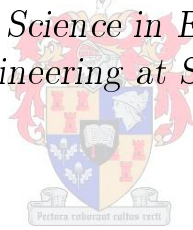


The Analysis and Optimization of an Axial Compressor

by

Richard Alan Hamman

*Thesis presented in partial fulfilment of the requirements for
the degree of Master of Science in Engineering (Mechanical)
in the Faculty of Engineering at Stellenbosch University*



Supervisors:

Prof. Gerhard Venter Dr. Johan van der Spuy

December 2015

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date:

Abstract

The Analysis and Optimization of an Axial Compressor

R. A. Hamman

*Department of Mechanical and Mechatronics Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.*

Thesis: MScEng (Mech)

December 2015

Axial compressors are widely used in a variety of contexts. When compared to centrifugal compressors, axial compressors offer higher pressure ratios and efficiencies. Due to the complex relationship between blade shape and compressor performance, as well as the sensitivity of axial compressors to massflow rates, the design of an axial compressor is a challenging problem. These design difficulties can be circumvented using numerical design optimization.

In this work, meta-model based design optimization (MBDO), a variant of numerical design optimization, is used to develop replacement blades for a low speed axial compressor. Two numerical models were developed: a low fidelity, computationally inexpensive single stage model and a high fidelity, computationally expensive three stage model. For reasons of computational cost, the single stage model was used to evaluate the objective function in the optimization process.

An optimized blade design was developed, which delivered a 9.83% increase in pressure coefficient over the original design, when evaluated with the single stage model. This increase was not sustained when evaluating the optimized design with the three stage model. After investigation, the cause was found to be a high incidence angle near the hub and shroud, just outside the range in which incidence angle was constrained in the design problem. To compensate for this, the blade angles were manually adjusted, in order to lower the incidence angle. The adjusted blades delivered an increase of 8.54% over the original design, when evaluated with the three stage model.

Uittreksel

Die Analise en Optimering van 'n Aksiale kompressor

R. A. Hamman

*Departement Meganiese en Megatroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.*

Tesis: MScIng (Meg)

Desember 2015

Aksiale kompressors word benut in 'n wye verskeidenheid omgewings. In vergelyking met sentrifugale kompressors bied aksiale kompressors hoër werksverrigting en doeltreffendheid. Die komplekse verhouding tussen lemvorm en kompressorwerkverrigting sowel as die sensitiwiteit van aksiale kompressors vir massavloeiempo maak die ontwerp van 'n aksiale kompressor 'n uitdagende taak. Hierdie ontwerpsuitdagings kan omseil word deur van numeriese ontwerpsoptimering gebruik te maak. In hierdie werk word meta-model gebaseerde ontwerpsoptimering (MBDO), 'n variasie van numeriese ontwerpsoptimering, gebruik om nuwe lemme vir 'n lae-spoed aksiale kompressor te ontwerp. Twee numeriese modelle was ontwerp: 'n lae-akkuraatheid, lae-berekeningskoste enkel stadium model, en 'n hoë-akkuraatheid, hoë-berekeningskoste drie-stadium model. As 'n gevolg van berekeningskoste is die enkel stadium model gebruik om die doel funksie in die optimeringsproses te evalueer.

'n Optimale lem ontwerp is ontwikkel wat 'n verhoging van 9.83% in die druk-koëffisiënt toon teenoor die oorspronklike ontwerp indien dit met die enkel stadium model ge-evalueer word. Hierdie verbetering is nie sigbaar met die evaluasie van die drie-stadium model nie. Die oorsaak hiervoor was 'n hoë invalshoek naby die middelpunt en deelspanmantel wat net buite die omvang van die invalshoek se beperkinge in die ontwerpsoptimering was. As kompensasie hiervoor, is die lemhoeke aangepas om die invalshoek te verminder. Die aangepaste lemme toon 'n verhoging van 8.54% teenoor die oorspronklike ontwerp asdit met die drie-stadium model ge-evalueer word.

Acknowledgements

I would like to thank Professor Gerhard Venter and Doctor Johan van der Spuy for all their help and guidance during this long journey. I'm forever grateful for your monk-like patience. I would also like to thank the National Research Foundation (NRF) and Project BALLAST for their support in this work.

Thanks to Andrew Gill for his valuable input into the modeling of axial compressors and his helpfulness in supplying experimental data. I'd like to thank Martin Venter and Francois Wessels for all the office discussions, coffee and camaraderie.

Lastly and most importantly, I would like to thank my mother, Lyn Hamman, and my fiancé, Suretha Koegelenberg, for their love and support during this long journey. The most valuable thing I have learned over the last four years is what is truly important to me.

Contents

Declaration	i
Abstract	ii
Uittreksel	iii
Acknowledgements	iv
Contents	v
List of Figures	viii
List of Tables	xi
1 Introduction and background	1
1.1 Introduction	1
1.2 Background	1
1.3 Past work at Stellenbosch University	3
1.4 The objectives of this work	4
1.5 Navigating this document	5
2 Analytical and experimental analysis of the Rofanco compressor	6
2.1 Introduction	6
2.2 The fundamentals of axial compressors	6
2.3 The Rofanco compressor	12
2.4 Conclusion	20
3 Numerical analysis of the Rofanco compressor	21
3.1 Introduction	21
3.2 Overview of the simulation process	21
3.3 Model domain and boundary conditions	23
3.4 The meshing process	23
3.5 Some specific modeling choices	27
3.6 Estimating the stall margin	29

<i>CONTENTS</i>	vi
3.7 Comparing numerically predicted and experimentally measured pressure coefficients	30
3.8 Comparing numerically predicted and experimentally measured velocity profiles	31
3.9 Comparing model run-times	36
3.10 Conclusion	36
4 Optimization and meta-modeling	38
4.1 Introduction	38
4.2 Background to optimization and meta-models	39
4.3 Distributing samples in the design space	41
4.4 Evaluating meta-model error	42
4.5 Data transforms	44
4.6 Common meta-modeling techniques	46
4.7 Efficient global optimization	52
4.8 Conclusion	56
5 The optimization of blade profiles	57
5.1 Introduction	57
5.2 Formulation of the optimization problem	57
5.3 Solving the optimization problem	61
5.4 Results of the optimization problem	63
5.5 Conclusion	65
6 Analysis of the optimized blade design	66
6.1 Introduction	66
6.2 Examining the optimized blade design	66
6.3 Comparing the original and optimized blade designs	68
6.4 Adjusting the optimized blade design to lower the incidence angle	73
6.5 Conclusions	77
7 Conclusion	78
7.1 Introduction	78
7.2 The modeling of axial compressors for design purposes	78
7.3 Meta-modeling and efficient global optimization	80
7.4 The compressor blade design problem formulation	82
7.5 Engineering software and computation	84
7.6 The new compressor blades	85
7.7 Evaluating the objectives	85
7.8 Conclusion	86
Appendices	87
A Analytic calculations relating to the Rofanco compressor	88
A.1 Introduction	88

A.2	Solving the velocity triangles	88
A.3	Calculating flow quantities of interest	89
B	Introduction to optimization and meta-modeling	91
B.1	Introduction	91
B.2	R^2 in the context of response surfaces	91
B.3	Kriging correlation and regression models	92
B.4	Some efficient global optimization infill criteria	93
B.5	Optimization algorithms used in this work	94
C	An efficient global optimization example problem	97
C.1	Introduction	97
C.2	Problem description	97
C.3	The optimization process	98
C.4	Results	101
D	Optimization problem detailed results	104
	List of References	124

List of Figures

1.1	Section view of an axial compressor	2
2.1	Velocity triangles in an axial compressor	8
2.2	Blade angles and definitions	9
2.3	The inception of rotating stall in axial compressors	12
2.4	Rofanco pressure coefficient as a function of β_2	16
2.5	Rofanco compressor velocity ratios used to evaluate the de Haller stall criterion	16
2.6	Rofanco compressor static-to-static pressure coefficient as a function of radius	17
2.7	The relationship between pressure rise over a single stage and multiple identical stages	20
3.1	The compressor simulation process	22
3.2	The ATM Optimized mesh topology	24
3.3	Convergence of pressure rise as a function of mesh parameters	26
3.4	Rofanco compressor rotor blade mesh and boundary layer at various locations	27
	(a) The rotor blade	27
	(b) Close-up at mid-chord	27
	(c) Close-up near blade tip	27
3.5	Pressure on upstream and downstream interfaces when applying the mixing plane model	29
3.6	Determining compressor stall from three static simulations	30
	(a) The stalled case	30
	(b) The unstalled case	30
3.7	Numerical and experimental pressure coefficients as a function of mass flow	31
3.8	Axial velocity data for the Rofanco compressor	33
	(a) R1 axial velocity measurements	33
	(b) R2 axial velocity measurements	33
	(c) R3 axial velocity measurements	33
3.9	Tangential velocity data for the Rofanco compressor	35
	(a) R1 tangential velocity measurements	35

(b)	R2 tangential velocity measurements	35
(c)	R3 tangential velocity measurements	35
4.1	A poor and optimal latin hypercube	42
(a)	A poor latin hypercube	42
(b)	A optimal latin hypercube	42
4.2	A visual explanation of principal component analysis	45
4.3	The stepwise forward regression process	47
4.4	The soft margin loss function used in support vector machines	50
4.5	Minimum and maximum kernel values of significance in SVM	51
(a)	Minimum kernel radius of significance	51
(b)	Maximum kernel radius of significance	51
4.6	The infill optimization process	56
5.1	The construction of a compressor blade from radial stations.	59
5.2	Expected improvement and best design of the compressor design problem	64
(a)	Expected improvement as a function of design iterations	64
(b)	Best pressure coefficient as a function of design iterations	64
5.3	Categorization of infill samples over the course of optimization.	65
6.1	The original and optimized compressor blades	67
(a)	Optimum rotor at station 1	67
(b)	Original rotor at station 1	67
(c)	Optimum rotor at station 2	67
(d)	Original rotor at station 2	67
(e)	Optimum rotor at station 3	67
(f)	Original rotor at station 3	67
(g)	Optimum stator at station 1	67
(h)	Original stator at station 1	67
(i)	Optimum stator at station 2	67
(j)	Original stator at station 2	67
(k)	Optimum stator at station 3	67
(l)	Original stator at station 3	67
6.2	Optimized mass flow-pressure coefficient curves	69
(a)	Single stage model	69
(b)	Full model	69
6.3	Axial velocity data for the original and optimized blade set	71
(a)	R1 axial velocity measurements	71
(b)	R2 axial velocity measurements	71
(c)	R3 axial velocity measurements	71
6.4	Tangential velocity data for the original and optimized blade set	72
(a)	R1 tangential velocity measurements	72
(b)	R2 tangential velocity measurements	72

(c)	R3 tangential velocity measurements	72
6.5	Blade angles and incidence for the optimized blade design	73
(a)	Flow and blade angles	73
(b)	Incidence upon the next rotor row	73
6.6	Pressure coefficient as a function of mass flow for the optimized blades	76
(a)	Single stage model	76
(b)	Full model	76
C.1	The meta-modeling test problem	98
(a)	The six-hump camelback objective function	98
(b)	The classification constraint function (g_1).	98
(c)	The regression constraint function (g_2).	98
C.2	An example report for the objective function during the first EGO iteration.	100
C.3	The kriging approximation to the example problem objective function	101
C.4	Expected improvement contour plot for the example problem	102
C.5	Expected and actual cross-validation error for the meta-model variants	102
C.6	The best sample found and expected improvement, as a function of the optimization iterations completed	103
(a)	The best solution as a function of iterations	103
(b)	Expected improvement as a function of iterations	103

List of Tables

1.1	Historical performance of axial compressors	3
2.1	Work done factor per compressor stage	11
2.2	Stagger, camber and solidity of the Rofanco compressor	13
2.3	Rotor and stator tip gap	13
2.4	Rofanco rotor outlet blade angles	14
2.5	Dimensionless performance parameters for the Rofanco compressor	14
3.1	Rofanco compressor mesh statistics	26
3.2	Average and maximum y^+ values for the Rofanco compressor	27
3.3	Area-averaged numerical and experimental axial velocity	34
3.4	Computational cost of the various compressor models	36
4.1	Data transform recipes	55
4.2	Meta-model recipes	55
5.1	The choice of models used in the optimization process.	62
5.2	The number and variety of infill samples	62
5.3	Objective function meta-model error	65
6.1	The current design, optimum design and upper and lower bounds.	68
6.2	Stage by stage pressure rise for the original and optimized compressor geometries	70
6.3	The current design, optimum design, adjusted design and upper and lower bounds.	75
6.4	The various designs discussed in this work and their relative improvements over the original design	75
B.1	Kriging regression models	93
C.1	The number and variety of infill samples used in the example problem	99
C.2	The choice of models used in the example problem	99
D.1	The sampled design points	105
D.2	The objective and constraint function values for each design point	114

Chapter 1

Introduction and background

1.1 Introduction

As performance requirements upon engineering systems become more stringent, so the need to develop more resource and energy efficient solutions arises. This drive for increased performance necessitates the use of engineering models of greater accuracy and increasingly sophisticated approaches to design. One such approach is numerical design optimization, where advanced numerical optimization algorithms are applied to engineering design problems in order to generate optimal or near-optimal solutions. This technique can be applied to problems of arbitrary complexity, provided the performance of a particular design can be evaluated in a timely manner.

In this work, numerical design optimization techniques are applied to the design of low speed axial compressor blades. The relationship between blade shape and compressor performance is a complex one, making the designing of axial compressor blades a challenging undertaking. The problem is further complicated by the sensitivity of axial compressor performance to mass flow rate. Due to these factors, as well as their wide use, the design of axial compressor blades is an ideal problem with which to investigate numerical design optimization techniques.

This chapter lays the basis for the analysis and optimization work of the next five chapters. Some background to axial compressors and numerical design optimization is presented in Section 1.2 and past work is introduced in Section 1.3. Finally, the objectives of this work are outlined in Section 1.4.

1.2 Background

Designing an axial compressor using numerical design optimization requires familiarity with both the analysis of axial compressors and the techniques of numerical design optimization. In the remainder of this section, brief outlines of these topics are presented. Further information on the experimental and

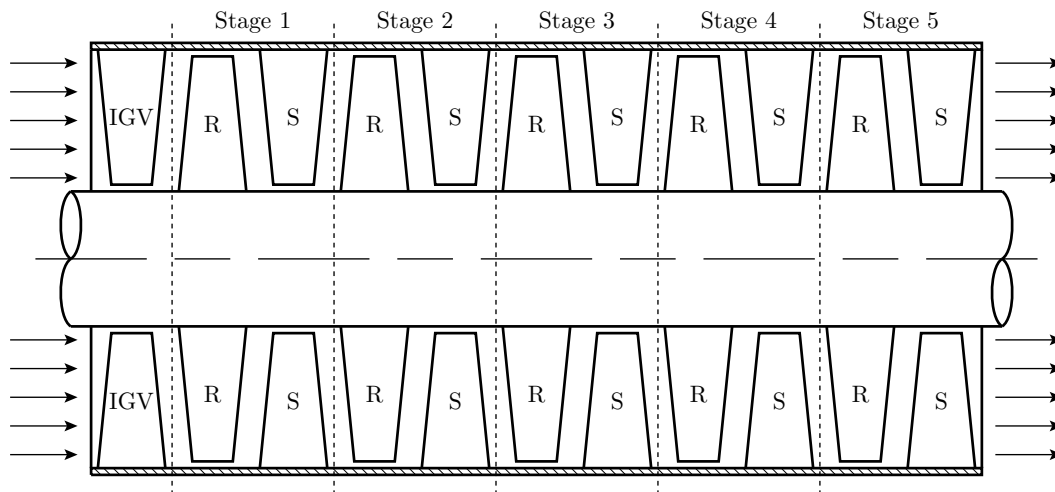


Figure 1.1: An section view of an axial compressor. Flow enters the inlet, and is guided onto the first rotor by an optional set of inlet guide vanes (IGVs). The rotating rotor blades accelerate the flow, while the stationary stator blades diffuse the flow to achieve a rise in static pressure over the blade row. This process is repeated over subsequent pairs of rotor (R) and stator (S) blades, referred to as stages.

analytic analysis of axial compressors is given in Chapter 2, while the focus of Chapter 3 is in developing a numerical axial compressor model, for use in the design process. The necessary optimization techniques are explored in Chapter 4.

1.2.1 Axial compressors

Axial compressors consist of alternating pairs of rotating and stationary blade rows, referred to as rotors and stators respectively, as shown in Figure 1.1. Flow enters and exits in an axial direction, parallel to the axis of rotation of the rotors. The rotor rows accelerate the flow while the stator rows diffuse the flow, increasing the static pressure. Additional rotor and stator pairs, referred to as stages, are combined to achieve the desired increase in pressure. It is possible to fit an additional pair of stationary blades, referred to as inlet guide vanes, to the inlet of the compressor in order to guide the incoming flow onto the first rotor row.

Axial compressors are integral components of gas turbine engines, which are used in jet aircraft, power stations and ships. They are commonly used in wind tunnels and in the process industry, where high volume flow rates are required. The idea of using a reversed axial turbine as a compressor has existed since at least 1884 (Meher-Homji (2000)). Unsurprisingly, these early axial compressors did not deliver stellar performance, due to aerodynamic stall of their blades (Dixon (2005)). It was not until the pioneering work of Griffith (1926) that the science of axial compressors had advanced sufficiently so as to provide a rational basis for the design of axial compressors.

Table 1.1: A historical view of the performance of some representative General Electric axial compressors. Data taken from Cumpsty (1989).

Year	Compressor model	Pressure ratio	Stages
late 50s	CJ805/J79	12.5	17
1969	CF6-50	13.0	14
1974	CFM56	12.0	9
1982	E^3 engine	23.0	10

As the data in Table 1.1 shows, the performance of typical axial compressors continued to advance over the course of the century. In this work, the massive increase in computational power afforded by Moore's Law is exploited to continue this trend, using the techniques of numerical design optimization. A new set of blades is developed for a Rofanco three stage low speed axial compressor used by the Department of Mechanical and Mechatronic Engineering at Stellenbosch University. This machine, manufactured by Royston Fan Co. Ltd. in 1974, is hereafter referred to as the Rofanco compressor. Further technical details of the Rofanco compressor are given in Section 2.3.

1.2.2 Numerical design optimization

Numerical design optimization is a technique involving the application of advanced numerical optimization techniques to solve design problems in the engineering field. Design problems are stated as an objective function to be maximized or minimized, which is acted on by a numerical optimizer. In the usual case, the function itself is unknown, but can be sampled by means of a numerical simulation. Any restrictions on the design are stated as equality or inequality constraints that cannot be violated by a candidate solution.

Numerical design optimization allows the substitution of computational power for creativity, by replacing the step of synthesizing a new design with the output of a numerical algorithm. In this work, a form of numerical design optimization, known as meta-model based design optimization (MBDO), is used. Instead of optimizing the objective function directly, an approximation or "meta-model" of the objective function is optimized. MBDO is discussed in more detail in Chapter 4.

1.3 Past work at Stellenbosch University

The Rofanco compressor has been thoroughly studied by previous students at Stellenbosch University. Since these past works provide valuable numerical and experimental data pertaining to the Rofanco compressor, it is worth briefly reviewing them.

The Rofanco compressor's original plastic blades were destroyed in testing in the 1980s. Benadé (1987) developed a computer program that was used to design a new, high reaction blade set, which did not require inlet guide vanes. A set of replacement blades were manufactured by the Atomic Energy Corporation and Lewis (1989) began the process of recommissioning the machine by installing the first blade row. The recommissioning was completed by Roos (1990), who had to make modifications to several blades, due to manufacturing errors. Roos (1990) also performed experimental analysis of the new blades.

In his Master's thesis, Gill (2006) evaluated several stall criteria across a group of test compressors, one of which was the Rofanco. This work will be relevant in Chapter 2. In his PhD thesis, Gill (2012) performed detailed numerical and experimental analyses of the Rofanco compressor in all four quadrants of operation. This work provides useful data for evaluating the numerical models developed in Chapter 3.

Raubenheimer (2011) used the Rofanco compressor as a test bed for a forced vibration excitation system, designed to enable investigations into fatigue in compressor blades caused by flutter. Brandsen (2013) continued the work of Gill and Raubenheimer and focused on developing a model of the Rofanco compressor that incorporated fluid structure interaction (FSI) effects. He was able to accurately predict the aerodynamic forces on the blades as well as their motions, using a one-way FSI model.

Relevant past optimization works performed at Stellenbosch University include those by Wise (2008) and Wessels *et al.* (2012). Wise (2008) built and optimized a support vector regression meta-model to approximate the performance of a low speed wind turbine calculated using the XFOIL analysis software. Wessels *et al.* (2012) continued this work and proposed several new airfoil parameterization schemes based on non-uniform rational B-splines (NURBS). Although different application areas, these works share many basic technologies with the current work.

1.4 The objectives of this work

The objectives of this work are listed below. Note that they are numbered for reference, not as an indication of priority.

1. The exploration of the techniques and processes involved in the application of shape optimization to airfoils.
2. Investigation into several different meta-modeling techniques, their training procedures, strengths and weaknesses.
3. The application of meta-model based design optimization to a practical design problem.
4. The design of a new set of blades for the Rofanco compressor.

1.5 Navigating this document

This document consists of seven chapters. Chapter 2 to Chapter 4 deal with the fundamental technologies required to apply numerical design optimization to the Rofanco compressor. Chapter 2 focuses on the basics of axial compressors, with an emphasis on an analytic and experimental analysis of the Rofanco compressor as well as determining the scope of potential improvement via numerical design optimization. Chapter 3 deals with the process of developing a numerical model of the Rofanco compressor for use in the design process. Chapter 4 is an introduction to meta-modeling techniques and the optimization code developed as part of this work. Chapter 5 applies the technologies introduced in the preceding three chapters to the optimization of the Rofanco compressor, while Chapter 6 analyses the optimized blade set proposed in Chapter 5. Finally, Chapter 7 summarizes and concludes the work. Note that there is no explicit literature study chapter; instead relevant literature is discussed in the body of the work.

Chapter 2

Analytical and experimental analysis of the Rofanco compressor

2.1 Introduction

The purpose of this Chapter is twofold: to introduce the basics of axial flow compressors and to describe the Rofanco compressor used as a test case in this work. Included in Section 2.2, which describes the fundamentals of axial compressors, is a discussion of blade-to-blade flow in axial compressors, dimensionless flow parameters, losses and their sources, stall and surge as well as structural loading. Section 2.3 deals with the Rofanco compressor, a low speed axial compressor installed at the Department of Mechanical and Mechatronic Engineering at Stellenbosch University. This section includes a description of the machine in question, a brief overview of some of the sources of experimental data used in this work and a semi-analytic analysis of the machine's current configuration. The possibility of improvement through numerical design optimization as well as the relationship between the pressure rise over identical compressor stages are also discussed.

2.2 The fundamentals of axial compressors

2.2.1 Comparing axial and centrifugal compressors

As discussed in Section 1.2.1, axial compressors consist of alternating pairs of rotating and stationary blade rows, referred to as rotors and stators respectively. When compared to centrifugal compressors, multi-stage axial compressors offer greater pressure rises and higher efficiencies, but at the expense of delivering performance that is more sensitive to mass flow rate. This is explained with reference to the conservation of rothalpy (Cumpsty (1989)). In the equation below, $h_2 - h_1$ is the change in enthalpy over a blade row, U is the rotational velocity of the blade and W is the velocity of the fluid relative

to the blade (see Figure 2.1):

$$h_2 - h_1 = \frac{1}{2}(U_2^2 - U_1^2) - \frac{1}{2}(W_2^2 - W_1^2) \quad (2.2.1)$$

In an axial compressor, $U_1 \approx U_2$, and most of the increase in enthalpy comes from $\frac{1}{2}(W_2^2 - W_1^2)$. In contrast, radial compressors cause a change in both U and W . This is significant, since according to Cumpsty (1989), the change in enthalpy attributed to $\frac{1}{2}(U_2^2 - U_1^2)$ is essentially loss free and subject to increase without aerodynamic limit, while the change in enthalpy attributed to $\frac{1}{2}(W_2^2 - W_1^2)$ is not. This topic is further discussed in Section 2.2.5.

