Universitá degli Studi di Padova

Andrea Scaramuzzi

Robust Design Optimisation of S-Ducts under Uncertainties

Dipartimento di Ingegneria Industriale

MSc in Aerospace Engineering

MSc Thesis

Academic Year: 2018-2019

Supervisors:

Prof. Ernesto Benini

Dr. Timoleon Kipouros

April 2020

Universitá degli Studi di Padova

MSc in Aerospace Engineering

MSc Thesis

Academic Year: 2018-2019

Andrea Scaramuzzi

Robust Design Optimisation of S-Ducts under Uncertainties

Supervisors:

Prof. Ernesto Benini

Dr. Timoleon Kipouros

April 2020

UNIVERSITÁ DEGLI STUDI DI PADOVA

Abstract

Dipartimento di Ingegneria Industriale

MSc in Aerospace Engineering

Robust Design Optimisation of S-Ducts under Uncertainties

by Andrea Scaramuzzi

Thanks to the increasing computational power available, Computational Fluid Dynamics (CFD) analysis and the Multi-objective optimisation methods have become increasingly important and widespread in the design engineering world in the recent period. In fact, these calculation and optimisation tools have allowed a tremendous improvement in the performances of aerodynamic aircraft elements such as wings or engine intakes. Over the past years, robust optimisations have become very popular. The aim of this new type of optimisation is to consider the sensitivity of the output results to small variations in the operating conditions or in the manufacturing constraints. In order to study such sensitivities, it is necessary an accurate and efficient method to quantify uncertainty in physical processes. In this context, the study of the behaviour of the flow in an S-duct intake has been conducted following the previous studies performed by Rigobello [1], D'Ambros [2], Dal Magro [3]. In particular, the purpose of all the above-mentioned research has been to optimize the shape of the duct in order to improve its performances. This project is the natural prosecution of these works. For this reason, in this thesis, the main objective has been to extend the robust design optimisation for S-ducts with more uncertainties input (Inlet velocity, flux deviation, bending angle) and output (pressure recovery and swirl). The study of uncertainties is a very complex topic and in literature it is possible to find different strategies to calculate their propagation to the final output results. Two different non-intrusive Polynomial Chaos techniques have been chosen: the non-intrusive point collocation and the non-intrusive spectral projection. The results of these two techniques have been compared to each other. For the first time, a manufacturing uncertainty (bending angle) has been introduced and its influence to the performances was analysed. Moreover, to analyse more S-ducts more quickly, a robust optimisation cycle was

implemented with a machine learning state-of-the-art technique developed by Longato [4].

Sommario

In questa tesi di laurea magistrale, cinque cicli di ottimizzazione robusta, più un sesto ciclo di ottimizzazione supportato da Machine Learning, sono stati sviluppati con lo scopo di migliorare le performance aerodinamiche di una presa particolare dinamica curvilinea chiamata S-duct. Il presente lavoro è stato supervisionato dal Professor Timoleon Kipouros dell' Univeritá di Cranfield (UK), ed un tempo pari a 6 mesi è stato necessario per programmare i codici e trovare i risultati finali. Come giá anticipato nell' Abstract, lo studio del comportamento di un flusso d' aria all'interno di un' S-duct, é stato condotto seguendo i precedenti lavori di Rigobello, Ď Ambros e Dal Magro. L'obbiettivo principale di questa tesi é quello di estendere l'ottimizzazione robusta, già precedentemente implementata, con maggiori incertezze in ingresso (velocità d'ingresso, deviazione del flusso, angolo di bending) e in uscita (pressure recovery e swirl).

Nel primo capitolo, i principali parametri fisici sono stati descritti ed analizzati. Inoltre, é stata data una panoramica generale dello stato dell'arte delle S-ducts ed una panoramica storica dei precedenti lavori riguardanti questo tipo particolare di prese dinamiche.

Nel capitolo due, il lettore è stato introdotto alla comprensione teorica dei concetti di Ottimizzazione e Incertezza. In piú , é son stati ben spiegati e formulati dal punto di vista matematico l'ottimizzazione con singolo oggetto (SOO) e l'ottimizzazione multi-oggetto (MOO), in presenza e non di incertezze in ingresso. Alla fine di questo capitolo é stata data una definizione esaustiva di ottimizzazione robusta.

Nel capitolo tre é stato studiato il complesso meccanismo della quantificazione e valutazone delle incertezze. Un' intera panoramica teorica é stata data al lettore, introducendo e successivamente analizzando con discreto dettaglio le tecniche di quantificazione più comuni con l'aiuto di esempi. Infine sono stati presentati schemi pratici delle due tecniche utilizzate in questa tesi.

Nel capitolo quattro, sono stati analizzati i principali algoritmi usati nei problemi di ottimizzazione, in particolare il Tabu Search ovvero quello utilizzato in questi mesi di lavoro.

Nel capitolo cinque é stato presentato il caso di studio. Dunque é stata esaminata

la geometria di partenza detta baseline, la parametrizzazione utilizzata nei vari cicli di ottimizzazione robusta, la mesh, le condizioni al contorno dell'analisi cfd e le incertezze in ingresso ai diversi cicli implementati. Alla fine di questo capitolo é stata infine compilata una precisa lista di tutti i parametri di ingresso per ogni ciclo di ottimizzazione robusta sviluppato.
Nell'ultimo capitolo, il sei, sono state divulgate tutte le analisi di post-processing effettuate sui design di interesse, distinguendo i risultati ottenuti in base alle due tecniche di quantificazione delle incertezze utilizzate (NIPC, NISP). Nelle Appendici finali A,B,C,D,E sono presenti passaggi e schemi con cui sono stati costruiti i cinque codici.

Acknowledgements

I would like to thank my supervisor Dr. Timoleon Kipouros, for the advice provided during the project and to gave me the opportunity to work in the challenging field of Design Engineering. A big thank you goes to Mattia, Luca and Alberto without whom this experience wouldn't have been special.
I would like to express my gratitude to Prof. Ernesto Benini, who made this amazing experience in Cranfield possible. A sincere thought goes to my best flatmate ever: I will never forget you Mudassir!
Finally, a special thank goes to my family and to Elena. They have continuously supported me despite the distance and always will.

Contents

List of Figures

List of Tables

Abbreviations

AIP	**A**erodynamics **I**nterface **P**lane
CDF	**C**umulative **D**istribution **F**unction
CFD	**C**omputational **F**luid **D**ynamics
FFD	**F**ree **F**orm **D**eformation
LHS	**L**atin **H**ypercube **S**ampling
MOO	**M**ulti-**O**bjective **O**ptimisation
MOTS	**M**ulti-**O**bjective **T**abu **S**earch
NIPC	**N**on-**I**ntrusive **P**oint-**C**ollocation
NISP	**N**on-**I**ntrusive **S**pectral-**P**rojection
ODE	**O**rdinary **D**ifferential **E**quation
PC	**P**olynomial **C**haos
PDF	**P**robability **D**ensity **F**unction
RANS	**R**eynolds-**A**veraged **N**avier-**S**tokes equations
SOO	**S**ingle-**O**bjective **O**ptimisation

Symbols

I_n	Wiener-Askey polynomial chaos of order n
H_n	Hermite polynomial chaos of order n
M	Mach number
P	pressure
T	temperature
U_∞	free-stream velocity
X	general second order random process
W	weighting function
PR	pressure recovery
CP	pressure coefficient

α	swirl
δ_{ij}	Kronecker delta
Φ	generic polynomial basis
μ	mean
Ψ	Hermite polynomial basis
ρ	density
σ	standard deviation
ξ	Gaussian standard random variable
ζ	generic random variable

Dedicated to Nonno Filippo, Nonna Anna, Papá Francesco and Mamma Rosaria...

Introduction

In this MSc thesis, five robust optimisation cycles plus one machine learning optimisation problem were developed in order to improve the aerodynamic performances of S-ducts intakes. The following work was supervised by Dr. Timoleon Kipouros at Cranfield University (UK) and it took 6 months to implement all the codes and to find the respective results. As it is written in the Abstract, the study of the behaviour of the flow in an S-duct intake has been conducted following the previous studies performed by Rigobello [1], D'Ambros [2], Dal Magro [3], and the main objective of this thesis has been to extend the robust design optimisation for S-ducts with more uncertainties input (Inlet velocity, flux deviation, bending angle) and output (pressure recovery and swirl).

In the first chapter, the main physical parameters have been analysed and a general overview of the S-duct state-of-the-art and an historical overview of previous works about S-ducts intakes is given.

In chapter two a theoretical introduction about Optimisation and Uncertainties has been given to the reader. In this section, the optimisation problem formulation, its division into SOO (Single-objective optimisation) or MOO (Multi-objectives optimisation) problems and their mathematical formulation in presence or not of uncertainties are well explained. At the end of this chapter has been given an exhaustive definition of robust optimisation.

In the third chapter, the complex task of the uncertainties quantification has been underlined. The entire theoretical overview and some of the most common techniques and examples have been presented. Moreover, practical schemes of two techniques have been reported to help the reader to understand the topic.

In chapter four, all the optimization algorithms have been analysed, in particular the Tabu Search, which was used in these studies.

In chapter five, the case study has been explained. In particular the geometry, the parametrization, the mesh, the cfd analysis and the uncertainties taken into

account. The geometry that was used is similar to the one utilized in Wellborn [6] experiment, and Delot [11], while the parameterization is the same used in D'Ambros [2]. At the end of this chapter, a precise list of all the five optimisation starting parameters was given.

In chapter six, all the results have been disclosed, and in particular, the results of all the optimizations implemented with two different uncertainties quantification technique (NISP and NIPC), have been compared.

Finally in the Appendices A,B,C,D,E there are schemes that help the reader to better understand how the five codes have been implemented.

Chapter 1

S-duct state-of-the-art

S-duct intakes are a particular type of inlet for modern aircraft propulsion systems, and they are so called because they are characterized by a bended shape. In military applications, this kind of intakes has been adopted in order to reduce mass, size, fuel consumption and to increase reactivity and engine operations range. Instead, in the civil field the S-duct represents a design solution approached by some aircraft manufacturers to get less noise, drag and lowering the engine position compared to straight through design. On the other hand, the particular shape of these intakes creates complex aerodynamics distortions at the AIP, that compromise the engine performances. The latter can be categorized as: Total Pressure distortion, Swirl Angle and Total Temperature distortion.

1.1 Total Pressure losses

The definition of the total pressure is: the pressure value when the fluid element is brought to rest isoentropically [5]. The total pressure can be defined as:

$$P_{tot} = P_{static} + \frac{1}{2}\rho v^2, \qquad \rho = cost \tag{1.1}$$

$$P_{tot} = P_{static}\left[1 + \frac{(k-1)}{2}M^2\right]^{\frac{k}{k-1}}, \qquad \rho \neq cost \tag{1.2}$$

The total pressure losses throughout a diffusing duct are generally described with the Pressure Recovery (PR) parameter. In general, this is defined as the ratio

between the average total pressure at the Aerodynamic Interface Plane (AIP) and inlet total Pressure:

$$PR = \frac{\overline{P}_{AIP}}{P_{tot,in}} \tag{1.3}$$

The total pressure losses are caused by the cross-section growth and by the duct bends. These two geometry conditions create an adverse pressure gradient that subsequently leads to flow separation and causes a reverse flow and vortices formation. As a matter of fact, turning the flow, as generally known, leads to boundary layer thickening and to a next separation of the fluid from the S-duct walls. The flow detachment, in diffusing S-duct, is placed right after the first bend and it occupies a wide downstream region generating a big wake, as proved by Wellborn's experiment [6]. It is important to underline that in all this thesis we will refer to the pressure recovery with another coefficient defined as follow:

$$CP = 1 - PR \tag{1.4}$$

1.2 Swirl

Swirl is a parameter that determines the distortion of the flow. Considering cylindrical coordinates, it is possible to divide the velocity vector in tangential (U_θ) and axial (U_z) components and to define the swirl as follow:

$$\alpha = \arctan\left(\frac{U_{\theta,\,AIP}}{U_{z,\,AIP}}\right) \tag{1.5}$$

From equation 1.5 it is easy to understand why the swirl represents the distortion of the flow. In fact, it is the angle between the local velocity vector and the normal vector in the AIP plane. In figure 1.1 it is possible to see a schematic representation of the swirl angle.

According to the literature, it is considered positive if it has the same direction of the rotation of the compressor. El-Sayed et al. in [7] affirms how the swirl phenomenon might cause severe problems in the aircraft engine, such as vibrations and surge. In fact, swirl phenomenon occurs specifically when the engine is installed inside the aircraft fuselage and the intake is connected with the core thanks to a double bend or a S-shaped duct like in our work.

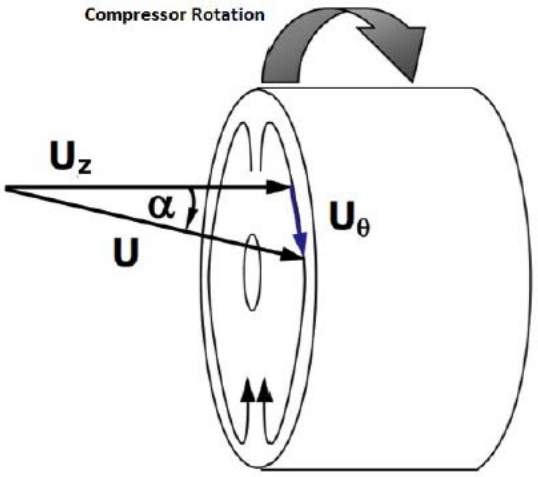


FIGURE 1.1: Representation of swirl angle.

Finally, it is necessary to specify that, in general, several typologies of swirl exist and they are identified as follows: Bulk Swirl, Paired Swirl, Tightly-Wound Vortex, Cross-Flow Swirl.

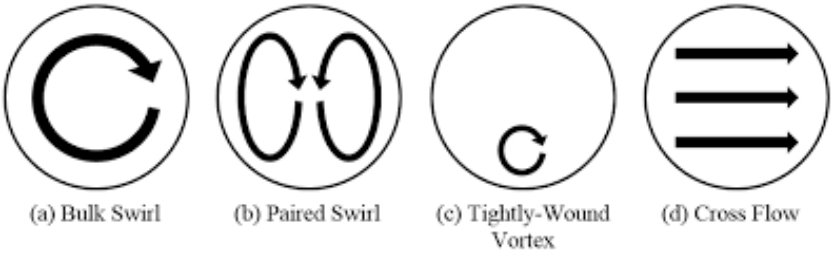


FIGURE 1.2: Swirl Classification.

1.2.1 Bulk Swirl

Bulk swirl occurs when the entire flow in the AIP is rotating in the same direction as it is represented in Figure 1.3. If the flow rotates in the same direction of the engine, it is called co-rotating swirl, otherwise it is named counter-rotating swirl. This phenomenon is the consequence of a non-symmetrical inlet pressure distribution that forces the fluid to rotate in a single direction. The co-rotating bulk swirl occurs when a vortex is ingested into the engine, on the other hand, the counter-rotating bulk swirl is internally generated.

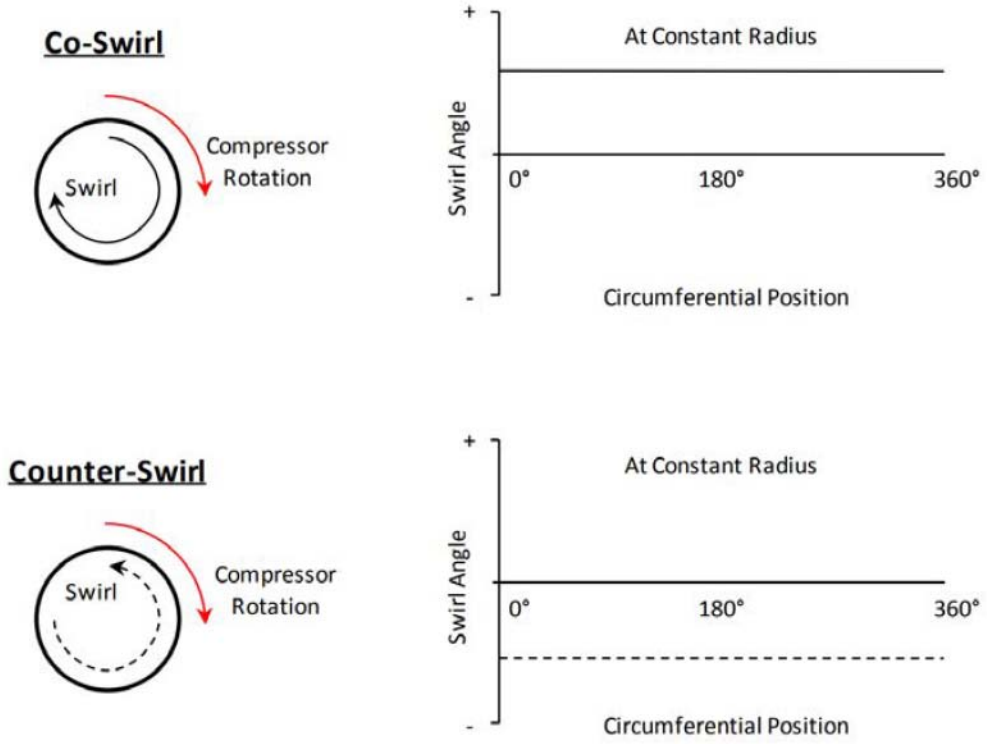


FIGURE 1.3: Bulk Swirl.

1.2.2 Paired Swirl

This type of swirl is the most important because it is the most common swirl that takes place in a bended duct. The paired swirl consists of two or more paired vortices rotating in opposite directions. If the vortices have the same magnitude, the swirl is called twin swirl, otherwise offset paired swirl (Figure 1.4).

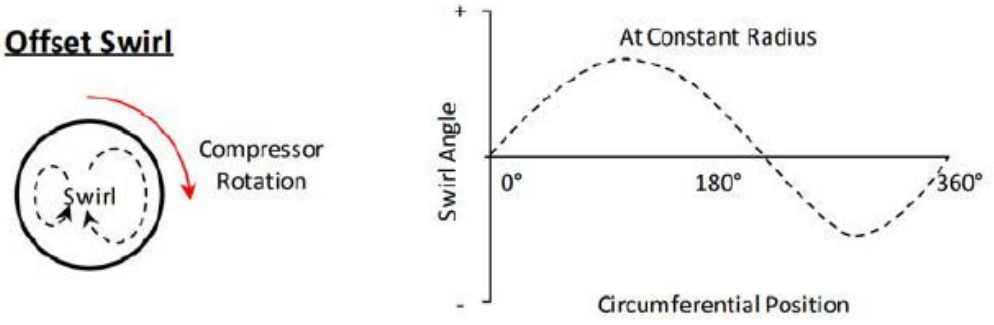


FIGURE 1.4: Offset Paired Swirl.

Its formation can be explained by simple considerations about the pressure and

momentum fields throughout a bended duct. Figure 1.5 can assist the understanding of the phenomenon. In figure 1.5.B it is possible to see the velocity distribution

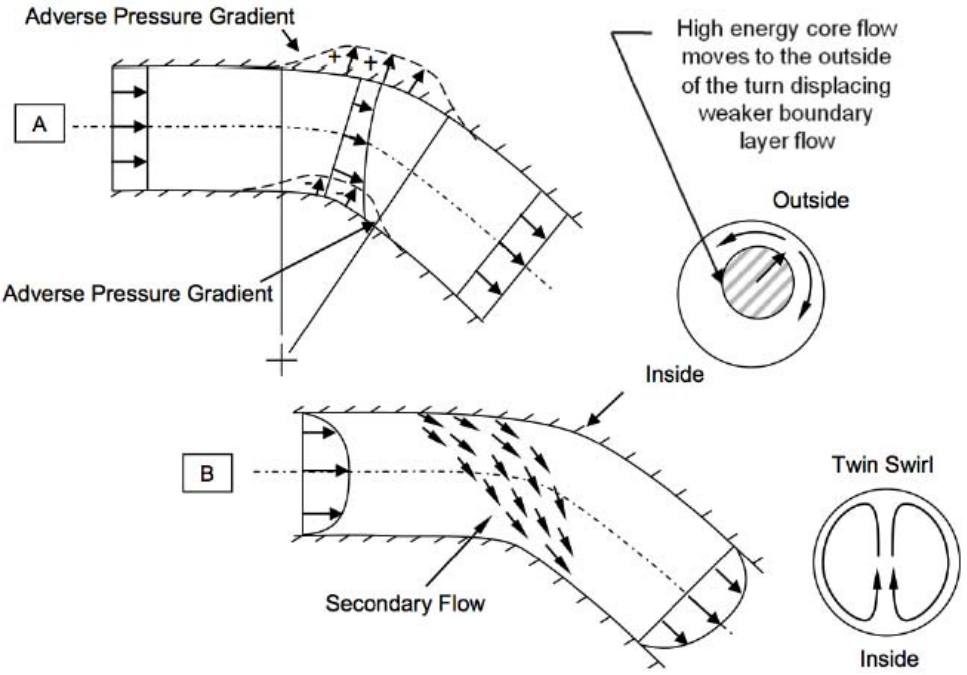


FIGURE 1.5: Formation of a paired swirl in a bended duct.

in a pipe. The value of the velocity is equal to zero at the walls, and it is maximum in the flow core. The natural consequence is that even the momentum distribution is either not uniform and there is a zone (the core) that have an high energy. When the flow travels through the bend, the core stream flow (high momentum flow) is pushed towards the upper wall by the centrifugal force (or better it tries to maintain the position, thanks to the higher momentum, until it finds the upper wall). This behaviour forces the low momentum flow to slip around the duct walls in a circular motion toward the internal bend as shown in figure 1.5.B. This phenomenon is not the only one that occurs. In fact, always thanks to the centrifugal force, when the flow travels through the bend, the pressure has to increase at the upper (external) wall and to decrease at the lower wall. Thus, a pressure gradient is created like in Figure 1.5.A. It is important to remark that the two phenomena are not separated, but superimposed. This is how the bend creates two counter-rotating vortices.

1.2.3 Tightly-Wound Vortex

It is essential to make a further consideration: this typology of swirl is created by several mechanisms, but all of them are characterized by three common elements that are: a stagnation point, a source of vorticity in the surrounding flow field and a flow sink. The Figures 1.6(a) and 1.6(b) exemplify two examples of what just explained. Specifically, in the first one we can see the stagnation point and the flow sink in the ground. In the second figure, the upstream disturbances in the flow field are the reason of the ingestion of fuselage and wing tip vortices.

(a)

(b)

FIGURE 1.6: Example of tightly-wound vortex.

1.2.4 Cross Swirl

This peculiar swirl is very close to the paired swirl just previously described. However, the main difference is that the velocity is uniform in the cross flow direction. In [8] is written that it is common to find this type of swirl in straight inlet ducts with the flow directions normal to the motion of the aircraft. Usually, it can be observed in lift fans, turboshaft and turboprop with bifurcated intake ducts.

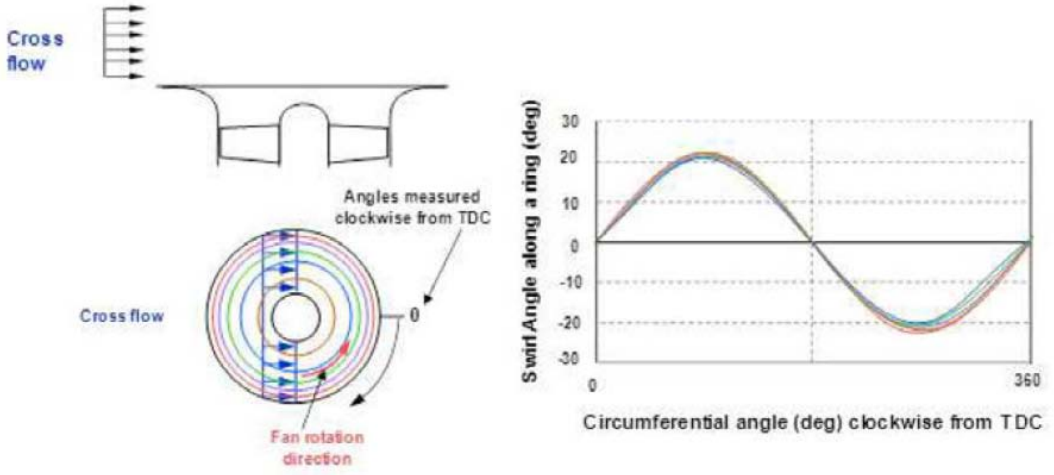


FIGURE 1.7: Cross-flow swirl distribution in a lift-fan installation. Source: [8]

1.3 DC60

The DCθ is a distortion parameter that represents the variation of the total pressure across the engine face. It has been derived by Rolls Royce and used extensively in the European fighter programs Tornado (Stocks and Bissinger, 1981) and Eurofighter (Bissinger and Jost, 2000) as reported by [9]. It is important to define this parameter because in this way we can quantify the total pressure distortion at the AIP. The DC60 is defined as follows:

$$DC\theta = \frac{\overline{P}_{tot,AIP} - \overline{P}_{tot,\theta}}{q_{f,AIP}} \tag{1.6}$$

Where $\overline{P}_{tot,\theta}$ is the lowest average total pressure of all sectors in the AIP (so it correspond to the worst sector), $\overline{P}_{tot,AIP}$ and $q_{f,AIP}$ are the average total pressure

and the dynamic head, respectively, at AIP. θ is the angular value of the sector and 60° or 15° are the value of θ that are most used.

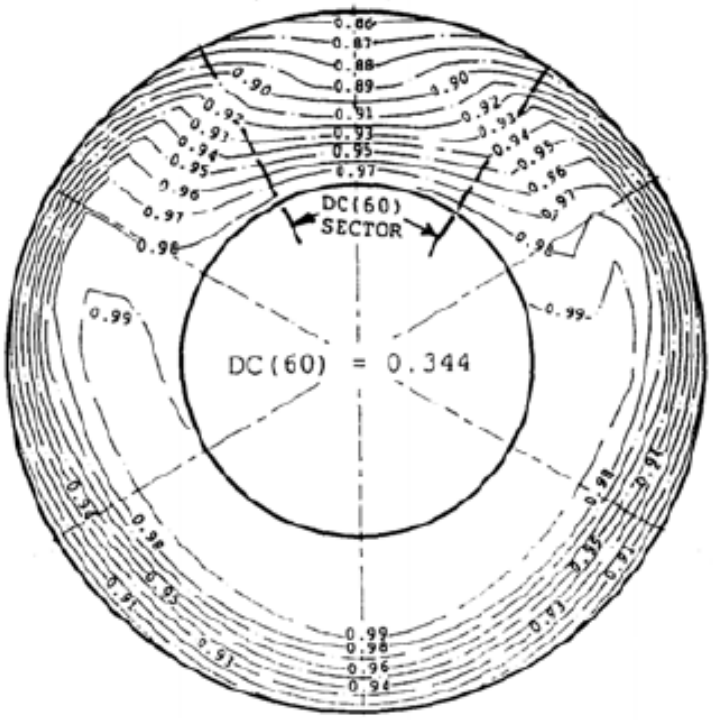


FIGURE 1.8: Example of DC60.

1.4 Historical Review

In order to determine the behaviour of the flow field throughout and at the exit of S-ducts, several experimental campaigns have been conducted over the years. The whole history of experimental test is well described in [10].

One of the very first experimental research was conducted by Weske. He studied the pressure and velocity field at the exit of elbow-shaped ducts with the final aim of improving the knowledge on aircraft intakes design parameters. The main finding was that the most influential parameter for the pressure drop downstream of the ducts was, more than the offset, the ratio between the exit and the inlet duct radii.

The effects of flow separation and skin friction on the pressure recovery were examined by John R. Henry in 1944, elaborating results from previous experimental investigations.

The presence of the two contra-rotating vortices at the exit of s-shaped intakes was investigated in detail only later, with the works of P.Bansod and R.W. Guo on simplified models of engine inlets.
All of the above reported experimental investigations have the goal to build a more concrete knowledge of the complex aerodynamic phenomena of s-shaped ducts. Most of these studies were conducted with simplified models, thick boundary layer and incompressible flow. Moreover, in some studies, the duct's radius was constant, so it did not involve the diffusion phenomena throughout the duct, hence reducing the presence of separation.

1.4.1 Wellborn experiment

A consistent breakthrough in the physical knowledge of the s-duct aerodynamics was achieved thanks to the experimental and computational campaigns performed by Wellborn [6] in the 1993. In these studies, both the effects of compressibility and diffusion were taken into account. All of the tests were performed at NASA Lewis Research Center. The settling chamber conditioned the incoming flow in the

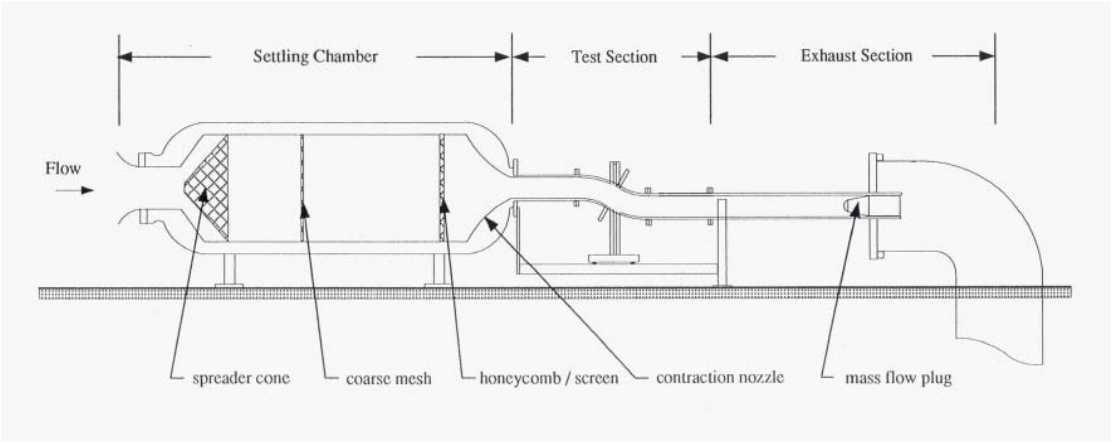


FIGURE 1.9: Schematic representation of Wellborn experiment.

following way. Air was drawn into the chamber through a large bell-mouth opening. A perforated spreader cone mixed the inlet flow. A coarse mesh conditioning screen reduced mean flow non uniformities. A honeycomb-screen combination removed large scale turbulence fluctuations. A seamless contraction section uniformly accelerated the flow from the settling chamber. An area contraction ratio of 59 to 1 ensured a low turbulence intensity flow and nearly uniform flow at the test section entrance. The test section for this experiment consisted of the diffusing S-duct and two constant area duct extensions. The first extension (10.21

cm diameter) served as the interface between the contraction exit and the S-duct entrance. The second extension (12.57 cm diameter) conveyed the flow from the S-duct to the exhaust region. The second extension was able to rotate about its centerline, when the facility was shut down. Each extension was 76.2 cm long and had the same internal surface finish as the S-duct. The exhaust section contained a circular cross-section pipe, a mass flow plug and a sub-atmospheric plenum. The purpose of the mass flow plug was to delete the influences of the exhaust plenum on the test section.

It must be remarked that a double circular arc diffusing duct was designed (Fig.1.9), with the aim to develop and study a complex three dimensional velocity field and flow separation. Visualization techniques and pressure measurements were used to investigate the flow. 220 pressure transducers at different stream-wise and circumferential positions were used to register the wall static pressure throughout the duct. Three- and five-hole probes were used to measure total pressure, static pressure and flow direction at different planes (planes A, B, C, D and E in Fig.1.10) for numerous radial and circumferential positions. The inlet Mach number was set to 0.6 and the Reynolds number of the inlet free-stream at $2.6 \cdot 10^6$. The study demonstrated the strong coupling between the boundary layer and the flow at the core of the duct. A vast separated region downstream of the first bend was observed. Furthermore, the curvature of the duct was found to be responsible of the generation of pressure-driven secondary flow which eventually induced the creation of the two contra-rotating vortices downstream.

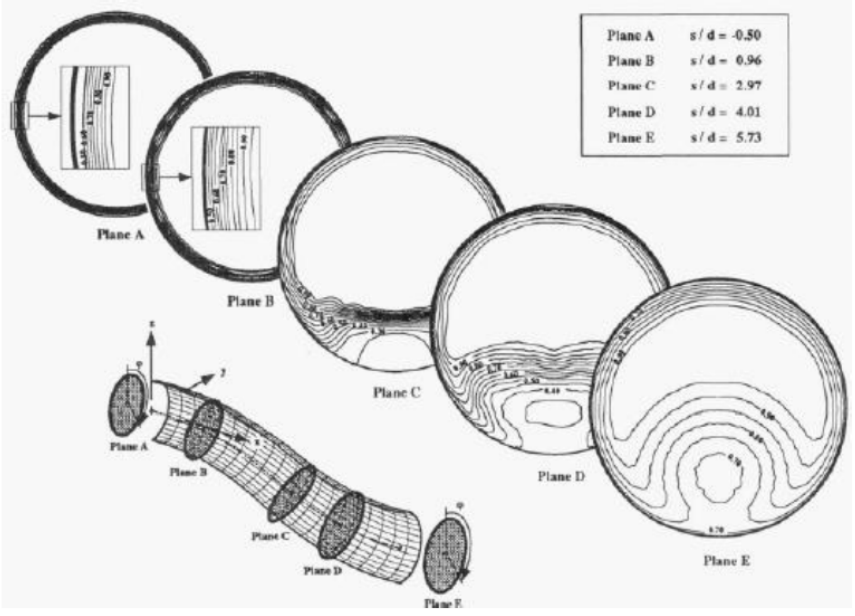


FIGURE 1.10: S-duct pressure contour on planes A, B, C, D and E.

1.4.2 Delot experiment

The Delot analysis [11], conducted in 2006 are the starting point to understand how to simulate a flow in a S-duct. Delot based her study on scaled-down Wellborn's S-duct geometry carrying out several computational tests to define the best set up that best reproduces real flow. She compared several meshes, solver codes and turbulence models; the project stated that Fluent solver best matches the separate region and well predicts low pressure region. For the purpose of our analysis, it is crucial to remind an important statement that she postulated after her research: the fluent solver well represents the separated region and the low pressure region, but the PR coefficient is higher than the one of the real flow.

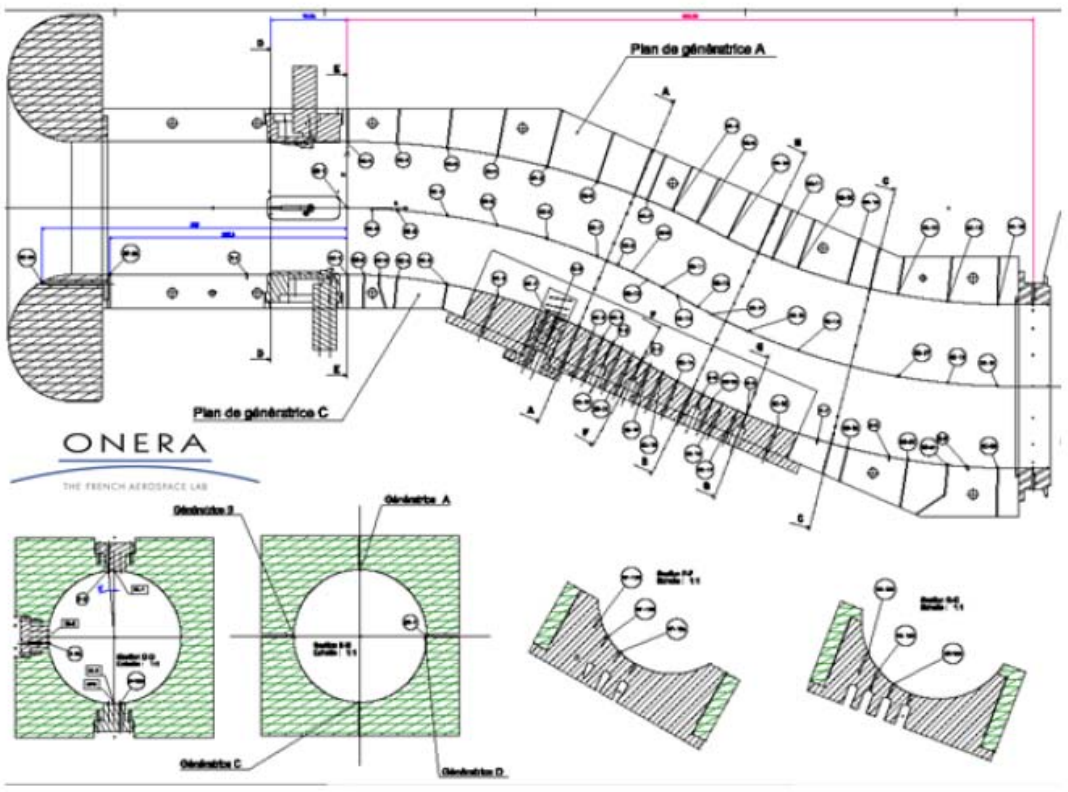


FIGURE 1.11: Scheme of Delot experiment.

1.4.3 Recent S-duct studies

In order to understand the simulation done in this thesis, we have to analyse the previous work done by Rigobello Aurora [1], R.Tridello [12], Alessio D' Ambros [2] and Davide Dal Magro [3]. Alessio D'Ambros and Davide Dal Magro are the most

important sources and the most recent. Precisely, D'Ambros research consisted in the optimization of the Delot geometry considering two objective functions: the pressure losses (CP) and the swirl (α). Instead Dal Magro research goal was the implementation of a robust optimisation cycle with an uncertainty input (Inlet Velocity) and two objective functions output: the CP_{mean} and CP_{stdv}.

Chapter 2

Optimisation and Uncertainties

Optimisation problems are present in everyday life. One common example, at the basis of optimisation theory, is the well-known *Travelling Salesman Problem*. Given a list of cities that a salesman has to visit to sell goods to customers, the question is how to find the shortest route that passes through every city and returns to the initial point. The more variables are introduced, the more complicated and highly non-linear the problem will become. Nowadays, optimisation is a very general automated design technique. When a designer faces to this technique for the first time, it is important to take into account that an optimisation problem is implemented in cycles, and all of them are composed of three main processes: *design analysis*, *results evaluation* and *new design creation*. The *design analysis* process is to determine the response of a specified design when it is subjected to a certain combination of input parameters. In other words, the goal is to find the value of one or more variables of interest. An example is to find the output pressure recovery of an S-duct as a result of certain fluid properties inputs at the inlet area. Instead, the *results evaluation* and the *new design creation* processes are used to verify if the last design is an optimum, and to create a new design for the next iteration. Efficient algorithms are therefore required to do the last two steps. At the end of the optimisation, the optimum design result will have the characteristics to satisfy specified performance and manufacturing constraints. In a numerical optimisation, we call design variables those parameters that can be changed in the system, while the code is searching for the best design that minimize or maximize one or more specific characteristics called objective functions. In mathematical terms, a simple deterministic optimisation problem can be stated as follows: given an independent variable $x \in \mathbb{R}$ called design variable, and an objective function

$f(x)$, the goal of a optimisation cycle is to automatically change the design variable in order to find a certain x^* such that $f(x^*)$ is a global minimum or maximum (it depends on the problem we are analysing). In a more general case, x can be a design vector of design variables $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. If only one objective function is present, the optimisation is said to be *single-objective* (SOO), otherwise is a *multi-objective* optimisation (MOO). The design to be acceptable it must also satisfy certain requirements. These requirements are called design constraints. Moreover, it must be remarked that a global optimum design might even not exist for complex problems: in these situations a *decision maker* comes into play by choosing one solution that may be of particular interest with respect to the others. This chapter starts with a brief review of various definitions, hence the focus is given to the mathematical difference between deterministic and stochastic optimisation problem and to the definition of Robust optimisation. After that we will see how the various single and multi-objective robust optimisation are mathematically implemented.

2.1 Optimisation problem formulation

Problem formulation is normally the most difficult part of the process. It is the selection of design variables, constraints, objectives, models and uncertainties.

- **Design variables**: Design variables are entities that can change the shape or properties of the system within a specified range during an optimisation design study. The design variables you create affect only the shape of the design system. For example, the control points of a parametrized S-duct can be considered design variables. In general, design variables can be continuous, discrete or boolean. They are often bounded, in fact, they often have maximum and minimum values. Depending on the solution method, these bounds can be treated as constraints or separately.

- **Constraints**: A constraint is a condition that must be satisfied to make the design feasible. Examples of constraints in a S-duct design can be related to manufacturing. Or, if after an iteration, the result of one or more objective functions exceed a certain value, that design is discarded.

- **Objective functions**: An objective is a variable of interest that has to be maximized or minimized. For example, a designer may wish to maximize performance or minimize weight. Many solution methods work only with single objective. The designer normally weights the various objectives and sums them to form a single objective. But other methods allow multi-objective optimization, such as the calculation of a Pareto front, which will be defined in Chapter 3.

- **Models**: The designer must also choose models to relate the constraints and the objectives to the design variables. These models are dependent on the discipline involved. They may be empirical models, such as a regression analysis of aircraft prices, theoretical models, such as from computational fluid dynamics. When the designer chooses the models, he must trade off the model fidelity with the computational time.

- **Uncertainties**: They are potential deficiencies in any phase or activity of the modelling process and they are due to lack of knowledge. Uncertainties are entirely stochastic and they are divided into aleatory uncertainties, which are intrinsically variable and often represented by probability density functions (PDFs), and epistemic uncertainties, which are usually due to a deliberate simplification or lack of understanding the modelled phenomena. An example is deliberately ignoring coupling effects to simplify models. This work will focus on aleatory uncertainties only, and for the moment, it is easier to think to them like random variable ζ. In Chapter 3 uncertainties will be better described and we will understand their quantifications.

2.2 Single and Multi-objectives problem

Once the design variables, constraints, objectives and the models have been chosen, the designer has to take into account the presence or not of uncertainties variables. Based on this, single and multi-objectives optimisation problem can be expressed in a deterministic or stochastic mathematical way.

2.2.1 Deterministic optimisation problem

Now we consider an objective function $f(\mathbf{x})$, where $\mathbf{x} \in \mathbb{X}$ represents vector of design variables. A *single-objective* minimisation problem is formulated in general as:

$$\begin{cases} f(\mathbf{x}^*) \leq f(\mathbf{x}) & \forall\, \mathbf{x} \in \mathbb{X} \\ s.to & g(\mathbf{x}) \leq 0,\ h(\mathbf{x}) = 0 \end{cases} \tag{2.1}$$

where $g(\mathbf{x})$ and $h(\mathbf{x})$ are constraints vectors and $\mathbb{X}$ is a set called design space.
As we can see, this is a deterministic single-objective problem where the objective function is only one, and there's no uncertainties input $\boldsymbol{\zeta}$.

Now we consider a set of objective functions $[f_1(\mathbf{x}); f_2(\mathbf{x}); \dots f_m(\mathbf{x})]$ where $\mathbf{x} \in \mathbb{X}$ represents a vector of design variables. A deterministic *multi-objective* minimisation problem is formulated as follow:

$$\begin{cases} \min\limits_{\mathbf{x}} [f_1(\mathbf{x}); f_2(\mathbf{x}); \dots f_m(\mathbf{x})] & \forall\, \mathbf{x} \in \mathbb{X} \\ s.to & g(\mathbf{x}) \leq 0,\ h(\mathbf{x}) = 0 \end{cases} \tag{2.2}$$

where, like before, $g(\mathbf{x})$ and $h(\mathbf{x})$ are constraints vectors and $\mathbb{X}$ is a set called design space.
The underlying difference between the single objective function and the MOO is that the latter does not provide an optimal solution that can ensure that all the objective functions are optimized.

2.2.2 Stochastic optimisation problem

In presence of uncertainties, the conventional deterministic optimisation becomes an optimisation under uncertainties. The designer has to take into account possible probabilistic information of these variables like the probability density function

(PDF) or the cumulative density function (CDF), which represents the identity card data of a design analysed under uncertainty. As said before this topic will be addressed in Chapter 3.

Now we consider an objective function $f(\mathbf{x}, \boldsymbol{\zeta})$, where $\mathbf{x} \in \mathbb{X}$ represents vector of design variables and $\boldsymbol{\zeta} \in \boldsymbol{\Omega}$ is vector of random variables: $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)$. Each element of ζ can be a design variable or another parameter. A *single-objective* minimisation problem is formulated in general as:

$$\begin{cases} f(\mathbf{x}^*, \boldsymbol{\zeta}) \leq f(\mathbf{x}, \boldsymbol{\zeta}) & \forall\, \mathbf{x} \in \mathbb{X},\ \boldsymbol{\zeta} \in \boldsymbol{\Omega} \\ s.to & g(\mathbf{x}) \leq 0,\ h(\mathbf{x}) = 0 \end{cases} \tag{2.3}$$

where $g(\mathbf{x})$ and $h(\mathbf{x})$ are constraints vectors and $\mathbb{X}$, $\boldsymbol{\Omega}$ are sets respectively called design space and random space.

Now we consider a set of objective functions $[f_1(\mathbf{x}, \boldsymbol{\zeta}); f_2(\mathbf{x}, \boldsymbol{\zeta}); \ldots f_m(\mathbf{x}, \boldsymbol{\zeta})]$ where $\mathbf{x} \in \mathbb{X}$ represents a vector of design variables, and $\boldsymbol{\zeta} \in \boldsymbol{\Omega}$ is vector of random variables. A deterministic *multi-objective* minimisation problem is formulated as follow:

$$\begin{cases} \min\limits_{\mathbb{X}} \left[f_1(\mathbf{x},\ \boldsymbol{\zeta}); f_2(\mathbf{x},\ \boldsymbol{\zeta}); \ldots f_m(\mathbf{x},\ \boldsymbol{\zeta})\right] & \forall\, \mathbf{x} \in \mathbb{X},\ \boldsymbol{\zeta} \in \boldsymbol{\Omega} \\ s.to & g(\mathbf{x}) \leq 0,\ h(\mathbf{x}) = 0 \end{cases} \tag{2.4}$$

where, like before, $g(\mathbf{x})$ and $h(\mathbf{x})$ are constraints vectors and $\mathbb{X}$, $\boldsymbol{\Omega}$ are sets respectively called design space and random space.
It must be remarked that each $f_i(\mathbf{x}, \boldsymbol{\zeta})$ is a random quantity induced by ζ, and as said before, it is not possible to find a unique solution that simultaneously optimise (in this case minimise) each objective function.

2.3 Robust Optimisation

As said in the previous sections, single and multi-objective optimisation aims at finding the design that maximises (or minimises) one or a certain set of objective functions. Optimised designs, however, may be very sensitive to small variations in the manufacturing or operating conditions: such variations can even be destructive in terms of drastic reductions of the objective functions. Davide Dal Magro et al., for instance, demonstrated in [3] how an optimised S-duct remarkably change its Pressure Recovery value when the Velocity input number deviated from the mean value. Since uncertainty is everywhere and, by definition, cannot be predicted, a scrupulous designer must take it into account when he's performing a robust optimisation, i.e. an optimisation that considers the sensitivity to uncertainty. The goal of robust optimisation is connected to the idea that in the presence of (input) uncertainty, the optimal design should be relatively insensitive (small output uncertainty). Consider, for instance, the plot in figure 2.1: for the same input variable variation $\pm\delta$, the objective function variations Δ and Δ' are considerably different on the two optimum points. Point B is the goal of the robust optimisation, i.e. a robust design, because is less sensitive to input uncertainty. It must be remarked that the latter may not necessarily coincide with the global optimum (point A in figure 2.1).

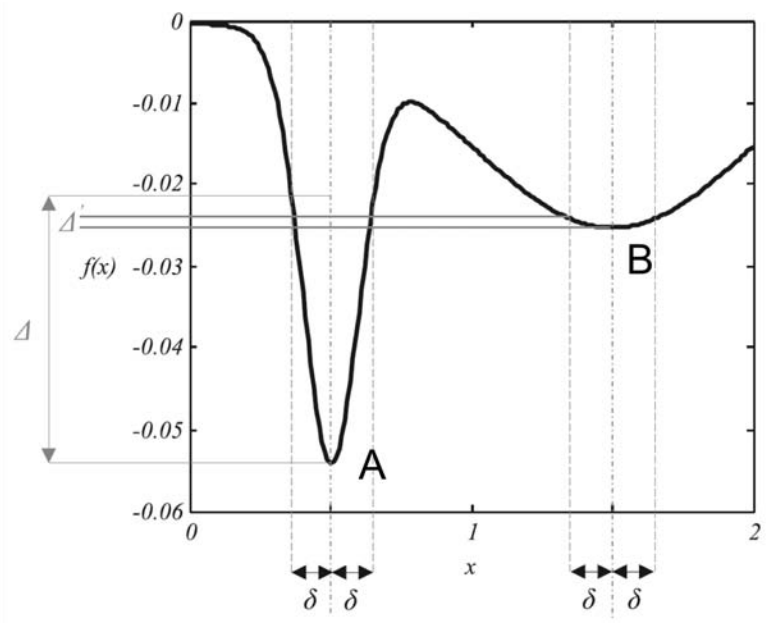


FIGURE 2.1: The effect of uncertainties on the objective function. Source: [13].

