. Park (1999). A new study on reliability-based design optimization. *J. Mech. Des.* 121, 557 – 564.
- van der Corput, J. G. (1935). Verteilungsfunktionen i and ii. Volume 38, pp. 813–821 and 1058– 1066.
- van Laarhoven, P. J. M. and E. H. L. Aarts (Eds.) (1987). *Simulated Annealing: Theory and Applications*. Norwell, MA, USA: Kluwer Academic Publishers.
- Vanderbei, R. J. (1994). LOQO: an Interior point code for quadratic programming. Technical report, Optim. Method. Softw.
- Vapnik, V. and O. Chapelle (2000). Bounds on error expectation for support vector machines. *Neural Comput.* 12(9), 2013–2036.
- Vapnik, V., S. E. Golowich, and A. Smola (1997). Support vector method for function approximation, regression estimation, and signal processing. In *Advances in Neural Information Processing Systems 9*, pp. 281–287. MIT Press.
- Vapnik, V. N. (1995). The nature of statistical learning theory. New York, NY, USA: Springer-Verlag.
- Vapnik, V. N. (1998). Statistical learning theory. Wiley.
- Vapnik, V. N. and A. Y. Chervonenkis (1971). On the uniiform convergence of relative frequencies of events to their probabilities. *Theor. Prob. Appl.* 16(2), 264–280.
- Vazquez, E. (2005). Modélisation comportementale de systèmes non-linéaires multivariables par méthodes à noyaux et applications. Ph. D. thesis, Université Paris XI Orsay, France.
- Venkayya, V., N. Khot, and V. Reddy (1968). Optimization of structures based on the study of strain energy distribution. Technical report, AFFDL, Aerospace Engineers, Structures Division.
- Verleysen, M. (2003). Learning high-dimensional data. In S. A. et al. (Ed.), *Limitations and Future Trends in Neural Computation 186*, pp. 141–162.
- Viana, F. A. C., G. Venter, and V. Balabanov (2010). An algorithm for fast optimal Latin Hypercube design of experiments. *Int. J. Numer. Methods Eng.* 82(2), 135–156.
- Walter, C. and G. Defaux (2015). Rare event simulation: A point process interpretation with application in probability and quantile estimation. In T. Haukaas (Ed.), *Proc. 12th International Conference on Applications of Statistics and Probability in Civil Engineering (ICASP12), Vancouver, Canada, July 12-15.*

- Wang, L. and S. Kodiyalam (2002). An efficient method for probabilistic and robust design with non-normal distributions. In 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 22-25, 2002, Denver, Colorado, USA.
- Wauquiez, C. and R. Zeitouni (2008). Reducing the scatter of vehicle crashworthiness. Technical report, PSA Peugeot Citroën.
- Wolpert, D. H. and W. G. Macready (1997). No free lunch theorems for optimization. *Trans. Evol. Comp* 1(1), 67–82.
- Wu, Y.-T., H. Millwater, and T. Cruse (1990). An advanced probabilistic structural analysis method for implicit performance functions. *AIAA* 43(4), 874–884.
- Xu, F., G. Sun, G. Li, and Q. Li (2013). Crashworthiness design of multi-component tailorwelded blank (TWB) structures. *Struct. Multidisc. Optim.* 48(3), 653–667.
- Youn, B., K. Choi, and L. Du (2005). Enriched performance measure approach for reliabilitybased design optimization. *AIAA Journal* 43(4), 874–884.
- Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst. 1, 3–28.
- Zhu, P., F. Pan, W. Chen, and S. Zhang (2012). Use of support vector regression in structural optimization: Application to vehicle crashworthiness design. *Math. Comput. Simul.*.
- Zimmermann, H.-J. (1996). *Fuzzy set theory and Its applications*. Norwell, MA, USA: Kluwer Academic Publishers.