2.2.2 Blade-to-blade flow in axial compressors

Consider the cross section of a compressor stage at constant radius, shown in Figure 2.1. The rotor row moves to the right with velocity U . C_1 is the absolute velocity of the flow coming into the rotor row, at angle α_1 to the axial direction. W_1 is the velocity of the incoming flow relative to the moving rotor, which in turn is at angle β_1 to the axial direction. The naming convention for the velocities into the stator are similar to the rotor. W_2 is the velocity relative to the rotor row, while C_2 is the absolute velocity into the rotor row.

The blade-to-blade perspective lends itself to experimental analysis by linear cascade, shown in Figure 2.2. Several important blade properties are also illustrated in this figure. The chord length of the blade (c) is defined as the distance between the leading edge and trailing edge of the blade. The angle between the chord and the vertical plane (ζ) is referred to as the stagger angle. χ_1 and χ_2 are the rotor inlet and outlet angles, while κ_1 and κ_2 are the stator inlet and outlet angles. The flow does not exactly follow the blade, instead the inlet flow angle (β_1 or α_1) is the sum of the blade inlet angle (χ_1 or κ_1) and some incidence, i , while the outlet flow angle (β_2 or α_2) is the sum of the blade outlet angle (χ_2 or κ_2) and some deviation, δ . The camber angle of the blade (θ) is a measure of the blade's curvature. It may be shown geometrically that $\theta = \chi_1 - \chi_2$. For blades with a circular camber profile, the camber half-angles θ_1 and θ_2 are equal due to symmetry i.e. $\theta_1 = \theta_2 = \theta/2$. Finally, the distance between two adjacent blades is referred to as the pitch, s and the location of maximum camber is given by a .

2.2.3 Dimensionless performance parameters for axial compressors

As is the convention in fluid mechanics, compressor characteristics are often stated in terms of dimensionless numbers. In this section, several important dimensionless numbers are briefly discussed.

Compressor performance is often stated in terms of the ratio between inlet and outlet pressure, P_r . Cumpsty (1989) shows that P_r can be written in terms

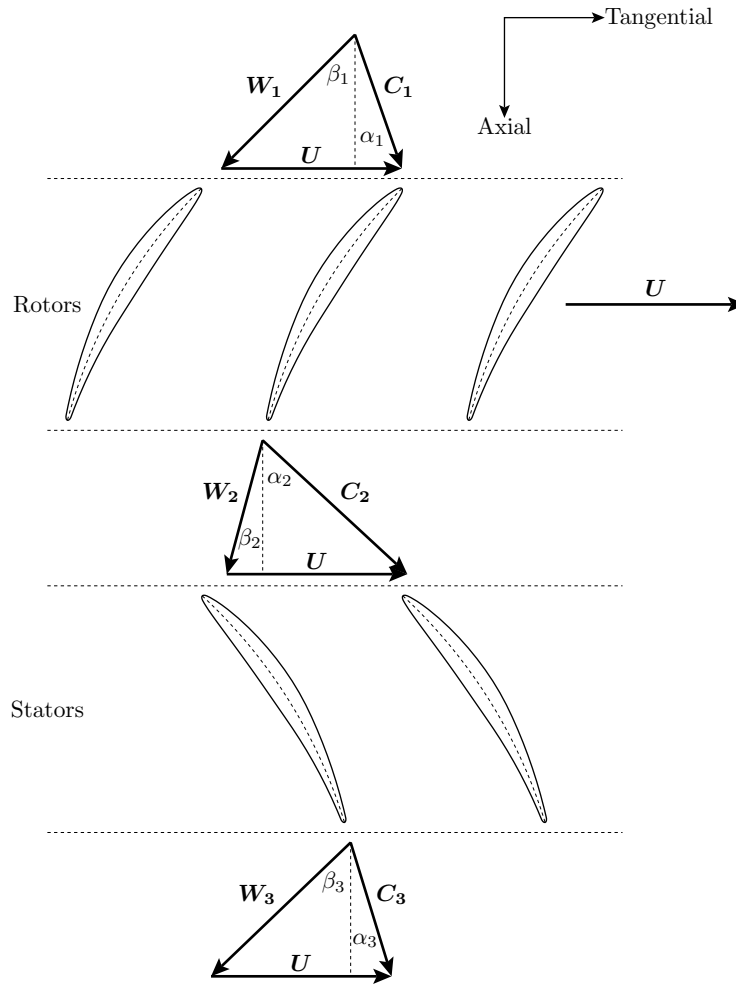


Figure 2.1: The velocity triangles of an axial compressor. C_1 and C_2 are the absolute velocities into the rotor and stator respectively, while W_1 and W_2 are those velocities relative to the rotor. U is the rotational velocity at the given radius.

of Mach number based on blade speed, M_u .

$$P_r = \frac{p_2}{p_1} = 1 + \gamma M_u^2 \frac{\Delta p}{\rho_1 U^2} \quad (2.2.2)$$

Since for the Rofanco compressor $M_u \ll 1$, it is more convenient to report the pressure rise using the pressure coefficient, ψ :

$$\psi = \frac{\Delta p}{\rho U^2} \quad (2.2.3)$$

The total-to-total, total-to-static and static-to-static pressure coefficients are denoted by ψ_{tt} , ψ_{ts} and ψ_{ss} respectively.

Another essential property in the performance of axial compressors is the ratio between axial velocity and the rotational speed of the rotor blades. This

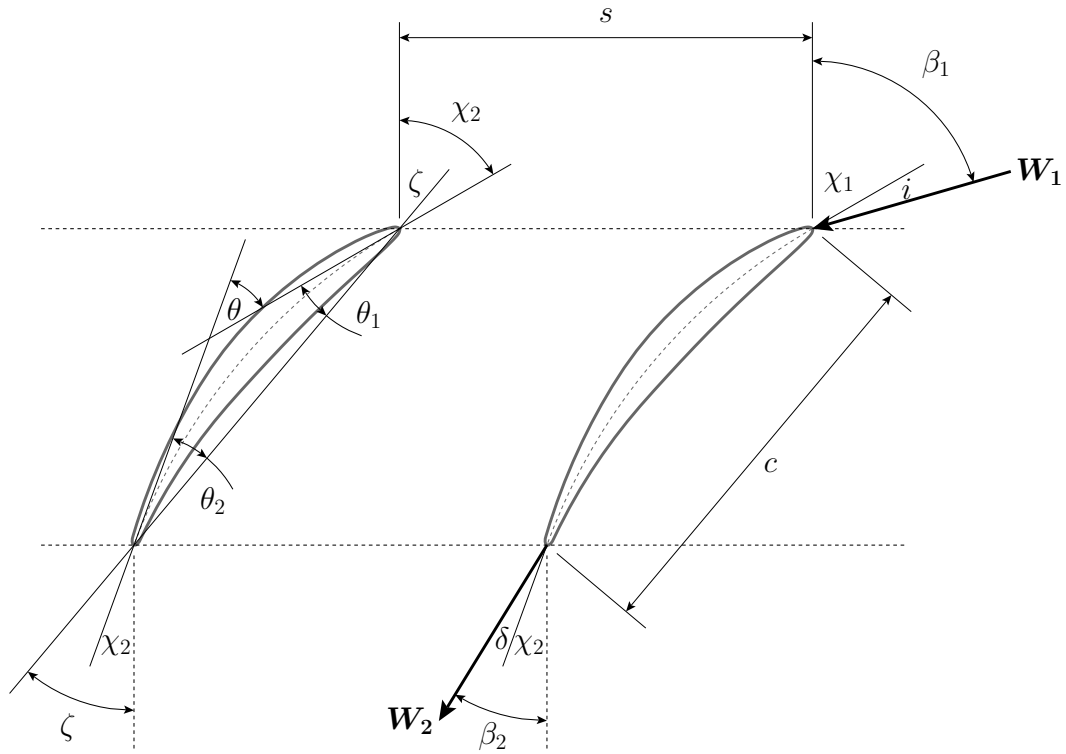


Figure 2.2: Two compressor blades in cascade configuration. Rotor blade (χ) and flow (β) angles are shown in the sketch. For stator blades, these should be substituted with stator blade (κ) and flow angles (α).

quantity is referred to as the flow coefficient and is defined by:

$$\phi = \frac{C_a}{U} \quad (2.2.4)$$

Simple trigonometry shows that the flow coefficient determines the inlet flow angle to the first rotor stage. Cumpsty (1989) states that the flow coefficient effectively dictates the performance of the stage and for typical compressors $\phi \in (0.3; 0.9)$.

Reaction ratio is a measure of relative magnitude of the rise in stagnation enthalpy in the rotor, when compared to the magnitude of the rise in stagnation enthalpy in the stage. Dixon (2005) shows that the reaction ratio may be expressed as:

$$R = \frac{1}{2} (1 + \phi(\tan \beta_2 - \tan \alpha_1)) \quad (2.2.5)$$

The stage loading coefficient (Ψ) provides a measure of the energy imparted to the fluid relative to the energy available. It is defined by the equation below (Sayers (1990)).

$$\Psi = \frac{h_{03} - h_{01}}{U^2} = \phi (\tan \alpha_2 - \tan \alpha_1) \quad (2.2.6)$$

The solidity of a blade set (σ) is defined as ratio between the inter-blade spacing and the blade chord.

$$\sigma = \frac{c}{s} \quad (2.2.7)$$

2.2.4 Losses in axial compressors

In order to better understand and control the losses that occur in axial compressors, it is useful to divide them into three categories: profile losses from the blade surfaces, skin friction losses from the walls of the hub and shroud and secondary losses, which is a grouping for all other losses. Fundamentally, all losses are due to viscous effects and shear work performed by the fluid (Cumpsty (1989)).

The most easily recognizable form of loss in an axial compressor is profile loss, produced by the operation of the compressor blades away from the end wall (Cumpsty (1989)). The flow in an axial compressor is more complex than the flow over an isolated airfoil. The flow field surrounding a single axial compressor blade is affected by the neighboring blades. In addition, the relative velocities of the oncoming flow vary with radius. Finally, the tip gap and end wall have a significant effect on the overall performance of the compressor (Cumpsty (1989)). Profile losses may be estimated using airfoil drag coefficients. They are not the most significant form of loss, if the airfoil is operated near its design conditions (Cumpsty (1989)).

Losses in the annulus boundary layer, also referred to as skin friction losses, are difficult to analyze due to the complex and three dimensional nature of the flow (Aungier (2003), Horlock (1958)). Despite the complexity of the phenomena, Howell (1942) states that annulus drag may be estimated by the simple expression:

$$C_{Da} = 0.02 \frac{s}{h} \quad (2.2.8)$$

There are other sources of loss in axial compressors, such as blade tip clearance loss and shroud seal leakage, which are grouped under the term secondary losses. Despite their name, secondary losses are of similar orders of magnitude to profile losses and skin friction. A common estimation of secondary losses, attributed to Howell (1942), is given by:

$$C_{Ds} = 0.018 C_L^2 \quad (2.2.9)$$

Due to the growth of the boundary layer through the machine, the axial velocity at the mean radius is higher than the mean axial velocity (Horlock (1958)). This blockage decreases the amount of work done by the stage. In order to account for this, the work-done factor (λ) is introduced:

$$T_{0,3} - T_{0,1} = \lambda U C_a (\tan \beta_1 - \tan \beta_2) / c_p \quad (2.2.10)$$

Cumpsty (1989) suggests that the work-done factor should be high at the inlet and steadily decreased as the boundary layers grow. Typical values of the

Table 2.1: Typical values of the work-done factor for various numbers of stages, interpolated from a graph in Sayers (1990).

Stage	1	2	3	4	5
λ	1.00	0.94	0.93	0.91	0.87

work-done ratio are given in Table 2.1. In Section 2.3.4, the work-done factor is used to account for the growth of the boundary layer through the machine while converting the pressure rise over a single stage into the pressure rise over N identical stages.

2.2.5 Operating limits of axial compressors

In normal operation, decreasing the mass flow rate through an axial compressor increases the pressure rise over the compressor. This process is not without limit however: a point is reached where decreasing mass flow further results in a sudden, dramatic change in the flow pattern through the compressor (Cumpsty (1989)). Lowering the mass flow rate beyond this point results in the compressor entering either stall or surge.

When discussing stall, it is important to distinguish between separated flow in a blade row and compressor stall, the point at which decreasing the mass flow through the machine results in a drop in the pressure rise over the machine. Compressors can perform satisfactorily with large regions of separated flow and separated flow in a particular blade row is not an indicator of compressor stall (Cumpsty (1989)).

Although separated flow is not a sufficient condition for compressor stall, it is the root cause of rotating compressor stall i.e. the propagation of a stall cell around the annulus, relative to the blades. This phenomenon is explained as follows by Horlock (1958): consider three adjacent blades A , B and C , shown in Figure 2.3. Initially some non-uniformity causes separation to occur at blade B , which causes blockage in the blade passage AB . This blockage causes the incoming flow to deflect around passage AB , which increases the incidence on blade A and decreases the incidence on blade C . The increased incidence causes flow over blade A to separate and the resulting decrease in incidence on blade B causes the flow to reattach, unblocking passage AB . In this manner the stall cell propagates around the blade row. This description is somewhat simplified, since multiple blade passages are involved in the initiation of rotating compressor stall (Cumpsty (1989)).

In contrast to compressor stall, surge is an unsteady flow condition where the average mass flow through the compressor changes as a function of time as the compressor switches between a stalled and unstalled state (Cumpsty (1989)). The surge process may cause flow reversal in the machine, as well as high structural loading. For a more complete description of this topic, the reader is referred to Horlock (1958), Cumpsty (1989) and Aungier (2003).

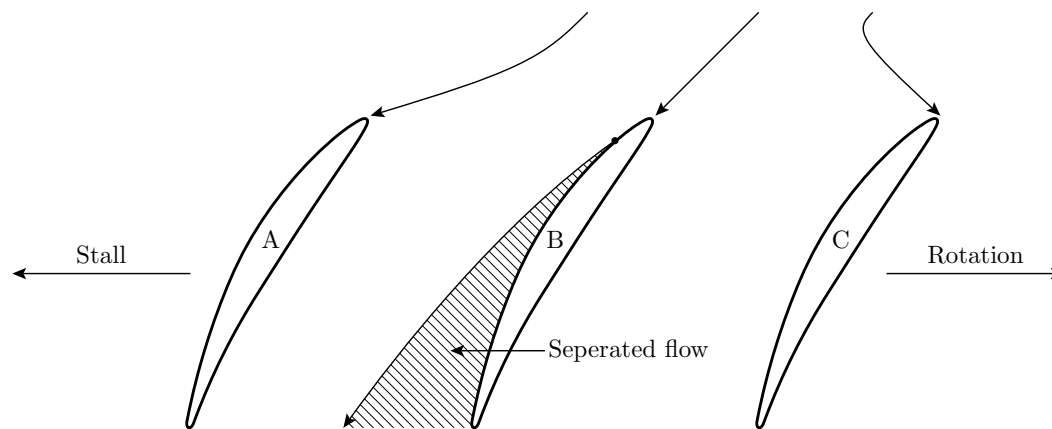


Figure 2.3: Rotating stall is caused by the separation of flow from one blade influencing the flow in surrounding blade passages. See the text for a simplified explanation of rotating compressor stall.

2.2.6 Structural loading on axial compressor blades

Axial flow compressor blades are typically long and slender and possess a small amount of mechanical damping (Cumpsty (1989)). It thus follows that the vibration of axial compressor blades can be a significant problem. Fortunately this problem is nearly always confined to the compressor rotor, since the large centrifugal loading on the rotor causes a tight contact with the hub, which results in low mechanical damping (Cumpsty (1989)).

A distinction is made between two types of vibration in axial compressors: forced vibration and flutter. Forced vibration is caused by rotors moving through disturbances, such as stator wakes and rotating stall cells. By contrast, flutter is self-excited and is much more difficult to predict. Consequently, flutter is of greater concern to the designer than forced vibration. A full treatment of these topics is far beyond the scope of this text; for more details see Raubenheimer (2011).

2.3 The Rofanco compressor

2.3.1 Introduction

The Rofanco compressor consists of three repeating stages of 43 rotor blades and 41 stator blades with removable inlet guide vanes. In this work, the inlet guide vanes are removed. The machine's original plastic blades were destroyed during testing and were subsequently replaced by aluminum NACA 65 series blades whose camber and stagger angles, as reported by Gill (2012), are given in Table 2.2. The machine has a hub diameter of 300 mm and a shroud diameter of 420 mm and operates at a design speed of 3000 rpm.

The effect of the tip clearance between the rotor and shroud and the stator

Table 2.2: The Rofanco rotor and stator stagger and camber angles as well as solidity, as a function of radius (Gill (2012)).

	Radius	ζ	θ	σ
Rotor	150 mm	38.00°	31.04°	1.305
	165 mm	45.00°	23.48°	1.186
	180 mm	49.40°	17.93°	1.088
	195 mm	53.00°	13.85°	1.004
	210 mm	56.10°	10.90°	0.932
Stator	150 mm	20.38°	46.28°	1.369
	165 mm	18.18°	43.39°	1.244
	180 mm	16.61°	41.05°	1.141
	195 mm	14.90°	40.57°	1.053
	210 mm	14.32°	40.00°	0.978

Table 2.3: The average rotor and stator tip gap, as measured by Gill (2012).

Blade row	Rotor tip gap	Stator hub gap
1	0.40 mm	0.68 mm
2	0.38 mm	0.54 mm
3	0.33 mm	0.50 mm
Average	0.37 mm	0.57 mm

and hub on performance is complicated and can have a large effect on the overall performance of the machine. The rotor and stator tip gap values, as recorded by Gill (2012), are presented in Table 2.3. In future numerical models, the average values of rotor tip gap and stator hub gap are used for all stages.

The Rofanco compressor, in its current configuration, has been studied in great detail. Preliminary performance charts were prepared by Roos (1995), who installed the blade set designed by Benadé (1987). More recently, exhaustive numerical and experimental studies of the Rofanco compressor, in all four quadrants of operation, were performed by Gill (2012). The experimental data of Gill, for the first quadrant of operation, is discussed in Section 3.7.

2.3.2 Analysis of current blade geometries

In this section an analysis of the Rofanco compressor is undertaken using semi-analytical two-dimensional methods. The purpose of this analysis is not to make performance predictions, but rather to gain a quantitative and qualitative understanding of the behavior of the machine.

Since inlet guide vanes are not used, axial inlet flow is assumed i.e. $\alpha_1 = 0^\circ$. Furthermore, since all three stages are identical, it may be assumed that outlet flow is also approximately axial i.e. $\alpha_3 = 0^\circ$. In order to determine the rotor outlet flow angle, it is first necessary to convert the stagger (ζ) and camber

Table 2.4: Rotor outlet flow angle (β_2) for each stage. The analytic result is calculated using the known blade angles in Table 2.2 and the deviation correlation of Equation 2.3.3. The experimental data is from Gill (2012). There is a good correlation between the analytic results and experimental data.

Method	Stage	165 mm	180 mm	195 mm
Analytic	-	39.66°	45.78°	50.53°
Five hole probe	1	42.15°	47.42°	51.84°
Five hole probe	2	41.60°	49.25°	54.11°
Five hole probe	3	38.70°	47.30°	51.45°
Hot film	1	38.73°	42.59°	49.55°
Hot film	2	38.64°	43.38°	50.06°

Table 2.5: Some dimensionless performance parameters. Note the high reaction ratio (R).

ϕ	R	Ψ
0.594	0.861	0.259

half angles (θ_1 , θ_2) to blade inlet and outlet angles (χ_1 and χ_2), using the geometry of Figure 2.2.

$$\chi_1 = \zeta + \theta_1 = \zeta + \theta/2 \quad (2.3.1)$$

$$\chi_2 = \zeta - \theta_2 = \zeta - \theta/2 \quad (2.3.2)$$

Using these equations, the blade angles for the current Rofanco blades may be determined. Outlet flow does not precisely follow the blade; there is some small deflection from the blade angle. This deflection may be estimated using a number of empirical correlations, such as those provided by Carter (1950), Lieblein (1960), Gostelow and Pollard (1967) or Howell (1942). In this work, the correlation of Howell (1942) is used, due to its simplicity and reported accuracy (Thomas (2005)):

$$\delta = \left(0.23 \left(\frac{2a}{l} \right)^2 + 0.1 \left(\frac{\chi_2}{50^\circ} \right) \right) \theta \left(\frac{s}{l} \right)^{\frac{1}{2}} \quad (2.3.3)$$

With the deflection and blade angles known, it is possible to calculate the approximate rotor outlet flow angles at the design point. These are summarized in Table 2.4 and provide reasonable agreement with experimental data. With the flow angles known, it is possible to calculate the flow coefficient, stage loading and reaction ratio of the compressor, summarized in Table 2.5. Details of these and earlier calculations are shown in Appendix A.

2.3.3 Capacity for improvement

Before undertaking any numerical design optimization, it is useful to first attempt to determine the potential for improvement, using the techniques of Section 2.3.2. For a given mass flow rate, the axial component of velocity (C_a) is approximately constant through the machine, since the Rofanco compressor operates in the incompressible flow regime. Rotational speed (ω) is also fixed. Finally, prescribing a value for β_2 implies a value for α_2 and vice versa, provided that U and C_a are fixed. The net result of this is that β_2 is the only flow parameter which may be freely modified. In this section, a two-dimensional analysis is undertaken for a range of β_2 values at various radii.

In the process of designing a new set of blades for the Rofanco compressor, the goal is to increase the static pressure out of the machine. With this in mind, the first property examined as a function of β_2 is the static-to-static pressure coefficient (ψ_{ss}) which is plotted as a function of β_2 in Figure 2.4. Examination of this figure reveals two trends: for a given flow angle a higher radius results in a higher pressure coefficient and decreasing β_2 results in an increase in pressure coefficient (ψ_{ss}). Both these phenomenon can be explained with reference to the second velocity triangle of Figure 2.1, specifically the fact that:

$$U = \omega r = W_2 \sin \beta_2 + C_2 \sin \alpha_2 = W_{2,x} + C_{2,x} \quad (2.3.4)$$

Since C_a is constant due to continuity, increasing r or decreasing β_2 will increase C_2 . Assuming this increased absolute velocity can be efficiently diffused, increasing r or decreasing β_2 will thus result in a higher static pressure rise over the machine.

Critically, efficient diffusion can not be guaranteed for all β_2 . Eventually, decreasing β_2 results in flow separating from the rotor blades. This separation of flow over the blades can lead to stall of the machine as a whole (discussed in Section 2.2.5). It is thus desirable to increase the pressure coefficient as far as possible, while maintaining unstalled flow.

Gill (2006) evaluated the accuracy of several empirical stall criteria, when applied to the Rofanco machine. He found the de Haller criterion gave the most accurate results. Fortuitously, this criteria is extremely simple: stall is deemed to occur when the velocity out of a blade row is less than 72% of the inlet velocity i.e. $W_2/W_1 \leq 0.72$.

Despite the de Haller criterion being the best of stall criteria Gill (2006) studied, it is still not sufficiently accurate for direct application. The rotor inlet-to-outlet velocity ratio is plotted in Figure 2.5 for several radii. The velocity ratios associated with the current blade set at design mass flow, range from approximately 0.57 to 0.63. These ratios are substantially below the de Haller limit of 0.72. The de Haller criteria can still provide useful indirect information relating to performance trends, if it is assumed that stall takes place at some unknown but approximately constant value of W_2/W_1 .

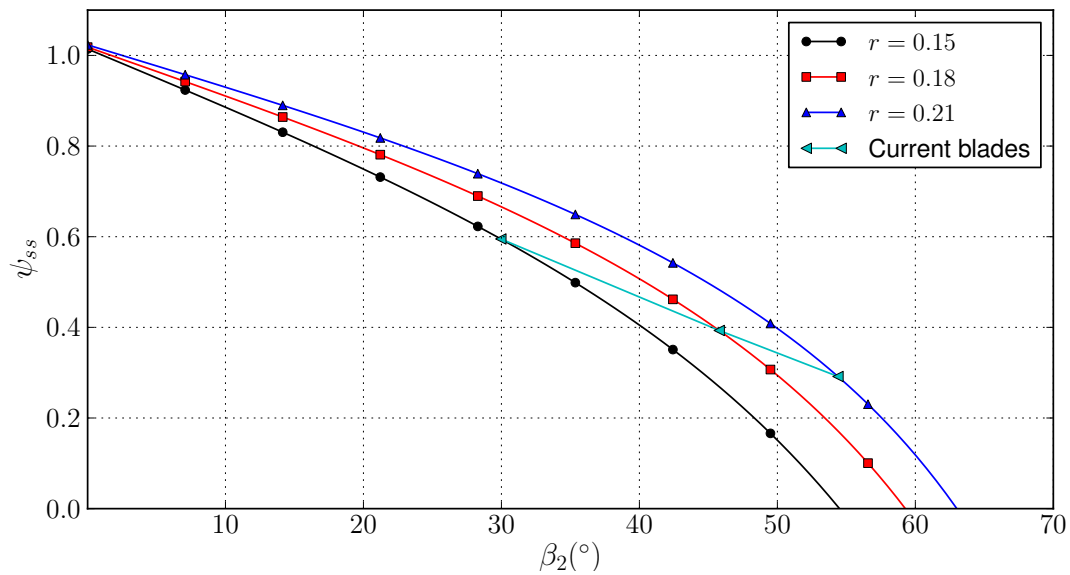


Figure 2.4: The pressure coefficients, as a function of β_2 , for various radii, at the design mass flow rate. Note that decreasing β_2 or increasing r will result in a higher ψ_{ss} . This is explained by reference to Equation 2.3.4.