Mirjalili et al. [13] provided an overview of several approaches to perform a robust optimisation:

- Type I (expectation measure): the objective function is averaged in a neighbourhood of the solution, finding an expectation measure that replaces the 'crude' objective function value in the optimisation;
- Type II (variance measure): the original objective functions are optimised here, but the process is subject to an additional constraint on the variance of the solution. In other words, any found optima must not exceed a pre-defined variation Δ in the objective functions space;
- Hybrids of Type I and II: an average weighted on the importance of the neighbourhoods distributions is computed on the objective functions to find the expected measure.

An alternative to the above methods (that could although be seen as a Type II, variance measure approach), utilised in [14, 15], is to directly consider as objective functions the stochastic properties, i.e. mean and standard deviation, of the variable of interest. In this case the goal is to maximise (or minimise) the mean μ and always minimise the standard deviation σ. It is easy to see that with this approach a robust optimisation is always multi-objective: even for one variable of interest, two objective functions, the mean and the standard deviation, have to be considered.
The above cited approach has been chosen to perform a robust optimisation in this work. In fact, the explicit calculation (and visualisation) of the stochastic properties of the output can give the designer a more detailed overview of the problem, allowing more rational choices. Hence, accurate and efficient tools to model uncertainty and compute the stochastic properties of a random process are necessary: these will be discussed in Chapter 3.

2.3.1 Alternative Robust MOO

The alternative method mentioned above, can be written in the following mathematical form:

$$\begin{cases} \min\limits_{\mathbb{X}} \ \mu(\mathbf{x}) \\ \min\limits_{\mathbb{X}} \ \sigma^2(\mathbf{x}) \\ s.to \qquad\qquad g(\mathbf{x}) \leq 0, \ h(\mathbf{x}) = 0 \end{cases} \tag{2.5}$$

where μ is the mean, σ is the standard deviation and $g(\mathbf{x})$, $h(\mathbf{x})$ are constraints vectors.
With this approach, one variable of interest (one objective) is splitted in two objective functions, the mean and the standard deviation. In this case a challenge is posed by the increase in dimensionality. Indeed, an original $\boldsymbol{m}$ multi-objective problem turns into $m \times n$ multi-objective problem, where $\boldsymbol{n}$ is the number of the statistical moments (stochastic properties) used in the problem formulation. In the above system, $\boldsymbol{n} = 2$ (μ, σ) and $\boldsymbol{m} = 1$.

2.4 Dominance concept

In the previous sections we underlined that the aim of a MOO is to optimise (minimise or maximise) not only one, but a set of objective functions. However, it is not possible to find an unique solution that simultaneously optimise each objective function. To better understand this concept let's introduce the dominance concept as reported in [3].
In a minimisation problem with two objective functions, the design solution x_A^* dominates a design solution x_B^* if the following statement is verified:

$$f_j(x_A^*) \geq f_j(x_B^*) \quad \forall\, j = 1, \ldots, m \tag{2.6}$$

where m is the number of objective functions.
If $f_j(x_A^*) < f_j(x_B^*)$ for at least one $j = 1, \ldots, m$ and vice versa, then both x_A^* and x_B^* are non-dominated solutions.
The non dominated solutions are chosen and are considered as *Pareto-optimal set*, in fact all these points have the peculiarity that if one objective function improves, the other gets worse. At the end of the optimisation process, the best

non-dominated solutions are usually represented as a Pareto front such the one in figure 2.2. In figure 2.2, the Pareto front is two-dimensional because it is a result of a two objectives optimisation. The notion of Pareto optimal front can be extended to an arbitrary number of objectives, but visualising a Pareto front in more than three dimensions can be challenging.

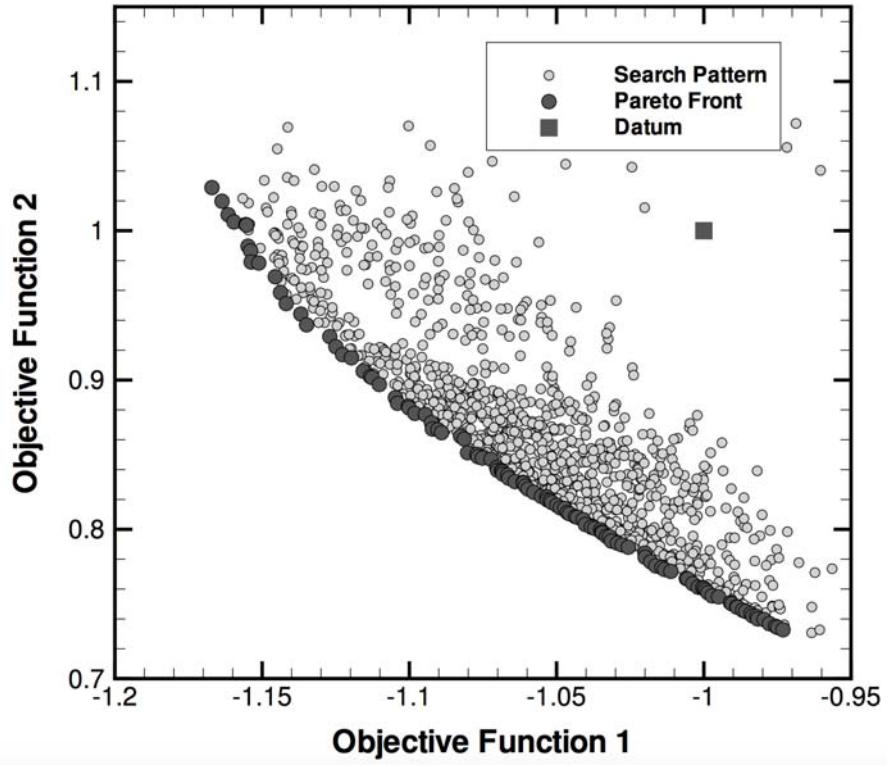


FIGURE 2.2: Example of a Pareto-optimal set (Pareto Front highlighted in dark grey). Source: [16].

Chapter 3

Uncertainty quantification

Over the past decades, thanks to the continuously increasing of the computational power available, Computational Fluid Dynamics (CFD) has become a convenient and trustworthy alternative to experimental tests to solve complex engineering problems. However, it is well known that reality cannot be modelled without errors and uncertainties. For this reason, Walter and Huyse in [17] pointed out the AIAA definitions of error and uncertainty:

- **Error**: A recognisable deficiency in any phase of or activity of modelling and simulation that is not due to lack of knowledge.
- **Uncertainty**: A potential deficiency in any phase or activity of the modelling process that is due to lack of knowledge.

A further subdivision of these two categories has then been defined by Oberkampf and Helton in [18]. An error can be either acknowledged or unacknowledged. Acknowledged errors can be, for examples, finite arithmetic precision in a computer or the discretisation of a continuous process. In this the analyst is typically aware of the magnitude of these deficiencies, nevertheless he/she can choose to accept them because of the excessive cost needed for the correction. Conversely, unacknowledged errors are deficiencies which the analyst is not aware of, but they are recognisable, such as errors in the source code of a program. Uncertainty can be further subdivided in aleatory and epistemic:

- **Aleatory**: Quantities affected by aleatory uncertainty can take values that will randomly change from time to time, but these are in a known range and

follow a known or assumable distribution. These uncertainties are commonly modelled with Probability Distribution Functions (PDFs).

- **Epistemic**: Epistemic uncertainty is caused by any lack of knowledge in any phase of the modelling process. For instance, the simple assumption of modelling an aleatory uncertainty with a particular PDF is a source of epistemic uncertainty, or, more generally, it can arise from assumptions introduced in the mathematical model or simplifications related to the correlation between physical processes. In order to correct epistemic uncertainties, therefore, more knowledge of the physical process is necessary (e.g. more experimental data).

This work will focus on aleatory uncertainty only. Oberkampf and Helton then stated the most common occurrences of uncertainty in engineering problems:

- **parametric**: these mainly include physical or chemical parameters. They are usually modelled with a Probability Distribution Function (PDF) if there is enough knowledge or must be guessed from the analysis.
- **modelling**: these include the uncertainties that occur when there are some inadequate understanding of the model implemented for physical/chemical processes. The typical example could be an uncompleted amount of data.
- **scenario abstraction**: these include all the parameters that are not take into account in the simulations, but that can occur.

Another classification of uncertainty has been made by Huyse et al. in [19], according to the impact and the frequency of an event (figure 3.1).

The task of uncertainty quantification and management (from now on the term aleatory will be omitted) can be significantly costly and key in many engineering designs, hence it must be carefully performed. In order to model uncertainty the literature gives us several schemes and methodologies. All of them are possible to be applied but now take a further look to stochastic methodologies.

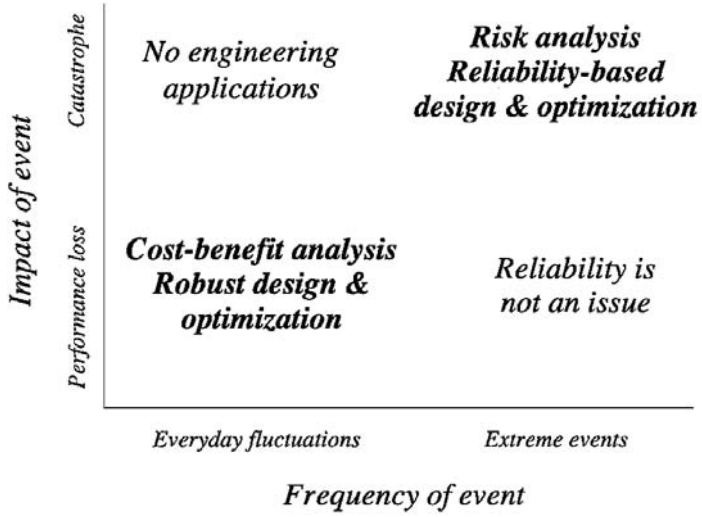


FIGURE 3.1: Uncertainty classification according to [19].

3.1 Stochastic approaches for aleatory uncertainty quantification

The stochastic approach is based on the main assumption that the uncertainty input variable has to be considered as random ζ, and with a probability distribution function. This assumption implies that also the output variable $f(x, \zeta)$ will be random with its own PDF. It must be remarked that, in the general case, there could be more than one input uncertainty variables and/or multiple outputs. As stated in [18], a classical mathematical model is considered deterministic, in the sense that for every fixed input it will produce a unique value for every output variable. Hence, to obtain a stochastic model of the process, it is necessary to run the deterministic model a certain number of times. In this section the most common stochastic schemes are analysed.

3.1.1 Monte Carlo Method

As reported in [20], the basic Monte Carlo Method follows a very simple approach:

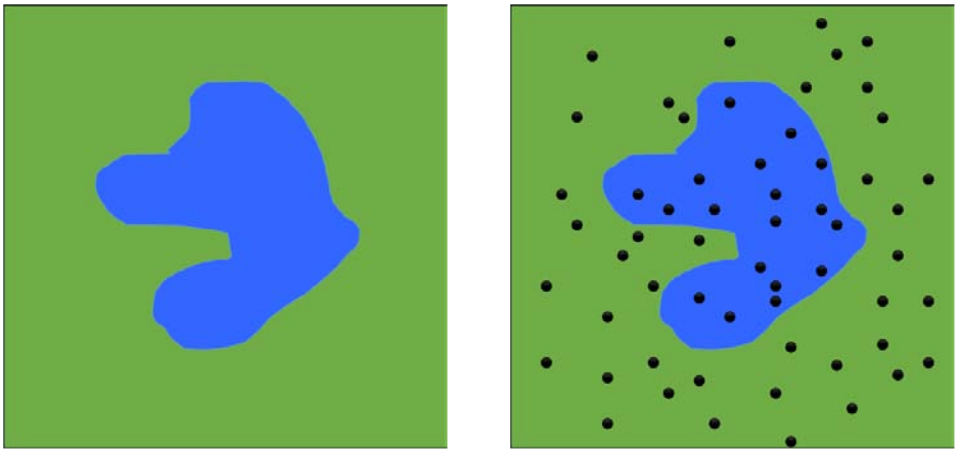
- sample the input variables following a known (or assumed) probability distribution.
- compute the deterministic output for each of the sampled input values.

- build a probability density function for the output variables.

Furthermore, this method has some positive peculiarities, such as:

- the method converge to the precise stochastic solution.
- the solutions are not directly linked to the number of the random variables.
- the method is easy to be implemented.
- it is applicable in any sort of problem and it does not need to be modified.

In many papers, in order to deeply understand the Monte Carlo Method, it is used the following example: determine the unknown surface of a lake contained in a square of known side, as shown in figure 3.2(a). The problem can be solved by



(a) Lake of unknown surface. (b) Random shots on the square.

FIGURE 3.2: Determination of the surface of a lake with random shots.

randomly shooting with a cannon inside the square and count the shots landed on earth N_{earth}. If N is the total number of shots, the surface of the lake S_{lake} can be approximated by:

$$\frac{S_{lake}}{S_{terrain}} = \frac{N - N_{earth}}{N_{earth}} \tag{3.1}$$

where $S_{terrain}$ is the known surface of the square. It is clear that the above estimation is accurate if the number of shots is high and uniformly distributed.
The Monte Carlo Method is known to converge to the exact stochastic solution for an infinite number of samples. For this reason, it is frequently used as a base for comparison with other methods.

Nevertheless, the two major drawbacks of this scheme are the need for a good random numbers generator (to obtain a uniform distribution of the 'shots') and, most of all, the slow convergence rate. The latter is of the order of $1/\sqrt{N}$ with N being the number of samples. As a matter of fact, the integration with very high time-consuming optimisation processes (e.g. CFD simulations) becomes infeasible.

3.1.2 Taylor Series Method

This model performs the Taylor Expansion of the output variables around the mean value [20]. It is usually more efficient and fast than Monte Carlo Method (depending on the derivatives evaluation), but it shows some issues in the treatment of highly non-linear or discrete functions, and it is not accurate in zones far from the mean value [21].

3.1.3 Sigma Point

Presented by Padulo et al. in [21] and [22], the Sigma Point is a stochastic method. The main idea is that is better to approximate the inputs value rather than the outputs. In order to do that, the procedure consists in choosing the sigma points, that are input points, symmetrically distributed around the mean value. As a result, the deterministic models will be computed only for these points. Usually, the random variable is Gaussian and it is called ξ. If μ_ξ is the mean and σ_ξ the standard deviation.

$$\xi_0 = \mu_\xi \tag{3.2}$$

$$\xi_{p\pm} = \mu_\xi \pm h_{sp}\sigma_\xi \tag{3.3}$$

The coefficient h_{sp} is arbitrarily chosen by the user and all the sampling points depend on this parameter. This method has been successfully tested by Moro in [15], even though it has been shown that results heavily depend upon the choice of h_{sp}.

3.1.4 Polynomial Chaos

The Polynomial Chaos scheme (PC) views the inputs and the outputs random variables as an infinite summation of terms where it is possible to identify deterministic coefficients multiplied for random basis polynomials. This theory has its fundamentals on the homogeneous chaos written by Wiener in [23], and it is based on the assumption that a second-order random process (i.e. a process with a finite variance σ^2), as the majority of the engineering process, can be expressed in terms of orthogonal polynomials.
Several types of polynomials are used to implement the polynomial Chaos, but the original ones is the Hermite Polynomials in terms of Gaussian random variables. In fact, this concept has been generalized by Xiu and Karniadakis, into every orthogonal polynomials that belong to the Askey-Scheme, only after 64 years the Weiner article. The generalization has been made also for some other general PDF that are not included in the Askey-Scheme [24].
The Polynomial chaos is a method that is used in many works, especially to integrate an optimization work, as in [25–27], and it is very attractive because of its very high convergence rate (exponential or quasi-exponential). However, many issues have been observed [28]:

- an analytical PDF must exist for every uncertain variable. If not known, it must be guessed;
- it is computational efficient for a small number of random variables only;
- it is intrusive, i.e. it requires modifications in the solver.

In order to solve some of these issues many modifications have been made, in particular on the intrusiveness. In fact in this thesis, we will use non intrusive variant of the Polynomial Chaos.

3.2 Polynomial Chaos formulation

Polynomial chaos expansion is based on the homogeneous chaos developed by Wiener [23]. Wiener used Hermite polynomials in terms of Gaussian random variables as the basis for an expansion of random processes. The Cameron–Martin

theorem [29] proved that the expansion could represent any second-order random process in terms of orthogonal polynomials.

3.2.1 Hermite polynomials chaos

With the Hermite polynomials, a second-order process $X(\theta)$ can be expanded as follow:

$$\begin{aligned} X(\theta) = a_0 H_0 + \\ + \sum_{i_1=1}^{\infty} a_{i_1} H_1(\xi_{i_1}(\theta)) + \\ + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} H_2(\xi_{i_1}(\theta), \xi_{i_2}(\theta)) + \\ + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} H_3(\xi_{i_1}(\theta), \xi_{i_2}(\theta), \xi_{i_3}(\theta)) + \\ + \ \ldots \end{aligned} \tag{3.4}$$

where $H_n(\xi_{i_1}, \ldots, \xi_{i_n})$ denote the Hermite polynomials of order n in terms of the multi-dimensional independent standard Gaussian random variables $\boldsymbol{\xi} = (\xi_{i_1}, \ldots, \xi_{i_n})$ with *zero mean* and *unit variance*. Instead, $a_{i_1} \ldots a_{i_n}$ are deterministic coefficients. The above equation is the discrete version of the original Wiener polynomial chaos expansion, where the continuous integrals are replaced by summations. The general expression of the polynomials is given by:

$$H_n(\xi_{i_1}, \ldots, \xi_{i_n}) = (-1)^n e^{\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}} \frac{\partial^n}{\partial \xi_{i_1} \cdots \partial \xi_{i_n}} e^{-\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}} \tag{3.5}$$

For example, if $\boldsymbol{\xi} = (\xi_1)$, the one-dimensional Hermite polynomials are:

$$\Psi_0 = 1, \quad \Psi_1 = \xi, \quad \Psi_2 = \xi^2 - 1, \quad \Psi_3 = \xi^3 - 3\xi, \ldots \tag{3.6}$$

and if $\boldsymbol{\xi} = (\xi_1, \xi_2)$, the two-dimensional Hermite polynomials are:

$$\Psi_0 = 1, \quad \Psi_1 = \xi_1, \quad \Psi_2 = \xi_2, \quad \Psi_3 = \xi_1^2 - 1, \quad \Psi_4 = \xi_1 \xi_2, \quad \Psi_5 = \xi_2^2 - 1, \ldots \tag{3.7}$$

For notational convenience, equation 3.4 can be rewritten as:

$$X(\theta) = \sum_{j=0}^{\infty} \hat{a}_j \Psi_j(\boldsymbol{\xi}) \tag{3.8}$$

where there is a one-to-one correspondence between the functions $H_n(\xi_{i_1}, \ldots, \xi_{i_n})$ and $\Psi_j(\boldsymbol{\xi})$ and also between the coefficients $\hat{a}_j$ and $a_{i_1} \ldots a_{i_n}$.
For clarity, the two-dimensional expansion is shown here, both in the fully expanded form (See Eq 3.4):

$$\begin{aligned} X(\theta) = a_0 H_0 + a_1 H_1(\xi_1) + a_2 H_1(\xi_2) + a_{11} H_2(\xi_1, \xi_1) + \\ + a_{12} H_2(\xi_2, \xi_1) + a_{22} H_2(\xi_2, \xi_2) + \cdots \end{aligned} \tag{3.9}$$

and in the simplified form (see Eq 3.8):

$$\begin{aligned} X(\theta) &= \hat{a}_0 \Psi_0 + \hat{a}_1 \Psi_1 + \hat{a}_2 \Psi_2 + \hat{a}_3 \Psi_3 + \hat{a}_4 \Psi_4 + \hat{a}_5 \Psi_5 + \cdots \\ &= \hat{a}_0 + \hat{a}_1 \xi_1 + \hat{a}_2 \xi_2 + \hat{a}_3 (\xi_1^2 - 1) + \hat{a}_4 (\xi_1 \xi_2) + \hat{a}_5 (\xi_2^2 - 1) + \cdots \end{aligned} \tag{3.10}$$

The polynomial basis Ψ_j of Hermite-Chaos forms a complete orthogonal basis, i.e.:

$$\langle \Psi_i \Psi_j \rangle = \left\langle \Psi_i^2 \right\rangle \delta_{ij} \tag{3.11}$$

where δ_{ij} is the Kronecker delta and $\langle \cdot, \cdot \rangle$ denotes the ensemble average, which correspond to the inner product in the Hilbert space of Gaussian variables:

$$\langle f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) \rangle = \int_{\Omega} f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) W(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{3.12}$$

where $W(\boldsymbol{\xi})$ is the weighting function corresponding to the polynomial basisΨ_j.
For the Hermite polynomials this weighting function is:

$$W(\boldsymbol{\xi}) = \frac{1}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}\boldsymbol{\xi}^T \boldsymbol{\xi}} \tag{3.13}$$

The Hermite polynomials are paired with the Gaussian distribution because Hermite polynomials are expressed in terms of Gaussian variables and, by definition, these polynomials are orthogonal to the weighting function $W(\boldsymbol{\xi})$ which has the form of the multi-dimensional independent Gaussian probability distribution with unit variance. [30].

3.2.2 Generalisation to any random distribution

The Cameron-Martin theorem assures the Wiener-Hermite chaos exhibits an exponential convergence rate when applied to Gaussian distributions. However, for general non-Gaussian random inputs, such as Beta or Uniform distributions, the convergence rate is not fast and in some cases the convergence rate is, in fact, severely deteriorated. [30]. In order to deal with more general random inputs, we introduce the Wiener-Askey polynomial chaos expansion as a generalization of the original Wiener-Chaos expansion. Similar to equation 3.4, we represent the general second-order random process $X(\theta)$ as:

$$\begin{aligned} X(\theta) = a_0 I_0 + \\ + \sum_{i_1=1}^{\infty} c_{i_1} I_1(\zeta_{i_1}(\theta)) + \\ + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} c_{i_1 i_2} I_2(\zeta_{i_1}(\theta), \zeta_{i_2}(\theta)) + \\ + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} c_{i_1 i_2 i_3} I_3(\zeta_{i_1}(\theta), \zeta_{i_2}(\theta), \zeta_{i_3}(\theta)) + \\ + \ \ldots \end{aligned} \tag{3.14}$$

where $I_n(\zeta_{i_1}, \ldots, \zeta_{i_n})$ denotes the Wiener-Askey polynomials of order n in terms of the multi-dimensional random variables $\boldsymbol{\zeta} = (\zeta_{i_1}, \ldots, \zeta_{i_n})$ and $c_{i_1} \ldots c_{i_n}$ are deterministic coefficients.
Again for notational convenience, we rewrite Eq 3.14 as:

$$X(\theta) = \sum_{j=0}^{\infty} \hat{c}_j \Phi_j(\boldsymbol{\zeta}) \tag{3.15}$$

where there is a one-to-one correspondence between the functions $I_n(\zeta_{i_1}, \ldots, \zeta_{i_n})$ and $Phi_j(\zeta)$ and their coefficients $\hat{c}_j$ and $c_{i_1} \ldots c_{i_n}$. The orthogonality relation of the Wiener-Askey polynomial chaos takes the form:

$$\langle \Phi_i \Phi_j \rangle = \left\langle \Phi_i^2 \right\rangle \delta_{ij} \tag{3.16}$$

where δ_{ij} is the Kronecker delta and $\langle \dots \rangle$ is the ensemble average, which corresponds to the inner product in the Hilbert space of the variables ζ.

$$\langle f(\boldsymbol{\zeta})g(\boldsymbol{\zeta})\rangle = \int_{\Omega} f(\boldsymbol{\zeta})g(\boldsymbol{\zeta})W(\boldsymbol{\zeta})d\boldsymbol{\zeta} \tag{3.17}$$

As stated in [24], some polynomials from the Askey scheme have as weighting functions $W(\boldsymbol{\zeta})$ the same as the PDF of certain random distributions. As a matter of fact, it is convenient to choose the type of random variables $\boldsymbol{\zeta}$ according to their probability distributions, as shown in table 3.1.

TABLE 3.1: Types of random variables and their corresponding chaos.

	Random variables $\boldsymbol{\zeta}$	**Wiener-Askey chaos $\Phi(\boldsymbol{\zeta})$**	**Support**
Continuous	Gaussian	Hermite Chaos	$(-\infty, +\infty)$
	Gamma	Laguerre Chaos	$[0, +\infty)$
	Beta	Jacobi Chaos	$[a, b]$
	Uniform	Legendre Chaos	$[a, b]$
Discrete	Poisson	Charlier Chaos	$\{0, 1, 2, \dots\}$
	Binomial	Krawtchouk Chaos	$\{0, 1, \dots, N\}$
	Negative Binomial	Meixner Chaos	$\{0, 1, 2, \dots\}$
	Hypergeometric	Hahn Chaos	$\{0, 1, \dots, N\}$

3.3 Example: Stochastic ODE

The below example reports the overall procedure applied to a simple ODE.
Consider the following ordinary differential equation:

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}t} = -ky \\ y(0) = \hat{y} \end{cases} \tag{3.18}$$

where the coefficient $k = k(\theta)$ is a random variable with a Gaussian distribution. Therefore the Wiener-Hermite chaos and its specific notations will be used. k has a mean μ_k and a standard deviation σ_k, and it is possible to explicit it as:

$$k(\theta) = \mu_k + \sigma_k \xi_1(\theta) \tag{3.19}$$

where θ is the random variable dimension, and $\xi_1(\theta)$ is the value of the random variable depending, in this specific example, on the Gaussian distribution. In order

to simplify the notation, the $\xi_1(\theta)$ is rewritten as ξ, since the random variable taken into account in this case, is one dimensional.
Treating the problem as stochastic, the Wiener-polynomial chaos expansion is applied to both the random variable $k(\theta)$ and the solution $y(t,\theta)$:

$$k(\theta) = \sum_{i=0}^{\infty} k_i \Psi_i(\xi), \quad y(t,\theta) = \sum_{i=0}^{\infty} y_i(t)\Psi_i(\xi) \tag{3.20}$$

For practical applications, the infinite summations must be truncated to a finite number. The new upper summation limit depend on the number of random dimensions and the desired order of the polynomials:

$$P + 1 = \frac{(n+p)!}{n!p!} \tag{3.21}$$

where P is the new upper summation limit, p the desired order of the polynomials, and n the number of random dimensions. Therefore, we obtain:

$$k \approx \sum_{i=0}^{P} k_i \Psi_i(\xi), \quad y \approx \sum_{i=0}^{P} y_i(t)\Psi_i(\xi) \tag{3.22}$$

The above expansion can be considered a spectral expansion where the stochastic process is divided into random basis polynomials and deterministic coefficients (k_i and $y_i(t)$).
Substituting equations 3.22 in the original equation 3.18, the resulting differential equation becomes:

$$\sum_{i=0}^{P} \frac{\mathrm{d}y_i(t)}{\mathrm{d}t}\Psi_i(\xi) = -\sum_{i=0}^{P}\sum_{j=0}^{P} k_i\, y_j(t)\, \Psi_i(\xi)\Psi_j(\xi) \tag{3.23}$$

The truncation inevitably introduces error, and a Gelerkin projection is used to ensure that the error is orthogonally projected to the reduced Hermite polynomial basis Ψ_l. We will do this by taking the inner product of the equation with each basis $\langle \cdot\,, \Psi_l \rangle$:

$$\left\langle \sum_{i=0}^{P} \frac{\mathrm{d}y_i(t)}{\mathrm{d}t}\Psi_i,\ \Psi_l \right\rangle = -\left\langle \sum_{i=0}^{P}\sum_{j=0}^{P} k_i\, y_j(t)\, \Psi_i\Psi_j,\ \Psi_l \right\rangle \tag{3.24}$$

where $l = 0, 1, \ldots, P$.
Remembering the orthogonality equation 3.11, the left hand side of the equation is

always equal to zero, except when $i = l$, therefore, it becomes $\frac{dy_l(t)}{dt} \langle \Psi_l^2 \rangle$. Instead, the right hand side doesn't reduce because the orthogonality relation doesn't work to the inner product of three polynomials. Therefore we obtain:

$$\frac{\mathrm{d}y_l(t)}{\mathrm{d}t} = -\frac{1}{\langle \Psi_l^2 \rangle} \sum_{i=0}^{P} \sum_{j=0}^{P} k_i \, y_j(t) \, e_{ijl} \tag{3.25}$$

where $e_{ijl} = \langle \Psi_i \Psi_j \Psi_l \rangle$ and $l = 0, 1, \ldots, P$. The formula of inner product 3.17 allows the calculation of $\langle \Psi_l^2 \rangle$ and $\langle \Psi_i \Psi_j \Psi_l \rangle$, and removes all the random parameters from the equations. In this way, we will have a set of $P + 1$ deterministic equations where the polynomial chaos coefficients y_j are the unknowns.
In this specific case, the $P + 1$ equations can be further simplified thanks to the known behaviour of $k(\theta)$, so the coefficients k_i can be computed.

$$k(\theta) = \mu_k + \sigma_k \xi \approx \sum_{i=0}^{P} k_i \Psi_i(\xi) \tag{3.26}$$

If the right hand side is expanded:

$$k(\theta) = \mu_k + \sigma_k \xi = k_0 + k_1 \xi + k_2(\xi^2 - 1) + \cdots \tag{3.27}$$

it can be seen that $k_0 = \mu_k$, $k_1 = \sigma_k$ and $\forall i > 1, k_i = 0$.
If a third order chaos is used ($p = 3$), since the present problem has only one dimension ($n = 1$), according to equation 3.21, $P + 1 = 4$. At the end, if we put $l = 0$ and we calculate the summation with index i, equation 3.25 is reduced to:

$$\frac{\mathrm{d}y_0(t)}{\mathrm{d}t} = -\mu_k \sum_{j=0}^{P} y_j(t) \, \langle \Psi_0 \Psi_j \Psi_0 \rangle \; - \; \sigma_k \sum_{j=0}^{P} y_j(t) \, \langle \Psi_1 \Psi_j \Psi_0 \rangle \tag{3.28}$$

Hence, the first equation of the system is

$$\begin{aligned} \frac{\mathrm{d}y_0(t)}{\mathrm{d}t} = & -\mu_k [y_0(t) \, \langle \Psi_0 \Psi_0 \Psi_0 \rangle + \; y_1(t) \, \langle \Psi_0 \Psi_1 \Psi_0 \rangle + \\ & + y_2(t) \, \langle \Psi_0 \Psi_2 \Psi_0 \rangle + \; y_3(t) \, \langle \Psi_0 \Psi_3 \Psi_0 \rangle] + \\ & - \; \sigma_k [y_0(t) \, \langle \Psi_1 \Psi_0 \Psi_0 \rangle + \; y_1(t) \, \langle \Psi_1 \Psi_1 \Psi_0 \rangle + \\ & + y_2(t) \, \langle \Psi_1 \Psi_2 \Psi_0 \rangle + \; y_3(t) \, \langle \Psi_1 \Psi_3 \Psi_0 \rangle] \end{aligned} \tag{3.29}$$

All the other P deterministic equations can be found for $l = 1, \ldots, P$. After that, once the system is resolved and all the y_i coefficients are determined (y_0, y_1, y_2 and

y_3), they can be substituted back into the original $y(t;\theta)$ expansion displayed in equation 3.22:

$$y(t,\theta) = y_0(t) + y_1(t)\Psi_1(\theta) + y_2(t)\Psi_2(\theta) + y_3(t)\Psi_3(\theta) \tag{3.30}$$

The first statistic moment, the mean, is the first polynomial chaos coefficient:

$$\mu_y = y_0(t) \tag{3.31}$$

whilst the standard deviation can be computed as:

$$\sigma_y = \sqrt{\sum_{j=1}^{P} y_j^2(t) \left\langle \Psi_j^2 \right\rangle} \tag{3.32}$$

Assuming k is a *Gaussian* random variable, having thus probability density function

$$f(k) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \tag{3.33}$$

therefore $\mu_k = 0$ and $\sigma_k = 1$, Xiu and Karniadakis in [24] calculated the solution of the problem, reported in figure 3.3. The power of the Polynomial Chaos method

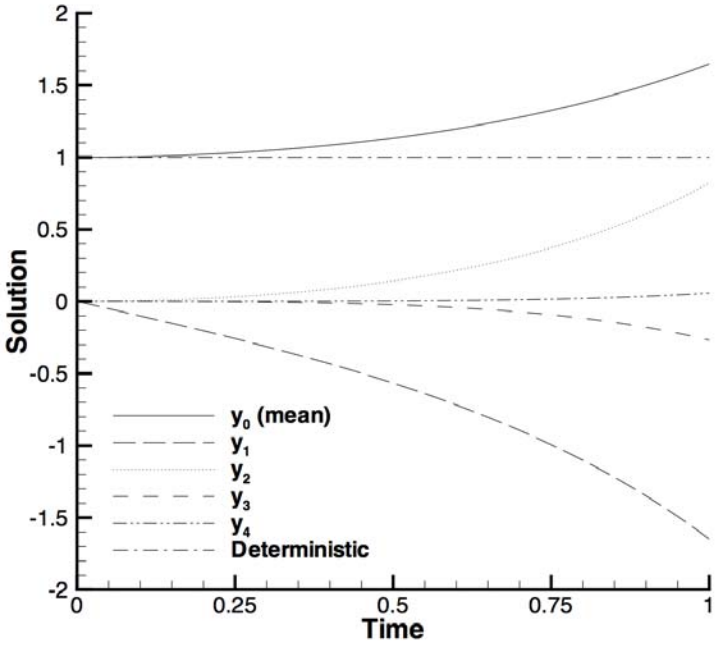


FIGURE 3.3: Plot of the polynomial chaos coefficients up to order 4 for the example ODE. Source: [24].

is clear when it is considered that, for all the PDFs reported in table 3.1, an

exponential or quasi-exponential convergence rate is achieved, meaning that the error decreases exponentially with the PC order. Ghisu et al. in [31] showed that it is also possible the use of nonstandard probability density functions, even though the exponential convergence is not granted.

3.4 Non-intrusive methods

In paragraph 3.1.4 the intrusiveness of the Polynomial Chaos method has been presented as the major drawback. This is now more clear by looking at equation 3.29: in order to solve directly the deterministic system and find the PC coefficients, the intrusive Polynomial Chaos method requires to directly modify the code of a deterministic solver. This can be inconvenient and complex, especially if the problem deals with complicated analyses as CFD simulations. In order to overcome this issue, several non-intrusive approaches have been proposed and can be found in literature: the common idea is to treat the deterministic solver as a black box, and find the PC coefficient after having run the necessary deterministic code.

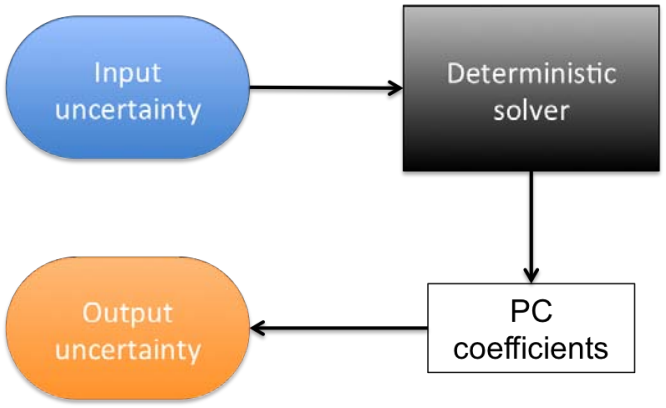


FIGURE 3.4: Schematic of the non-intrusive methods.

In the present work, three non-intrusive schemes have been analysed.

3.4.1 Non-Intrusive Spectral Projection

Non Intrusive Spectral Projection (NISP) is based on the observation that the PC coefficients for a variable a (i.e., $a_0, \ldots, a_P$) can be obtained by projecting the

deterministic solution onto the PC basis Φ_i and using the orthogonality relation 3.11.

$$a = \sum_{i=0}^{\infty} a_i \Phi_i$$

$$\langle a,\ \Phi_i \rangle\ = \left\langle \sum_{i=0}^{\infty} a_i \Phi_i, \Phi_i \right\rangle$$

$$a_i = \frac{\langle a,\ \Phi_i \rangle}{\langle \Phi_i^2 \rangle} \tag{3.34}$$

Now we change the above equation with a new nomenclature:

$$a_i = \frac{\left\langle a^d(\boldsymbol{\zeta}),\ \Phi_i \right\rangle}{\langle \Phi_i^2 \rangle} \tag{3.35}$$

where $a^d(\boldsymbol{\zeta})$ represents the deterministic solution corresponding to a particular realization $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)$, and n is the number of random dimensions. After that, if we apply the inner product (Eq 3.17) to the equation 3.35 we will have the following equation form for the PC coefficients:

$$a_i = \frac{\left\langle a^d(\zeta)\Phi_i \right\rangle}{\langle \Phi_i^2 \rangle} = \int_{\Omega} \cdots \int_{\Omega} \left[a^d(\boldsymbol{\zeta}) \frac{\Phi_i(\boldsymbol{\zeta})}{\langle \Phi_i^2 \rangle} \prod_{k=1}^{n} W(\zeta_k) \right] \mathrm{d}\zeta_1 \ldots d\zeta_n \tag{3.36}$$

where W is the weighting function. The above integral can be approximated using the Gaussian quadrature method, by sampling each random variable ζ_i in m Gaussian quadrature points.

$$a_i = \sum_{m_1=1}^{m} \cdots \sum_{m_n=1}^{m} a^d(\zeta_{m_1}, \ldots, \zeta_{m_n}) \frac{\Phi_i(\zeta_{m_1}, \ldots, \zeta_{m_n})}{\langle \Phi_i^2 \rangle} \prod_{k=1}^{n} \omega_{m_k} \tag{3.37}$$

With the new mathematical notations, ζ_{m_k} corresponds to the old one-dimensional random variable notation ζ_i when $k = 1 \ldots, n$. Instead, with $m_k = 1, \ldots, m$ we define the m couples $(\zeta_{m_k}, \omega_{m_k})$ that comes from sampling each one-dimensional random variable. In fact, the two elements of these m couples are respectively the Gaussian quadrature point and it's weight.

Moreover, it must be remarked that the Gaussian quadrature sampling depends on the probability distribution of the random variable. For example, if the latter follows a Gaussian distribution, the quadrature sampling will be called Hermite-Gauss quadrature. On the other hand, if the random variable follows a Beta distribution, the Gaussian quadrature sampling will be called Jacobi-Gauss quadrature.

The process involves the following steps:

1. define the behaviour of the uncertainties input variable (PDF), the random variables (in this example ζ_{m_k}) is sampled in m points using the Gaussian quadrature technique;
2. compute the basis functions Φ_i with $i = (0, \ldots, P)$ and the weights ω for each m quadrature points;
3. for each m sampled input variable, compute the deterministic solutions $a^d(\zeta_1), \ldots, a^d(\zeta_m)$;
4. the polynomial chaos coefficients are then computed with the equation 3.37

Following the indications of Le Maitre [32], the Gauss-Hermite quadrature requires $m = p + 1$ (p being the Polynomial Chaos order) sample points for each random variable to be exact, therefore the total number of deterministic solutions required is $(p+1)^n$, that are notably higher than the $P+1$ solves required by the intrusive Polynomial Chaos presented in the previous sections. This remark introduces the so called *curse of dimensionality* from which the NISP is affected: the number of deterministic solves grows exponentially with the number of random dimensions. Even though they could easily become a quite high number, they are considerably less than the solves required by Monte Carlo Method [17].

3.4.2 Non-Intrusive Point-Collocation

The Non-Intrusive Point-Collocation method (NIPC) has been proposed for the first time by Hosder et al. in [33]. This method consists in sampling $P+1$ collocation points from the random vector $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_n)$, and running the deterministic solver for each of these collocation points. In contrast to NISP, this new method does not need a Galerkin projection but it directly solves a linear system in order to find the NIPC coefficients:

$$\begin{pmatrix} \Phi_0(\boldsymbol{\zeta}_0) & \Phi_1(\boldsymbol{\zeta}_0) & \cdots & \Phi_P(\boldsymbol{\zeta}_0) \\ \Phi_0(\boldsymbol{\zeta}_1) & \Phi_1(\boldsymbol{\zeta}_1) & \cdots & \Phi_P(\boldsymbol{\zeta}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_0(\boldsymbol{\zeta}_P) & \Phi_1(\boldsymbol{\zeta}_P) & \cdots & \Phi_P(\boldsymbol{\zeta}_P) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_P \end{pmatrix} = \begin{pmatrix} a^d(\boldsymbol{\zeta}_0) \\ a^d(\boldsymbol{\zeta}_1) \\ \vdots \\ a^d(\boldsymbol{\zeta}_P) \end{pmatrix} \tag{3.38}$$

As we can see, the vector on the right hand side contains the deterministic results computed for each sample input, and to find them we need to compute the inverse of the polynomial chaos matrix.

$$\vec{a} = \mathbf{\Phi}^{-1}\,\vec{a^d} \tag{3.39}$$

Nevertheless its solution is non unique, it depends on the sampling of the $P+1$ collocation points.
Sampling can be performed following several procedures. Hosder in [34] studied three widely used techniques:

- Random Sampling: this is the easiest technique, but the results obtained are not accurate [34].
- Latin Hypercube Sampling: this is an algorithm that divides in $P+1$ sections the cumulative density function, and randomly selects one point for each section. In this way, all the portions of the input range are represented. We have to specify that this technique is more accurate than the random ones [35].
- Hammersley Sampling: this algorithm is based on the prime numbers, and its output results unique [36].

The number of function evaluations needed with NISP and NIPC is plotted against the number of random variables and the PC order in figure 3.5.

For a small number of random variables the evaluations required by the two methods are comparable, but there is a notable difference when n increases. This constitutes the great advantage of the NIPC method, that is not being subjected to the *curse of dimensionality*: in fact, it needs only $P+1$ deterministic solves, with P computable from equation 3.21. Oversampling is also possible: in this case the system 3.38 is solved using the least squares method. Hosder in [34] showed that with $2(P+1)$ collocation points instead of the required $P+1$, NIPC gives more accurate results, however the computational cost increases.

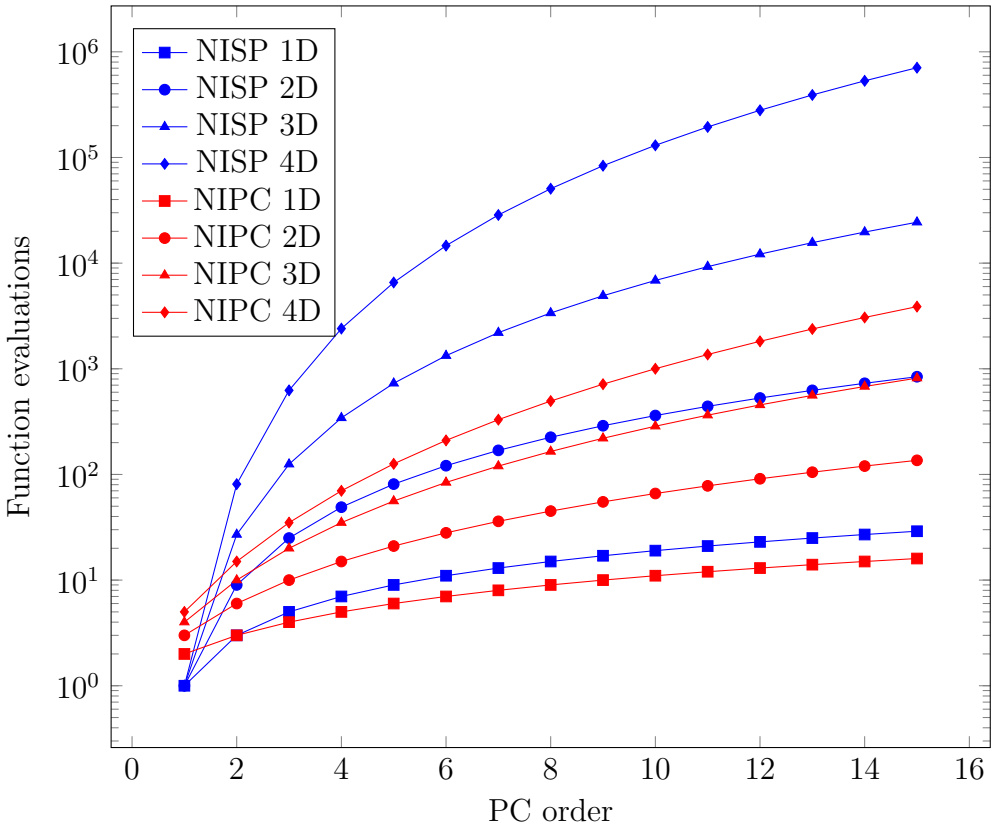


FIGURE 3.5: Function evaluations needed with NIPC and NISP methods. Source: [34].

3.4.3 Sparse Grid

This method has been proposed by [37] to alleviate the *curse of dimensionality*. It is based on the Smolyak algorithm that strategically approximates multivariate functions by tensor products of univariate interpolating formulas, avoiding complex multivariate rules. A grid of points (*sparse grid*) is then constructed and the PC coefficients are evaluated at those points. The Sparse Grid method has been successfully used in conjunction with generalised Polynomial Chaos by Walter in [38]: it has shown a fully exponential convergence rate for many test functions with 10 dimensions, allowing a fast calculation of the PC coefficients. If the problem deals with many uncertain variables, therefore, Sparse Grid constitutes a promising methodology to be integrated with PC for uncertainty quantification.

Chapter 4

Optimisation Algorithms

In chapter 2 it was remarked that two of the three main processes, *results evaluation* and *new design creation*, are performed by an optimisation algorithm. In fact, the aim of the algorithm is to decide how to change the decision variable (design variables input) in order to find better solutions. There are different typologies of algorithm, and according to EL-Sherbeny, [39], they can be classified into:

- **exact algorithms**: they look for the exact mathematical solution. Most of the time, these algorithms are not easy to be implemented, and the difficulties increase with the increment of the complexity of the problem. Another problem is the time they required in order to be programmed.
- **heuristic algorithms**: they look for an immediate and approximated solution that can be considered satisfactory for practical purposes. The problem of this kind of algorithm is that the solution is usually coarse and so, improvable. An example is the *trial and error* procedure;
- **metaheuristic algorithms**: In this algorithm the decision of how to change the decision variable is guided by a strategy, that explores the design space, and that tries to be as efficient as possible.

As said at the end of chapter 2, it does not exist a solution point (i.e. a design) that minimizes two objective functions at the same time. As a consequence, it is necessary a *decision maker* that express a preference between the two objective functions or choose a compromise.
According to Miettinen [40], the role of the decision maker in multi-objective

optimisations allows another classification of the available algorithms present in literature:

- **no-preference methods**: the decision maker does not play a role, i.e. the compromise is selected randomly, which basically means without any strategy.
- **a priori methods**: the decision maker selects the compromise solution a priori, so without waiting the results of the simulations. However, the main problem with this type of method is that it may have too optimistic or too pessimistic expectations, because the decision maker unknowns the results.
- **a posteriori methods**: in this typology of algorithm the decision maker makes an overview of the results before choosing the solutions. In order to do that, it is created a Pareto-optimal set, as it is possible to see in the figure 2.2. The main advantage is the acquisition of consciousness in the choice, but the drawback is the computational time that is requested.
- **interactive methods**: in these approach the decision maker is allowed to interact with the process while it is running, by expressing preferences and 'guiding' the algorithm to the desired direction.

This work is focused on *a posteriori*, *metaheuristic methods*. The most common families of these methods include simulated annealing, genetic algorithms and Multi-Objective Tabu Search.

4.1 Simulated Annealing

This concept has been proposed for the first time by Kirkpatrick in 1983 [41]. The idea comes from the observance of the physical behaviour of a metal when it cools down from the liquid state and undergoes a solidification process. If the process is slow enough, the crystals tend to go towards a configuration of minimal energy. The Simulated Annealing method exploits this concept employing an algorithm where a new solution is searched starting randomly from the current state: if the new solution is better than the current, it is automatically accepted. Otherwise, it is accepted with a certain probability (the 'temperature') that is decreased with time. In this way, bad solutions are accepted with less and less probability, and it

allows the algorithm to converge to the good solution. Simulated Annealing has a very slow convergence rate, therefore it is inadequate for the current study that is characterised by many time-consuming CFD analyses.

