POLITECNICO DI TORINO

SCUOLA DI DOTTORATO
Dottorato in Ingegneria Aerospaziale - XXIV ciclo

Tesi di Dottorato

Space Systems Conceptual Design

Analysis methods for engineering-team support

Guido Ridolfi

Tutori
Prof. Sergio Chiesa
Dr. Sabrina Corpino
Prof. B.A.C. Ambrosius
Dr. Erwin Mooij

Coordinatore del corso di dottorato
Prof. Fulvia Quagliotti

Aprile 2013

Acknowledgements

It all started on a napkin, in a bar, drinking coffee at midnight. The transformation from napkin to dissertation has been an entire journey, made possible by the people I have met, technology used, the environment around me and personal dedication (and caffeine). Just like any other achievement resulting from the human intellect, this PhD dissertation is the result of a balance between factors that have influenced my personal and professional life. The human factor is certainly the most important one. It encompasses all the people that helped me, facilitating my path and making me feel supported. Thank you all.

Professor Chiesa and professor Ambrosius, you believed in my ideas. You put me in the position to pursue them and to explore the boundaries of my limits at the same time. Thank you.

Erwin, you supported me since my master's thesis. You introduced DoE to me and after four years this is the result. Your attention to the smallest details strikes me, and the determination with which, day-by-day, you help students like me to achieve their goals (just because you can) strikes me even more so. You taught me valuable lessons, in life and research, but there is something I will never-learn: hyphenation. Remember, *Barolo* is the answer, even if she did not say so. Thank you.

Sabrina, you trusted me for working on something that was very important to you. Resultantly, it become very important to me as well, and made me feel part of something bigger. That closing with *Jack Nicholson* was legendary, I am sure you will never forget it, I will not. Thank you.

I am thankful to all the committee members for the time they dedicated to reading this thesis. Many thanks to Ron Noomen: your point of view regarding the clarity (or still unclarity after all) of figures and passages in my thesis has been very valuable to me. Kees Sudmeijer, your advice on *hot* materials and cooling mechanisms was fundamental. Further, having you as co-coach for the DSE was a unique experience, thank you. Dominic I truly appreciate your patience guiding me through the complicated maze that your *C++* code is to me. Thank you. Fabrizio, your experience and point of view always played an important role for me. You helped me a lot through the process and I am very thankful for that. Marc and Wouter, without you many of the simulations that frustrated me so much could not be possible, thank you.

The space industry in Europe is overwhelmed by excellent professionals. The engineering group headed by Piero Messidoro in Thales Alenia Space - Italy, and the people at the Concurrent Design Facility in ESA, headed by Massimo Bandecchi, gave me plenty of support and valuable feedback. Thank you all. I would also like to acknowledge the support from the people at J-CDS. J-CDS is a very innovative, open-minded, and fun place to work. Thank you.

One of the privileges of working at a university is that you meet extraordinary people, coworkers at first and friends the day after, who, like you, are following a vision and challenging the *status quo*. To colleagues and friends at Politecnico di Torino and Delft University of Technology: it has been an honor sharing this journey with you. The aspect that amazed me most, is that you always have an answer, an idea, a point of view, no matter how complex or futile the topic is. Thank you. A special thanks to you Relly. Facilitating is your job, and you do it with great passion.

Strangely enough, besides my PhD I also had a life and I was very fortunate to have met great people along the way. Alby and Dave, thank you for the good glass of whisky in-between simulations and our sporadic (failed) attempts at staying in shape. *Club230*, you hosted me several times and I felt at home each time. You are the best *bunch* of people one can hope to eat *stamppot* with. Friends all over Europe and beyond, (former) Dutch expats, *neo* British and *neo* German, *Primers*, former *terroni* now perfectly adapted to northern habits, and you lifelong friends from Velina. Thank you all, you made this journey much more pleasurable and interesting.

Relatives from all over the boot, you made me feel like I was doing something really big even if I knew I was not, thank you.

Cristina and Filippo, you always gave me shelter when I needed even without me asking. I will always be grateful. Thank you.

A special thanks to my parents. You always pushed me upwards and onwards. You showed me how to fight for what I believe and yet to live according to the values you taught me. If I was able to overcome the obstacles that, day-by-day, showed up at my door, it is also thanks to you. Obstacles in life are there for a reason, no matter if they are external or they come directly from the depths of your soul. Obstacles are there to be overtaken and one should look forward to the next one. If you find it too difficult for you, then change perspective. Look at your problem from another point of view; you may find it is not too hard after all. If that does not work out, ask for help. You are certainly not alone in your journey. Capito Mari?

Iole, you shared with me every single moment. When I get carried away with programming I am not always easy to get along with (and that's clearly an understatement), but you always did it! Thank you for helping me through; the end of this journey is just another step forward in our own.

Guido Ridolfi,
January 2013.

Summary

The cover of this thesis shows some humorous illustrations representing partial points of view of several discipline/domain experts contributing to the design of a spacecraft. These different conceptualizations of the spacecraft are purposely exaggerated. Nevertheless, they clearly show that there is a need for balancing requirements and technical solutions for the final system to be the best compromise between often conflicting design *forces*. We believe that collaboration between the stakeholders of the design cycle is the key to successfully designing new systems. The purpose of this thesis is to propose and describe design methods capable of supporting the stakeholders during the design cycle, facilitating their decision process as to reach balanced design solutions in a more consistent and structured manner. The research can be framed as work performed in the field of Operational Research, a field of research that encompasses approaches for facilitating decision making and promote design efficiency.

The initial phase of the design cycle of a space system, also called conceptual design phase, is crucial for the success of the system and the mission it will complete. Up to 70% of the costs are locked-in during this phase, and most of the decisions (technical and managerial) taken at this stage will heavily affect the entire life of the system. Though so important, a very limited amount of resources is allocated for the completion of the conceptual design phase (when compared to the resources used for the entire life of the system, including detailed design, manufacturing, and operations). The tendency of space agencies and industries has been to request (paradoxically) faster, better, and cheaper conceptual design baselines. This paradox has been faced by adopting concurrent design during conceptual phases in place of the more common sequential approach. A concurrent approach to the design of a system means that discipline/domain experts together with the customer all work in parallel (as opposed to working one after each other, sequentially), at the same time; maintaining a high level of collaboration and communication between each other. This facilitates the exchange of technical information and promotes discussion and confrontation. This approach finally leads to early discovery and resolution of potential design show-stoppers and/or inconsistencies. Real-time exchange of technical data and information between engineering team members has been demonstrated to have a large potential. In the European Space Agency, for instance, more than fifteen years of experience in implementing concurrent design for conceptual phases has led to an effective reduction of the costs (by a factor of 2) and development time (by a factor of 4). This was made possible by the utilization of a concurrent design infrastructure (the Concurrent Design Facility, CDF), a state-of-the-art facility that allows a team of experts from several disciplines/domains to apply the concurrent engineering approach. The CDF is not unique in the world; other concurrent design infrastructures, developed by space agencies and private organizations, exist.

Concurrent design infrastructures are reaching maturity level, therefore we believe that time is mature enough for integrated applications to be used on top of them. We believe that conceptual design in general could benefit from the utilization of more structured analysis methods, specifically developed or adapted for this important design phase. The main objective of the thesis is therefore to provide several design approaches to support the engineering team during the conceptual design activities. The goal is to promote efficient exploitation of the models during concurrent design, enhancing the exchange of information and promoting discussions even more.

One of the assumptions that drives this research is that the conceptual design of a new, complex space system is performed by using mathematical models of the system, its elements and its operating environment. This is the case for all engineering fields, where mathematical models are used to predict the performance of the system given the settings of the design parameters influencing it. The design parameters represent the degrees of freedom of the engineering team whose main objective during the technical design is to set these parameters such that the system performs as required, possibly at the minimum cost.

At first we introduce the concept of sampling. Sampling the design space (*i.e.,* the mathematical space having the design parameters as dimensions) means selecting the points in the design space that will be used to simulate mathematical models to compute the performance. This is the most practical way to quantitatively assess the effect of the design parameters on the performance of the system. The most common approach to sampling is certainly a (pseudo)random one. In this thesis we propose alternative methods for more efficient sampling in the presence of a mix of continuous and discrete design factors. Sampling represents the foundation on which all the analysis methods presented here are based. Performing an efficient sampling of the design space allows saving computational resources and thus time during the analysis. Enabling fast and accurate analyses is the key requirement for analysis methods to be used at a conceptual level.

The first analysis method that we describe is the Regression Based Sensitivity Analysis, RBSA. Sensitivity analysis is a tool that allows the engineering team to evaluate the importance of the design parameters in determining the performance of the system and resultantly to set priorities amongst them. With sensitivity analysis, cause-effect relationships can easily be discovered, thus models can be checked (by model developers) and better exploited (by users of the models that have not developed them). RBSA was developed by us to bring the benefits of sensitivity analysis at a significantly reduced computational cost when compared to other methods for sensitivity analysis.

Optimization is often regarded as a method to be used for detailed analyses, possibly later in the design cycle. In this thesis we demonstrate that (multi-objective) optimization may bring advantages also to conceptual design. This technique is able to present only the best solution(s) to the engineering team, preventing it from investing time in non-promising areas of the design space, thus saving time during the analysis. Further, the optimization techniques considered in this thesis may guarantee a thorough exploration of the design space, with mechanisms preventing local optima thus focusing on the global ones.

There might be cases in which solutions are judged equally optimal from a mathematical point of view. However, from an engineering perspective they may differ substantially. Robustness is a concept that is important in engineering, besides optimality, to assess the suitability of a certain solution. This is especially true at a conceptual stage where more sources of uncertainty are present, when compared to advanced phases of the design cycle. Uncertainty and robustness analysis techniques, alone and in conjunction with optimization methods, are considered in this thesis. Indeed, we present the Pareto-Robust Optimization Algorithm, PROA, and the Double-Repository Archive Maintenance Scheme as two approaches for dealing with uncertainties during optimization, providing robustness (together with optimality) information to the engineering team.

Several test cases are used in the discussion to demonstrate the working principles of the proposed methods. These test cases are introduced step-by-step, and the details of the mathematical models are provided in the appendix of this thesis.

The research that is the subject of this thesis culminated with the utilization of some of the analysis methods presented here in two concurrent design infrastructures from two different organizations: the Concurrent Design Facility at the European Space Agency, and JAQAR-Concurrent Design Services. These organizations gave us the opportunity to experiment using their concurrent design infrastructure. The results were very positive, demonstrating that the methods proposed in this thesis bring benefits both technically and process-wise. These experiences are described in detail in Chapter 6.

Concluding the summary of the main activities and objectives of this thesis, we would like to emphasize the fact that the proposed methods are valid in general and that they are independent from the mathematical models that one is using for the analysis. We mostly deal with space systems in this thesis, but we always treat the models from the input/output interface. It is for this reason that we can conclude that the outcome of this thesis can be applicable also to other engineering fields using mathematical models for design purposes.

A mathematical model of the system under study is a means that shall help the engineering team in taking decisions. The methods presented here are only a way of better supporting their activity. These methods are not meant to substitute the people responsible for the design process. There are many aspects of conceptual design that go beyond an efficient utilization of the available mathematical models, where the contribution of the human factor is fundamental for obtaining a final product with a high effectiveness/cost value. An interesting opportunity for further investigation of the research presented here is exploring the possibility of coupling it with methods for supporting the project managers and team leaders in directing people of various skills and social attitudes during the decision-making process. The *Delphi* method, for instance, could be one of them. We are also of the opinion that tools for gathering, keeping, and properly reusing the knowledge would bring extra benefits to the team. The final goal of such tools should be that of shifting the importance of the individual from knowledge holder to innovation pusher and knowledge aggregator.

Innovation and creative thinking is what differentiates us as people from computers. Methods and tools of the future should free the engineering team from the repetitive tasks of the design, leaving room to the creative and more fascinating aspects of the process.

Riassunto

Per la creazione della copertina della tesi sono state utilizzate alcune illustrazioni umoristiche che rappresentano un satellite (un sistema spaziale) immaginato da punti di vista differenti. Queste diverse visioni del sistema appartengono ai vari esperti che contribuiscono al suo sviluppo. É chiaro il tono ironico di tali illustrazioni. Esse sono state utilizzate per la potenza con la quale esprimono il messaggio della necessità di bilanciare requisiti e soluzioni tecniche affinché il sistema risulti essere il miglior compromesso possibile tra forze progettuali spesso in conflitto tra loro.

Siamo dell'idea che per ottenere una buona progettazione di nuovi sistemi sia necessario che tutti gli attori del ciclo di progettazione collaborino in maniera efficace. A tal proposito, lo scopo di questa tesi è di proporre e descrivere in dettaglio metodi di progettazione e analisi che mirino a creare strumenti di progettazione che possano essere di supporto al team di ingegneri impegnati nella definizione concettuale del sistema e della missione che esso dovrà compiere. Il lavoro presentato è riconducibile alle metodologie in supporto alla Ricerca Operativa.

La fase iniziale del ciclo di vita di un sistema spaziale, chiamata anche progetto preliminare o concettuale (*conceptual design* in inglese), è fondamentale affinché il sistema svolga propriamente la sua missione. Durante il progetto preliminare vengono impegnati i costi dell'intero programma fino al 70%, e la maggior parte delle decisioni prese sia a livello tecnico sia a livello manageriale influenzeranno notevolmente tutte le fasi successive della progettazione e utilizzo del sistema. Nonostante il progetto preliminare sia una fase così importante, le risorse rese disponibili per il suo svolgimento sono molto limitate rispetto a quelle impiegate per tutto il ciclo di vita del sistema. La tendenza generalizzata delle agenzie spaziali mondiali e dell'industria del settore è stata quella di richiedere che il progetto preliminare venisse svolto, paradossalmente, in più breve tempo, in maniera migliore, e inoltre ad un costo ridotto rispetto al passato. Una soluzione a questo dilemma è stata trovata grazie all'utilizzo di metodi di progettazione collaborativa (*concurrent/collaborative engineering* in inglese) in sostituzione dei precedenti metodi cosiddetti sequenziali. Un approccio collaborativo alla progettazione prevede che i vari esperti delle varie discipline lavorino parallelamente invece che in modo sequenziale. Questo consente di mantenere alto il livello di comunicazione, facilita lo scambio di informazioni tecniche e allo stesso tempo permette di prevenire di incomprensioni tra esperti e inconsistenze di progetto.

Il notevole potenziale della progettazione collaborativa è stato capitalizzato dall'industria spaziale negli ultimi vent'anni. Si pensi solo che, ad esempio, l'agenzia spaziale europea (ESA - *European Space Agency*) è stata in grado di ridurre della metà i costi legati al progetto preliminare e del 75% i suoi tempi di sviluppo. Nello specifico, questo è stato possibile utilizzando un'infrastruttura adibita alla progettazione collaborativa per le fasi preliminari del ciclo di vita di nuovi sistemi spaziali: la cosiddetta *Concurrent Design Facility*, CDF. La CDF è una struttura dotata di sistemi informatici allo stato dell'arte che consentono a un team di ingegneri di svolgere la loro attività seguendo tutti i principi della progettazione collaborativa. Dati i vantaggi dimostrati negli anni, anche altre organizzazioni governative e private hanno deciso di uniformarsi dotandosi di tali infrastrutture.

Siamo convinti che le attività di progettazione preliminare di nuovi sistemi possano trarre beneficio dall'utilizzo di metodi di analisi strutturati, sviluppati *ad hoc* oppure adeguatamente adattati per questa fase importante del ciclo di vita. É per questo che lo scopo principale della tesi è di proporre approcci per il supporto al team di ingegneria durante le fasi di progettazione preliminare. Crediamo inoltre che tali metodi, messi a disposizione del team di ingegneria, possano garantire un valore aggiunto alle infrastrutture per la progettazione collaborativa.

Uno dei presupposti su cui si basa il lavoro presentato in questa tesi è che il progetto preliminare di un nuovo sistema spaziale venga svolto utilizzando modelli matematici che rappresentano il sistema, i suoi elementi e l'ambiente operativo. Generalmente questa è un'ipotesi ragionevole, considerando che in tutti gli ambiti dell'ingegneria vengono usati modelli matematici per stimare il comportamento del sistema prima che esso sia realmente costruito. Lo scopo, infatti, è quello di aiutare nella progettazione correlando le variabili di progetto (i gradi di libertà dei progettisti) con le performance del sistema. Uno dei compiti del team di ingegneria è proprio quello di selezionare la combinazione più appropriata delle variabili di progetto in modo che il sistema si comporti come richiesto, possibilmente ad un costo minimo.

Tra i metodi proposti per supportare il team di ingegneria in questa attività c'è il campionamento. Campionare lo spazio di progetto (lo spazio matematico che ha come dimensioni i parametri di progetto) significa selezionare i punti in tale spazio (combinazioni dei livelli dei parametri di progetto) che saranno successivamente utilizzati con lo scopo di simulare il comportamento del sistema. L'approccio più comune per il campionamento, in questi casi, è il campionamento pseudo-casuale. In questa tesi vengono proposti metodi alternativi per processi di campionamento più efficienti, quando sono presenti nella stessa analisi parametri continui e parametri discreti. Il campionamento è alla base di tutti gli altri metodi presentati in questa tesi. Campionare in maniera efficiente lo spazio di progetto consente di risparmiare in termini di risorse informatiche dunque di risparmiare in termini di tempo necessario per le analisi dei dati.

Successivamente, vengono passati in rassegna metodi per l'analisi di sensitività, descrivendone uno in particolare sviluppato *ad hoc* per le fasi di progettazione preliminare: *Regression-Based Sensitivity Analysis*, RBSA. L'analisi di sensitività è un metodo che consente di determinare l'importanza dei parametri di progetto in relazione alla loro influenza sulle performance del sistema. É dunque uno strumento in grado di svelare relazioni di causa-effetto in modo tale da consentire un controllo accurato dei modelli matematici da parte dei loro sviluppatori, e di consentirne un migliore utilizzo da parte di chi lavora con essi pur non avendoli personalmente sviluppati. Il metodo da noi sviluppato mira all'ottenimento di una maggiore efficienza e velocità nell'analisi della sensitività, rispetto ad altri metodi qui discussi. Questo è in linea con l'esigenza di ridurre tempi e costi del progetto preliminare pur mantenendo gli standard di qualità richiesti.

L'ottimizzazione è spesso considerata solo in fasi di progettazione dettagliata, quindi a valle rispetto al progetto preliminare. Nel lavoro presentato in questa tesi viene dimostrato come metodi di ottimizzazione, anche multi-obiettivo, possano portare notevoli vantaggi in fase di progettazione preliminare oltre che in fasi più avanzate. Infatti, essi permettono di calcolare le soluzioni migliori nello spazio di progetto consentendo di mostrare soltanto quelle al team di ingegneria. Questo consente di evitare agli esperti partecipanti al progetto di soffermarsi su soluzioni sub ottime, velocizzando il processo di progettazione.

Soluzioni considerate ugualmente buone dal punto di vista delle performance calcolate con modelli matematici potrebbero risultare notevolmente differenti da un punto di vista ingegneristico. La robustezza è un concetto importante che contribuisce alla definizione dell'adeguatezza di una soluzione progettuale rispetto ai requisiti. Questo è particolarmente vero in fase di progettazione preliminare dove è presente un maggior numero di fattori di incertezza rispetto a fasi più avanzate. In questa tesi vengono presi in considerazione metodi per l'analisi di incertezza e l'analisi di robustezza delle soluzioni progettuali, sia come metodi a sé stanti sia in combinazione con metodi di ottimizzazione. Infatti, vengono presentati due metodi sviluppati *ad hoc* in grado di fornire informazioni di robustezza oltre che di ottimalità: *Pareto-Robust Optimization Algorithm*, PROA, e *Double-Repository Archive Maintenance Scheme*.

Per dimostrare i concetti e i metodi esposti vengono utilizzati alcuni casi applicativi descritti progressivamente nella tesi insieme ai dettagli dei modelli matematici utilizzati riassunti nei capitoli in appendice.

L'attività svolta in questi quattro anni è culminata con un'esperienza molto formativa presso due organizzazioni che hanno voluto sperimentare nelle loro infrastrutture alcuni dei metodi da noi presentati in questa tesi. Il riferimento è alla Concurrent Design Facility dell'agenzia spaziale europea, e all'azienda JAQAR-Concurrent Design Services. Entrambe queste esperienze hanno prodotto risultati positivi sia in termini di interesse generato sia in termini di vantaggi tecnici effettivamente riscontrati. I dettagli vengono riportati nel sesto capitolo.

I metodi descritti nella tesi sono di validità generale. Essi sono indipendenti dai modelli matematici utilizzati, sebbene in questa tesi ci si sia soffermati maggiormente su casi applicativi riguardanti i sistemi spaziali. Per questo motivo tali metodi possono essere utilizzati anche in altri campi dell'ingegneria, rendendo la ricerca presentata di interesse in settori differenti da quello aerospaziale.

Il modello matematico di un sistema è solo uno strumento necessario a prendere decisioni da parte del team di ingegneria e i metodi presentati in questa tesi sono solo un modo per supportare meglio tale processo. Modelli matematici e metodi di analisi non devono in alcun modo essere considerati sostituivi dell'essere umano nel processo di progettazione. Ci sono infatti molti aspetti del progetto (non solo preliminare) che esulano dall'utilizzo di modelli matematici, dove il contributo del fattore umano è fondamentale per ottenere sistemi di alto valore e costo ridotto. Un aspetto su cui la ricerca svolta nell'ambito di questa tesi potrebbe focalizzarsi, come obiettivo successivo, sarebbe quello di esplorare la possibilità di utilizzare dei metodi di supporto alla decisione per aspetti qualitativi del progetto in combinazione con i metodi quantitativi presentati qui.