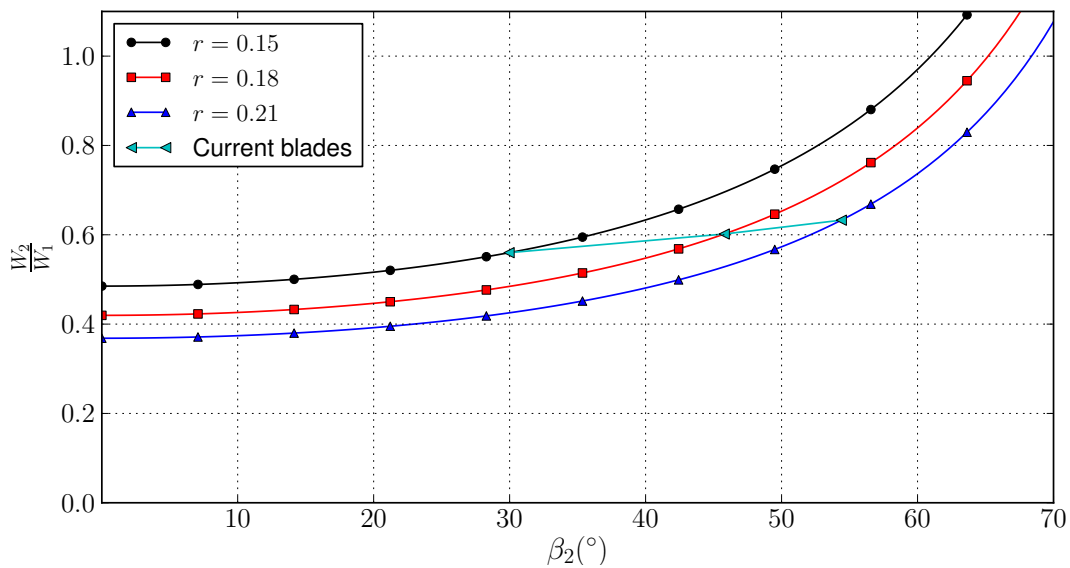


Figure 2.5: The rotor velocity ratio, as a function of β_2 for the Rofanco compressor at stall mass flow rate. For the range of β_2 values shown, the stator velocity ratio is higher than that of the rotor and can therefore be safely ignored.

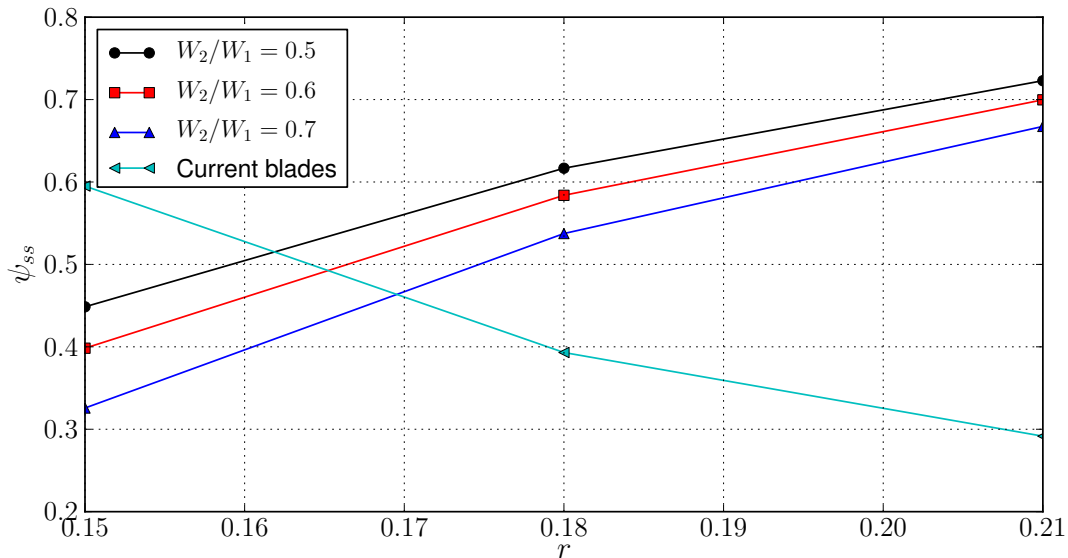


Figure 2.6: Static-to-static pressure coefficient as a function of radius for the current blades, as well as blades with a prescribed rotor inlet-outlet velocity ratio (de Haller’s criteria). Note how blades designed according to de Haller’s criteria deliver an increasing pressure coefficient as a function of radius, while the performance of the current blades deteriorates radially.

The static-to-static pressure coefficient (ψ_{ss}) for the current blade angles is plotted in Figure 2.6. Also plotted in this figure are several hypothetical blades generated by prescribing a value of W_2/W_1 and computing the blade angles which deliver this value at each radii. A clear trend is visible: blades generated by prescribing a constant value of the de Haller criterion produce a static-to-static pressure coefficient that increases as a function of radius, while for the current blades the pressure coefficient decreases as a function of radius.

Although the original design report is no longer available, it appears likely that the current blades were designed by applying the free-vortex condition:

$$C_x r = \text{constant} \quad (2.3.5)$$

This constraint is introduced in order to limit radial flow in the blade row, as the two dimensional methods used in this section assume negligible radial flow (Dixon (2005)). It should be noted that although increased radial flow commonly results in an increase in secondary losses (Dixon (2005)), it is not always detrimental. Indeed, an interesting consequence of allowing non-negligible radial flow is that on average the fluid leaves the machine at a greater radius than it entered with. This centrifugal velocity component increases the enthalpy rise over the machine (see Equation 2.2.1). It is common for axial compressors to exploit this effect by increasing the hub diameter through the machine (Cumpsty (1989)).

It seems probable that the performance of the Rofanco compressor may be increased through the dual mechanisms of removing the restriction on radial

flow and increasing the relative rotor outlet flow angle (β_2). It appears that there is sufficient prospect of obtaining a significant increase in static pressure rise to warrant the application of numerical design optimization techniques.

2.3.4 Computing the total pressure rise from the stage pressure rise

Early in the process, it was decided to design one set of blades for all three stages. This approach provides a significant advantage, as it drastically reduces the number of design variables required to describe the compressor. Since the computational cost of solving an optimization problem usually is a super-linear function of the number of design variables, reducing the number of design variables by a factor of three results in a significantly smaller computational problem.

The use of repeating stages does necessarily result in a lower increase in performance than designing different blades for each stage, since there are fewer free parameters, but this is deemed acceptable for the reasons outlined above. It should be noted that the properties of the fluid through the machine do not drastically differ due to the low amount of energy imparted to the fluid, making the implicit assumption of identical conditions in each stage reasonable.

Aside from computational cost, repeating stages offer another advantage, discussed in this section. Since the Rofanco compressor is a low pressure machine, the properties of the working fluid entering each stage are approximately identical i.e. $P_{03} \approx P_{01}$ and $T_{03} \approx T_{01}$. This suggests that it might be possible to estimate the pressure rise over the entire machine by simulating a single stage only. A first approach to calculating the total pressure rise would be to simply calculate the pressure rise over a single stage and multiple this by the number of stages in the machine.

$$\Delta p_{total} = n \Delta p_{stage} \quad (2.3.6)$$

This approach neglects the growth of the boundary layer through the machine, which reduces the performance of successive stages. In this section a more accurate approach is derived, using the work-done factor (λ). For an isentropic process, the ratio between temperature and pressure is given by:

$$\frac{p_a}{p_b} = \left(\frac{T_a}{T_b} \right)^{\gamma/(\gamma-1)} \quad (2.3.7)$$

For a compressor stage, it is possible to relate the actual temperature rise in the compressor, ($\Delta T_{0,ab} = T_{0,b} - T_{0,a}$) to the temperature rise that would have occurred, were the process isentropic and the axial velocity profile uniform ($\Delta T_{0,ab}^*$) using the isentropic efficiency ($\eta_{c,ab}$) and the work-done factor (λ_{ab}):

$$\Delta T_{0,ab} = \lambda_{ab} \eta_{c,ab} \Delta T_{0,ab}^* \quad (2.3.8)$$

From Equations 2.3.7 and 2.3.8, the following expression may be derived.

$$\Delta T_{0,ab}^* = \frac{T_{0,b}}{\lambda_{ab}\eta_{c,ab}} \left(\left(\frac{p_{0,a} + \Delta p_{0,ab}}{p_{0,a}} \right)^{\gamma-1/\gamma} - 1 \right) \quad (2.3.9)$$

Due to the small amount of energy imparted to the fluid by the machine and the fact that all compressor stages are identical, it is reasonable to assume the following:

$$\frac{T_{0,3}}{T_{0,1}} \approx 1 \quad \frac{p_{0,3}}{p_{0,1}} \approx 1 \quad \frac{\eta_{13}}{\eta_{35}} \approx 1 \quad (2.3.10)$$

From Equation 2.2.10 it can be deduced that $T_{0,13}^* = T_{0,35}^*$ when C_a is constant. Since the cross-sectional area of the Rofanco compressor is constant and it operates in the incompressible flow regime, this assumption is reasonable. By setting $T_{0,13}^* = T_{0,35}^*$ using Equation 2.3.9 and using the assumptions listed in Equation 2.3.10, it is possible to relate $\Delta p_{0,13}$ to $\Delta p_{0,35}$.

$$\Delta p_{0,35} = p_{0,3} \left[\left(1 + \frac{\lambda_{35}}{\lambda_{13}} \left[\left(\frac{p_{0,1} + \Delta p_{0,13}}{p_{0,1}} \right)^{(\gamma-1)/\gamma} - 1 \right] \right)^{\gamma/(\gamma-1)} - 1 \right] \quad (2.3.11)$$

At first glance, Equation 2.3.11 appears to be highly non-linear, however examination of Figure 2.7 reveals that for the domain of interest Equation 2.3.11 is well approximated by the simple relation:

$$\Delta p_{0,35} \approx \frac{\lambda_{35}}{\lambda_{13}} \Delta p_{0,13} \quad (2.3.12)$$

With the additional assumptions that $c_5 \approx c_3$ and $c_1 \approx c_3$, a similar relation can be derived for static pressure.

$$\Delta p_{35} \approx \frac{\lambda_{35}}{\lambda_{13}} \Delta p_{15} + \frac{1}{2} \left(\frac{\lambda_{35}}{\lambda_{13}} (c_5^2 - c_3^2) + (c_1^2 - c_3^2) \right) \approx \frac{\lambda_{35}}{\lambda_{13}} \Delta p_{13} \quad (2.3.13)$$

Thus, for a three stage compressor, where only the first stage is simulated, the total pressure rise accounting for the growth of the boundary layer through the machine is:

$$\begin{aligned} \Delta p_{17} &= \Delta p_{13} + \Delta p_{35} + \Delta p_{57} \\ &\approx \Delta p_{13} \left(1 + \frac{\lambda_{35}}{\lambda_{13}} + \frac{\lambda_{57}}{\lambda_{13}} \right) \\ &= \Delta p_{13} \left(\frac{3\lambda_{17}^{avg}}{\lambda_{13}} \right) \end{aligned} \quad (2.3.14)$$

Using the values from Table 2.1 for a three stage axial compressor, such as the Rofanco, yields:

$$\begin{aligned} \Delta p_{17} &= \Delta p_{13} \left(\frac{3(0.925)}{(0.96)} \right) \\ &= 2.891 \Delta p_{13} \end{aligned} \quad (2.3.15)$$

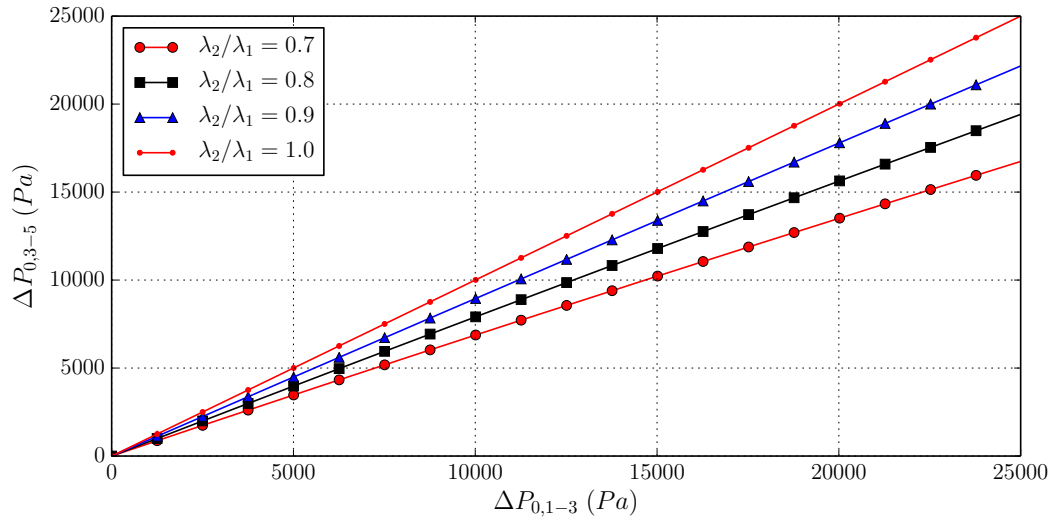


Figure 2.7: Equation 2.3.11 at atmospheric input conditions for a range of ΔP_{13} values. Despite its complicated appearance this equation is very nearly linear for the range of values considered.

It should be noted that when approximating the performance of the entire machine using a single stage simulation, it is necessary to constrain the outlet flow angle to be within some acceptable tolerance of the inlet blade angle, in order to keep the rotor incidence angle within acceptable limits. This topic is dealt with in more detail in Chapter 4.

2.4 Conclusion

This chapter introduced axial compressors in general and the specific machine used as a test bench in this work. Axial compressor theory was introduced in order to justify major design and simulation choices in subsequent chapters. The current configuration of the machine was studied and the chance of increasing performance through the application of numerical design optimization techniques was determined to be high. Finally, a method of reducing the computational domain of the numerical problem using the work-done factor was introduced. In the next chapter, the analytic and experimental analysis of this section is extended to include numerical analysis.

Chapter 3

Numerical analysis of the Rofanco compressor

3.1 Introduction

In this chapter, four computational fluid dynamics (CFD) models are discussed. These four models are generated from the combinations of two model domains and two sets of boundary conditions. The focus of this chapter is the development, evaluation and validation of these models, as well as the selection of a single model to be used in the design process. Section 3.2 provides an overview of the various components that make up the compressor model used in this work. Section 3.3 to Section 3.5 discuss specific modeling choices relating to boundary conditions, meshing, turbulence and other topics. Predicted compressor stall is discussed in Section 3.6. The developed models are evaluated by comparing the pressure coefficient as a function of mass flow and the velocity profiles as a function of radius to the experimental data gathered by Gill (2012), in Section 3.7 and Section 3.8 respectively. Finally, in Section 3.9, the run-times of the various models are compared.

3.2 Overview of the simulation process

Figure 3.1 is a diagrammatic representation of the process of simulating a single design point. Initial inputs to the simulation are a vector describing the compressor blades (\mathbf{x}), simulation parameters such as mass flow rate and solver settings, such as the number of model partitions. The simulation parameters and solver settings are input into a Python script which procedurally generates another Python script referred to as the model runner script. The model runner script is used to customize and run the CFX model, as well as generate CFX Command Language (CCL) fragments. These CCL fragments are executed by CFX, in order to set simulation parameters and export data of interest. The input vector is converted to an intermediate internal represen-

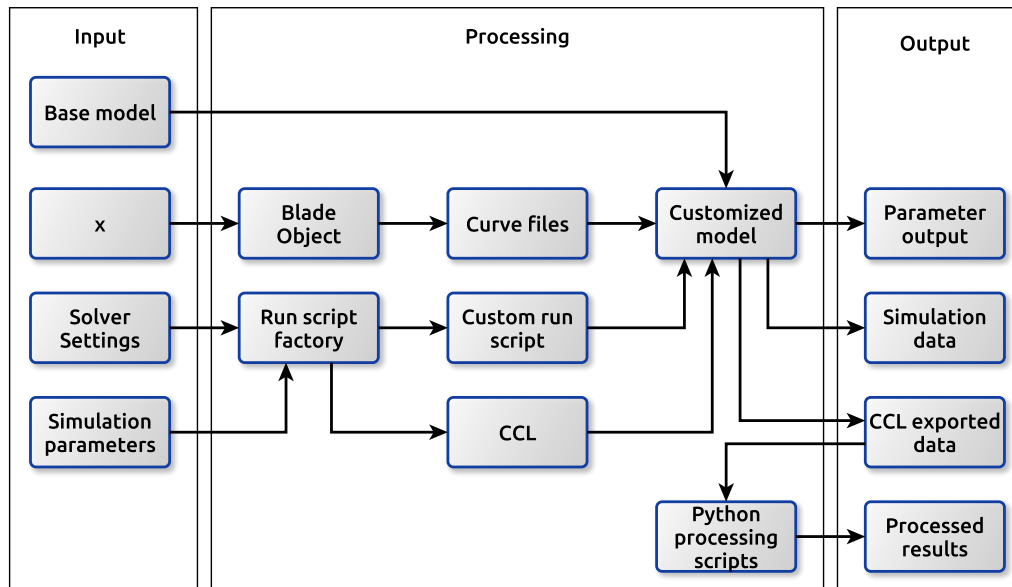


Figure 3.1: The simulation process in flowchart form. Model inputs are on the left, processing steps in the center and the outputs on the right. Not shown in this figure is the Python framework that orchestrates the simulation workflow.

tation of a blade, via a process described in Section 5.2.3. Using an internal blade representation allows the same simulation pipeline to be used no matter how the blade is parameterized. A converter was written to export the internal blade representation to files representing hub, profile and shroud curves, which could be imported by the customized model. Further details on the curve files are available in the ANSYS Turbogrid User’s Guide (2009).

In conjunction with the curve files, the generated run script and CCL files are used to create the customized, simulation specific model from the base model (a hand built template specific to a particular analysis type). The base numerical model is an ANSYS Workbench system, which uses ANSYS Turbogrid to import the model geometry and mesh the computational domain, ANSYS CFX to perform the simulation itself and ANSYS CFD Post for initial post-processing.

The model is executed by a Python job management script, which invokes ANSYS Workbench and executes the model runner script using Workbench’s IronPython interpreter (not shown in Figure 3.1). The model runner is robust and handles model errors, supports resuming models and retries model update failures. The retry procedure is important, because it distinguishes between models that fail to update for physical reasons, such as a failure to converge with iteration limits, from those that fail due to bugs in the ANSYS software or temporary unavailability of the necessary license server.

As output, the simulation produces the raw data at each node, a summary of important model parameters and specific outputs created by the CCL code

generated earlier in the process. The CCL output is further post-processed in order to derive additional results, such as the flow angle out of each stage.

3.3 Model domain and boundary conditions

Two different domains are modeled: a single compressor stage and all three compressor stages. As the name suggests, in the single stage model only one of the three identical stages is simulated. Equation 2.3.14 is used to convert the pressure rise over a single stage into the pressure rise over three stages, while accounting for boundary layer growth through the machine. The full compressor model includes all three stages of the Rofanco compressor, so no conversions are necessary.

In addition to the two different model domains, two alternative sets of boundary conditions are explored. In the first, model pressure and temperature are specified at the rotor inlet while mass flow is specified at the stator outlet. This configuration is common in turbomachinery and recommended by the ANSYS CFX Reference Guide (2010). This model is hereafter referred to as the “mass flow model”. In the second model, a velocity profile is specified at the rotor inlet, while pressure and temperature are specified at the stator outlet. This model is referred to as the “profile model”.

As its name suggests, the mass flow model provides direct control over mass flow through the machine. It only requires knowledge of the rotor-inlet temperature and pressure, which are known from atmospheric conditions. From an implementation perspective, the mass flow model is substantially easier to script, as all properties can be specified directly via CCL.

By contrast, the profile model allows only indirect control of mass flow through the machine and the inlet pressure at the rotor inlet. Control over mass flow is accomplished by linearly scaling the velocity profile in proportion to the ratio between the mass flow at which the profile was recorded and the desired mass flow. Implicit in this approach is the assumption that the form of the velocity profile remains consistent over the range of mass flows that are of interest. The inlet pressure can only be approximately controlled by estimating the pressure rise over the machine and setting the outlet pressure so that the desired inlet pressure is maintained. The velocity profile used in this work is an experimental velocity profile recorded by Gill (2012).

3.4 The meshing process

3.4.1 Overview

Candidate geometries were meshed using TurboGrid. TurboGrid is a meshing tool designed to enable the construction of structured hexahedral meshes of

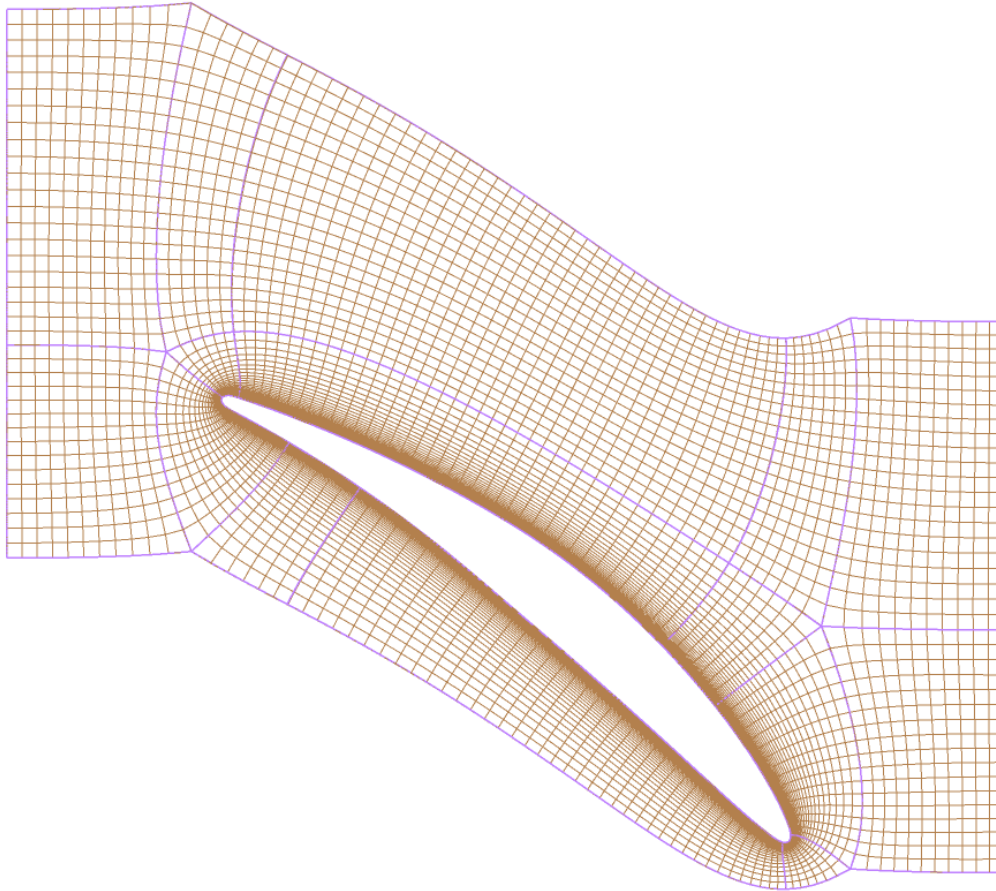


Figure 3.2: The ATM Optimized feature allows the generation of H-O grid meshes in a near automated fashion. Shown in this figure is a mesh of a Rofanco rotor blade created using the ATM Optimized feature

turbomachine blade passages. The primary consideration for the mesh generation process was the ease with which creation of the mesh could be automated, as it was expected that several thousand candidate blade geometries would need to be evaluated in the course of the design process.