4.2 Genetic Algorithms

Presented by Mitchell [42], the algorithms of this family try to reproduce an evolution of a population according to the Darwin theory. They are based on the principle that strong parents produce best children. Therefore, the strongest members of the actual population are selected and *crossed over* to generate children that ideally will be stronger than the parents. The strongest available children are then taken as parents for the next iteration and the procedure is repeated until a very strong population (the optimal solution) is generated. To better explain the concept, we consider the following example taken from [42].
It is supposed a string of 8 bits to be the variable of interest: the objective is to form a string of all ones. A genetic algorithm to solve the problem can be performed as follows:

1. Start with a *random* generation of an initial population. In this example, suppose the latter is composed by four individuals:
   A = 11101110;
   B = 10100101;
   C = 00010000;
   D = 01000100.

2. Evaluate the *fitness* of the current individuals (corresponding to the objective functions evaluation). Here the fitness is represented by the quantity of ones in the string, therefore:
   f(A) = 6;
   f(B) = 4;
   f(C) = 1;
   f(D) = 2.

3. *Select* two candidates couples that will form the parents. The probability of selection increases with increasing fitness. Suppose that here the couples A/B and B/D are selected.

4. *Cross over* the two couples of parents to generate two couples of children. The cross over takes place with a user-defined probability in a randomly chosen bit: this action exchanges the digits before and after the chosen bit. Suppose that here both couples are crossed over: couple A/B after the fifth digit to form E/F and couple B/D after the second digit to form G/H. The new population will be:
   E = 11101101;
   F = 10100110;
   G = 10000100;
   H = 01100101.

5. *Mutate* the new population at each locus with a user-defined probability, usually very low. In this example the mutation is represented by the flip of the digit from 0 to 1 or vice versa. Suppose that in the population E, F, G, H only the last bit of individual H is mutated to 0. Therefore H now is 01100100 = H' and the final population is:
   E = 11101101;
   F = 10100110;
   G = 10000100;
   H' = 01100100.

6. The fitness of the new population is: f(E) = 6;
   f(F) = 4;
   f(G) = 2;
   f(H') = 3.
   Note that, even though in the first two individual the fitness has not changed, the global average has increased from 13/4 to 15/4. The iteration of the process from step 3 will bring, after a certain number of repetitions, all four strings to be 11111111.

A more complex and engineering-based discussion of the genetic algorithm goes beyond the scope of this work. For more details, the reader should refer to [42].

Applications of genetic algorithms in multi-objective optimisations can be found in [13, 43–48].

4.3 Tabu Search

The Tabu Search was first proposed by Glover in 1989 [49]. It is an efficient method that exploits the usage of three types of memories to intelligently explore the design space. Recently visited points (i.e. recently designs) are not allowed to be selected as next move (flagged as *tabu*) to prevent cycling moves. It has been proven to be accurate and efficient [50], along with having already been carried out by [15, 16, 51–53]. In this work, a multi-objective optimisation using Tabu Search has been performed.
Several variants of the Tabu Search exists in literature: the version used here has been proposed by Kipouros et al. [50] and implemented in the software *Multi-Objective Tabu Search* (*MOTS*). As stated at the beginning of the section, three types of memories are used during the search:

- **Short Term Memory**: it contains all the recently visited points that are marked as *tabu*, in the sense that they cannot be visited again.
- **Medium Term Memory**: it contains the current Pareto Front points (i.e current Pareto-optimal set). This is the starting point for the *intensification move*: if the current search is not giving good solutions, one point from the actual Pareto-optimal set is selected as the next point, therefore focusing the search in promising zones.
- **Long Term Memory**: it contains all the visited points since the beginning of the algorithm. It is accessed when a *diversification move* is required: if the intensification moves are not successful, the search is moved towards other regions of the design space.

The Tabu Search algorithm starts from an initial design provided by the user, then a Hooke and Jeeves (H&J) move [54] is performed to explore the design space. If one new point is better than the current, it is automatically accepted as the next point. If two or more new points dominate the current, one of them is randomly selected as the next move. If none of them dominates the current, the search is

directed towards different zones and the comparison is performed again. A third move, the *step-reduction move*, is performed if neither the intensification nor the diversification have been successful. In fact, the search starts again from one point of the current Pareto-optimal set (i.e current Pareto Front) and the step size of the H&J move is reduced, in order to further improve the optimal set.
A visual description of the H&J move and the usage of the memories is shown in figure 4.2. On the other hand, the complete flow chart of the MOTS algorithm, taken from [50], is shown in figure 4.1. The latter has been simplified here. In fact the real one contains checks on the step size reduction and on the diversification move, to prevent too big changes.

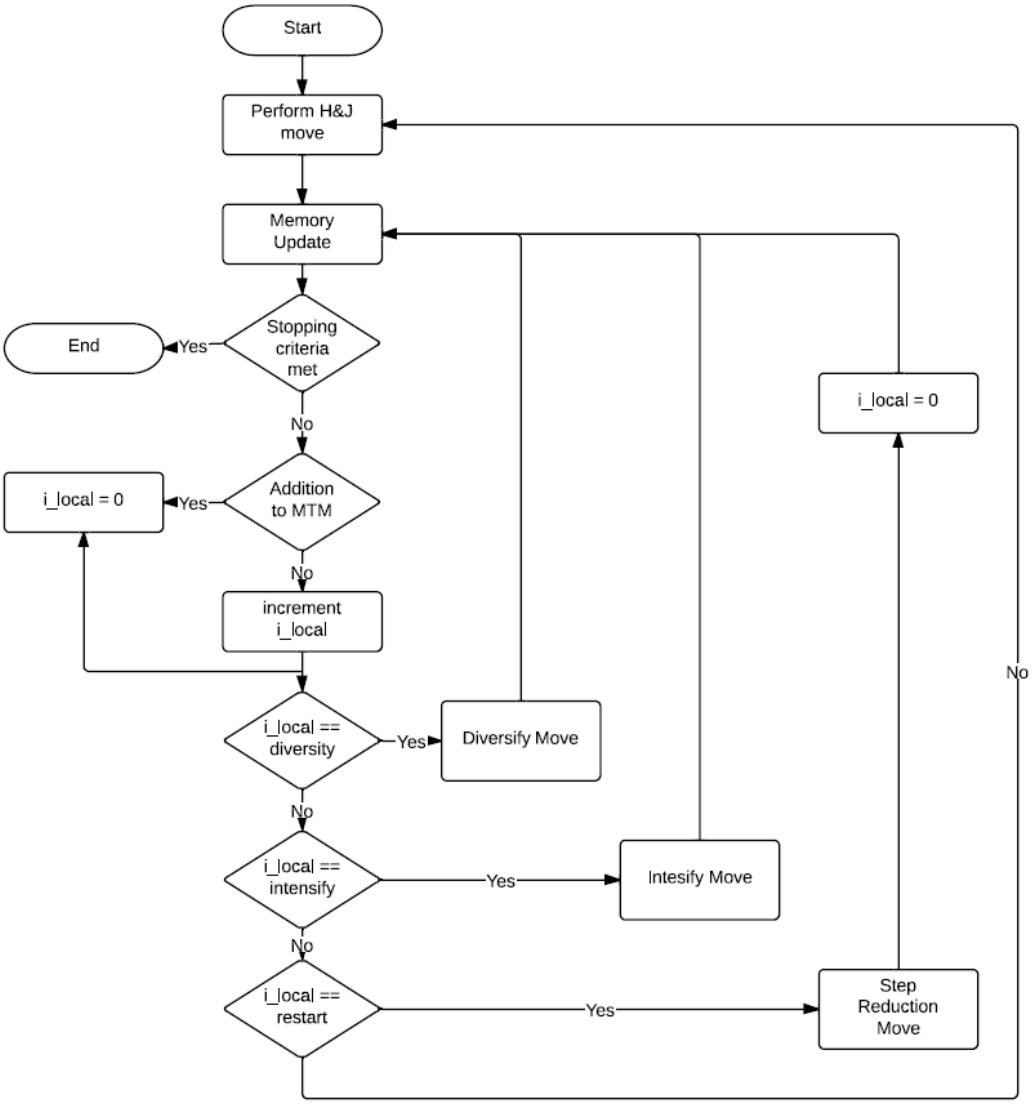


FIGURE 4.1: The MOTS flow chart. Source: [50].

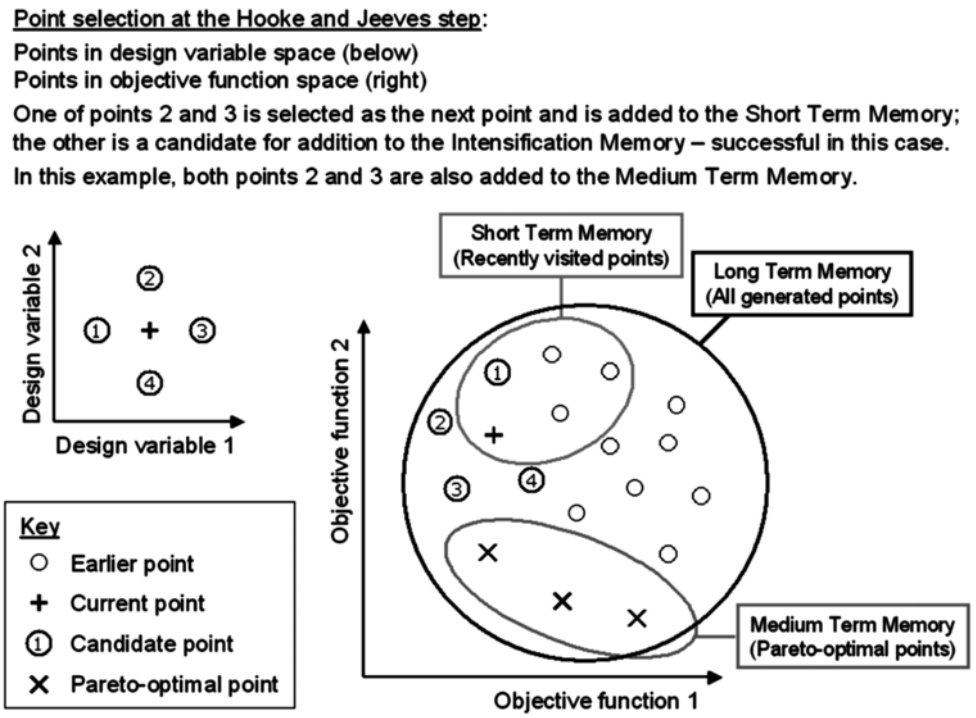


FIGURE 4.2: The MOTS memories and the H&J move. Source: [50].

4.4 Selected software description

The algorithm used here is coded in Python and allows the user to easily integrate its modules for the constrain handling and the objective function evaluation, which is crucial to the integration of non-intrusive uncertainty quantification techniques such as Non Intrusive Spectral Projection and Non Intrusive Point Collocation. In order to perform an optimisation in a reasonable amount of time, even when the evaluation procedure is performed by a set of complex and time consuming tools such as ANSYS ICEM and ANSYS Fluent, the MOTS software used here integrates the multi-process approach through the Master and Slaves paradigm. The brain of the optimiser is located on the Master process. It chooses the moves to be performed and the configurations to be evaluated. On the other hand, the multiple Salves performs the work of setting up the evaluation environment for their specified configuration and run ICEM and Fluent (which is effectively the most time consuming step of an optimisation loop). Then it returns the objective functions to the Master that decides what next move should be. This kind of scheme is called *Functional Decomposition*. This method is also combined with the usual *Domain Decomposition* used by Fluent to perform a CFD simulation on

multiple cores. In the best case, the optimisation uses $n_slaves \times n_fluent + 1 cores$ in the server.

Chapter 5

Case Study: S-duct optimisation with uncertainties

The aim of this work is to improve the robust optimisation of a S-duct made by Davide Dal Magro [3]. In the previous work, Dal Magro focused his studies using as uncertainty input the inlet velocity, as objective functions to minimise, the CP_{mean} and the CP_{stdv} (i.e. $CP_{standard-deviation}$), and finally as constraints the Swirl mean and the Swirl standard deviation. Now instead, the optimisation problem has been improved with more combinations of uncertainties input and output, that we will better explain later in this chapter. The optimization, and in particular the uncertainties quantification, is made with two different non intrusive techniques: NIPC and NISP, in order to compare the final results and to not focus the studies on only one uncertainties quantification method. The geometry of the baseline, the parameterisation, the mesh, the CFD parameters and all the optimisation problem implemented will be analysed in this chapter.

5.1 Baseline geometry configuration

The baseline geometry configuration is defined as the starting design point of the optimisation cycle. The first geometrical S-duct model, implemented as baseline configuration, was designed by Wellborn et al. [6] and reviewed by Delot [11] and D'Ambros [2] in order to simplify the parametrisation of the geometry.

5.1.1 Wellborn geometry

The duct centerline is defined by two planar circular arcs with same radii R, and subtended angles $\theta_{max}/2$. Its coordinates are defined by the following equations: For $0 \leq \theta \leq \theta_{max}/2$

$$\begin{aligned} x_{cl} &= R\sin\theta \\ y_{cl} &= R\cos\theta - R \\ z_{cl} &= 0 \end{aligned} \tag{5.1}$$

For $\theta_{max}/2 \leq \theta \leq \theta_{max}$

$$\begin{aligned} x_{cl} &= 2R\sin\theta_{max}/2 - R\sin(\theta_{max} - \theta) \\ y_{cl} &= 2R\cos\theta_{max}/2 - R\cos(\theta_{max} - \theta) - R \\ z_{cl} &= 0 \end{aligned} \tag{5.2}$$

All cross-section perpendicular to the centerline are circular with radius defined as follow:

$$\frac{r}{r_1} = 1 + 3\left(\frac{r_2}{r_1} - 1\right)\left(\frac{\theta}{\theta_{max}}\right)^2 - 2\left(\frac{r_2}{r_1} - 1\right)\left(\frac{\theta}{\theta_{max}}\right)^3 \tag{5.3}$$

where r_1 and r_2 are the inlet and the outlet radius respectively. Both centerline and radius distribution are a function of the angle θ. The value of θ_{max}, R, r_1 and r_2 reported in table 5.1.

TABLE 5.1: Wellborn S-Duct baseline geometry parameters.

Parameter	**Value**
θ_{max}	$60°$
R	$102.1\ cm$
r_1	$10.21\ cm$
r_2	$12.57\ cm$

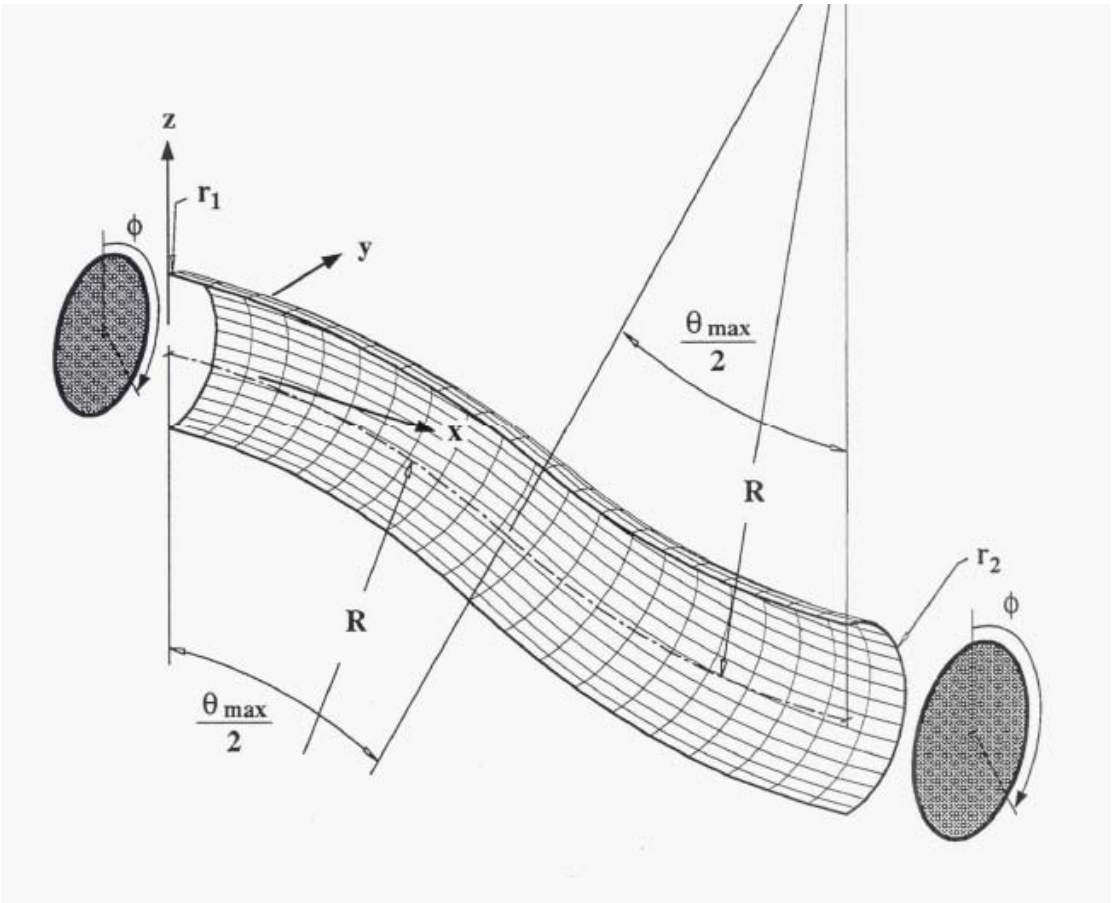


FIGURE 5.1: Wellborn baseline geometry. Source: [6].

5.1.2 Delot geometry

This baseline geometry is similar to the Wellborn one. In fact, they have the same mathematical construction but different values of θ_{max}, R, r_1 and r_2. Their values are the same of the Delot experiment [11] and they are reported in table 5.2. In

TABLE 5.2: Delot baseline geometry parameters.

Parameter	Value
θ_{max}	$60°$
R	$0.6650\ m$
r_1	$0.0665\ m$
r_2	$0.0820\ m$

order to increase the uniformity in the inlet condition and to obtain more accurate results, two additional parts have been introduced by [55]:

1. at the inlet, a cylindrical duct eight times longer than the inlet radius. Its purpose is to ensure uniform inlet conditions.

2. at the outlet, a cylindrical duct six times longer than the outlet radius. Its purpose is to guarantee that the outlet conditions do not have any influence on the upstream flow.

Figure 5.2 represents a section of the overall baseline geometry in the $x-y$ plane, the symmetry plane of the duct. The values of the parameters have been reported in table 5.3.

TABLE 5.3: Modified Delot baseline geometry parameters.

Parameter	**Value**
$Offset$	$60°$
L_{S-duct}	R
L_{inlet}	$8r_1$
L_{outlet}	$6r_1$
$L_{AIP} = L_{inlet} + L_{S-duct} + r_1$	$9r_1 + R$
$L_{Tot} = L_{inlet} + L_{S-duct} + L_{outlet}$	$14r_1 + R$

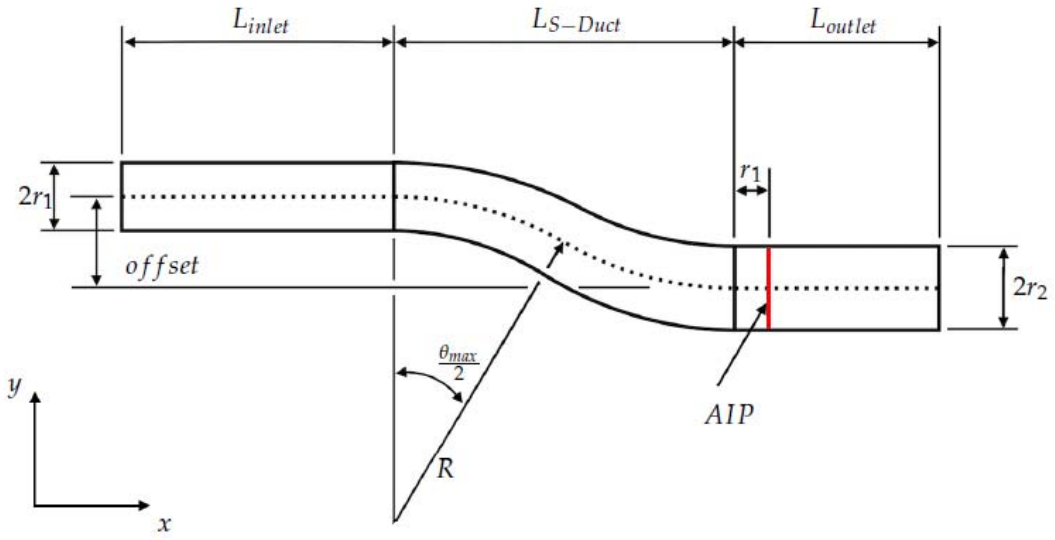


FIGURE 5.2: Modified Delot baseline geometry. Source: [2].

In this thesis, the Modified Delot Baseline geometry has not been used as starting point for all the implemented robust optimisation cycles. In fact the true starting Baseline geometry used in this work is a little bit different from the one described above (i.e Modified Delot Baseline). This difference is due to the new parametrisation adopted by D'Ambros in his work [2].

5.2 Geometry parametrisation

The purpose of parametrisation is to reduce the number of geometric parameters (decision variables) necessary to draw the geometry, in order to reduce the overall optimization computational cost. Furthermore, parametrisation should also allow an efficient modification of the shape of the S-Duct. As it is reported in [55], the FFD (Free Form Deformation) is the method employed to parametrise and deform the baseline geometry. In general, it consists of embedding the considered geometry into a 3D parallelepipedic lattice regularly subdivided which, nodes are called control points (Figure 5.3).

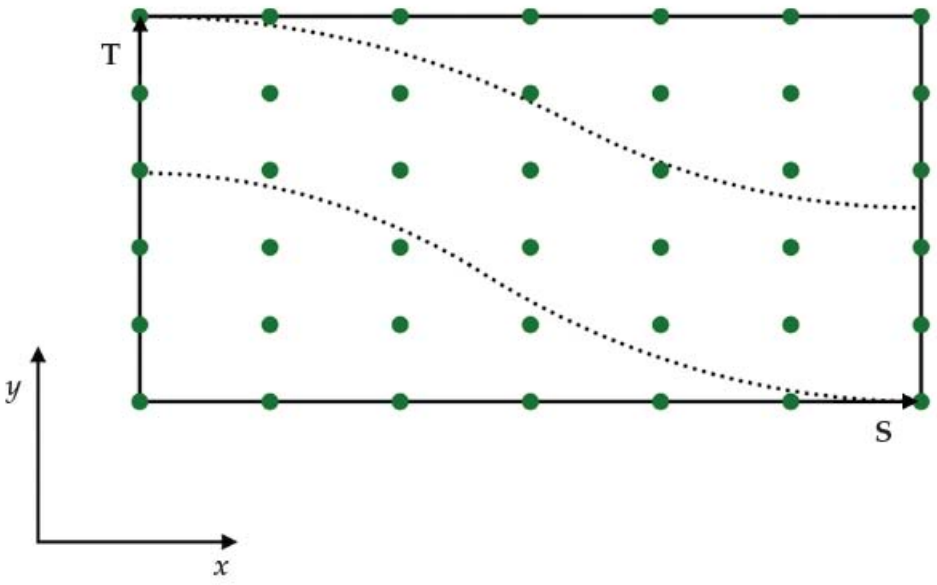


FIGURE 5.3: S-duct parallelepipedic lattice (Dotted lines represent the duct projection). Source: [2].

The position of each point of the geometry is mathematically described by a weighted sum of the control points position as follow:

$$X_{ffd} = \sum_{i,j=0}^{l,m} B_i(s) B_j(t) P_{ij} \tag{5.4}$$

where:

- X_{ffd} is a vector containing the Cartesian coordinates of the S-duct displaced point.

- l, m are the numbers of control point in S and T direction respectively.
- $B_k(u)$ are the Bernstein polynomials of degree 3.
- s, t are the generic coordinate in the $S-T$ system of reference $(0 \leq s \leq 1, 0 \leq t \leq 1)$.
- P_{ij} is a vector containing the Cartesian coordinates of the control point.

In this work and also in [2, 3], the general FFD method used in [55], was simplified and modified as follow:

- we decide to design and simulate only half of the S-duct in order to reduce the computational cost. In fact Wellborn [6] and Delot [11] demonstrated that the stream flow is symmetric respect to the $x-y$ plane.
- we consider the cylindrical ducts added after and before the S-Duct, as manufacturing constraints. This means that the only part to be parametrised is the S-Duct itself.
- Since there is nothing inside the duct, the best position for the control point would be on the surface of the S-Duct. Following this reasoning, and working with the Modified Delot's surface geometry, $l = 7$ equally spaced semicircular cross-section perpendicular to the centerline were defined and, on each of them, $m = 6$ equally spaced control points were placed. In this way, this solution does not represent a parallelepipedic lattice.

The main problem of placing the control points in this new way occurs when the FFD is performed (i.e. point control interpolation, Equation 5.4). The result surface we obtain is similar but not equal to the Modified Delot surface (Figure 5.4). To obtain a closer baseline result to the Modified Delot geometry, D'Ambros modified the control points position as follow:

- In every cross-section, the deformed geometry is described by a Bezier interpolation curve.

$$X_{ffd} = \sum_{i=0}^{m} B_j(t) P_i \tag{5.5}$$

  Fixed $m = 6$, and imposing the following constraints:

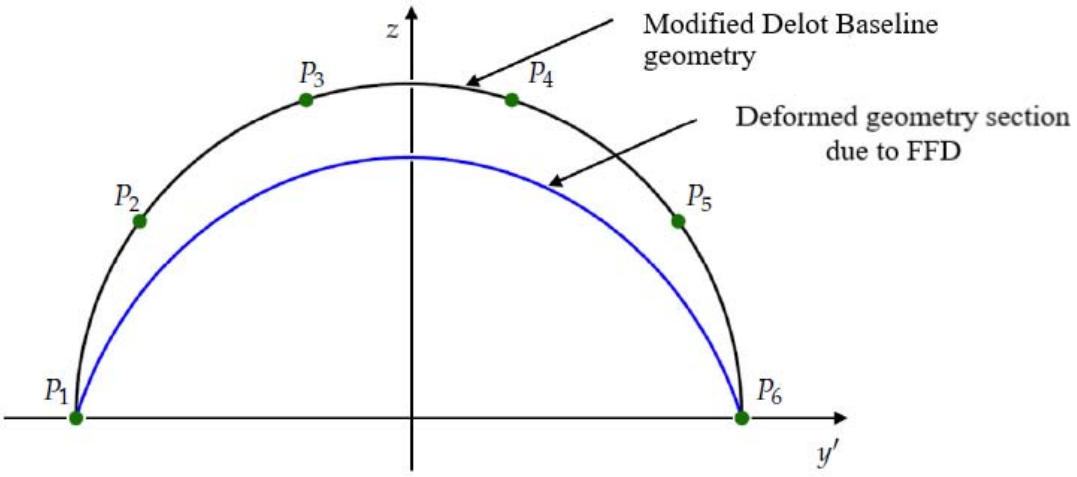


FIGURE 5.4: Generic cross section ($P_1 \dots P_6$ control points). Source: [2].

* $y'_{P_1} = -r$
* $y'_{P_1} = y'_{P_2}$: tangency condition
* $z_{P_3} = z_{P_4}$: symmetry condition
* $z_{P_2} = z_{P_5}$: symmetry condition
* $y'_{P_3} = -y'_{P_4}$: symmetry condition
* $y'_{P_5} = y'_{P_6}$: tangency condition
* $y'_{P_6} = r$

where r is the semicircle radius in each cross section, D'Ambros inverted equation 5.5 in order to find the control points position that interpolate the Modified Delot cross sections as near as possible. After some calculations he obtained:

$$z_{P_2} = z_{P_5} = r\frac{4(8\sqrt{2} - 9)}{15} \tag{5.6}$$

$$z_{P_3} = z_{P_4} = r\frac{2(21 - 8\sqrt{2})}{15} \tag{5.7}$$

$$y'_{P_3} = -y'_{P_4} = r\frac{2(64\sqrt{2} - 79)}{45} \tag{5.8}$$

- In order to guarantee tangential condition at the inlet and at the outlet, D'Ambros copied and translated the control points in the inlet section l_1 shortly after, and the control points in the outlet section l_7 shortly before. This means that cross sections l_1, l_2 and cross sections l_6, l_7 are identical.

As we said, in previous works [55] the parametrisation of the same baseline geometry was performed with a 3D parallelepipedic lattice, and this method allowed to recreate the precise baseline geometry. Now instead, with the parametrisation of D'Ambros, even tough the starting baseline is similar but not identical to the Modified Delot's one, this new parametrisation method permits to modify the geometry of the S-duct with more accuracy compared to the other previous works.

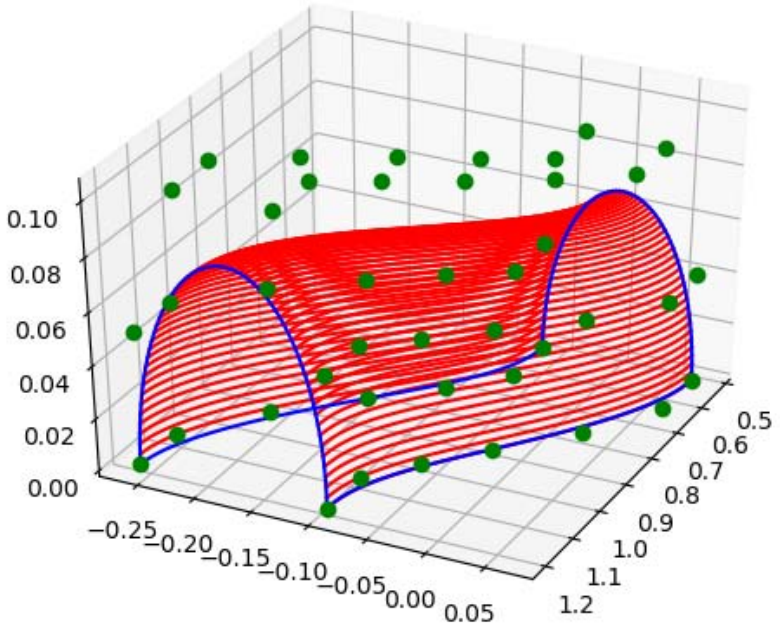


FIGURE 5.5: D'Ambros Baseline geometry.

5.2.1 Parametrisation Degrees of Freedom

The degrees of freedom (dof) of the new parameterization can be defined as follow [2]:

- The control point in the first two cross-section from the S-Duct inlet and the last two before the outlet are fixed. This is due to manufacturing constraints.
- in every other cross-section we have:

- Point on the symmetry plane (P_1, P_6) can only move on the symmetry plane $x - y$ (2dof).
- To maintain tangency condition, point P_2 and P_5 have the same x and y coordinates as P_1 and P_6 respectively. They can move only in z-direction (1dof).
- point P_4 and P_5 can move in all the space (3dof)

This means that every cross-section have 12 dof. In previous work [55], 36 dof were imposed. Therefore, to maintain the same number, three cross-sections between the two fixed section at the inlet and outlet were imposed in our parametrisation. This is why previously we set $l = 7$. All the 36 parameters described above are free to move inside an imaginary box that encloses the S-Duct:

* x-direction: between S-Duct inlet and outlet.
* y-direction: $[-10.5r_1,\ 9r_1]$.
* z-direction: $[-4.5r_1,\ 9r_1]$.

5.2.2 Control Points constraints

In addition to the parametrisation constraints, the following were defined in order to avoid infeasible geometry during optimization:

1. for line upper (UP) and lower (DW) curves in the symmetry plane:

$$y_{UP}(x) > y_{DW}(x) \tag{5.9}$$

2. if $y_{P_4} < y_{P_3}$:

$$y_{P_4} - y_{P_3} < r_1 \tag{5.10}$$

3. with $X_{P_j}(i)$ we indicate the j control point x-range in the generic i cross-section:

$$X_{P_j}(i-2) \leq X_{P_j}(i) \leq X_{P_j}(i+2) \tag{5.11}$$

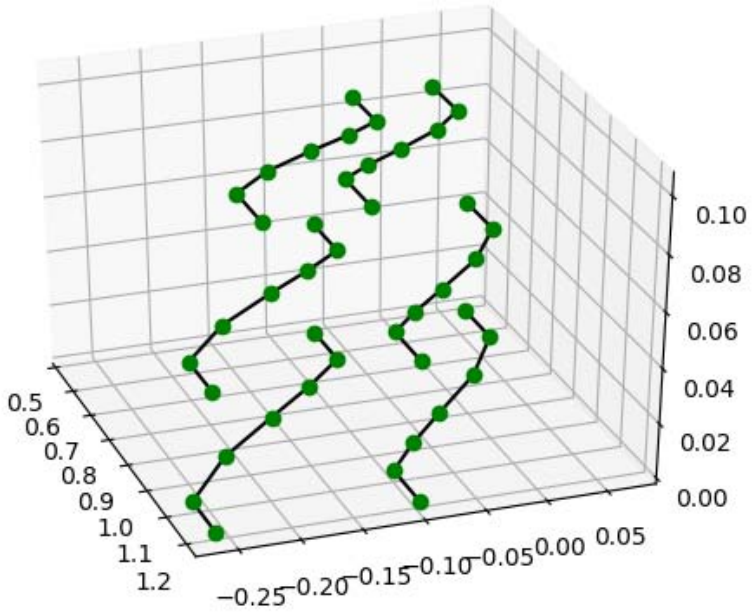


FIGURE 5.6: Control point starting position.

5.3 Flow Simulation

The analysis that we carried out is a steady state RANS simulation, with the ANSYS Fluent solver. The $K-\omega$ SST model was adopted in the optimization study since it provided similar results at a reasonable computational cost, compared to the four-equation transition SST model, that is the best match with experimental data [2]. During the optimization, the simulations were carried out running the first 200 iterations with the first order of solution accuracy for all the flow parameters. For the next 500 iterations all the parameters were set to the second order. A total of 700 iterations was performed in order to secure every residual below 10^{-5}.

The boundary conditions are the same applied by Delot [11] and also in D'Ambros e Dal Magro [2, 3]. However, the main difference is that in this thesis we introduce more than one uncertainty variables that will be discussed later. The boundary conditions of Delot are reported in table.

TABLE 5.4: Delot S-duct boundary conditions parameters.

Parameter	**Value**
Inlet total pressure	88.744 kPa
Inlet static pressure	69.575 kPa
Outlet static pressure	78.982 kPa
Total temperature	286.2 K

5.4 Mesh

To built the mesh, it was decided to emulate the mesh created by Dal Magro [3] in order to extend and improve his robust optimisation study and to compare the final results. It was decided to adopt the software ICEM. The mesh was created with the same topology of [11], and D'Ambros changed the number of nodes. D'Ambros noticed that as the number of mesh cells increased, the results of the CFD simulations were getting closer and closer to the experimental result. Moreover, with a number of cells greater than 1.7×10^6, the numerical results seem to start to oscillate around an average value as we can see in figure 5.8. Thanks to this, D'Ambros chose to set the number of celle to 1.8×10^6. For every new geometry created by the optimiser, every mesh shares the same general properties in order to guarantee comparable results. An H-grid structure was imposed in the center of the duct section and an O-grid structure around the walls (Figure 5.7).

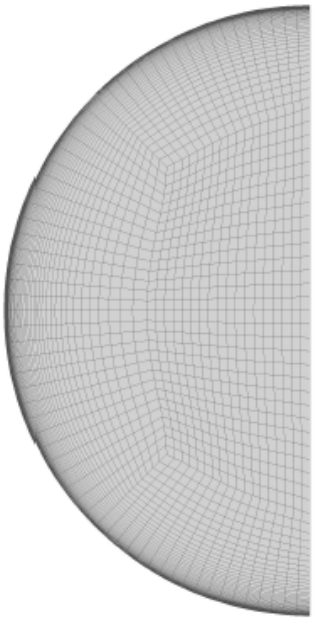
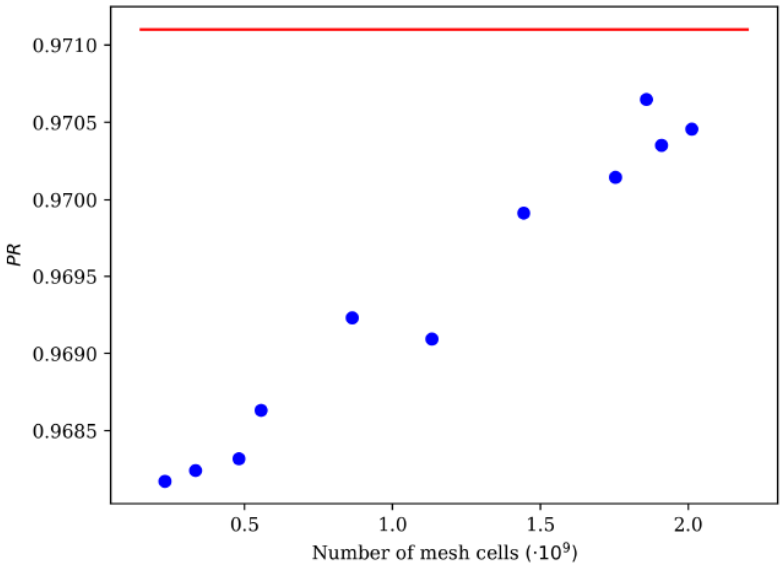
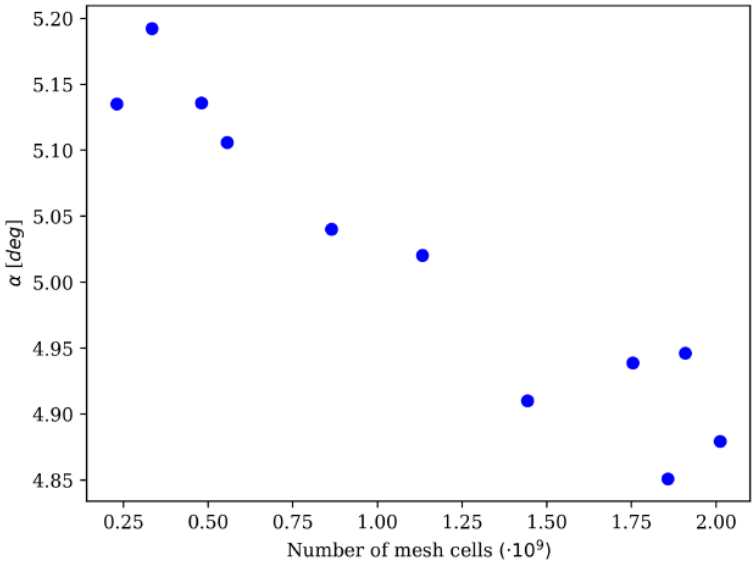


FIGURE 5.7: Cross-section mesh topology. Source: [2].

The first layer thickness on the wall was imposed to ensure that the y^+ would be smaller than 1 over the full domain: with a first layer thickness of 2×10^{-6} we obtained a maximum y^+ of about 0.8. The expansion ratio from the wall was set equal to 1.05. The number of nodes in each cross-section is approximately 6000, while the number of cross-sections is 360.



(a)



(b)

FIGURE 5.8: PR and α as a function of the number of mesh elements. The red solid line in (**a**) represent the experimental result from [55]. Source: [2].

5.5 Uncertainties variables

The first step of an uncertainty quantification is the knowledge of the behaviour of the variable, that means defining its probability density function. Thanks to the research conducted by Liatsikouras [56], one of the parameter chosen as uncertainty is the Inlet velocity. Its mean value is equal to the inlet velocity of Delot, and it follows a Gaussian distribution. The other two main uncertainties used in this work are the flux inlet deviation and the bending angle of the S-duct. The first one refers to the possible inlet velocity deviation respect to the main axial direction. It is easy to understand that this uncertainty is strictly related to the Inlet velocity and its purpose is to extend the speed range at the input of the S-duct. As we know, in reality the inlet velocity is not necessarily perfectly axial but it can be slightly deviated. The mean value chosen for the flux deviation is zero and it follows a Gaussian distribution. The last but not least uncertainty parameter is the bending angle of the S-duct, a manufacturing variable that is normally set to 60° (θ_{max}) in the baseline geometry. In literature, there are no references about the uncertainty quantification of this variable. To overcome this lack of knowledge, we supposed a Gaussian distribution for the θ_{max} variable, in fact the bending tubes technique is always affected by some manufacturing errors like the elastic springback.

5.6 NIPC and NISP Robust optimisations

The robust optimizations implemented in this thesis has been computed using two non intrusive polynomial chaos techniques:

- Non intrusive Spectral Projection (NISP).
- Non intrusive Point collocation (NIPC).

Now we analyse all the NIPC and NISP robust optimisation problems implemented:

1. One uncertainty input (Inlet Velocity), four objective functions outputs (CP_{mean}, CP_{stdv}, $Swirl_{mean}$, $Swirl_{stdv}$) and two constraints ($Swirl_{mean} <$

$5°,\ Swirl_{stdv} < 1°$).
This problem has been built with the following parameters:

- Uncertainty in the Inlet Velocity v. $n = 1$.
- $\mu_v \approx 196.53$ m/s , $\sigma_v = 10$ m/s.
- PC order $p = 3$.
- Type polynomial: Hermite Chaos (Gaussian random input)

The sampling technique used for the NIPC method is the latin Hypercube Sampling (LHS). Remembering the Non Intrusive Point Collocation theory and the Equation 3.21, $P + 1 = 4$ deterministic evaluations are necessary to find the $P + 1 = 4$ PC coefficients and to model the stochastic outputs. As a matter of fact, Inlet Velocity was sampled 4 times.
On the other hand,, for the NISP method was used the Hermite-Gauss quadrature sampling technique. To evaluate the $P + 1 = 4$ PC coefficient and find the stochastic outputs, the NISP method needs $(p + 1)^n$ deterministic solutions and $m = p + 1$ samples for each uncertainty. In this case $(p + 1)^n = P + 1 = 4$ and $m = p + 1 = 4$.

2. Two uncertainties input (Inlet velocity, Inlet flux deviation), two objective functions ($CP_{mean},\ CP_{stdv}$) and two constraints ($Swirl_{mean} < 5°,\ Swirl_{stdv} < 1°$).
This problem has been built with the following parameters:

   - Uncertainty in the Inlet Velocity v and Inlet Flux Deviation β. $n = 2$.
   - $\mu_v \approx 196.53$ m/s , $\sigma_v = 10$ m/s.
   - $\mu_\beta = 0°$, $\sigma_\beta = 3°$ m/s.
   - PC order $p = 2$.
   - Type polynomial: Hermite Chaos (Gaussian random input)

   Like before, the sampling technique used for the NIPC method is the latin Hypercube Sampling (LHS). This time, in order to find the $P + 1 = 6$ PC coefficient and to model the stochastic outputs, it is necessary to perform $P + 1 = 6$ deterministic evaluations. As a matter of fact, the Inlet Velocity and the Inlet Flux Deviation were sampled 3 times, forming 9 possible pairs (3^2). Of these nine couples, $P + 1 = 6$ has been chosen.

Instead, for the NISP method was used the Hermite-Gauss quadrature sampling technique like before. Now $(p+1)^n = 9$ deterministic solutions and $m = p+1 = 3$ samples for each uncertainty are needed to find the $P+1 = 6$ PC coefficient and to model the stochastic outputs. It is possible to observe that with more than one uncertainty input, the NISP method needs more deterministic evaluation than the NIPC method, hence the computational cost increases.

In figure 5.9, the Robust optimisation loop implemented to solve the two problem described above is shown.

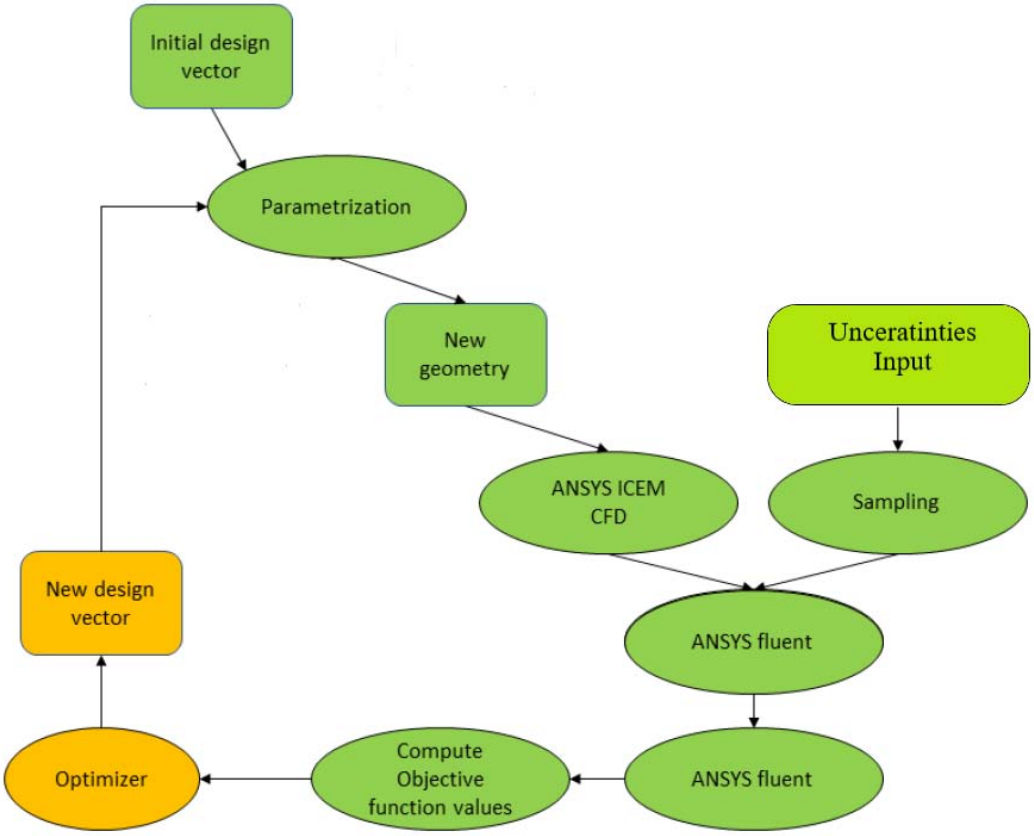


FIGURE 5.9: Robust optimisation loop scheme.

3. Two uncertainties input (Inlet velocity, Bending angle), two objective functions (CP_{mean}, CP_{stdv}) and two constraints ($Swirl_{mean} < 5.5°$, $Swirl_{stdv} < 1°$).

   This problem has been built with the following parameters:

   - Uncertainty in the Inlet Velocity v and Bending angle α. $n = 2$.

- $\mu_v \approx 196.53$ m/s , $\sigma_v = 10$ m/s.
- $\mu_\alpha = \theta_{max} = 60°$, $\sigma_\alpha = 4°$ m/s.
- PC order $p = 2$.
- Type polynomial: Hermite Chaos (Gaussian random input)

In this optimisation problem, only the NIPC method has been used to evaluate the stochastic output, and $P + 1 = 6$ deterministic solver has been performed in order to find the $P + 1 = 6$ PC coefficients. The LHS technique was performed to sample the two input uncertainties.

In figure 5.10 the Robust optimisation loop implemented to solve the problem with the manufacturing uncertainty is shown.