Siamo dell'opinione che il fine ultimo dovrebbe essere quello di spostare l'equilibrio dei membri del team di ingegneria dall'essere proprietari della conoscenza a innovatori e aggregatori di conoscenza. La conoscenza dovrebbe essere sempre più a disposizione dell'organizzazione vista come collettività di individui che lavorano con un obiettivo comune: progettare sistemi migliori. Innovazione e creatività sono due aspetti che ci contraddistinguono nettamente dai computer (per ora). Metodi di progettazione del futuro dovrebbero permettere al team di ingegneria di essere sollevato dai compiti più ripetitivi lasciando spazio agli aspetti creativi del progetto e dell'intero processo.

Contents

Notations

AAO	All At Once
ANOVA	Analysis of Variance
BLISS	Bi-Level Integrated System Synthesis
BOL	Beginning Of Life
BPA	Basic Probability Assignment
CAD	Computer Aided Design
CCD	Central Composite Design
CD	Concurrent Design
CDF	Concurrent Design Facility
CDP^{TM}	Concurrent Design Platform ^{TM}
CDR	Critical Design Review
CSD	Complex System Design
CSSO	Concurrent Sub Space Optimization
CO	Collaborative Optimization
COBiL	Collaborative Bi-Level
COTS	Commercial Off-the-Shelf
DoE	Design of Experiments
ECSS	European Cooperation for Space Standardization
ELR	End-of-Life Review
EOL	End Of Life
ESA	European Space Agency
FAST	Fourier Amplitude Sensitivity Test
GNC	Guidance Navigation and Control
GPS	Global Positioning System
HD	Hierarchical Decomposition
IDE	Integrated Design Environment
IDF	Individual Disciplinary Feasible
IDM	Integrated Design Model
J-CDS	JAQAR - Concurrent Design Services B.V.
LHS	Latin Hypercube Sampling
MDF	Multi-Disciplinary Feasible
MDO	Multi-Disciplinary Optimization
MDR	Mission Definition Review
MIT	Massachusetts Institute of Technology
MOEA	Multi-Objective Evolutionary Algorithm
MOO	Multi-Objective Optimization
MOPSO	Multi-Objective Particle Swarm Optimization
NASA	National Aeronautics and Space Administration
ND	No Decomposition

NHD	Non-Hierarchical Decomposition
NSGA	Non-Dominated Sorting Genetic Algorithm
OA	Orthogonal Array
PDF	Probability Density Function
PDR	Preliminary Design Review
POD	Picosatellite Orbital Dispenser
PROA	Pareto Robust Optimization Algorithm
QR/AR	Qualification Review/Acceptance Review
RBSA	Regression-Based Sensitivity Analysis
RSM	Response Surface Method
SA	Sensitivity Analysis
SE	Systems Engineering
SRC	Standardized Regression Coefficient
SRR	System Requirements Review
SSP	Sub-Satellite Point
SVD	Singular Value Decomposition

Chapter 1
Introduction

The research presented in this thesis can be framed as work performed in the field of operational research. It is a field of research that encompasses approaches for facilitating decision making and design efficiency. We propose design methods and techniques to support the engineering team during the conceptual design of complex space systems.

In the last decades man-made systems have gained in overall complexity. From a technical point of view, a complex system may be defined as one in which there are complex relationships between functions and hardware, and multiple interactions between many different elements and many different disciplines concurring to its definition. Speaking in more general terms, complexity does not only regard the system *per se*, but it is also related to the whole life-cycle management of the system. This encompasses all the activities needed to support the program development from the requirements definition to the verification, validation, operation, and end-of-life of the system in the presence of a large number of different stakeholders (internal and/or external). These views of complexity from different perspectives converge to a general definition of a system as *a construct formed by a set of interdependent functions and elements (e.g., hardware, software, policies, documents, people) that complete one or more functions defined by requirements and specifications*.

The Systems Engineering (SE) process has been increasingly adopted and implemented by enterprise environments to face this increased complexity, especially in the space industry. The purpose is to ensure that the customer needs are satisfied with the required quality, promoting a reduction of costs and development time.

Systems Engineering can be defined as a discipline executing an interdisciplinary and iterative process of technical management, acquisition and supply, system design, realization, verification and validation, and technical evaluation at each level of the system, beginning at the top (*i.e.,* the system level) and propagating throughout all the elements of the system (*i.e.,* sub-system, elements, and components level). This process takes place in the form of nested iterations of analysis/synthesis through its entire life cycle.

The life cycle of a system is the set of phases into which the life of the system can be divided. It comprehends the phases that go from the conceptual design of the system to its end of operational life. The phases of the life cycle are marked by milestones, *i.e.,* typically formal meetings in which the main contractor, the sub-contractors, and the customer discuss the achieved goals and agree upon the next design phase. The Systems Engineering process, implemented at different levels of aggregation of the system and through the entire life cycle, is schematically shown in Figure 1.1.

The phases identified by the European Cooperation for Space Standardization (ECSS, 1996) are:

- Pre-Phase A/Phase 0, conceptual design
- Phase A, preliminary analysis and feasibility

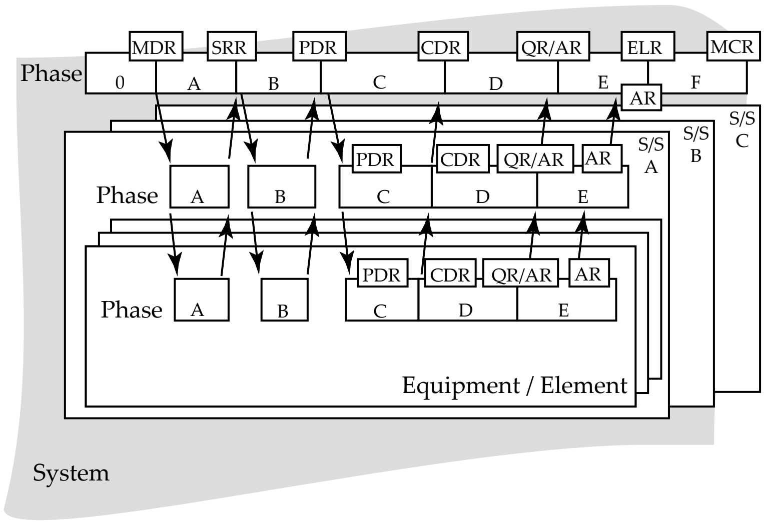


Figure 1.1 The Systems Engineering process implemented through the entire life-cycle of a space system (ECSS, 1996).

- Phase B, definition
- Phase C, detailed design
- Phase D, development and qualification
- Phase E, operations phase
- Phase F, disposal/end of life

The level of detail of the design increases enormously during the life cycle, from the conceptual phase to the production drawings. In the advanced design phases, the engineering team-members are called to perform a detailed design using their specific knowledge and sets of dedicated software. Those phases may take months to years to be fully completed, depending on the complexity of the system and the available resources. The conceptual design, instead, is usually completed on a much shorter time-scale. Weeks to months are needed/allowed for its completion. The main objective of conceptual design is the definition of the mission to perform to satisfy the customer's requirements. This is obtained by establishing multiple system-design concepts and, after their evaluation, defining the system baseline with technology, programmatic, and cost assessment. A large number of design options evaluated at this stage, will increase the chances of propagating a successful concept to subsequent design phases. Successful system concepts are those that will allow meeting all the customer's requirements, with minimum overall cost (*i.e.*, development, production, launch, and operations costs).

Conceptual design is characterized by having hard constraints in terms of costs and resources, and it is easy to understand the reason. The European Space Agency (ESA) alone, for instance, performs 10 to 20 conceptual-design studies per year, of which only 10% on average gets to subsequent phases, while the remaining are not further developed. This means that 90% of the resources invested in conceptual design end up at an impasse. The limited time available, and a potentially large number of design options to evaluate, usually limit the engineering team to a certain preliminary level of the analysis. However, experience in space-systems design demonstrated that despite the fact that most of the costs are expended in the advanced phases of the life cycle (*i.e.*, production and operations) the great majority of them is determined by the choices taken during the conceptual design, Figure 1.2.

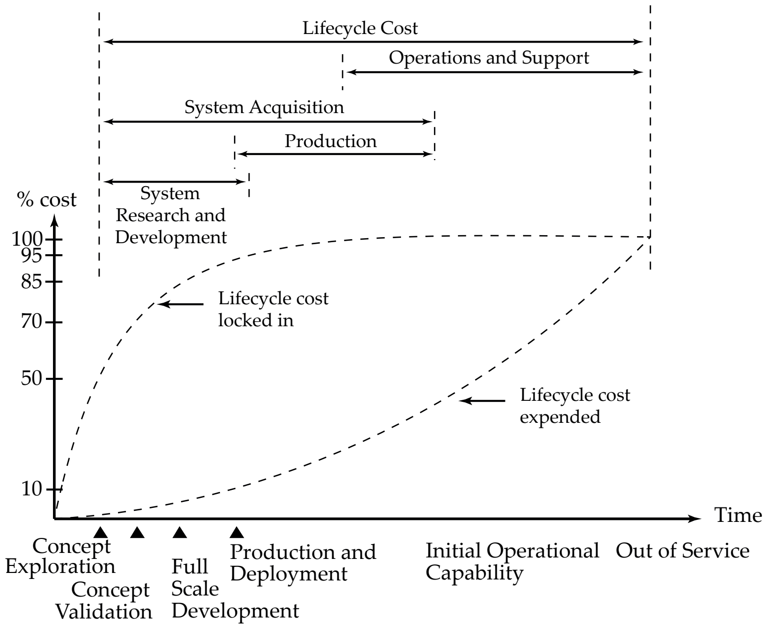


Figure 1.2 Percentage of costs locked-in and costs expended by life-cycle phase, (Larson, 1999).

Therefore, a poor conceptual design will lead to an even worse and expensive system at the end of the process. This is very clear in the space community, and a possible solution to prevent poor, or non-successful, conceptual designs of space missions was found in the implementation of Concurrent Design (CD). With the CD approach all the aspects related to the spacecraft and the mission it will perform are taken into account at the same time (concurrently) from the very beginning of the life cycle. All the technical discipline-experts, with risk-, cost- and programmatics engineers, together with the customer are in constant communication between each other, enabling the possibility to efficiently keep track of the system requirements and their evolution. In 1998, ESA made its first step towards a systematic implementation of CD during the conceptual phases of space missions and systems by creating the Concurrent Design Facility, CDF. The CDF is a design meeting room that makes use of state-of-the-art information technology to create an Integrated Design Environment where the communication between the experts is made possible and efficient. In the CDF communication happens at all the levels, also at the level of the mathematical models that the experts use for the preliminary analyses. This is done because a modification in one single discipline or subsystem immediately reflects on all the other disciplines and subsystems, creating a much higher level of awareness of the evolution of the design amongst the members of the engineering team. The experience of the CDF has radically modified the classical sequential design approach, allowing to capture more knowledge at the beginning of the process and preserve design freedom for later phases to give the possibility to fully benefit from additional knowledge gained by analysis, experimentation, and human reasoning, Figure 1.3.

The approach adopted in the ESA CDF is being replicated at industrial level in the space sector, because it demonstrated that to respond to the increasing request of complete conceptual solutions in a short period of time, and with limited resources, an Integrated Design Environment is an efficient approach.

The focus of the technical activities during conceptual design of space systems has historically been on the development and utilization of (preliminary) mathematical models able to describe the behavior of the system and its parts. Mathematical models are important to help *explaining* the system and performing trade-off studies before the system is actually built. This is true for conceptual design phases carried out with or without concurrent design and with

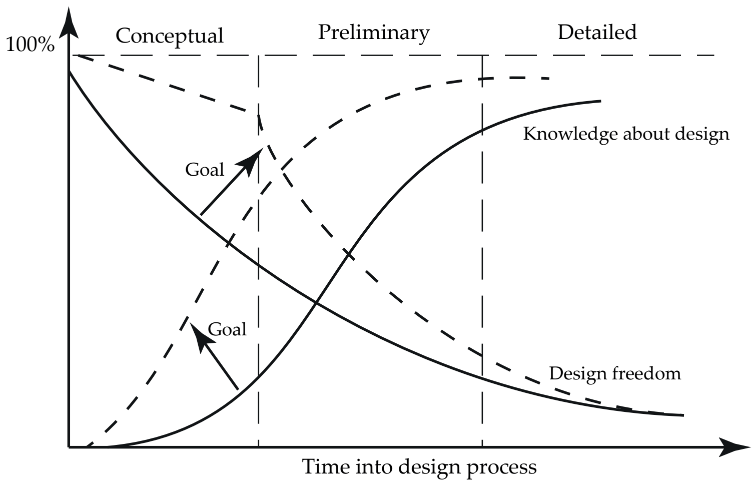


Figure 1.3 Comparison between the classical design process (continuous lines) and the target for a more efficient one (dashed lines).

or without integrated design environments. We are of the opinion that preliminary does not necessarily mean easy-to-solve, limited, poor quality, or fast-to-execute. The complete preliminary mathematical model of a space system and the mission it will perform, for instance, considering 15 to 20 different disciplines linked to each other, can become already very hard to manage. Further, when considering a mathematical model developed for an integrated design environment, thus distributed in nature, the complexity increases even more.

Despite the profuse effort on model development, we detect a lack of specific and standardized design-analysis techniques to be used by engineers, and designers in general, during conceptual design. The mathematical models need to be properly used in order to find well balanced solutions, to spot relevant phenomena in the model, driving factors, interactions amongst elements and disciplines and eventually exploiting them to improve the system performance as a whole. One-factor-at-the-time technique (trial and error) is very intuitive and largely used, but it is clearly not able to support the design activities under such increased demand for quality, and reduction of cost and development time, as discussed later in this thesis. We believe that there is a need for more advanced design and analysis techniques to be developed or specifically adapted for conceptual design. These techniques should be specific in the sense that they should allow quantitative analyses to be performed quickly, because time and resources are limited for conceptual design. We refer to design techniques such as quantitative sensitivity analysis or optimization that are usually only used for more advanced phases of the design life-cycle. Analysis methods for conceptual design should also be standardized because the output of engineering analyses of different disciplines should be comparable in the form and level of detail, to promote discussions and confrontation between discipline experts.

Currently, the mathematical models used for conceptual design (also in the collaborative environments) provide exceptional engineering-data exchange between experts, but often lack in providing structured and common design methods involving all the disciplines at the same time, leaving the type of analysis to be performed and type of results to be produced to the judgment of the team members with no integration and no standardization. This may result in the risk of incurring in misunderstandings from one side, and the risk of eventually underexploiting the available concurrent design infrastructure models from the other.

We believe that the enormous effort made to conceive, implement, and operate concurrent engineering for conceptual design can be consolidated and brought to a more fundamental level, if also specific and standardized analytical design methods and tools could be concurrently exploited during conceptual design.

For all these reasons, the principal problem definition of this thesis work can be expressed as follows:

How and to what extent can design techniques, usually implemented for advanced design phases, assist the engineering team during the conceptual design of complex systems? And in what way can these techniques contribute to obtain better, faster, and eventually cheaper design processes?

To be able to answer these questions, several design techniques and methods to efficiently use mathematical models of multi-element systems are encompassed in this thesis. The common denominator is to limit the computational effort required to obtain meaningful results to support the engineering team and the decision makers in their activities. This is required for allowing a more efficient utilization of potentially long-running models already at conceptual design level. Further, the methodological approaches described in this thesis are presented in a form that is general enough to be in principle applicable to any type of (integrated) complex system, not only space-related, using models of any level of detail, thus potentially also for more advanced phases of the design process.

1.1 Analysis methods for engineering-team support

The activity of designing a system is related to the possibility of predicting its performance and characteristics before the system is actually produced and operated. This means that a (mathematical) model of the system shall allow to simulate its behavior, given the values of the design factors as inputs. The design factors represent the degrees of freedom of the engineering team, that adjusts them in such a way as to obtain the desired performance. During the design activity, at any level of detail, thus also during conceptual design, typical design questions arise. It is answering these questions that the design actually takes place and the system is shaped and refined. To answer the main research questions presented in the previous section, we will try to give an answer to these typical design questions that will help the engineering team to obtain a faster, better and possibly cheaper design, specifically at conceptual-design level.

Amongst all the design factors of the system model, what are those actually influencing the performance of interest? To what extent do these factors influence the performance?

The determination of the most influential factors is obtained performing sensitivity analysis. The sensitivity analysis is usually implemented for screening the input factors and determine those that influence the variability of a certain phenomenon of interest most (Saltelli *et al.*, 2004). This is accomplished in most of the cases by using a Monte-Carlo based approach. In this thesis an alternative, and possibly more efficient, method to compute the sensitivity analysis is proposed and discussed. The method is based on a particular implementation of factorial design for sampling the design space and computing the sensitivity indices, using a variance-based parametric approach. A great advantage of using this approach is that the number of model evaluations is radically reduced when compared to the computational effort required by the Monte-Carlo based techniques, thus enabling the sensitivity analysis to be used even when the models require a long time to execute for each single evaluation. Sensitivity analysis is also a powerful tool to be used to check and validate the mathematical model by comparing the output to the designers expectations.

In case of uncertainties in the factors influencing the performance of the system, how do they propagate through the model? And what are the factors that are mostly responsible for performance uncertainty?

When designing a complex system, there can be at least two types of uncertainty. Epistemic uncertainty (*i.e.,* systematic uncertainty arising when there is not enough information to determine a certain quantity, not even to estimate a probability distribution) is one type of uncertainty that the engineers need to deal with. Stochastic uncertainty (*i.e.,* intrinsic uncertainty of the system design due to a non-controllable factor, *e.g.,* environmental) shall also be taken into account. Epistemic uncertainty is related to gaps of knowledge, *e.g.,* the utilization of a new technology never used before may lead to epistemic uncertainty in its behavior. Stochastic uncertainty, instead, can be associated with phenomena like flipping a coin. In performing subsequent coin-flipping, one is uncertain about every single outcome, but there are mathematical ways for estimating the long term confidence in obtaining one specific side of the coin.

In this thesis the parametric-design method used for sensitivity analysis in case of controllable design variables is extended to be able to deal with stochastic uncertainty as well. The uncertainty from input to output is efficiently propagated to determine the Probability Density Function (PDF) of the performance, given the PDFs of the factors of interest. One of the applications on which we use uncertainty propagation in this thesis is to determine design margins and system budgets.

What is the shape of the design space? And what are the best parameter settings to optimize the objectives and meeting the constraints?

The information gathered during the sensitivity/uncertainty analysis can be interpreted as a roadmap for the engineering team to efficiently direct the design effort. The non-influential design factors can be fixed to a pre-determined level, because they will not affect the performance much, *de facto* reducing the dimensions of the design search-space. However, the influential design variables and the behavior of the system under the effects caused by their variation and their interactions shall be investigated in more detail. The results of the simulations used for sensitivity and uncertainty analysis are also used to compute response surfaces linking the most influential design factors to the performance. This provides the engineering team with a clear insight in the shape of the design regions of interest. The response surfaces are presented in the form of contour plots, in which also constraint violation regions are superimposed. This compact visualization of the design space represents an easy and direct way to understand the effect of a change of the values of the design variables.

How robust is (are) the baseline(s)?

Robustness can have different meanings, depending on the context to which it is applied. One may think of robustness as that characteristic of the system for which its behavior does not change much given off-design settings of the environmental factors, that are not directly controllable at design level. However, robustness of a system may be also assessed from the SE process perspective. The design obtained at the end of the conceptual design phase (*i.e.,* what is typically called the baseline) may still be modified, partially at least, in subsequent phases of the design process. In this case one is interested in understanding the robustness of the design baseline, in terms of performance, given the modification of the controllable design factors. The sampling methods and the uncertainty propagation techniques presented in this thesis, will be used to demonstrate that they are flexible enough to support decision makers in both types of robustness analysis.

What are the settings of the variables for which the performance(s) is (are) optimized?

Optimization techniques are generally used during detailed design, at discipline level, to determine the maxima and/or minima of the problem of interest. The problem of designing and optimizing a space system, considering its operative environment and the mission it will accomplish, is highly constrained and characterized by having multiple objectives, with continuous and discrete (*e.g.*, architectural) variables. Many techniques have been developed that could in principle be used to solve such problems, providing solutions in the form of Pareto fronts. In this thesis we discuss on the applicability of some known techniques for multi-objective optimization to the optimization of system models during conceptual design. The Pareto front demonstrates to be very effective in narrowing down the options to show to the engineering team. Indeed, only those solutions that are considered optimal are present on the Pareto front.

What are the settings of the variables for which we obtain performance(s) that is (are) both optimal and robust?

It is empirically proven that excellent results can be obtained using optimization techniques on relatively complex mathematical models. However, optimal solutions are not all equal to each other, especially from an engineering perspective. Some optima could be not robust. This means that they could be the result of a particular combination of design variables that will exhibit a steep drop in performance when the values of these variables are only slightly modified. Especially during conceptual design of space systems, the design variables are only frozen after several design iterations. Thus, there is a risk that the selected design baseline may suffer from performance degradation in subsequent phases of the design cycle. In this sense, a more robust solution can be considered a less risky one. In this thesis we demonstrate that by integrating the methods developed for studying sensitivity analysis with any optimization algorithm it is possible to support the engineering team by generating robust-optimal solutions.

1.2 Thesis layout

The design methodologies presented in this thesis are developed to support the engineering team *and* the decision-makers during the conceptual design of complex systems with potentially long-running and distributed mathematical models. In this thesis we describe the advantages and limitations of using them, providing several design test cases that demonstrate their feasibility and potential.