3.4.2 Automating the creation of meshes

The ATM Optimized feature of TurboGrid was used to create meshes in an automated fashion. This feature allows the creation of H-O grid meshes while avoiding the need for manual control point adjustment by providing several constants controlling mesh refinement (ANSYS TurboGrid User's Guide (2009)). It should be noted that the ATM Optimized feature should only be used for blades with rounded edges (ANSYS CFX-Solver Modeling Guide (2010)). A sample mesh, generated with the ATM Optimized feature is shown in Figure 3.2.

The ATM optimized feature requires the selection of three constants: the global size factor (GSF), boundary layer proportional refinement factor (BLPRF) and spanwise proportional refinement factor (SPRF). The GSF is a multiplier to all individual element counts. Increasing GSF results in a super-linear increase in the number of elements in the model. BLPRF is a multiplier of GSF, responsible for increasing the resolution of the mesh within the O-grid. Increasing BLPRF increases the resolution of the grid near the blade surface and so increases the resolution in the boundary layer. SPRF controls the distribution of elements in the spanwise direction. Increasing SPRF increases the number of elements in the spanwise direction to maintain the approximate relation, where AR denotes aspect ratio:

$$\text{AR}_{\text{element}} \approx \frac{\text{AR}_{\text{blade}}}{\text{SPRF}} \quad (3.4.1)$$

3.4.3 Finding acceptable mesh constants

In order to automatically generate meshes for candidate compressor geometries, it is necessary to choose values for BLPRF, SPRF and GSF. The aim of this section is to find the combination of GSF, BLPRF and SPRF that leads to meshes of acceptable quality with the minimum number of nodes and elements required for convergence. To ensure that the results of this investigation are applicable to the procedurally generated meshes used to evaluate design geometries, no special refinements are made to any generated meshes.

Early experimentation revealed that low refinement in the passage and high refinement in the boundary layer (SPRF = 0.5, BLPRF = 3.0) resulted in faster convergence. These parameters result in a mesh that is highly refined in the boundary layer, while allocating the lowest permissible number of elements to the blade passage. Figure 3.3 plots the pressure rise over a single stage of the compressor as GSF is increased, for constant values of BLPRF = 3.0 and SPRF = 0.5.

Convergence occurs at GSF = 1.4. This corresponds to $n_{\text{elements}} \approx 844(10^3)$, which is similar to the result obtained by Gill (2012), who reported convergence at $n_{\text{elements}} \approx 2.8(10^6)$ when simulating all three stages of the compressor. In all future simulations, meshing parameters of BLPRF = 3.0, SPRF = 0.5 and GSF = 1.4 are used.

3.4.4 Evaluating mesh statistics

Although the true measure of mesh quality is the independence of fluid quantities of interest from the mesh itself, it is helpful to examine common heuristic measures of mesh quality. Table 3.1 lists several important mesh statistics, as measured by the CFX solver, along with the acceptable limits according to the ANSYS CFX-Solver Modeling Guide (2010). All measures of mesh quality are

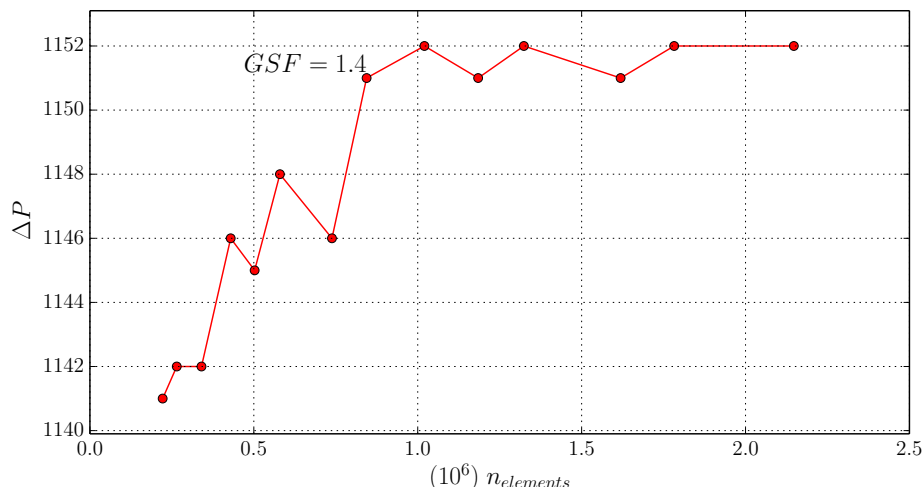


Figure 3.3: Convergence for various mesh parameters. All meshing settings converge to $\Delta P \approx 1022$ Pa except $GSF = 4$, which showed unusually poor mesh quality.

Table 3.1: All compressor mesh statistics fall into the acceptable range defined by the CFX solver, with 99% falling into the good range. All elements with a high aspect ratio occur in the boundary layer and are aligned with the flow.

Mesh statistic	Acceptable	Good	Rotor	Stator
Maximum aspect ratio	$\leq 10(10^3)$	$\leq 100(10^3)$	6517	4450
Minimum orthogonality	≥ 20 deg	≥ 50 deg	40.6 deg	39.0 deg
Maximum expansion ratio	≤ 20	≤ 5	14	12

in the “acceptable” range and in fact over 99% of volumes fall into the “good” range.

It is worth briefly discussing the maximum aspect ratio, since although it is within the acceptable limits prescribed by the CFX solver, the maximum value of 6517 indicates an extremely skewed element. The limits prescribed for aspect ratio depend on whether the simulation is run in single or double precision. If run in single precision, the acceptable limit is 10^3 , but in double precision that limit is $100(10^3)$. Therefore, a mesh containing elements with an aspect ratio of 6517 is only suitable when the solver is run in double precision mode. A moderating factor with respect to the high aspect ratio is that all skewed elements occur in the boundary layer and are aligned with incoming flow. According to the ANSYS CFX Reference Guide (2010), aspect ratios in the boundary layer can be much higher (on the order of 10^5 to 10^6).

3.4.5 Meshing in the boundary later

The dimensionless number y^+ is a measure of how close the first grid cell is to the wall (Versteeg and Malalasekera (1995)). The choice of turbulence model

Table 3.2: y^+ values for various surfaces in the Rofanco compressor. These low values indicate that the mesh resolves the laminar sublayer and is thus suitable for use with low Re turbulence models.

Surface	Rotor		Stator	
	Avg.	Max	Avg.	Max
Blade	1.56	5.33	0.92	3.69
Hub	1.45	4.85	1.15	1.96
Shroud	1.75	4.05	0.74	2.37

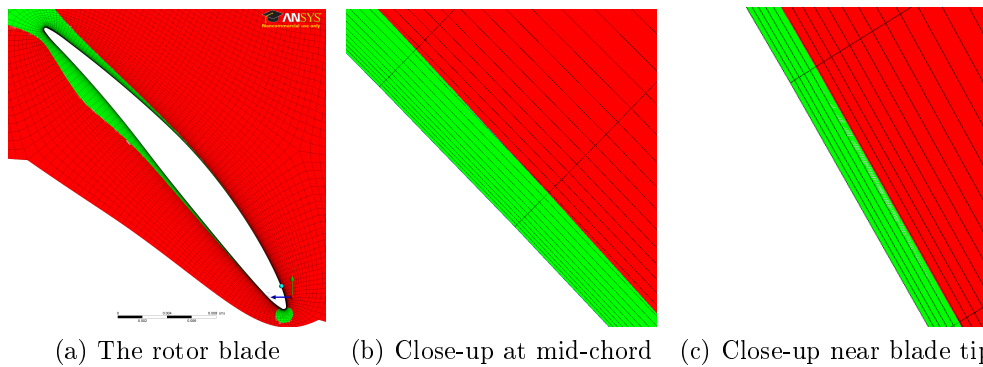


Figure 3.4: The rotor blade mesh and boundary layer at mid-span. Notice from the closeups of the mesh at mid-chord, that there are at more than ten points in the boundary layer.

can place restrictions on the minimum or maximum allowable y^+ value and the number of grid nodes inside the boundary layer. Table 3.2 lists the average and maximum y^+ values for the hub, shroud and blade surfaces. The average value for the rotor and stator blades is around $y^+ \approx 1$, indicating that the mesh typically resolves the laminar sublayer and is suitable for use with low Re turbulence models.

A two level contour plot of the velocity field and mesh around the rotor at mid-span is shown in Figure 3.4. Contour levels are set at $V_{contour} = 50$ m/s well below the relative velocity of the fluid with respect to the rotor ($V_1 \approx 65$ m/s). This means that any estimates of the boundary layer in Figure 3.4 are conservative, since $V_{contour} \leq (0.99)V_1$. At mid-chord, there are more than ten points in the boundary layer.

3.5 Some specific modeling choices

3.5.1 Turbulence modeling and near wall treatment

Two important and related modeling choices are the choice of turbulence model and choice of near wall treatment. In their work modeling the Rofanco com-

pressor, Raubenheimer (2011) and Gill (2012) both used the Spalart-Allmaras turbulence model (Spalart and Allmaras (1992)). Unfortunately, this model is not officially supported in the version of CFX used in this work, so another model had to be chosen.

The $k-\omega$ SST model was developed in Menter (1993) and expanded in Menter (1994). It combines the advantages of the $k-\epsilon$ and $k-\omega$ models by blending them in such a way as to make the $k-\omega$ model active near the model surface and the $k-\epsilon$ model active in the free steam (Carregal-Ferreira *et al.* (2002)). The ANSYS CFX Reference Guide (2010) suggests $k-\omega$ SST model for turbomachinery. This model was investigated for use in modeling the Rofanco compressor by Gill (2012), who found that it delivered results of comparable accuracy to the Spalart-Allmaras model, but resulted in a model that was more computationally expensive to solve.

The automatic wall function treatment available in CFX switches between a low Re model and a wall function model, based on y^+ value. It allows smooth refinement from coarse to fine mesh, without the need to ensure that the cell nearest the wall is in the log-law region (i.e. $y^+ \geq 12$) for wall functions or that the laminar sublayer is resolved (i.e. $y^+ \approx 2$) for low Re models.

In this work, it was found that the $k-\omega$ SST turbulence model, combined with the automatic wall function treatment yielded the best results. This conclusion was later reached independently by Brandsen (2013). It is required for the $k-\omega$ SST turbulence model that at least ten nodes be in the boundary layer and that the node next to the wall have a y^+ value of approximately 20 or less (ANSYS CFX Reference Guide (2010)). From Table 3.2 and Figure 3.4 it can be observed that both these conditions are met.

3.5.2 Rotor-stator interfaces

The rotor stator interface is modeled using the “stage” model in CFX, also referred to as a “mixing plane” model in literature. This model circumferentially averages results on the upstream interface and applies these averaged results to the downstream interface. Figure 3.5 shows static pressure at a rotor outlet and the associated pressure field at the inlet to the downstream stator when applying a mixing plane model.

This averaging process imposes a mixing loss upon the flow, equivalent to assuming that all upstream axial velocity profiles are mixed out (ANSYS CFX-Solver Modeling Guide (2010)). While mixing plane models account for averaged rotor-stator interactions, they cannot capture transient interactions such as rotating stall. As such, the mixing plane model is not suitable for compressors with significant wake interactions or for the prediction of blade loading (ANSYS CFX Reference Guide (2010)). Due to the large gaps between blade rows in the Rofanco compressor, it is reasonable to expect that substantial mixing takes place, making the mixing plane model an acceptable choice.

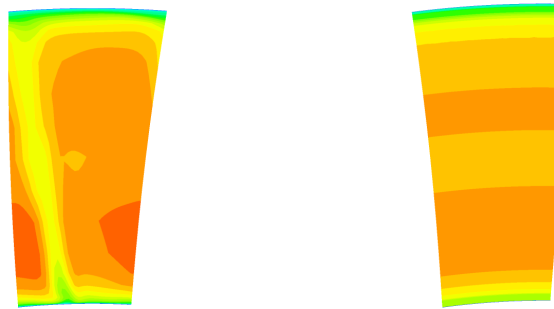


Figure 3.5: Pressure on upstream (a) and downstream (b) interfaces using the mixing plane approach to deal with rotor-stator iterations. Note the circumferential averaging on the downstream interface.

3.5.3 Convergence

Several criteria were required to be simultaneously satisfied in order to judge the simulation as converged. It was required that the RMS residual values were less than 10^{-5} and the maximum values of the residuals were less than 10^{-4} . It was further required that the mass flow imbalance between inlet and outlet be less than 1%. Finally, it was required that the difference in the static pressure rise between inlet and outlet in subsequent iterations be less than 1%.

3.6 Estimating the stall margin

Accurately predicting the stall point of a compressor using numerical methods is a difficult undertaking. Numerical methods have a tendency to under-predict the stall margin (ANSYS CFX Reference Guide (2010)). Fortunately, for the purposes of this work, it is not necessary to accurately predict the stalling mass flow. Instead, the aim is to preserve the current stall margin in any new designs. Assuming the ratio between the numerically predicted and experimentally observed stalling mass flow remains constant, it is sufficient to require that any proposed new blade design have a predicted stalling mass flow less than that of the current design. Of course, accurate predictions of stall margin, while not strictly needed, are desirable as they increase confidence in the models.

Before attempting to predict the occurrence of stall, it is first necessary to clarify what is meant by stall in this case. For the purposes of design, the stall point is taken to mean the point at which flow through a blade row is sufficiently separated so as to cause a decrease in the pressure rise over the machine, as mass flow is decreased. By this definition of stall, it is possible for a portion of a blade or blades to be stalled, yet the compressor as a whole to be unstalled.

In order to determine whether a candidate geometry is stalled using only

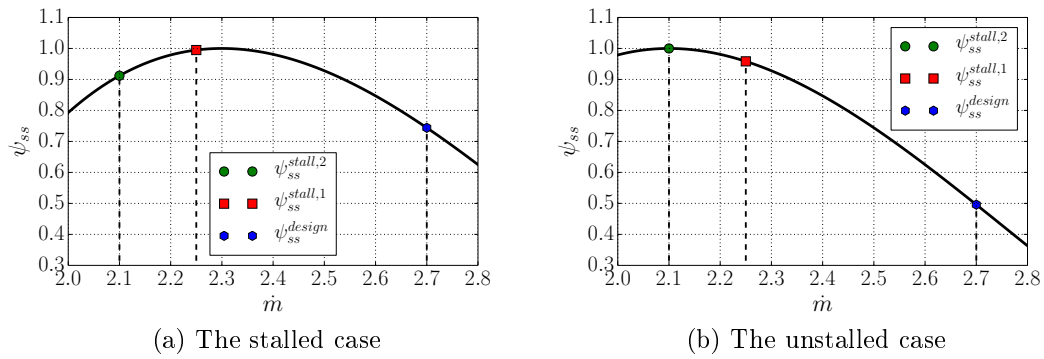


Figure 3.6: Three static simulations can be used to estimate whether the compressor is stalled. If the pressure coefficient at the lowest mass flow is lower than that of the intermediate or design mass flow, then the stall point must necessarily occur at a mass flow greater than the minimum mass flow.

static simulations, three simulations are run at different mass flows. The first simulation is run at the design mass flow rate, m_{design} and is used to evaluate the performance of the compressor blade geometry under consideration. The remaining pair of simulations are run at the maximum allowable stall mass flow and a mass flow slightly below this. These mass flow rates are referred to as $m_{stall,1}$ and $m_{stall,2}$, with $m_{stall,2} < m_{stall,1}$.

Together, these three samples can be used to determine whether or not the compressor is in stall. If either of ψ_{ss}^{design} or $\psi_{ss}^{stall,2}$ are greater than $\psi_{ss}^{stall,1}$, it is necessary that the peak of the performance curve (i.e. the stall point) occurs at a greater mass flow than $m_{stall,1}$. This reasoning is illustrated in Figure 3.6. Restating this in terms of inequalities:

$$\begin{aligned}
 \text{Unstalled:} & \quad \psi_{ss}^{stall,2} \leq \psi_{ss}^{stall,1} \leq \psi_{ss}^{design} \\
 \text{Stalled:} & \quad \psi_{ss}^{stall,1} \leq \psi_{ss}^{stall,2} \quad \text{or} \quad \psi_{ss}^{design} \leq \psi_{ss}^{stall,1}
 \end{aligned} \tag{3.6.1}$$

3.7 Comparing numerically predicted and experimentally measured pressure coefficients

The numerically predicted static-to-static pressure coefficient is plotted against the experimental data measured by Gill (2012) in Figure 3.7. Since the validity of the non-transient models used in this work are doubtful for mass flows less than the stalling mass flow, numerical predictions are not shown for these mass flows. For mass flow rates near the design point ($2.6 \text{ kg/s} \leq m \leq 3.1 \text{ kg/s}$) both the full velocity and mass flow compressor under-predicted the static-to-static pressure coefficient, although the profile model is noticeably more

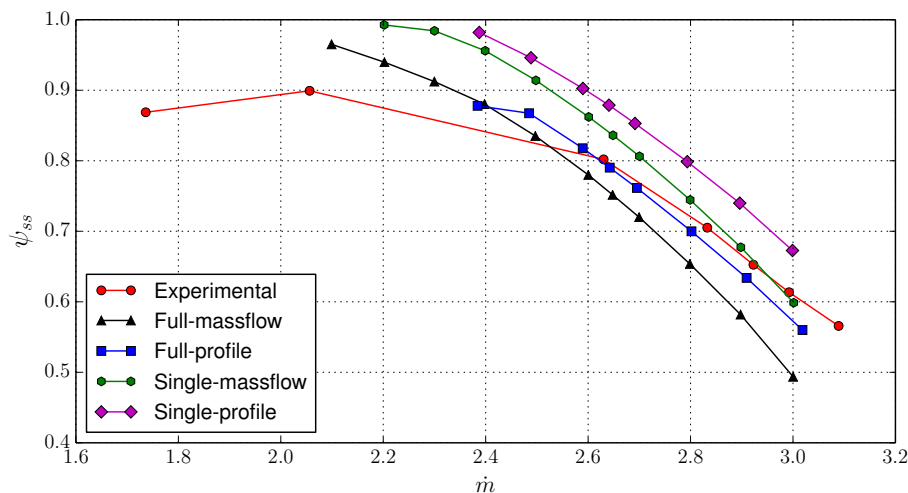


Figure 3.7: In this figure the experimental data of Gill (2012) is plotted against the numerical predictions of the basic and single stage models. It is clear that there is reasonable agreement in the region of the design point ($m = 2.7$ kg/s) and that the stall point is correctly predicted ($m \approx 2.1$ kg/s)

accurate than the mass flow model. It is possible that this underprediction is due to losses imposed by the mixing-plane interface model, described in Section 3.5.2. The full profile model makes the most accurate predictions in the region of the design point and in general, the profile models predict higher pressure rises than the mass flow models.

It is difficult to determine the experimental stall point, due to the sparsity of experimental data in the region. It appears likely that stall occurs in the range $2.2 \text{ kg/s} \lesssim m_{stall,exp} \lesssim 2.4 \text{ kg/s}$. The mass flow models stall at approximately 2.1 kg/s , slightly lower than $m_{stall,exp}$, while the profile models stall at 2.4 kg/s , slightly higher $m_{stall,exp}$. It is usual for numerical models to stall at slightly higher mass flow rates than is observed experimentally (ANSYS CFX Reference Guide (2010)).

3.8 Comparing numerically predicted and experimentally measured velocity profiles

3.8.1 Introduction

In this section, the numerically computed velocities from the profile models are compared to those obtained via experimentation. The velocities of the mass flow models are significantly different to both those obtained via experimentation and from the profile models and are thus excluded from further analysis. Since only a limited number of measurements are available for radial velocity

and it is an order of magnitude lower than other components, only axial and tangential velocities are discussed. It is believed that the five-hole probe data are more accurate than the hot-film data as the hot-film data are susceptible to distorted readings from unsteady effects.

3.8.2 Axial velocity predictions

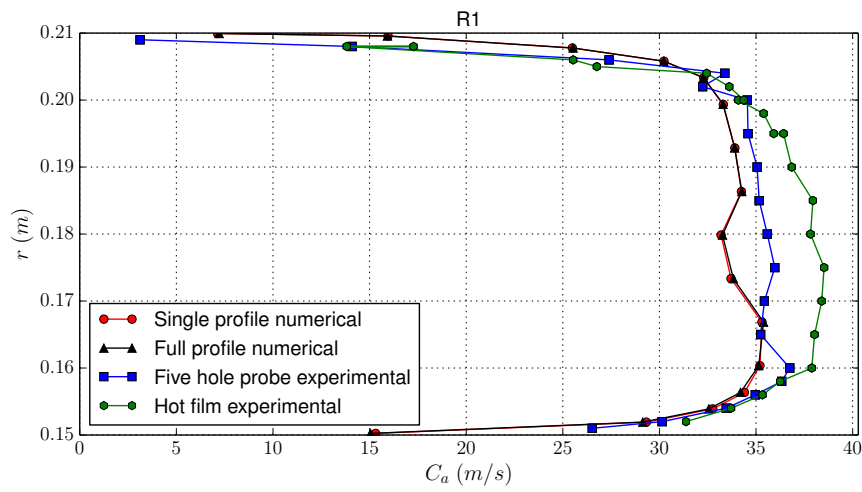
Axial velocity measurements are shown in Figure 3.8. An initial examination of axial velocity at R1, R2 and R3 suggests that there is reasonable agreement between the numerical and experimental results. At R1, the form of the numerical velocity profile is similar to that of the five-hole probe, but different to that of the hot-film data. As expected, the single stage profile model and full profile model provide nearly identical predictions at R1. The width of the hub and shroud boundary layers for the numerical model as well as the five-hole probe and hot-film sensor are similar. At R2, the width of the shroud boundary layer is under-predicted while the width of the hub boundary layer is over-predicted. Once again, there is similarity in magnitude and form between the numerical data and five-hole probe data, with the hot-film data different to both. There is no hot-film data available for R3 and again the width of the shroud boundary layer is under-predicted while the width of the hub boundary layer is over-predicted. At R3, the difference in form between the numerical and five-hole probe data is most pronounced, with the numerical data not showing the bulge occurring at $0.155\text{m} \lesssim r \lesssim 0.175\text{m}$. Brandsen (2013) expresses concerns that the five hole probe may be protruding into the flow, resulting in the larger than expected shroud boundary layer observed in the experimental data.

Table 3.3 provides the area-weighted average axial velocity for the numerical simulation, as well as that recorded by the five-hole probe and hot film sensor at R1, R2 and R3. This data shows the variability of the hot-film data, which fluctuates between 98.65% and 105.71% of the inlet average-weighted average axial velocity, as measured by the five-hole probe.

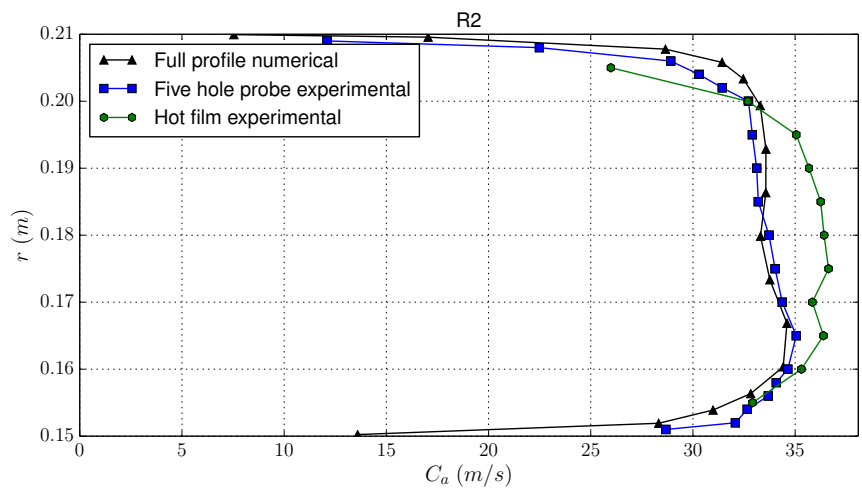
The data of Table 3.3 provides another interesting observation: at R2, the five-hole probe velocity is 96.66% of that at the inlet and at R3, the five-hole probe velocity is 101.15% of that at the inlet. It is expected that density increases through the machine, as pressure increases. This means that by conservation of mass, axial velocity should decrease through the machine, provided area stays constant. Despite this, axial velocity increases by just under 5% between R2 and R3. This deviation helps establish a probable lower bound of approximately 5% on experimental error.