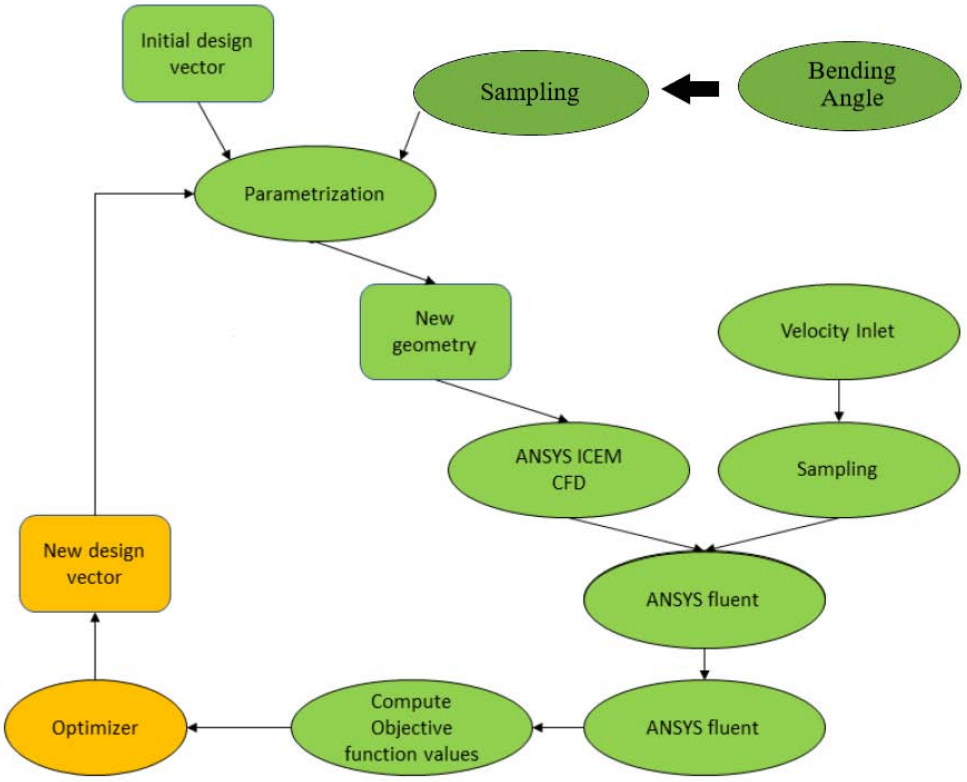


FIGURE 5.10: Robust optimisation loop scheme with manufacturing uncertainty.

Chapter 6

Results

From now on, robust optimizations with a single input uncertainty will be called 1D, while those with two input uncertainties will be labeled as 2D.
As explained at the end of Chapter 5, three robust optimisation problems plus a Machine Learning optimisation cycle were performed. The first two robust optimisation problems (1D, 2D) were both implemented with two uncertainties quantification techniques (NIPC, NISP). On the other hand, the third one, the manufacturing problem (MNF), was implemented only with the NIPC technique. During the project, a Machine Learning optimisation cycle was developed in order to collect more data from the 1D and 2D optimisation problems and speed up the research of optimum designs. In total, five robust optimisations cycles plus a Machine learning one, were implemented:

- NIPC 1D, 2D
- NISP 1D, 2D
- NIPC MNF
- ML 1D,2D

6.1 1D Robust Optimisation Problem

The 1D optimisation problem is characterised by the following starting parameters:

- **Uncertainty input:** Inlet Velocity with mean $\mu_v \approx 196.53$ m/s and standard deviation value $\sigma_v = 10$ m/s. This uncertainty variable is described with a Gaussian PDF.
- **Objective Functions:** CP_{mean}, CP_{stdv}, $Swirl_{mean}$, $Swirl_{stdv}$
- **Constraints:** The mean and the standard deviation values of the Swirl were setted as constraint. $Swirl_{mean} < 5°$, $Swirl_{stdv} < 1°$.
- **Polynomial Chaos type and order:** the typology is the Hermite Chaos polynomial and the order p is set to 3.

6.1.1 NIPC 1D

To implement the NIPC 1D cycle, the uncertainty input was sampled with the Latin Hypercube Sampling technique (LHS). The MOTS software was stopped after having successfully evaluated 324 configurations. As explained at the end of Chapter 5, the NIPC uncertainty quantification technique needs $P+1$ deterministic evaluations to find the PC coefficients and the stochastic outputs (i.e objective functions). In this case the number of evaluations were equal to four. This means that $324 \times 4 = 1296$ CFD Fluent simulations converged. The time for a single fluent evaluation is ≈ 38.5 minutes, so the time that was necessary to obtain all these results was ≈ 830 hours. However, the amount of time above estimated does not take into account all the configurations that were analysed but subsequently discarded if they did not respect the Swirl mean and Swirl standard deviation constraints or if a structured geometry mesh was not possible to be created by ANSYS ICEM. The amount of discarded design is 51.
As mentioned at the end of Chapter 2 it is not possible to represent a Pareto optimal front with more than three dimensions (i.e objective functions). To overcome this issue, the post processing analysis was also supported with the use of a Parallel coordinates software developed by Cambridge University.
To better understand the complexity of a four dimensional Pareto front plot it is possible to imagine it thinking of a cube. On each face of the cube, is plotted a bi-dimensional Pareto front: $CP_{stdv} - CP_{mean}$, $Swirl_{stdv} - Swirl_{mean}$, $CP_{mean} - Swirl_{mean}$, $CP_{stdv} - Swirl_{stdv}$, $CP_{mean} - Swirl_{stdv}$, $Swirl_{mean} - CP_{stdv}$.

Three of these six cube faces were plotted:

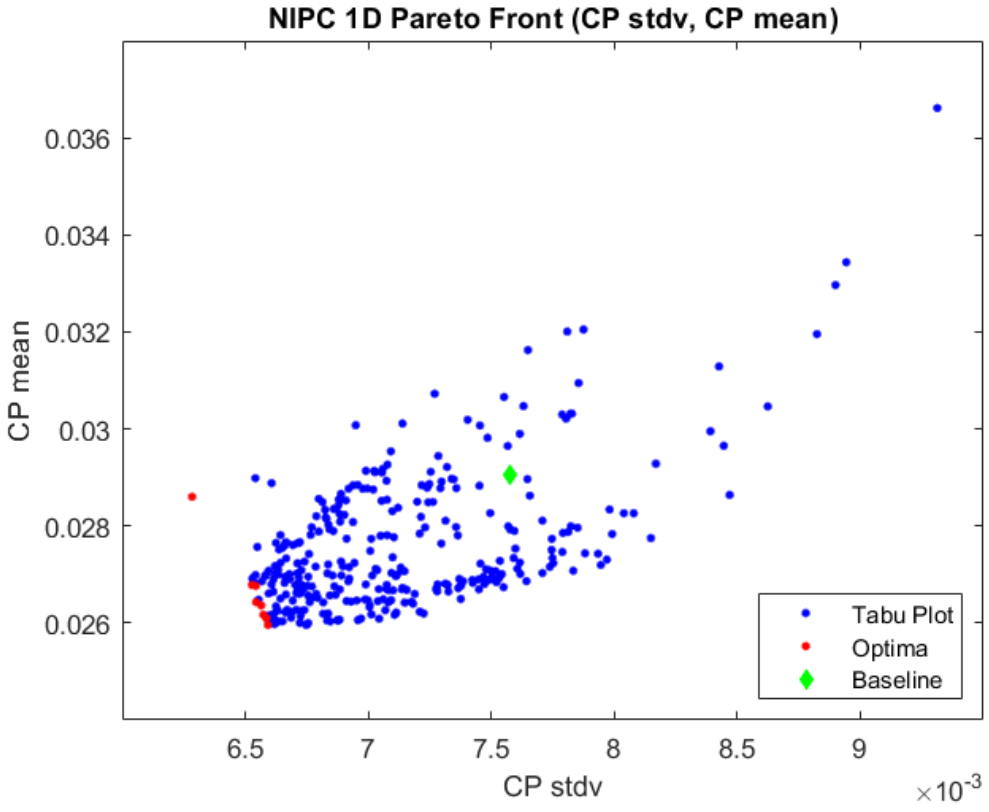


FIGURE 6.1: NIPC 1D: CP_{stdv}, CP_{mean}.

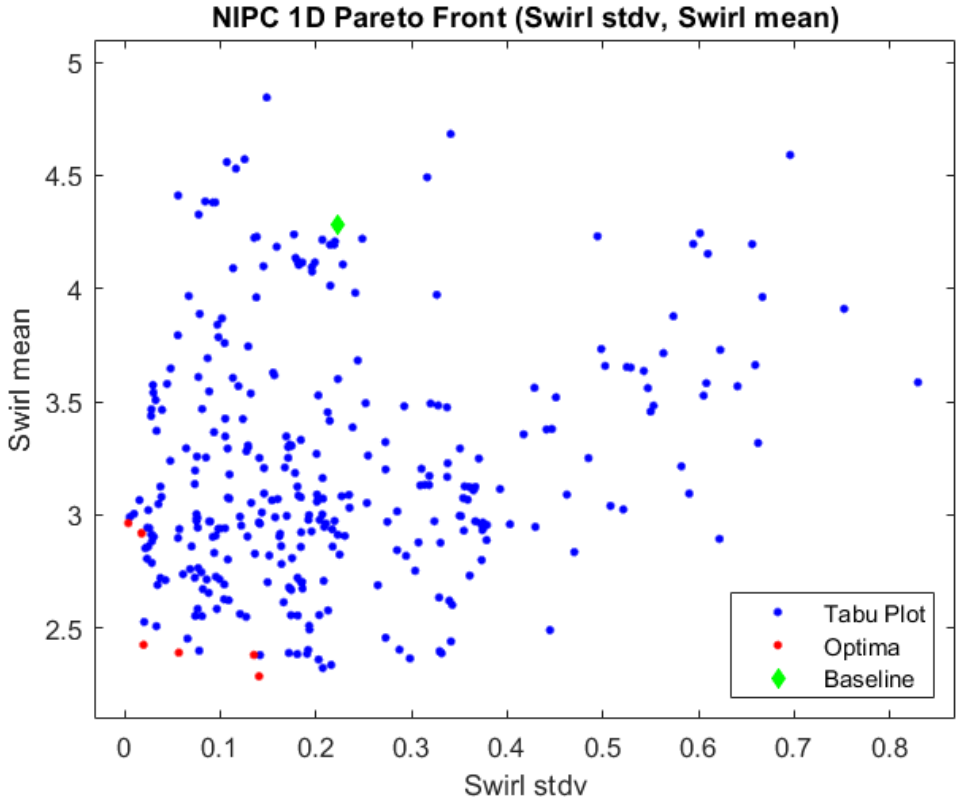


FIGURE 6.2: NIPC 1D: $Swirl_{stdv}$, $Swirl_{mean}$.

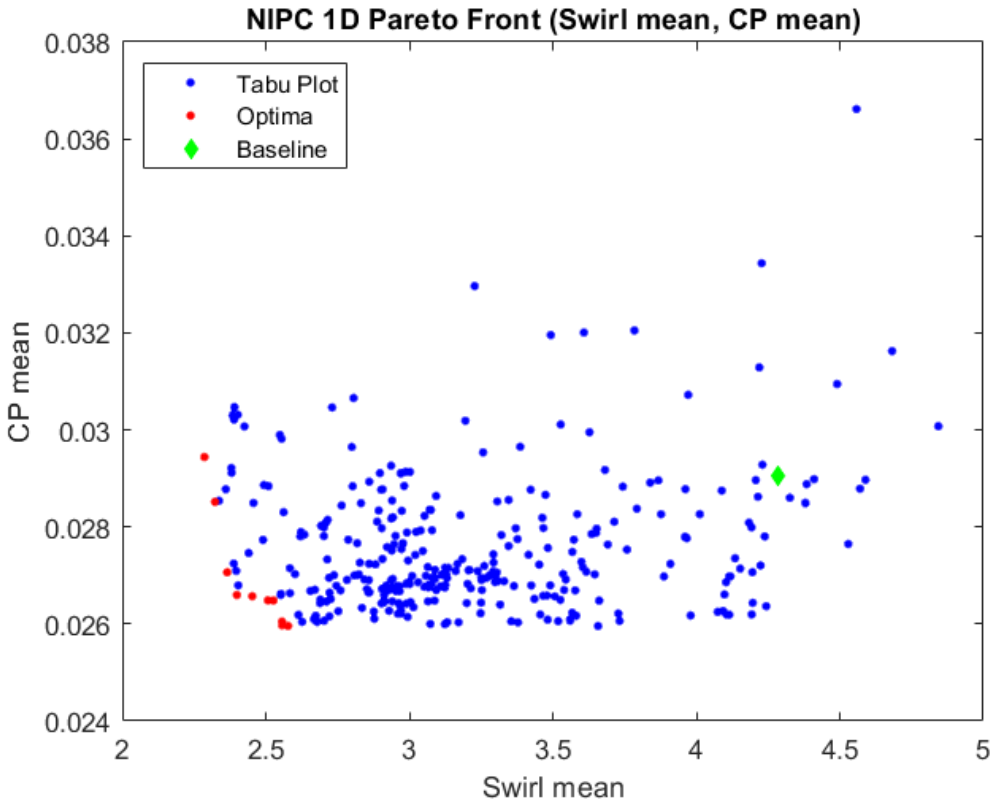


FIGURE 6.3: NIPC 1D: $Swirl_{mean}$, CP_{mean}.

In the three Pareto front figures 6.1 6.2 6.3, the bi-dimensional Pareto optima design points of each faces are highlighted in red, the starting geometry (i.e Baseline) in green and all the MOTS evaluated designs in blue.

It is important to underline that these bi-dimensional plots are not good to represent a multi-dimensional Pareto front and its optimum designs. As a matter of fact, from these three figures, only the Best design were chosen and analysed: Best CP_{mean}, Best CP_{stdv}, Best $Swirl_{mean}$, Best $Swirl_{stdv}$. These four Best design are so labelled because only one of their four objective functions has the overall minimum value. The results of the Best designs and the Baseline are shown in table 6.1:

TABLE 6.1: NIPC 1D Best design results.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Best CP_{mean}	0.025957	0.006592	2.577800°	0.212810°
Best CP_{stdv}	0.028601	0.006284	4.326070°	0.077612°
Best $Swirl_{mean}$	0.029442	0.007285	2.28655°	0.140704°
Best $Swirl_{stdv}$	0.026427	0.007081	2.963150°	0.003968°
Baseline	0.02905	0.007574	4.283636°	0.223726°

Only the use of the parallel coordinates technique can help the designer to find the designs that try to minimise all the four objective functions simultaneously: the *trade-off designs*.
In fact, thanks to the parallel coordinates plot (figure 6.21), other four designs were chosen and analysed, but only one of them was highlighted in this thesis.
The Parallel Coordinates is a very powerful tool. It was developed with the aim to give to the designer the possibility to see all the evaluated design space in all its dimensions (in this thesis 36 design parameters), and to link each design geometry (composed by 36 design parameters) with its objective functions. In other words, this tool allows to choose the designs that minimise all the four objective functions simultaneously and to verify if there are zones of the design space that have not been fully evaluated. The results of the chosen *trade-off designs* are illustrated in table 6.2:

TABLE 6.2: NIPC 1D optima designs (*trade-off designs*) results obtained with the use of the parallel coordinates.

Design	CP_{mean}	CP_{stdv}	$Swirl_{stdv}$	$Swirl_{stdv}$
optima 1	0.026596	0.006719	2.39949°	0.078154°
optima 2	0.02657	0.006755	2.45296°	0.066027°
optima 3	0.026618	0.006708	2.5533°	0.073955°
optima 4	0.026634	0.006725	2.58439°	0.076555°
Baseline	0.02905	0.007574	4.283636°	0.223726°

It is very important to specify that the output values of the objective functions, obtained from the NIPC 1D (and also for the NIPC 2D) optimisation loop, are slightly different from the ones obtained after the post-processing analysis of the chosen designs. These small differences in the results are the effect of two events:

1 The sampled velocities were not recorded during the optimisations loop.

2 The LHS method chose randomly the inlet velocity values inside the cumulative density function curve (CDF).

In other words, the sampled inlet velocities of the post-processing design analysis are different from the optimisation cycle ones. It must be remarked that the above mentioned effect does not occur in the NISP optimisation cycles because of the different nature of the Hermite-Gauss quadrature sampling. In fact, the latter chooses always the same values inside the Gaussian probability density function

(PDF). From now, until the end of the thesis, all the contour drawings will refer to the post-processing objective function values.

6.1.1.1 NIPC 1D: Best CP mean

The first configuration taken into account in this paragraph is the design that has achieved the best overall result in terms of CP_{mean}. Table 6.3 highlights the differences between the objective functions values obtained from the optimisation loop and the post-processing analysis ones. Moreover, the same table shows the impact that different inlet velocities had to the $Swirl_{mean}$ and $Swirl_{stdv}$ values.

TABLE 6.3: NIPC 1D Best CP_{mean}: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Optimisation	0.025957	0.006592	2.577800°	0.212810°
Post-proc.	0.025958	0.006738	3.657614°	0.491281°
Baseline	0.02905	0.007574	4.283636°	0.223726°

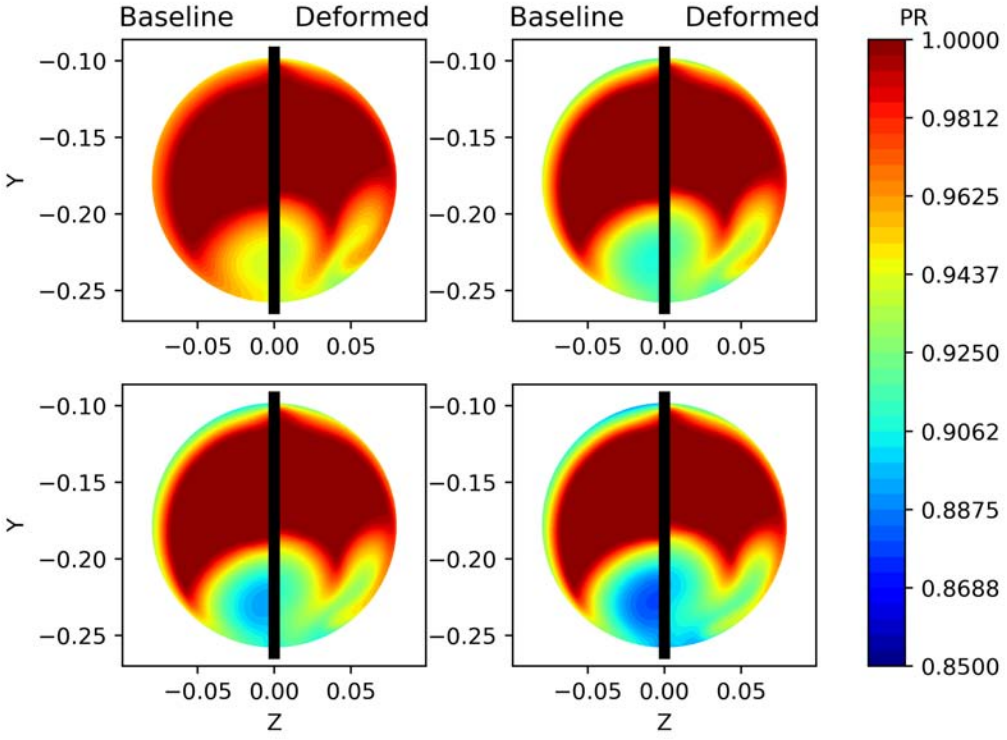


FIGURE 6.4: NIPC 1D Best CP_{mean}: PR contour comparison at the AIP.

In figure 6.4, it is possible to see the pressure recovery value (PR) evaluated at the AIP surface for each inlet velocity sample. The order is from the lowest velocity, top left, to the highest one, bottom right. The left half of each AIP contours drawing represents the pressure recovery at the AIP surface of the baseline. On the other hand, the right one represents the deformed one.
It is possible to observe that in all the four contour comparison, the values of the Pressure recovery of the deformed design are higher than the baseline values. In fact, in table 6.4, all the values of $CP = 1 - PR$ that were found for each inlet velocity sample, are reported.

TABLE 6.4: NIPC 1D Best CP_{mean}: CP value for each velocity sample.

	CP **Baseline**	**NIPC 1D Best** CP_{mean}	**Improvement**
Sample 1	0.02278	0.01932	−15.15%
Sample 2	0.024020	0.02458	+2.33%
Sample 3	0.030367	0.02678	−11.81%
Sample 4	0.040817	0.03240	−20.62%

In figure 6.4 the deformed geometry shows two zones with the lowest value of pressure recovery instead of only one present in the Baseline configuration. This phenomenon is due to the presence of a sharp lateral widening of the sections near the AIP that allows the formation of a new area of detachment of the flow (figure 6.5). In this new detachment area the pressure recovery decreases as the inlet

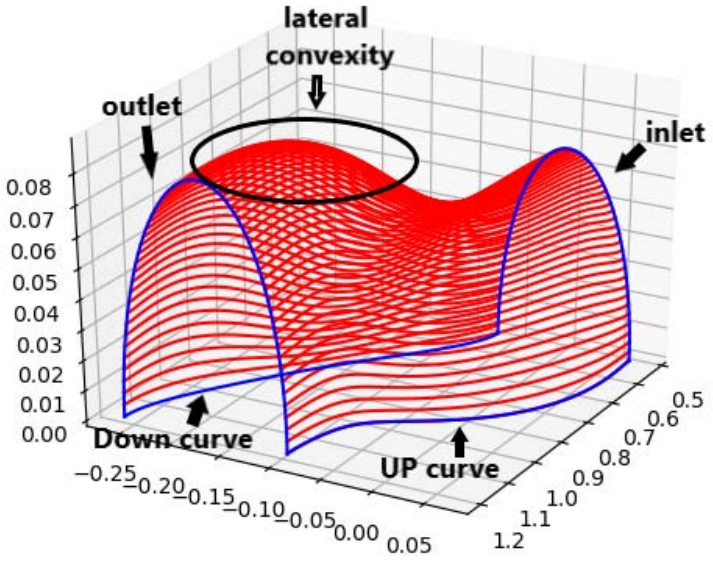


FIGURE 6.5: NIPC 1D Best CP_{mean}: Lateral convexity.

velocity increases. Moreover, the two flow detachment areas tends to become one as the velocity increases. In fact, it is possible to imagine that as the velocity increases, the flow has more and more difficulties in adhering to the side wall.
The Swirl mean and standard deviation angle, as already explained, were used in the NIPC 1D robust optimization not only as objective functions but also as constraints. The trend of the swirl in the AIP is illustrated in the figure 6.6. The differences between the deformed geometry and the baseline are not so evident. It can be observed that in the deformed design, as velocity increases, the secondary swirl area, attached to the S-duct wall, tends to get closer and to link with the primary zone in the middle.

For S-ducts intakes, another important parameter is the DC60 and DC15. These

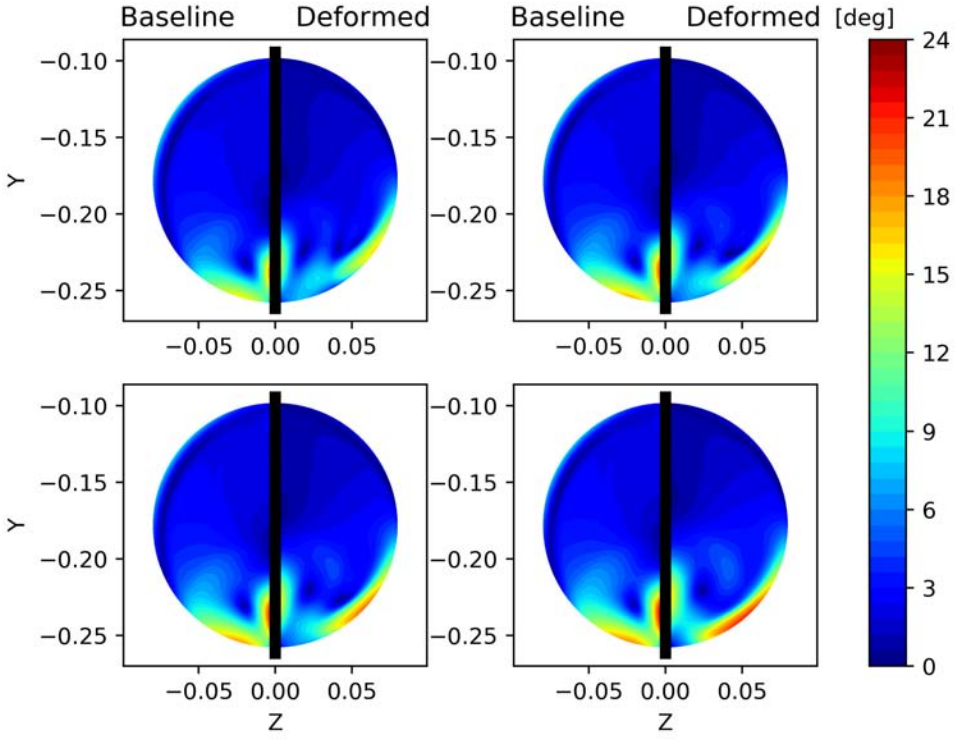


FIGURE 6.6: NIPC 1D Best CP_{mean}: Swirl contour comparison at the AIP.

values are not taken into account as objective functions inside all the optimization cycles, but it is interesting to observe how they have changed with respect to the baseline (figure 6.7 6.8). The DC60 did not improve, but with a smaller sectors distortion analysis (i.e DC15), it is possible to see how the DC15 values, of the deformed design, increased in the angle range $100° \leq \theta \leq 140°$. This variation is due to the presence of a secondary flow detachment area that lowers the value $\overline{P}_{tot,\theta}$ and increases the numerator of equation 1.6.

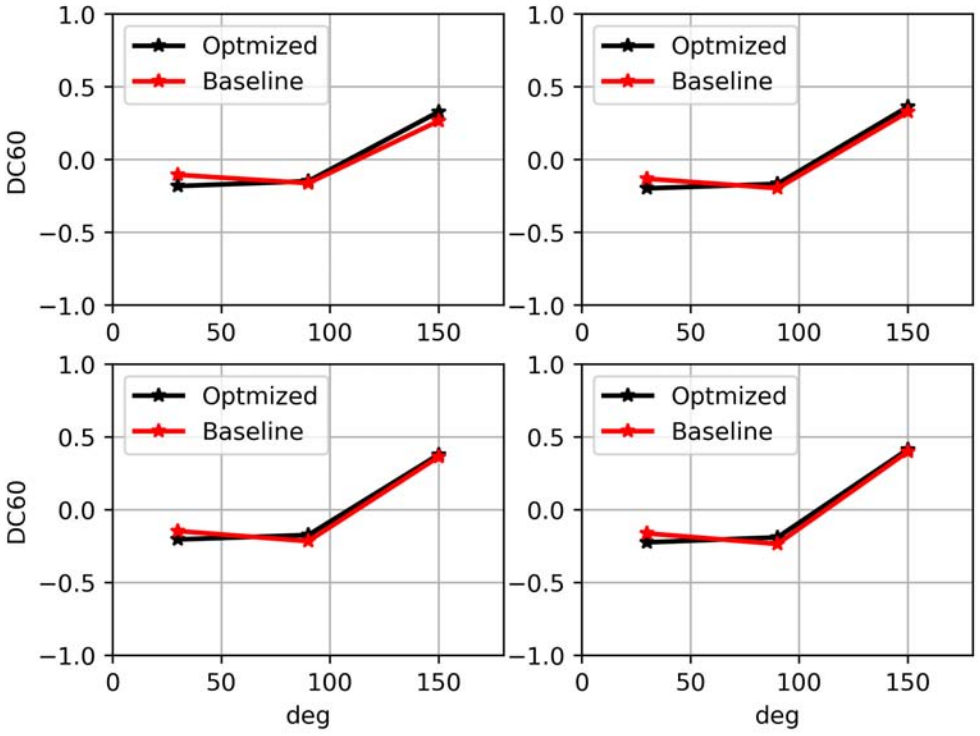


FIGURE 6.7: NIPC 1D Best CP_{mean}: DC60.

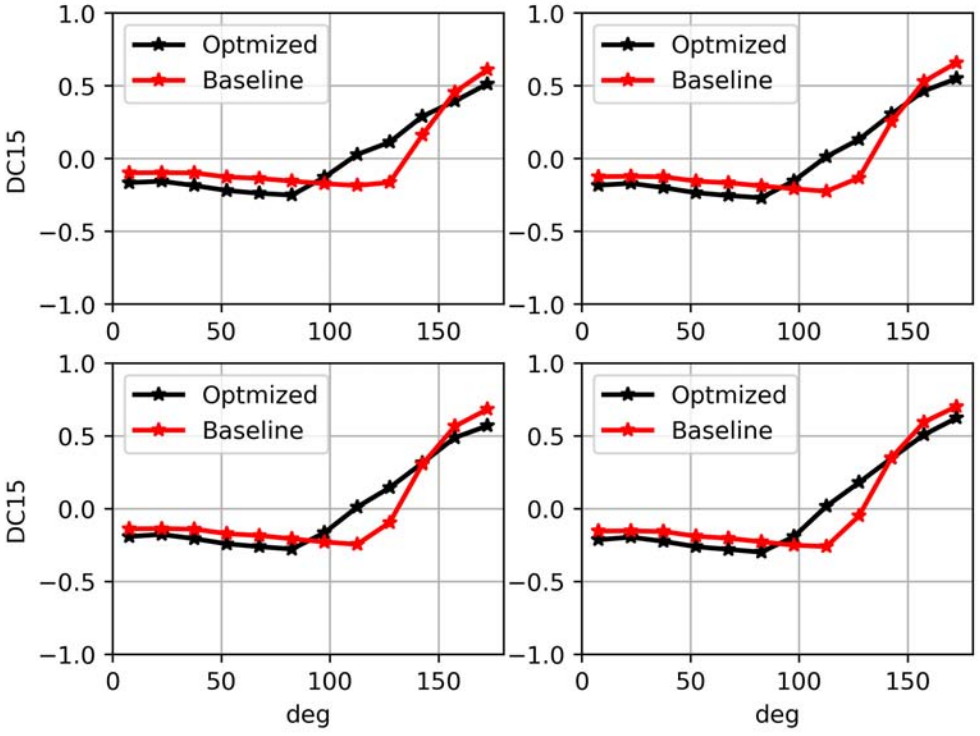


FIGURE 6.8: NIPC 1D Best CP_{mean}: DC15.

6.1.1.2 NIPC 1D: Best CP stdv

Now we look at the Best CP_{stdv} design configuration. The following table 6.5 highlights the differences between the objective functions values obtained from the optimisation loop and the post-processing analysis ones.

TABLE 6.5: NIPC 1D Best CP_{stdv}: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Optimisation	0.028601	0.006284	4.326070°	0.077612°
Post-proc.	0.028585	0.006652	4.314047°	0.160728°
Baseline	0.02905	0.007574	4.283636°	0.223726°

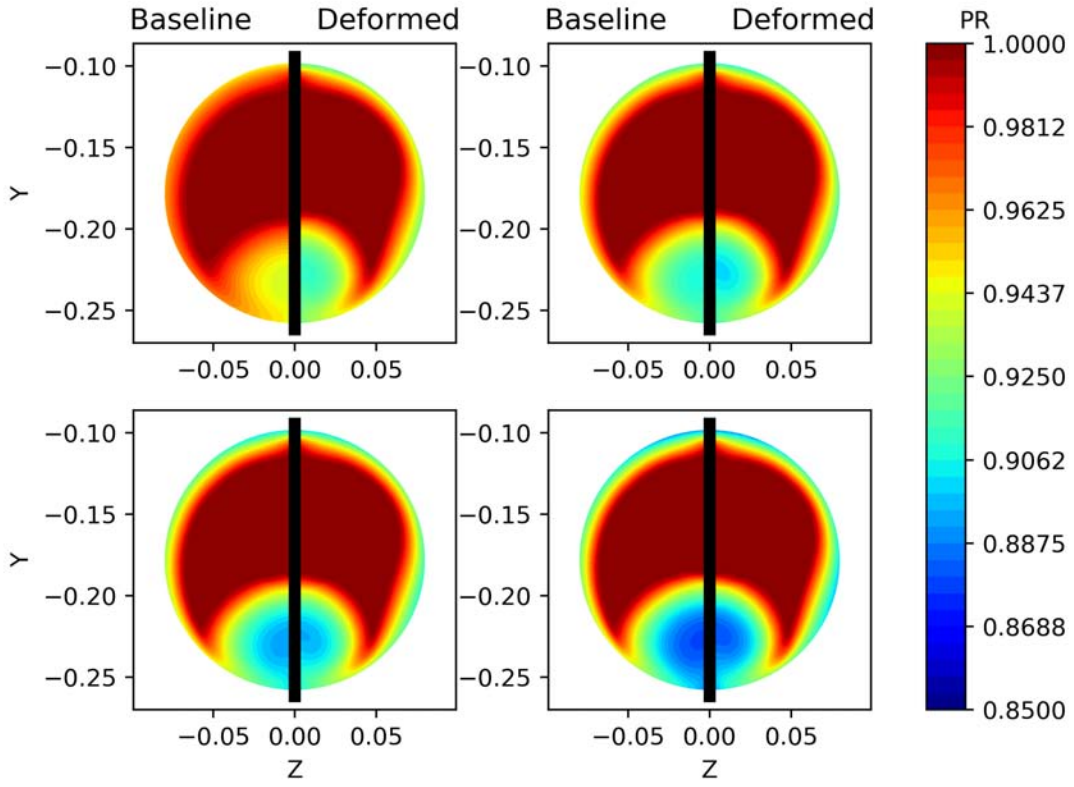


FIGURE 6.9: NIPC 1D Best CP_{stdv}: PR contour comparison at the AIP.

In figure 6.9 it is very easy to understand why this design has the Best CP_{stdv}. In fact, the deformed design changes its PR values more slowly than the baseline ones. This trend is evident when we look at the contour colors in the flow detachment area that is located at the bottom of all the half sections.

In the Swirl contour the differences between the samples are not evident and moreover, the deformed geometry has the same Swirl behaviour of the Baseline (figure 6.10).

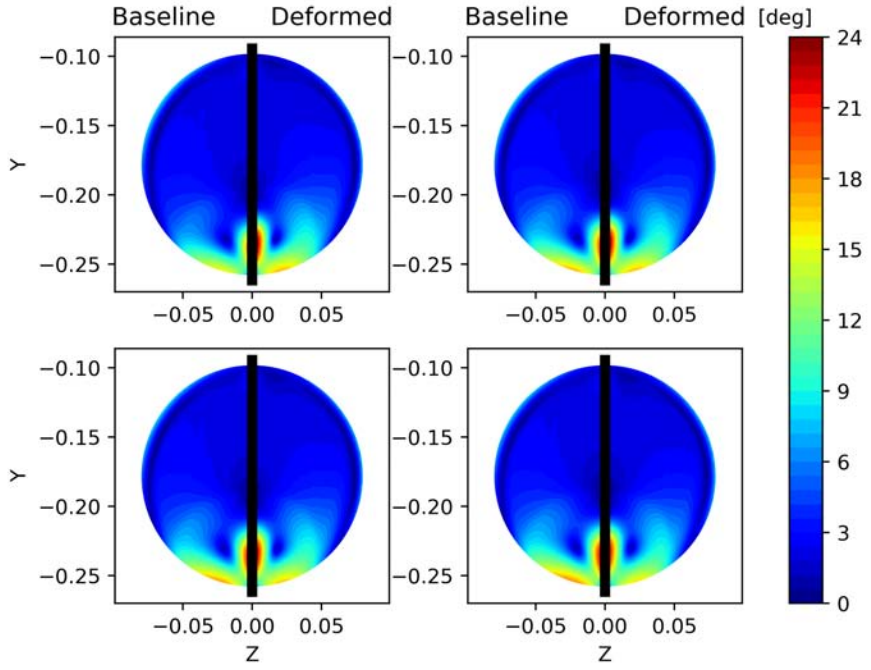


FIGURE 6.10: NIPC 1D Best CP_{stdv}: Swirl contour comparison at the AIP.

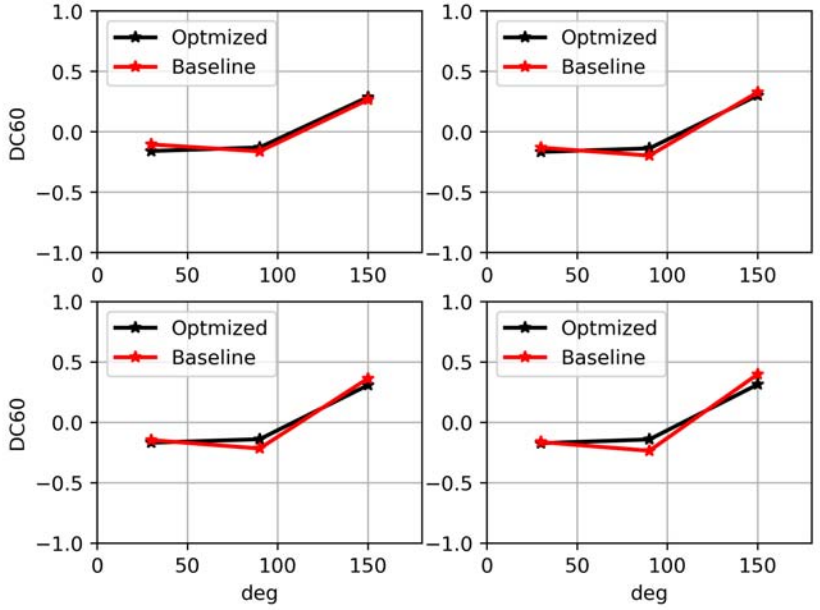


FIGURE 6.11: NIPC 1D Best CP_{stdv}: DC60.

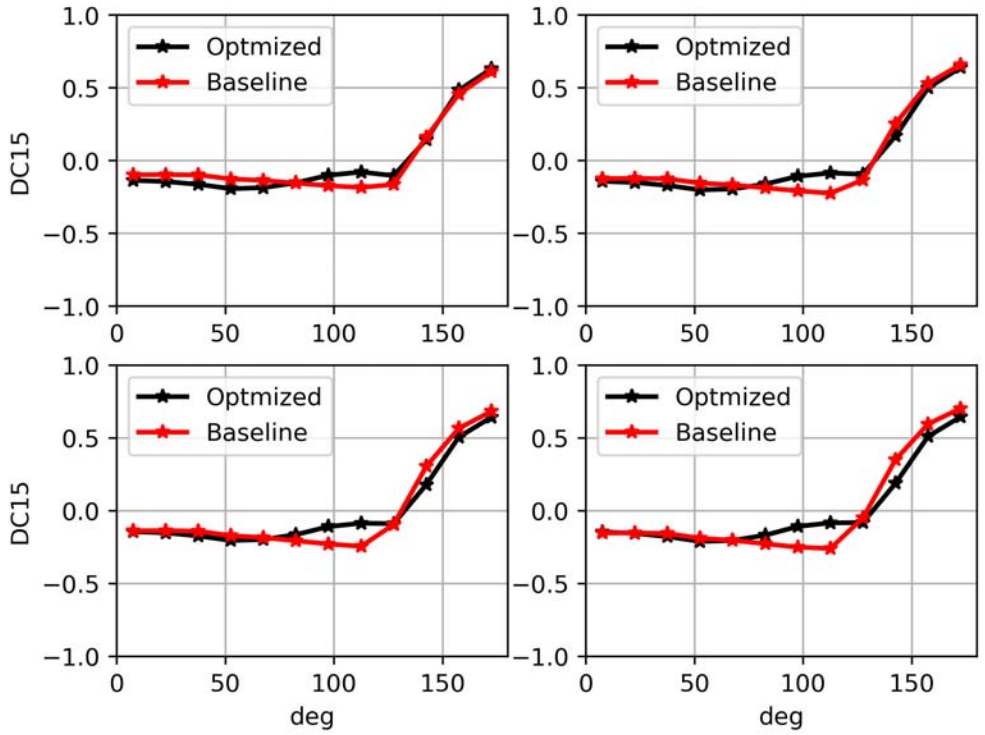


FIGURE 6.12: NIPC 1D Best CP_{stdv}: DC15.

In the DC15 plot (figure 6.12), the red (Baseline) and the black (optimized) curves are almost superimposed in the angle range $135° \leq \theta \leq 180°$. This is usually the case in which the two geometries have a very similar behaviour in the flow detachment area.

6.1.1.3 NIPC 1D: Best Swirl mean

The third configuration to analyse is the Best $Swirl_{mean}$ design. In table 6.6, the differences between the objective functions values obtained from the optimisation loop and those obtained from the post-processing analysis are displayed.

TABLE 6.6: NIPC 1D Best $Swirl_{mean}$: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Optimisation	0.029442	0.007285	2.28655°	0.140704°
Post-proc.	0.029454	0.007275	2.289931°	0.138313°
Baseline	0.02905	0.007574	4.283636°	0.223726°

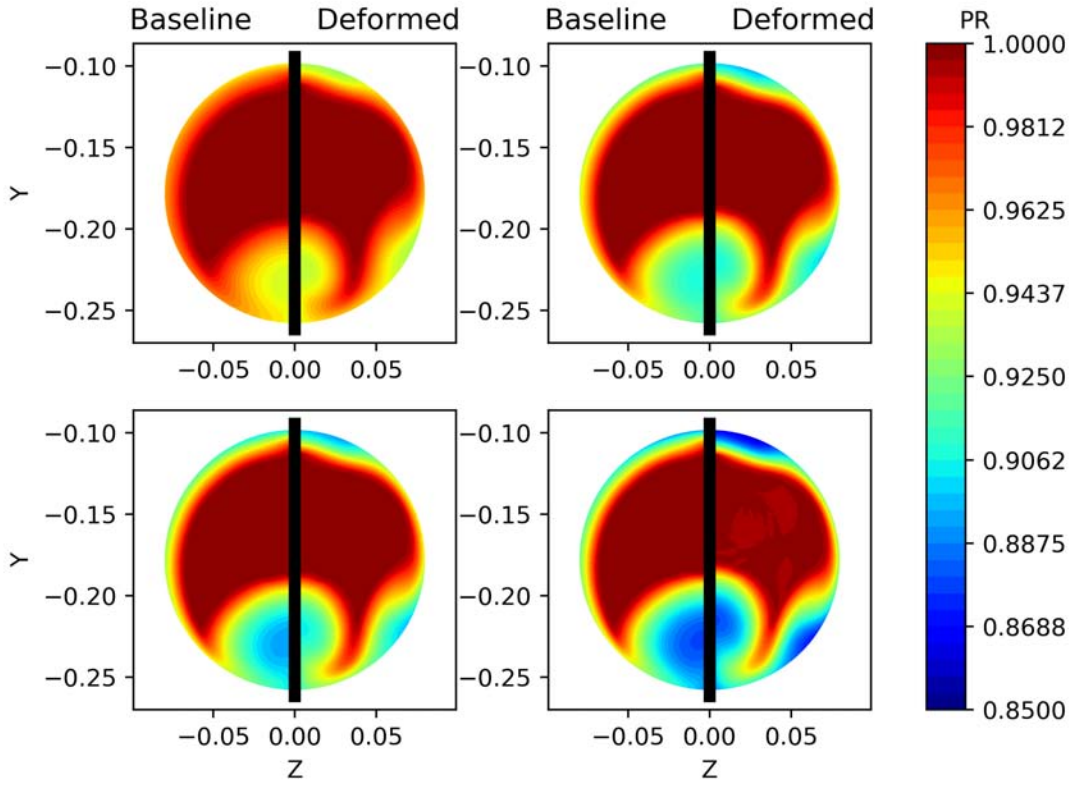
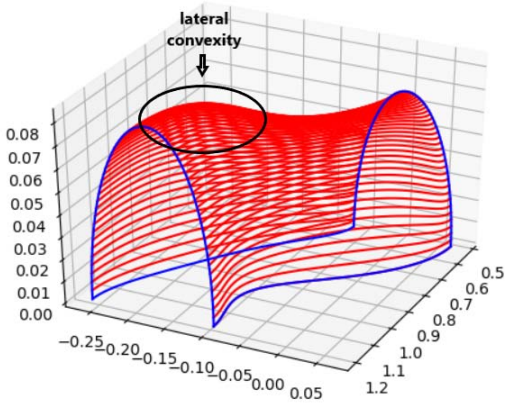


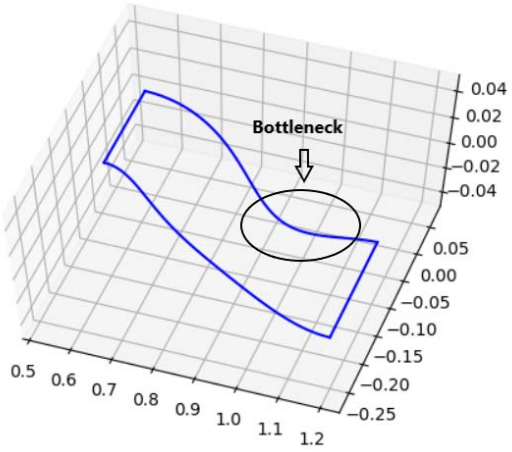
FIGURE 6.13: NIPC 1D Best $Swirl_{mean}$: PR contour comparison at the AIP.

The design configuration taken into account shows in figure 6.13 the formation of a secondary (side wall) and a tertiary (up wall) flow detachment areas.
In this case, the secondary area is more evident compared to the Best CP_{mean} design before being analysed. In fact, if in the CP_{mean} design, the new secondary area seemed like an arm of the primary one, now the two areas are almost completely distinct. Also here, the birth of this new flow detachment area is due to the presence of a lateral convexity in the sections near the AIP (figure 6.14(a)). Moreover, it is possible to see that this convexity is not as big as the Best CP_{mean} design.
On the other hand, the birth of a small third flow detachment area is due to a bottleneck section near the AIP, located on the upper wall of the S-duct (figure 6.14(b)).

A big improvement was obtained in the Swirl values. It is possible to observe in figure 6.15 that in all the four contour comparison, the values of the Swirl of the deformed design are lower than the baseline values and the secondary swirl area (side wall) is disappeared. In fact, in table 6.7, all the values of Swirl that were found for each inlet velocity sample are reported.



(a) Lateral convexity



(b) Bottleneck.

FIGURE 6.14: NIPC 1D Best $Swirl_{mean}$: Lateral convexity and bottleneck.

The DC15 plot of the Best $Swirl_{mean}$ design has better results than the Baseline ones (figure 6.16) for $0° \leq \theta \leq 30°$ and for $105° \leq \theta \leq 135°$, where there is the second flow detachment area. Moreover, the values of DC15 improved a little for $135° \leq \theta \leq 180°$. This improvement is due to a narrower primary detachment zone. Instead, the DC60 had no improvements.

TABLE 6.7: NIPC 1D Best $Swirl_{mean}$: Swirl value for each velocity sample.

	***Swirl* Baseline**	**NIPC 1D Best $Swirl_{mean}$**	**Improvement**
Sample 1	4.09554°	2.17297°	−46.94%
Sample 2	4.13263°	2.23661°	−45.88%
Sample 3	4.32343°	2.24825°	−48.00%
Sample 4	4.63012°	2.46851°	−46.69%

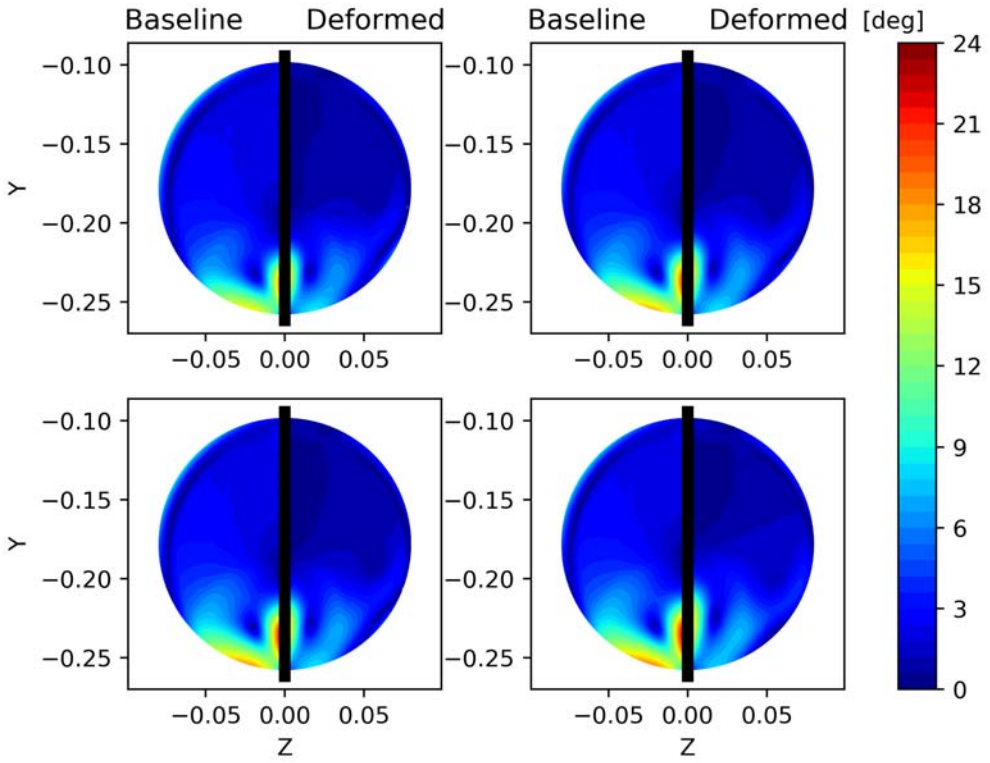


FIGURE 6.15: NIPC 1D Best $Swirl_{mean}$: Swirl contour comparison at the AIP.