1.2.1 Chapter 2

In Chapter 2 we provide some basic definitions, terminology, and design settings of the class of problems of interest that are used in the thesis. The different approaches that may be used to model a system made of multiple elements is also discussed. Modeling is not the main focus of the thesis, but mathematical models of space and non-space systems will be used to demonstrate the design techniques that are presented. At the end of Chapter 2 we provide some preliminary information on the problems used as test cases. The choice of the mathematical models for the problems presented here, and the level of detail considered, is related to the possibility of demonstrating the working principle of the analysis methods, but also to be representative of a hypothetical conceptual design phase. All the assumptions are provided in the appendix sections of this thesis.

1.2.2 Chapter 3

Sampling the design space is the first design activity discussed in Chapter 3. Sampling is the cornerstone for a successful, accurate, and computationally cheap analysis using mathematical models. All the sampling techniques presented in this thesis are based on innovative combinations of existing techniques that will be discussed case by case. In Chapter 3 we also show the advantages of having sensitivity analysis as a tool to predict the importance of the design factors in the determination of the performance of interest. This is a fundamental analysis technique for a decision maker and it will be tackled using the Regression-Based global Sensitivity Analysis method (RBSA). RBSA is an innovative approach that we developed to obtain quantitative, variance-based, sensitivity indices of the design factors of a mathematical model. It provides very accurate results with a significant reduction of the number of required model evaluations, compared to other methods.

Uncertainty is always an ingredient of the design of engineering systems, especially at a conceptual level. In this chapter we also show that specific sampling techniques can promote uncertainty and robustness analysis, meant as propagation of input uncertainty into the model or also as methods to assess the effect of modeling uncertainties on the performance. The uncertainties considered in this thesis work are all assumed to be uncorrelated. We do not take input correlation structures into account, however we provide some references to link this work with methods for sampling considering correlation of the inputs. On the other hand, we do not impose any constraints in terms of correlation to the output of the mathematical models. The output is solely determined by the mathematical relationships between inputs.

1.2.3 Chapter 4

In Chapter 4 we focus our attention on methods that allow the engineering team to efficiently and systematically explore many design options, rather than studying only few of them in detail, as it is possible with the analysis methods presented in Chapter 3. In particular, here we focus on global multi-objective constrained optimization that provides at the end a set of optimal solutions known as the Pareto front. The main objective is to use optimization techniques to facilitate the design process, we are not interested in developing them. However, coupling of global methods and local ones will be deeply explored. In this chapter, indeed, we introduce the Pareto-Robust Optimization Approach (PROA), a concept that was developed in the scope of this thesis. It is based on some of the local analysis techniques described in Chapter 3 to be implemented in the design region in the neighborhood of the Pareto-optimal solutions. PROA allows for estimating a metric for the Pareto-Robustness, to allow the engineering team to strive for optimal-robust solutions, and contributes to improve the quality of the final set of Pareto solutions.

1.2.4 Chapter 5

In Chapter 5 we couple the local and global approaches described in Chapters 3 and 4 right from the beginning of the optimization process. The proposed approach is called robust optimization or reliability optimization, depending on the meaning that one gives to the uncertain design variables that are involved. In robust multi-objective optimization it is common practice to optimize the average performance instead of the nominal objective functions. To compute the average performance, and to determine the compliance of the solutions to the constraints, sampling is needed in a neighborhood of each individual, and the performance of each sample point must be evaluated. This drives the computational cost of robust optimization up. In this chapter, we present a repository-based approach that limits the number of evaluations needed during robust optimization, instead. Sampling methods only will be used for the propagation of uncertainty through the mathematical models, and the rationale will be clear on a case-by-case basis.

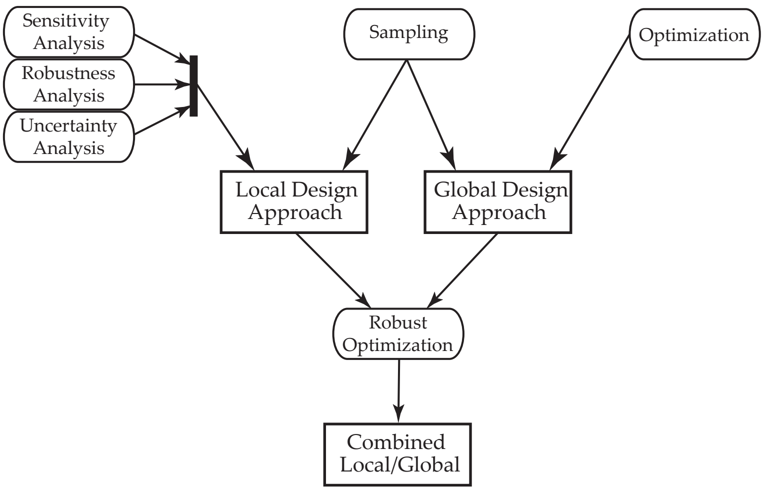


Figure 1.4 Key elements of the thesis study.

1.2.5 Chapter 6

In Chapter 6 we present the implementation of some of the methods discussed in this thesis in real integrated design environments. In particular, uncertainty analysis was used in the ESA CDF to study the effect of the uncertainties on the mass budget of the first cubesat mission ever designed in the CDF. Further, we also provided support to the concurrent design process implemented by JAQAR-Concurrent Design Services (J-CDS) for both space and non-space industry. We used sensitivity analysis, regression analysis, sampling techniques for continuous and discrete design spaces, and uncertainty analysis to support several design activities. We supported system-level decisions for the reduction of the price of a new product to be produced by a non-space organization. Further, these design methods were also used at discipline-domain level to support design activities in the absence of input data from other disciplines and to negotiate on requirements.

1.2.6 Key elements and limitations

In Figure 1.4 the key elements of the thesis and their interactions have been identified. The construction with which they are linked together was briefly introduced before. Some limitations apply to each of the elements shown in Figure 1.4. These limitations are listed hereafter.

Sampling

- No random sampling (Monte-Carlo). Despite the fact that all the analysis methods discussed in this thesis can be coupled with random sampling, we use alternative (possibly more efficient) sampling techniques, either from literature or developed by us.
- Correlation between input factors is not taken into account. All the input factors are considered uncorrelated.

Sensitivity analysis

- A limited number of sensitivity analysis methods is considered for comparison with Regression Based Sensitivity Analysis, developed by us. We take only the most representative of the sensitivity analysis methods that provide variance-based results into account.

Robustness and uncertainty analysis

- Only sampling-based approaches have been considered for the propagation of the uncertainty (no analytical methods).

Optimization

- Local optimization methods are not considered.
- Only heuristic methods for global optimization are taken into account.

Mathematical models

- Use of available models from the literature.

Chapter 2
Design of Complex Systems

One very important aspect for the successful conceptual design of a system is the availability of a mathematical model that represents the main phenomena of interest. The system under analysis may be formed by many subsystems, and many disciplines concur to the determination of its performance and its engineering characteristics. Many approaches may be used to actually implement such a mathematical model, but some of them are more suitable than others for using the model concurrently, for conceptual design. In this chapter some of the most common modeling techniques are described, highlighting pros and cons in Section 2.2. But first, in Section 2.1 we describe the basic terminology regarding the design of a system using its mathematical representation. This chapter is concluded with Section 2.3 where we describe the test cases used in subsequent chapters to demonstrate the working principles and the applicability of the design methods proposed in this thesis.

2.1 Definitions

The discussion and the methodologies presented in this thesis are based on the assumption that the activity of designing a complex system is performed by a team of designers (the engineering team), using **mathematical models** to determine the physical and functional characteristics of the system itself. A mathematical model is a set of relationships, *i.e.*, equations, providing figures-of-merit on the **performance(s)** of the system as output to the engineering team when certain **inputs** are provided. The inputs are represented by the **design variables**, *i.e.*, factors that are responsible for influencing the performance(s) of the system. For this motivation, the design variables will also be called **design factors**, or more generally inputs, or simply variables. The domain of existence of the design variables forms the **design space**, where they can assume certain **values**, or **levels**, between a minimum and a maximum. The design variables are also called **controllable** factors, since their value can be set at design time by the engineering team. The **design-variable range** determined by the minimum and the maximum can, of course, only be as large as the domain of existence of the variable. Mimima and maxima for the design variables are usually set by the engineering team to limit the analysis to a specific region of the design space or to avoid infeasible conditions. For instance, the design range of the eccentricity e of a closed orbit about the Earth should not exceed the interval $0 \leq e < 1$. In the upper-left diagram of Figure 2.1 a hypothetical design space formed by two variables, *i.e.*, eccentricity and semi-major axis, is shown. The limits of the individual variable ranges are represented by the dash-dotted lines. The subspace of the design space determined by all the design-variable ranges is addressed as the **design region** of interest, and it is represented in Figure 2.1 as the area between the two vertical and the two horizontal dash-dotted lines in the top-left diagram. These dash-dotted lines also represent the **boundary conditions** Design variables can be **continuous** or **discrete**. A continuous variable can assume

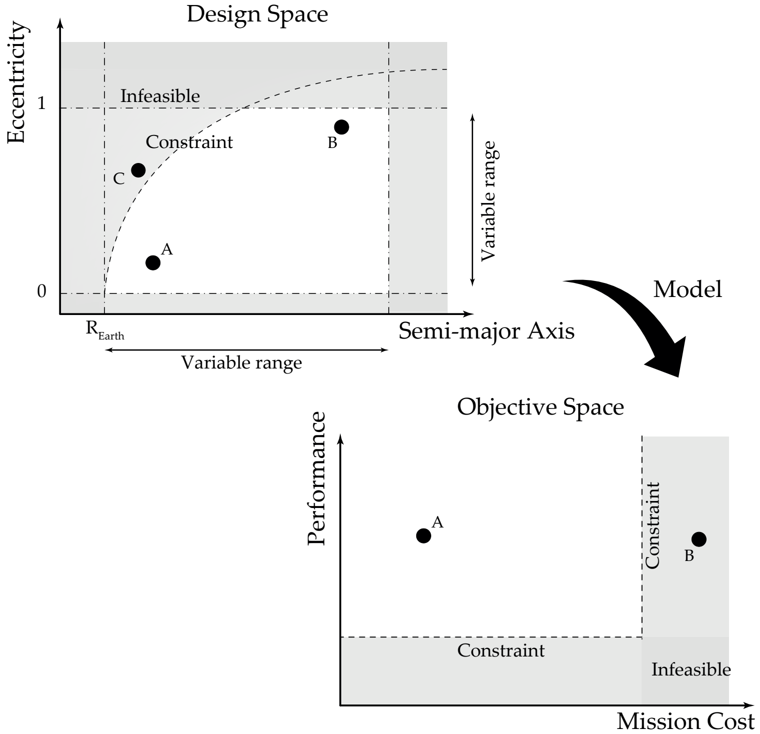


Figure 2.1 Schematic representation of the design space and the objective space of the mathematical model of a system. The gray area is infeasible.

all the values between a minimum and a maximum. A discrete variable, instead, can assume only a few specific values in the design-variable range. Discrete variables can be further distinguished into two classes, namely **ordinal** or **categorical**. The *length* of a solar array on a satellite system, for instance, is a continuous variable. It can assume, in principle, any value between a minimum and a maximum set to limit the mass or to provide a minimum performance under certain conditions. The *number of cells* used to build the array is an ordinal variable. It can only assume the levels represented by the natural numbers, and certain characteristics increase (decrease) when the number of cells increases (decreases), *e.g.,* the total mass. The *type of solar cell*, instead, is a categorical variable. This means that it can only assume certain levels (*e.g., type#1*, *type#2*, and so on), but in this case the order is not important. It is not always the case that, for instance, the *efficiency* of the solar cells increases going from the first type to the second type and so on. It depends on the order in which they appear in a database, for instance, that may be an arbitrary choice of the engineering team. The model of the system may also be subject to other sources of variability representing the non-deterministically known parameters typical of the operating environment of the system. The residual atmospheric density in orbit, the solar radiation, and the orbit injection errors, just to mention a few, are factors that may not be directly controlled, therefore they must be taken into account in a statistical sense. These factors are called **uncontrollable**.

One of the main tasks of the engineering team during the **design process** of the system is to set the values and/or the levels of the design variables in such a way that the performance(s) of the system assume(s) a certain optimal level under *certain conditions* (**optimal design**), and/or such that the final system is insensitive (within a certain extent) to variations of the (un)controllable factors (**robust design**). The performance(s) of interest is (are) called **objective(s)** of the analysis. The space in which the objectives can be represented, *i.e.,* the domain of the images of the mathematical equations of the model, is called **objective space**. Thus, the model is responsible for relating points in the design space with points in the objective space. The term *certain conditions* is used to indicate the constraints and boundary conditions of the analysis. As already mentioned, the **boundary conditions** are represented by the design-variable ranges, the dash-dotted lines of Figure 2.1. The **constraints**, instead, are determined by an infeasible condition in the objective space, *e.g.,* the mass of the satellite is exceeding the mass that the launcher is able to deliver in a given orbit. Further, the constraints can also be determined by infeasible conditions in the design space, when certain combinations of the values or levels of the design variables are not allowed. This may happen, for instance, with the eccentricity and the semimajor-axis of an Earth-orbiting satellite: their combination must ensure that the perigee radius of the orbit is at least larger than the radius of the Earth. Constraints may be linear or non-linear, continuous or discrete. The dashed lines in Figure 2.1 represent the constraints in the design space (non-linear in this case), and in the objective space (linear in this case). The thick dots A,B,C in Figure 2.1 represent **design points**. In the design space, they are a representation of the values of the design variables, while in the objective space dots D and E represent the corresponding set of output values. Considering a deterministic model, there is a one-to-one correspondence between one point in the design space and one point in the objective space. However, the engineering team must make sure to provide design points that do not violate constraints in the design space. For instance, an orbit with a semi-major axis of 7000 km and an eccentricity of 0.7 would lead to a negative value of the satellite altitude at perigee (*i.e.,* non-existing orbit) thus with the impossibility of computing relevant parameters such as, for instance, *time-in-view* at perigee passage over a specific region on Earth. Therefore, in Figure 2.1 the design point C does not have a corresponding image on the objective space.

The methodologies presented in this thesis, are designed to help the engineering team and the **decision makers** in the activity of *exploring* the design space of complex-system models. **Design-space exploration** is the fundamental activity with which the model of the system is sampled to understand the effect of the design choices on the performance(s) and to set the values of the variables in such a way that the final product will perform as required by the customer. This activity often involves many stakeholders, with many objectives to be balanced, many constraints and many design variables, thus posing the problem to be extremely difficult to solve *by hand*. Thus the scope of the thesis is to provide a guideline for exploring the design space of models of different complexity in an automated and efficient way.

2.2 Characteristics and modeling peculiarities of a complex system

Dr. Sobieszczanski-Sobieski, in one of his lectures at the Massachusetts Institute of Technology (MIT) once said:

. . . if you cannot model it, you cannot optimize it . . .

This expression could not be more agreeable. Indeed, a mathematical model (being very preliminary or quite detailed, depending on the type of analysis to be performed) is fundamental to understand in advance, before the system is built and operated, the behavior of the system, *i.e.,* the result of the decisions taken during the design on its performance (cause $\rightarrow$ effect). The problem of obtaining the mathematical model of a complex system may be treated considering two main sub-problems, namely *problem decomposition* and *problem formulation*, (Sobieszczanski-Sobieski, 1989b; Cramer *et al.*, 1993; Tedford and Martins, 2006).

The decision whether to *decompose* the mathematical model or treat the problem using a monolithic mathematical model and what kind of *formulation* to adopt depends on the complexity of the problem and models involved and the number of people working with them. Before entering into the details of the most-widely adopted approaches to manage the model of a complex system, a comparison between the Multi-Disciplinary Optimization (MDO) and the Complex System Design (CSD) problems will be made. The reason is that in literature

	MDO Problem	CSD Problem
Multiple Disciplines	Multiple disciplines are applied to the design of a single element/system.	Every element of the complex system may or may not require more than one single discipline to be designed.
Multiple Elements	Usually not more than one.	It is the core of the CSD problem. Several elements have to be designed concurrently.
Mathematical Models	Level of detail depends on the objectives of the analysis.	
Continuous Variables	Most of the MDO problems involve continuous variables.	In use mostly for elements/components analysis.
	Gradient-based sensitivity-analysis not applicable with discontinuities.	
Discrete Variables	Usually very few, ad-hoc optimization algorithms.	Used for system analysis. Especially when different *architectures* have to be judged (ordinal and/or categorical variables).
Design and Optimization	Usually numerical techniques are applied, gradient-based or stochastic. The objective is to obtain the best possible solution: *push-and-go* techniques. Human intellect can be out-of-the-loop.	*Push-and-go* not applicable. The objective is to enable trade-offs: the human intellect is in-the-loop.

Table 2.1 Comparison of the characteristics of the MDO and the CSD problem.

many applications and design techniques have been developed to deal with MDO (almost nothing is available for CSD), and even if CSD and MDO are conceptually different problems, many common aspects exist that allow for the re-utilization of part of the knowledge developed for the MDO problem also in the CSD one.

2.2.1 Complex system design vs. multi-disciplinary optimization

The field of MDO encompasses the activities of efficiently analyzing and optimizing a design problem governed by multiple coupled disciplines (Ridolfi *et al.*, 2010). A classical example of an MDO problem is the aero-elastic problem encountered when dealing with flexible wing design, where mainly structural and aerodynamics calculations are executed to obtain solutions that optimize the overall design of a wing (De Baets *et al.*, 2004). Several disciplines, in this case only two, are applied concurrently to a single system/element, in this case the wing of a non-conventional aircraft. The CSD problem is conceptually different, by definition. The main concern is to concurrently design different elements of a system, whose design procedures can, as it happens in most of the cases, or cannot involve several disciplines. The main differences and common aspects of the two problems can be found in Table 2.1.

The distinction between the MDO and the CSD problems given in Table 2.1 is the result of two different definitions. Some readers may consider it not strong enough to actually make a distinction between them, some may agree, some other readers, instead, may consider MDO as encompassing both problems considered above. However, in this thesis the aspect of supporting the *human in the loop* during the design of systems made of multiple elements, that may require more than one discipline to be designed, is the fulcrum of the discussion. Therefore, the rationale behind Table 2.1 shall be considered as a preliminary definition of the problem of interest. The identification of some commonalities between the problems also serves as a jus-

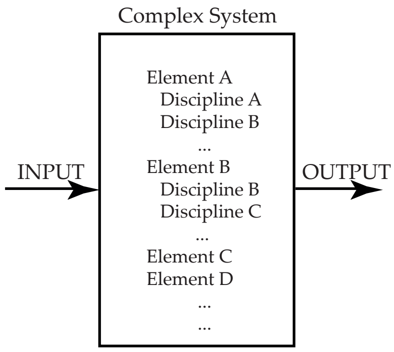


Figure 2.2 Schematic of the No-Decomposition approach for complex system models.

tification for inheriting the terminology and common practices in terms of *modeling techniques* from MDO to CSD. These are discussed in the following subsections.

2.2.2 Problem decomposition

No-decomposition approach, ND

One of the first and probably most straightforward ideas to decompose the model of a complex system has been not to decompose it at all. A single design model is implemented, which takes inputs and constraints providing outputs to the user(s). All the governing equations of the elements and disciplines involved are kept together in this monolithic model, see Figure 2.2. The ND approach has the advantage of potential direct linkages of the monolithic model with an optimization algorithm, which may use the model as if it were an (a set of) objective function(s) to obtain the feasibility/optimality of the design. Of course, this kind of approach becomes harder to manage and to execute as the number of design variables, disciplines and couplings between them, increase. Further, the resulting methodology would not be so flexible, since the concept of modularity is not used at all.

Hierarchical decomposition approach, HD

For complex systems, the decomposition of the mathematical model into smaller sets is highly advised. It allows to efficiently manage the complexity, providing ease of maintainability of the code, modularity and scalability. In the literature, authors propose several model-decomposition techniques. However, two main classes may be identified, namely *Hierarchical Decomposition* and *Non-Hierarchical Decomposition* methods (Sobieszczanski-Sobieski and Haftka, 1995; Alexandrov and Hussaini, 1995). The Hierarchical Decomposition methods (HD) treat the system model by dividing it into several independent sub-models. Each one of those sub-models has its own local variables while the global parameters are specified at the system level. The element models in which the system model is divided are independent from the local parameters of other element models. There may also be a weak dependency between governing equations of one element and local variables of another. In those cases, the coupling is neglected and the elements/disciplines are considered uncoupled. The schematic in Figure 2.3 better explains this approach. The HD method is certainly more flexible than an ND method since it fully exploits the concept of modularity, giving the user the possibility to separately treat the models of the several elements/disciplines involved. The disadvantage of using HD methods is that the coupling between the *blocks*, or the elements, only involves global parameters.

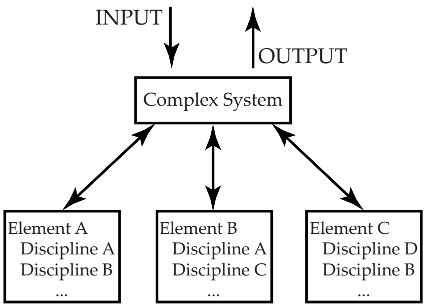


Figure 2.3 Schematic of the Hierarchical Decomposition approach for complex systems models.

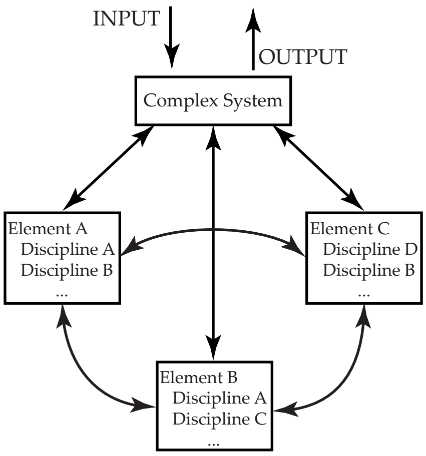


Figure 2.4 Schematic of the Non-Hierarchical Decomposition approach for complex systems models.