3.8.3 Tangential velocity predictions

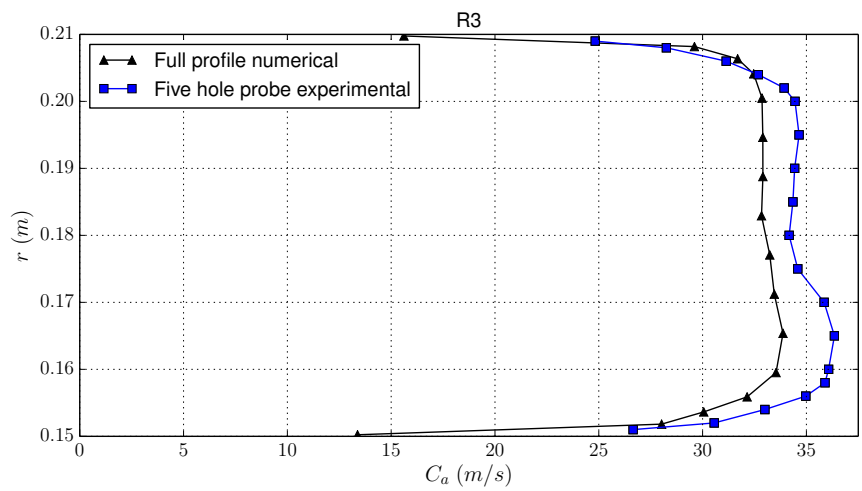
Tangential velocity measurements are shown in Figure 3.9. Once again, the single stage and full profile models provide nearly identical predictions at R1,



(a) R1 axial velocity measurements



(b) R2 axial velocity measurements



(c) R3 axial velocity measurements

Figure 3.8: In this figure, the axial velocity predictions of the profile model are plotted against the experimental data of Gill (2012).

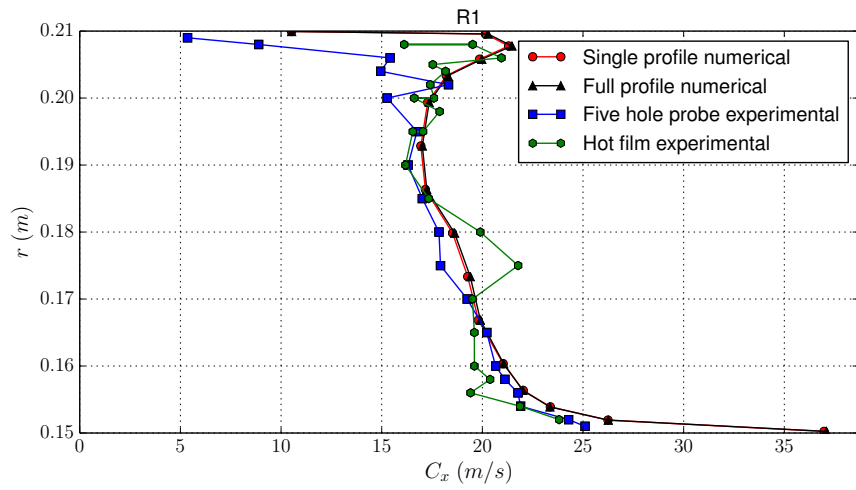
Table 3.3: Area-averaged axial velocity. Note the $\sim 5\%$ increase in axial velocity between R2 and R3 for the five hole probe, likely attributable to experimental error.

Blade row	Data source	$\overline{C_a}$ (m/s)	$\overline{C_a}/\overline{C_a}^{five-hole,R1}$
S0	Five hole probe	33.782	100.00%
	Hot film	33.325	98.65%
	Full profile numerical	32.897	97.38%
R1	Five hole probe	33.645	99.60%
	Hot film	35.710	105.71%
	Full profile numerical	32.625	96.58%
R2	Five hole probe	32.653	96.66%
	Hot film	34.884	103.26%
	Full profile numerical	32.426	95.99%
R3	Five hole probe	34.171	101.15%
	Full profile numerical	32.223	95.39%

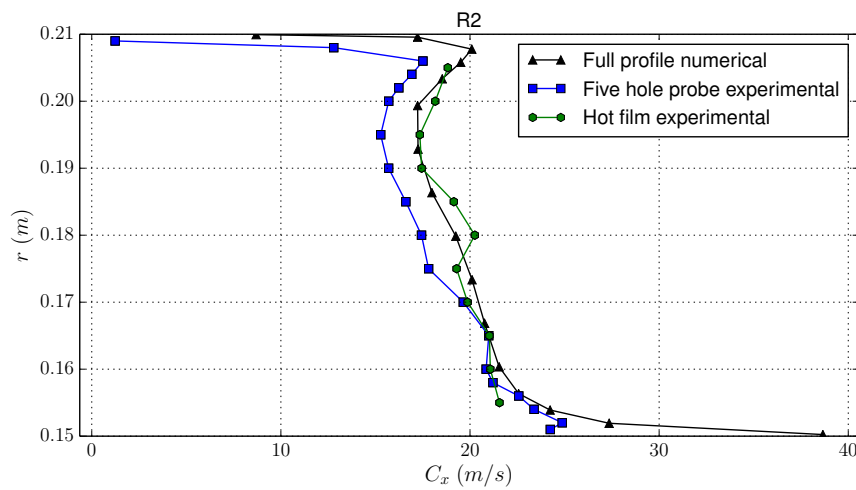
as expected. There is good agreement between the five-hole probe data and that of the numerical model for $0.15 \text{ m} \leq r \leq 0.195 \text{ m}$. For the region $0.195 \text{ m} \leq r \leq 0.210 \text{ m}$, there is divergence between the numerical results and five-hole probe experimental data, but agreement between the hot-film data and numerical results. The divergence of the two sources of experimental data may be symptomatic of an area of increased experimental error, possibly due to the effect of the shroud boundary layer, rotor tip vortices or the five-hole probe protruding into the flow stream (Brandsen (2013)).

At R2, there is close correspondence between the five-hole probe data, hot-film data and numerical data for $0.15 \text{ m} \leq r \leq 0.175 \text{ m}$. For $0.175 \text{ m} \leq r \leq 0.21 \text{ m}$ the numerical predictions lie between the measurements of the five-hole probe and hot-film sensor. In this range, the form of the numerical data closely corresponds to that of the five-hole probe data, despite a difference of approximately 2 m/s in magnitude.

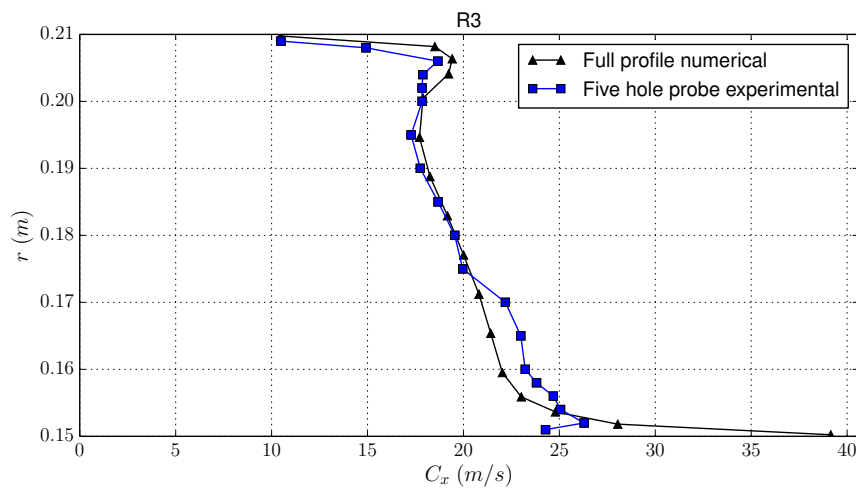
At R3, there is no hot-film data available. There is close correspondence between the five-hole probe data and the numerical data for $0.175 \text{ m} \leq r \leq 0.21 \text{ m}$, but these results deviate for $0.15 \text{ m} \leq r \leq 0.175 \text{ m}$. It is interesting that R2 and R3 results are respectively more and less accurate for radii greater than 0.175 m. When compared to the numerical data, it appears that the five-hole probe data is “shifted” left for R2 and “shifted” right for R3, possibly due to flow effects averaged away by the mixing-plane model for rotor-stator interaction or possibly due to experimental error. Both R2 and R3 show a pronounced step in velocity occurring at $r \approx 0.17 \text{ m}$. This flow feature is not captured by the numerical model. Unlike axial velocity, there is no easy way to bound the experimental error for tangential velocity. In the absence of bounds, it is reasonable to assume a similar magnitude of error as for axial velocity, as similar experimental techniques were used.



(a) R1 tangential velocity measurements



(b) R2 tangential velocity measurements



(c) R3 tangential velocity measurements

Figure 3.9: In this figure, the tangential velocity predictions of the profile model are plotted against the experimental data of Gill (2012).

Table 3.4: Computational cost of the various CFD models, using a single solver process. Notice that single stage models execute in approximate one third of the wall clock time.

Simulation	Runtime (s)
Single stage profile	549.33
Single stage massflow	677.40
Full profile	1869.63
Full massflow	2020.18

3.9 Comparing model run-times

In Table 3.4, a representative set of run-times for the various compressor compressor models is listed. All simulations were performed using a single process on a computer with an Intel® Core™ i7-4770K processor, 16 GB of DDR3 RAM and two 128 GB SSDs striped in RAID 0. The single stage simulations are approximately one third of the computational cost of the full models. Additionally, due to lower memory requirements, four single stage simulations can be undertaken in parallel on the given hardware, while only two full simulations can be run. These two factors mean that 12 single stage simulations can be executed in the same time as 2 full simulations on the hardware in question.

3.10 Conclusion

In this chapter, four CFD models have been examined: two variants in terms of boundary conditions (the mass flow and profile models) and two variants in terms of domain (single stage and full models). It is necessary to choose a single model for use in optimization. To assist in the selection process, it would have been of great assistance to have experimental data for another blade set, in order to establish the generalizability of the model to other geometries. Since no such data is available, accuracy will have to serve as a surrogate for generalizability.

It is tempting to choose a mass flow model, due to its simplicity and the direct control over the mass flow rate it provides. Conversely, the profile models offer reasonably accurate prediction of velocity profiles through the machine and appear to offer better predictions of pressure over the machine. Single stage models are desirable due to their much reduced execution time, but full models have increased accuracy. While it might be possible that a single stage approximation of a less accurate full model is more accurate due to the canceling of error terms, this is not a desirable sort of accuracy.

Since the full profile model appears to provide more accurate predictions of experimental results than the full mass flow model, the velocity boundary

conditions are chosen. The single stage domain is chosen as it appears to provide a reasonable approximation of the full model, at approximately one third of the computational cost (Table 3.4). Thus, the single profile model is used for future design tasks in this work.

It is difficult to determine whether the stalling mass flow of the single stage profile model accurately predicts the stalling mass flow of the experimental data in Figure 3.7, due to the lack of experimental data in the region of stall. The static simulations developed in this chapter are unable to capture inherently transient stall phenomenon, such as rotating stall. This inaccuracy can be tolerated, as all that is required for design purposes is preserving the numerically predicted stall margin.

In the next chapter, the focus of this work shifts from the development of an accurate model of the Rofanco compressor to the introduction of the necessary numerical design optimization techniques. In Chapter 6, the focus returns to the analysis of the optimized compressor blades developed in Chapter 5.

Chapter 4

Optimization and meta-modeling

4.1 Introduction

The preceding two chapters deal with the development of an accurate and computationally inexpensive model of the Rofanco compressor. In this chapter, the focus is shifted toward developing the numerical techniques necessary to produce an optimized blade design.

The use of statistical techniques to approximate the output of computationally expensive computer simulations has become commonplace in the context of numerical design optimization (Simpson *et al.* (2001)). These approximations, variously referred to as meta-models, surrogates and regressors in the engineering context, allow the processing of physical and numerical experimental results in ways that would not otherwise be possible. In this chapter, meta-model based design optimization (MBDO) techniques are introduced so that they may be applied to the compressor design problem in subsequent chapters.

A brief introduction to optimization, as well as specific background on MBDO is presented in Section 4.2. The problem of bootstrapping a meta-model by distributing an initial sample set in the design space, referred to as design of experiments, is discussed in Section 4.3. Robustly assessing the quality of fit of meta-models is discussed in Section 4.4. Section 4.5 and Section 4.6 discuss data transforms and meta-modeling techniques which are used to approximate the output of the simulations from Chapter 3. Section 4.7 brings the focus to efficient global optimization (EGO), a probabilistic optimization technique.

4.2 Background to optimization and meta-models

4.2.1 Introduction

In this section, the optimization process is introduced. The standard form of an optimization problem, along with methods of solving this problem, are discussed in Section 4.2.2. Section 4.2.3 introduces meta-modeling and discusses some reasons for applying meta-modeling techniques. Section 4.2.4 introduces some basic meta-modeling terminology.

4.2.2 An optimization problem in standard form

The standard formulation of an optimization problem is as follows. Given a vector of design variables, \mathbf{x} , hereafter referred to as the design vector, minimize some objective function $f(\mathbf{x})$, subject to a set of inequality constraints $\mathbf{g}(\mathbf{x}) \leq 0$ and equality constraints $\mathbf{h}(\mathbf{x}) = 0$, within some bounds $\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u$. This is expressed in mathematical notation in Equation 4.2.1.

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{Subject to: } & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0 \\ & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \end{aligned} \tag{4.2.1}$$

When solving an optimization problem it is desirable to find the best possible solution, with the smallest amount of computational work. In the context of typical engineering optimization, the performance of an optimization algorithm is measured in terms of the number of times the objective function must be evaluated, as a single function evaluation likely constitutes an expensive numerical simulation.

Generally speaking, methods for solving optimization problems can be divided into two groups: gradient based methods and gradient free methods. Typically, gradient based methods involve using the gradient to find a search direction expected to minimize the objective function, as well as an optimal step size to travel in that direction. Gradient based methods can be highly efficient if the problem is convex over the set of allowable inputs. Gradient based methods require the gradient of the objective and constraint functions to be smooth and can become trapped in local minima when applied to multimodal functions. An introduction to gradient based methods is available in Snyman (2005).

By contrast, gradient free methods do not explicitly form a gradient and thus do not rely on the gradient being smooth, or even existent. Typically gradient free methods are less computationally efficient than gradient based

methods, but are much more robust and can better deal with problems of a highly multi-modal nature. In this work, the gradient free methods used include particle swarm optimization (PSO) and genetic algorithms (GA). For a review of gradient free methods, see Rios and Sahinidis (2013).

4.2.3 Reasons for using meta-models

Despite Moore's Law providing an exponential growth in the computational power afforded to engineers over the last forty years (Schaller (1997)), computational cost still remains a major limitation in numerical design optimization. This is attributed to the rapid increase in fidelity and complexity of simulation codes as well as the human factors that dictate practical simulation run times (Venkataraman and Haftka (2004), Koch *et al.* (1999)). This observation is not unique to engineering simulations. In the field of computer graphics, it is referred to as Blinn's Law: the observation that any increase in the available amount of computational power will result in the practitioner increasing the complexity of the model so as to negate that increase (Andrade *et al.* (2007)).

The continual consumption of the available computation power by the simulation component of design suggests that an alternative approach is required to deal with the computational cost of numerical design optimization. One such approach is meta-model based design optimization (MBDO). Instead of applying numerical optimization algorithms directly to the objective function, MBDO constructs an approximation to the objective function or "meta-model". This meta-model can be sampled cheaply and thus optimized cheaply. Meta-modeling is a multi-disciplinary exercise, incorporating advances from the fields of statistics, computer science and engineering.

Aside from a potential reduction of computational cost, MBDO offers several other advantages. Meta-modeling allows the decoupling of the simulation steps of the optimization process from the minimization steps, reducing the complexity of the software engineering problem and making the process more robust to a single simulation failure (Simpson *et al.* (2001)). The global approximation provided by meta-models is useful for rapid exploration of the design space (Simpson *et al.* (2001)). Since the bulk of the computational expense in the meta-modeling process is in generating the initial training set, much of the computational expense can be performed in a data-parallel manner, at the start of the process (Wang and Shan (2007)). Meta-models also filter noise and discontinuity from the dataset, as they can provide smooth and continuous approximations to a noisy dataset (Wang and Shan (2007)).

4.2.4 Basic meta-modeling terminology

At this stage, it is necessary to introduce some meta-modeling terminology. Given an unknown function $y = f(\mathbf{x})$, sampled at n points (\mathbf{x}_i, y_i) , let $\hat{y}(\mathbf{x})$ with estimated standard deviation $\hat{s}(\mathbf{x})$ be the meta-model approximation to

that function. The vector $\mathbf{x} = [x_1 \dots x_p]$ is referred to as the design vector and x_i is referred to as a model feature. The model error is the difference between the predicted and actual values i.e. $e_i = y_i - \hat{y}_i$.

If the values y_i are restricted to some discrete set, the problem is a classification problem, while if y_i is free to assume a continuum of values, then the problem is a regression problem. In a classification problem, each discrete y_i value is referred to as a class label. Only binary classification is used in this work and by convention class labels of “-1” and “1” are used.

4.3 Distributing samples in the design space

When creating a meta-model, the question of how to distribute sample points in the design space arises. This problem is referred to as design of experiments and is widely studied for both numerical and physical experiments. Typically computer experiments are deterministic; repeating a computer experiment gives the same result. In contrast, physical experiments contain a component of random measurement error. The presence of random measurement error means that physical experiments require different experimental designs to those of numerical experiments (Sacks *et al.* (1989)).

The latin hypercube, originally proposed by McKay *et al.* (1979), is a commonly used experimental design for computer experiments. It is a form of stratified sampling; the design space is distributed into levels and only one sample is allowed per level. Two latin hypercubes are shown in Figures 4.1a and 4.1b. Despite being a valid latin hypercube, Figure 4.1a is a poor experimental design, as all samples are collinear and the data provided to the meta-model are effectively one-dimensional. Conversely, a design such as that shown in Figure 4.1b, where samples are evenly distributed, will result in a better model fit.

The space-filling properties of a latin hypercube can be improved by incorporating the additional condition that the minimum distance between samples be maximized. Latin hypercubes that incorporate this additional condition are referred to as optimal latin hypercubes. The latin hypercube shown in Figure 4.1b is in fact an optimal latin hypercube.

The problem of generating an optimal latin hypercube in a high dimensional space is difficult, as evaluating a candidate latin hypercube requires a computationally expensive nearest-neighbors search. In this work, optimal latin hypercubes are generated using a genetic algorithm described in Appendix B.

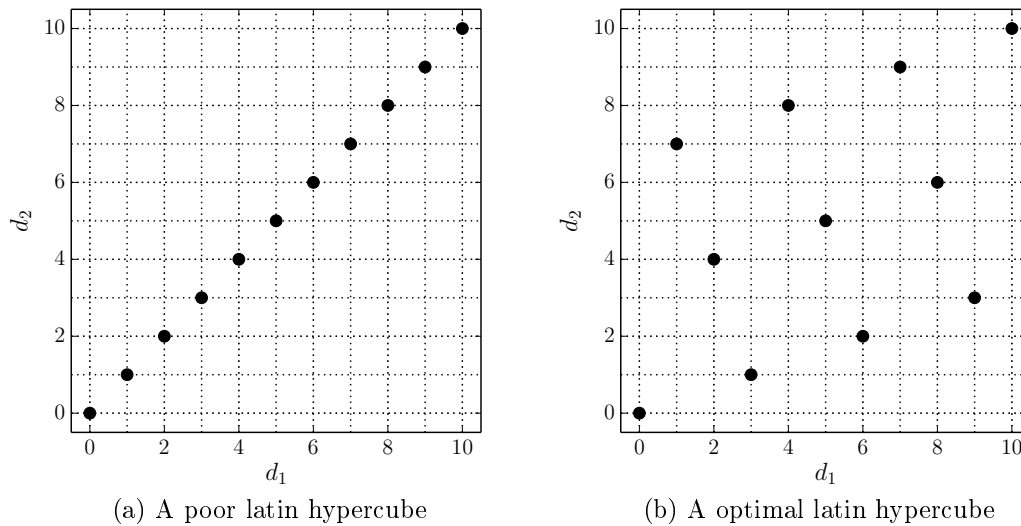


Figure 4.1: In this figure two latin hypercubes are displayed. Each has eleven levels and two dimensions. Figure 4.1a is an example of a poor latin hypercube, as information is provided in one dimension only. Figure 4.1b is an optimal latin hypercube, as the minimum distance between samples is maximized.

4.4 Evaluating meta-model error

4.4.1 Introduction

In this section, evaluating the error associated with a particular meta-model is discussed. Cross-validation, a technique for obtaining robust error estimates without a separate test set, is introduced in Section 4.4.2 and various error norms are discussed in Section 4.4.3.

4.4.2 Cross-validation

Many meta-models require the choice of model parameters, referred to as model hyper-parameters. Commonly, these hyper-parameters are chosen so as to minimize the estimated model error. It is tempting to compute model error by comparing the predicted and actual values at training points. This practice is hazardous when applied to models with a capacity to fit arbitrary data, as it can lead to over-fitting and a loss of generalizability. Ideally an independent set of data are used to evaluate model error, but often this is not available, due to the cost of generating training samples.

Cross-validation is a technique for estimating model error without the need for an independent testing data set. It involves iterative partitioning of data into two subsets: the training subset, which is used to train a candidate model, and the testing subset, which is used to evaluate the candidate model.

There are many different forms of cross validation, which use different data partitioning strategies (Kohavi *et al.* (1995)). Leave-one-out (LOO) cross-validation involves using each sample in turn as the testing subset, while the other samples are used as the training subset. Although LOO is a computationally expensive form of cross-validation it is used exclusively in this work, since the cost of generating training samples dwarfs the cost of performing LOO cross-validation. In subsequent sections the notation e^{-i} is used to denote the error associated with predicting the value of the i -th sample using LOO cross-validation.

4.4.3 Measurement of error

In order to compare two vectors of cross validation error, it is necessary to choose a norm to compute their magnitudes. Two commonly encountered measurements of error are the root-mean-square error (RMSE) and maximum absolute error (MAE). They are defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (e^{-i})^2} \quad (4.4.1)$$

$$MAE = \max \{e^{-i}\} \quad (4.4.2)$$

The R^2 coefficient is a measure of the variance explained by the model, relative to the variance in the data (Myers *et al.* (2009)). As such, higher values of R^2 indicate a more accurate model. An R^2 value of near one, indicates that the model explains most of the variance present in the model, while an R^2 value of near zero indicates that the model explains very little of the variance in the model. R^2 is defined in terms of the total sum-of-squares (SS_{total}) and residual sum-of-squares (SS_{res}):

$$SS_{total} = \sum_{i=1}^n (y_i - \bar{y}_i)^2 \quad (4.4.3)$$

$$SS_{res} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.4.4)$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{total}} \quad (4.4.5)$$

For a discussion of the R^2 coefficient in terms of response surfaces, see Appendix B.

4.5 Data transforms

4.5.1 Introduction

Often the accuracy with which a given dataset can be approximated is improved by applying some sort of transformation to the data before beginning the fitting process. These transformations take the form $\mathbf{x}' = f(\mathbf{x})$ and are usually invertible and continuous in the domain being considered. In this section, three common data transforms used in this work are discussed: normalization of model features, the Box-Cox family of power transforms and principle component analysis.

4.5.2 Normalizing features

Normalizing of model features is both a good practice and a requirement for meta-modeling techniques which are not scale invariant, such as support vector regression (discussed in Section 4.6.4). Typically, data are normalized by removing the mean ($\bar{\mathbf{x}}$) and scaling to unit variance ($\sigma = 1$):

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sigma^2} \quad (4.5.1)$$

4.5.3 Box-Cox transform

A wide range of transformations have been proposed in the context of response surface methods (Sakia (1992)). These transforms give response surface models, which are usually quadratic, the ability to accurately approximate a wider range of data. One of the most useful of these transforms is the range of power transforms proposed by Box and Wilson (1951), which has come to be referred to as the Box-Cox transform:

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda \dot{y}^{\lambda-1}} & \lambda \neq 0 \\ \dot{y} \log y_i & \lambda = 0 \end{cases} \quad (4.5.2)$$

Note that $\log y_i$ is the natural logarithm of y_i and that \dot{y} implies the geometric mean of all y_i . As λ is varied a range of different transforms are applied to the input data, making the Box-Cox transform very flexible.