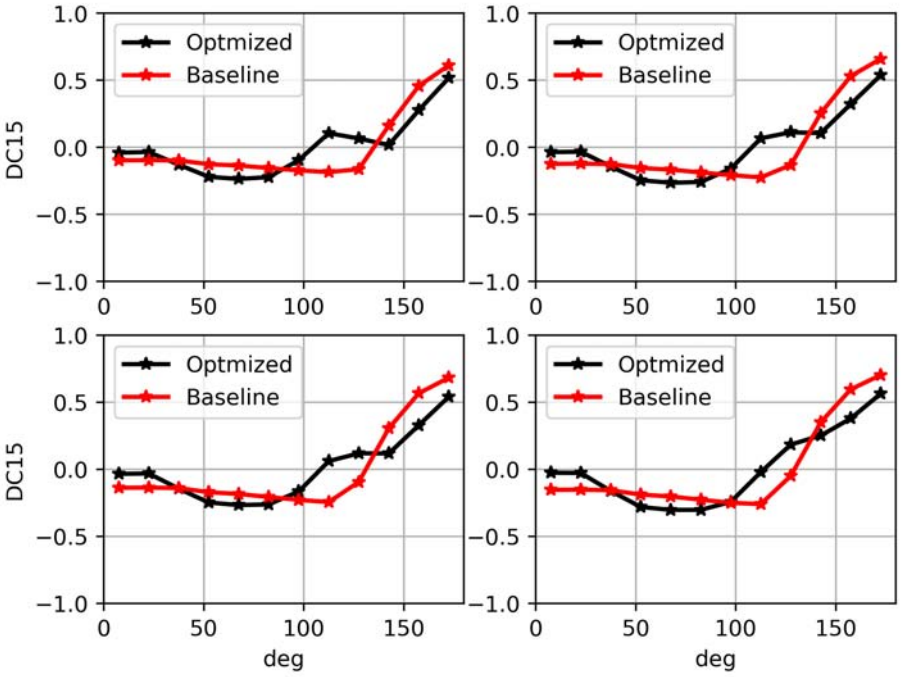


FIGURE 6.16: NIPC 1D Best $Swirl_{mean}$: DC15.

6.1.1.4 NIPC 1D: Best Swirl stdv

The fourth configuration to analyse is the Best $Swirl_{stdv}$ design. In table 6.8, the differences between the objective functions values obtained from the optimisation loop and those obtained from the post-processing analysis are displayed.

TABLE 6.8: NIPC 1D Best $Swirl_{stdv}$: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Optimisation	0.026427	0.007081	2.963150°	0.003968°
Post-proc.	0.026412	0.007121	2.962503°	0.005442°
Baseline	0.02905	0.007574	4.283636°	0.223726°

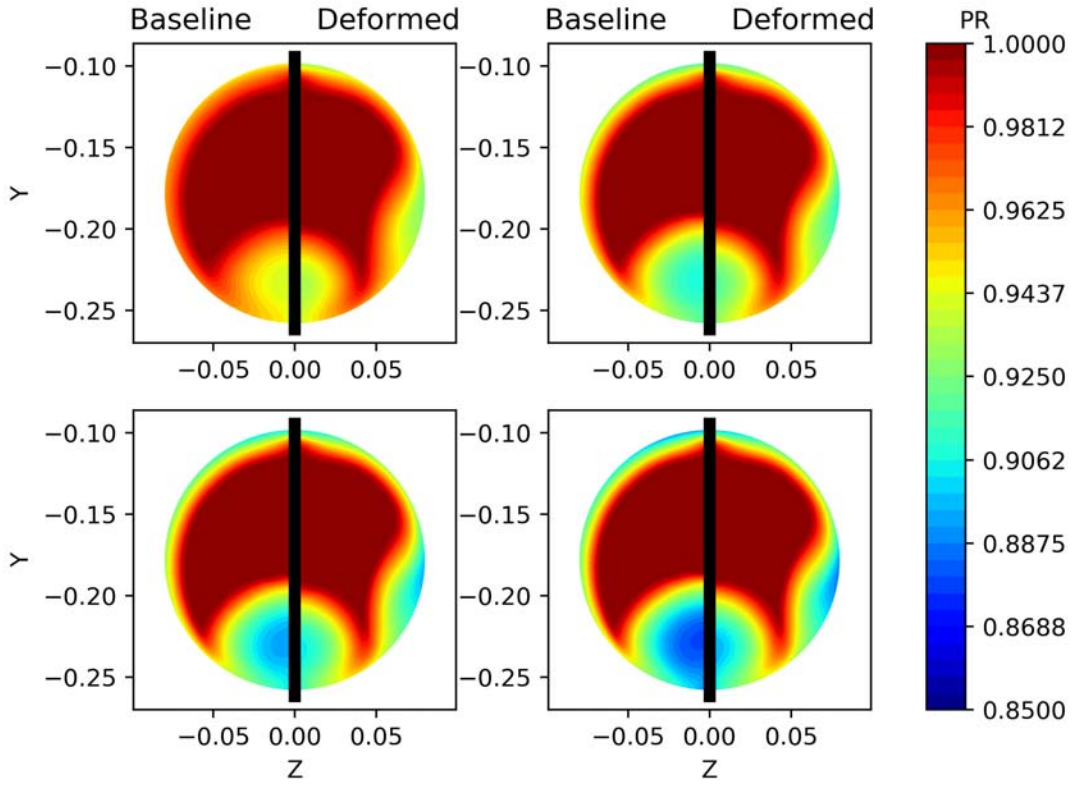


FIGURE 6.17: NIPC 1D Best $Swirl_{stdv}$: PR contour comparison at the AIP.

The Best $Swirl_{stdv}$ design has good results for all the four objective functions (table 6.8), in particular, the $Swirl_{mean}$, and the $Swirl_{stdv}$ are very low and it is possible to see the effects into the Swirl Contour plot (figure 6.18).
Moreover, the effects of these good results are also visible in the DC60 plot. The sector $60° \div 120°$ has a distortion ≈ 0 in all the four sub-plots, and also the sector $120° \div 180°$ has improved (figure 6.19).

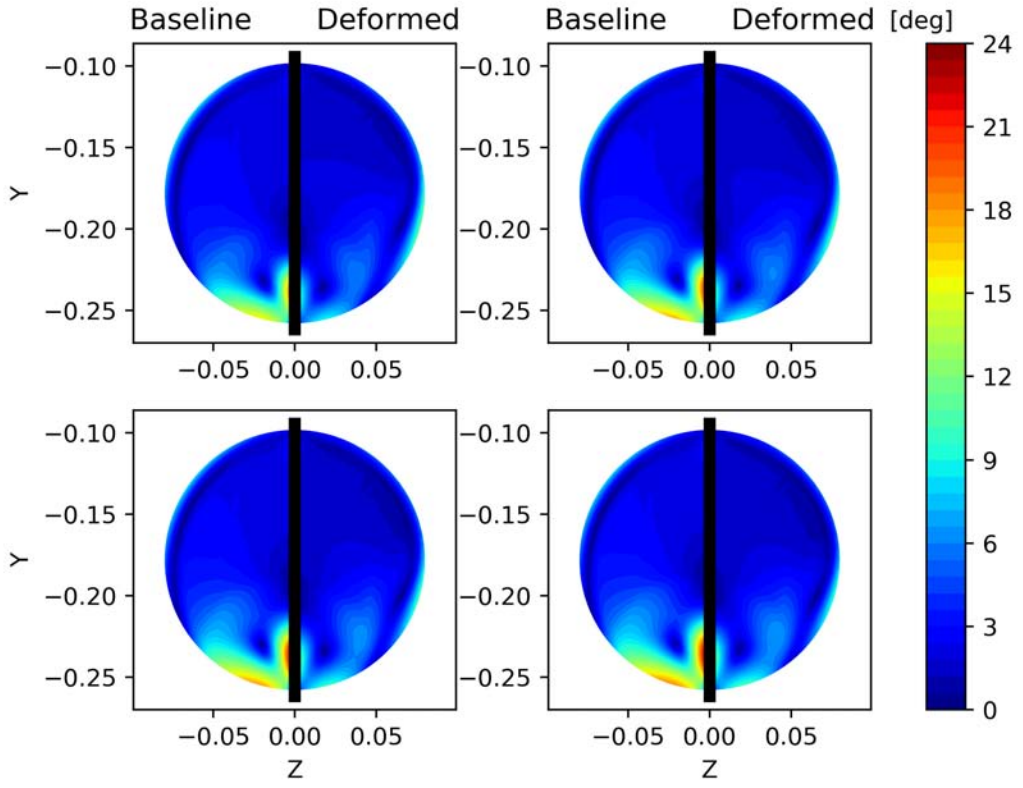


FIGURE 6.18: NIPC 1D Best $Swirl_{stdv}$: Swirl contour comparison at the AIP.

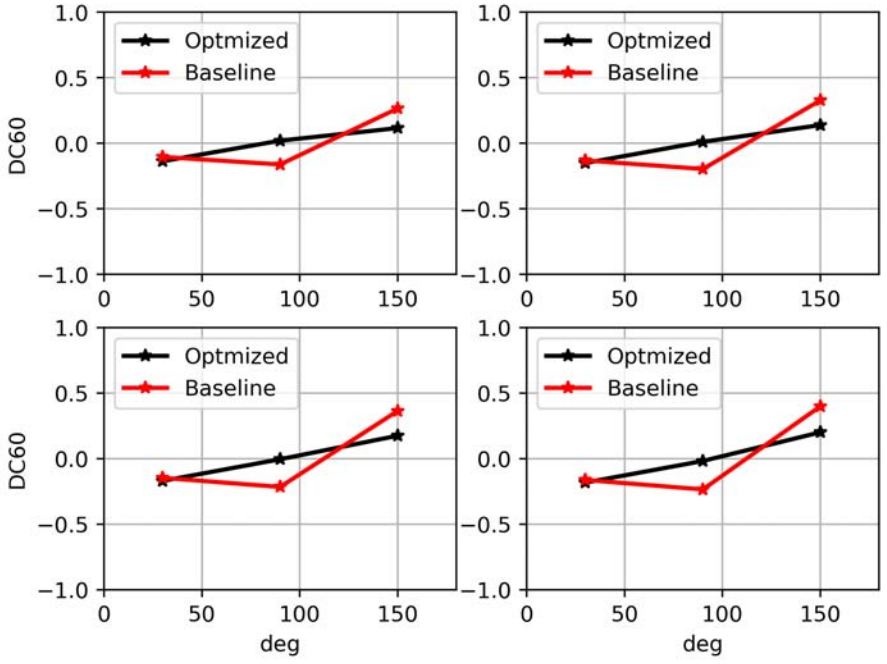


FIGURE 6.19: NIPC 1D Best $Swirl_{stdv}$: DC60.

The only thing to highlight in the DC15 plot 6.20 is the improvement of the values for $145° \leq \theta \leq 180°$. In fact, it is quite easy to see in figure 6.17 that the primary detachment zone of the deformed geometry is smaller than the baseline one.

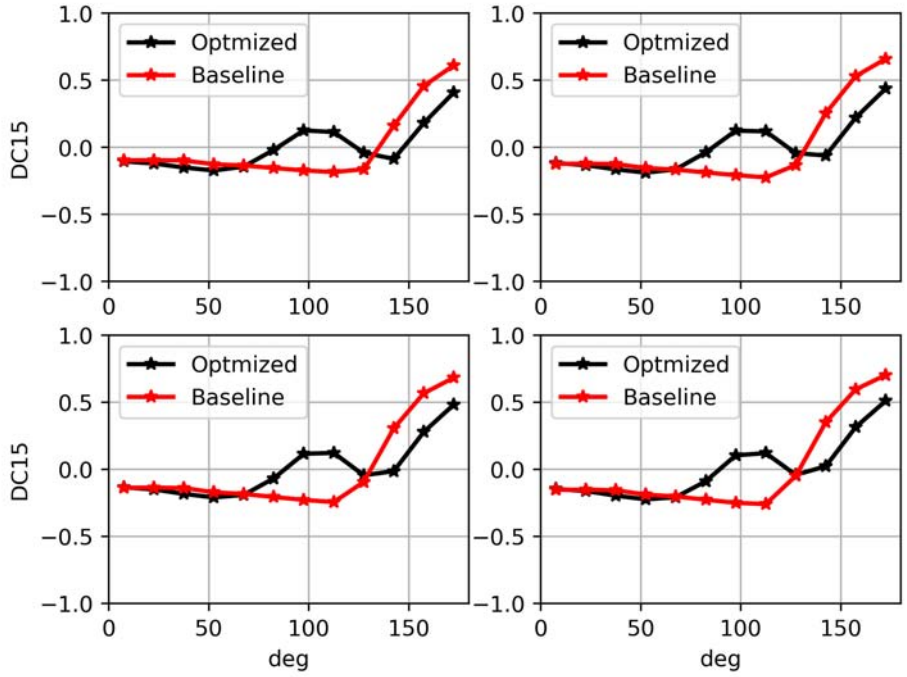


FIGURE 6.20: NIPC 1D Best $Swirl_{mean}$: DC15.

6.1.1.5 NIPC 1D: Parallel coordinate optima design

As said before, only the use of the parallel coordinates technique can help the designer to find the designs that try to minimise all the four objective functions simultaneously. In this section, the first optima design of table 6.2 was analysed.

TABLE 6.9: NIPC 1D optima 1: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Optimisation	0.026596	0.006719	2.39949°	0.078154°
Post-proc.	0.026610	0.006714	2.399510°	0.079944°
Baseline	0.02905	0.007574	4.283636°	0.223726°

The cause of the presence of a second big flow detachment zone (figure 6.22) is well highlighted in figure 6.23.

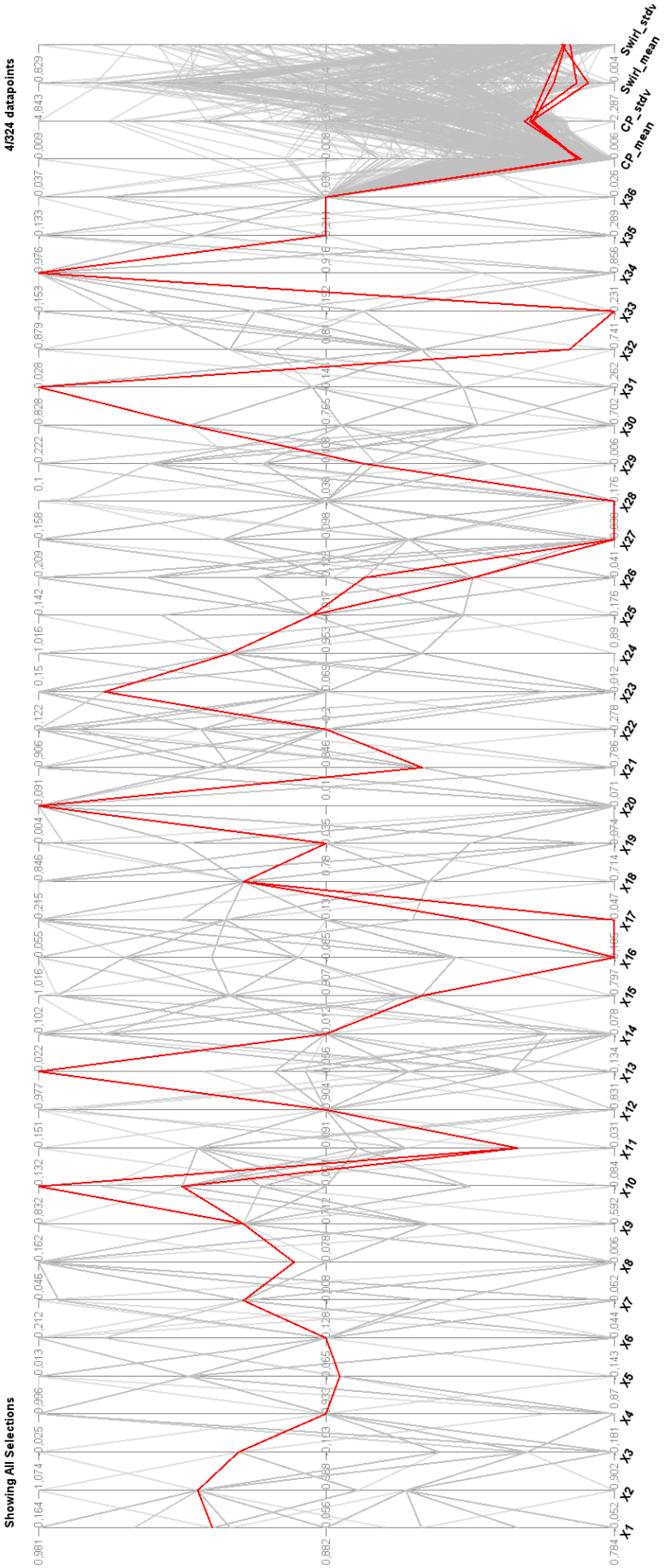


FIGURE 6.21: NIPC 1D: Parallel Coordinate.

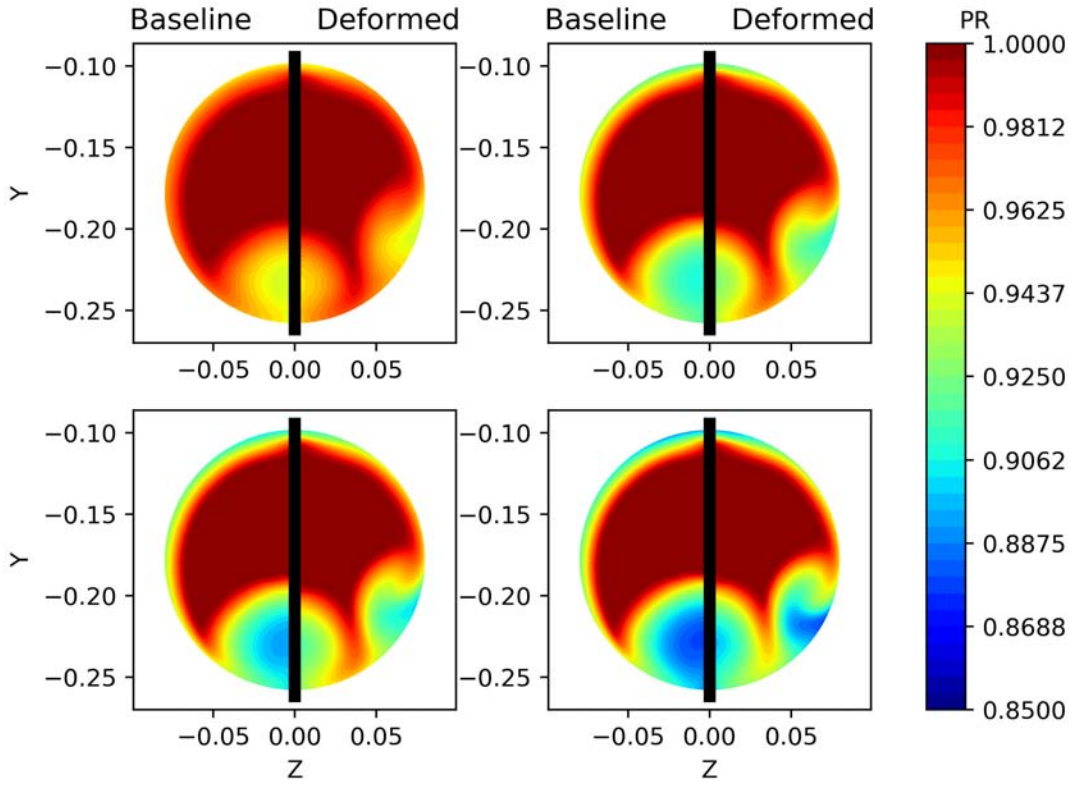


FIGURE 6.22: NIPC 1D Best optima 1: PR contour comparison at the AIP.

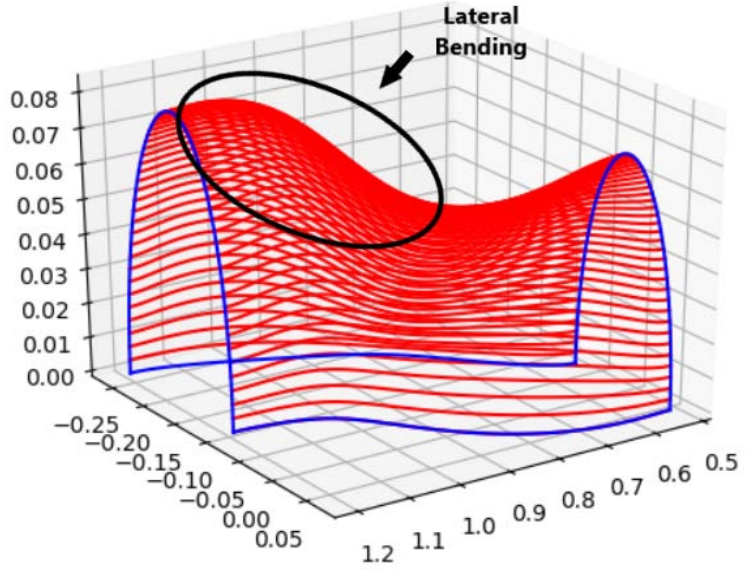


FIGURE 6.23: NIPC 1D optima 1: Lateral bending.

The Swirl contour plot in figure 6.24 has excellent results. The deformed design has no high swirl angle values and also there are small differences between the four deformed sub-plots.

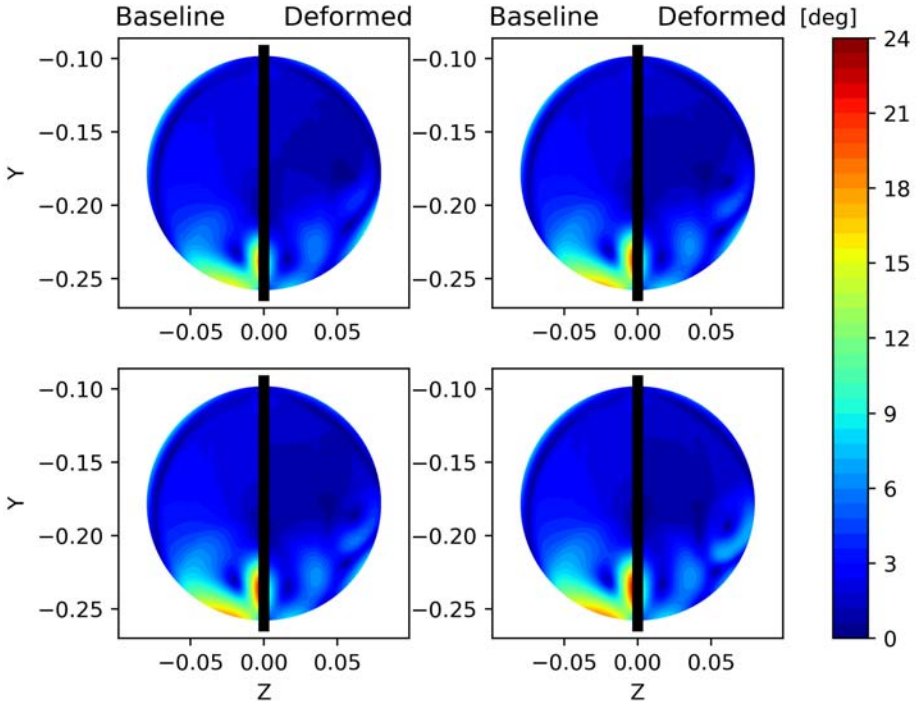


FIGURE 6.24: NIPC 1D optima 1: Swirl contour comparison at the AIP.

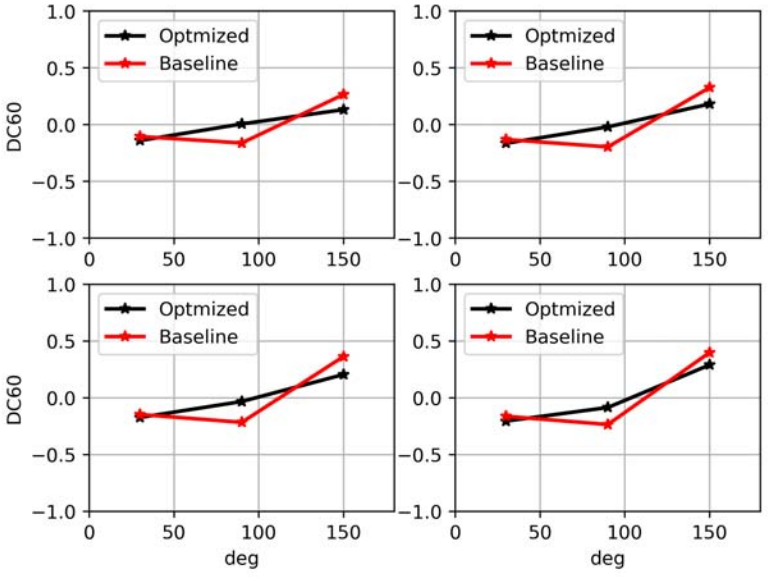


FIGURE 6.25: NIPC 1D optima 1: DC60.

The DC60 has good values. In figure 6.25 the sector $60° \div 120°$ has a distortion value ≈ 0 in all the four sub-plots, and also the sector $120° \div 180°$ has improved. The behaviour of the DC60 is similar to the Best $Swirl_{stdv}$ design that was analysed before. The DC15 sub-plots (figure 6.26) show as the small size of the primary

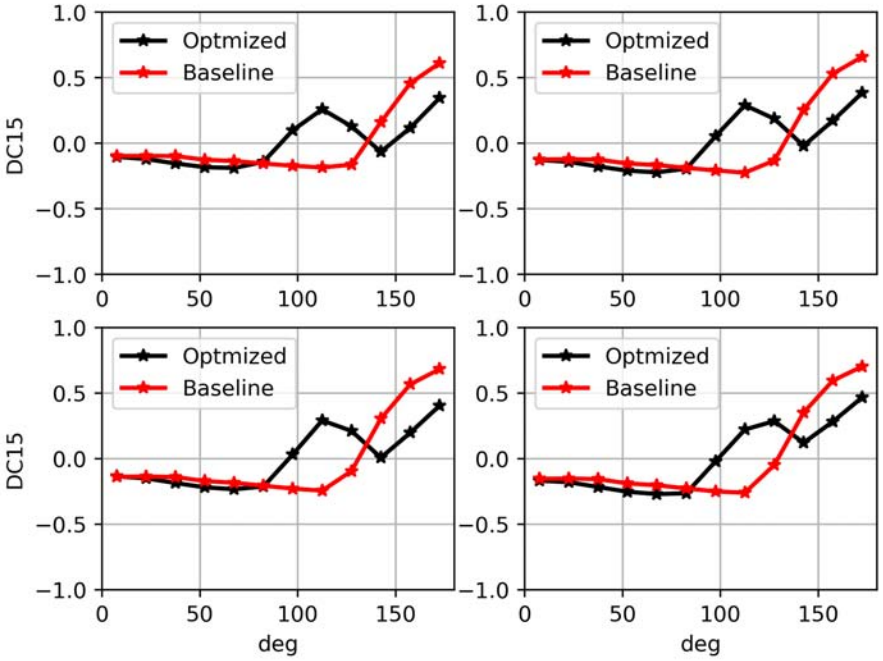


FIGURE 6.26: NIPC 1D optima1: DC15.

flow detachment area allows the improvement of the DC15 values (i.e $145° \div 180°$). On the contrary, it is not possible to say the same thing of the secondary flow detachment area because it improves some sectors but makes others worse inside the range $90° \div 140°$.

6.1.2 NISP 1D

To implement the NISP 1D cycle, the uncertainty input was sampled with the Hermite-Gauss Sampling technique (HG). The MOTS software was stopped after having successfully evaluated 317 configurations. As explained at the end of Chapter 5, the NIPC uncertainty quantification technique needs $(p+1)^n$ deterministic evaluations to find the $P+1$ PC coefficients and the stochastic outputs (i.e objective functions). In this case the number of evaluations were equal to four. This means that $317 \times 4 = 1268$ CFD Fluent simulations converged. The time

for a single fluent evaluation is ≈ 37.5 minutes, so the time that was necessary to obtain all these results was ≈ 801 hours. However, the amount of time above estimated does not take into account all the configurations that were analysed but subsequently discarded if they did not respect the Swirl mean and Swirl standard deviation constraints or if a structured geometry mesh was not possible to be created by ANSYS ICEM. The amount of discarded design is 63.
However, also here, the amount of time above estimated does not take into account all the configurations that were analysed but subsequently discarded for the same reasons already listed for the NIPC 1D. The post processing analysis was carried out following the same procedure adopted for the NIPC 1D. To help the reader to understand the problem of a four-dimensional Pareto front, three faces (i.e bi-dimensional Pareto front) of the imaginary cube were plotted (figure 6.27 6.28 6.29). From these three plots, four Best designs were chosen and analysed. At the end, other eight optima designs were chosen and analysed with the support of the parallel coordinates in figure 6.40, but only one was highlighted in the thesis. In tables 6.10 6.11, all the chosen designs were listed.

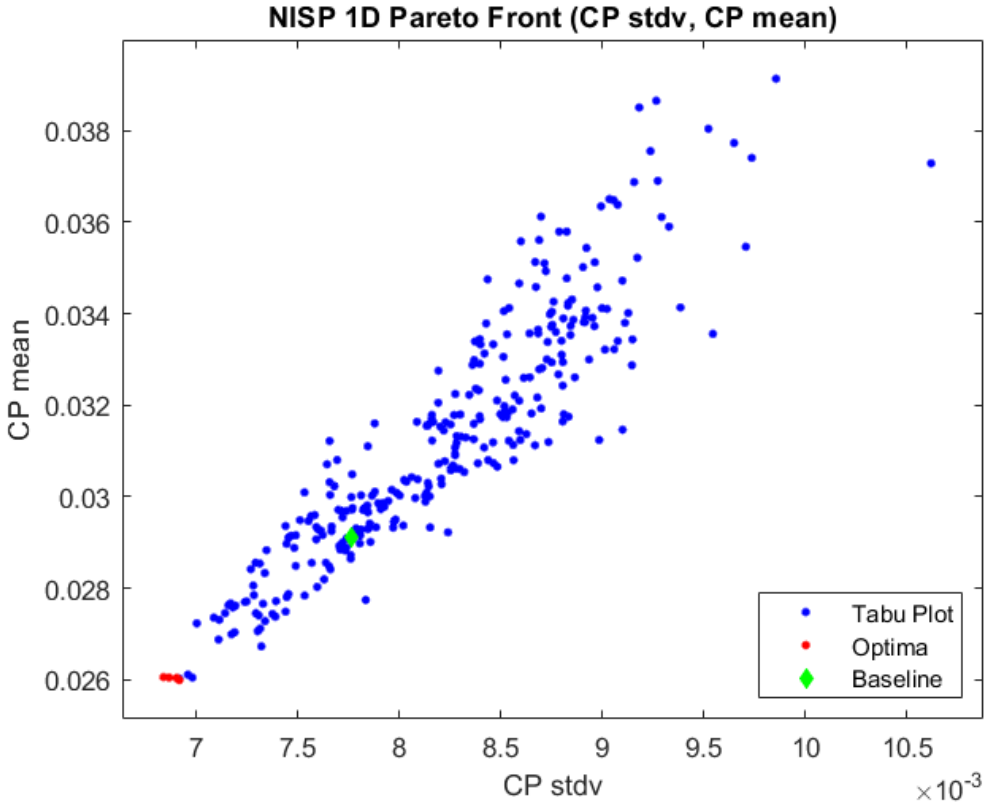


FIGURE 6.27: NISP 1D: CP_{stdv}, CP_{mean}.

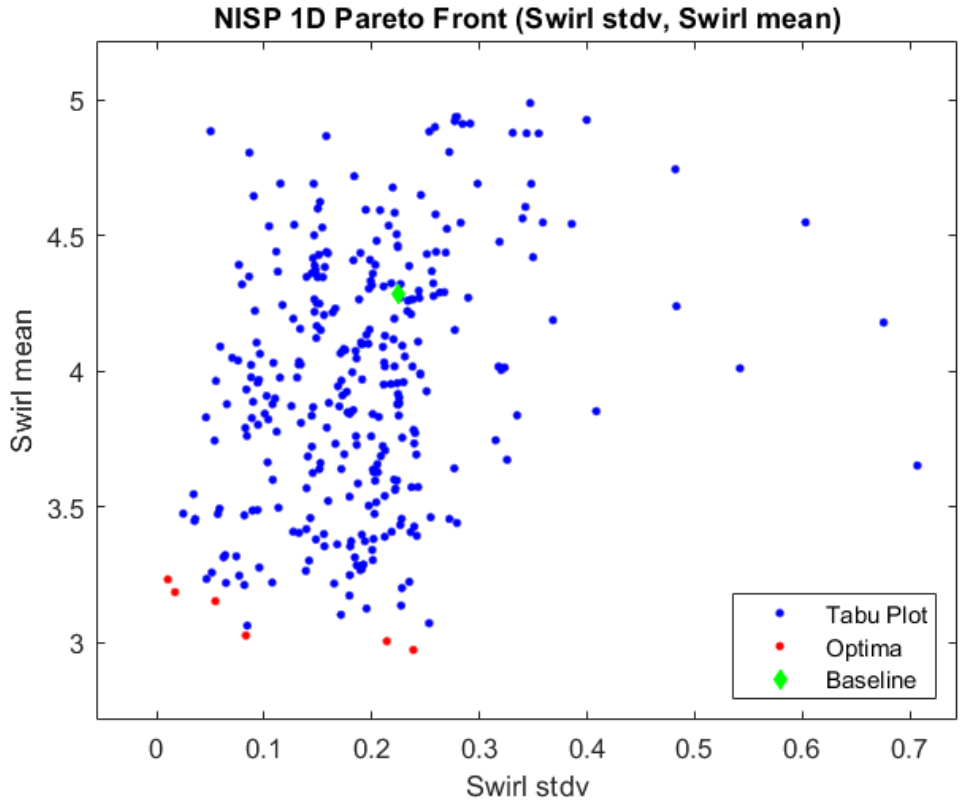


FIGURE 6.28: NISP 1D: $Swirl_{stdv}$, $Swirl_{mean}$.

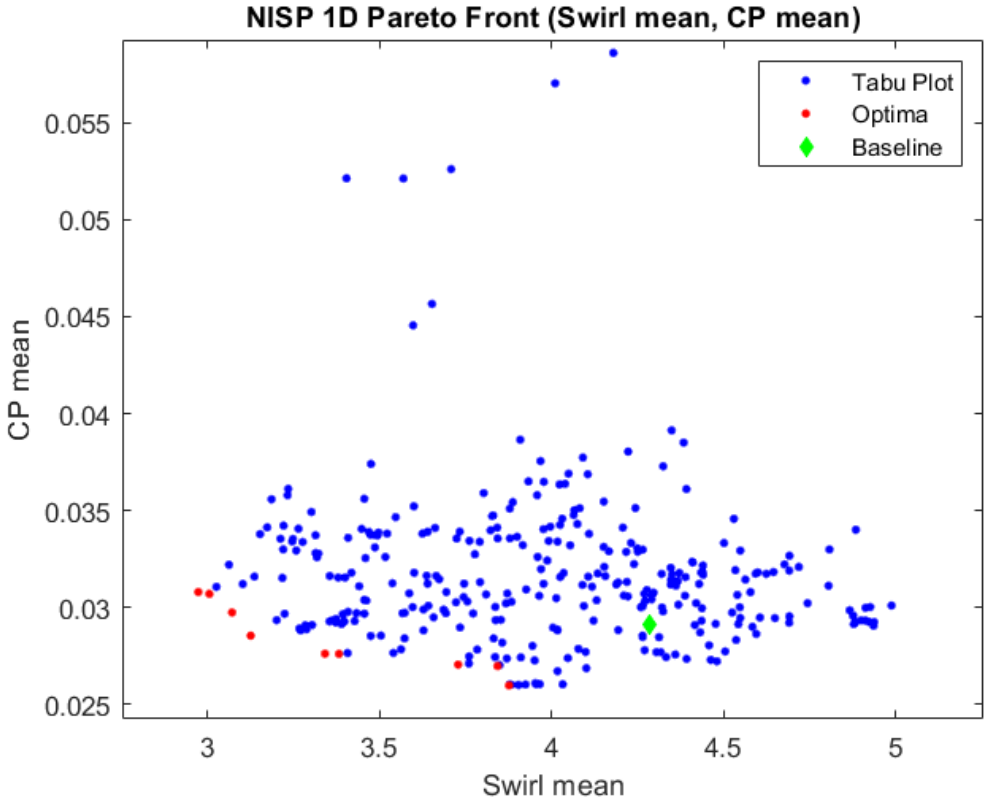


FIGURE 6.29: NISP 1D: $Swirl_{mean}$, CP_{mean}.

Table 6.10: NISP 1D Best design results.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Best CP_{mean}	0.025992	0.006918	3.87643°	0.224438°
Best CP_{stdv}	0.026057	0.006840	3.965730°	0.171665°
Best $Swirl_{mean}$	0.030804	0.007695	2.974180°	0.238561°
Best $Swirl_{stdv}$	0.035583	0.008600	3.186860°	0.017106°
Baseline	0.029124	0.007765	4.283883°	0.224200°

Table 6.11: NISP 1D optima designs (*trade-off designs*) results obtained with the use of the parallel coordinates.

Design	CP_{mean}	CP_{stdv}	$Swirl_{stdv}$	$Swirl_{stdv}$
optima 1	0.027629	0.00716	3.34194°	0.200365°
optima 2	0.02909	0.007742	3.2847°	0.185958°
optima 3	0.028994	0.007722	3.28496°	0.190507°
optima 4	0.028924	0.007706	3.26762°	0.189412°
optima 5	0.029106	0.007754	3.3046°	0.200914°
optima 6	0.028883	0.007741	3.28847°	0.192288°
optima 7	0.02883	0.007734	3.27152°	0.190673°
optima 8	0.028556	0.00764	3.1264°	0.195091°
Baseline	0.029124	0.007765	4.283883°	0.224200°

6.1.2.1 NISP 1D: Best CP mean

The first configuration taken into account in this paragraph is the design that has achieved the best overall result in terms of CP_{mean} for the NISP 1D. Table 6.12 highlights the differences in the objective functions values between the Best CP_{mean} design and the Baseline.

Table 6.12: NISP 1D Best CP_{mean}: differences in the objective functions values between the Best CP_{mean} design and the Baseline.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Best CP_{mean}	0.025992	0.006918	3.87643°	0.224438°
Baseline	0.029124	0.007765	4.283883°	0.224200°

In table 6.12 and in figure 6.30 it is possible to see how the CP_{mean} value drastically improved. In fact, in table 6.13, are reported all the values of $CP = 1 - PR$ that were found for each inlet velocity sample. On the other hand, the flow detachment area became higher and narrower than the Baseline one.

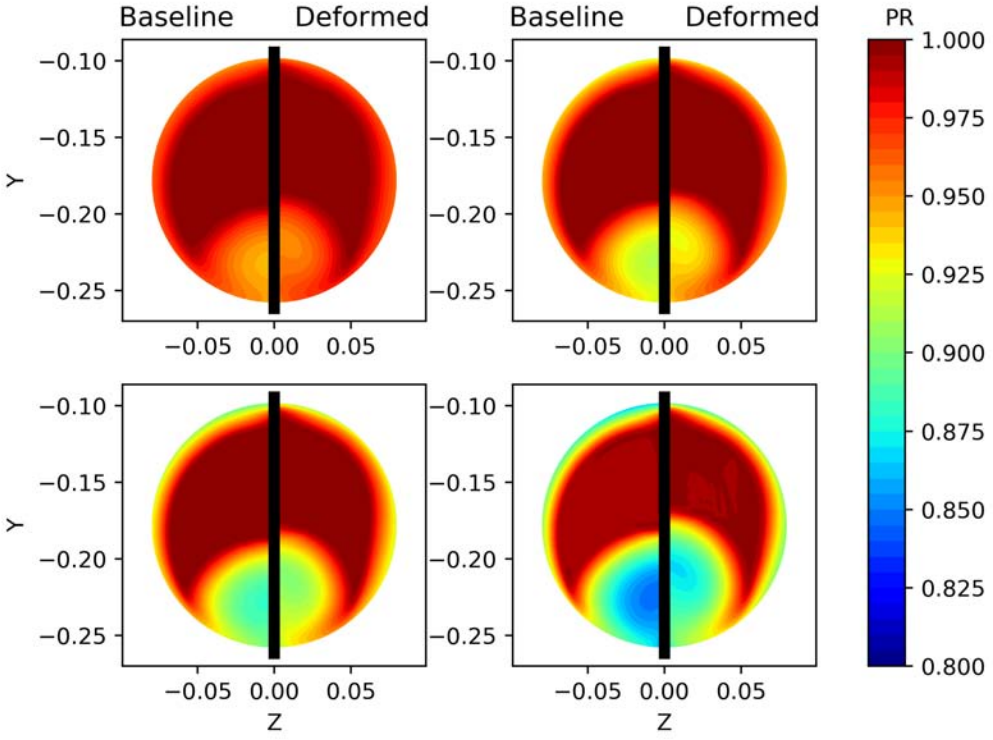


FIGURE 6.30: NISP 1D Best CP_{mean}: PR contour comparison at the AIP.

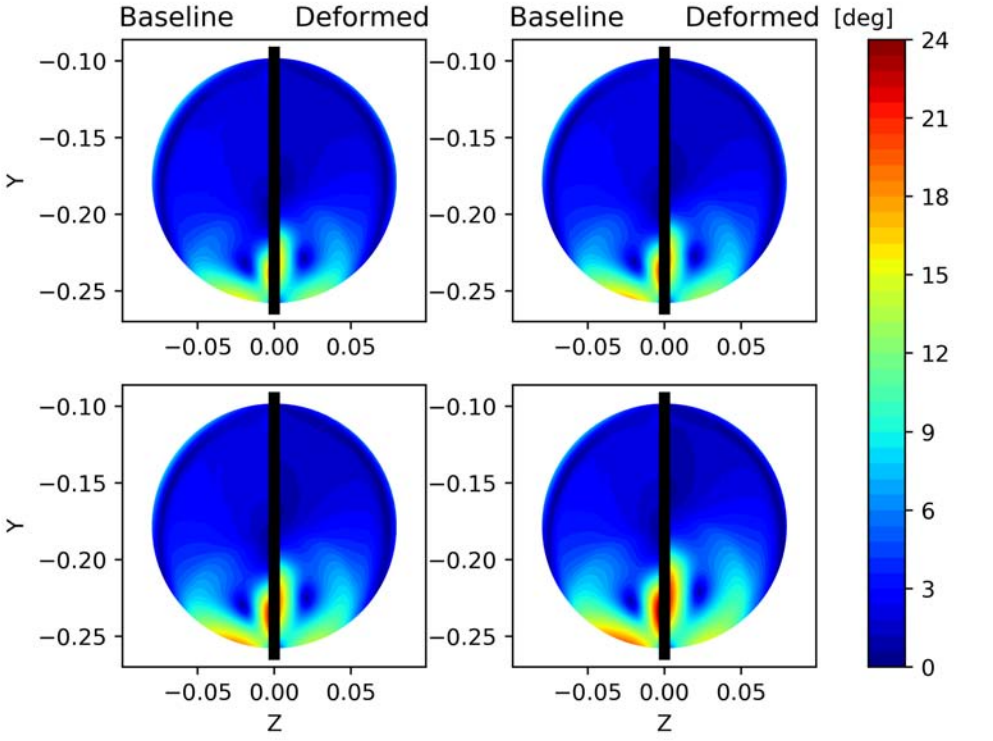


FIGURE 6.31: NISP 1D Best CP_{mean}: Swirl contour comparison at the AIP.

TABLE 6.13: NISP 1D Best CP_{mean}: CP value for each velocity sample.

	CP **Baseline**	**NISP 1D Best** CP_{mean}	**Improvement**
Sample 1	0.01411	0.01288	-8.72%
Sample 2	0.02329	0.02084	-10.52%
Sample 3	0.03423	0.03041	-11.16%
Sample 4	0.05134	0.04637	-9.68%

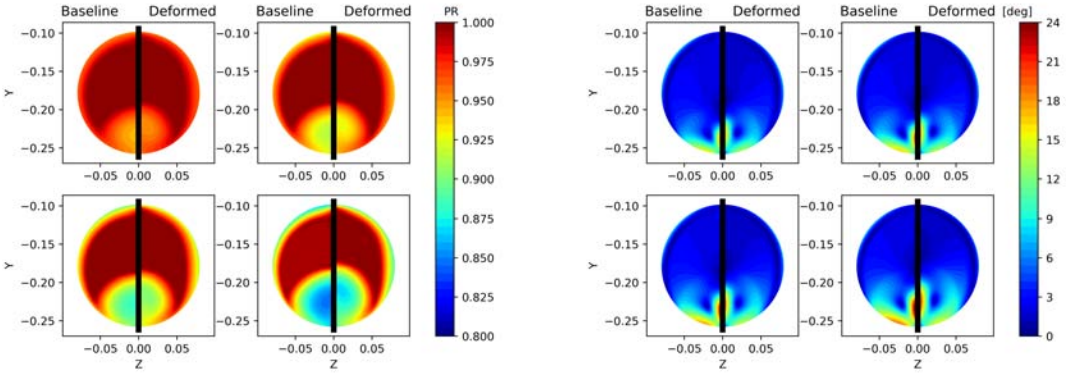
There were no important improvements for parameters DC15 and DC60. The same was for the swirl in figure 6.31. The only thing to note is that the central swirl zone of the deformed geometry is longer than the baseline one and the lower swirl area has a less swirl magnitude.

6.1.2.2 NISP 1D: Best CP stdv

Now we consider the Best CP_{stdv} design configuration. Table 6.14 highlights the differences in the objective functions values between de Best CP_{stdv} design and the Baseline.

TABLE 6.14: NISP 1D Best CP_{stdv}: differences in the objective functions values between the Best CP_{stdv} design and the Baseline.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Best CP_{stdv}	0.026057	0.006840	3.965730°	0.171665°
Baseline	0.029124	0.007765	4.283883°	0.224200°



(a) PR contour

(b) Swirl contour.

FIGURE 6.32: NISP 1D Best CP_{stdv}: Contour sub-plots.

The Best CP_{stdv} design and the Best CP_{mean} design have very similar Pressure Recovery and Swirl contour sub-plots (figure 6.32 6.30 6.31). On the other hand, the value of the DC60 has improved in the sector $60° \div 120°$ (figure 6.33) and the same was for the DC15 sectors in the range $60° \leq \theta \leq 120°$ in figure 6.34.

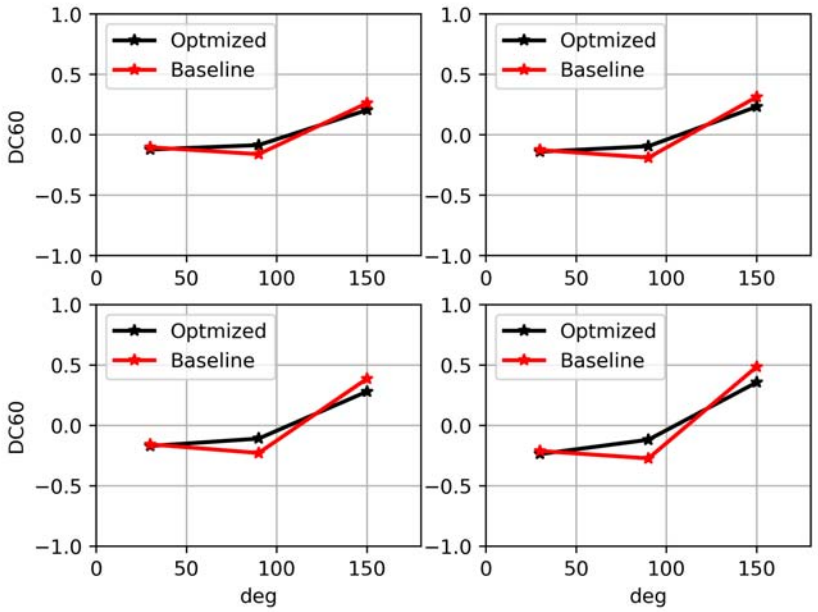


FIGURE 6.33: NISP 1D Best CP_{stdv} : DC60.