Non-Hierarchical decomposition approach, NHD

Non-Hierarchical Decomposition methods (NHD) shall be used when there is no clear separation between two or more elements/disciplines, *i.e.,* when the coupling between them is not negligible *a priori*. In those cases the information flow is much more complex when compared to the HD method, since it has to go in vertical directions (from element level to system level and *vice versa*), but also in lateral directions (from one element to another), see Figure 2.4. This causes an increase of the number of variables since with the NHD approach also the so-called coupling variables between elements, and the consequent coupling equality constraints, must be taken into account. For instance, the variable *Solar Array Area*, for the design of an Earth-oriented satellite system, is used to compute the available power input, but also the drag exerted on the satellite by the residual atmosphere. The power subsystem model and the drag model of the satellite system *share* this variable. Its value must be equal in both mathematical models at the end of the design iterations. The *propellant mass* of the AOCS (Attitude and Orbit Control System) is used to compute the structural mass of the subsystem and the volume to be stored in the tanks. The propulsion subsystem and the structure must work with the same value of this variable. In these two examples the *Solar Array Area* and the *propellant mass* are coupling variables. In other words, coupling variables are *virtual* duplications of existing variables (and coupling equality constraints are *virtual* constraints) that are used to maintain a net separation of the system mathematical model into element mathematical models, and yet not eliminating the coupling between them.

Problem decomposition, conclusions

The choice between an ND method and one of HD or NHD certainly depends on the complexity of the problem, but also on some other relevant issues. An ND method will not allow efficient code re-utilization, modularity, maintainability, and scalability. An enterprise environment, paying in complexity of the overall architecture, may most likely require those characteristics. On the other hand, an ND method is much easier to implement and to use for designing, because it can be used as a *black box* on individual machines. The choice between HD and NHD methods strongly depends on the particular class of problems the designers have to deal with. It depends, for instance, on the amount of coupling that exists between the involved elements. In general, the mathematical models of the elements of a complex system are developed by different individuals and subsequently linked together through an input/output software architecture. Therefore independence between the elements cannot be assessed *a priori* (actually one of the main goals of designing using all the element models concurrently is to understand and exploit the interactions), thus leaving the natural choice of steering for an NHD method. The NHD approach is flexible enough to allow for a *plug-&-play* management of the mathematical models of the elements (it enables scalability and code re-utilization), and it naturally behaves as a hierarchically decomposed model in the case of completely uncoupled elements. As discussed later in this chapter, the mathematical models of a collaborative environment are usually developed using an NHD approach. For these motivations, the discussion from this point is meant to be applicable for non-hierarchically decomposed systems.

2.2.3 Problem formulation

The formulation of the CSD problem is related to the allocation of the resources to the various elements of the architecture. The nature of the design process of a complex system is iterative, at least partially. The formulation of the CSD problem influences the convergence level reached after each design iteration. The goal is to obtain a model that provides a consistent output, possibly a feasible solution, when an input is provided. In the following subsections three main typologies of problem formulation will be discussed, namely Multi-Disciplinary Feasible (MDF), Individual Disciplinary Feasible (IDF), and All-at-once (AAO) (Cramer *et al.*, 1993; Balling and Sobieszczanski-Sobieski, 1996; Tedford and Martins, 2006).

Multi-Disciplinary Feasible, MDF

The Multi-Disciplinary Feasible problem formulation foresees that for each input a *converged* solution must be obtained for the design of all the elements of the system and of the complete system. Convergence is obtained by iterating the element analyses until all the coupling variables between the elements converge, *i.e.,* do not change significantly over successive iterations. This also means that the coupling equality constraints are satisfied. The advantage of using this method is that for every design-variable set provided as input a consistent output is obtained. One of the drawbacks is that design iterations (that are resource-consuming) must be performed even when the final solution is far from being the optimal or the desired one. With this approach, the analysis is managed at a system level only, see Figure 2.5, working with the design variables and leaving the determination of the values of the coupling variables to the models themselves, during the iterations to convergence. The relationship between the mathematical models in the MDF formulation and the designer(s) is better explained in Figure 2.5. The dashed circles of Figure 2.5 represent the iterations to be executed on the coupling variables (the largest circle) and on the design variables (the smaller circles) for every input provided by the user(s) to obtain a consistent (converged) solution.

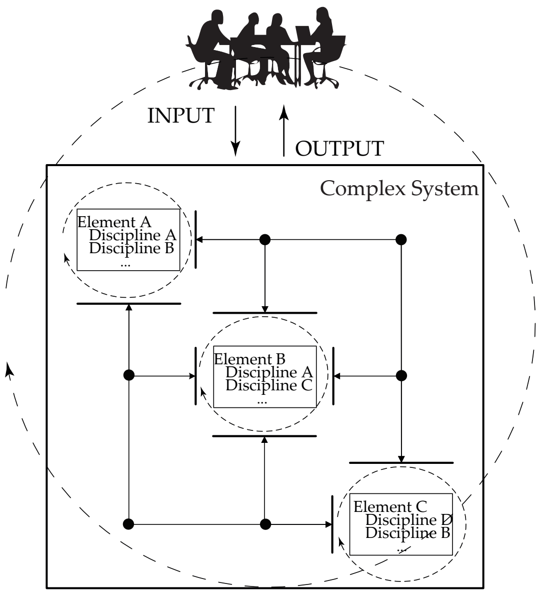


Figure 2.5 Schematic of the Multi-Disciplinary Feasible formulation for complex systems models.

Individual Disciplinary Feasible, IDF

In a problem stated in terms of an Individual Disciplinary Feasible formulation, only the individual elements convergence is enforced for every design variable set provided as input. This means that every element will iterate on the design variables according to the underlying equations and to the inputs provided (Tedford and Martins, 2006). In this way, the overall convergence of the system, *i.e.,* the convergence of the coupling-variable values, needs to be guaranteed in a subsequent phase. This can be done either manually by the user(s) or by using an optimizer. Compared to the MDF formulation, the IDF is considered to be more efficient from the computational point of view (Hulme and Bloebaum, 2000), and more flexible due to the possibility of executing the mathematical models of each element separately, eventually in parallel, because they are virtually decoupled. The most evident drawback is that for a given input, a consistent solution at system level is not guaranteed. The IDF approach is explained in Figure 2.6. The feedback arrow that goes from the elements to the user(s) represents the additional analysis effort that is needed to solve the coupling equality constraints. As in Figure 2.5, the dashed circles represent the iterations to be executed on the design variables (the smaller circles) for every input provided by the user(s).

All-At-Once, AAO

In the All-At-Once problem formulation, also the individual elements convergence is not enforced at each design iteration, as in the IDF approach. All the design and coupling variables are managed at system level either manually by the user(s) or by an optimizer, see Figure 2.7. Therefore, the actual number of variables used at system level is much larger than in the previous two approaches, with more constraints to be satisfied. This implies that the solution of the problem posed in these terms is more complex and harder to achieve. Further, convergence is not guaranteed until the termination of the design/optimization process. Cramer *et al.* (1993) and Tedford and Martins (2006) demonstrate that the AAO problem formulation is the most effective in terms of computational effort. This can be explained considering the fact that the convergence is guaranteed only at the end of the process, thus avoiding many iterations for every input from the user(s). However, one drawback of such formulation approach is that the

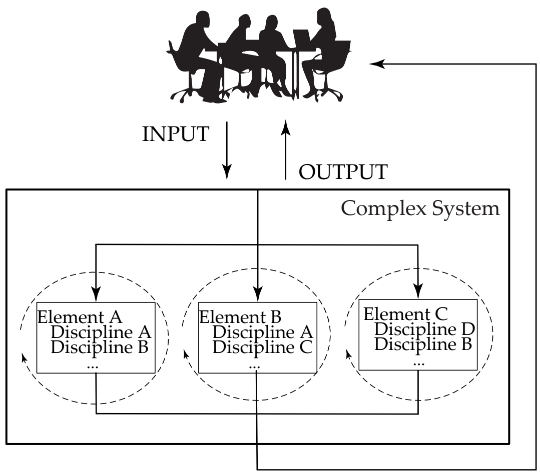


Figure 2.6 Schematic of the Individual Disciplinary Feasible formulation for complex systems models.

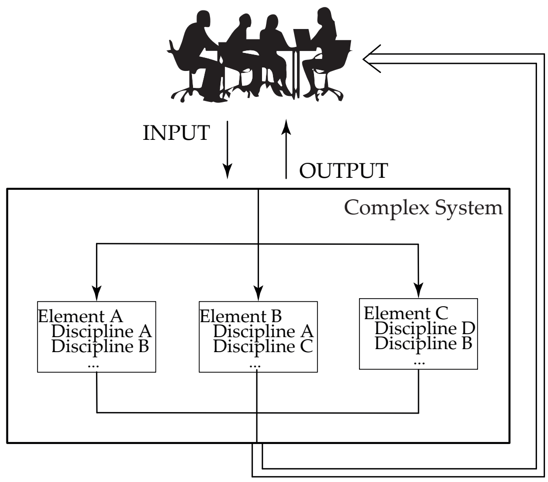


Figure 2.7 Schematic of the All At Once formulation for complex systems models.

analysis becomes much more complex. This increased complexity is also represented in Figure 2.7 where the double feedback line indicates the additional analysis effort that is needed to solve the coupling equality constraints and the convergence of the variables of all the elements of the system.

Multi-level formulations

The formulation methods presented up to this point can be classified as *single-level* methods. This means that in all cases the architecture is conceived so that the analysis, or the optimization, of the system has to be performed in one single location, *i.e.*, at the model/analyzer or model/optimizer interface. This usually happens at the top of the architecture, where the *team icon* is located in Figures 2.5 to 2.7, considering the underlying model as a *black box*. Single-level architectures can be executed in one single location by a complete engineering team, or eventually by a single designer, having full authority on the entire model. This can be useful when fast and preliminary/preparatory analyses have to be performed on the model. However, collaborative environments have the characteristics of being distributed environments. This means that the team experts can participate to the design of the system from different

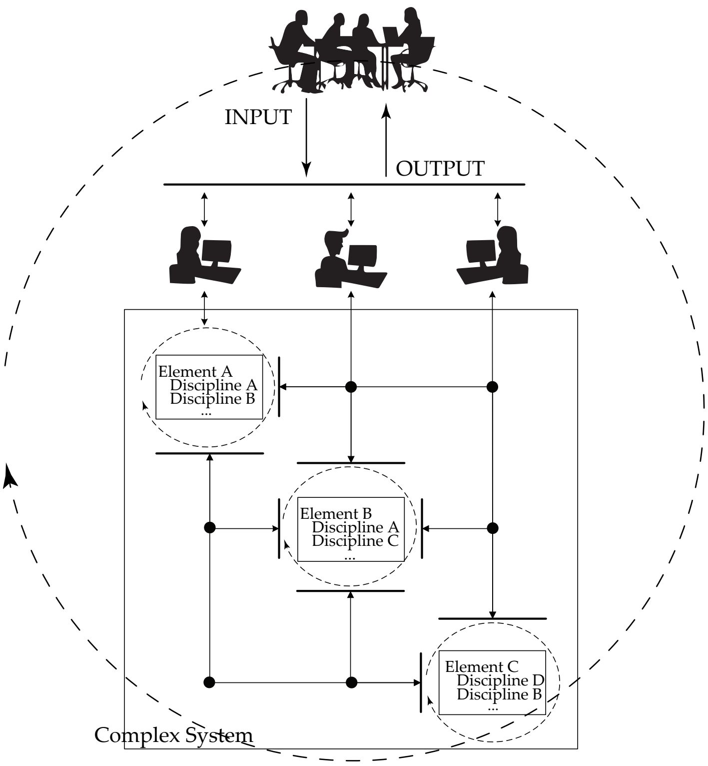


Figure 2.8 Schematic of the Collaborative Bi-Level (COBiL) formulation for complex systems models.

locations. Multiple-level formulations can be found in the literature. Yi, Shin, and Park (2008) provide an extensive bibliography on single and multiple level MDO-problem formulations and a comparison of their performances with mathematical examples. The multiple-level formulations taken into account are the following:

- CSSO, Concurrent Sub Space Optimization, proposed by Sobieszczanski-Sobieski (1989a).
- BLISS, Bi-Level Integrated System Synthesis, proposed by Sobieszczanski-Sobieski *et al.* (2002).
- CO, Collaborative Optimization, proposed by Braun and Moore (1996).

A thorough description of these formulation approaches is beyond the scope of this discussion. The interested reader is referred to the original works of the authors. It suffices to say that all the multiple-level formulations tend to modify the relationship of a non-hierarchical structure into a hierarchical one, in such a way to be able to use an analyzer/optimizer for every single element and one at system level. The multi-level architecture of the CSD problem considered in this thesis is similar to the CO proposed by Braun and Moore but with an underlying MDF formulation and the presence of *human* in the loop, see Figure 2.8. It is a Bi-Level architecture; indeed there are team members responsible for element analysis and others responsible for system analysis. The thick lines represent input/output interfaces. The experts have full authority on the element of interest and can allow flow of data from/to the other element models and from/to the system level model.

Problem formulation, conclusions

The choice between the various formulation approaches mainly depends on the objectives of the analysis, on the complexity of the problem and on the design environment in which they will be adopted. From a practical point of view, the MDF approach can be considered the easiest and most straightforward. It inherits its working principle from the classical engineering design approach: multiple iterations until convergence, for every given design variable set. On the other hand the IDF and AAO seem promising, in terms of parallelization of the analyses and reduced computational effort (Cramer *et al.*, 1993; Tedford and Martins, 2006). In Figures 2.5 through 2.7 the choice of using a *team* on top of the various architectures is not casual. Indeed, the discussion presented in this thesis is intended to provide a general design approach in *support* of a team working in a collaborative environment. This means that the analysis should be tuned based on the type of models implemented in this kind of environments. Thus, since in a collaborative environment the *human* is still the fulcrum of the design process, the mathematical models can only be implemented in such a way that a converged solution is presented for every input provided, *ergo* using an MDF formulation. Non-feasible solutions would be discarded by the domain experts before sharing the results with the team. An additional consideration is that *de-facto* time and resource limits are often the termination criteria for the analysis, thus requiring meaningful results to be obtained as soon as possible during the design process. Considering the team as the final recipient does not mean that automated techniques, *optimization* for instance, will not be taken into account. A thorough discussion on the design techniques used on top of the mathematical models is provided in the following chapters.

This concludes the brief digression on the modelization peculiarities of a complex system. Single, see Figure 2.5, and multiple, see Figure 2.8, level architectures, both with MDF formulation and Non-Hierarchical Decomposition, are the two types of modelization architectures of complex systems that are taken into account throughout this thesis. All the models used as test case, which are described in the following section, are developed and implemented with a single-level MDF formulation. However, the methods described in this thesis have also been tested on models that present a COBiL formulation, see Chapter 6.

2.3 Test cases

The test cases used in this thesis serve the purpose of demonstrating the working principle of the analysis methods that we propose. These methods are developed and discussed following a bottom-up approach. This means that we present local methods first, describing the global ones later. Local methods are mainly developed to support design activities performed to better evaluate a design baseline, therefore with limited dimensions of the design space. Global methods, on the other hand, are used to support the engineering activities in the exploration of design spaces of large dimensions. The concept of *local* and *global* analysis shall not be confused with the levels of the system decomposition. Indeed, both local and global analysis methods will be used at both subsystem and system level, depending on the objective of the analysis.

2.3.1 The communication and power subsystems design

The design of the communication subsystem and the power subsystem of a spacecraft is not a complex activity *per se*, if the analysis is tuned to estimating their design characteristics at a preliminary stage. However, their performance is affected by, and affects, many other aspects of the design of the entire mission and satellite system. The orbit, the payload, the attitude control and the supporting structure of the spacecraft, the ground segment (just to mention a few) all have a relevant influence on the fundamental parameters responsible for the determination of the link with a (several) receiver(s) on Earth and the estimation of the power required on board together with the subsystems mass. For instance, some of these parameters are the communication path length, the data-rate of the link, the pointing accuracy, and the antenna characteristics of the ground segment(s). The design of the coupled communication and power subsystem is performed with the objective of providing the spacecraft with enough capabilities of sending (receiving) to (from) Earth the required amount of data, to (from) the required location(s), in the required time, and making sure that the quality of the communication is sufficient enough for a correct reconstruction of the data. This shall be obtained with a minimum mass and with a minimum (programmatic) risk. The coupled Communication and Power subsystem mathematical models are used to demonstrate the working principle of the Regression-Based Sensitivity Analysis presented in Chapter 3, and to show that uncertainty analysis, already performed at conceptual level, may bring benefits to the design lifecycle of the system. Detailed information on the mathematical models and the design settings is provided in Appendix A.

2.3.2 Satellite system for Earth-observation

The model of the satellite system for Earth Observation is used to study two different missions. We increment the complexity when compared to the previous case, going from subsystem-level to system-level analysis. The first mission has the objective of obtaining world-wide coverage, providing disaster-management capabilities. In Chapter 3 we use sensitivity and regression analysis for the design of the system baseline. Further, uncertainty analysis is implemented for a quantitative determination of the mission cost, mass, and power-consumption margins. The second mission is again an Earth-observation mission, but in this case we focus on a specific area around the Bay of Bengal. We use this mission in Chapter 4 to demonstrate the implementation of the Pareto Robust Optimization Algorithm. Further, we show that local analysis techniques coupled with global methods can provide the engineering team with quantitative information on the system that allows to speed-up the decision-making process maintaining high-quality standards. Detailed information on the mathematical models and the design settings is provided in Appendix B.

2.3.3 Lunar space-station mission design

The design of the Lunar space-station mission foresees the utilization of multiple systems whose characteristics have to be balanced. The main mission objective is to deliver a manned space station in low orbit around the Moon. The mathematical model developed for this analysis is characterized by having a discrete configuration of the mission architecture. Indeed, several building blocks and delivery strategies are considered for the analysis, see Appendix C for more details. The actual transfer to the Moon is not part of the study presented here, therefore we consider standard values for the velocity changes of each required maneuver. When it comes to conceptual design of new missions and systems, models with mixed continuous-discrete variables are very common. This test case is a good example to demonstrate that global multiobjective techniques can be used by the engineering team to choose amongst the different mission architectures and at the same time selecting the best combination of design variables that optimize them.

2.3.4 Atmospheric entry vehicles design

The design of unmanned entry capsules, considering continuous shape-variation models, aerothermodynamics, flight mechanics, and thermal protection system models at the same time, is considered to be a valuable test-bed for the robust-optimization methods presented in this thesis. The model has been adapted from the previous work of Dirkx (2011); Dirkx and Mooij (2011), adding a one-dimensional lumped-parameters thermal model for the thermal protection system (TPS). All the details regarding the shape, aerothermodynamics, and flight mechanics can be found in the original literature by the authors. The types of TPS, the material properties used for the analyis, and the validation of the model are presented in Appendix D, instead. In Chapter 5 we discuss the results of minimizing the mass of the capsules while maximizing the internal volume and the re-usability.

2.3.5 Ops-Sat, a cubesat mission in the ESA Concurrent Design Facility

Ops-Sat was a recent study in the Concurrent Design Facility (CDF) at ESA aimed at the design of an in-orbit demonstrator to test innovative mission-control and operations concepts by using a *3U* cubesat. A *3U* cubesat is a satellite with the dimensions of approximately $10x10x30$ cm. It is called *3U*, because its volume is three times larger than the *1U* cubesat which measures $10x10x10$ cm. Due to the limited complexity of the mission, in the Ops-Sat study the design was far more detailed than a classical conceptual study. The convenience of using a system margin of 20%, as is typically the case in CDF studies, would maybe not represent the optimal choice. Further, it was the first time that a satellite of such a small scale was designed in the CDF, thus it was clear from the beginning that the mass-margin philosophy commonly accepted (and corroborated by experience) for larger-scale satellites would be hardly applicable in this case. For these reasons, it was decided to adopt a statistical approach to the mass-budget management in parallel to the classical one. This analysis was performed using some of the analysis methods presented in this thesis. In particular, we used the Augmented Mixed Hypercube sampling approach and the uncertainty analysis described in Chapter 3 to propagate the uncertainty distribution of the mass of the elements of Ops-Sat from subsystem level to system level. The utilization of these methods in the CDF proved to be very effective in capturing the knowledge of the team members and synthesizing it for supporting the activity of the systems engineer in determining the mass budget. Indeed, using detailed information from the subsystem experts as input data, it was possible to determine a baseline-mass of Ops-Sat that was 4.75% lower than the initial estimate.

2.3.6 Support of the Concurrent Design Platform^{TM} at JAQAR-Concurrent Design Services B.V.

The purpose of this cooperation with JAQAR-Concurrent Design Services B.V. (J-CDS) is to demonstrate that the design methods described in this thesis can be effectively used as an application layer on top of the concurrent engineering infrastructures to boost the concurrent design process by supporting the activities of the engineering team members. In particular, we cooperated with J-CDS implementing some design methods described in Chapter 3 in the Concurrent Design Platform (CDP^{TM}). As a first test case, we supported the design process of J-CDS for the business analysis of a new non-space product to be commercialized (a medical devise). The results highlight the best policy in terms of make-buy of the various elements of the product and identify cost-reducing trends in the design parameters. The design within the scope of a training session for a fictional scientific space mission was considered as a second test case. The purpose of the mission is to place an instrument on the Moon's surface. We supported the activity of the subsystems engineers in dealing with flexible requirements and enhancing the functionality of the CDP^{TM} in allowing the design also when data from other disciplines are missing.