4.5.4 Principle component analysis

Principle component analysis (PCA), variously referred to as proper orthogonal decomposition, singular value decomposition and the Karhunen-Loeve transform in different fields, is an orthogonal linear transform that transforms data into a new coordinate system where the first dimension has the greatest variance, the second dimension has the second greatest variance and so on (Jolliffe (2005)). Figure 4.2 shows PCA applied to a two-dimensional example.

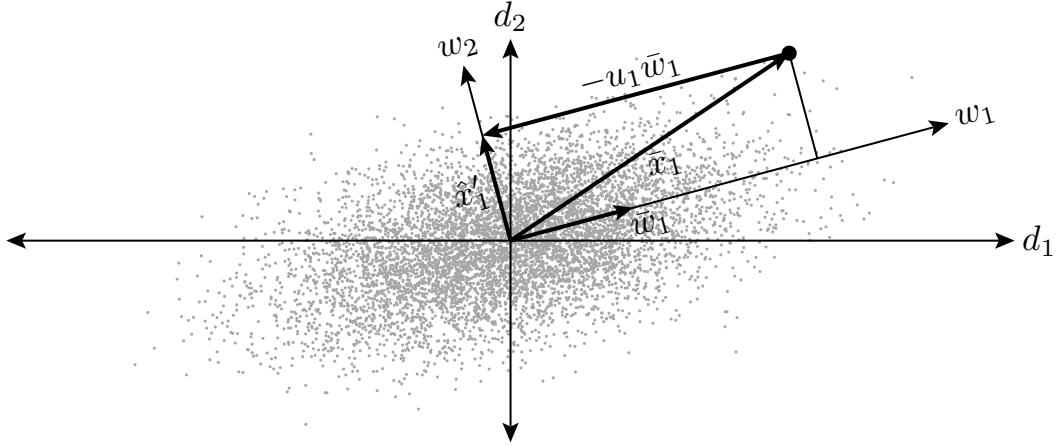


Figure 4.2: In this figure, principal component analysis is applied to a two dimensional dataset. Principal component analysis involves deriving a new orthogonal set of dimensions whereby each dimension seeks to maximize the variance of the data it explains.

Consider the problem of finding a new dimension, \mathbf{w} , formed out of a weighted linear combination of the old dimensions, \mathbf{x} , so that the variance of u_i , the component of \mathbf{x}_i in the direction \mathbf{w}_j is maximized i.e $u_i = \mathbf{x}_i \mathbf{w}_j^T$. If \mathbf{u} and X are defined by $\mathbf{u} = [u_1 \dots u_n]$ and $X = [\mathbf{x}_1 \dots \mathbf{x}_n]$ then this problem may be stated as given $\mathbf{u} = X\mathbf{w}$ find some \mathbf{w} such that $\text{Var}(\mathbf{u})$ is maximized, subject to $\|\mathbf{w}\| = 1$. Assuming centered data ($\mu = 0$), the variance of \mathbf{u} can be expressed in terms of a Rayleigh quotient:

$$\text{Var}(\mathbf{u}) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 = \frac{1}{n} \mathbf{u}^T \mathbf{u} = \frac{1}{n} \mathbf{w}^T X^T X \mathbf{w} = \frac{\mathbf{w}^T X^T X \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \quad (4.5.3)$$

Since this quantity is a Rayleigh quotient, it reaches a maximum value of the largest eigenvalue of $X^T X$, λ_{\max} , when $\mathbf{w} = \boldsymbol{\nu}_{\max}$, the eigenvector associated with λ_{\max} . Subsequent dimensions can be found by subtracting the component of the data in direction \mathbf{w} :

$$\hat{X} = X - \mathbf{u}\mathbf{w}^T = X - X\mathbf{w}\mathbf{w}^T \quad (4.5.4)$$

By repeating the process of Equation 4.5.3, the PCA transform can be defined, where $X = [\mathbf{x}_1 \dots \mathbf{x}_n]^T$ and $W = [[\boldsymbol{\nu}_1 \dots \boldsymbol{\nu}_n]]$:

$$X' = XW \quad (4.5.5)$$

In this work, principle component analysis is used as an optional preprocessing treatment to meta-modeling data. For further details regarding PCA, the reader is referred to Jolliffe (2005).

4.6 Common meta-modeling techniques

4.6.1 Introduction

In this section, three types of meta-models commonly used in MBDO are discussed. Section 4.6.2 is a discussion of polynomial response surfaces, while Section 4.6.3 deals with kriging and Section 4.6.4 introduces support vector machines.

4.6.2 Polynomial response surface methods

The use of polynomial response surfaces can be traced to the early work of Box and Wilson (1951). Reviews of the progress of the field are found in Myers *et al.* (1989), Mead and Pike (1975), Hill and Hunter (1966) and Myers *et al.* (2004). For a complete introduction to polynomial response surfaces, see Myers *et al.* (2009).

Although originally used to model physical experiments (Box and Draper (1987)), response surfaces have become widely used to model the results of numerical experiments. Despite widespread use, there is some debate about the applicability and usefulness of polynomial response surfaces to deterministic computer experiments, due to their limited capacity to model highly non-linear functions (Sacks *et al.* (1989), Welch *et al.* (1990)). While this means that response surfaces are necessarily inexact when modeling complex functions, it does make them very resistant to overfitting, provided a low order polynomial is used.

A second order response surface model with cross terms is of the form:

$$\hat{y} = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{j=2}^k \sum_{i=1}^{j-1} \beta_{ij} x_i x_j = \beta_0 + \boldsymbol{\beta}_1^T \mathbf{x} + \mathbf{x}^T \boldsymbol{\beta}_2 \mathbf{x} \quad (4.6.1)$$

where $\boldsymbol{\beta}_1$ is a $k \times 1$ vector and $\boldsymbol{\beta}_2$ is a $k \times k$ upper triangular matrix. A full second order model thus possesses $n = \frac{1}{2}k^2 + \frac{3}{2}k + 1$ terms. Let $\boldsymbol{\beta}$ and \mathbf{X} be defined as follows:

$$\boldsymbol{\beta} = [\beta_0 \quad \beta_1 \dots \beta_n \quad \beta_{ij}^{i \neq j} \quad \beta_{11} \dots \beta_{nn}] \quad (4.6.2)$$

$$\mathbf{X} = [\mathbf{1} \quad \mathbf{x}_0 \dots \mathbf{x}_n \quad \mathbf{x}_i \mathbf{x}_j^{i \neq j} \quad \mathbf{x}_{11}^2 \dots \mathbf{x}_{nn}^2] \quad (4.6.3)$$

Given a set of training data \mathbf{x} and \mathbf{y} it is possible to compute optimal values for $\boldsymbol{\beta}$ using least squares. The two-norm of the difference between the predicted and actual value at a given data point is given by:

$$E = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \mathbf{y}^T \mathbf{y} - 2\boldsymbol{\beta}^T \mathbf{X}^T \mathbf{y} + \boldsymbol{\beta}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\beta} \quad (4.6.4)$$

Setting the derivative with respect to $\boldsymbol{\beta}$ equal to zero and rearranging results in a solution for $\boldsymbol{\beta}$ which minimizes E , yields:

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (4.6.5)$$

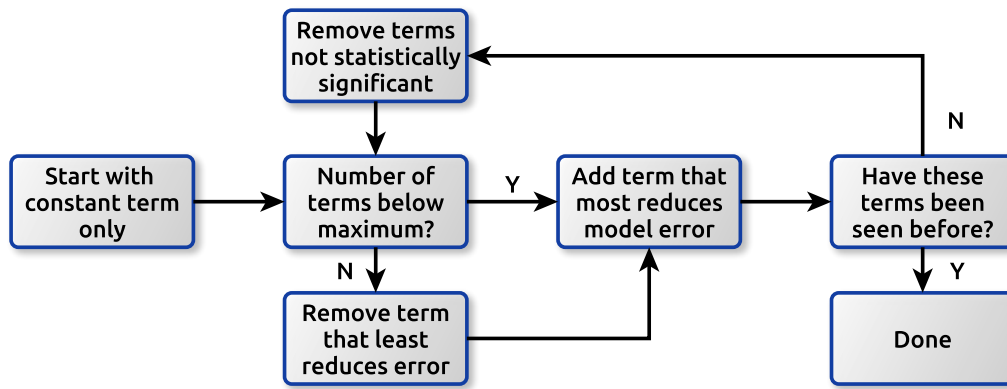


Figure 4.3: The stepwise forward regression algorithm used to determine the polynomial terms for response surface methodology. The search for terms is terminated when a cycle is detected, at which point the polynomial with the lowest R^2 is selected.

This approach is referred to as “solving the normal equations”. In practice, least-squares systems should not be solved in this way, as computing $(\mathbf{X}^T \mathbf{X})^{-1}$ is inefficient and poorly-conditioned. Least squares systems can be efficiently solved using QR, SV or Cholesky decomposition or by iterative schemes, such as the conjugate gradient method.

A large amount of data is required to fit a full second order model, especially since it is good practice to include at least $\frac{3}{2}n$ training samples, to minimize the risk of overfitting. Furthermore, it is possible that many of the terms in the full quadratic model are insignificant. This suggests fitting an incomplete polynomial, yet which terms to exclude is unclear. This problem is solved using stepwise forward regression, an iterative scheme in which terms are added and removed from a candidate model, based on their statistical significance and contribution to R^2 . An overview of the stepwise forward regression algorithm used in this work is shown in Figure 4.3.

4.6.3 Kriging

Kriging, also known as Gaussian processes, is a technique developed in the field of geostatistics by Krige (1951), in order to evaluate gold ore reserves in the Witwatersrand using a limited number of samples. Kriging was further developed and formalized by Matheron (1963). Subsequently, kriging has become a popular technique in the field of meta-modeling and has seen widespread use (Simpson *et al.* (1998), Simpson *et al.* (2001), Martin and Simpson (2004), Hoyle (2006), Forsberg and Nilsson (2005)).

The basic kriging model is given by:

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^k \beta_j f_j(\mathbf{x}) + Z(\mathbf{x}) \quad (4.6.6)$$

This equation consists of two components: the first component is a linear regression which models the variation in the mean. The functions f_j are referred to as the regression functions, while β_j are the weights assigned to those functions. The second component consists of a stationary Gaussian process with zero mean and a covariance given by:

$$\text{cov}(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 \mathcal{R}(\theta, \mathbf{x}_1, \mathbf{x}_2) \quad (4.6.7)$$

$\mathcal{R}(\theta, \mathbf{x}_1, \mathbf{x}_2)$ is referred to as the correlation function, σ^2 is the model variance and θ is a model parameter that controls the influence of nearby points. The correlation function controls several important model properties, such as the smoothness of the model, the influence of nearby points and the differentiability of the response (Martin and Simpson (2004)). When $\mathcal{R}(\theta, \mathbf{x}_1, \mathbf{x}_2) = 1$ when $\mathbf{x}_1 = \mathbf{x}_2$ and zero otherwise, kriging reduces to ordinary least squares. A correlation function of this sort is referred to as a “nugget model”. For more on the similarities and differences between kriging and conventional polynomial regression, see Kleijnen (2009).

In practice, the regression models f_i and correlation model $\mathcal{R}(\mathbf{x})$ are not known in advance. Instead they are chosen from a set of models shown to have good empirical performance. Kriging regression and correlation models are discussed in more detail in Appendix B.

The method of maximum likelihood can be used to estimate the model parameters. Given a particular set of data and a particular model for that data, the method of maximum likelihood seeks to find the parameters (θ , σ and β) that make the observed results (\mathbf{y}) most probable. The principles of maximum likelihood is not discussed in detail; for more information see Edwards (1974). Since the expression for likelihood, $\mathcal{L}(\theta, \sigma, \beta | \mathbf{y})$, is rather complicated it is more common to work with the log-likelihood function, $\ell(\theta, \sigma, \beta | \mathbf{y}) = \log(\mathcal{L}(\theta, \sigma, \beta | \mathbf{y}))$ (Martin and Simpson (2004)):

$$\begin{aligned} \ell(\theta, \sigma, \beta | \mathbf{y}) = & -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2} \log \|\mathbf{R}\| \\ & - \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{F}\beta)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\beta) \end{aligned} \quad (4.6.8)$$

The matrix \mathbf{R} is a m by m matrix, where m is the number of observations and \mathbf{F} is m by k matrix. The components of these matrices are given by:

$$F_{i,j} = f_j(\mathbf{x}_i) \quad (4.6.9)$$

$$R_{i,j} = \mathcal{R}(\theta, \mathbf{x}_i, \mathbf{x}_j) \quad (4.6.10)$$

If \mathbf{R} is not a function of θ , it is possible to generate analytic expressions for β and σ^2 by setting the partial derivative of Equation 4.6.8 with respect to β and σ^2 to zero (Martin and Simpson (2004)):

$$\beta = (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{y}) \quad (4.6.11)$$

$$\sigma^2 = \frac{1}{n} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) \quad (4.6.12)$$

Now, if \mathbf{R} is a function of θ , there is no analytical solution for $\boldsymbol{\beta}$ and σ^2 . For a given θ value, $\boldsymbol{\beta}$ and σ^2 can be inferred from the analytical solutions in Equations 4.6.12 and 4.6.11 when assessing Equation 4.6.8. Often, an optimization algorithm such as COBYLA, developed by Powell (1994), is used to search for the θ value that maximizes Equation 4.6.8. Since evaluating this function is computationally inexpensive when compared to generating training samples, in this work θ is chosen by testing a range of values. Similarly, regression and correlation models are chosen from those in Appendix B.3 by fitting all combinations and using the pair that best approximates the data..

For the purposes of this work, the foremost advantage of kriging meta-models are that they provide estimates of both the function mean and variance. In Section 4.7, the availability of variance estimates will be exploited in order to develop a scheme to update meta-models with new samples that have a high probability of improving the current optimum.

4.6.4 Support vector machines

Support vector machines (SVMs) are a powerful tool for regression or classification. Initially proposed by Cortes and Vapnik (1995), support vector machines operate by non-linearly mapping input data into another space, where linear classification or regression takes place. Originally intended for classification, SVMs were generalized for regression by Drucker *et al.* (1997). Both support vector classification (SVC) and support vector regression (SVR) are used in this work. Since the derivations of SVC and SVR are similar, only SVR is discussed here. For an explanation of SVC, see Gunn (1998).

Initially, consider the problem of fitting a linear function $f = \mathbf{w} \cdot \mathbf{x} + b$, to data y_i with an allowable error ϵ , as shown in Figure 4.4. Since it is possible that there exists no function capable of meeting this error constraint, it is necessary to introduce the slack variables ξ_i^u and ξ_i^l when formulating the optimization problem. Furthermore, it is desired that the function be as smooth as possible, i.e. w^2 is minimized. This problem may thus be stated as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i^u + \xi_i^l) \\ \text{Subject to} \quad & y_i - f(x_i) - \epsilon - \xi_i^l \leq 0 \\ & f(x_i) + y_i - \epsilon - \xi_i^u \leq 0 \\ & -\xi_i^l \leq 0 \\ & -\xi_i^u \leq 0 \end{aligned} \quad (4.6.13)$$

The constant C controls the trade-off between error and smoothness. From

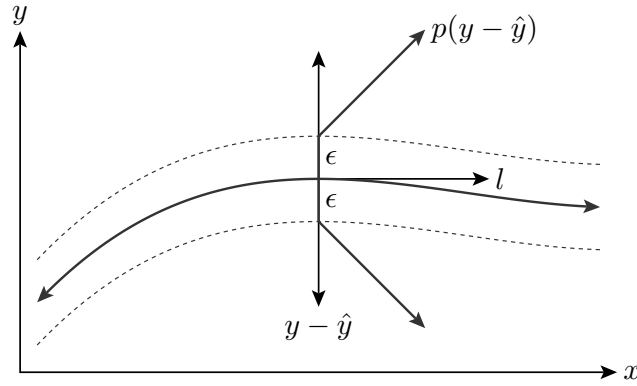


Figure 4.4: This figure illustrates the soft margin loss function. Note that $p(y - \hat{y}) = 0$ when $y_i - \epsilon \leq \hat{y}(\mathbf{x}_i) \leq y_i + \epsilon$

this primal problem, it is possible to construct an equivalent Lagrangian dual problem (Smola and Schölkopf (2004)).

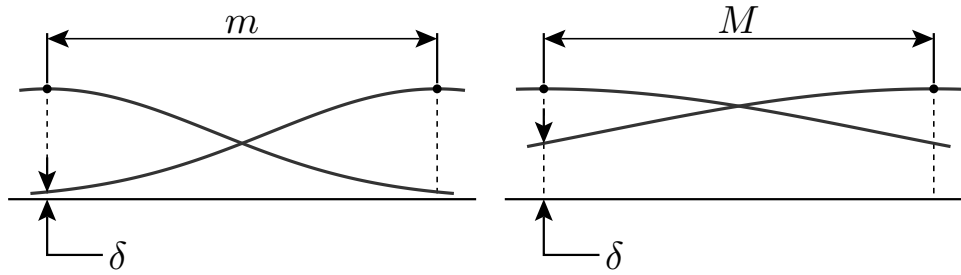
$$\begin{aligned} \max \quad & -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) \mathbf{x}_i \cdot \mathbf{x}_j - \\ & \epsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i + \alpha_i^*) \end{aligned} \quad (4.6.14)$$

$$\begin{aligned} \text{Subject to:} \quad & \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ & \alpha_i, \alpha_i^* \in [0, C] \end{aligned}$$

It may also be shown that $f(x)$ can be written in terms of Lagrange multipliers, to obtain what is commonly referred to as the support-vector expansion (Smola and Schölkopf (2004)):

$$f(x) = w \mathbf{x}_i \cdot \mathbf{x} + b = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \mathbf{x}_i \cdot \mathbf{x} + b \quad (4.6.15)$$

Equations 4.6.14 and 4.6.15 allow several important observations to be made. The first is that \mathbf{w} is only dependent on the difference between the dual variables α and α^* . For points where the error is less than the soft margin this difference is zero. In other words $-\epsilon \leq f(x_i) - y_i \leq \epsilon$ implies $\alpha_i = \alpha_i^*$. Thus, only points outside of the soft margin contribute towards the regressor. These points are the “support vectors” to which support vector regression refers. Secondly, notice that Equation 4.6.14 is a convex quadratic programming problem, for which advanced solvers are widely available. Finally, note that both the regressor and training problem are defined in terms of the dot product. This is significant when introducing non-linearity into the problem.



(a) Minimum kernel radius of significance (b) Maximum kernel radius of significance

Figure 4.5: A kernel with a value less than δ at the minimum distance between samples does not contribute significantly to the decision function. Similarly, a kernel with a value of greater than $1 - \delta$ at the maximum distance between samples may be replaced with a constant term with minimal loss in accuracy.

By replacing the dot product $\mathbf{x}_i \cdot \mathbf{x}$, with a kernel function $k(\mathbf{x}_i, \mathbf{x})$, it is possible to model non-linear functions using SVR. This process may be thought of as non-linearly mapping the input vectors into another space, applying the dot product in that space and then applying the inverse mapping. Instead of providing a function for the mapping, a kernel function is provided to account for the whole process.

There are many possible choices of valid support vector machine kernels, such as the polynomial kernel, sigmoid kernel and Gaussian kernel. Of these, the Gaussian kernel is the most commonly used and thus used in this work (Scholkopf and Smola (2001), Lin *et al.* (2003)). The Gaussian kernel is given by:

$$k(\mathbf{x}_1, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}\|^2}{2\sigma^2}\right) = \exp(-\gamma\|\mathbf{x}_1 - \mathbf{x}\|^2) \quad (4.6.16)$$

It would be possible to train multiple SVM predictors for a given dataset, each using different kernels. Because three different varieties of meta-models, combined with three different data transforms are already being used, this is deemed unnecessary.

The Gaussian kernel requires the choice of kernel parameter γ . Furthermore, the training process requires the choice of the error penalty parameter C . In this work, a simple grid search is used to find appropriate C and γ values as the cost of a training a candidate model is negligible compared to the cost of generating training samples.

It is necessary to determine levels and bounds for this grid search; an often used approach is to use levels of 2^n for $n \in [-8, 8]$ (Chang and Lin (2011)). By considering the scaling of the particular problem, a much narrower set of bounds can be deduced. First define a minimum objective function value of significance, δ . It would be sensible to choose $\delta = \epsilon$, but this does not have to be the case. Also, let the maximum distance between pairs of samples in the domain be M and the minimum distance be m .

Consider the case of the closest two samples in the domain. A particular kernel only significantly contributes to the decision function (Equation 4.6.15) if it exceeds δ when $\|\mathbf{x}_1 - \mathbf{x}\| = m$, provided C is not excessively large. Using this knowledge, it is possible to derive an upper bound for γ :

$$\delta \leq \exp(-\gamma m) \quad (4.6.17)$$

$$\gamma \leq -\frac{1}{m} \ln \delta \quad (4.6.18)$$

Similarly, any kernel that is greater than $1 - \delta$ when $\|\mathbf{x}_1 - \mathbf{x}\| = M$ can be replaced by the constant term in the decision function (Equation 4.6.15), provided C is not excessively small. This observation enables a lower bound for γ to be specified:

$$1 - \delta \geq \exp(-\gamma M) \quad (4.6.19)$$

$$-\frac{1}{M} \ln(1 - \delta) \leq \gamma \quad (4.6.20)$$

A key assumption of Equations 4.6.17 and 4.6.19 is that any point that does not make a contribution to reducing cross-validation error is negligible. Since cross-validation is used to evaluate the quality of candidate meta-models in this work, this assumption is reasonable. Depending on the distribution of samples, it may be sensible to choose m and M differently. If some pairs of samples are radically further apart or closer together than the mean distance between samples, then it may be worth excluding them.

4.7 Efficient global optimization

4.7.1 Introduction

In the previous section, three types of meta-models were introduced. These meta-models serve as surrogates for the computationally expensive simulations needed to evaluate the performance of a candidate set of compressor blades. It is possible to apply standard optimization algorithms directly to these meta-models to generate an improved blade design. Since the areas of the design space yielding optimal or near optimal solutions are not known a priori, this would require dense sampling over the entire domain.

A better approach is an iterative update scheme, such as efficient global optimization (Jones *et al.* (1998)). EGO bootstraps a meta-model using an initial set of samples, often generated via latin hypercube sampling. New samples are subsequently added, by optimizing various infill criteria, derived from estimates of meta-model mean and variance. At its core EGO, is a reinforcement learning problem: an agent, in this case the optimizer, takes action, in this case selecting infill samples, to maximize reward, in this case the objective function. Inherently, such problems require a trade off between

exploration of unsampled areas and exploitation of problem knowledge and are thus termed exploration-exploitation problems (Kaelbling *et al.* (1996)). In the remainder of this section, the process of applying EGO to a practical optimization problem is discussed.

4.7.2 Infill criteria

Probably the most widely used infill criteria is expected improvement, which is an estimate of the probability that a particular sample will improve on the model optimum. The improvement of the current point Y , over the current optima, y^* , is given by $I = \max\{0, y^* - Y\}$. Jones *et al.* (1998) show that given this definition of I , expected improvement is given by Equation 4.7.1. Note that $\Phi(x)$ denotes the normal cumulative density function, while $\phi(x)$ denotes the normal probability density function.

$$E(I) = \begin{cases} (f_{min} - \hat{y}) \Phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right) + \hat{s} \phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right) & \hat{s} \neq 0 \\ 0 & \hat{s} = 0 \end{cases} \quad (4.7.1)$$

Sasena *et al.* (2002) gives a comparison of several other infill criteria, such as the maximum variance criterion, where samples are placed in areas of high variance, in order to decrease the uncertainty of the model predictions. For further discussion of the infill criteria used in this work, see Appendix B.

4.7.3 Generating multiple infill samples per iteration

The samples generated by optimizing an infill criterion are referred to as infill samples. In this section, the algorithm used to generate infill samples is described. As originally proposed by Jones *et al.* (1998), EGO only adds a single point during the infill step. This approach is parsimonious in terms of functional evaluations, but does not make efficient use of the massive amount of parallel computing power available. Several authors have explored different avenues for generating multiple infill samples per EGO iteration (Chaudhuri *et al.* (2012), Henkenjohann and Kunert (2007), Ponweiser *et al.* (2008), Ginsbourger *et al.* (2007), Viana *et al.* (2010)).