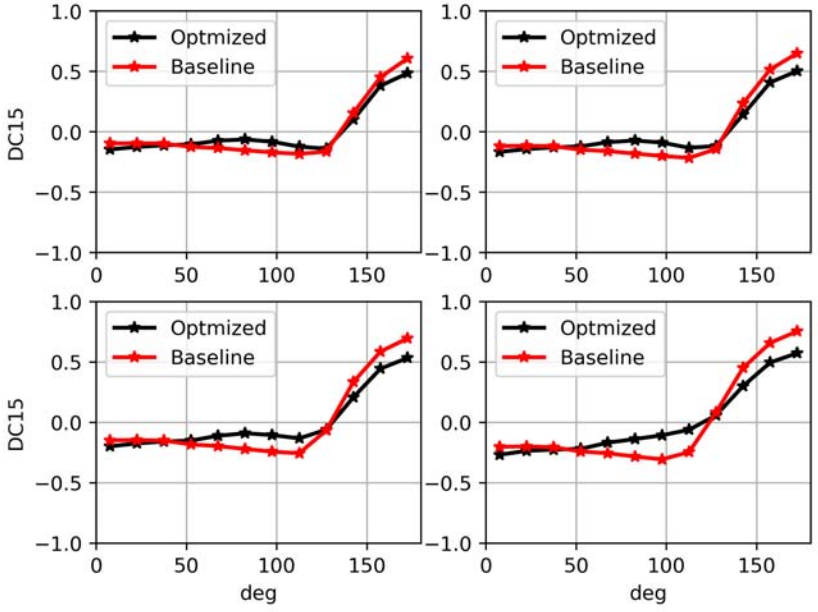


FIGURE 6.34: NISP 1D Best CP_{stdv}: DC15.

6.1.2.3 NISP 1D: Best Swirl mean

The third configuration to analyse is the Best $Swirl_{mean}$ design. In table 6.15, the differences in the objective functions values between de Best $Swirl_{mean}$ design and the Baseline are displayed.

TABLE 6.15: NISP 1D Best $Swirl_{mean}$: differences in the objective functions values between the Best $Swirl_{mean}$ design and the Baseline.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Best $Swirl_{mean}$	0.030804	0.007695	2.974180°	0.238561°
Baseline	0.029124	0.007765	4.283883°	0.224200°

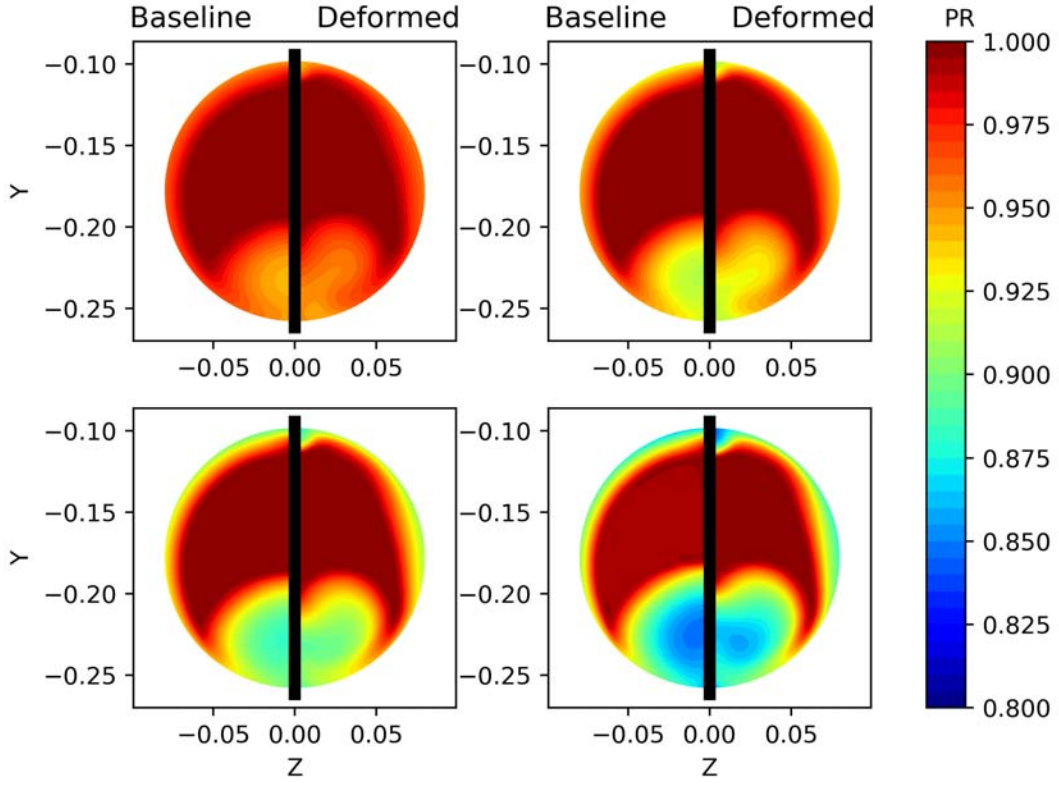


FIGURE 6.35: NISP 1D Best $Swirl_{mean}$: PR contour comparison at the AIP.

In this design configuration, the value of the CP_{mean} is worse than the baseline one, but it is interesting to analyse the shape of the lowest pressure recovery area in figure 6.35. The origin of this shape is probably attributable to the particular geometry of the duct inlet, which is clearly visible in figure 6.36. Figure 6.36 shows the axial velocity magnitude inside the entire S-duct when the velocity inlet is the

maximum one of the four samples. This means that the boundary conditions of the axial velocity plot were the same of the bottom right contour sub-plot of figure 6.35.

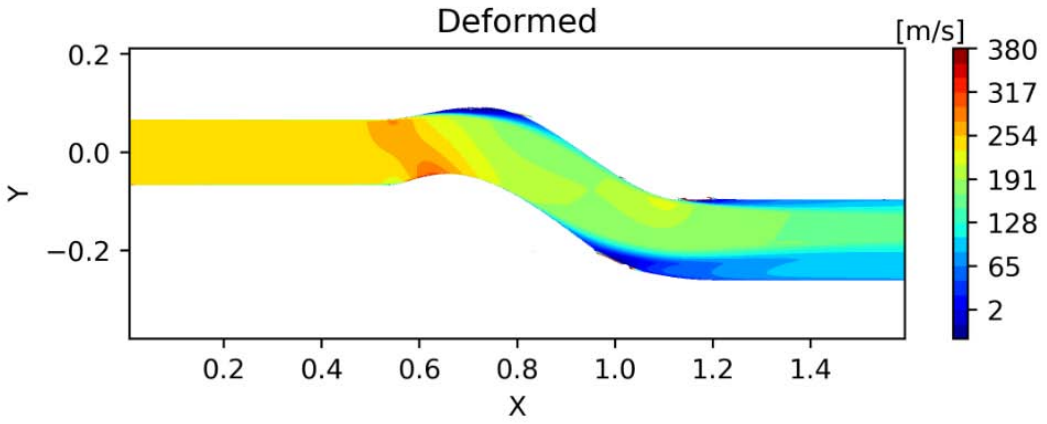


FIGURE 6.36: NISP 1D Best $Swirl_{mean}$: Axial velocity in the S-duct.

In the axial velocity plot, the presence of a second region of flow detachment is clearly visible on the up wall and it is also distinguishable in the fourth pressure recovery contour sub-plot.

A good Swirl contour result was obtained, mostly for low inlet velocity (figure 6.37). But as the inlet velocity increases, the swirl behaviour worsens. This phenomenon is confirmed by the high $Swirl_{stdv}$ value and also from table 6.16.

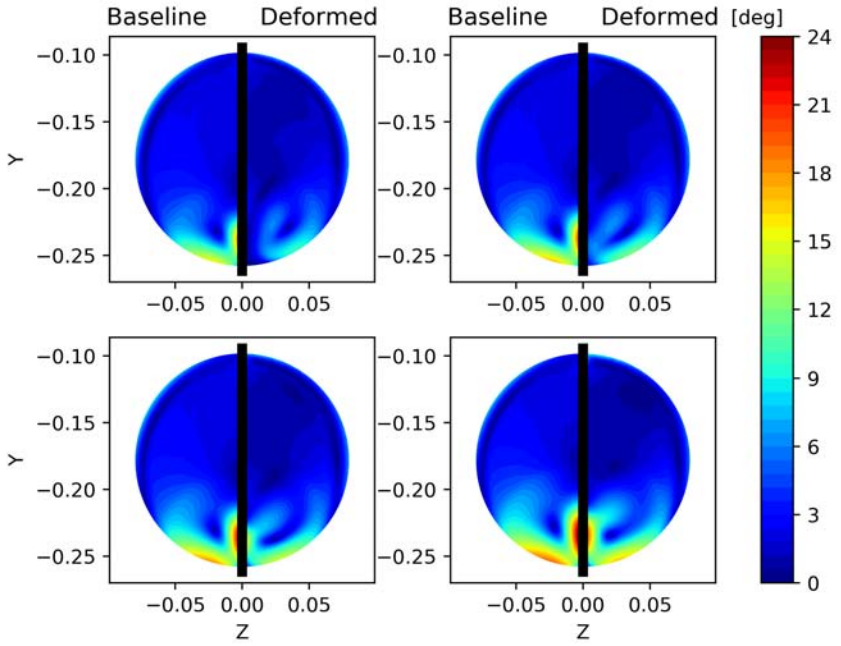


FIGURE 6.37: NISP 1D Best $Swirl_{mean}$: Swirl contour comparison at the AIP.

TABLE 6.16: NISP 1D Best $Swirl_{mean}$: Swirl value for each velocity sample.

	Swirl **Baseline**	**NISP 1D Best** $Swirl_{mean}$	**Improvement**
Sample 1	3.84619°	2.56679°	−33.26%
Sample 2	4.11063°	2.79463°	−32.01%
Sample 3	4.43925°	3.12398°	−29.63%
Sample 4	4.89146°	3.70047°	−24.35%

6.1.2.4 NISP 1D: Best Swirl stdv

The fourth configuration to analyse is the Best $Swirl_{stdv}$ design. In table 6.17, the differences in the objective functions values between de Best $Swirl_{stdv}$ design and the Baseline are displayed.

TABLE 6.17: NISP 1D Best $Swirl_{stdv}$: differences in the objective functions values between the Best $Swirl_{stdv}$ design and the Baseline.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
Best $Swirl_{stdv}$	0.035583	0.008600	3.186860°	0.017106°
Baseline	0.029124	0.007765	4.283883°	0.224200°

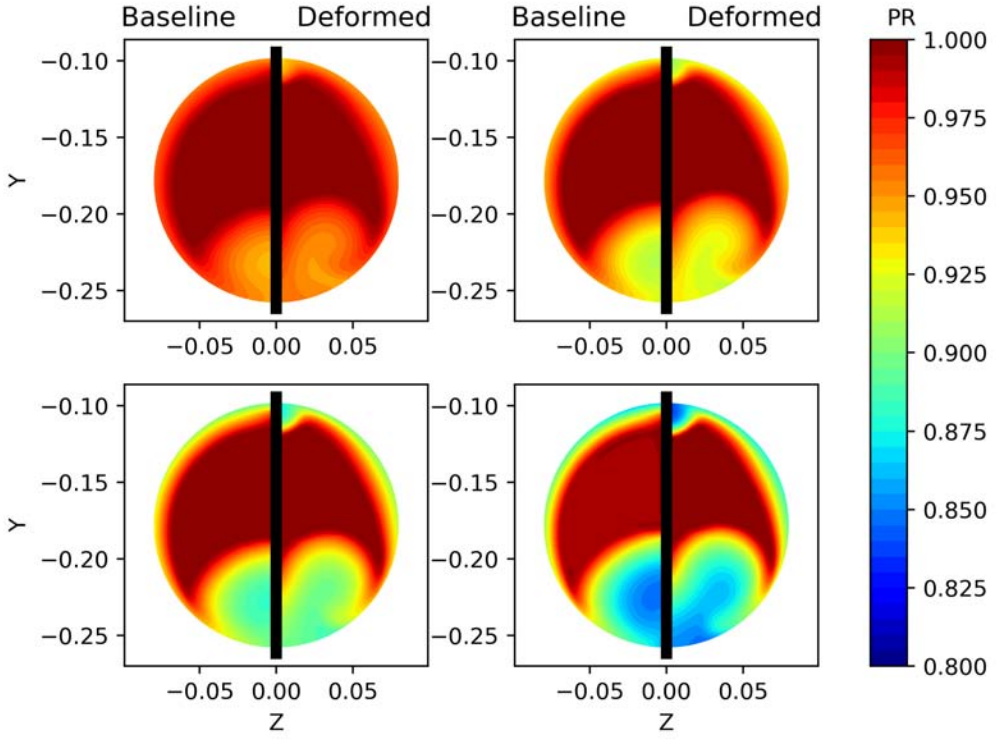


FIGURE 6.38: NISP 1D Best $Swirl_{stdv}$: PR contour comparison at the AIP.

In this new design configuration, the pressure recovery contour (figure 6.38) is similar to the Best $Swirl_{mean}$ one (figure 6.37). Moreover, the CP_{mean} value has worsened considerably. On the other hand, good results were obtained from the swirl contour. The four deformed swirl contours are almost identical and for the firs time the high swirl value area (in the middle) has completely disappeared in figure 6.39.

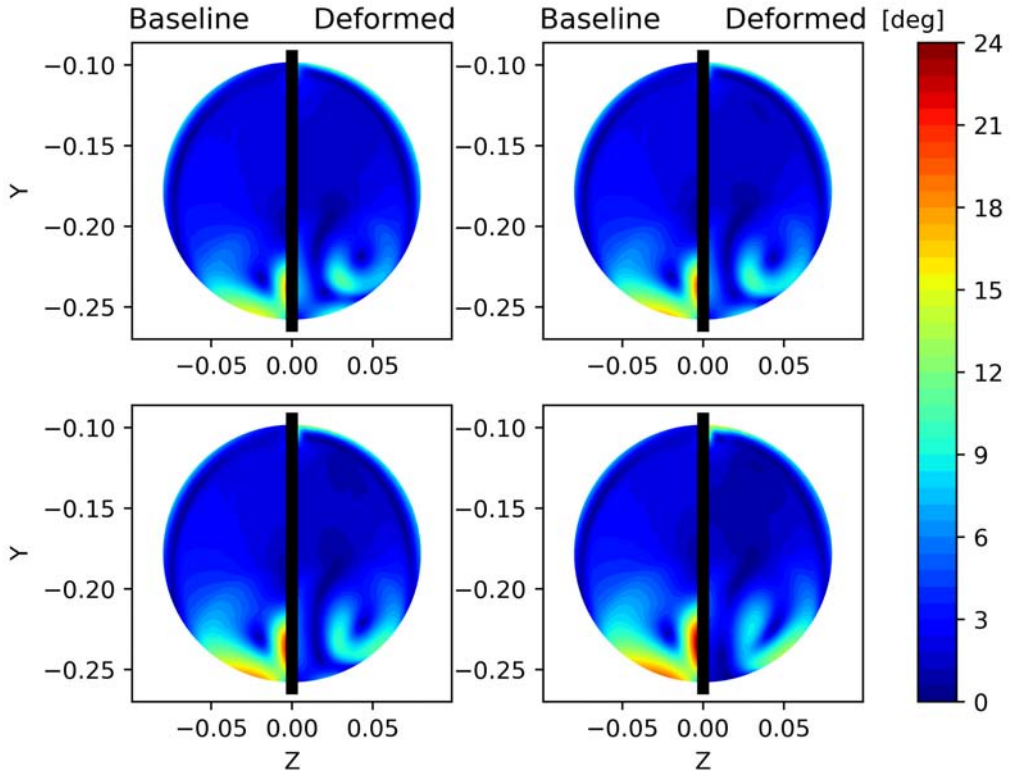


FIGURE 6.39: NISP 1D Best $Swirl_{stdv}$: Swirl contour comparison at the AIP.

6.1.2.5 NISP 1D: Parallel coordinate optima design

In this section, the first optima design of table 6.11 was analysed.

TABLE 6.18: NISP 1D optima 1: differences in the objective functions values between the optima 1 design and the Baseline.

Design	CP_{mean}	CP_{stdv}	$Swirl_{mean}$	$Swirl_{stdv}$
optima	0.027629	0.00716	3.34194°	0.200365°
Baseline	0.029124	0.007765	4.283883°	0.224200°

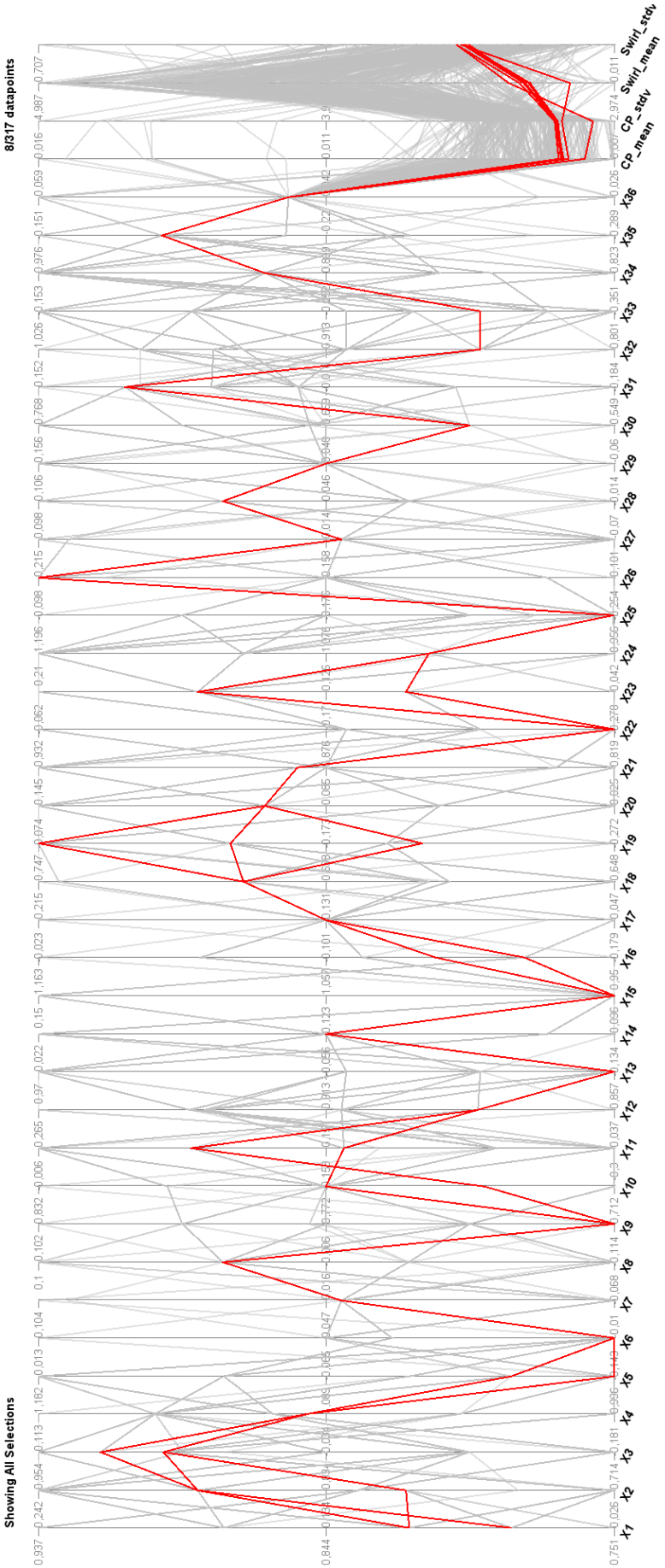


FIGURE 6.40: NISP 1D: Parallel Coordinate.

The PR contour plot(figure 6.41) and the Swirl contour plot (figure6.42) are very similar to the contour plots of the NISP 1D Best CP_{mean} design (figure 6.30, 6.31).

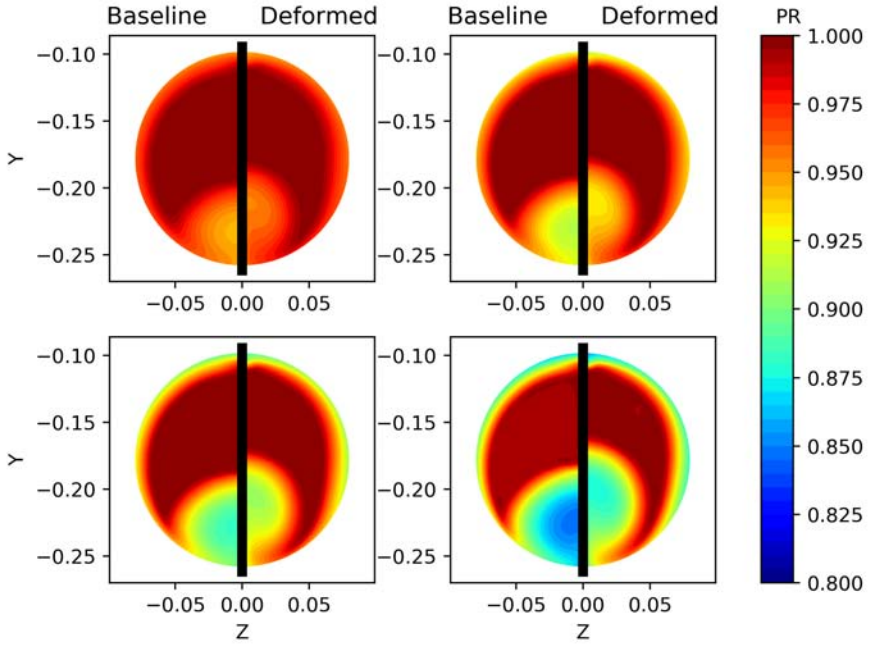


FIGURE 6.41: NISP 1D optima 1: PR contour comparison at the AIP.

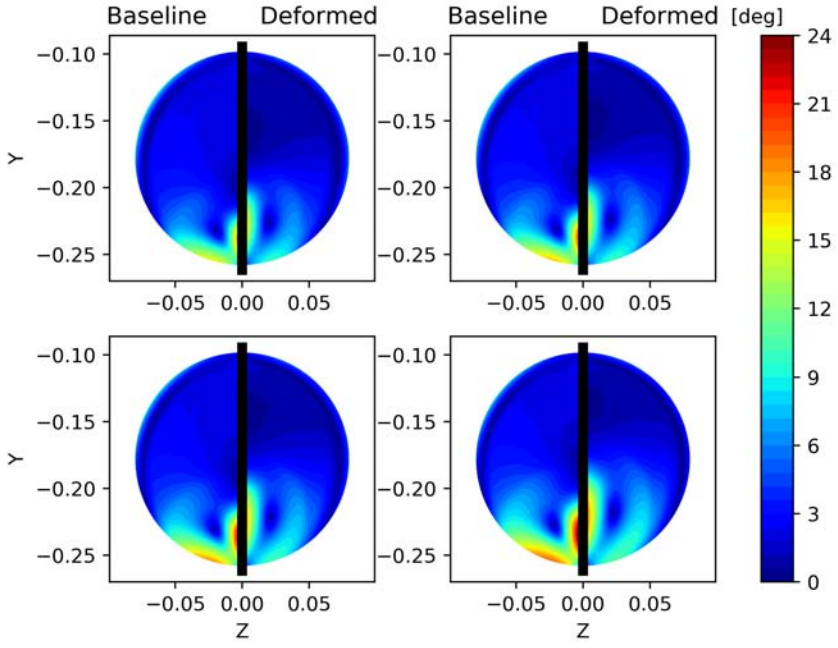


FIGURE 6.42: NISP 1D optima 1: Swirl contour comparison at the AIP.

The only difference between the two designs can be found in the Swirl values. These values are closer to zero in the first 90° of the optima 1 contour sub-plots design. Good values of the DC60 were obtained in the second and third sector (i.e $60° \div 180°$), mostly in the third and fourth velocity sample (figure 6.43). On the other hand, the DC15 had an excellent improvement of its values for all the sectors inside the range $45° \leq \theta \leq 180°$ (figure 6.44).

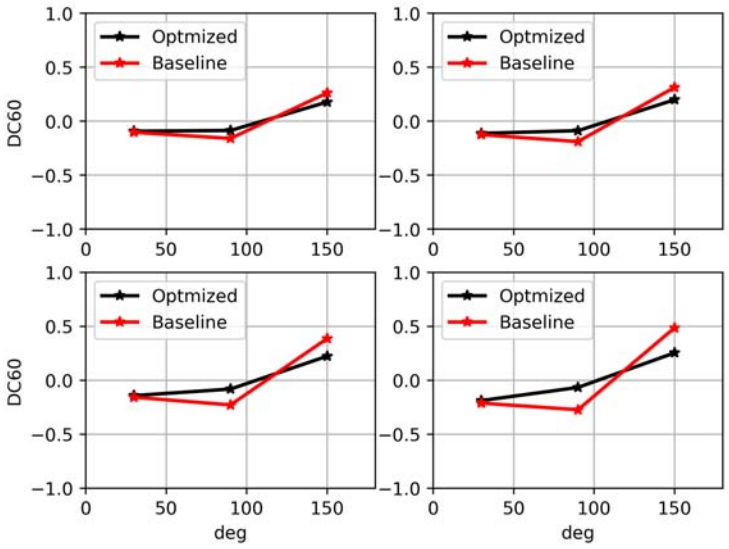


FIGURE 6.43: NISP 1D optima 1: DC60.

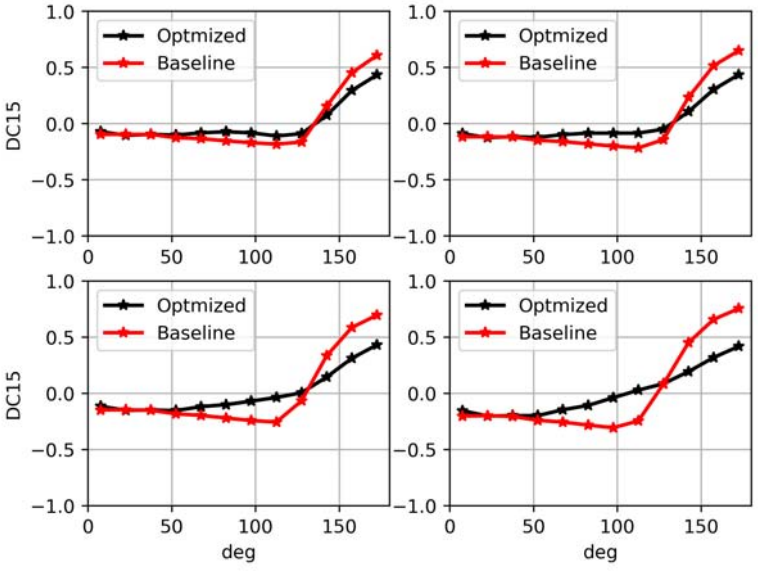


FIGURE 6.44: NISP 1D optima 1: DC15.

6.2 2D Robust Optimisation Problem

The 2D optimisation problem is characterised by the following starting parameters:

- **Uncertainty input:** Inlet Velocity with mean $\mu_v \approx 196.53$ m/s and standard deviation value $\sigma_v = 10$ m/s. Inlet Flux Deviation with mean $\mu_\beta = 0°$ m/s and standard deviation value $\sigma_\beta = 3°$ m/s. These two uncertainties variables are described with a Gaussian PDF.
- **Objective Functions:** CP_{mean}, CP_{stdv}
- **Constraints:** The mean and the standard deviation values of the Swirl were setted as constraints. $Swirl_{mean} < 5°$, $Swirl_{stdv} < 1°$.
- **Polynomial Chaos type and order:** the typology is the Hermite Chaos polynomial and the order p is set to 2 to lower the computational cost.

6.2.1 NIPC 2D

To implement the NIPC 2D cycle, the two uncertainties input were both sampled with the Latin Hypercube Sampling technique (LHS). The MOTS software was stopped after having successfully evaluated 192 configurations. As explained at the end of Chapter 5, the NIPC uncertainty quantification technique needs $P + 1$ deterministic evaluations to find the PC coefficients and the stochastic outputs (i.e objective functions). In this case the number of evaluations were equal to six. This means that $192 \times 6 = 1152$ CFD Fluent simulations converged. The time for a single fluent evaluation is ≈ 43.9 minutes, so the time that was necessary to obtain all these results was ≈ 843 hours. However, the amount of time above estimated does not take into account all the configurations that were analysed but subsequently discarded if they did not respect the Swirl mean and Swirl standard deviation constraints or if a structured geometry mesh was not possible to be created by ANSYS ICEM. The amount of discarded designs is 54.
This time the objective functions are two and a bi-dimensional Pareto front plot is sufficient to show all the MOTS evaluated points and the optimised designs. The Pareto front that was obtained is illustrated in figure 6.45.

The numerical results of the six points of the Pareto front are reported in the table 6.19.

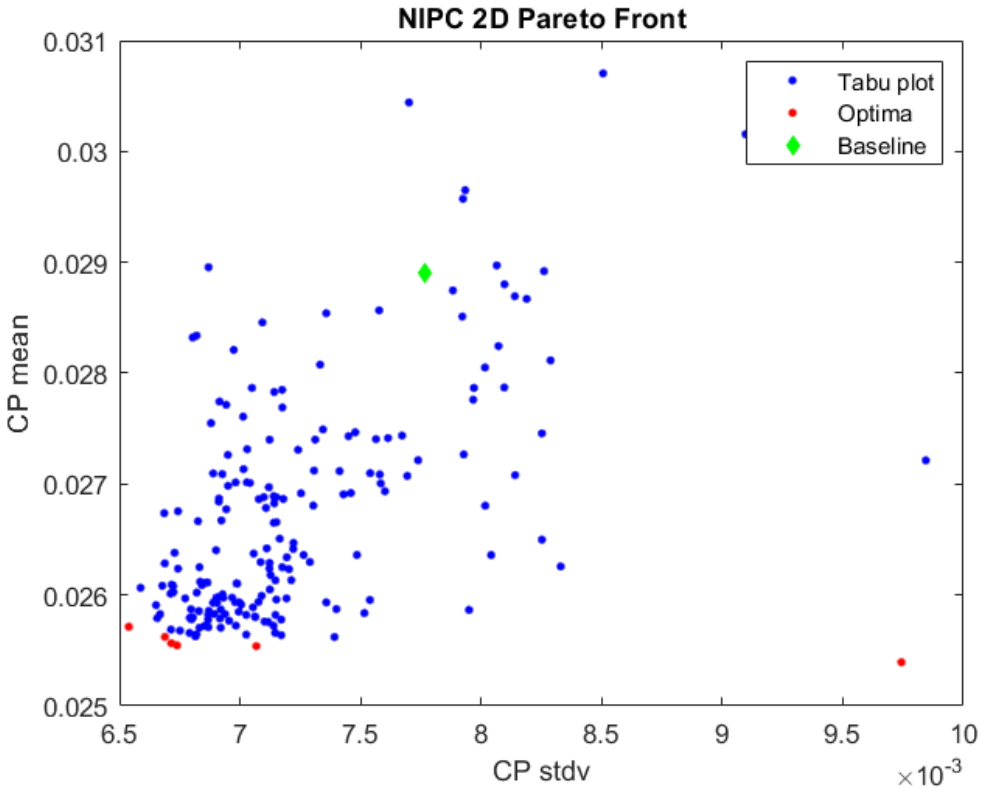


FIGURE 6.45: NIPC 2D: Pareto front.

TABLE 6.19: NIPC 2D optima designs results obtained from the Pareto front figure 6.45.

Design	CP_{mean}	CP_{stdv}
optima 1	0.025545	0.006739
optima 2	0.025393	0.009743
optima 3	0.025622	0.006689
optima 4	0.025713	0.006539
optima 5	0.025565	0.006715
optima 6	0.025538	0.007068
Baseline	0.028905	0.007768

Three of the six configurations were analysed and they will be labelled as follow:

- NIPC 2D: Best CP_{mean} design (optima 2 of table 6.19)
- NIPC 2D: Best CP_{stdv} design (optima 4 of table 6.19)
- NIPC 2D: Trade-off design (optima 1 of table 6.19)

6.2.1.1 NIPC 2D: Best CP mean

The first configuration taken into account in this paragraph is the optima design that has achieved the best result in terms of CP_{mean}. Table 6.20 highlights the differences between the objective functions values obtained from the optimisation loop and the post-processing analysis ones. Moreover, we can see how these differences are greater in comparison with the NIPC 1D and this is due to the presence of a second uncertainty (i.e Inlet Flux Deviation). But this latter statement is not always true, it depends on the inlet sampled couple.

TABLE 6.20: NIPC 2D Best CP_{mean}: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}
Optimisation	0.025393	0.009743
Post-proc.	0.025643	0.006805
Baseline	0.028905	0.007768

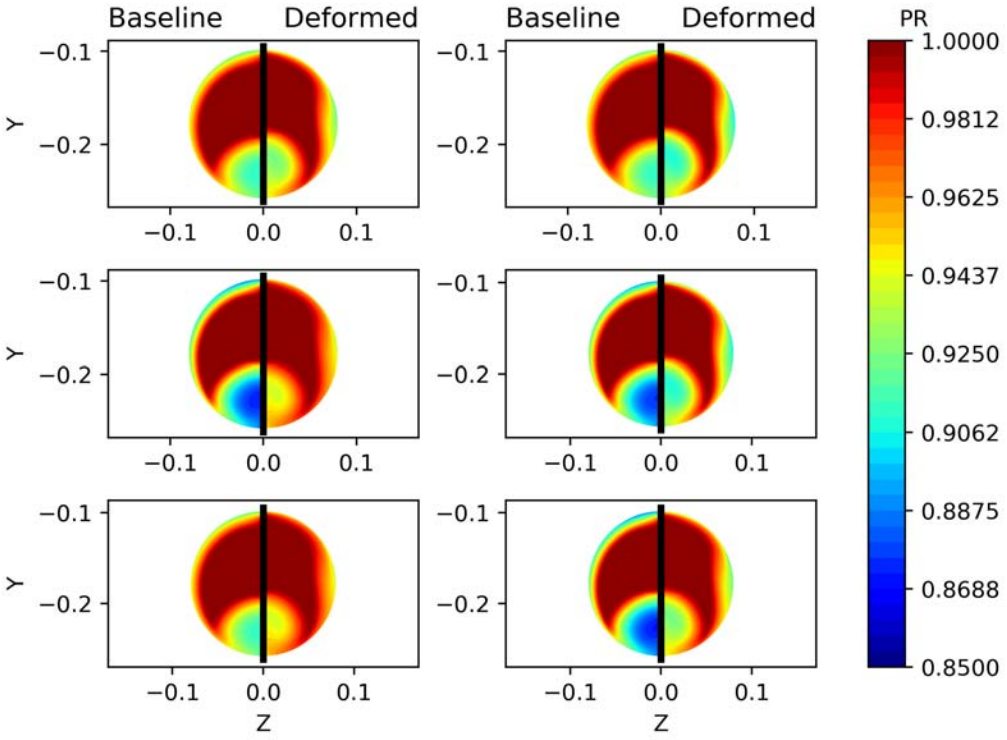


FIGURE 6.46: NIPC 2D Best CP_{mean}: PR contour comparison at the AIP.

In figure 6.46, it is possible to see the pressure recovery value (PR) evaluated at the AIP surface for each inlet velocity and flux deviation sampled couple. The order

of the sub-plots is random because the couples of the two sampled uncertainties were chosen randomly. In Appendix B it is possible to find how these uncertainties couples were chosen. As in the NIPC 1D and NISP 1D post-processing analysis, the left half of each AIP contours drawing represents the pressure recovery at the AIP surface of the baseline, instead the right one represents the deformed one.
It is possible to observe that in all the six contour comparison, the values of the Pressure recovery of the deformed design are higher than the baseline values. In fact, in table 6.21, all the values of $CP = 1 - PR$ that were found for each sampled couple of inlet velocity and flux deviation are reported.

TABLE 6.21: NIPC 2D Best CP_{mean}: CP value for each sampled couple.

	CP **Baseline**	**NIPC 2D Best** CP_{mean}	**Improvement**
Sample 1	0.02324	0.02308	-0.7%
Sample 2	0.02960	0.0293	-1.0%
Sample 3	0.02897	0.01727	-40.38%
Sample 4	0.03347	0.02950	-11.86%
Sample 5	0.02341	0.01712	-26.87%
Sample 6	0.03277	0.02326	-29.02%

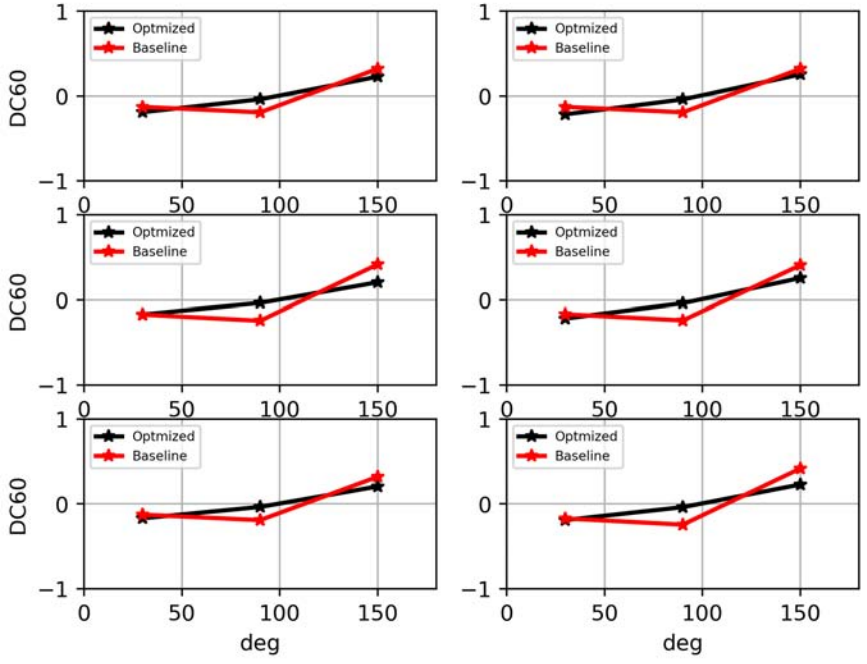


FIGURE 6.47: NIPC 2D Best CP_{mean}: DC60.

The DC60 values in figure 6.47 have improved in the second and third sector (i.e $60° \div 180°$) and also the DC15 in figure 6.48 has improved a lot in all the sectors with $\theta \geq 70°$. Moreover, their values have reached ≈ 0 in the sectors inside the range $70° \div 100°$.

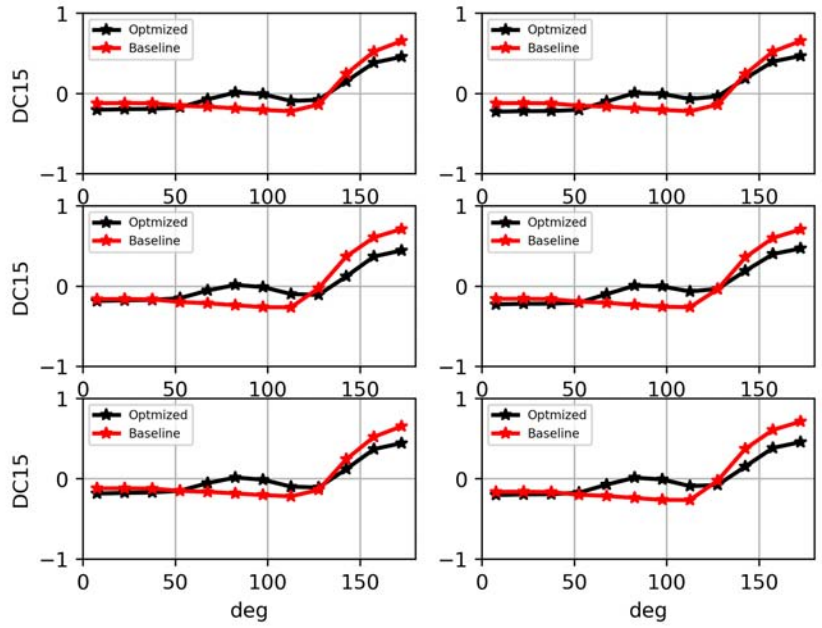


FIGURE 6.48: NIPC 2D Best CP_{mean}: DC15.

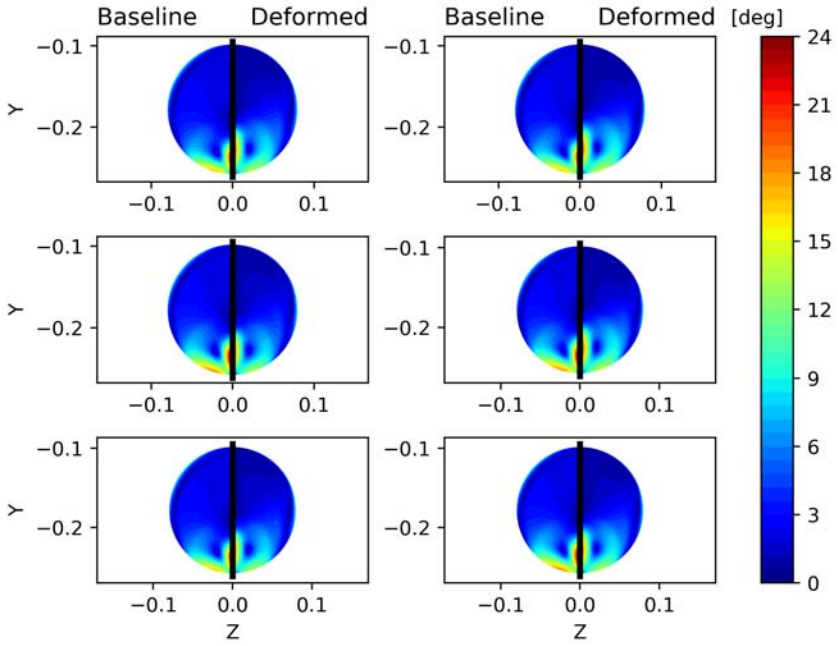


FIGURE 6.49: NIPC 2D Best CP_{mean}: Swirl contour comparison at the AIP.

6.2.1.2 NIPC 2D: Best CP stdv

Now we look at the Best CP_{stdv} design configuration. The following table 6.22 highlights the differences between the objective functions values obtained from the optimisation loop and the post-processing analysis ones.

TABLE 6.22: NIPC 2D Best CP_{stdv}: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}
Optimisation	0.025713	0.006539
Post-proc.	0.025808	0.007090
Baseline	0.028905	0.007768

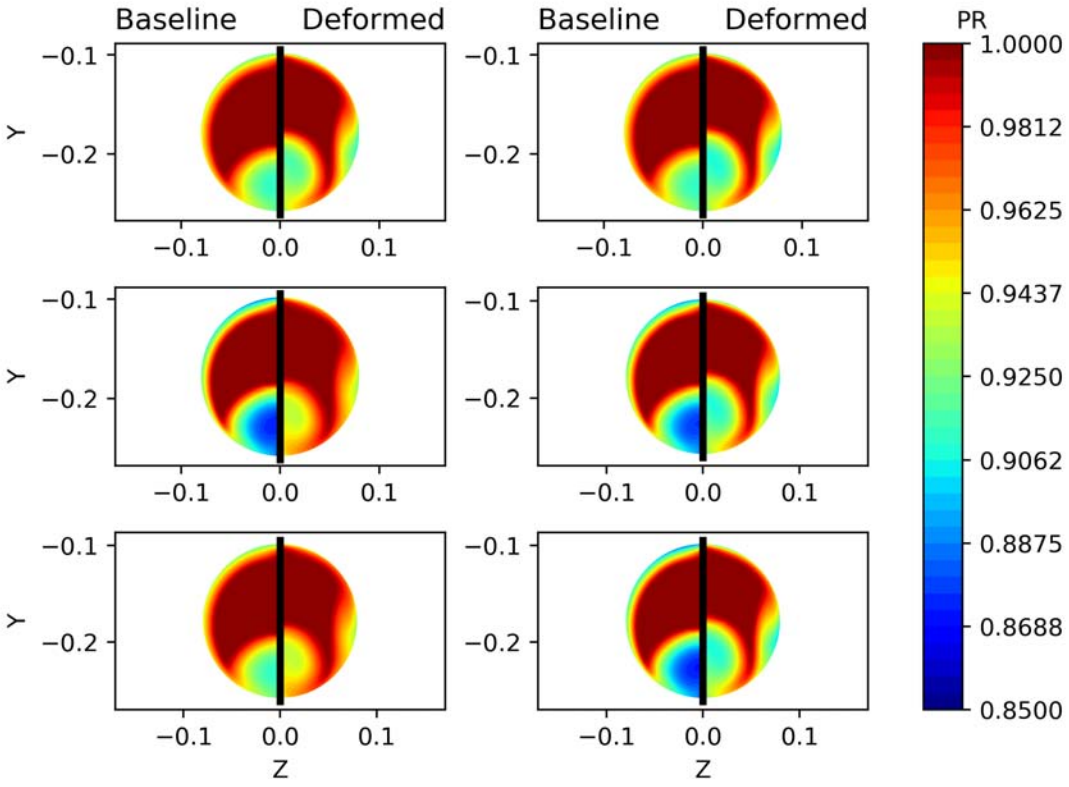


FIGURE 6.50: NIPC 2D Best CP_{stdv}: PR contour comparison at the AIP.

In figure 6.50 it is possible to see the formation of a new flow detachment area and in table 6.23 all the values of $CP = 1 - PR$ that were found for each sampled couple of inlet velocity and flux deviation are reported.

TABLE 6.23: NIPC 2D Best CP_{stdv}: CP value for each sampled couple.

	CP **Baseline**	**NIPC 2D Best** CP_{stdv}	**Improvement**
Sample 1	0.02324	0.02714	+16.78%
Sample 2	0.02960	0.03103	+4.83%
Sample 3	0.02897	0.01956	−32.48%
Sample 4	0.03347	0.03078	−8.04%
Sample 5	0.02341	0.01936	−17.30%
Sample 6	0.03277	0.03110	−5.10%

Similary to the Best CP_{cmean} design, also the DC60 values of the Best CP_{stdv} in figure 6.51 have improved in the second and third sector (i.e $60° \div 180°$).
The same was for the DC15 values in figure 6.52 for all the sectors with $\theta \geq 70°$. Moreover, its values have reached ≈ 0 in the sectors inside the range $70° \div 100°$.

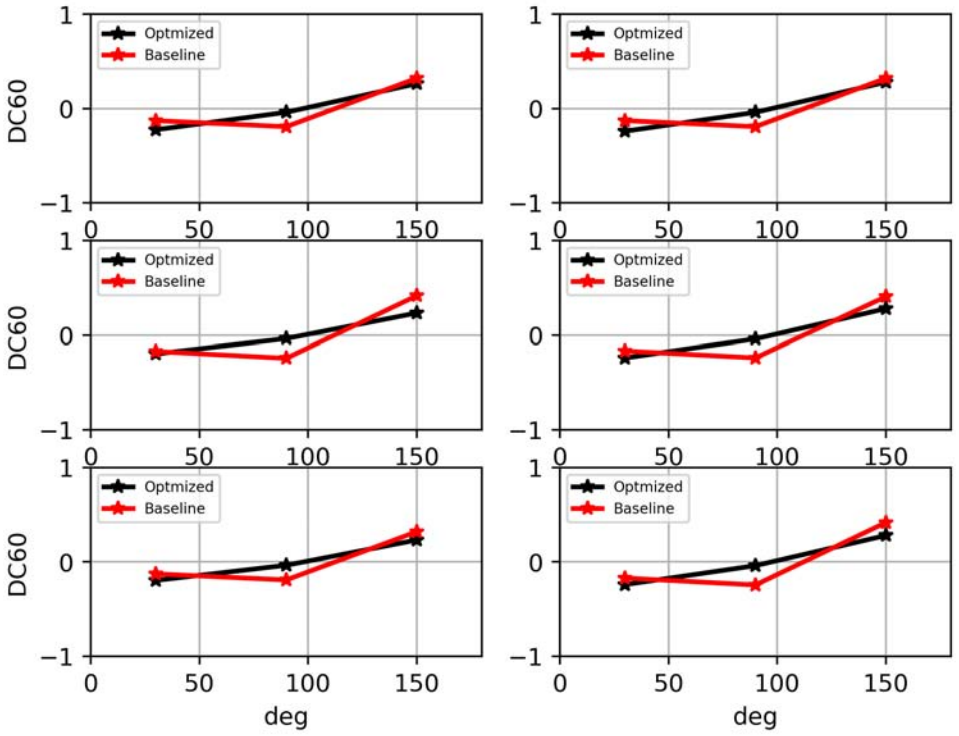


FIGURE 6.51: NIPC 2D Best CP_{stdv}: DC60.