2.4 Summary

This ends the analysis on the most relevant aspects related to the design of a complex system, that will be considered in this thesis. Under the main assumptions that:

- the design of a complex system is performed based on the utilization of a mathematical model describing it;
- the analysis of the model is performed considering multiple objectives, that can be linear or non-linear, and continuous or not. It may also present multiple constraints, which can be expressed as linear or non-linear, and continuous or discontinuous quantities;
- the design variables that characterize the model can be continuous or discontinuous. The last class can be further distinguished in ordinal and categorical;
- the model can include uncertain parameters;
- the model is developed and implemented according to a single or multiple-level architecture with an MDF formulation and an NHD decomposition approach;

then, the main objective of the thesis is to provide a structured approach to support the engineering team in the conceptual-design activities, eventually performed in a collaborative environment. The design activities of interest are factor screening and sensitivity analysis, determination of the design-space shape, robustness analysis, uncertainty analysis, optimization and robust optimization. They can be performed at a local level on a small subset of the design space, and/or at a global level on the entire design space, on a single element only, and/or on the complete system.

Chapter 3

Local Design Approach

This chapter presents a thorough analysis of the problems encountered when designing a complex system and the methods proposed to solve them, at a local level. By *local* we mean design techniques that are useful to support the engineering team when a limited portion of the design space has to be analyzed. This is valid, for instance, when a certain design option has to be studied in detail to compare it to other design options. The digression on design techniques for local analysis begins with sampling, in Section 3.1. Then, sensitivity analysis is discussed in Section 3.2, and response surfaces and graphical support to the engineering team in Section 3.3. In Section 3.4 we discuss uncertainty analysis and robust design. In this chapter we use two of the mathematical models briefly introduced in the previous chapter. In particular we demonstrate the application of the analysis methods discussed in this chapter to the *communication and power subsystems* model (Sections 3.2.4 and 3.4.4) and the *satellite mission for Earth-observation* model (Sections 3.2.6, 3.3.3 and 3.4.2).

3.1 Sampling the design space

Sampling the design space is the first step necessary when the mathematical model of a system needs to be studied. A sample is a set of points in the design space (a k-dimensional hyperspace) whose coordinates are the values of the design variables taken from their variability ranges, $(x_1, x_2, \cdots, x_k)$. In Figure 2.1, in the previous chapter, the black dots represented sample points in the design space and their *image* in the objective space. The model is executed using each sample point as input. The corresponding output, *i.e.*, the performance, can then be studied in detail to make conclusions on the correlation between input and output, and set the design factors such that the system model behaves as required. The purpose of this section is to discuss the advantages and disadvantages of sampling techniques for continuous and discrete variables. This discussion will bring us, at the end of this section, to propose the mixed-hypercube approach. It is a mixed sampling method that will allow us to take both continuous and discrete design variables into account, at the same time.

3.1.1 Pseudo-random sampling

A random sample is obtained from the design region of interest using numbers which are *casually* generated, usually by a computer. Therefore, random sampling is sometimes referred to as *pseudo-random*. Indeed, the apparently casual numbers generated by a computer are the result of a deterministic process. Many algorithms for producing pseudo-random numbers are provided in Press *et al.* (2007). When one refers to Monte-Carlo techniques, very often pseudo-random sampling is behind it. In this chapter we will highlight advantages and disadvantages of using Monte-Carlo techniques, that are considered only one of the many options available

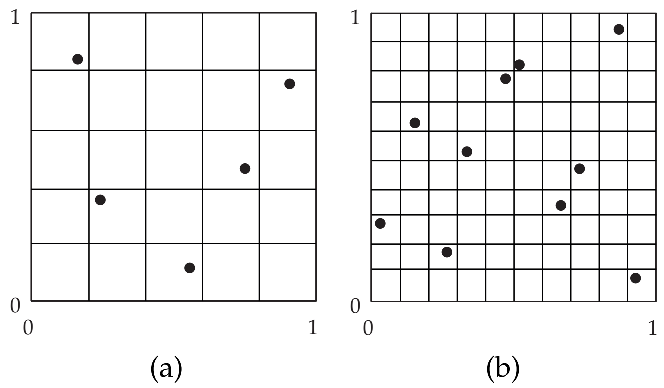


Figure 3.1 Latin Hypercube sampling with 2 variables ($k = 2$) and (a) $N = 5$ (b) $N = 10$.

for sampling the design space of a mathematical model. All the design methods presented in this thesis may be implemented simply using the Mote-Carlo approach for sampling. One of the main intents, is actually to define what the best sampling technique is (*i.e.,* that allows computationally cheaper analyses) given specific problem settings.

3.1.2 Stratified sampling

In statistical jargon, it is said that stratified sampling allows to improve the representativeness of the sample (Press *et al.*, 2007). In practice, this means that stratified sampling allows to obtain a more uniform coverage of the design region of interest, compared to a random-sampling method. The idea is to partition the design region of interest into disjoint k-dimensional subsets, called *strata*, in such a way to obtain a multidimensional grid partitioning the design region of interest. Then, random sampling is used to collect sample points from each stratum.

Latin hypercube sampling

Latin hypercube sampling (LHS) is a subclass of the stratified sampling technique. According to the original implementation of the LHS method proposed by McKay *et al.* (1979), the design space is divided into k^N disjoint subsets (strata). This means that the variability range of each variable is divided into N intervals. The N sample points are taken in such a way that a point is taken once from each stratum, see Figure 3.1. Due to its nature, the LHS provides in general an increased coverage of the design space if compared to the stratified sampling and the random sampling. However, the fact that a sample is taken from each stratum randomly, does not always guarantee to obtain a sample with good space-filling qualities, as demonstrated by Viana *et al.* (2010). As a consequence, since its first implementation LHS has been the object of the application of many optimization methods to obtain samples at the maximum relative distance between each other (Morris and Mitchell, 1995; Ye *et al.*, 2000; Jin *et al.*, 2005; Grosso *et al.*, 2009). Viana *et al.* (2010) propose an algorithm for near-optimal Latin hypercube designs without using formal optimization. This method provides results with a negligible computational effort if the number of design variables k is not so large. According to our experience using the algorithm proposed by Viana *et al.* (2010), it requires the generation of matrices with at least 2^k elements, irrespectively of the number of sample points actually required. The number of matrix entries to be stored to compute the near-optimal LH-design can already become cumbersome for 20 variables.

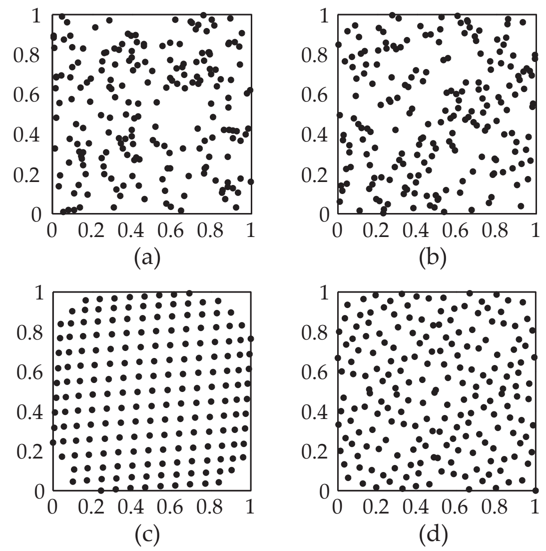


Figure 3.2 Scatter plots of 100 sample points in a 2-dimensional design space based on (a) pseudo-random sampling, (b) Latin hypercube sampling, (c) sub-optimized Latin hypercube sampling, (Viana *et al.*, 2010), (d) Modified Sobol' LP_τ sequence.

3.1.3 Quasi-random sampling

The class of quasi-random sampling techniques is developed with the main objective of obtaining a set of points that are *well* spaced in the hyperspace determined by the design-variable ranges (the design region of interest). It is usually said that these algorithms provide *low-discrepancy* sample points, with the discrepancy indicating a measure of *non-uniformity* of the sample (Bratley and Fox, 1988). Three alternate approaches are proposed by Faure (1982), Halton (1960), and Sobol' (1979) respectively. Bratley and Fox (1988) provide a good review and references of all of them. However, the most widely used approach is the one proposed by Sobol'. Bratley and Fox (1988) and Press *et al.* (2007) give useful indications on how a Sobol' LP_τ sequence, or its variant proposed by Antonov and Saleev (1979), can be computed.

The modified LP_τ algorithm is extensively used in this thesis as basis for all the sampling methods described hereafter. The modified Sobol' LP_τ sequence was selected for its particular characteristic of providing a sequence of sampling points for which successive points at any stage *know* how to fill in the gaps in the previously generated distribution (Press *et al.*, 2007). This aspect is particularly useful for limiting the computational load in performing a sensitivity analysis as described in Section 3.2.3. A comparison of the methodologies in sampling a 2-dimensional design region is presented in Figure 3.2, while the *gap-filling* properties of the same methodologies are presented in Figure 3.3. The modified Sobol' LP_τ sequence demonstrates that the additional sample points, the circles in Figure 3.3, are placed in such a way to fill the gaps following a sort of pre-defined pattern, allowing the re-utilization of the samples previously generated, because they are placed in the design space in a low-discrepancy sense.

3.1.4 Design of experiments

An experiment is a test executed to evaluate the outcome of a process given certain settings of the factors believed to influence it. The *experiments* considered here are all *computer experiments* performed on the mathematical model in correspondence of the sample points. However, the Design of Experiment (DoE) practice has older origins than the computer era, indeed it was first introduced by Fisher in 1935, (Fisher, 1971). The sampling methods belonging to the category of DoE can be distinguished in Factorial Designs (full or fractional), Orthogonal Arrays

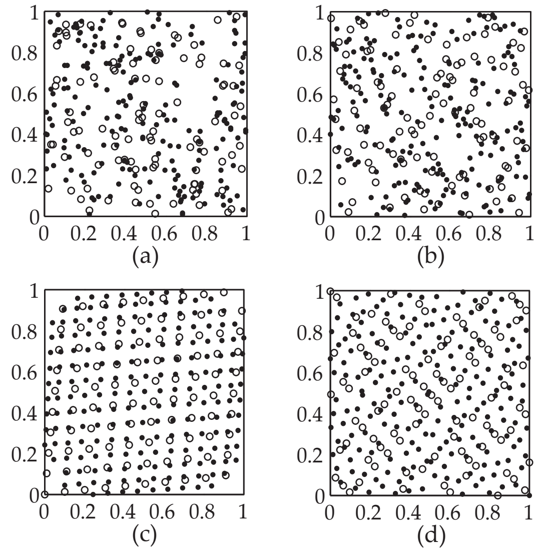


Figure 3.3 Scatter plots of sample points in a 2-dimensional design space based on (a) random sampling, (b) Latin hypercube sampling, (c) sub-optimized Latin hypercube sampling, (Viana *et al.*, 2010), (d) Modified Sobol' LP_τ sequence. • Initial set of sample points. ○ Additional set of sample points.

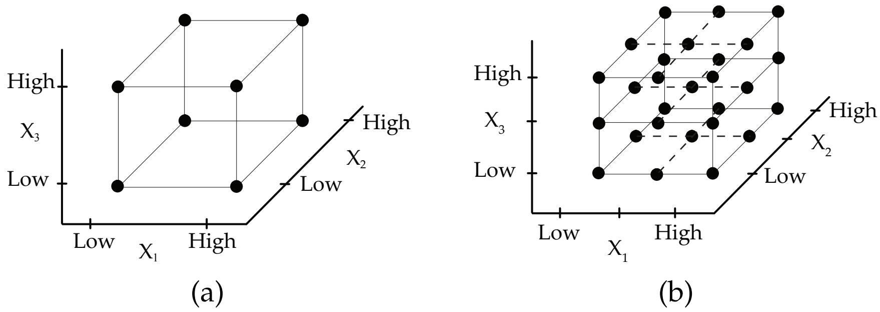


Figure 3.4 Full factorial design with (a) 2 variable-levels and (b) 3 variable-levels in a 3-dimensional design space.

and other methods, amongst which, for instance, Central Composite Design (CCD). The common characteristic of these sampling methods is that they are all deterministic: the samples are placed in the design space according to a certain pre-defined geometry. This allows also *ordinal* and *categorical* variables to be used in the analysis. Random or quasi-random sampling methods do not directly provide this feature. In the case of DoE, the values of the variables are more appropriately called *levels*.

Factorial design

Full factorial design is a sampling method that foresees one experiment for each possible combination of the factor levels. If factor A has a levels, factor B has b levels and factor C has c levels, the total number of experiments is $N = a \cdot b \cdot c$. There are special cases of factorial design where for all the factors only 2 or 3 levels are considered. They are usually called 2^k- and 3^k- factorial designs, respectively, where k indicates the number of factors. The experimental structure obtained for 2^k- and 3^k- factorial designs is shown in Figure 3.4, where the dots indicate the sample points.

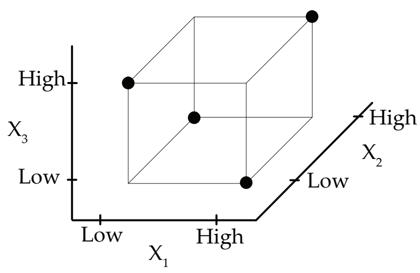


Figure 3.5 Fractional Factorial design in 3-dimensional design space.

Experiment	Factors A	B	C
1	1	1	1
2	1	1	2
3	1	2	1
4	1	2	2
5	2	1	1
6	2	1	2
7	2	2	1
8	2	2	2

Table 3.1 Full factorial design, 3 variables at 2 levels.

In general, full-factorial design may also be called *grid search*: all the sample points are placed in the design space in such a way to form a grid, see Figure 3.4 for instance. Full-factorial design requires a number of experiments that increases with the power of the number of factors. Thus, already in the case of 2^k or 3^k factorial designs, the experimentation (*i.e.,* the simulation of the model) can become cumbersome very soon. Therefore, fractional-factorial designs were introduced as an attempt to reduce the computational effort for the analysis. As the name suggests, fractional-factorial designs only foresee a fraction of the number of experiments required by a full-factorial design with the same number of factors and the same number of levels. For instance a *one-half* fractional factorial design, or 2^{k-1} design, requires half of the experiments of the original 2^k design, see Figure 3.5.

All the designs belonging to the category of DoE are also called matrix designs. Indeed their visualization, and their construction, is better understood if represented in the form of a matrix with the factors in the columns and the experiments to do in the rows. A graphical structure for more than 3 variables becomes hard to visualize. In Table 3.1 we show, for instance, the matrix visualization of a 2^k design, with $k = 3$. The level 1 indicates the *low* level, while 2 indicates the *high* level of the design factors. A 2^{k-1} design is also called *Resolution 5* design (for $k > 4$). It requires half of the simulations required by a full-factorial design, with the same number of parameters, with two levels. It is also possible to generate fractional-factorial designs that require less experiments than *Resolution 5*. However, the smaller the number of experiments, the lesser the information that can be obtained. In Section 3.2.2 we will demonstrate that the information that is available after performing a matrix design strongly depends on the type of chosen design, *i.e.,* on the number of performed experiments.

Orthogonal arrays

Orthogonal Arrays (OAs) are special matrix designs originally developed by Taguchi (1987). In Table 3.2, we present an L8 orthogonal array, *i.e.,* 8 experiments with a maximum of 7 factors at 2 levels. To represent the orthogonal array of Table 3.2 in a graphical form, as shown in Figure 3.5, a hypercube of 7 dimensions (because 7 factors are taken into account) would be required.

Experiment	Factors						
	A	B	C	D	E	F	G
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

Table 3.2 L_8 orthogonal array.

Experiment	Factors			
	A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

Table 3.3 L_9 orthogonal array.

An orthogonal array has the advantage of reducing the computational effort in exploring the design space. Indeed, with a full factorial design the number of required experiments would have been 2^7, 128.

The term *orthogonal* is related to the balancing property, which means that for any pair of columns, all combinations of factor levels are present an equal number of times. For instance, in Table 3.2, in the first two columns all the factor combinations (*i.e.,* 1 1, 1 2, 2 1, and 2 2) are present two times. The same happens for any other pair of columns. This property assures that once all the simulations are completed, for any couple of factors all the factor-level combinations are evenly tested. The utilization of matrix designs for experiment planning is straightforward. Referring to the OA presented in Table 3.2, for instance, in each column we read the level that has to be assigned to the design variables for each experiment. In each row, we read the variable sets for every single experiment to be performed. Therefore, the number of rows represents the total number of experiments. The L_8 orthogonal array of Table 3.2 is only one amongst the many OAs discussed by Taguchi (1987). It is also possible to build three-, four-, and five-level OAs, and also mixed-levels OAs for factors having a heterogeneous number of levels (Phadke, 1989). An L_9 OA is presented in Table 3.3, where four factors are considered at three possible levels. As will be shown in Section 3.2.2, three-level arrays will allow to study also quadratic effects of the factors on the performance of interest. An efficient algorithm to generate three-level OAs is discussed by Mistree *et al.* (1994) while standard tables for other types of orthogonal arrays can be found in (Taguchi, 1987) and (Phadke, 1989).

Other experimental designs

The major distinction amongst the experimental designs is usually made between first- and second-order designs, as already hinted before. In the first case the design variables can assume only two levels, while in the second case at least three levels per design variable are

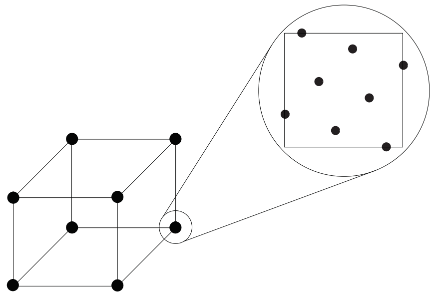


Figure 3.6 Mixed-hypercube sampling with 3 discrete and 2 continuous variables.

considered. The development of second-order designs is mainly related to the need of obtaining information on the curvature of the design space for fitting second-order response surfaces. Box *et al.* (1979) present a method to compute fractional 3^k factorial designs, the Box-Bhenken designs, obtained by combining two-level factorial designs with balanced incomplete block designs. The Central Composite Design (CCD) introduced by Box and Wilson (1951), is built using a 2^k-factorial design, plus n_0 central points (in the geometric center of the design hyperspace), plus $2k$ points on the axis of every design variable at a distance α from the center. In a hyperspace normalized in the interval $[-1, 1]$, a CCD with $\alpha \neq 1$ will present 5 levels for each variables, while with $\alpha = 1$ it will only require the variables to assume 3 different levels. The value of α can be chosen by the designer to fulfill certain characteristics. With $\alpha = 2^{\frac{k}{4}}$, for instance, one can obtain a rotable design, the preferred class of CCD (Montgomery, 2001). In a rotable design, the variance of the predicted performance is not biased in any direction, it is only a function of the distance from the center. For computer experiments $n_0 = 1$ is a logical choice, since there is no variability associated with executing the model with the same settings of the design variables for more than one simulation. The interested reader is referred to Box *et al.* (1979) and Montgomery (2001) for a good overview and discussion on the many types of available experimental designs.

3.1.5 The mixed-hypercube approach

The mixed-hypercube approach is a mixed sampling method that we developed to be able to take both continuous and discrete variables into account (**????**). In particular, with the mixed-hypercube approach we use both sampling for continuous variables and DoE. The main idea is to separate the continuous and discrete variables in two groups. A matrix design is then created for the discrete variables while for every row of the matrix design (*i.e.*, for every design point of the factorial design), a Sobol' sequence is generated for the continuous variables. An example with three discrete and two continuous variables is presented in Figure 3.6.

The advantage of using a matrix design instead of a space-filling technique for the discrete variables is that it allows to deterministically select the levels of the factors. When only few factor levels can be selected (*e.g.*, in a database there is a certain number of batteries, or only a limited number of thrusters is considered in the analysis of a satellite system) the maximum number of simulations is determined by a full-factorial design. Then, depending of the type of analysis, and the available resources, one could choose for a fractional-factorial design. This will allow to reduce the computational effort while avoiding to disrupt the balance characteristics of the sampling matrix. The modification of a random or pseudo-random technique for sampling only at certain levels does not immediately provide such a balance, especially when the number of samples is kept low. On the other hand, in case of continuous variables matrix designs alone are less flexible in *filling* the design region and less suitable for the *re-sampling* process than the Sobol' technique. The proposed mixed-hypercube sampling approach allows for covering the design region more uniformly when compared to all the other techniques mentioned in this section, already with a low number of samples. The sensitivity-analysis technique described in Section 3.2.3, will directly benefit from these characteristics, since convergence of the variance is obtained with a reduced computational effort, for instance. A more detailed description of the implications of using specific implementations of the mixed-hypercube sampling method in combination with the design approaches discussed in this chapter is presented in the following sections.

3.2 Sensitivity analysis

Sensitivity analysis (SA) is a technique used in many scientific and technical environments with different purposes, such as the determination of the quality of a certain model, validation of assumptions, or as a method to identify important factors. From an engineering perspective, the team is usually interested in understanding the consequences of certain settings of the design parameters on the performance of the system. In this context, sensitivity analysis can be described as the study of the *effect* of a certain model input X_i (or group of inputs) on a given model output Y_j. It allows to identify design drivers, *i.e.,* those factors or group of factors that shall be carefully assessed by the designer, because they will be the principal responsible for determining the performance of the system. The extent of the influence identified, may be useful for checking the adequacy of the model being used for the analysis and for corroborating the underlying analysis assumptions. In other words, sensitivity analysis performed already at an early stage of the design process will speed up the process itself, while at the same time it will provide the engineering team with *an X-ray machine* that allows to understand the effect of their design choices on the system.