A straight forward way to generate multiple infill samples per iteration is to use multiple infill criteria. For example, by applying both maximum variance and expected improvement infill criteria, a balance is struck between improving the model as a whole and finding samples with a high likelihood of improving the optimum. The relative numbers of maximum variance and expected improvement samples might be adjusted depending on the estimated accuracy of the model. The optima may also be added as an infill sample, in order to validate predicted performance.

Multiple infill samples can be generated from a single infill criterion by iteratively optimizing this criteria, provided there is some way for later infill

samples to account for the effect of earlier samples. One way to achieve this is to introduce a minimum radius between infill samples in order to prevent them from clumping together. This can be accomplished using a so called exclusion radius ($r_{exclusion}$); a minimum radius between infill samples, included as a constraint on the infill optimization problem (Chaudhuri *et al.* (2012)). The approach taken in this work accounts for the fact that most infill criteria are dependent on both estimates of sample mean, as well as the variance of those estimates. When a new infill sample (\mathbf{x}_{infill}) is found, the objective model (\hat{f}) is updated by adding a new data point, with $y_{infill} = \hat{f}(\mathbf{x}_{infill})$. In this way, the infill model's estimate of sample mean is not affected, but the estimate of the variance is updated to account for the new infill sample. This implies that subsequent infill samples can account for the change in variance caused by previous infill samples.

4.7.4 Optimizing infill criteria

By definition, model variance is zero at points that have been sampled and non-zero between them. If a sample space has d dimensions and contains n samples, and it is assumed that there is one optima in variance between each sample and its $2d$ neighbors, then there are $O(nd)$ local optima in variance. Even for moderately sized problems, this corresponds to a large number of optima.

Aside from the highly multi-modal nature of meta-model variance upon which most infill criteria are based, the infill optimization problem is further complicated by the fact that infill criteria such as expected improvement are relatively flat near sampled points.

As the infill optimization problem is both highly multi-modal, as well as flat near sampled points, a gradient based optimizer is likely to be unsuitable. Accordingly, a PSO optimizer, described in Appendix B, is used. PSO has no requirement that constraint gradients be continuous when optimizing infill, so constraints on the infill optimization problem are simply handled by masking the infill criteria to zero in areas in which constraints are violated.

4.7.5 EGO based convergence criteria

The concept of expected improvement also provides a convenient way to judge whether or not to terminate the optimization process, as it provides a measure of how likely the current optima can be improved upon. Jones *et al.* (1998) recommend that the optimization process be terminated when there is less than 1% chance that the next iteration will improve upon the current optima i.e. $\max E(I) < 0.01$. Relying on expected improvement in this way presumes the underlying model of sample mean and variance are of sufficient fidelity to make estimates of expected improvement accurate. Accordingly, another convergence criteria can be added: meta-models must be of sufficient accuracy,

Table 4.1: Data transform recipes

Recipe Name	Description
<code>scaled</code>	Feature normalization, as given by Equation 4.5.1
<code>powerscaler{N}</code>	The Box-Cox transform, as given by Equation 4.5.2. The parameter λ is given by <code>{N}</code> .
<code>pca</code>	Principle component analysis, as outlined in Section 4.5.4.

Table 4.2: Meta-model recipes

Recipe Name	Description
<code>rs</code>	A full second-order response surface, with coefficients determined via stepwise forward regression
<code>quad-rs</code>	A second-order response surface with no cross-terms
<code>kriging</code>	A kriging meta-model, as outlined in Section 4.6.3.
<code>svr</code>	A support-vector regressor, as outlined in Section 4.6.4.
<code>svc</code>	A support-vector classifier, as outlined in Section 4.6.4.

assessed using the criteria described in Section 4.4.3. Finally, for practical reasons, it is useful to impose a minimum and maximum number of iterations. The choice of these values is dually dependent on the problem being solved and the resources allocated to solving the problem.

4.7.6 Using meta-models as part of the EGO process

Before explaining the optimization process itself, it is necessary to understand how the data transforms of Section 4.5 and meta-models of Section 4.6 are combined to approximate the simulations of Chapter 3. The Python library `metamod`, developed as part of this work, provides convenience functions to implement the transforms and meta-modeling libraries of Section 4.5 and Section 4.6. This library introduces the concept of “recipes”. These recipes are simple text definitions of data transforms and meta-models to be applied to a data set. The data transforms and meta-models used in this work are outlined in Table 4.1 and 4.2 respectively. The recipe `scaled pca quad-rs` indicates that the data is first normalized, whereafter principle component analysis is applied and finally, a second order response surface without cross terms is fit to the data. An abstraction is provided in the library to fit an arbitrary number of recipes to a given dataset and retrieve the most accurate model recipe.

4.7.7 The EGO optimization process

The optimization process is illustrated in Figure 4.6. Initially, an optimal latin hypercube is used to distribute samples in the computational domain. Meta-

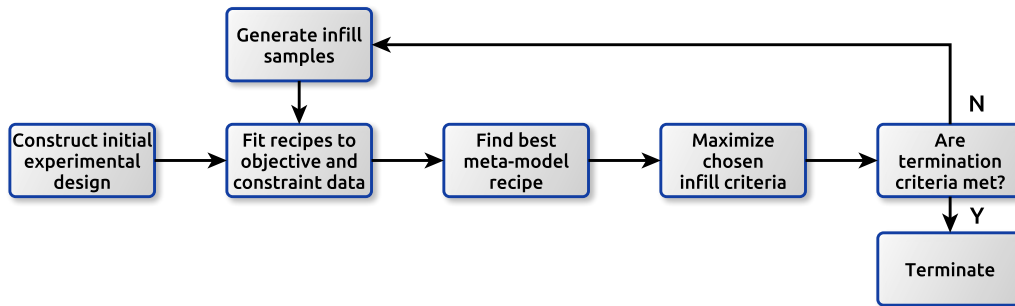


Figure 4.6: The optimization process. Model recipes, made up from the transforms in Table 4.1 and meta-models in Table 4.2, are fitted to the currently available data. The best meta-models are selected and used to evaluate the chosen infill criteria. This process continues until convergence has been achieved.

models are fit to the objective and constraint functions, using a collection of predefined model recipes associated with each function. The best meta-model is identified, using the R^2 coefficient calculated by cross-validation. Various infill criteria are optimized, using the estimates of mean and variance from the best meta-model. If the best meta-model does not provide an estimate of variance, the variance estimate from the best kriging model is used along with the mean estimate from the meta-model in question. Convergence is checked after optimizing infill criteria, using the approach described in Section 4.7.5. If the optimization process is adjudged not converged, then the infill samples are evaluated and new meta-models are trained for the objective and constraint functions. This process is repeated until convergence is achieved. In Appendix C the algorithm described in this section is applied to an example problem.

4.8 Conclusion

In this section three data transforms and three meta-modeling techniques were introduced. Their purpose is to replace the computationally expensive CFD simulation codes in the optimization process. An optimization technique was presented, referred to as efficient global optimization. This technique is specifically developed for use with meta-models which provide probabilistic information about the underlying function, such as the kriging meta-model. In the next chapter, efficient global optimization will be used to develop an optimized blade design for the Rofanco compressor.

Chapter 5

The optimization of blade profiles

5.1 Introduction

In this chapter, the concepts introduced in the preceding three chapters are used to develop a new blade design for the Rofanco compressor. Section 5.2 deals with the formulation of the optimization problem, particularly blade parameterization, problem bounds and the necessary constraints. Solving these problems, using the EGO techniques of Section 4.7, is discussed in Section 5.3. The results of the optimization problems are presented in Section 5.4.

5.2 Formulation of the optimization problem

5.2.1 Introduction

In this section, some concepts needed to develop the blade design optimization problem are presented. The NACA 65 parameterization scheme, used to control airfoil shape, is introduced and the makeup of the design vector from individual airfoil sections is explained. Bounds are introduced on this design vector and the specific constraints necessary to control blade stall and incidence angle are also discussed.

Before proceeding, it is helpful to review the parameters that can influence the performance of an axial flow compressor. The overall performance of an axial flow compressor is determined by the flow coefficient (Equation 2.2.4), pressure coefficient (Equation 2.2.3) and degree of reaction (Equation 2.2.5). These can in turn be related to physical properties of the compressor itself, such as flow area, speed of rotation, mean radius, number of blades, blade chord and blade shape. Because the aim of this work is to design a new set of compressor blades compatible with the existing Rofanco compressor, axial blade spacing, flow area, speed of rotation, mean radius and the number of blades are all fixed by the problem definition. The remaining free parameters pertain to blade shape, as well as blade chord and thickness. Of these, blade

chord and thickness are set to those of the current machine and repeating stages are used, for reasons of computational cost. As a result, blade shape is the only remaining free parameter.

5.2.2 NACA 65 airfoils

NACA 65 airfoils were originally designed to maximize the range of laminar flow and for approximately uniform loading along the chord of the airfoil (Aungier (2003)). They have been in use for at least the past 70 years (Abbott *et al.* (1945)), in a variety of fields, such as wind turbines (Timmer (2009)), jet aircraft (Whitcomb (1974)) and compressors (Aungier (2003)).

The NACA 65 airfoil series were originally intended for use as isolated airfoils in a free stream. In order to increase their suitability for compressor applications, it is common to make modifications to standard NACA 65 blades. The original NACA 65 airfoils are not structurally sound in the compressor context, due to their thin trailing edge (Aungier (2003)). Therefore, the modified NACA 65 thickness distribution of Kovach and Sandercock (1961) is used, which has a thicker trailing edge. Another modification used in this work is to replace the original NACA 65 camber line with a circular arc camber line, as this has become the normal camber line to use with NACA 65 blades (Cumpsty (1989)).

NACA 65 airfoils are commonly specified using the six-digit airfoil code (not explained here) from which the NACA 6 airfoil series take their name. Since the thickness distribution of the modified NACA 65 airfoils is known and the combination of blade inlet and outlet angles completely specifies a circular arc camber line, blade angles can be used to specify the modified NACA 65 airfoil. In this work, NACA 65 blades are specified using inlet (χ_1 or κ_1) and outlet (χ_2 or κ_2) blade angles.

5.2.3 Makeup of the design vector

The inlet and outlet blade angles are controlled at three radial stations. If the rotor and stator blade angles are denoted by χ and κ respectively, then the fragment of the design vector relating to the i -th radial station is given by:

$$\mathbf{b}_i = [\chi_1^i \quad \chi_2^i \quad \kappa_1^i \quad \kappa_2^i]^T \quad (5.2.1)$$

The whole design vector is made up of three radial station fragments:

$$\mathbf{x} = [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3]^T \quad (5.2.2)$$

In order to construct a compressor blade from the three radial stations, cubic splines are used to interpolate blade data at each station, at a constant chord fraction, as shown in Figure 5.1.

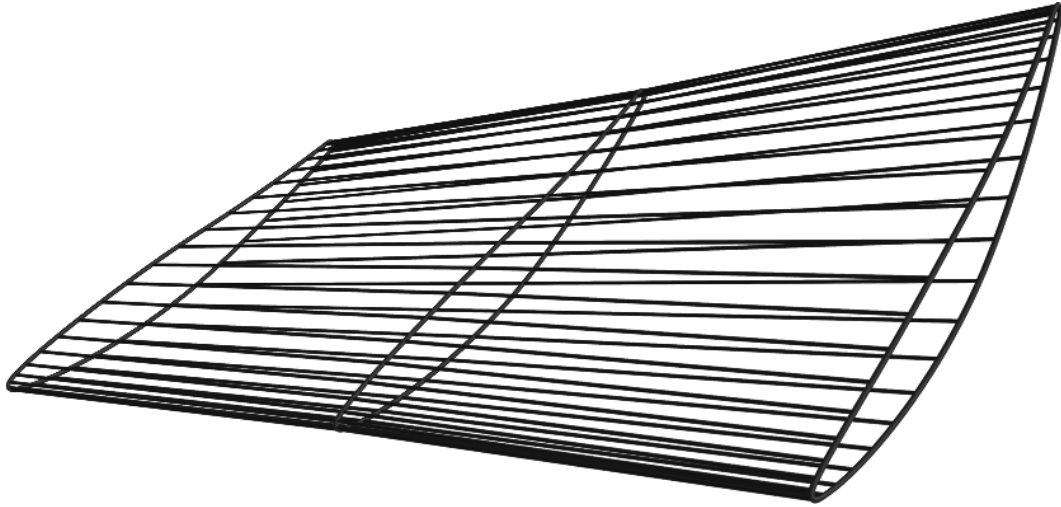


Figure 5.1: Compressor blades are constructed from three radial stations (the airfoil shapes) by interpolating station data radially, using cubic splines.

5.2.4 Bounding the design vector

It is necessary to establish upper (\mathbf{x}_u) and lower bounds (\mathbf{x}_l) on the design vector. The choice of bounds is problem dependent, but not critical as the search space can be expanded if necessary. The arbitrarily chosen design space is centered around the current design for the Rofanco compressor ($\bar{\mathbf{x}}$) and a constant offset of 10° is used to establish the upper and lower bounds of the design space. The bounds are listed in Table 6.1 and given by the equations below.

$$\mathbf{x}_l = \bar{\mathbf{x}} - 10^\circ \quad (5.2.3)$$

$$\mathbf{x}_u = \bar{\mathbf{x}} + 10^\circ \quad (5.2.4)$$

5.2.5 Outlet flow angle constraints

To ensure the validity of the single-stage model, it is necessary to constrain the angle of the flow out of the stator so that the incidence upon the next rotor blade is within acceptable limits. The incidence upon the rotor row at radius r can be determined from the blade angle and outlet flow angle, which is related to the outlet flow velocity:

$$i(r) = \chi_1(r) - \beta_3(r) = \chi_1(r) - \arctan\left(\frac{U(r) - C_t(r)}{C_a(r)}\right) \quad (5.2.5)$$

Then the constraint takes the form:

$$i_{min} \leq i(r) \leq i_{max} \quad (5.2.6)$$

Since the inlet blade angles are only specifically known at the radial stations, they are interpolated using cubic splines. This constraint is applied at

five equally spaced radii, outside of the boundary layer. The boundary layer is excluded for two reasons. Firstly, the flow fields near the hub and shroud are complex, due to tip gap and boundary layer effects. No detailed validation of the flow fields in this area was undertaken and there were concerns that valid designs could be excluded due to model error. Secondly, during post-processing the CFX results, it was observed that there was a significantly greater variance in flow quantities from grid points near the hub and shroud. Because of this, it was questionable whether constraining the incidence angle calculated from the average axial and tangential velocity had physical meaning.

5.2.6 Stall margin constraints

It is necessary to include a constraint to exclude designs which have a stalling mass flow greater than the design stalling mass flow. Stall is evaluated using the techniques of Section 3.6: three static simulations are run at different mass flows and stall is determined by comparing their pressure coefficients.

As a population based optimizer is used to solve the infill optimization problem, there is no requirement that the stall constraint be continuous. Therefore, the stall prediction problem is modeled as a classification problem, in order to avoid having to train three separate meta-models to predict $\psi_{ss}^{stall,1}$, $\psi_{ss}^{stall,2}$ and ψ_{ss}^{design} individually.

$$S(\mathbf{x}, m_{stall,1}, m_{stall,2}) = \begin{cases} -1 & \psi_{ss}^{stall,1} \in [\psi_{ss}^{stall,2}, \psi_{ss}^{design}] \\ 1 & \psi_{ss}^{stall,1} \notin [\psi_{ss}^{stall,2}, \psi_{ss}^{design}] \end{cases} \quad (5.2.7)$$

For the Rofanco compressor, these stall mass flows are set as follows:

$$m_{stall,1} = 2.30 \text{ kg/s}, \quad m_{stall,2} = 2.25 \text{ kg/s}, \quad m_{design} = 2.70 \text{ kg/s} \quad (5.2.8)$$

5.2.7 Stagger and camber angle constraints

In order to restrict blades to those that are physical, constraints on stagger and camber angles are introduced. A stagger angle of $\zeta = 0^\circ$ implies blades aligned with axial flow, while a stagger angle of $\zeta = 90^\circ$ implies blades perpendicular to the oncoming flow. Stagger angle is thus constrained to be between these angles, in radians:

$$0 \leq \zeta \leq \frac{\pi}{2} \quad (5.2.9)$$

Substituting for rotor and stator blade angles, yields four constraints:

$$0 \leq \frac{1}{2}(\chi_1 + \chi_2) \leq \frac{\pi}{2} \quad (5.2.10)$$

$$0 \leq \frac{1}{2}(\kappa_1 + \kappa_2) \leq \frac{\pi}{2} \quad (5.2.11)$$

Similarly, a blade of zero camber $\theta = 0^\circ$ does not turn the flow, while a blade of $\theta > 90^\circ$ turns the flow back in the direction it came. Putting into constraint form, with the angle in radians again:

$$0 \leq \theta \leq \frac{\pi}{2} \quad (5.2.12)$$

Once again, substituting blade angles for camber angle, yields four additional constraints:

$$0 \leq \chi_1 - \chi_2 \leq \frac{\pi}{2} \quad (5.2.13)$$

$$0 \leq \kappa_1 - \kappa_2 \leq \frac{\pi}{2} \quad (5.2.14)$$

5.2.8 Standard form optimization problem

Using the information of this section, it is possible to construct an optimization problem in standard form. The pressure coefficient, ψ_{ss} is maximized, subject to the incidence being greater than the minimum allowable incidence ($i_{min} = -5^\circ$) and less than the maximum allowable incidence ($i_{max} = 5^\circ$). Further, the compressor must not stall before some critical stalling mass flow ($S(m_{stall,1}, m_{stall,2}) \leq 0$) as discussed in Section 5.2.6 and the blades must be physically viable, as discussed in Section 5.2.7.

$$\begin{aligned} \min & & & -\psi_{ss} \\ \text{subject to:} & & i(R) - i_{max} & \leq 0 \\ & & i_{min} - i(R) & \leq 0 \\ & & S(m_{stall,1}, m_{stall,2}) & \leq 0 \\ & & \Upsilon_1^j + \Upsilon_2^j - \pi & \leq 0 \\ & & -(\Upsilon_1^j + \Upsilon_2^j) & \leq 0 \\ & & (\Upsilon_1^j - \Upsilon_2^j) - \frac{\pi}{2} & \leq 0 \\ & & (\Upsilon_2^j - \Upsilon_1^j) & \leq 0 \\ & & \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u & \end{aligned} \quad (5.2.15)$$

$$\begin{aligned} \text{where:} & & R \in \{i \in [0, 4] : 0.16 + 0.1i\} \\ & & j \in [1, 3] \\ & & \Upsilon = \chi, \kappa \end{aligned}$$

5.3 Solving the optimization problem

5.3.1 Introduction

The optimization problem in Section 5.4 can be solved with the methods of Section 4.7. In this section, some problem specific aspects of this process are described. An initial optimal latin hypercube of 120 samples was trained.

Table 5.1: The choice of models used to solve the optimization problems of Section 5.4 are listed in the table below. For explanations and definitions of each of these recipes, see Section 4.7.7.

Response surfaces	SVM	Kriging
scaled rs	scaled svr	scaled kriging
scaled quad-rs	scaled pca svr	scaled pca kriging
scaled pca rs		
scaled pca quad-rs		

Table 5.2: The number and type of infill samples. The choices in this table are problem dependent, but were shown to perform well in test problems.

Technique	Count	Aim
Maximum variance	2	Improve model quality by reducing variance
Expected improvement	5	Maximize the probability of improving the optimum
Meta-model optima	3	Sample the predicted optimum, in order to realize performance gains
Total	10	

These candidate designs are summarized in Appendix D. Meta-models and data transforms of the types outlined in Section 5.3.2 were fitted to this initial data, whereafter infill samples are generated using the criteria of Section 4.7.2 and sampled using the numerical models of Chapter 3. These new samples are used to train new models and the process is repeated until the termination criteria of Section 5.3.5 are met, at which point the optimization process is terminated.

5.3.2 Choosing the meta-models

It is necessary to choose meta-model varieties from the combinations of data transforms in Section 4.5 and the models in Section 4.6. The models used in this work are listed in Table 5.1 and based on the recipes described in Section 4.7.7. All models are scaled for numerical conditioning reasons.

5.3.3 Choosing the infill samples

Ten infill points per EGO iteration were sampled, as this allowed a cadence of one EGO iteration per day. The breakdown of these ten infill samples is listed in Table 5.2. In order to generate n meta-model optima infill samples, the optima of the n most accurate meta-models are used.

5.3.4 The infill optimizer settings

In order to generate multiple infill samples per iteration criterion, using the approach of Section 4.7.3, it is necessary to set the value of $r_{exclusion}$, a minimum radius between infill samples included as a constraint on the infill optimization process. It is possible to set $r_{exclusion}$, based on the minimum distance between design vectors that can be realized in a physical manifestation of the design. It is probable that the manufacturing tolerance on critical blade angles is greater than 0.5° . Accordingly, the exclusion radius can be set to a distance corresponding to a difference of 0.5° on each blade angle:

$$r_{exclusion} = \sqrt{12 \left(0.5^\circ \left(\frac{\pi}{180^\circ}\right)\right)^2} = 32.23(10^{-3}) \quad (5.3.1)$$

5.3.5 Termination criteria

Section 4.7.5 outlines convergence criteria for an EGO based optimization scheme. There must be less than a threshold probability of improvement i.e. $\max\{E(I)\} \leq 0.01$. The objective function must be well approximated by the meta-model i.e. $R^2 \geq 0.7$. At least $I_{min} = 10$ iterations must have been completed before convergence has been achieved. The maximum number of iterations is set at $I_{max} = 20$.

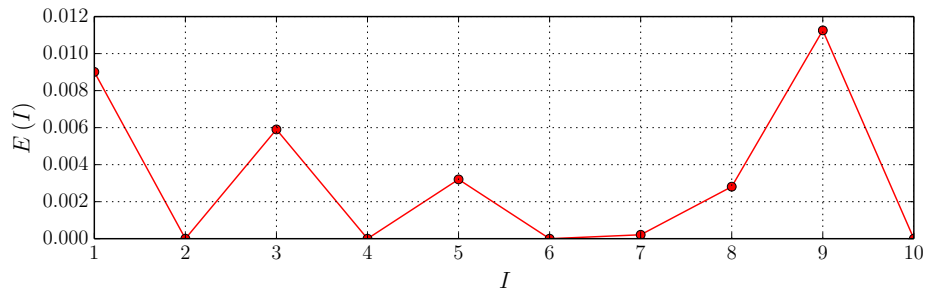
5.4 Results of the optimization problem

Ten iterations of EGO optimization were performed on the problem described in Section 5.2.8. Over the course of the optimization process ten designs were discovered which outperformed the original blade design. In iterations two, six and nine a new optima was found. The best single stage pressure coefficient achieved was $\psi_{ss}^* = 0.330$. This constituted an improvement of 8.55% over the original value of $\bar{\psi} = 0.304$. This optimized design is analyzed further in Chapter 6.

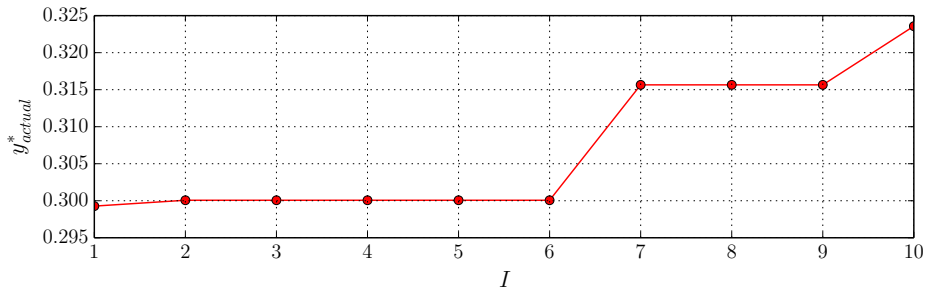
Figure 5.2 shows the expected improvement and model optima, over the course of the optimization process. Peak expected improvement of just over 1% is achieved in iteration nine. The improvements in iterations one and nine are preceded by substantial peaks in expected improvement, but the improvement in iteration six is associated with an expected improvement of $E(I) \approx 0$.