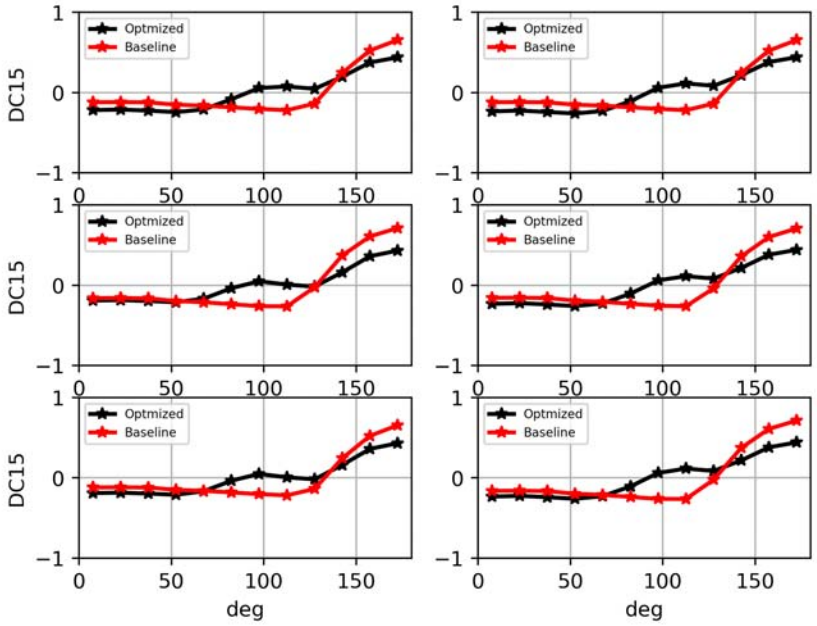


FIGURE 6.52: NIPC 2D Best CP_{stdv}: DC15.

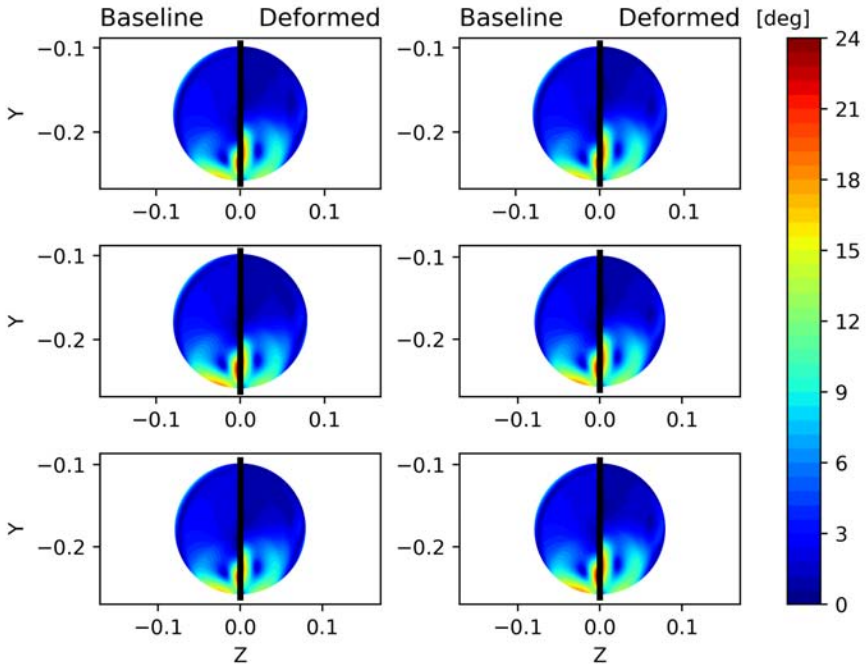


FIGURE 6.53: NIPC 2D Best CP_{stdv}: Swirl contour comparison at the AIP.

6.2.1.3 NIPC 2D: Trade-off design

The last analysed configuration is the trade-off between the Best CP_{mean} design and the Best CP_{stdv} one. The following table 6.24 highlights the differences between the objective functions values obtained from the optimisation loop and the post-processing analysis ones.

TABLE 6.24: NIPC 2D Trade-off: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}
Optimisation	0.025545	0.006739
Post-proc.	0.025676	0.006818
Baseline	0.028905	0.007768

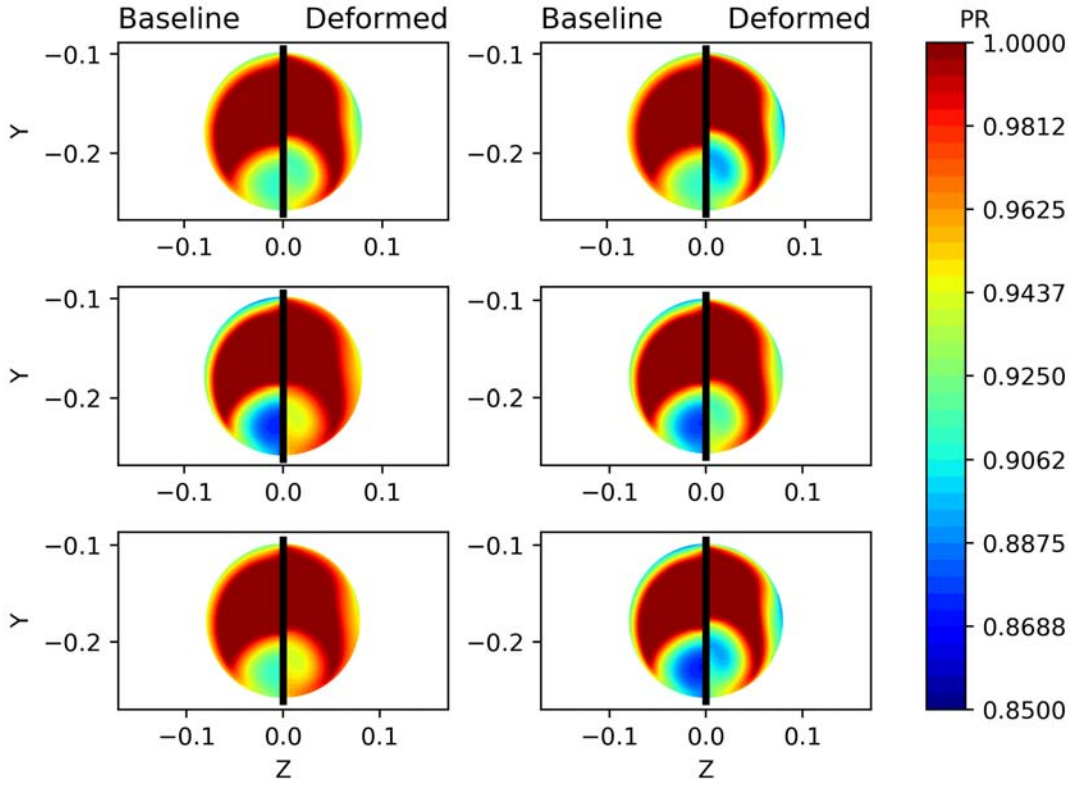


FIGURE 6.54: NIPC 2D Trade-off: PR contour comparison at the AIP.

The great improvement in the DC15 values of the sectors between $65° \div 125°$ and in the DC60 value of the middle sector (i.e $60° \div 120°$) are very interesting (figure 6.55 6.56).

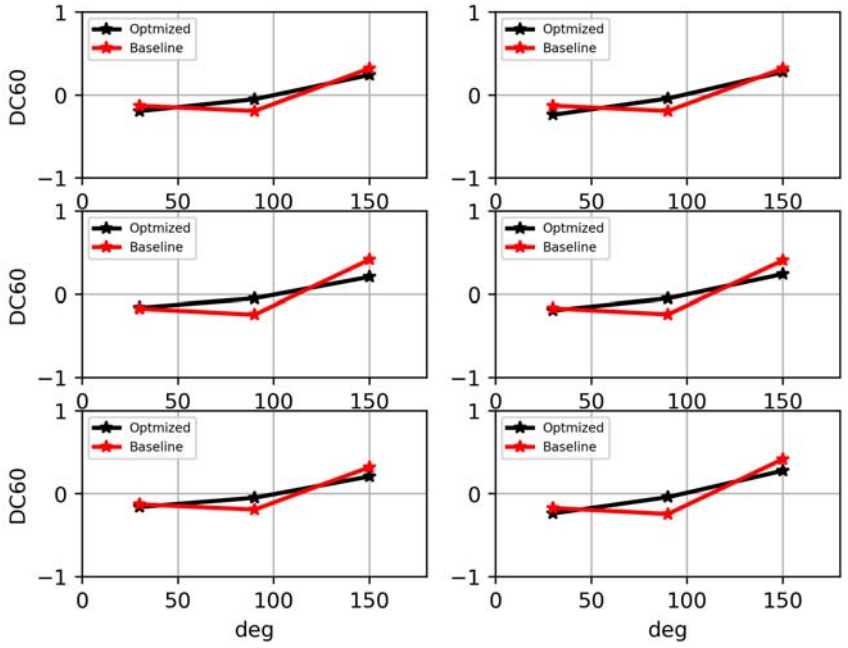


FIGURE 6.55: NIPC 2D Trade-off: DC60.

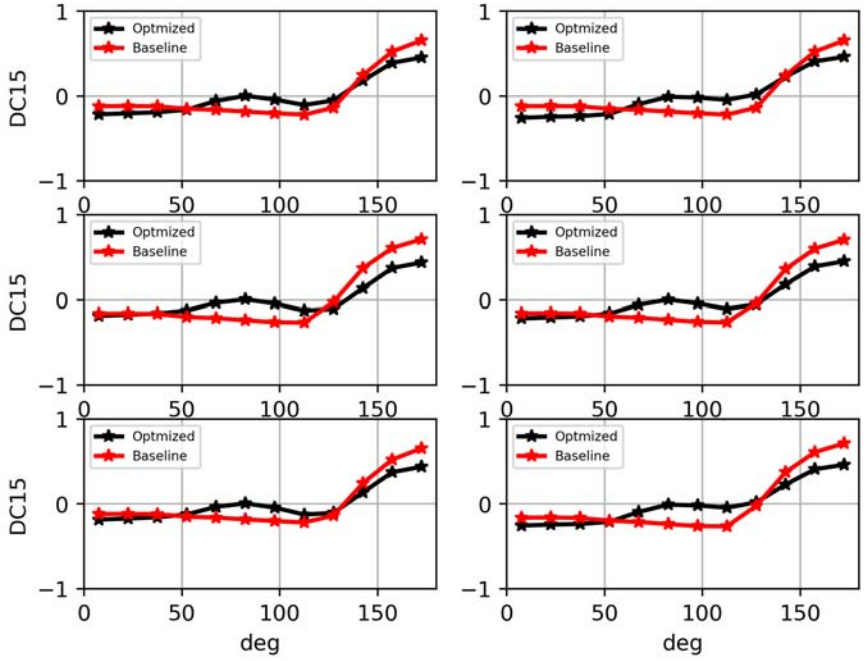


FIGURE 6.56: NIPC 2D Best Trade-off: DC15.

6.2.2 NISP 2D

To implement the NISP 2D cycle, the two uncertainties input were sampled with the Hermite-Gauss Sampling technique (HG). The MOTS software was stopped after having successfully evaluated 135 configurations. As explained at the end of Chapter 5, the NIPC uncertainty quantification technique needs $(p+1)^n$ deterministic evaluations to find the $P+1$ PC coefficients and the stochastic outputs (i.e objective functions). In this case the number of evaluations were equal to nine. This means that $135 \times 9 = 1215$ CFD Fluent simulations converged. The time for a single fluent evaluation is ≈ 38.8 minutes, so the time that was necessary to obtain all these results was ≈ 785 hours. However, the amount of time above estimated does not take into account all the configurations that were analysed but subsequently discarded if they did not respect the Swirl mean and Swirl standard deviation constraints or if a structured geometry mesh was not possible to be created by ANSYS ICEM. The amount of discarded designs is 30.
The objective functions are two as in the NIPC 2D, and a bi-dimensional Pareto front plot is sufficient to show all the MOTS evaluated points and the optimised designs. The Pareto front that was obtained is illustrated in figure 6.57.

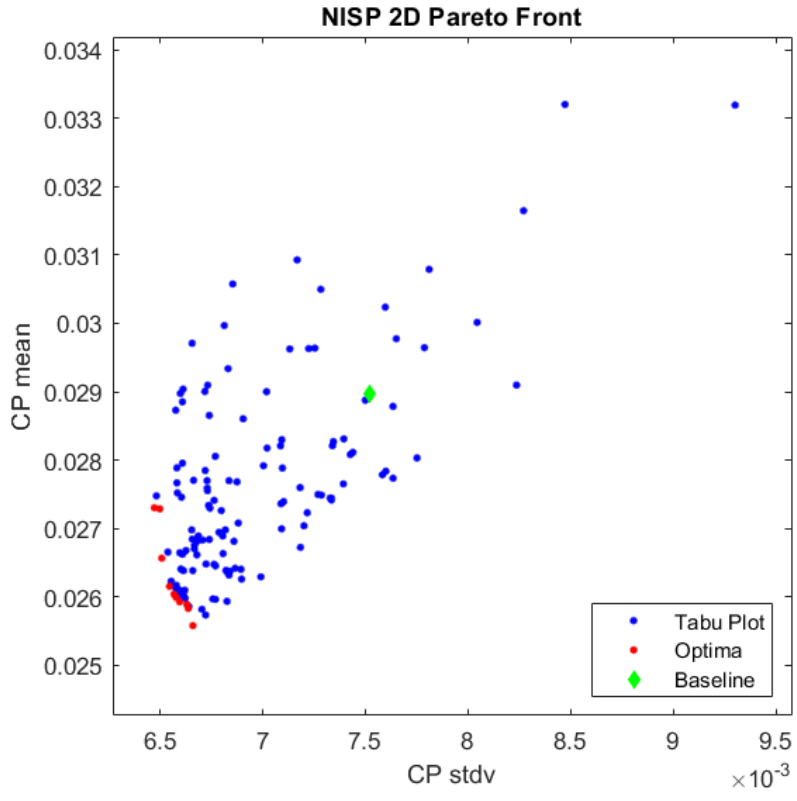


FIGURE 6.57: NISP 2D: Pareto front.

The numerical results of the ten points of the Pareto front are reported in the table 6.25.

TABLE 6.25: NISP 2D optima designs results obtained from the Pareto front figure 6.57.

Design	CP_{mean}	CP_{stdv}
optima 1	0.025891	0.00663
optima 2	0.027287	0.006499
optima 3	0.025833	0.006637
optima 4	0.02558	0.00666
optima 5	0.027304	0.006472
optima 6	0.026155	0.006546
optima 7	0.026566	0.006508
optima 8	0.025929	0.006595
optima 9	0.02604	0.006568
optima 10	0.025994	0.006578
Baseline	0.028973	0.007522

Three of the six configurations were analysed and they will be labelled as follow:

- NISP 2D: Best CP_{mean} design (optima 4 of table 6.25)
- NISP 2D: Best CP_{stdv} design (optima 5 of table 6.25)
- NISP 2D: Trade-off design (optima 7 of table 6.25)

In the three subsequent analyses, it will not be possible to show all the plots obtained from the three analysed designs. This is due to the excessive number of samplings carried out (i.e nine), which has drastically increased the number of contour sub-plots per figure. Therefore, it was decided to show only single plots referring to a single sampling of the nine for the DC60 and DC15 values.
Furthermore, there will not be any descriptions for the pressure recovery and swirl results of the three designs because their contour plots are very similar between them. This choice was made to avoid the risk of having a part of the thesis with redundant figures.
Only the contours of the Best CP_{mean} design will be plotted in order to show at least one to the readers.

6.2.2.1 NISP 2D: Best CP mean

The first configuration taken into account in this paragraph is the optima design that has achieved the best result in terms of CP_{mean} for the NISP 2D. Table 6.26 highlights the differences in the objective functions values between the Best CP_{mean} design and the Baseline.

TABLE 6.26: NISP 2D Best CP_{mean}: differences in the objective functions values between the Best CP_{mean} design and the Baseline.

Design	CP_{mean}	CP_{stdv}
Best CP_{mean}	0.02558	0.00666
Baseline	0.028973	0.007522

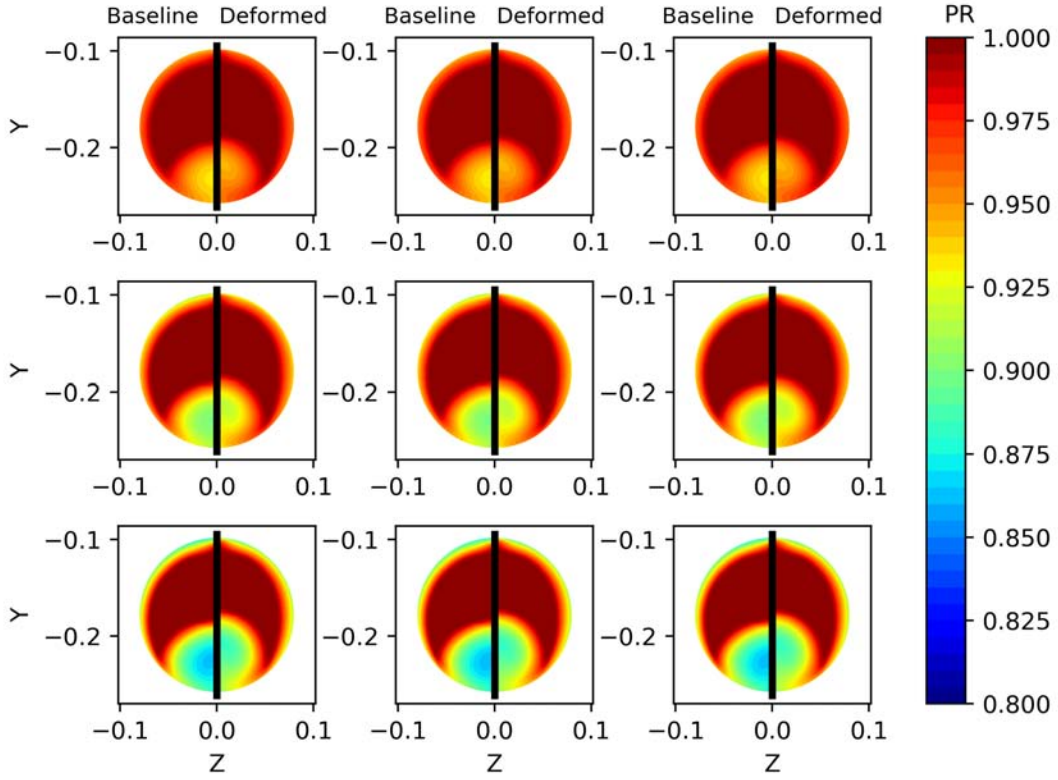


FIGURE 6.58: NISP 2D Best CP_{mean}: PR contour comparison at the AIP.

It is important to specify that in the NISP 2D, the order of the sub-plots follows the order of the sampled couples. As a matter of fact, the velocity value increases row by row, while the Flux deviation increases column by column. There was fair improvements in the DC60 and DC15 parameters. In the following plots (figure

6.59 6.60), we can observe the DC60 and DC15 values of every sector for one of the nine samples. At least, there were no improvements in swirl values (figure 6.61).

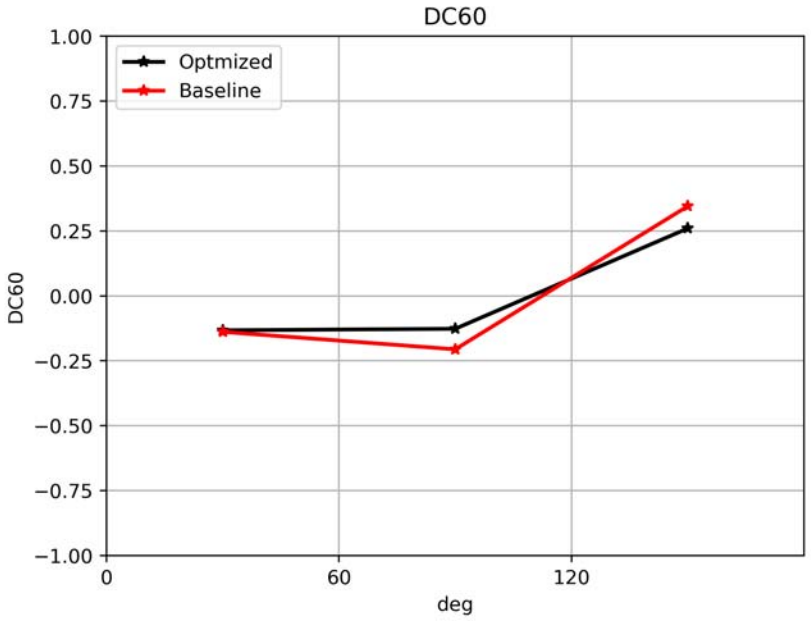


FIGURE 6.59: NISP 2D Best CP_{mean}: DC60 values for only one sample.

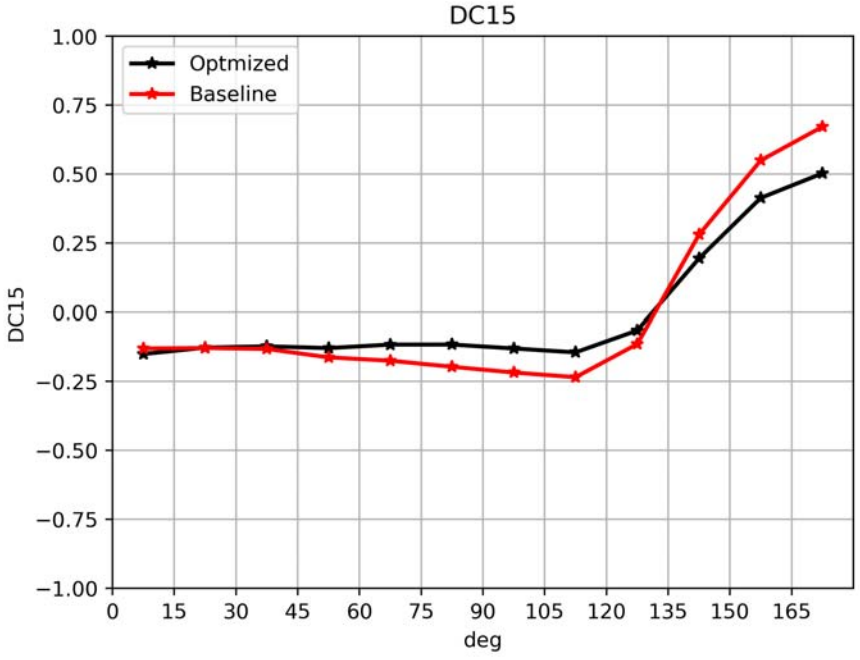


FIGURE 6.60: NISP 2D Best CP_{mean}: DC15 values for only one sample.

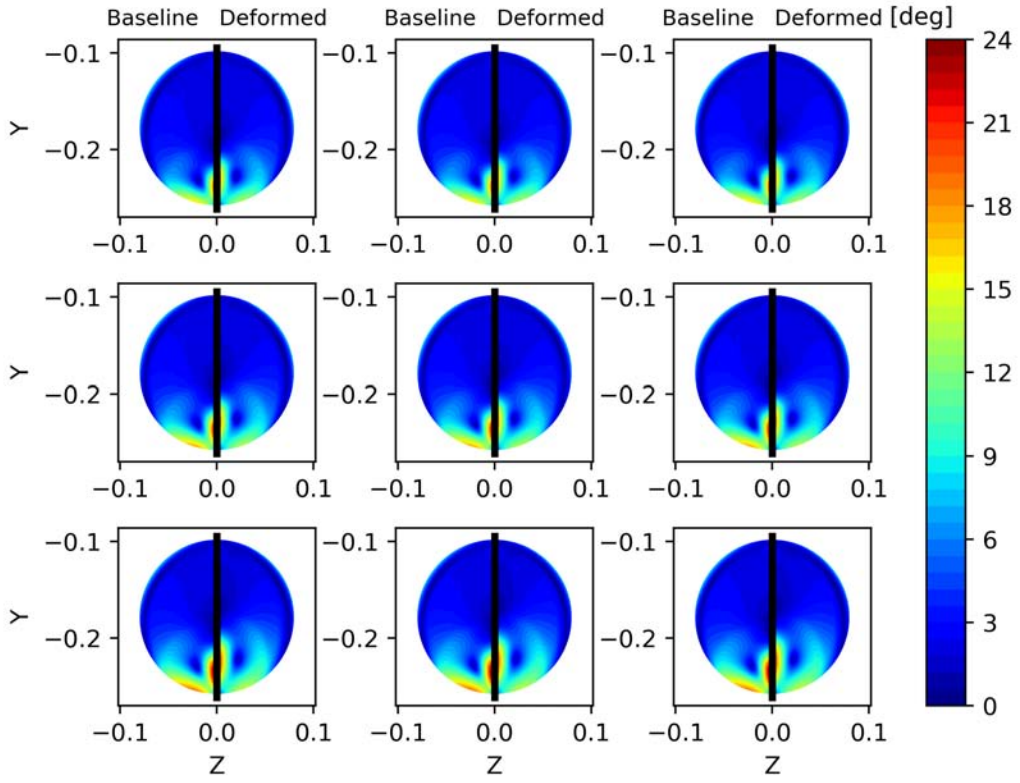


FIGURE 6.61: NISP 2D Best CP_{mean}: Swirl contour comparison at the AIP.

6.2.2.2 NISP 2D: Best CP stdv

The second configuration taken into account is the optima design that has achieved the best result in terms of CP_{stdv}. Table 6.27 highlights the differences in the objective functions values between the Best CP_{stdv} design and the Baseline.

TABLE 6.27: NISP 2D Best CP_{stdv}: differences in the objective functions values between the Best CP_{stdv} design and the Baseline.

Design	CP_{mean}	CP_{stdv}
Best CP_{stdv}	0.027304	0.006472
Baseline	0.028973	0.007522

For the first time ever, in the deformed design, there were DC60 and DC15 values better than the baseline one in all the sectors for all the samples. However, these values are not close enough to zero. As said before, only one DC60 and DC15 plot of only one sample was displayed.

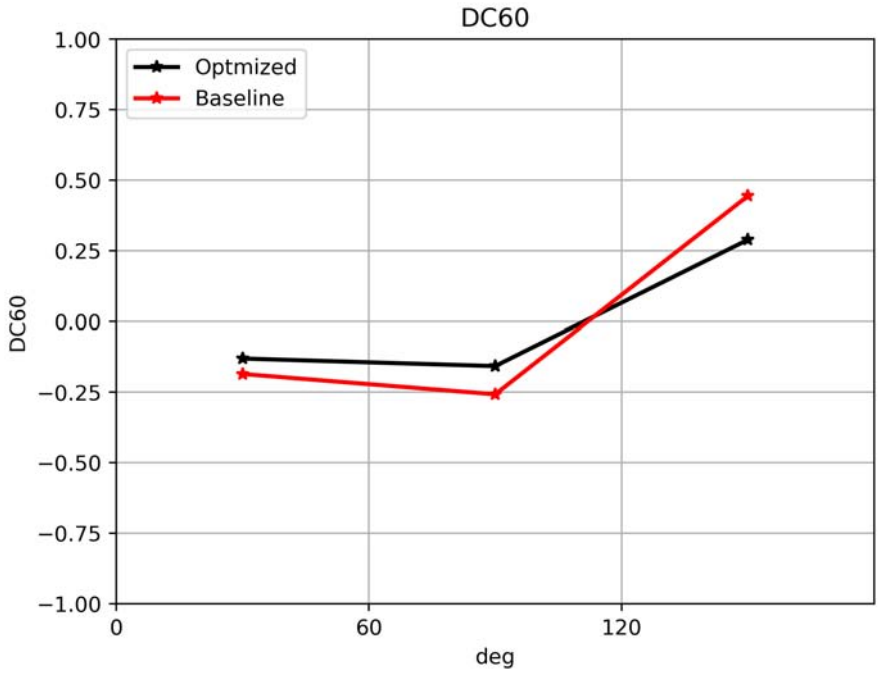


FIGURE 6.62: NISP 2D Best CP_{stdv}: DC60 values for only one sample.

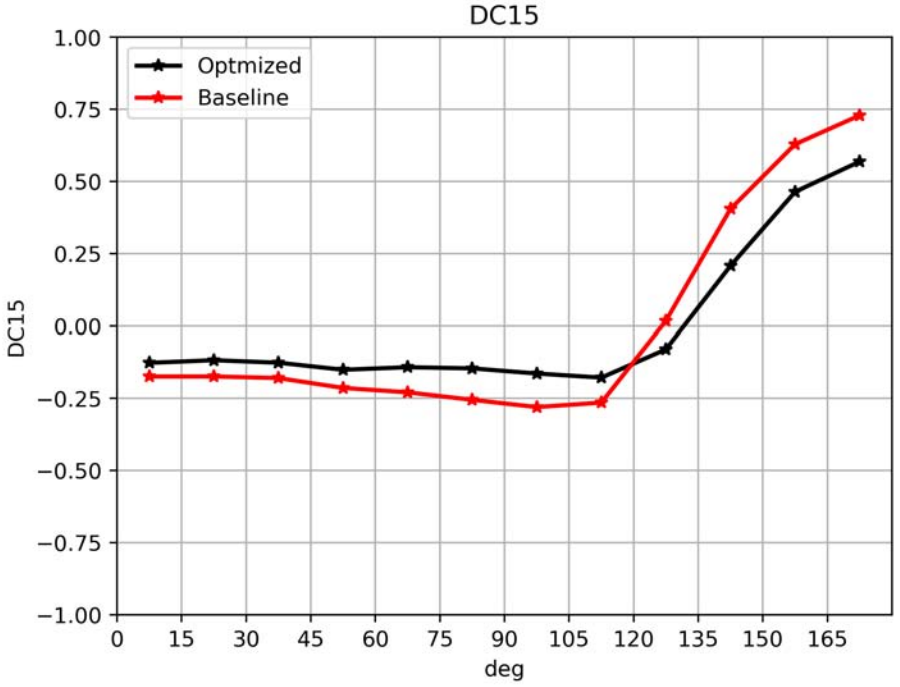


FIGURE 6.63: NISP 2D Best CP_{stdv}: DC15 values for only one sample.

6.2.2.3 NISP 2D: Trade-off design

The last analysed configuration is the trade-off between the Best CP_{mean} design and the Best CP_{stdv} one. Table 6.28 highlights the differences in the objective functions values between the Best CP_{stdv} design and the Baseline. There is nothing

TABLE 6.28: NISP 2D Trade-off: differences in the objective functions values between the Trade-off design and the Baseline.

Design	CP_{mean}	CP_{stdv}
Trade-off	0.026566	0.006508
Baseline	0.028973	0.007522

to highlight in this configuration, because all the plot results and values are very similar to the NISP 2D Best CP_{mean} design.

6.3 Manufacturing Robust Optimisation Problem

For the first time ever, a manufacturing uncertainty input was inserted in a robust design optimisation problem for S-ducts. This new uncertainty parameter is the Bending Angle, and together with the inlet velocity, they are the input couple of uncertainties of this robust optimisation problem. In other words, another 2D problem was implemented and analysed, and the following are the starting parameters:

- **Uncertainty input:** Inlet Velocity with mean $\mu_v \approx 196.53$ m/s and standard deviation value $\sigma_v = 10$ m/s. Bending angle with mean $\mu_\alpha = \theta_{max} = 0°$ m/s and standard deviation value $\sigma_\alpha = 4°$ m/s. These two uncertainty variables are described with a Gaussian PDF.
- **Objective Functions:** CP_{mean}, CP_{stdv}
- **Constraints:** The mean and the standard deviation values of the Swirl were setted as constraint. $Swirl_{mean} < 5.5°$, $Swirl_{stdv} < 1°$.
- **Polynomial Chaos type and order:** the typology is the Hermite Chaos polynomial and the order p is set to 2.

The Non Intrusive Point Collocation method (NIPC) was used as uncertainties quantification method. The reasons of this choice are the following:

- The LHS sampling technique randomly samples the uncertainties input. In this way a new configuration is tested every time.
- The NIPC method is not affected by the *course of dimensionality*.

In the next sections we will see which uncertainty quantification method was chosen, and how the the output objective functions values and the behaviour of the fluid inside the duct were influenced by the Bending Angle uncertainty input.

6.3.1 NIPC MNF

To implement the NIPC MNF cycle, the two uncertainties input were both sampled with the Latin Hypercube Sampling technique (LHS). The MOTS software was stopped after having successfully evaluated 69 configurations. As explained at the end of Chapter 5, the NIPC uncertainty quantification technique needs $P + 1$ deterministic evaluations to find the PC coefficients and the stochastic outputs (i.e objective functions). In this case the number of evaluations were equal to six. This means that $71 \times 6 = 426$ CFD Fluent simulations converged. The time for a single fluent evaluation is ≈ 49.6 minutes, so the time that was necessary to obtain all these results was ≈ 342 hours. However, the amount of time above estimated does not take into account all the configurations that were analysed but subsequently discarded if they did not respect the Swirl mean and Swirl standard deviation constraints or if a structured geometry mesh was not possible to be created by ANSYS ICEM. The amount of discarded designs is 52, the 42.9% of the total evaluated ones. As a matter of fact, the two constraint in the Swirl mean and standard deviations, are very stringent. Moreover, in this new robust optimisation cycle, for each sampled values of the Bending angle, a new different geometry was built. In this way, ANSYS ICEM had to built for six times (i.e six deterministic evaluation) different structured mesh. Therefore, this high variability in the geometry and mesh was the second cause of the high amount of discarded design. In fact, ANSYS ICEM was often not able to create a structured mesh for each sampled values of the Bending angle.

It must be remarked that this manufacturing robust optimisation work is not finished yet because more data need to be collected in order to find better results. As the other 2D problems, the objective functions are two and a bi-dimensional Pareto front plot is sufficient to show all the MOTS evaluated points and the optimised designs. The Pareto front that was obtained is illustrated in figure 6.64.

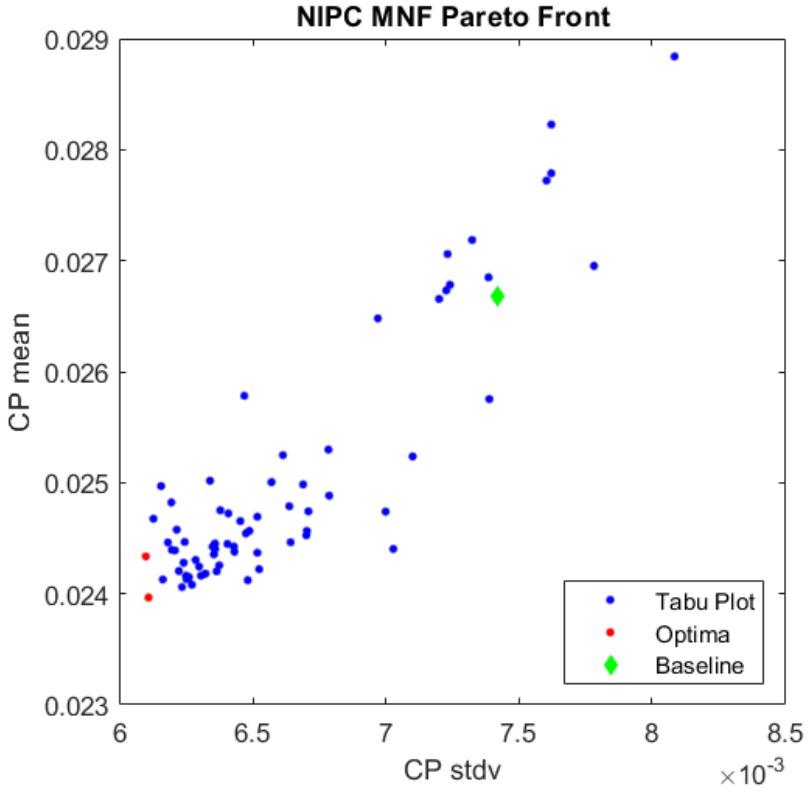


FIGURE 6.64: NIPC MNF: Pareto front.

The optima results obtained from the cycle were only two (table 6.29), and this was due to the low number of successfully evaluated designs.

TABLE 6.29: NIPC MNF optima designs results obtained from the Pareto front figure 6.64.

Design	CP_{mean}	CP_{stdv}
optima 1	0.023966	0.006108
optima 2	0.024336	0.006098
Baseline	0.026676	0.007418

Despite the few designs evaluated so far, the first results are encouraging. In fact, the 0.025 threshold for the CP_{mean} values has been broken maintaining the standard deviation values in the average of those seen so far. As in the NIPC 1D and NIPC 2D optimisation loops, it is very important to specify that the output values of the objective functions obtained from the NIPC MNF optimisation loop are slightly different from the ones obtained after the post-processing analysis of the chosen optima designs. As a matter of fact, also here the sampled velocities and bending angles were not recorded during the optimisation loop. Moreover as we know, the LHS method chooses randomly the inlet velocities and bending angles values inside their own CDF.

6.3.1.1 NIPC MNF: optima 1

The first configuration taken into account in this paragraph is the optima 1 design. Table 6.30 highlights the differences between the objective functions values obtained from the optimisation loop and the post-processing analysis ones.

TABLE 6.30: NIPC MNF Best optima 1: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}
Optimisation	0.023966	0.006108
Post-proc.	0.024038	0.006146
Baseline	0.026676	0.007418

As in the NIPC 2D, the order of the contour sub-plots is random because the couples of the two sampled uncertainties were chosen randomly.
In the contour figure 6.65, it is not easy to see the improvement in the CP values. As a matter of fact, in table 6.31, all the values of $CP = 1 - PR$ that were found for each inlet sampled couple of uncertainties are reported, in order to help the reader to understand and see the improvements.
The swirl values have not improved and we can see it in figure 6.68. Moreover, it is interesting to observe how the DC60 an DC50 values have changed a lot from sample to sample but it is not possible to say that the values have improved (figure 6.66 6.67).

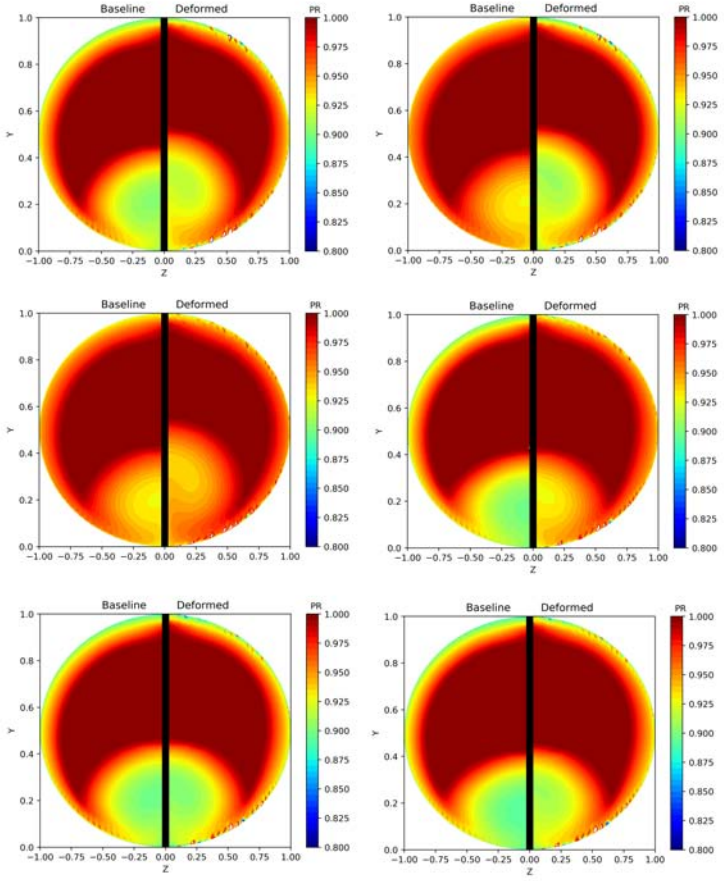


FIGURE 6.65: NIPC MNF optima 1: PR contour comparison at the AIP.

TABLE 6.31: NIPC MNF optima 1: CP value for each velocity sample.

	CP **Baseline**	**NIPC MNF optima 1**	**Improvement**
Sample 1	0.02796	0.0257	−8.08%
Sample 2	0.01897	0.0286	+50.76%
Sample 3	0.018945	0.0195	−2.93%
Sample 4	0.02802	0.0191	−31.83%
Sample 5	0.03031	0.0285	−5.97%
Sample 6	0.03045	0.0255	−16.25%

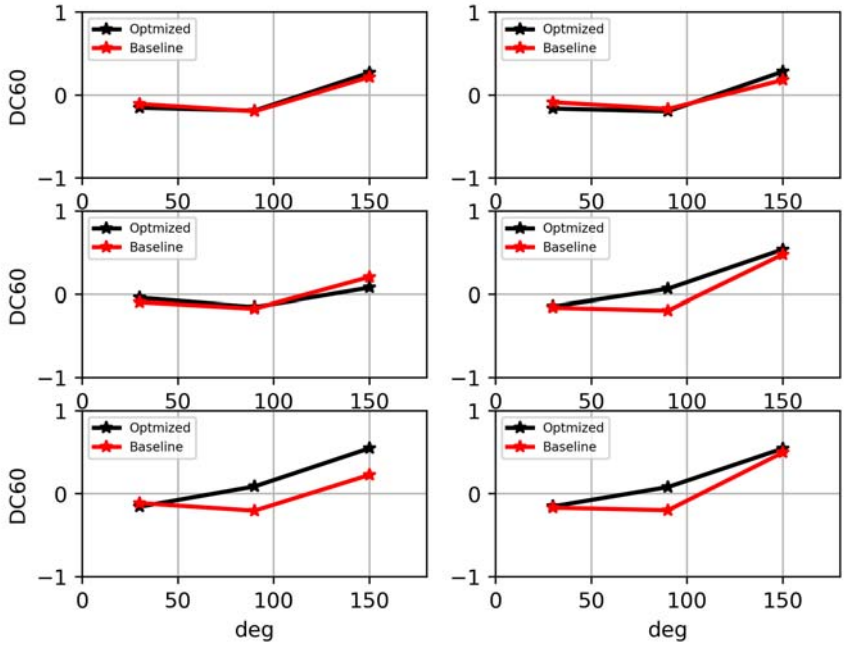


FIGURE 6.66: NIPC MNF optima 1: DC60.

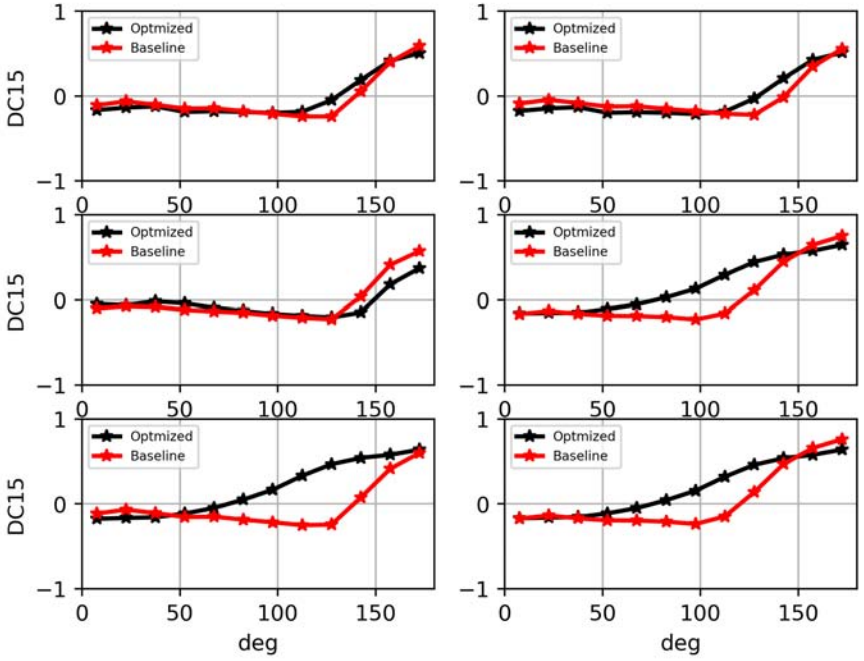


FIGURE 6.67: NIPC MNF optima 1: DC15.

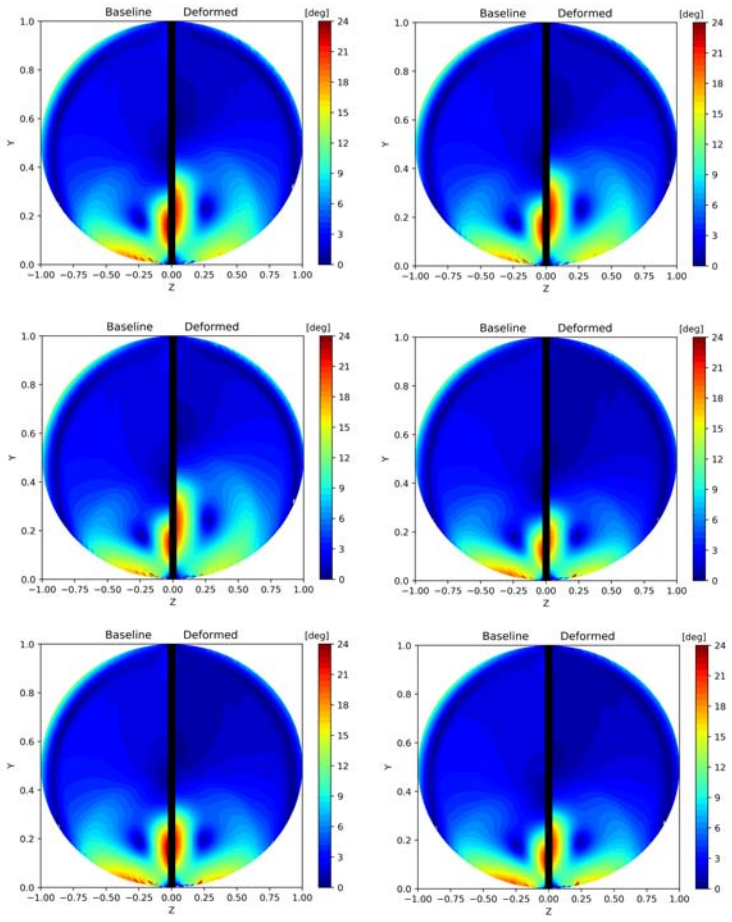


FIGURE 6.68: NIPC MNF optima 1: Swirl contour comparison at the AIP.

6.3.1.2 NIPC MNF: optima 2

The first configuration taken into account in this paragraph is the optima 2 design. Table 6.32 highlights the differences between the objective functions values obtained from the optimisation loop and the post-processing analysis ones.

TABLE 6.32: NIPC MNF Best optima 2: differences in the objective functions values due to the nature of the LHS sample technique.

Design	CP_{mean}	CP_{stdv}
Optimisation	0.024336	0.006098
Post-proc.	0.024357	0.006181
Baseline	0.026676	0.007418

The physical behaviour of the optima 2 configuration is very similar to that of the optima 1 and all the two designs plots and objective functions results confirm this

similarity. In fact, the two designs are very close to each other in the Pareto front (figure 6.64) and therefore it is normal that they behave in similar way.
If we had evaluated more points inside the design space, then we would have, probably analysed different designs with recognizable physical behaviours.
In table 6.33, all the values of $CP = 1 - PR$ that were found for each inlet sampled couple of uncertainties are reported.

TABLE 6.33: NIPC MNF optima 2: CP value for each velocity sample.

	CP **Baseline**	**NIPC MNF optima 2**	**Improvement**
Sample 1	0.02796	0.0250	−10.59%
Sample 2	0.01897	0.0301	+58.67%
Sample 3	0.018945	0.0303	+59.93%
Sample 4	0.02802	0.0298	+6.35%
Sample 5	0.03031	0.0254	−16.20%
Sample 6	0.03045	0.0211	−30.71%

From table 6.33, we can observe that the CP results of the optima 2 design are not always better than the Baseline ones, though they have a remarkable difference in the CP_{mean} result. This is due to the randomly nature of the LHS sampling technique.