In this section we introduce the Regression-Based Sensitivity Analysis method (RBSA). It is developed to provide global, quantitative measures of sensitivity with a computationally cheap approach. The objective of this section is to discuss the basis for global sensitivity analysis, to introduce RBSA, and to compare its performance with other known global sensitivity analysis methods.

3.2.1 Global sensitivity analysis

The *effect* mentioned before can be the result of a local measure, *e.g.,* the measure of a derivative, *e.g.,* $(\partial Y/\partial X)_{X=x^*}$, which requires an infinitesimal variation of the input X around a specific value x^*. Local sensitivity analysis is usually considered when studying operating-point stability of dynamic systems, for instance. It requires, amongst others, the determination of the stability properties in the presence of small variations of the factors that influence the dynamics of the system. Consider, for instance, the scheme in Figure 3.7(a). A local measure of sensitivity is intended as the determination of the variation of the performance Y, locally, when $X = x^*$. This measure, however, does not represent the behavior of the performance Y in the entire interval of interest ΔX.

The measure of sensitivity can also be obtained when the input varies over a specified finite interval ΔX. In this case, sensitivity analysis is valid over the entire interval of variation spanned by the input factor (*i.e.,* the design region of interest) rather than only directly around a single (operating) point. Therefore, this type of sensitivity analysis is often called *global*.

The settings of the problem of designing a system by selecting the most appropriate combination of input-factor levels is particularly suitable for the implementation of global sensitivity

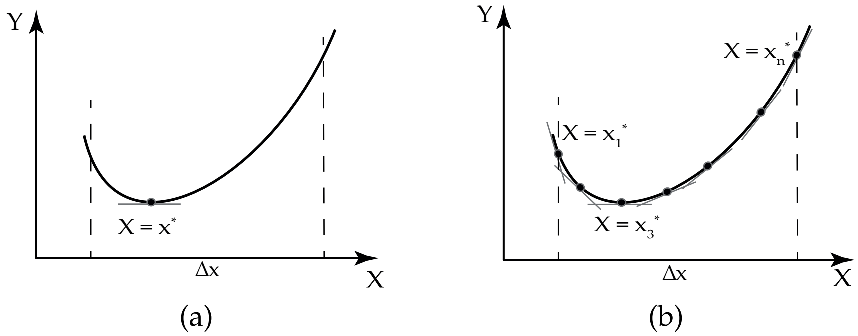


Figure 3.7 Sensitivity Analysis. (a) Local approach to sensitivity analysis, (b) global approach to sensitivity analysis.

analysis. Indeed, in this context sensitivity analysis is aimed at finding the set of relevant factors in the determination of the output, and the global implementation provides an answer that is valid over the entire design region, even if it represents only a (small) subset of the design space. Global sensitivity analysis is locally implemented, in the discussion presented in this chapter. This means that the design region of interest is small, thus the analysis is local with respect to the entire design search space. However, within the design region of interest, the analysis is global, according to the definition given before.

The importance of a factor can be determined on the basis of the reduction of the (unconditional) variance of the output Y, $V(Y)$, by fixing that factor to a certain (yet unknown) value. This also means how much of the total output variance is determined by the variation of that specific factor. In a local sense the focus would be on computing the conditional variance $V(Y|X_i = x_i^*)$. Since the value of x_i^* is unknown, it is reasonable to compute the average over all possible values of X_i within the design region of interest, thus obtaining $E(V(Y|X_i = x_i^*))$. The conditional expectation $E(\cdot)$ is computed over X_j, $j \neq i$ with $X_i = x_i^*$. The variance, instead, is computed considering all the possible values of x_i^* (Saltelli *et al.*, 2004, 2008). In Figure 3.7(b), we show that to obtain a global measure of sensitivity, many points $X = x_i^*$ must be taken into account.

The smaller $E(V(Y|X_i = x_i^*))$, the larger the factor influence on the output is. The term $V(Y|X_i = x_i^*)$ is related to the variance computed with a fixed value for the variable of interest, with all the other variables varying in their intervals. Therefore, if the expected value of the variance computed in this way is low (in percentage with respect to the total variance $V(Y)$), then it means that the variability caused by the variable of interest must be high.

The computation of global sensitivity based on the variance of the response is a growing (and also logical) practice. It allows for taking the dimensions of the design region of interest into account to provide multi-dimensional averaged information on the effect of the factors on the output. The knowledge of the *importance* of the factors in their contribution to the output variance, is used to identify and fix the non-influential factors (or those with a limited influence) on the determination of the output of the model. The most important ones may be ranked and their effect may be studied in more detail.

Global, variance-based, sensitivity indices can be estimated using qualitative or quantitative methods; it depends on the purpose of the analysis, on the complexity of the problem and on the available computational resources.

A qualitative approach, like the method of Morris (1991), allows to determine the importance of the factors with a relatively limited computational effort. As we will demonstrate later, this comes with a limitation. The method of Morris, and other qualitative methods, can only rank input factors in order of importance. The method is unable to determine a quantitative measure of the contribution of the factors to the variability of the performance. Therefore, these methods are usually used as a preliminary analysis to detect and fix the unimportant factors. Therefore, qualitative methods are also called *screening methods*. On the other hand, quantitative techniques like the method of Sobol', Sobol (1993) and Saltelli *et al.* (2004), or the FAST (Fourier Amplitude Sensitivity Test), Cukier *et al.* (1978), require a large number of model evaluations to provide sensitivity indices of the design factors, especially the terms like V_{ij}, or $V_{ij\cdots k}$ (Helton and Davis, 2003). This may be a limitation when a large number of input factors are taken into account, or when the model is computationally time consuming.

Regression analysis has been, and still is nowadays, extensively used to assess the effects of the input factors on the output. The RBSA introduced in this section, is a method for global sensitivity analysis based on the regression of a general polynomial model. The sensitivity indices, for each design factor, are computed by decomposing the global variance detected by the model, apportioning it to the various factors and combinations of them, rather than simply relying on the regression coefficients (that are a local measure of sensitivity, for more-than-linear effects). This allows for obtaining global sensitivity indices, and providing information on the first-order as well as on higher-order factor effects on the model's output.

3.2.2 Methods for sensitivity analysis

Sobol' method

The sensitivity indices introduced by Sobol' can be used as possible indicators of the relative importance of the factors in the determination of the variance. The method of Sobol' for computing the sensitivity indices is based on the decomposition of the total (unconditional) variance in the contribution of each single factor. Consider, for instance, $Y = f(X)$ as the model of interest. Y is the output vector while $X = (x_1, x_2, \cdots, x_k)$ is the vector of the k independent input factors. The method of Sobol' discussed here and the regression-based sensitivity analysis described later are in general valid for independent (*i.e.*, non-correlated) input factors. The case with correlated inputs implies that the correlation structure must be taken into account during the sampling of the design space, leading to higher computational cost on one hand and to a non-direct applicability of the method on the other hand (Saltelli *et al.*, 2004). An effective technique for imposing the correlation between input variables has been proposed by Iman and Conover (1982).

To compute the sensitivity, a sample of N points is taken from $Y = f(X)$ by evaluating the model N times. The unconditional variance $V(Y)$ can be decomposed as follows (Sobol, 1993):

$$V(Y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \cdots + V_{12\cdots k} \tag{3.1}$$

where V_i is the variance of Y due to factor i, V_{ij} is the variance of Y due to the interaction between factor i and factor j. All the terms of this relationship are conditional variances of the factors indicated by the subscripts. For factor i, for instance, $V_i = V(E(Y|x_i))$. For the interaction factor V_{ij}, instead, $V_{ij} = V(E(Y|x_i, x_j)) - V_i - V_j$, which is the combined effect of the factors x_i and x_j minus their individual conditional variances V_i and V_j.

Since the following relationship holds, $V(Y) = V(E(Y|x_i)) + E(V(Y|x_i))$, and since the unconditional variance $V(Y)$ is constant, an important factor will lead to a small value of $E(V(Y|x_i))$, as anticipated before, or equivalently to a large value of $V(E(Y|x_i))$ (Saltelli *et al.*, 2004). Therefore, each term in Eq. (3.1) can be used as a measure of global sensitivity. Indeed, Sobol' sensitivity indices are defined as follows (Sobol, 1993):

$$S_i = \frac{V(E(Y|x_i))}{V(Y)} \tag{3.2}$$

S_i is sometimes called the *first-order sensitivity index* to distinguish it from *higher-order sensitivity indices* S_{ij}, S_{ijw}, or S_{ii}, which represent the effects of the interactions between factors or the effect of higher-order terms on the unconditional variance.

Another measure of sensitivity is represented by the so-called total-order sensitivity indices, S_{Ti}. A total-order sensitivity index takes the contribution to the unconditional variance of a certain variable i into account, considering the first-order and all higher-order effects that involves it. A total sensitivity index provides an indication of the overall effect of a certain variable on the response of the model. The total-order sensitivity indices can be computed as follows (Saltelli *et al.*, 2004):

$$S_{Ti} = 1 - \frac{V\left(E\left(Y\left|X_{-i}\right.\right)\right)}{V\left(Y\right)} \tag{3.3}$$

where $V\left(E\left(Y\left|X_{-i}\right.\right)\right)$ indicates the contribution to the variance due to all factors with the exception of x_i. The vector $X = [x_1, x_2, \ldots, x_i, \ldots, x_k]$ contains all the design factors. The vector $X_{-i} = [x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k]$ contains all the factors except x_i.

To provide a quantitative example of the results that can be obtained using sensitivity analysis, let us consider the following simple problems. Y_1 is a linear additive model, in Y_2 a non-linear (fourth-order) term is added, and Y_3 introduces an interaction term between x_1 and x_2, see Eq. 3.4 to Eq. 3.6.

$$Y_1 = x_1 + x_2 + x_3 \tag{3.4}$$

$$Y_2 = x_1 + x_2 + x_3 + x_3^4 \tag{3.5}$$

$$Y_3 = x_1 + x_2 + x_3 + x_1 x_2 \tag{3.6}$$

where $x_1 \in [-9, 9]; x_2 \in [-5, 5]; x_3 \in [-3, 3]$. The first-order sensitivity indices computed with Eq. (3.2) are presented in Table 3.4. In Table 3.5 the total sensitivity indices computed with Eq. (3.3) are reported instead. As shall be demonstrated in Section 3.2.5, the sensitivity indices are *sensitive* to the number of samples used to compute them. For these examples, the results have been obtained using a sample size of 5000, which provides a good approximation of the sensitivity indices, that does not significantly change for larger sample sizes.

In Tables 3.4 and 3.5 one would expect the sensitivity indices of the factors to be the same in the case of Y_1, since x_1, x_2, and x_3 have the same multiplicative coefficient (*i.e.*, 1). Sensitivity indices also take the range of variation into account, and in this case x_1, x_2, and x_3 are allowed to vary over different ranges, and this is reflected in the sensitivity indices.

The discrepancy between first-order and total-order sensitivity indices depends on the type of model and type of factor under analysis. When interactions are present, as in the case of Y_3 for the factors x_1 and x_2, the sum of the total-order sensitivity indices is larger than one. This is an expected result since the effect of the interaction $x_1 x_2$ is considered both when determining the effect of x_1 and the effect of x_2. In fact, in this case, the vector $X_{-1} = [x_2, x_3]$, so we study the sensitivity of Y_3 with respect to x_1 and $x_1 x_2$. The vector $X_{-2} = [x_1, x_3]$, so we study the sensitivity of Y_3 with respect to x_2 and $x_1 x_2$. The sensitivity of Y_3 with respect to $x_1 x_2$ is considered twice.

The Fourier Amplitude Sensitivity Test

The Fourier Amplitude Sensitivity Test (FAST) introduced by Cukier *et al.* (1978), is an alternative method that allows to compute the sensitivity indices of Eq. (3.2). In an analytical sense, the expected outcome of Eq. (3.2) should be computed by solving a k-dimensional integral, where k is the number of design factors. Suppose the model we consider is $y = f(X)$, with $X = [x_1, x_2, \ldots, x_k]$. Suppose now that X is a random vector with a certain Probability Density Function (PDF), $P(X) = P(x_1, x_2, \ldots, x_k)$. The x_i are not necessarily random variables. If

	S_i			Sum
	x_1	x_2	x_3	
Y_1	0.709	0.214	0.0793	1.0025
Y_2	0.0560	0.0156	0.924	0.996
Y_3	0.0728	−0.0043	0.0116	0.0801

Table 3.4 First-order sensitivity indices computed using the method of Sobol'. Results obtained with a sample size of 5000.

	S_{Ti}			Sum
	x_1	x_2	x_3	
Y_1	0.716	0.2198	0.0791	1.0149
Y_2	0.0540	0.0166	0.9235	0.994
Y_3	0.9675	0.886	0.0114	1.865

Table 3.5 Total-order sensitivity indices computed using the method of Sobol'. Results obtained with a sample size of 5000.

the PDFs are considered uniform, $P(X)$ can be considered the hypercube with samples drawn from a Sobol' sequence, for instance. The r^{th} moment needed to compute the indices of Eq. (3.2) is then found analytically by solving the following multidimensional integral (Saltelli *et al.*, 1999):

$$\langle y^r \rangle = \int f^r (x_1, x_2, \ldots, x_k) \, P(x_1, x_2, \ldots, x_k) \, dx \tag{3.7}$$

In the FAST algorithm the k-dimensional integral is converted into a single integral, that can be approximated using a random Monte-Carlo technique. Each factor x_i is associated with a certain frequency ω_i, using the relationship $x_i(s) = G_i(\sin(\omega_i s))$. The function G_i is a parametric equation that matches the variation of s to the variation of the parameter x_i in its proper range. Saltelli *et al.* (1999) proposed the Extended-FAST as an improved version of FAST. The limitation of FAST is that it allows for computing the *first-order* sensitivity indices only (Cukier *et al.*, 1978). With EFAST the total-order sensitivity indices of Eq. (3.3) can be estimated as well (Saltelli *et al.*, 1999).

The method of Morris

The method of Morris (1991) is a qualitative method for sensitivity analysis. It is based on the so-called elementary effect, which is a measure of the sensitivity in the form of incremental ratios, *i.e.*, an approximation of a local gradient within a finite interval of variation of the variable. As such, the elementary effect is a local measure of sensitivity:

$$d_i(X) = \frac{[y(x_1, \ldots, x_{i-1}, x_i + \Delta, x_{i+1}, \ldots, x_k) - y(\mathbf{x})]}{\Delta} \tag{3.8}$$

Each $d_i(X)$ is the result of a one-factor-at-a-time experiment. However, in the method of Morris, the final value attributed to the sensitivity of each design variable is obtained by averaging several elementary effects computed at different points of the input space (Morris, 1991). In Eq. (3.8), Δ is the width of the step in the i^{th} dimension of the design region needed to compute the incremental ratio. To compute the sensitivity measures for all the factors, the design region is fractioned into a grid of dimensions $k \times P$, where k is the number of factors and P is the number of levels in which every dimension is subdivided. The influence of a factor is determined by computing several elementary effects (the number of elementary effects is indicated by R) at points that are randomly selected from the grid. The method of Morris provides two qualitative measures of sensitivity, namely the mean, μ, and the standard deviation, σ, of the elementary effects. Large values of μ indicate that a factor has a prominent overall influence on the output. Large values of σ, instead, are the result of interactions of the factors with other factors or non-linear effects on the output. To compute μ and σ the method of Morris requires a sampling matrix made of $k+1$ sample points, for each elementary effect R to be computed. Therefore, the computational cost of the method of Morris is linear with the number of factors, *i.e.*, equal to $R \times (k+1)$. It was recognised that the method of Morris may present some limitations with non-monotonic problems. Some elementary effects may cancel each other out in these cases, thus providing low values of μ even when the factor is important. Campolongo *et al.* (2007) propose an alternative measure of the parameter μ, namely μ^* to avoid misleading results with non-monotonic models. For more information on the method of Morris we refer the reader to the original literature.

Analysis of Variance

The Design of Experiments presented in the previous section, is very often used in conjunction with the analysis technique called Analysis of Variance. Some authors refer to the DoE as to *statistical design of experiments* (Montgomery, 2001). In fact, they correctly indicate that the planning of the experiments with the DoE shall be done properly so that the data can be analysed using statistical methods, providing valid conclusions at the end. DoE and the type of analysis and results that can be obtained are closely related. Analysis of variance (ANOVA) is a technique that can be used to compute sensitivity analysis in the presence of discrete factors, *i.e.*, in the presence of samples coming from a DoE. Its main purpose is to partition the total variability detected during the simulations into the contribution of the various factors. This is a definition that is very much in line with the definition of global sensitivity analysis provided at the beginning of this section. For this reason, ANOVA can be considered a global sensitivity analysis method.

Given a set of observations of a mathematical model, the variance of the data can be computed with the well-known formula:

$$\hat{V} = \frac{\sum_{i=1}^{N} (Y_i - E(\mathbf{Y}))^2}{N-1} \tag{3.9}$$

where $E(\mathbf{Y})$ is the expected value, or mean value, of the model output. The numerator of this equation can also be called *sum of squares*. It is the sum of the deviations of all the observations from the mean, squared. In the following example we show how to compute the sensitivity analysis from a matrix design using ANOVA. Suppose having a model, with two factors that can assume three levels each. Suppose to simulate the model according to a full-factorial design, as shown in Table 3.6. There, the term $Y_{i.}$ indicates the sum of the outputs computed with factor A at constant levels. The term $Y_{.j}$, instead, indicates the sum of the outputs computed with factor B at constant levels. $Y_{..}$ is the sum of all the n outputs computed from the matrix experiment.

The total sum-of-squares can be computed as follows:

$$SS_T = \sum_{i=1}^{3}\sum_{j=1}^{3} Y_{ij}^2 - \frac{Y_{..}^2}{n} = (0)^2 + (2)^2 + (4)^2 + \ldots - \frac{(54)^2}{9} = 282 \tag{3.10}$$

The sum-of-squares due to factor A only, can be computed as follows:

		Levels of Factor B			
		1	2	3	$Y_{i.}$
Levels of Factor A	1	0	1	2	3
	2	2	6	10	18
	3	4	11	18	33
	$Y_{.j}$	6	18	30	54

Table 3.6 Matrix design, 2 factors at 3 levels. Performance of the model.

$$SS_A = \frac{1}{3}\sum_{i=1}^{3} Y_{i.}^2 - \frac{Y_{..}^2}{n} = (3)^2 + (9)^2 + (15)^2 + \ldots - \frac{(54)^2}{9} = 150 \tag{3.11}$$

The sum-of-squares due to factor B only, can be computed as follows:

$$SS_B = \frac{1}{3}\sum_{j=1}^{3} Y_{.j}^2 - \frac{Y_{..}^2}{n} = (6)^2 + (18)^2 + (30)^2 + \ldots - \frac{(54)^2}{9} = 96 \tag{3.12}$$

The sum-of-squares of the interaction AB, can be easily computed as the difference between the total sum-of-squares and the sum-of-squares of A and B. Therefore $SS_{AB} = 282-150-96 = 36$. A similar procedure for computing the sum-of-squares of the factors is discussed in Montgomery (2001). The difference here is that we deal with deterministic computer experiments, therefore we do not experience any sum-of-squares due to errors. We do not experience errors in the *measurements* of the performance.

The sensitivity indices are computed as the ratio between the sum-of-squares of the factors and the total sum-of-square:

$$S_A = \frac{SS_A}{SS_T} = 0.532 \quad S_B = \frac{SS_B}{SS_T} = 0.340 \quad S_{AB} = \frac{SS_{AB}}{SS_T} = 0.127 \tag{3.13}$$

As anticipated earlier, the amount of information that is available using ANOVA is directly related to the number of experiments performed with DoE. Full-factorial designs allows for determining main effects and factor interactions with no confounding, at the expenses of a large number of simulations required for increasing number of factors. *Resolution 5* factorial design allows for experimentation with no main effect or two-factor interaction confounded with any other main effect or two-factor interaction, although two-factor interactions can be confounded with higher-order interactions, *e.g.,* three-factor interactions ($A \times B \times C$). The subclass of fractional-factorial designs called *Resolution 4* requires half of the design points required by *Resolution 5*. It has no main effects confounded with any other main effect or with any two-factor interaction, but two-factor interactions can be confounded with each other and with higher-order interactions.

The confounding effect is typical of matrix designs, and can be explained as follows. Suppose we want to study the effect of the interaction of two of the parameters assigned to the columns of Table 3.2 on a performance parameter Y. For each experiment we obtain a certain value of the performance, so that at the end of all the experiments we will obtain a performance vector $\hat{Y} = [y_1, y_2, \ldots, y_n]$, with n equal to the number of experiments.

Looking at Table 3.2, we see that when factor C is at level 1 the combinations of A and B are A_1B_2 and A_2B_1, while when the factor C is at level 2 the combinations of A and B are A_1B_1 and A_2B_2. Thus, it is not possible to distinguish the effect of factor C from the interactions of factor A and factor B. The effect of factor C is confounded with the effect of the interaction $A \times B$. This also means that the third column of the OA of Table 3.2 does not represent factor C, it represents $A \times B$, instead.

One of the limitations of ANOVA is that it provides proper sensitivity indices only when a balanced matrix design is used for the simulations. The sensitivity indices will not be valid in general anymore if different sampling techniques, *e.g.*, random, would be used (Montgomery, 2001).

Alternative approaches

Some alternative approaches have been developed in the past years for the computation of sensitivity indices for computer models. A thorough discussion of all of them is beyond the scope of the present thesis. Helton and Davis (2002) present an analysis of the methods that are most widely used. The study provides results on the comparison of the performances of the following sampling-based procedures and measures of sensitivity: correlation coefficients, rank correlation coefficients, common means, common locations, common medians, statistical independence, standardized regression coefficients, partial correlation coefficients, standardized rank regression coefficients, partial rank correlation coefficients, stepwise regression analysis and scatter plots. Despite the limited computational effort required by most of the mentioned procedures, many of them provide local measures of sensitivity while many other only provide a qualitative indication on the ranking of the importance of the design variables in the determination of the output.

As stated previously, regression analysis is a popular method to assess the effects of input factors on performances. In particular, least-squares procedures are used to construct linear regression models in the following form

$$\hat{Y} = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i \tag{3.14}$$

where $\left[\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k\right]$ are the estimated regression coefficients. The validity of the results obtained with regression analysis is related to the fraction of the variation of Y that is accounted for by the regression model, $\hat{Y}$. When the variability *detected* using $\hat{Y}$ is lower than the variability of the data obtained with the simulations, lack-of-fit is present. More on the consequences of lack-of-fit for regression and sensitivity analysis will be discussed later, in Section 3.2.3. For now, it suffices to say that a linear regression model like Eq. (3.14), very often results to be poor in approximating the behavior of the model Y, thus the regression coefficients undergo the risk of being quantitatively meaningless and sometimes also qualitatively misleading. The same happens when the regression coefficients are expressed as Standardized Regression Coefficients (SRCs), *i.e.,* normalized coefficients, to eliminate the effect of the units in which Y and x_i are expressed, and the effect of the range of variation of the variables.

A general polynomial regression model, obtained by adding higher-order terms (including interactions) to the model of Eq. (3.14), will bring benefits in terms of a reduction of lack-of-fit, as will be demonstrated later in Section 3.2.3. In the following section the proposed global Regression-Based Sensitivity Analysis (RBSA) method is described. The main advantage of RBSA is that it is a computationally cheap method able to provide quantitative measures of the relative and absolute importance of the factors in the determination of the performance of an engineering system described by a mathematical model. It extends the concept of regression analysis, usually used for computing SRCs, coupling it with the idea of computing sensitivity based on the contribution of the factors to the variability of the model output.

3.2.3 Regression-Based Sensitivity Analysis method

If the design region of interest is not stretched out so much, a polynomial regression model is often sufficient to accurately describe the behavior of the engineering model under analysis.

This is true for typical models of engineering systems, even complex ones, especially when the source of complexity is represented by a large number of elements and their interrelated behavior rather than the mathematical models of every single component. However, also when the complexity is related to the highly non-linear and non-smooth behavior of the mathematical equations linking the design variables, considering a relatively small portion of the design space a polynomial regression model is still able to describe the system and explain most of the variability of the data.

The Regression-Based Sensitivity Analysis (RBSA) method proposed here, is general enough to be applicable to regression models of any order. However, the choice of the regression order depends on several aspects that will be discussed throughout this section. For ease of discussion the method will be explained using a second-order model as a reference:

$$\hat{Y} = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i + \sum_{i=1}^{k} \hat{\beta}_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \hat{\beta}_{ij} x_i x_j \tag{3.15}$$

Here, $\hat{\beta}_i$, $\hat{\beta}_{ii}$ and $\hat{\beta}_{ij}$ are the estimated regression coefficients that are calculated by fitting a response surface through the points sampled from the model.

The purpose of RBSA is to provide variance-based sensitivity information by decomposing the variance into the contributions of the individual design factors. The process is conceptually similar to what has been described already for the ANOVA and the method of Sobol'. However, we propose a faster approach if compared to the computation of the Sobol' sensitivity indices, without the limitation of ANOVA of being used only in the presence of DoE. Further, with RBSA we are also able to distinctively explain the interaction effect on the variance, unlike the other global sensitivity analysis methods presented in this thesis.

A review of the least-squares method

Let us consider a general mathematical model using a more compact notation:

$$Y = \beta_0 + \sum_{j=1}^{l} \beta_j x_j \tag{3.16}$$

where x_j represents any functional involving any of the design variables, for instance $x_j = x_2^2$ or $x_j = x_1 x_2$. In this case the coefficients β_j are the true (unknown) ones, which will be estimated by the coefficients $\hat{\beta}_j$.

Using a least-squares method to estimate the l regression coefficients of the model, at least $N \geq l$ samples are needed. The least-squares method computes an estimation of the regression coefficients minimizing the sum of squares of the errors ϵ_i:

$$Y_i = \beta_0 + \textstyle\sum_{j=1}^{l} \beta_j x_j + \epsilon_i, \qquad i = 1, 2, \ldots, N \tag{3.17}$$

In Eq. (3.17), Y_i represents the observed response for the i^{th} design-variable set $\mathbf{X}_i$. If the model in Eq. (3.17) is rewritten in matrix form, *i.e.*, $\mathbf{Y} = \mathbf{X}\beta + \epsilon$ the least-squares method is easier to present and to implement. Here we have used the following definitions:

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1l} \\ 1 & x_{21} & x_{22} & \cdots & x_{2l} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nl} \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_l \end{bmatrix}, \epsilon = \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

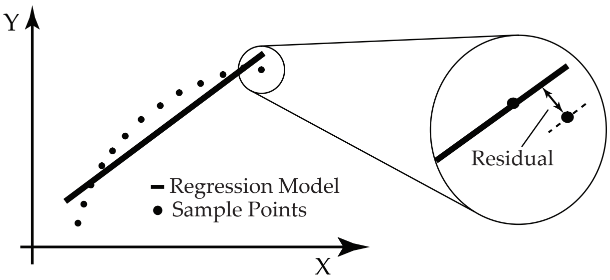


Figure 3.8 Regression error for mathematical models.

The least-squares estimate of the regression coefficients is computed as follows:

$$\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} \tag{3.18}$$

The utilization of a decomposition method, such as QR factorization or singular value decomposition (SVD), to work with the matrix $\mathbf{X}^T\mathbf{X}$ in Eq. (3.18) is highly recommended. That matrix may be close to be singular in some cases, also said *ill-conditioned*, and these factorization or decomposition methods are considered numerically stable also with ill-conditioned matrices.

The least-squares model is therefore represented by the following relationship:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} \tag{3.19}$$

As stated already, the least-squares method provides an estimate of the regression coefficients that minimizes the error between the regression model and the observations. In general, this error may have two main sources. The first source of error is the lack-of-fit of the regression model, when the model for which the regression has been computed does not have enough parameters to explain the real model. This happens, for instance, when a linear model is used to fit a nonlinear one (Figure 3.8). The difference between each sample point and the output computed using the regression model is called *residual*. The second source of error is the measurement performed to collect the sample; in this case it is called *pure error*, or *measurement error*. Since in this case regression analysis is applied to deterministic mathematical models, the pure error is zero. Indeed, given a certain combination of design variables values, the response will always be the same.

Decomposition of the variance

As shown already in Eq. (3.9), the total sum of squares of a set of observations of a mathematical model can be expressed as follows:

$$SS_T = \sum_{i=1}^{N} \left(Y_i - E(\mathbf{Y})\right)^2 \tag{3.20}$$

The sum of squares of the regression only, instead, can be computed as follows:

$$SS_R = \sum_{i=1}^{N} \left(\hat{Y}_i - E(\mathbf{Y})\right)^2 \tag{3.21}$$

SS_R represents the portion of the total variability that can be explained by the regression model. For instance, a linear regression model cannot explain all the variability of a nonlinear one, as schematically shown in Figure 3.8. In case the regression model perfectly fits the data, which does not always mean that the regression model perfectly matches the real one as will be explained later, then $SS_T = SS_R$. When residuals are present the portion of the total variability not explained by the regression model can be computed in the form of the error sum of squares, SS_E:

$$SS_E = \sum_{i=1}^{N} \left(Y_i - \hat{Y}_i\right)^2 \tag{3.22}$$

To obtain the sensitivity indices of all the factors that contribute to the total variability of the regression model, the regression sum of squares SS_R should be partitioned in its components. The main idea is to associate a sensitivity index to the additional variability calculated when a factor is added to the regression model. To do so, a matrix notation for the sum of squares is now introduced. Combining Eqs. (3.18) and (3.19), the regression model can be expressed as follows:

$$\hat{\mathbf{Y}} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} = \mathbf{H}\mathbf{Y} \tag{3.23}$$

The matrix $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$ is called the *hat* matrix. It transforms the vector of the observed responses $\mathbf{Y}$ into the vector of the fitted values $\hat{\mathbf{Y}}$. Using the hat matrix, the total, regression and error sums of squares can be expressed with the following relationships (Kuri and Cornell, 1996):

$$SS_T = \mathbf{Y}^T\left[\mathbf{I} - \frac{1}{N}\mathbf{J}\right]\mathbf{Y} \qquad SS_R = \mathbf{Y}^T\left[\mathbf{H} - \frac{1}{N}\mathbf{J}\right]\mathbf{Y} \qquad SS_E = \mathbf{Y}^T\left[\mathbf{I} - \mathbf{H}\right]\mathbf{Y} \tag{3.24}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix, and $\mathbf{J}$ is an $N \times N$ matrix of ones.

In literature there are three methods that are most widely used to obtain the variance decomposition of Eq. (3.15), namely the sequential sum-of-squares, the classical sum-of-squares and the partial sum-of-squares decomposition (Draper and Smith, 1998). As one of the possible models that Eq. (3.15) can describe, let us consider the following, with 3 factors:

$$\begin{aligned}\hat{Y} =&\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_{11} x_1^2 + \hat{\beta}_{22} x_2^2 + \hat{\beta}_{33} x_3^2 + \\ &+ \hat{\beta}_{12} x_1 x_2 + \hat{\beta}_{13} x_1 x_3 + \hat{\beta}_{23} x_2 x_3\end{aligned} \tag{3.25}$$

In the following discussion, $SS(Y_{x_1})$ represents the sum of squares associated with the model computed with only the factor x_1 (*i.e.*, $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1$). $SS(x_2|\, Y_{x_1})$ represents the sum of squares associated with a regression model where x_2 is added to the model given that x_1 is already present, it will also be indicated as $SS(x_2)$ since it is the sum of squares associated with x_2 only. This indicates the additional variability explained by adding x_2 to the model.

The sequential sum-of-squares decomposition, or Type-I sum of squares, is used for the stepwise regression analysis for computing sensitivity indices, see for instance (Helton and Davis, 2002). The Type-I sum of squares decomposition is influenced by the order in which the parameters are added to the model, thus it does not provide a unique sensitivity index for each factor. The Type-II sum-of-squares decomposition, or classical sum of squares, indicates the change in the variability explained by the regression model due to adding an extra term to the model, given that all other terms have been added except for the terms that contain the effect under test. For instance, the sum of squares of factor x_3, with x_1 and x_2 in the model, with all interactions (two and three factor-interactions) can be computed as follows:

$$SS(x_3) = SS(x_3|\, Y_{x_1 x_2 x_{12}}) = SS(Y_{x_1 x_2 x_3 x_{12} x_{13} x_{23}}) - SS(Y_{x_1 x_2 x_{12}}) \tag{3.26}$$

The third method computes the contribution to the variability explained by the regression model due to adding an extra term, given that all other terms are already in the model, including the interactions and higher-order factors involving the term under investigation. The sum of squares of Type-III for the factor x_3 of the model in Eq. (3.25) would be as follows:

$$SS(x_3) = SS(x_3|\,Y_{x_1x_2x_{12}}) = SS(Y_{x_1x_2x_3x_{12}x_{13}x_{23}}) - SS(Y_{x_1x_2x_{12}x_{13}x_{23}}) \tag{3.27}$$

In Eqs. (3.26) and (3.27) the term $Y_{(\cdot)}$ represents the regression model with all the factors and interactions indicated by the subscripts. Given the sum of squares associated with every factor of the regression model, the sensitivity indices can be computed with a relationship that is similar to that presented in Eqs. (3.2) and (3.3):

$$S_{x_i} = \frac{SS(x_i)}{SS_R + SS_E} \tag{3.28}$$

Indeed, the sensitivity measures computed using Eq. (3.28) can be interpreted in terms of the first- and total-order sensitivity indices. When $SS(x_i)$ is computed with the Type-II decomposition (Eq. (3.26)), it describes the contribution of a factor considering, simultaneously, all the interactions and higher-order effects involving it. Thus it provides information on the total effect of that factor. Using the Type-III decomposition to compute $SS(x_i)$ (Eq. (3.27)), instead, we obtain the contribution of each term of the polynomial regression model (*e.g.*, x_1, x_1^2, or x_1x_2) to the total variability computed with the regression model. This allows to compute the contribution to the variance of individual effects in a way that is not allowed with other approaches discussed in the previous section. And this is possible with no additional simulations.

In the case of RBSA we inherit the terminology from ANOVA calling the effects of the individual factors (computed with Eqs. (3.28) and (3.27)) *first-order* effects. In these cases it would be more appropriate calling them *individual-order* effects since they refer to individual terms in the regression model, therefore also the quadratic (*e.g.*, x_1^2) or interaction (*e.g.*, x_1x_2) terms. This is an added advantage of RBSA. With Sobol' method, or FAST, it is only possible to compute the actual first-order sensitivity indices (*e.g.*, sensitivity indices of x_1, x_2, *etc.*). The presence of interactions or higher-order terms may only be detected by comparing the first- and total-order sensitivity indices. The quantitative knowledge of the actual contribution of an interaction between factors, or of an higher-order term, computed with RBSA is certainly more insightful than the understanding of the fact that such effects may most likely be present in the model.

The algorithm for RBSA

The RBSA algorithm begins with an educated hypothesis on the behavior of the model in the design region of interest. Eq. (3.15) could be used, for instance, as an initial assumption. However, if later in the process *inacceptable* lack-of-fit is detected, this assumption could be reviewed by modifying the regression model adding cubic (*e.g*, x_i^3) or higher-order interaction terms (*e.g*, $x_ix_jx_k$), for instance. For the moment, let us use the model presented in Eq. (3.15).

The second step consists in the creation of an input sample matrix $\mathbf{M}$, made of *k* columns (the number of design variables taken into account) and *N* rows. Each row represents a design vector with a value for each design variable; each row represents a sample point. The sample size *N* shall be larger than the number of regression coefficients to estimate. For instance, $N > 2k + k(k-1)/2$ samples are needed for the regression analysis on the model of Eq. (3.15).

$$\mathbf{M} = \begin{pmatrix} x^{11} & x^{12} & \cdots & x^{1k} \\ x^{21} & x^{22} & \cdots & x^{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x^{N1} & x^{N2} & \cdots & x^{Nk} \end{pmatrix} \tag{3.29}$$

The output vector $\mathbf{Y}$ is obtained by executing the mathematical model with the rows of $\mathbf{M}$ as inputs.

The next step is to build the matrix $\mathbf{X}$ that will be used to compute the sum of squares and the sensitivity indices. The construction of $\mathbf{X}$, and the methodology to compute the sensitivity indices, will only be presented specifically for the model in Eq. (3.15). The derivation for regression models of different orders is similar. First, the two matrices $\mathbf{R_1}$ and $\mathbf{R_2}$ with dimensions $N \times k(k-1)/2$ shall be obtained by a re-arrangement of the columns of $\mathbf{M}$.

$$\mathbf{R}_1 = \begin{pmatrix} \mathbf{M}_{(1,1)} & \dots & \mathbf{M}_{(1,1)} & \mathbf{M}_{(1,2)} & \dots & \mathbf{M}_{(1,2)} & \dots & \mathbf{M}_{(1,k\text{-}1)} \\ \mathbf{M}_{(2,1)} & \dots & \mathbf{M}_{(2,1)} & \mathbf{M}_{(2,2)} & \dots & \mathbf{M}_{(2,2)} & \dots & \mathbf{M}_{(2,k\text{-}1)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \underbrace{\mathbf{M}_{(N,1)} \quad \dots \quad \mathbf{M}_{(N,1)}}_{k\text{-}1} & & & \underbrace{\mathbf{M}_{(N,2)} \quad \dots \quad \mathbf{M}_{(N,2)}}_{k\text{-}2} & & & \dots & \underbrace{\mathbf{M}_{(N,k\text{-}1)}}_{1} \end{pmatrix}$$

$$\mathbf{R}_2 = \begin{pmatrix} \mathbf{M}_{(1,2)} & \dots & \mathbf{M}_{(1,k)} & \mathbf{M}_{(1,3)} & \dots & \mathbf{M}_{(1,k)} & \dots & \mathbf{M}_{(1,k)} \\ \mathbf{M}_{(2,2)} & \dots & \mathbf{M}_{(2,k)} & \mathbf{M}_{(2,3)} & \dots & \mathbf{M}_{(2,k)} & \dots & \mathbf{M}_{(2,k)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \underbrace{\mathbf{M}_{(N,2)} \quad \dots \quad \mathbf{M}_{(N,k)}}_{k\text{-}1} & & & \underbrace{\mathbf{M}_{(N,3)} \quad \dots \quad \mathbf{M}_{(N,k)}}_{k\text{-}2} & & & \dots & \underbrace{\mathbf{M}_{(N,k)}}_{1} \end{pmatrix}$$

The matrices $\mathbf{R_1}$ and $\mathbf{R_2}$ can be visualized in blocks. The first $k-1$ columns of $\mathbf{R_1}$ are $k-1$ replications of the first column of $\mathbf{M}$. The second block of $k-2$ columns is made of the replication of the second column of $\mathbf{M}$, and so on until the last-but-one column of $\mathbf{M}$, which appears only once. $\mathbf{R_2}$ is built with a different approach, but the visualization by blocks is still possible. The first $k-1$ columns of $\mathbf{R_2}$ are replications of the second to last column of $\mathbf{M}$. The second block of $k-2$ columns consists of the third to last column of $\mathbf{M}$, and so on until the last column of $\mathbf{M}$, which appears only once. Therefore, the elements of $\mathbf{R_1}$ and $\mathbf{R_2}$ can be interpreted as follows: $\mathbf{M}_{(\mathbf{1},\mathbf{1})} = \mathbf{x}^{\mathbf{11}}$, $\mathbf{M}_{(\mathbf{1},\mathbf{k})} = \mathbf{x}^{\mathbf{1k}}$, and $\mathbf{M}_{(\mathbf{N},\mathbf{k})} = \mathbf{x}^{\mathbf{Nk}}$.

The coefficient-wise (*i.e.*, Hadamart, indicated by $\circ$) product of $\mathbf{R_1}$ and $\mathbf{R_2}$ gives the matrix $\mathbf{R}$:

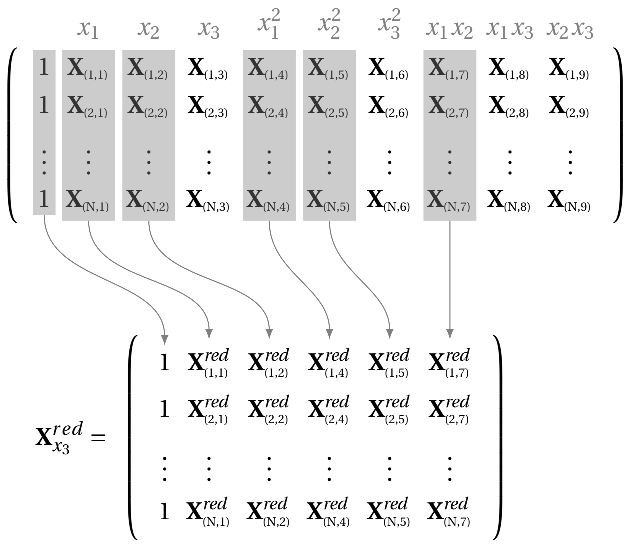
$$\mathbf{R} = \mathbf{R_1} \circ \mathbf{R_2}$$

Each element of $\mathbf{R}$ is obtained by multiplying the corresponding elements of $\mathbf{R_1}$ and $\mathbf{R_2}$, *i.e.*, $\mathbf{R}_{ij} = \mathbf{R}_{1(ij)} \times \mathbf{R}_{2(ij)}$. $\mathbf{R}$ will be used to compute the interaction effects for the sensitivity indices. The matrix $\mathbf{X}$ to be used for the regression analysis is obtained by re-arranging the columns of $\mathbf{M}$ and $\mathbf{R}$:

$$\mathbf{X} = \begin{pmatrix} 1 & \mathbf{M}_{(1,1)} & \dots & \mathbf{M}_{(1,k)} & \left(\mathbf{M}_{(1,1)}\right)^2 & \dots & \left(\mathbf{M}_{(1,k)}\right)^2 & \mathbf{R}_{(1,1)} & \dots & \mathbf{R}_{(1,k(k\text{-}1)/2)} \\ 1 & \mathbf{M}_{(2,1)} & \dots & \mathbf{M}_{(2,k)} & \left(\mathbf{M}_{(2,1)}\right)^2 & \dots & \left(\mathbf{M}_{(2,k)}\right)^2 & \mathbf{R}_{(2,1)} & \dots & \mathbf{R}_{(2,k(k\text{-}1)/2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{M}_{(N,1)} & \dots & \mathbf{M}_{(N,k)} & \left(\mathbf{M}_{(N,1)}\right)^2 & \dots & \left(\mathbf{M}_{(N,k)}\right)^2 & \mathbf{R}_{(N,1)} & \dots & \mathbf{R}_{(N,k(k\text{-}1)/2)} \end{pmatrix}$$

Once $\mathbf{X}$ is available, SS_T, SS_R, and SS_E can be computed using the *hat* matrix $\mathbf{H}$ and the relationships presented in Eq. (3.24).