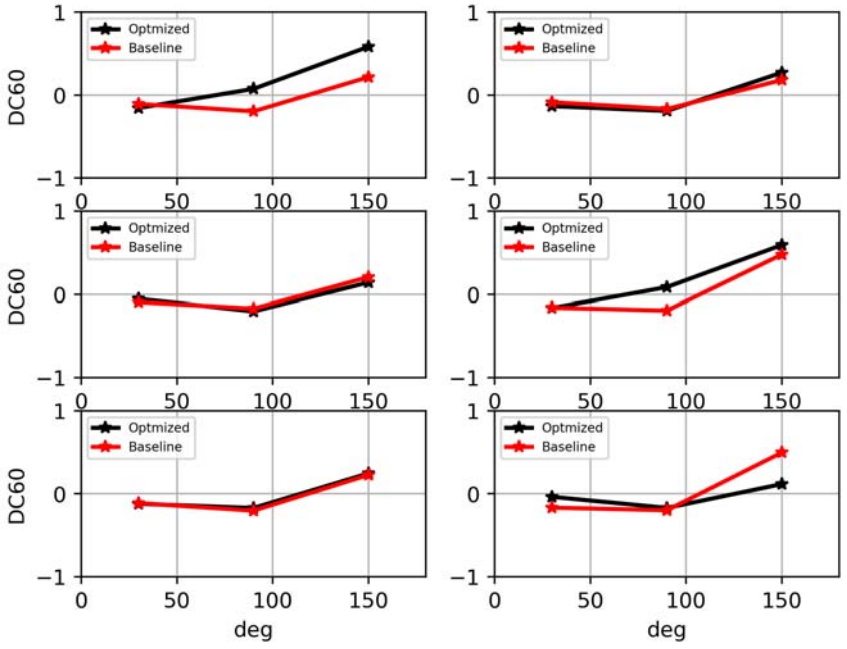


FIGURE 6.69: NIPC MNF optima 2: DC60.

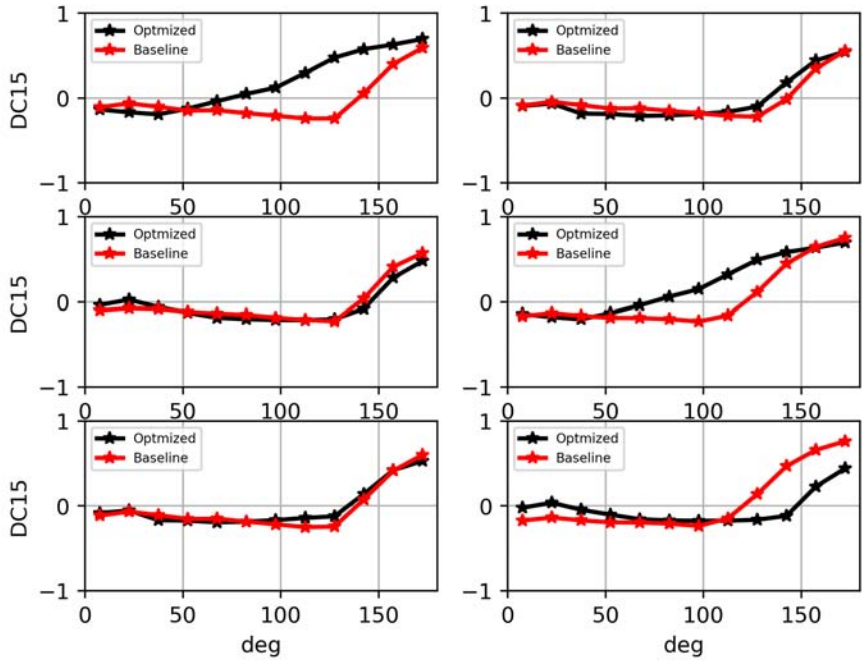


FIGURE 6.70: NIPC MNF optima 2: DC15.

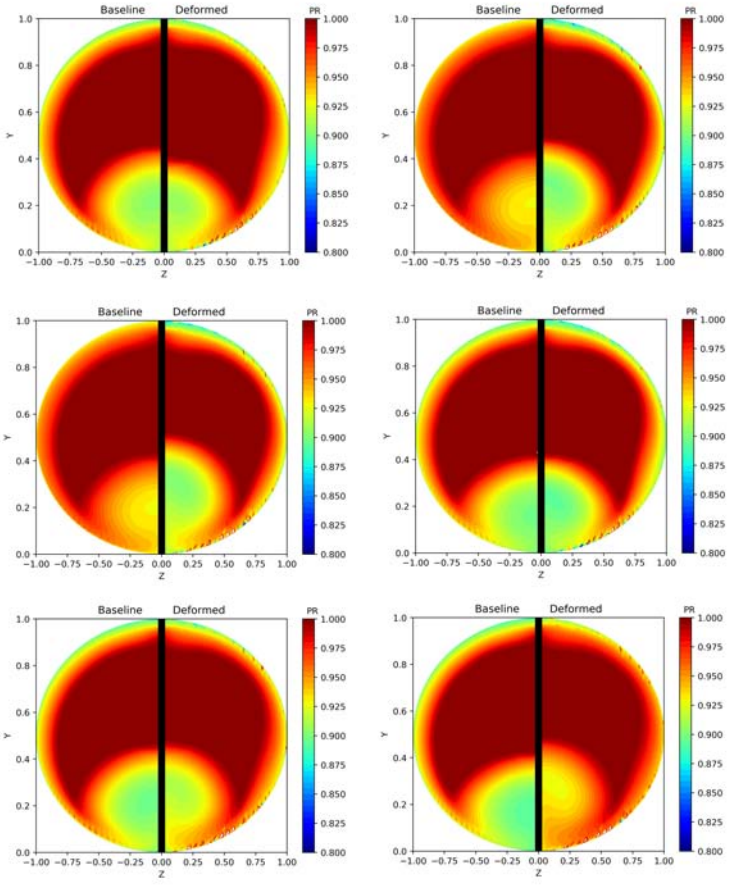


FIGURE 6.71: NIPC MNF optima 2: PR contour comparison at the AIP.

6.4 MOTS Machine Learning optimisation

A Machine learning optimisation cycle was developed in collaboration with Mattia Longato. Unfortunately, this optimisation cycle is partially developed but it is correct to mention it because we obtained the first predicted Pareto front results from it.

The aim of this machine learning cycle is to support the above mentioned robust optimisation problems in order to speed up the research of new optima designs.

First of all, two-thirds of the evaluated designs and their respective results were collected from the above mentioned robust optimisation problems and used to built a *prediction model* (i.e an algorithm) for two objective functions (CP_{mean}, CP_{stdv}). After that, these models were implemented inside a MOTS (Multi- objective Tabu Search) and used to predict the objective functions values of new design configurations created by the loop.

It must be remarked that as first *prediction model* for S-duct built so far, it was decided to use as input variables only the 36 design parameters without considering the input uncertainties. This choice gave us the possibility to use the results of all the 1D and 2D problems to build the model.

The MOTS software was stopped after having successfully evaluated 500 configurations. The pareto front that was obtained is displayed in figure 6.72.

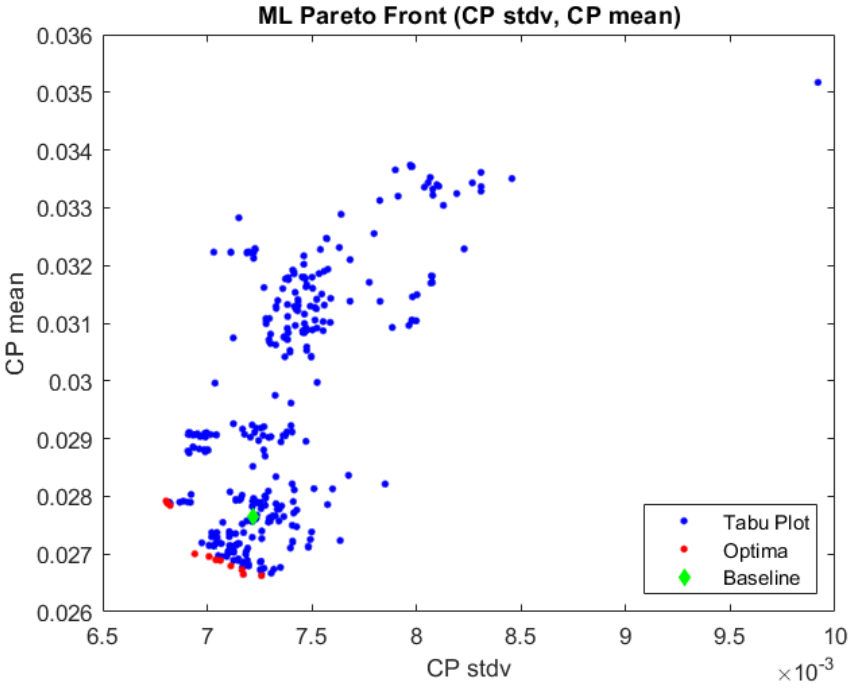


FIGURE 6.72: MOTS Machine Learning Pareto front

The prediction of the results of the new designs is accurate only if these new designs are located, in the design space, near the ones used to built the model. In other word, the further we move away from the design space used to build the model, the more inaccurate will be the results.
Therefore, we decided to evaluate and predict only 500 new designs because in this way it was possible to evaluate the new designs that share, more or less, the same design space as the ones used to build the model.

No post-processing analysis on the optima design has been made because the accuracy of this Machine Learning optimisation cycle is not too high. In order to improve the accuracy in the predicted results, it would be better if also the uncertainties variables were added as input parameters to create the *prediction model*.

Chapter 7

Conclusion

In this thesis many objectives have been pursued. The existent robust optimisation of Dal Magro [3] was extended developing 5 different robust optimisations cycles with two different Non Intrusive uncertainty quantification techniques (NIPC and NISP). The 1D problems or better those problems with only one uncertainty input (Inlet Velocity) have been developed and improved with the addition of other two important objective functions ($Swirl_{mean}$ and $Swirl_{stdv}$) to the existing ones (CP_{mean} and CP_{stdv}). The results obtained from the 1D problems are encouraging. In fact, good values of all the four objective functions were obtained simultaneously with excellent improvements, mostly in the swirl.

Another step forward has been made with the development of the 2D problems, or better those problems with two input uncertainties. To develop these kinds of problems, a good comprehension of the Non Intrusive uncertainty quantification techniques was required.

The Inlet Flux deviation has been the second uncertainty input to be added. In the 2D problem, it was decided to minimise only two objective functions (CP_{mean}, CP_{stdv}) in order to speed up the research of the optima designs performed by the algorithm (Tabu Search).

Good CP_{mean} results were obtained from the NIPC 2D and NISP 2D cycles A good improvement in the CP_{mean} values was also achieved if compared to the ones obtained by Dal Magro in his 1D problem [3].

For the first time, a manufacturing uncertainty in the bending angle was added to the input uncertainties. This new manufacturing uncertainty together with the Inlet Velocity uncertainty were quantified with the NIPC technique and used as the input of a new 2D problem labelled NIPC MNF. Very interesting and encouraging

results were found with a very big improvement in the CP_{mean} objective function. However, it is correct to specify that the research of the optima designs of this problem is not completed. In fact, one of my future works will be to finish the research and analysis of the optima designs for a publication.
The last work realised in this thesis is the implementation of a Machine Learning algorithm inside an optimisation problem in order to support the above mentioned robust optimisation cycles and to speed up the research of new optima designs. At the moment, this is a preliminary work because only the design parameters were used as input to build the *prediction model* but the Pareto front obtained is very promising. In the next future, The Machine learning optimisation cycle will be developed and completed with the addition of the uncertainties as input of the model.
In fact, this last purpose, together with the NIPC MNF will be the main focus of my next scheduled publication.

Bibliography

[1] A. Rigobello. A multi-objective shape optimization of an s-duct intake through nsga-ii genetic algorithm. Master's thesis, Universitá degli Studi di Padova, 2015.

[2] A. D'Ambros, T. Kipouros, P. Zachos, M. Savill, and E. Benini. Computational design optimization for s-ducts. *Design MDPI*, 2018.

[3] D. Dal Magro. Implementation of uncertainty management techniques in the design of s-ducts intakes. Master's thesis, Universitá degli Studi di Padova, 2019.

[4] M. M. Longato. Support s-duct studies with state of art machine learning techniques. Master's thesis, Universitá degli Studi di Padova, 2020.

[5] Jr. John D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill, 1221 Avenue of the Americas, New York, NY 10020, fifth edition edition, 2010.

[6] Steven R. Wellborn, Bruce A. Reichert, and Theodore H. Okiishi. A study of the compressible flow in a diffusing s-duct. Technical report, NASA, 1993.

[7] Ahmed F. El-Sayed and Mohamed S. Emeara. Intake of aero-engines: A case study. *International conference of engineering sciences and applications*, 2016.

[8] A. Mehdi. *Effect of swirl distortion on gas turbine operability*. PhD thesis, Cranfield University, 2014.

[9] Norbert C. Bessinger and Thomas Breuer. Basic principles - gas turbine compatibility - intake aerodynamic aspects. *Encyclopedia of Aerospace Engineering*, 2010.

[10] F. Carruso. S-duct aerodynamics; evaluation of uncertainties for s-duct experimental campaigns. Master's thesis, Cranfield University, 2015.

[11] *A Comparison of Several CFD Codes with Experimental Data in a Diffusing S Duct*, 2006.

[12] R. Tridello. Comparison of genetic and tabu search alghoritms in aerodynamic design of s-ducts. Master's thesis, Universitá degli Studi di Padova, 2017.

[13] S. Mirjalili, A. Lewis, and S. Mostaghim. Confidence measure: A novel metric for robust meta-heuristic optimisation algorithms. *Information Sciences*, 317: 114–142, 2015.

[14] M. Mattioli. Aerodynamic design sensitivities and uncertainty management. Master's thesis, Cranfield University, 2015.

[15] M. Moro. Robust aerodynamic optimisation of a high-lift configuration wing with polynomial chaos. Master's thesis, Cranfield University, 2013.

[16] G. Trapani. *The Design of High Lift Aircraft Configurations Through Multi-Objective Optimisation*. PhD thesis, Cranfield University, 2014.

[17] R. W. Walters and L. Huyse. Uncertainty analysis for fluid mechanics with applications. Technical report, NASA Langley Research Center, Hampton, VA, USA, 2002.

[18] W. L. Oberkampf, J. C. Helton, and K. Sentz. Mathematical representation of uncertainty. *AIAA Non-deterministic approaches Forum*, Seattle, WA, USA, 2002.

[19] L. Huyse, S. L. Padula, R. M. Lewis, and W. Li. Probabilistic approach to free-form airfoil shape optimization under uncertainty. *AIAA Journal*, 40(9): 1764–1772, 2002.

[20] H. W. Coleman and W. G. Steele. *Experimentation, Validation and Uncertainty Analysis for Engineers*. John Wiley & Sons, Hoboken, NJ, USA, 3rd edition, 2001.

[21] M. Padulo, M. S. Campobasso, and M. D. Guenov. Comparative analysis of uncertainty propagation methods for robust engineering design. *International Conference on Engineering Design*, Paris, France, 2007.

[22] M. Padulo, M. S. Campobasso, and M. D. Guenov. Novel uncertainty propagation method for robust aerodynamic design. *AIAA Journal*, 49(3):530–543, 2011.

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

[24] D. Xiu and G. E. Karniadakis. The wiener-askey polynomial chaos for stochastic differential equations. *SIAM Journal of Scientific Computing*, 24(2):619–644, 2002.

[25] T. Ghisu, G. T. Parks, J. P. Jarrett, and P. J. Clarkson. Robust design optimization of gas turbine compression systems. *Journal of Propulsion and Power*, 27(2):282–295, 2011.

[26] F. Simon, P. Gullien, P. Sagaut, and D. Lucor. A gpc-based approach to uncertain transonic aerodynamics. *Computer Methods to Applied Mechanics and Engineering*, 199:1091–1099, 2010.

[27] L. Bruno, C. Canuto, and D. Fransos. Stochastic aerodynamics and aeroelasticity of a flat plate via generalised polynomial chaos. *Journal of Fluid and Structures*, 25:1158–1176, 2011.

[28] G. E. Karniadakis, C.-H. Su, D. Xiu, D. Lucor, C. Schwab, and R. A. Todor. Generalized polynomial chaos solution for differential equations with random inputs. *Under review in SIAM review for Seminar fur Angewandte Mathematik Eidgenossische Technische Hochschule*, Zurich, Switzerland, 2005.

[29] R. H. Cameron and W. T. Martin. The orthogonal development of non-linear functionals in series of fourier-hermite functionals. *Annals of Mathematics*, 1947.

[30] D. Xiu and G. E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187: 137–167, 2003.

[31] T. Ghisu, G. T. Parks, J. P. Jarrett, and P. J. Clarkson. Adaptive polynomial chaos for gas turbine compression systems performance analysis. *AIAA Journal*, 48(6):1156–1170, 2010.

[32] O. P. Le Maitre, M. T. Reagan, H. N. Najm, R. G. Ghanem, and O. M. Knio. A stochastic projection method for fluid flow. part ii: Random process. *Journal of Computational Physics*, 181:9–44, 2002.

[33] S. Hosder, R. Walters, and R. Perez. A non-intrusive polynomial chaos method for uncertainty propagation in cfd simulations. *Proceedings of the 44^{th} AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 2006.

[34] S. Hosder, R. Walters, and M. Balch. Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. *Proceedings for the 48^{th} AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Honolulu, HI, USA, 2007.

[35] S.-K. Choi, R. V. Grandhi, R. A. Canfield, and C. L. Pettit. Polynomial chaos expansion with latin hypercube sampling for predicting response variability. *44^{th} AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference*, Norfolk, VA, USA, 2003.

[36] A. A. Giunta, S. F. Wojtkiewicz, and M. S. Eldred. Overview of modern design of experiments methods for computational simulations. *41^{st} AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 2003.

[37] F. Xiong, S. Greene, W. Chen, Y. Xiong, and S. Yang. A new sparse grid based method for uncertainty propagation. *Structural and Multidisciplinary Optimisation*, 41(3):335–349, 2010.

[38] M. A. T. Walter. *A stochastic expansion-based approach for design under uncertainty*. PhD thesis, Georgia Institute of Technology, 2013.

[39] N. A. El-Sherbeny. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University*, 22: 123–131, 2010.

[40] J. Branke, K. Deb, K. Miettinen, and R. Slowinski. *Multiobjective Optimization – Interactive and Evolutionary Approaches*, chapter 1, pages 1–26. Springer, Berlin, 2008.

[41] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[42] M. Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, Ma, USA, 1999.

[43] M. Dodson and G. T. Parks. Robust aerodynamic design optimization using polynomial chaos. *Journal of Aircraft*, 46(2):635–646, 2009.

[44] H. Kato, H. Mashiko, Y. Tokuyama, K. Funazaki, and J. Takida. Robust aerodynamic shape optimization of supersonic turbine using non-intrusive polynomial chaos expansion. *9th World Congress on Structural and Multidisciplinary Optimization*, Shizuoka, Japan, 2011.

[45] F. Xiong, B. Xue, Z. Yan, and S. Yang. Polynomial chaos expansione based robust design optimization. *Proceedings of the 2011 International Conference on Quality, Reliability, Risk, Maintenance and Safety Engineering*, Xi'an, China, 2011.

[46] L. Zhao, W. N. Dawes, G. Parks, J. P. Jarrett, and S. Yang. Robust airfoil design with respect to boundary layer transition. *Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Palm Springs, CA, USA, 2009.

[47] S. Mirjalili and A. Lewis. Novel frameworks for creating robust multi-objective benchmark problems. *Information Sciences*, 300:158–192, 2015.

[48] S. Mirjalili, A. Lewis, and S. A. M. Mirjalili. Multi-objective optimisation of marine propellers. *Procedia Computer Science*, 51:2247–2256, 2015.

[49] F. Glover. Tabu search: Part i. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[50] T. Kipouros, D. M. Jaeggi, W. N. Dawes, G. T. Parks, A. M. Savill, and P. J. Clarkson. Biobjective design optimization for axial compressors using tabu search. *AIAA Journal*, 46(3):701–711, 2008.

[51] F. Lorenzi. Design sensitivities for aircraft/engine configurations. Master's thesis, Cranfield University, 2013.

[52] J. A. Exposito Carrillo. Aeroacoustic design optimisation for take-off configuration. Master's thesis, Cranfield University, 2011.

[53] G. Trapani. Multi-objective optimization of 2d high-lift airfoil configurations using tabu search. Master's thesis, Cranfield University, 2009.

[54] R. Hooke and T. A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1960.

[55] N. Chiereghin, L. Guglielmi, M. Savill, E. Manca T. Kipouros, A. Rigobello, and E. Benini M. Barison. Shape optimization of a curved duct with free

form deformations. *23rd AIAA Computational Fluid Dynamics Conference*, 2017.

[56] Athanasios G. Liatsikouras, Varvara G. Asouti, Kyriakos C. Giannakoglou, Guillaume Pierrot, and Mustafa Megahed. Aerodynamic shape optimization under flow uncertainties using non-intrusive polynomial chaos and evolutionary algorithms. *2nd ECCOMAS Thematic Conference on Uncertainty Quanti cation in Computational Sciences and Engineering*, 2007.

Appendix A

NIPC 1D

The NIPC 1D work-flow is presented below. One uncertainty input (Inlet velocity), four objective functions output (CP_{mean}, CP_{stdv}, $Swirl_{mean}$, $Swirl_{stdv}$,). This optimisation problem was constructed with the following parameters:

- Uncertainty in the Inlet Velocity, $n = 1$.
- $\mu_v = 196.53$ m/s, $\sigma_v = 10$ m/s, $v(\xi) = \mu_v + \xi\sigma_v$ where ξ is a Gaussian random variable.
- PC order $p = 3$.
- Type polynomial: Hermite chaos.

According to equation 3.21, $P + 1 = 4$ deterministic evaluations are needed to model the stochastic output. The Inlet velocity was sampled with LHS, giving:

$$v_0, \qquad v_1, \qquad v_2, \qquad v_3 \tag{A.1}$$

The CFD analysis was then performed with these four samples, finding:

$$\begin{gathered} CP(v_0), \qquad CP(v_1), \qquad CP(v_2), \qquad CP(v_3) \\ \alpha(v_0), \qquad \alpha(v_1), \qquad \alpha(v_2), \qquad \alpha(v_3) \end{gathered} \tag{A.2}$$

From the Polynomial chaos theory it was possible to write:

$$CP(\xi_i) = a_0 + a_1\xi_i + a_2(\xi_i^2 - 1) + a_3(\xi_i^3 - 3\xi_i), \qquad i = 0, \ldots, 3 \tag{A.3}$$

and a linear system of equations was then built:

$$\begin{pmatrix} \Psi_0(\xi_0) & \Psi_1(\xi_0) & \Psi_2(\xi_0) & \Psi_3(\xi_0) \\ \Psi_0(\xi_1) & \Psi_1(\xi_1) & \Psi_2(\xi_1) & \Psi_3(\xi_1) \\ \Psi_0(\xi_2) & \Psi_1(\xi_2) & \Psi_2(\xi_2) & \Psi_3(\xi_2) \\ \Psi_0(\xi_3) & \Psi_1(\xi_3) & \Psi_2(\xi_3) & \Psi_3(\xi_3) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} CP\,(v_0) \\ CP\,(v_1) \\ CP\,(v_2) \\ CP\,(v_3) \end{pmatrix} \tag{A.4}$$

Where

$$\begin{aligned} \Psi_0 &= 1 \\ \Psi_1 &= \xi \\ \Psi_2 &= \xi^2 - 1 \\ \Psi_3 &= \xi^3 - 3\xi \end{aligned} \tag{A.5}$$

and where

$$\xi_i = \frac{v_i - \mu_v}{\sigma_v}, \qquad i = 0, \ldots, 3$$

Here of course the vector $\boldsymbol{\xi} = \xi$ because there is only one uncertainty and ξ was sampled 4 times. By solving the linear system the PC coefficients a_i can be found:

$$a_0, \qquad a_1, \qquad a_2, \qquad a_3, \tag{A.6}$$

The same linear system was built for the Swirl (α) and the four PC coefficient b_i was found.
Thanks to equation A.3 and remembering that $\langle \Psi_i^2 \rangle = i!$:

$$\begin{aligned} \mu_{CP} &= a_0 \\ \mu_\alpha &= b_0 \end{aligned} \tag{A.7}$$

$$\begin{aligned} \sigma_{CP} &= \sqrt{\sum_{i=1}^{P} a_i^2 \langle \Psi_i^2 \rangle} \\ \sigma_\alpha &= \sqrt{\sum_{i=1}^{P} b_i^2 \langle \Psi_i^2 \rangle} \end{aligned} \tag{A.8}$$

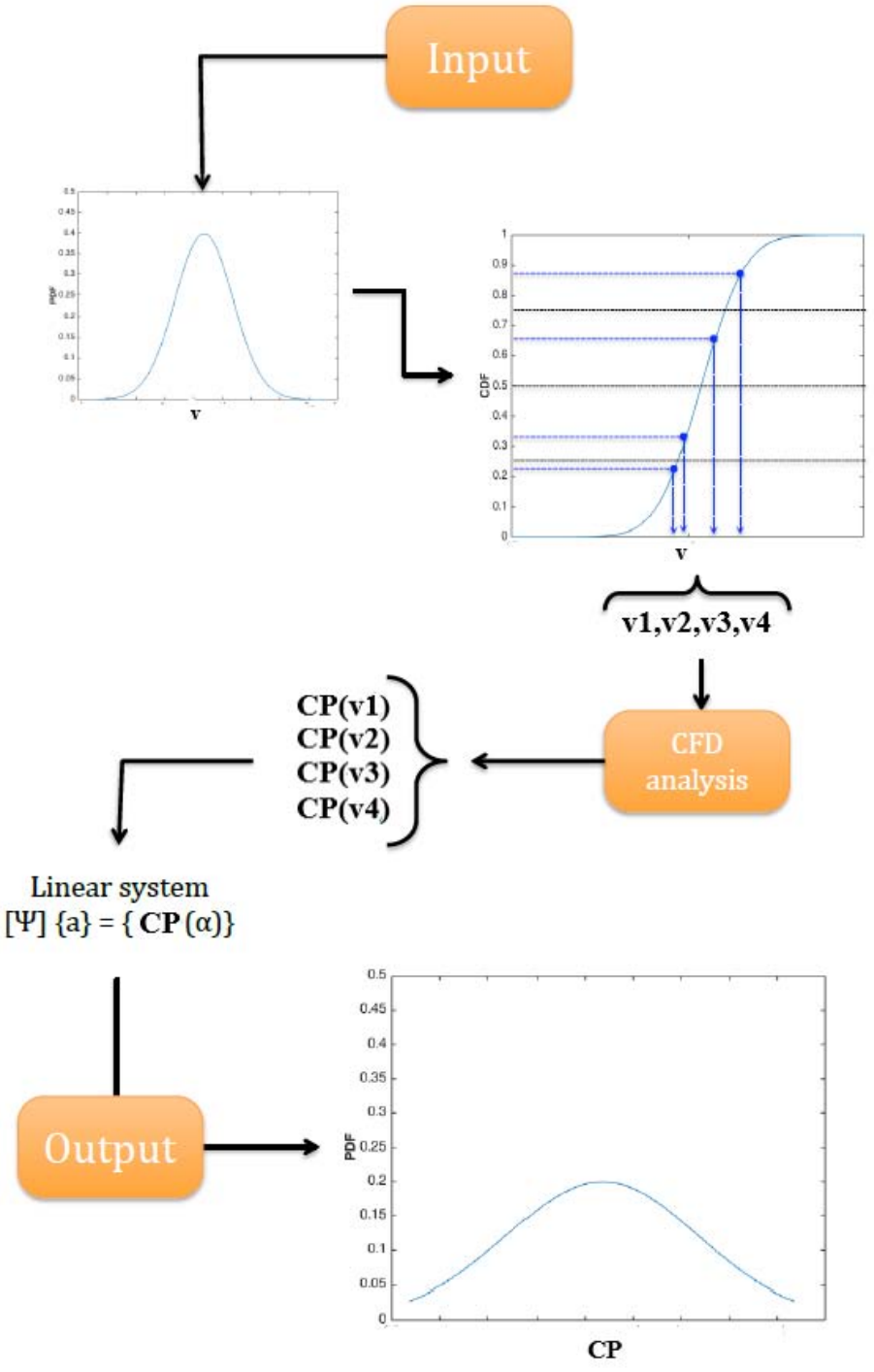


FIGURE A.1: NIPC 1D workflow.

Appendix B

NIPC 2D

The NIPC 2D work-flow is presented below. Two uncertainties input (Inlet Velocity, Inlet Flux Deviation), two objective functions output (CP_{mean}, CP_{stdv}) The following parameters were chosen:

- Uncertainty in the Inlet Velocity and Inlet flux deviation, $n = 2$.
- $\mu_v = 196.53$ m/s, $\sigma_v = 10$ m/s, $v(\xi) = \mu_v + \xi\sigma_v$ where ξ is a Gaussian random variable.
- $\mu_\beta = 0°$, $\sigma_\beta = 3°$, $\beta(\xi) = \mu_\beta + \xi\sigma_\beta$ where ξ is a Gaussian random variable.
- PC order $p = 2$.
- Hermite chaos.

According to equation 3.21, $P + 1 = 6$ deterministic evaluations are needed to model the stochastic output and to find the $P + 1 = 6$ PC coefficients. In this case the standard random vector is a 1×2 vector:

$$\boldsymbol{\xi} = (\xi_v, \xi_\beta) \tag{B.1}$$

And the basis functions are:

$$\begin{aligned}
\Psi_0 &= 1 \\
\Psi_1 &= \xi_v \\
\Psi_2 &= \xi_\beta \\
\Psi_3 &= \xi_v^2 - 1 \\
\Psi_4 &= \xi_v \xi_\beta \\
\Psi_5 &= \xi_\beta^2 - 1
\end{aligned} \tag{B.2}$$

Like in the 1D case, the two uncertain variables were sampled with LHS. To obtain the required 6 samples of $\boldsymbol{\xi}$, 3 samples for both variables were computed. In this case:

$$\begin{aligned}
v_0, \quad \beta_0 \\
v_1, \quad \beta_1 \\
v_2, \quad \beta_2
\end{aligned} \tag{B.3}$$

This leads to a maximum of 9 possible combinations: 6 of them were chosen randomly to form the samples of $\boldsymbol{\xi}$:

$$\begin{aligned}
\boldsymbol{\xi}_0 &= (\xi_{v_0}, \xi_{\beta_0}) \\
\boldsymbol{\xi}_1 &= (\xi_{v_0}, \xi_{\beta_1}) \\
\boldsymbol{\xi}_2 &= (\xi_{v_0}, \xi_{\beta_2}) \\
\boldsymbol{\xi}_3 &= (\xi_{v_1}, \xi_{\beta_1}) \\
\boldsymbol{\xi}_4 &= (\xi_{v_1}, \xi_{\beta_2}) \\
\boldsymbol{\xi}_5 &= (\xi_{v_2}, \xi_{\beta_2})
\end{aligned} \tag{B.4}$$

where

$$\begin{aligned}
\xi_{v_i} &= \frac{v_i - \mu_v}{\sigma_v} \\
\xi_{\beta_i} &= \frac{\beta_i - \mu_\beta}{\sigma_\beta}
\end{aligned} \tag{B.5}$$

Note that, according to Hosder [34], the choice of the combinations is up to the user. He demonstrated that better results are achieved by using more samples than the required $P + 1$ (in this case the linear system is solved in the least squares sense): however, given the limited amount of time available for the optimisation,

only 6 samples were used in this work.
At this point the procedure was similar to the 1D case. The CFD analysis was performed for the 6 samples giving:

$$\begin{aligned} &CP\left(v_0, \beta_0\right) \\ &CP\left(v_0, \beta_1\right) \\ &CP\left(v_0, \beta_2\right) \\ &CP\left(v_1, \beta_1\right) \\ &CP\left(v_1, \beta_2\right) \\ &CP\left(v_2, \beta_2\right) \end{aligned} \tag{B.6}$$

At the end, the linear system of equations was solved and the PC coefficients were found:

$$a_0, \qquad a_1, \qquad a_2, \qquad a_3, \qquad a_4, \qquad a_5 \tag{B.7}$$

Thanks to the following equation:

$$CP(\xi_{v_i}, \xi_{\beta_i}) = a_0 + a_1\xi_{v_i} + a_2\xi_{\beta_i} + a_3\left(\xi_{v_i}^2 - 1\right) + a_4\left(\xi_{v_i}\xi_{\beta_i}\right) + a_5\left(\xi_{\beta_i}^2 - 1\right) \tag{B.8}$$

and the relation $\langle \Psi_i^2 \rangle = i!$, it was possible to find the two objective functions:

$$\begin{aligned} \mu_{CP} &= a_0 \\ \sigma_{CP} &= \sqrt{\sum_{i=1}^{P} a_i^2 \left\langle \Psi_i^2 \right\rangle} \end{aligned} \tag{B.9}$$

However, it should be remarked that the PC expansion has been stopped at the second order here, hence for a more correct comparison with the 1D case the same PC order should be used.

Appendix C

NISP 1D

The NISP 1D work-flow is presented below. One uncertainty input (Inlet velocity), four objective functions output (CP_{mean}, CP_{stdv}, $Swirl_{mean}$, $Swirl_{stdv}$,). This optimisation problem was built with the following parameters:

- Uncertainty in the Inlet Velocity, $n = 1$.
- $\mu_v = 196.53$ m/s, $\sigma_v = 10$ m/s, $v(\xi) = \mu_v + \xi\sigma_v$ where ξ is a Gaussian random variable.
- PC order $p = 3$.
- Type polynomial: Hermite chaos.

According to the NISP theory, $(p+1)^n = 4$ deterministic evaluations and and $m = p + 1 = 4$ samples for each uncertainty were necessary to find the $P + 1 = 4$ PC coefficient and to model the stochastic output.
The sample points and weights were obtained through the Gaussian quadrature method. In Python, the Hermite-Gauss quadrature samples function gives to the user two arrays output. The first one contains the Gaussian random variable samples, instead the second one contains the respectively weights.

$$\begin{aligned} &\xi_0, \quad \omega_0 \\ &\xi_1, \quad \omega_1 \\ &\xi_2, \quad \omega_2 \\ &\xi_3, \quad \omega_3 \end{aligned} \tag{C.1}$$

It is important to underline that the value of ξ_i needs to be corrected by multiplying it by $\sqrt{2}$ because the Hermite-Gauss quadrature samples function, in Python, samples the Gaussian random variable following a different weighting function:

$$W(\boldsymbol{\xi}) = e^{-x^2} \tag{C.2}$$

To return to the original Gaussian weighting function this step was necessary. After that, to determine the PC coefficients for the CP and $Swirl$ it was possible to simplify equation 3.37 as follow:

$$a_i = \frac{1}{\sqrt{\pi}} \sum_{k=1}^{4} CP(v_k) \frac{\Psi_i(\sqrt{2}\xi_k)}{i!} \omega_k \tag{C.3}$$

$$b_i = \frac{1}{\sqrt{\pi}} \sum_{k=1}^{4} \alpha(v_k) \frac{\Psi_i(\sqrt{2}\xi_k)}{i!} \omega_k \tag{C.4}$$

where:

- $CP(v_k)$ and $\alpha(v_k)$ are the results of the CFD analysis for each sample with $v_k = \mu_v + \sqrt{2}\xi\sigma_v$.
- $\Psi_i(\sqrt{2}\xi_k)$ are the Hermite polynomials in terms of $\sqrt{2}\xi_k$
- $1/\sqrt{\pi}$ factor is necessary because of the different weighting function adopted in Python.

With the PC coefficient a_i it was possible to find the final objective functions:

$$\begin{aligned} \mu_{CP} &= a_0 \\ \mu_{\alpha} &= b_0 \end{aligned} \tag{C.5}$$

$$\begin{aligned} \sigma_{CP} &= \sqrt{\sum_{i=1}^{P} a_i^2 \left\langle \Psi_i^2 \right\rangle} \\ \sigma_{\alpha} &= \sqrt{\sum_{i=1}^{P} b_i^2 \left\langle \Psi_i^2 \right\rangle} \end{aligned} \tag{C.6}$$

In figure C.1 there is a schematic representation of the NISP workflow. It was taken from Moro's work [15].

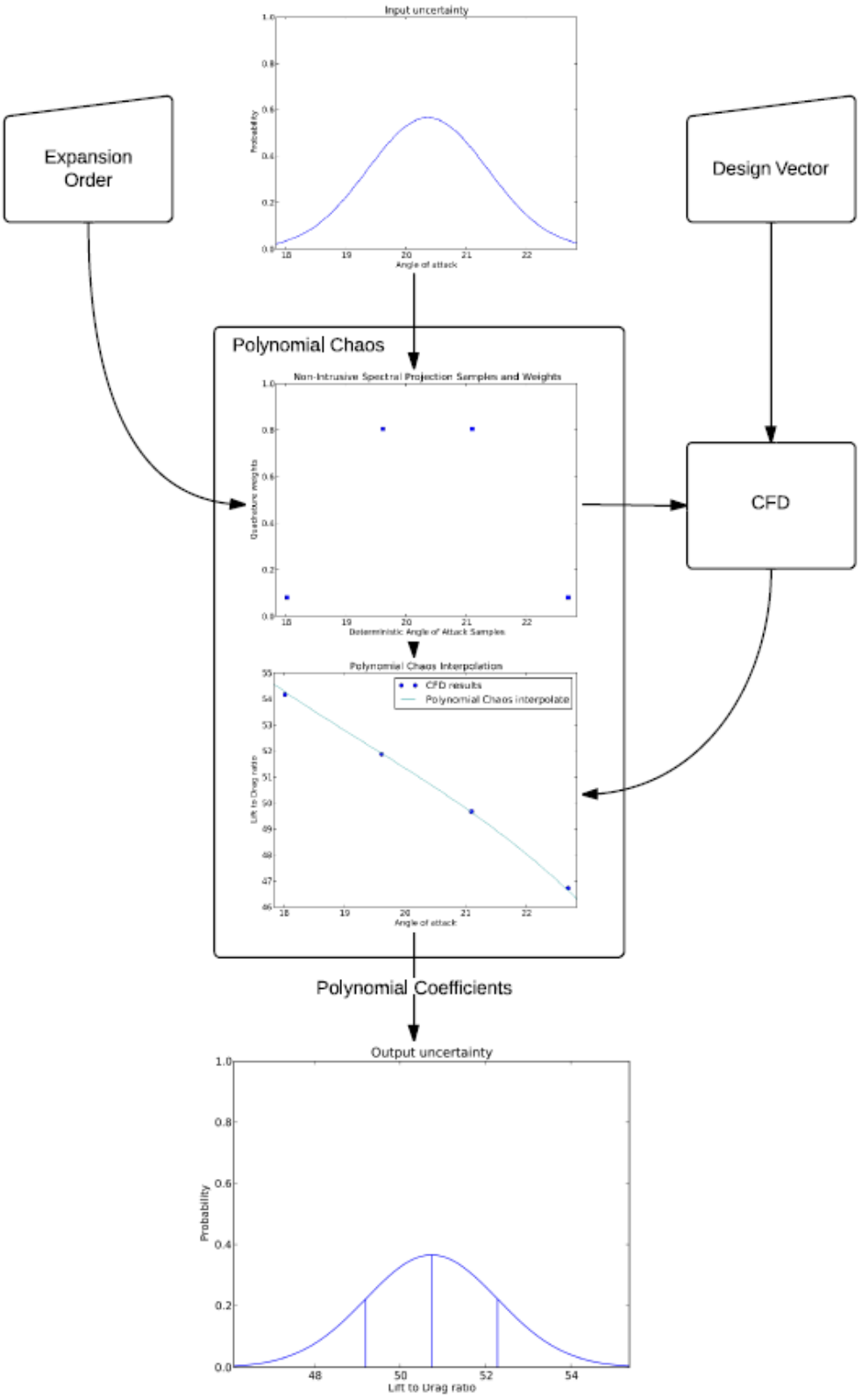


FIGURE C.1: NISP 1D workflow.

Appendix D

NISP 2D

The NISP 2D work-flow is presented below. Two uncertainties input (Inlet Velocity, Inlet Flux Deviation), two objective functions output (CP_{mean}, CP_{stdv})
The following parameters were chosen:

- Uncertainty in the Inlet Velocity an Inlet flux deviation, $n = 2$.
- $\mu_v = 196.53$ m/s, $\sigma_v = 10$ m/s, $v(\xi) = \mu_v + \xi\sigma_v$ where ξ is a Gaussian random variable.
- $\mu_\beta = 0°$, $\sigma_\beta = 3°$, $\beta(\xi) = \mu_\beta + \xi\sigma_\beta$ where ξ is a Gaussian random variable.
- PC order $p = 2$.
- Hermite chaos.

According to the NISP theory, $(p+1)^n = 9$ deterministic evaluations and $m = p + 1 = 3$ samples for each uncertainty were necessary to find the $P + 1 = 6$ PC coefficient and to model the stochastic output.
The sample points and weights of the two uncertainties input were obtained as it was described also in Appendix C for the NISP 1D, and from them it is possible to create nine combinations of (ξ_{v_i}, ξ_{β_i})
It is important to underline that all the values of ξ_{v_i}, ξ_{β_i} need to be corrected by multiplying them by $\sqrt{2}$ because the Hermite-Gauss quadrature samples function, in Python, samples the Gaussian random variable following a different weighting

function like in Appendix C for the NISP 1D. To find the PC coefficients it was used the modified equation 3.37:

$$a_i = \frac{1}{\sqrt{\pi}^n}\left[\sum_{m_1=1}^{m}\cdots\sum_{m_n=1}^{m} CP(\sqrt{2}\xi_{m_1},\ldots,\sqrt{2}\xi_{m_n})\frac{\Psi_i(\sqrt{2}\xi_{m_1},\ldots,\sqrt{2}\xi_{m_n})}{\langle\Psi_i^2\rangle}\prod_{k=1}^{n}\omega_{m_k}\right]$$

To help the reader, the above equation can be written for the a_1 coefficient as follow:

$$a_1 = \frac{1}{\sqrt{\pi}^2}\frac{1}{1!}[H]$$

where:

$$\begin{aligned}
H = &+CP(\sqrt{2}\xi_{v_0},\sqrt{2}\xi_{\beta_0})\Psi_1(\sqrt{2}\xi_{v_0},\sqrt{2}\xi_{\beta_0})\omega_{v_0}\omega_{\beta_0}+\\
&+CP(\sqrt{2}\xi_{v_0},\sqrt{2}\xi_{\beta_1})\Psi_1(\sqrt{2}\xi_{v_0},\sqrt{2}\xi_{\beta_1})\omega_{v_0}\omega_{\beta_1}+\\
&+CP(\sqrt{2}\xi_{v_0},\sqrt{2}\xi_{\beta_2})\Psi_1(\sqrt{2}\xi_{v_0},\sqrt{2}\xi_{\beta_2})\omega_{v_0}\omega_{\beta_2}+\\
&+CP(\sqrt{2}\xi_{v_1},\sqrt{2}\xi_{\beta_0})\Psi_1(\sqrt{2}\xi_{v_1},\sqrt{2}\xi_{\beta_0})\omega_{v_1}\omega_{\beta_0}+\\
&+CP(\sqrt{2}\xi_{v_1},\sqrt{2}\xi_{\beta_1})\Psi_1(\sqrt{2}\xi_{v_1},\sqrt{2}\xi_{\beta_1})\omega_{v_1}\omega_{\beta_1}+\\
&+CP(\sqrt{2}\xi_{v_1},\sqrt{2}\xi_{\beta_2})\Psi_1(\sqrt{2}\xi_{v_1},\sqrt{2}\xi_{\beta_2})\omega_{v_1}\omega_{\beta_2}+\\
&+CP(\sqrt{2}\xi_{v_2},\sqrt{2}\xi_{\beta_0})\Psi_1(\sqrt{2}\xi_{v_2},\sqrt{2}\xi_{\beta_0})\omega_{v_2}\omega_{\beta_0}+\\
&+CP(\sqrt{2}\xi_{v_2},\sqrt{2}\xi_{\beta_1})\Psi_1(\sqrt{2}\xi_{v_2},\sqrt{2}\xi_{\beta_1})\omega_{v_2}\omega_{\beta_1}+\\
&+CP(\sqrt{2}\xi_{v_2},\sqrt{2}\xi_{\beta_2})\Psi_1(\sqrt{2}\xi_{v_2},\sqrt{2}\xi_{\beta_2})\omega_{v_2}\omega_{\beta_2}
\end{aligned} \tag{D.1}$$

With the PC coefficient a_i it was possible to find the final objective functions:

$$\begin{aligned}
\mu_{CP} &= a_0\\
\sigma_{CP} &= \sqrt{\sum_{i=1}^{P} a_i^2\langle\Psi_i^2\rangle}
\end{aligned} \tag{D.2}$$

Appendix E

NIPC 2D Manufacturing

The NIPC 2D Manufacturing work-flow is presented below. Two uncertainties input (Inlet Velocity, Bending Angle), two objective functions output (CP_{mean}, CP_{stdv}) The following parameters were chosen:

- Uncertainty in the Inlet Velocity and Bending Angle, $n = 2$.
- $\mu_v = 196.53$ m/s, $\sigma_v = 10$ m/s, $v(\xi) = \mu_v + \xi\sigma_v$ where ξ is a Gaussian random variable.
- $\mu_\alpha = 60°$, $\sigma_\alpha = 4°$, $\alpha(\xi) = \mu_\alpha + \xi\sigma_\alpha$ where ξ is a Gaussian random variable.
- PC order $p = 2$.
- Hermite chaos.

According to equation 3.21, $P + 1 = 6$ deterministic evaluations are needed to model the stochastic output and to find the $P + 1 = 6$ PC coefficient. In this case the standard random vector is a 1×2 vector:

$$\boldsymbol{\xi} = (\xi_v, \xi_\alpha) \tag{E.1}$$

And the basis functions are:

$$\begin{aligned}
\Psi_0 &= 1 \\
\Psi_1 &= \xi_v \\
\Psi_2 &= \xi_\beta \\
\Psi_3 &= \xi_v^2 - 1 \\
\Psi_4 &= \xi_v \xi_\beta \\
\Psi_5 &= \xi_\beta^2 - 1
\end{aligned} \tag{E.2}$$

The two uncertain variables were sampled with LHS. To obtain the required 6 samples of $\boldsymbol{\xi}$, 3 samples for both variables were computed. In this case:

$$\begin{aligned}
v_0, \quad \alpha_0 \\
v_1, \quad \alpha_1 \\
v_2, \quad \alpha_2
\end{aligned} \tag{E.3}$$

This leads to a maximum of 9 possible combinations: 6 of them were chosen randomly to form the samples of $\boldsymbol{\xi}$:

$$\begin{aligned}
\boldsymbol{\xi}_0 &= (\xi_{v_0}, \xi_{\alpha_0}) \\
\boldsymbol{\xi}_1 &= (\xi_{v_0}, \xi_{\alpha_1}) \\
\boldsymbol{\xi}_2 &= (\xi_{v_0}, \xi_{\alpha_2}) \\
\boldsymbol{\xi}_3 &= (\xi_{v_1}, \xi_{\alpha_1}) \\
\boldsymbol{\xi}_4 &= (\xi_{v_1}, \xi_{\alpha_2}) \\
\boldsymbol{\xi}_5 &= (\xi_{v_2}, \xi_{\alpha_2})
\end{aligned} \tag{E.4}$$

where

$$\begin{aligned}
\xi_{v_i} &= \frac{v_i - \mu_v}{\sigma_v} \\
\xi_{\alpha_i} &= \frac{\alpha_i - \mu_\alpha}{\sigma_\alpha}
\end{aligned} \tag{E.5}$$

Note that, according to Hosder [34], the choice of the combinations is up to the user. He demonstrated that better results are achieved by using more samples than the required $P + 1$ (in this case the linear system is solved in the least squares sense): however, given the limited amount of time available for the optimisation, only 6 samples were used in this work.

At this point the procedure was similar to the others NISP 1D and 2D cases. The CFD analysis was performed for the 6 samples giving:

$$\begin{aligned} &CP\left(v_0, \alpha_0\right) \\ &CP\left(v_0, \alpha_1\right) \\ &CP\left(v_0, \alpha_2\right) \\ &CP\left(v_1, \alpha_1\right) \\ &CP\left(v_1, \alpha_2\right) \\ &CP\left(v_2, \alpha_2\right) \end{aligned} \tag{E.6}$$

At the end, the linear system of equations was solved and the PC coefficients were found:

$$a_0, \qquad a_1, \qquad a_2, \qquad a_3, \qquad a_4, \qquad a_5 \tag{E.7}$$

Thanks to the following equation:

$$CP(\xi_{v_i}, \xi_{\alpha_i}) = a_0 + a_1\xi_{v_i} + a_2\xi_{\alpha_i} + a_3\left(\xi_{v_i}^2 - 1\right) + a_4\left(\xi_{v_i}\xi_{\alpha_i}\right) + a_5\left(\xi_{\alpha_i}^2 - 1\right) \tag{E.8}$$

and the relation $\langle \Psi_i^2 \rangle = i!$ was possible to find the two objective functions:

$$\begin{aligned} \mu_{CP} &= a_0 \\ \sigma_{CP} &= \sqrt{\sum_{i=1}^{P} a_i^2 \left\langle \Psi_i^2 \right\rangle} \end{aligned} \tag{E.9}$$