Total-order sensitivity indices are computed for every design variable in the model. For each of the k design variables a reduced version of $\mathbf{X}$, namely $\mathbf{X}^{red}$, needs to be build. $\mathbf{X}^{red}$ is obtained by removing the columns of $\mathbf{X}$ that are related to all terms involving the related design variable for which the total sensitivity index is computed. For instance, consider the model of Eq. (3.15) with 3 design variables. The construction of $\mathbf{X}^{red}_{x_3}$ for the variable x_3 would be as shown below, by removing the *white* columns:



Using $\mathbf{X}^{red}$ the regression sum of squares, SS_R^{red}, can be computed using Equations (3.23) and (3.24):

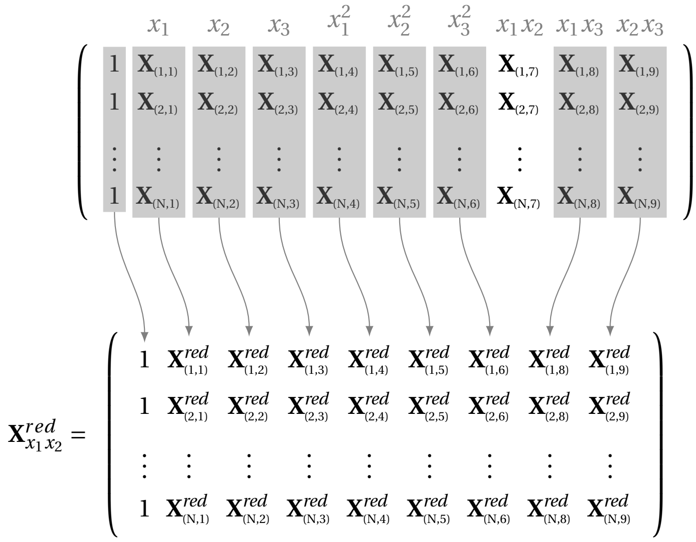
$$\mathbf{H}^{red} = \mathbf{X}^{red}\left(\mathbf{X}^{Tred}\mathbf{X}^{red}\right)^{-1}\mathbf{X}^{T,red} \tag{3.30}$$

$$SS_R^{red} = \mathbf{Y}^T\left[\mathbf{H}^{red} - \frac{1}{N}\mathbf{J}\right]\mathbf{Y} \tag{3.31}$$

SS_R^{red} of a certain design variable x_i indicates the variability that the model without the contribution of the terms that involve x_i is able to explain. The difference between the regression sum of squares computed with Equations (3.23) and (3.24) using $\mathbf{X}$ and the *reduced* regression sum of squares indicates the overall contribution of the design variable x_i to the variability detected by the full model (*i.e.*, Type-II sum of square). Thus, a total-order sensitivity index may be computed as follows:

$$S_{Ti} = \frac{SS_R - SS_R^{red}}{SS_R + SS_E} \tag{3.32}$$

First-order sensitivity indices can be obtained in a very similar fashion for each term of the model, including interactions and higher-order terms. For each term of the model $\mathbf{X}^{red}$ needs to be build, as in the previous case. $\mathbf{X}^{red}$ is again a *reduced version* of the matrix $\mathbf{X}$, but in this case it is obtained by removing only the column of $\mathbf{X}$ that is related to the term of interest. For instance, the construction of $\mathbf{X}^{red}_{x_1x_2}$ for the interaction term x_1x_2 would be as follows:



Using $\mathbf{X}^{red}$ the regression sum of squares SS_R^{red} is again obtained with Equations (3.30) and (3.31). The first-order sensitivity index for each term of the model may be computed as presented before:

$$S_i = \frac{SS_R - SS_R^{red}}{SS_R + SS_E} \tag{3.33}$$

In this case the difference between the regression sum of squares computed with Equations (3.23) and (3.24) using $\mathbf{X}$ and the *reduced* regression sum of squares indicates the Type-III sum of squares indicated in Eq. (3.27).

This approach to compute the sensitivity indices, based on regression analysis, is probably less intuitive than those presented in the previous sections, but it provides some advantages. First of all, the number of model evaluations, that is usually the most resource-consuming part of the analysis, is reduced (a numerical comparison is provided in Section 3.2.5). Second, the RBSA provides quantitative information (rather than qualitative as most of the screening or sample-based SA methods) also on the effects of interactions and higher-order terms on the performance of interest (rather than only first-order and total sensitivity indices as the method of Sobol′ or FAST). The fact that higher-order models are implemented, rather than linear models only, allows to explain a larger part of variability when compared to the SRCs method, for instance. One possible drawback of RBSA is that the validity of the results depends on the lack-of-fit of the regression model with respect to the sample data. Indeed, special attention must be paid to the ratio between the regression and the total sum of squares. If SS_R is close to SS_T, then the regression model is able to account for a large part of the output variance, and as a consequence the sensitivity indices are meaningful measures. If this is not the case, lack-of-fit is present meaning that important terms are missing from the initially assumed regression model. Lack-of-fit is important to decide whether to proceed with sensitivity analysis anyway or to modify the initial assumption and increase the order of the regression model by adding extra terms, *e.g.,* higher-order terms like cubic or higher-order interactions.

Testing for model adequacy

The quality of the regression model, used to fit the sample data is fundamental for the RBSA. As mentioned already, the sensitivity indices computed using RBSA indicate the contribution of the factors to the variance of the response. This is based on the decomposition of the regression sum of squares only, not taking into account the possible error sum of squares, arising

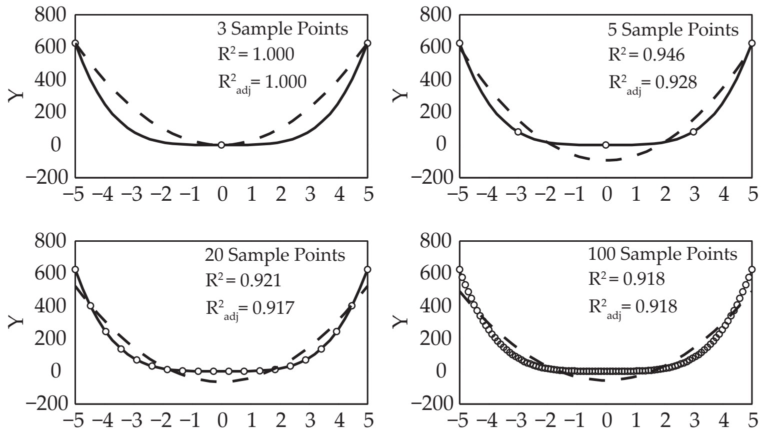


Figure 3.9 Effect of the number of samples on the quality of the model.

when lack-of-fit is present. The presence of lack-of-fit could be related to the fact that important terms have been neglected, or simply that a polynomial regression model is not entirely adequate to reproduce the relationships between the design variables, *e.g.*, in case of exponential or sinusoidal effects. Testing for model adequacy is a fundamental step since it is a means to validate the results of the sensitivity analysis, allowing to mitigate the effect of the lack-of-fit on the sensitivity indices by an iterative approach (see also Section 3.2.3).

The coefficient of determination, R^2, allows to detect the fraction of the model output variance accounted for by the regression model:

$$R^2 = \frac{\sum_{i=1}^{N}\left(\hat{Y}_i - \bar{Y}\right)^2}{\sum_{i=1}^{N}\left(Y_i - \bar{Y}\right)^2} = \frac{SS_R}{SS_T} \qquad 0 \leq R^2 \leq 1 \tag{3.34}$$

Very often the *adjusted* coefficient of determination, R^2_{adj}, is used instead of R^2:

$$R^2_{adj} = 1 - \left(\frac{N-1}{N-(l+1)}\right)\left(1 - R^2\right) \qquad 0 \leq R^2_{adj} \leq 1 \tag{3.35}$$

where l indicates the total number of regressors in the polynomial model (without the constant term β_0).

The motivation for introducing R^2_{adj} is that R^2 increases when terms are added to the model, even if those terms do not provide a relevant contribution to the variance (Kuri and Cornell, 1996). Thus, it cannot be used as a meaningful comparison of regression models with a different numbers of terms. R^2_{adj} does not suffer from this phenomenon. It only increases if relevant terms are added to the model, and when non-relevant terms are considered it decreases, thus providing more precise information on the general validity of the regression model (Kuri and Cornell, 1996).

Values of R^2_{adj} larger than 0.9 usually suggest a good fit of the data. The extreme case in which R^2_{adj} is equal to one, indicates that the regression model is able to account for all the variability of the model output, but this does not always mean that the regression model perfectly matches the true one in all points of the design region, as shown in the example of Figure 3.9.

A fourth order model $Y = x^4$ is used as the true model (*i.e.*, the model from which the samples are collected) in the design region spanned by $x \in [-5, 5]$. The trends of the true model are represented by the thick lines of Figure 3.9, while the dashed lines represent the behavior of a second-order regression model fitted using 3, 5, 20, and 100 sample points, indicated by the circles. Three sample points are sufficient to estimate the three coefficients needed for a

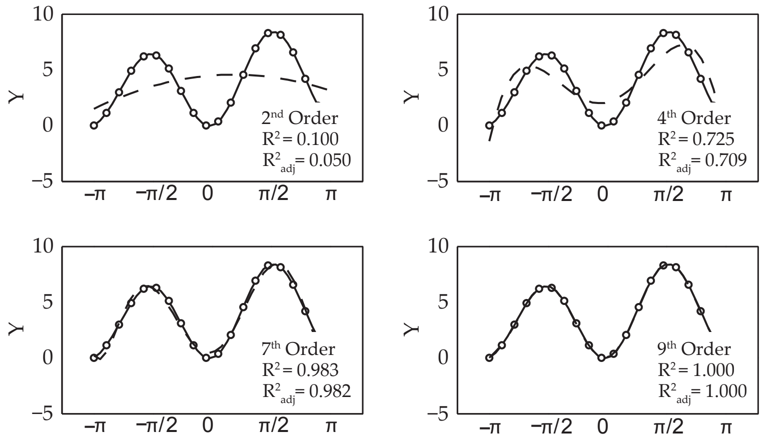


Figure 3.10 Effect of the order of the regression model on the quality of the regression.

second-order model of only one variable, but no degrees of freedom are left for the error, thus producing the result of the upper-left graph of Figure 3.9. In that case, both coefficients of determination indicate that the regression model perfectly fits the data. Indeed, the dashed line passes through all the sample points. However, the second-order regression model does not properly match the true one. With an increasing sample size, the estimation of the regression error becomes more and more reliable, while the quality of the regression model improves, but not significantly already after five sample points.

Increasing the sample size, is in general only partially beneficial to the reduction of the lack-of-fit. Increasing the order of the regression model could substantially help, instead, for a better reconstruction of the underlying relationships between the design variables. For instance, in case of the previous example, a fourth-order model with only 6 samples could already eliminate the lack-of-fit, instead of using up to 100 samples with a quadratic model. However, the solution may not always be so straightforward. In Figure 3.10 the results from the regression analysis of the model $Y = \sin(x) + 7\sin(x)^2 + 0.1x^3\sin(x)$, with $x \in [-\pi, \pi]$, is presented. The order of the regression models is increased from 2, top-left, to 9, bottom-right, with a constant number of samples, *i.e.*, 20. As in the previous example, the true model is represented by the thick lines, the dashed lines represent the behavior of the model regressed on the sample data, indicated by the circles. The results in Figure 3.10 demonstrate that there may be cases in which the quality of the analysis can be improved by increasing the order of the regression model, rather than adding samples.

Concluding, there is not a general and guaranteed approach to identify lack-of-fit. It is advised, though, to build the regression models with a number of samples that exceeds the actual number of terms needed to build the model. In this way, more degrees of freedom for the estimation of the error are provided, avoiding to obtain misleading values for R^2 or R^2_{adj}, as shown in Figure 3.9. The discussion on the model adequacy provided in this section is limited to the implementation needed for the proposed RBSA methodology. For a more complete analysis the interested readers may consider the books of Draper and Smith (1998) and Kuri and Cornell (1996).

The iterative approach to RBSA

In Table 3.7 a list of suggested regression models of increasing order, with the minimum number of samples required to compute all the coefficients, is presented. This particular choice is merely indicative, it shall be considered as an example to explain the iterative approach to RBSA. The minimum number of samples for every regression model is equal to the number of

Model order	Regression model	Minimum number of samples
1	$Y_1 = \beta_0 + \sum \beta_i x_i$	$2 + k$
$1^1/_2$	$Y_{1/_2} = Y_1 + \sum \beta_{ij} x_i x_j$	$2 + k + \binom{k}{2}$
2	$Y_2 = Y_{1/_2} + \sum \beta_{ii} x_i^2$	$2 + 2k + \binom{k}{2}$
3	$Y_3 = Y_2 + \sum \beta_{iii} x_i^3 + \sum \beta_{ijw} x_i x_j x_w$	$2 + 3k + \binom{k}{2} + \binom{k}{3}$
4	$Y_4 = Y_3 + \sum \beta_{i4} x_i^4$	$2 + 4k + \binom{k}{2} + \binom{k}{3}$
5	$Y_5 = Y_4 + \sum \beta_{i5} x_i^5$	$2 + 5k + \binom{k}{2} + \binom{k}{3}$
6	$Y_6 = Y_5 + \sum \beta_{i6} x_i^6$	$2 + 6k + \binom{k}{2} + \binom{k}{3}$
7	$Y_7 = Y_6 + \sum \beta_{i7} x_i^7$	$2 + 7k + \binom{k}{2} + \binom{k}{3}$

Table 3.7 Suggested regression models for the iterative procedure. k is the number of design variables.

factors present in the model plus extra sample points equal to the number of variables of the model. The addition of at least k extra sample points is suggested to prevent the phenomenon described in Figure 3.9. The decision to modify the initial assumptions on the regression model depends on the adequacy of the current one, determined by R^2_{adj}.

At the beginning of the process, the minimum number of samples for fitting a linear model is collected. If R^2_{adj} is lower than a certain threshold value, *e.g.*, 0.9, the sample size is increased (by a multiple of k, for instance), and R^2_{adj} is computed again. During the iterations, each time that the number of samples is sufficient to evaluate the next higher-order regression model, see Table 3.7, also R^2_{adj} of that model is tested. This procedure is repeated until at least one regression model provides satisfactory results, or if for increasing regression-model order and increasing sample size the value of R^2_{adj} does not significantly improve. RBSA is then computed with the regression model having the best performance in terms of R^2_{adj}.

At first sight, this iterative approach may seem inefficient, due to the re-sampling of the design region. However, with particular care on the sampling technique, the samples taken in one iteration can be re-used also for the subsequent one, as discussed already in Section 3.1, and as demonstrated in the following example.

3.2.4 Test case: the communication and power subsystems, sensitivity analysis

The mathematical models used for this example and all the assumptions needed to limit the analysis are described in Appendix A. The main purpose of the discussion in this subsection is to better explain the utilization of the iterative RBSA method and to show, step-by-step, its implementation starting from sampling the design space with the Mixed-Hypercube approach to the computation of the sensitivity indices. As described in Appendix A, we set up an analysis of the communication and power subsystems using five design variables, see Table 3.8, two performance indicators (namely, the down-link margin and the total mass of the two subsystems) and one constraint represented by the down-link margin itself demanded to be larger than 4 dB.

The three discrete variables give rise to a three-dimensional factorial design with 12 possible factor-levels combinations in total. For each combination of discrete-variable levels the RBSA routine initially generates 7 (2+5, see Table 3.7) sample points using a Sobol' sequence. With 7 sample points, a linear regression model is computed for both performances. In Figure 3.11 we present the results of the linear regression on the continuous variables, with the discrete variables at the lowest level, *i.e.*, a Horn antenna, a Silicon-cell solar array, and a TWTA transmitter.

The coefficients of determination indicate that the subsystems mass is well represented by a linear relationship. The down-link margin, instead, could be better approximated using a higher-order model. The decision whether to *re-sample* or continue with the RBSA shall be based on the value of R^2_{adj}. In this case a threshold of $R^2_{adj} = 0.95$ is used, which induces the

Design Variables		Code	Intervals Min	Max	Levels
Output RF power	[W]	A	1	50	–
Antenna diameter	[m]	B	0.05	1	–
Type of Antenna	[-]	C	1	2	2
Type of Solar Array	[-]	D	1	3	3
Type of Transmitter	[-]	E	1	2	2

Table 3.8 Settings of the design variables for the design of the communication and power subsystems.

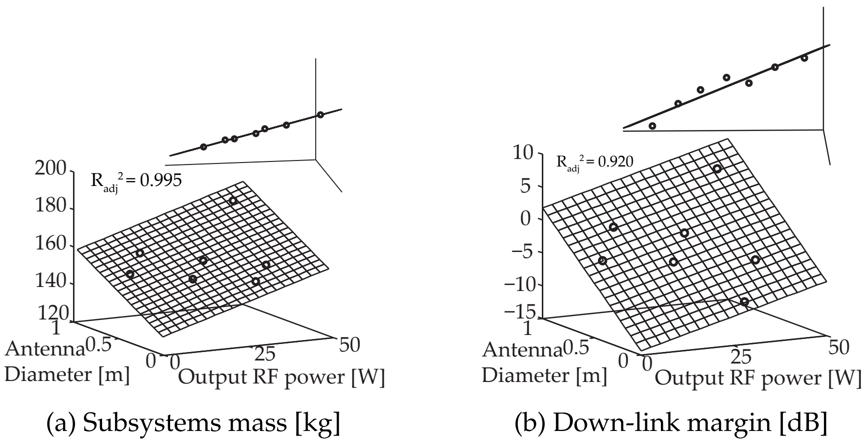
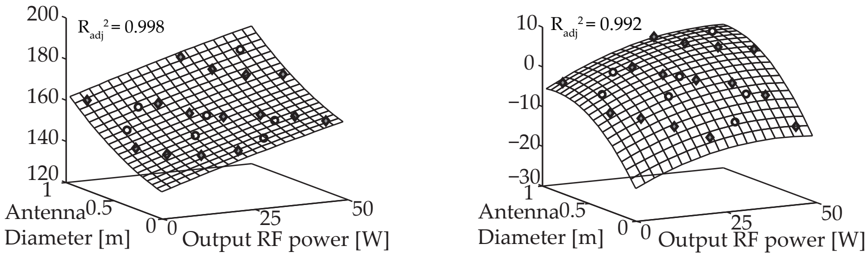


Figure 3.11 Linear regression of the performances using 7 sample points. On top, we show the side views of the surfaces.

iterative RBSA to add sample points to the analysis. A linear regression model with interaction terms is not sufficient to reach the threshold, which can be met only with a quadratic regression model. In this case we increment the number of sample points from 7 to 22 (2+10+10, see Table 3.7). The indications of Table 3.7 are only for the minimum number of sample points. The actual number of sample points to use, for each model order, is up to the user of RBSA. It depends on many aspects, including the type of model that one is using. In Figure 3.12, the results of the quadratic regression on the performances are shown. The white circles represent the previous 7 sample points while the gray diamonds are the additional samples produced for the subsequent two iterations of the RBSA, *i.e.*, the linear model with interaction effects and the quadratic model. As stated in the previous section, and confirmed by the results presented in Figure 3.12, the Sobol′ sequence provides additional samples that do not overlap with the previously generated ones placing them at a large distance between each other. This characteristic is very relevant for an efficient re-utilization of the previous model simulations, that is the time-consuming part of the method. The coefficients of determination, in this case, confirm that a quadratic regression model is suitable for representing the variability of the performances in the design region of interest, even in the case of the down-link margin.

In Figure 3.13 we show the difference between the linear and quadratic regression model for the Down-link margin.

With the same process described for the first combination of discrete design variables, the mathematical models of the communication and power subsystems are executed on the sam-



(a) Subsystems mass [kg] (b) Down-link margin [dB]

Figure 3.12 Quadratic regression of the performances. ○ Previous sample points. ◇ Additional sample points.

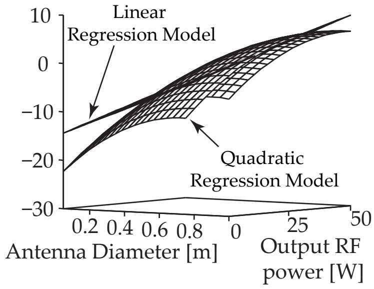


Figure 3.13 Comparison between the linear and quadratic regression model for the down-link margin performance [dB].

ple points for the other discrete-variable combinations. Then, with the RBSA the sensitivity indices can be estimated, using the relationship of Eqs. (3.32) and (3.33). In Figures 3.14 and 3.15 the *total order* and the *first order* sensitivity indices for the subsystems mass and the down-link margin performances, respectively, are shown. The bars represent the sensitivity indices, *i.e.*, the contribution of the factors indicated on the horizontal axis of the graphs, their interactions (when the product of two factors is indicated), and their quadratic effects (when the product of the factor by itself is indicated) to the variability of the performances. A sensitivity index equal to 0.2, for instance, indicates a contribution of that factor to the variance of the performance of interest equal to 20%. The contribution of all other effects that are not explicitly shown in the bar plots, including the regression error, are encapsulated in the bars named *Other*.

The influence of the *Antenna Diameter* (B) and the *Output RF Power* (A) on both the per-

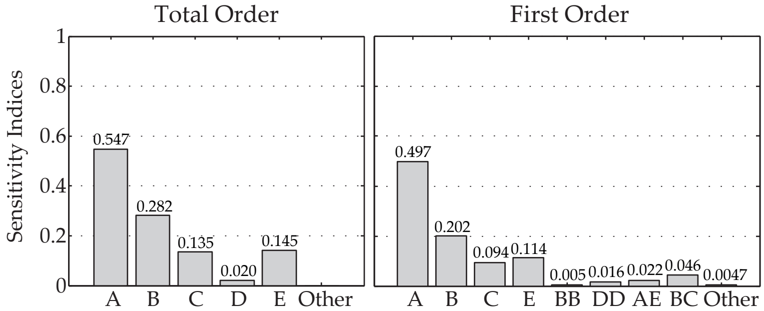


Figure 3.14 Sensitivity indices obtained with the Regression Based Sensitivity Analysis. Sub-systems mass.