A bar chart categorizing the ten infill samples per iteration is given in Figure 5.3. It is apparent from this figure that a substantial number of infill samples are placed into constrained regions of the design space. This suggests that a more sophisticated constraint handling strategy might have been beneficial, in order to deal with locally poor predictions of constraint boundaries.

It is expected that the area around the optima be highly infeasible, as it is likely that both incidence angle constraints and stall constraint will be active



(a) Expected improvement as a function of design iterations



(b) Best pressure coefficient as a function of design iterations

Figure 5.2: This figure contrasts the expected improvement with the predicted and actual model optima. Note the lack of a clear relationship between the optima of a given iteration between expected improvement of that iteration. This might be caused by locally poor predictions of the constraint boundary, as expected improvement does not take the possibility of infeasibility into account.

or near-active at the optimum. It is also interesting to note that most feasible designs either improved upon the initial Rofanco blade design or were in fact new optima. The fact that few poor feasible designs were found would seem to indicate that the optimizer narrowed in on areas of high performance from the infeasible side of the constraint boundary.

Infill samples are unseen data from the perspective of the models of the previous iteration. Accordingly, they constitute an independent test set which can be used to assess the quality of those models. As shown in Table 5.3, for this unseen infill data the objective function meta-model uniformly scored $R^2_{infill} > 0.9$ for all iterations. Also shown in this table is the best recipe for response surfaces, support vector regression and kriging meta-models. It is interesting to note that $R^2_{cv} > 0.95$ for all recipes, which indicates that the objective function models were of high quality. Further, the `scaled rs` recipe was consistently the best performing recipe. This indicates that the underlying function was well approximated by a quadratic polynomial, in the region of interest. The data used to train the meta-models in this section is available in Appendix D.

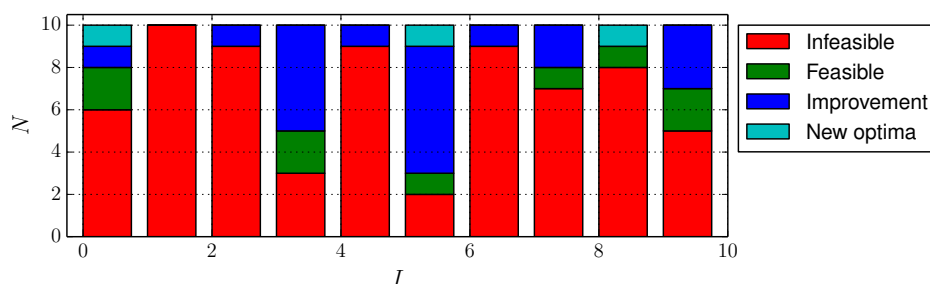


Figure 5.3: In this figure, infill samples are categorized over the course of the optimization. Note the abundance of constrained infill samples and the fact that most feasible designs found are actually better than the original Rofanco blade design. This seems to indicate that the optimizer approached the optimum from the infeasible region.

Table 5.3: The objective function meta-model error. Notice that the **scaled rs** recipe is the best performing meta-model and that the model performs well ($R_{infill}^2 > 0.9$) when evaluated with unseen infill data.

Best response surface	R_{cv}^2	Best SVR	R_{cv}^2	Best kriging	R_{cv}^2	R_{infill}^2
scaled rs	0.995	svr-rbf	0.983	krig	0.986	0.901
scaled rs	0.987	svr-rbf	0.955	scaled krig	0.981	0.980
scaled rs	0.992	scaled svr-rbf	0.964	krig	0.988	0.991
scaled rs	0.990	scaled svr-rbf	0.962	krig	0.989	0.980
scaled rs	0.993	scaled svr-rbf	0.972	scaled krig	0.992	0.990
scaled rs	0.992	scaled svr-rbf	0.969	krig	0.991	0.970
scaled rs	0.993	svr-rbf	0.971	scaled krig	0.993	0.977
scaled rs	0.991	svr-rbf	0.973	scaled krig	0.993	0.986
scaled rs	0.993	svr-rbf	0.976	krig	0.994	0.985
scaled rs	0.993	svr-rbf	0.977	scaled krig	0.993	0.960

5.5 Conclusion

In this chapter, an optimization problem in standard form was developed to redesign the Rofanco compressor blades. This problem was evaluated using the numerical simulation described in Chapter 3 and solved using the MBDO techniques of Chapter 4. Metrics from the optimization process were examined in Section 5.4. After ten iterations of EGO, an optimum blade design was found, which constituted a 8.55% improvement over the current design. In the next chapter, these new design will be subjected to more detailed analysis, comparable to the numerical experiments conducted for the Rofanco in Chapter 3.

Chapter 6

Analysis of the optimized blade design

6.1 Introduction

In Chapter 5, an optimized blade design for the Rofanco compressor was developed. The focus of this chapter is subjecting this new blade design to more detailed analysis, similar to that conducted in Chapter 3. Accordingly, the pressure coefficient of the new design is plotted as a function of mass flow in Section 6.3.2 and axial and tangential velocities are plotted as a function of radius in Section 6.3.3. The incidence angles are computed from this data and further examined in Section 6.3.4. In Section 6.4, the design is manually adjusted in to lower the incidence angles and improve performance.

6.2 Examining the optimized blade design

The original and optimized blade designs are shown in Figure 6.1, at the three radii at which shape is controlled. Table 6.1 lists the blade angles associated with each station, for the original ($\bar{\mathbf{x}}$) and optimum (\mathbf{x}^*) blade designs, along with the upper and lower bounds on the design space. From the analysis of Chapter 2, it is known that the turning angle must be increased in order to increase the performance of the compressor. This means that χ_2 must be increased and κ_2 must be decreased. This increase is clearly visible for stations one and two, but station three increases both χ_2 and κ_2 . This might be due to the incidence constraints only being applied between $0.15 \text{ m} \leq r \leq 0.20 \text{ m}$, while station three is at $r = 0.21 \text{ m}$. This means that the turning angle can be increased at station three, without concern for the incidence upon the next stage. χ_1 is also much lower for the optimized design than the original Rofanco design. This results in the maximum allowable incidence angle ($i_{max} = 5^\circ$) being exceeded, even without the effects of flow deviation (δ):

$$i = \beta_1 - \chi_1 = \kappa_2 + \delta - \chi_1 = 54.50^\circ - 46.37^\circ + \delta = 8.13^\circ + \delta \quad (6.2.1)$$

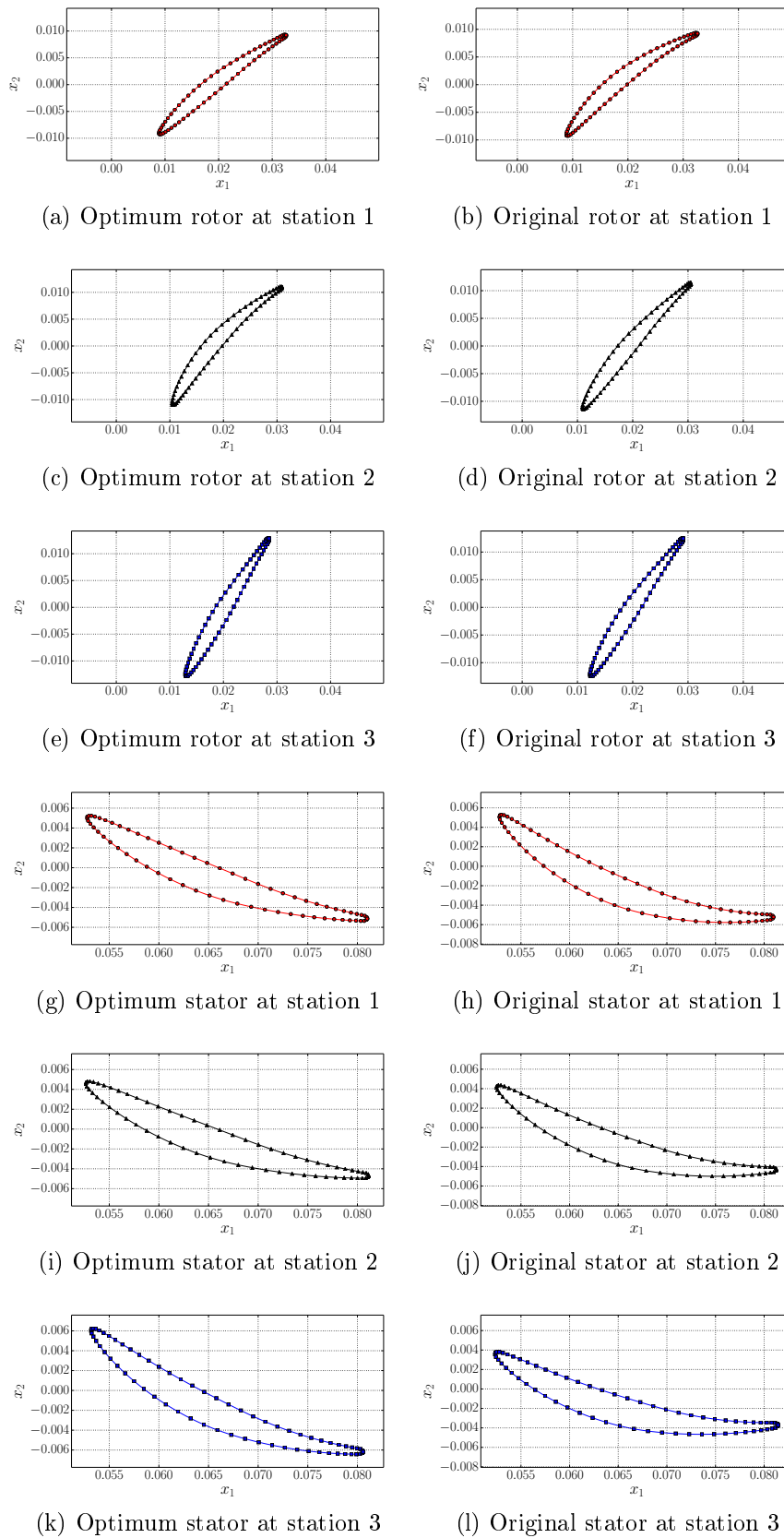


Figure 6.1: Shown in this figure are the original and optimized compressor blades for the Rofanco compressor.

Table 6.1: The current design, optimum design and upper and lower bounds.

	\mathbf{x}_l	$\bar{\mathbf{x}}$	\mathbf{x}^*	\mathbf{x}_u
χ_1^1	43.52°	53.52°	46.37°	63.52°
χ_2^1	12.48°	22.48°	29.48°	32.48°
κ_1^1	33.52°	43.52°	33.52°	53.52°
κ_2^1	-12.76°	-2.76°	6.47°	7.24°
χ_1^2	48.37°	58.37°	63.84°	68.37°
χ_2^2	30.44°	40.44°	30.44°	50.44°
κ_1^2	27.14°	37.14°	31.97°	47.14°
κ_2^2	-13.92°	-3.92°	4.54°	6.09°
χ_1^3	51.55°	61.55°	62.50°	71.55°
χ_2^3	40.65°	50.65°	55.41°	60.65°
κ_1^3	24.32°	34.32°	44.32°	44.32°
κ_2^3	-15.68°	-5.68°	4.32°	4.32°

Examining the blade angles in Table 6.1 shows that several design variables are pushing up against the design bounds. κ_1^3 and κ_2^3 are against the upper blade angle limit, while χ_2^2 and κ_1^1 are against the lower blade angle limit. This might indicate that there is more opportunity for further performance increases if the bounds are expanded.

6.3 Comparing the original and optimized blade designs

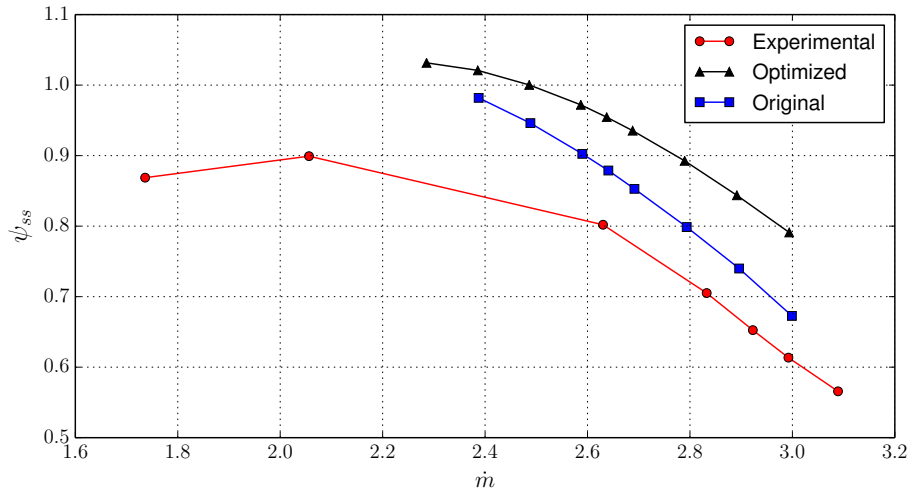
6.3.1 Overview

In this section, the performance of the optimized blade design is compared to that of the original blade set. In Section 6.3.2, the pressure coefficients of the two designs are compared, while in Section 6.3.3, the velocity profiles are compared. The incidence angles are computed from the velocity profiles and examined in Section 6.3.4.

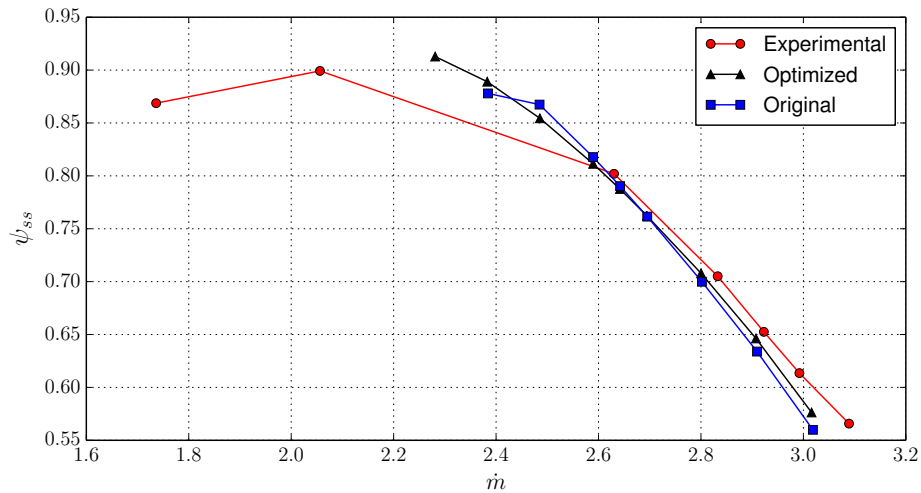
6.3.2 Pressure coefficient

Pressure coefficients for the new blade design are shown in Figure 6.2. In Figure 6.2a, the first stage pressure coefficient at the design point, reported by the single stage model, has increased by 8.55%, from $\psi_{ss} = 0.304$ to $\psi_{ss} = 0.330$. Unfortunately, Figure 6.2b shows that this increase does not translate into an increase over the full domain; in fact the pressure coefficient reported by the full model is essentially equal to that of the current design.

It is clear that the increases from the single stage model do not translate into equivalent increases over the full model for the optimum design. In



(a) Single stage model



(b) Full model

Figure 6.2: The static-to-static pressure coefficient for the original and optimized blade designs, as reported by the single stage and full models. Experimental data is also shown, for reference. Note the increase reported by the single stage model is not sustained in the full model.

Table 6.2: The pressure rise over each stage for the optimized and original Rofanco blade design are shown below. It is noticeable that optimized blades perform substantially better over the first stage, but that this performance deteriorates over the second and third stage.

Blade set	Model	Δp_1^{ss}	Δp_2^{ss}	Δp_3^{ss}
Original	Single stage	1151.02 Pa	-	-
Original	Full model	1164.07 Pa	904.21 Pa	923.73 Pa
Optimized	Single stage	1250.06 Pa	-	-
Optimized	Full model	1262.70 Pa	863.11 Pa	856.33 Pa

this regard, it is interesting to examine the pressure rise over each stage, as tabulated in Table 6.2. Clearly, the optimized model produces a significantly increased first stage pressure rise. This is unsurprising, as this is the quantity acted upon by the optimizer in Chapter 5. This increase is not carried over to the second and third stages, in which the pressure rise for the optimum design is in actual fact lower than that of the original design. The reason for this discrepancy is further explored in Section 6.3.3 and Section 6.3.4.

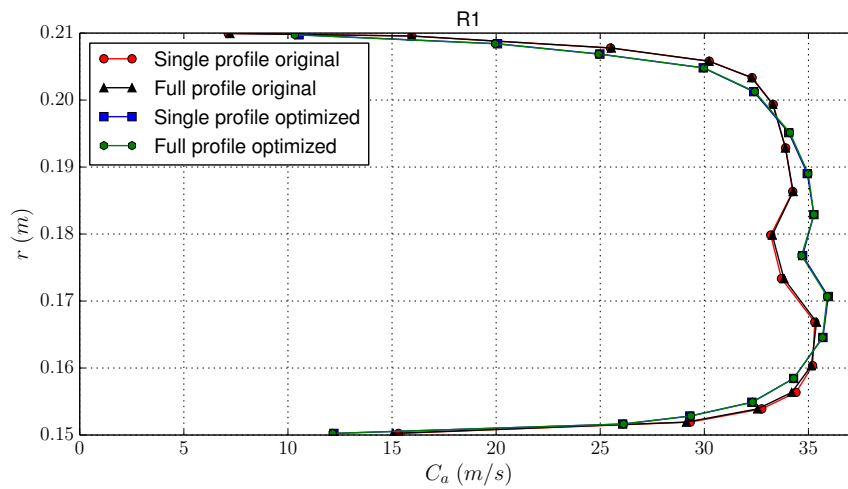
6.3.3 Velocity profiles

Axial velocity profiles for the original and optimized blade sets are shown in Figure 6.3. Axial velocity profiles for the optimized and original models are similar at R1. As the flow moves through the machine, they begin to diverge. The velocity profile for the optimized blades becomes more rounded, with the effect of lowering the axial velocity near the hub and shroud and significantly increasing the axial velocity near midspan. From Section 2.2.4, it can be recalled that the reason for introducing the work-done factor was the decrease in energy imparted to the fluid caused by the increased midspan axial velocity. This increase in axial velocity through the machine is probably part of the cause of the low performance of the second and third stage noticed in Section 6.3.2.

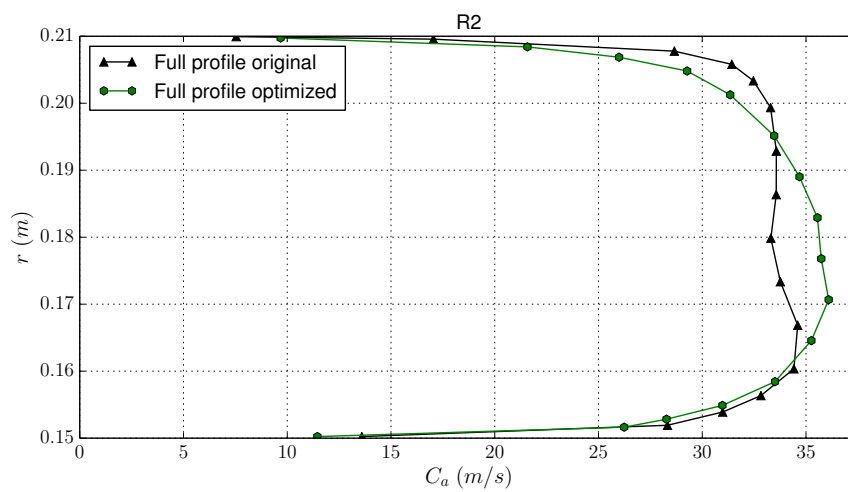
Tangential velocity profiles for the original and optimized blade sets are shown in Figure 6.4. The optimized and original velocity profiles show similar forms, with the optimized velocity profiles having higher magnitude. Since increasing the tangential velocity results in an increased amount of work being done on the fluid, this change is expected in an optimized blade set.

6.3.4 Incidence

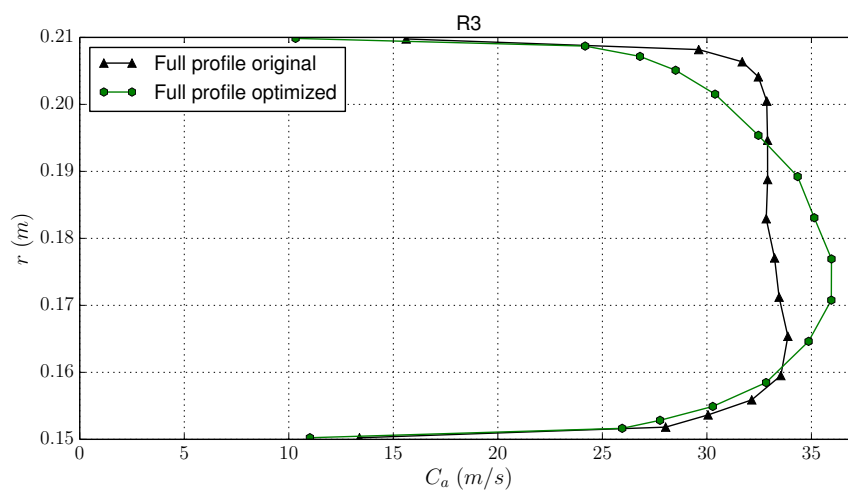
Using the data from Figures 6.3 and 6.4, it is possible to compute the flow angle out of the stator, as a function of radius. Flow angle out of the first stage as well as blade angles, interpolated via cubic splines, are shown in Figure 6.5a. Using these two quantities, incidence upon the second stage rotor blades can



(a) R1 axial velocity measurements

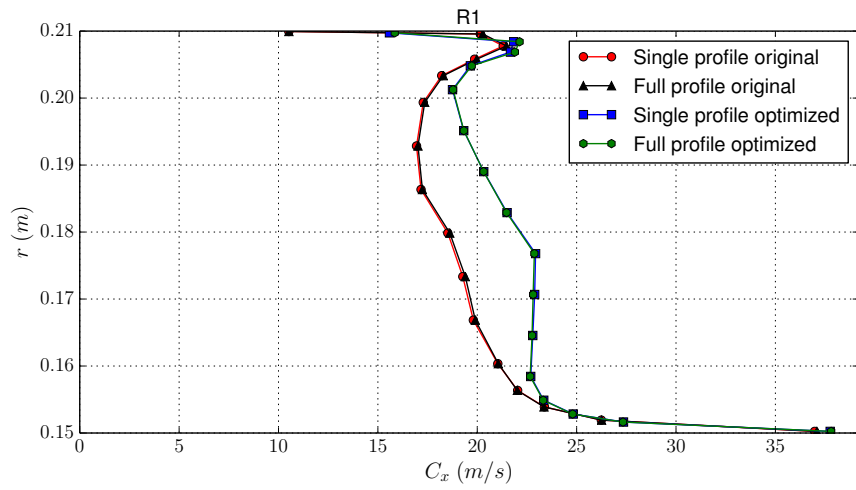


(b) R2 axial velocity measurements

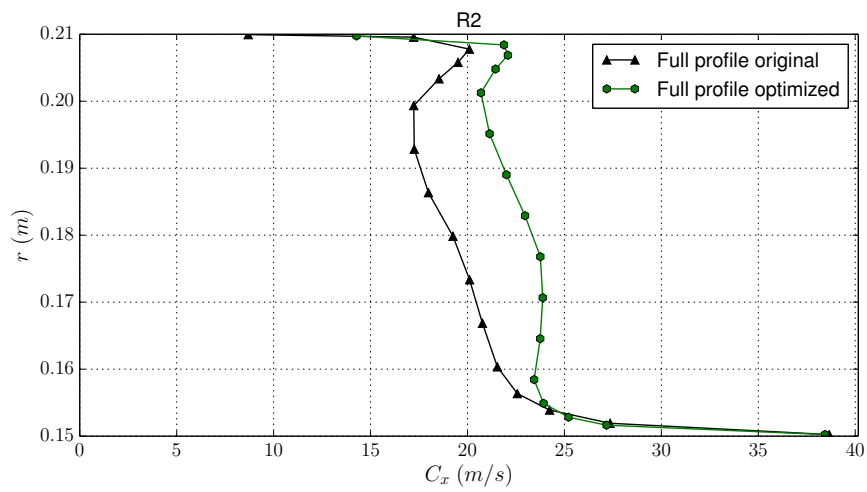


(c) R3 axial velocity measurements

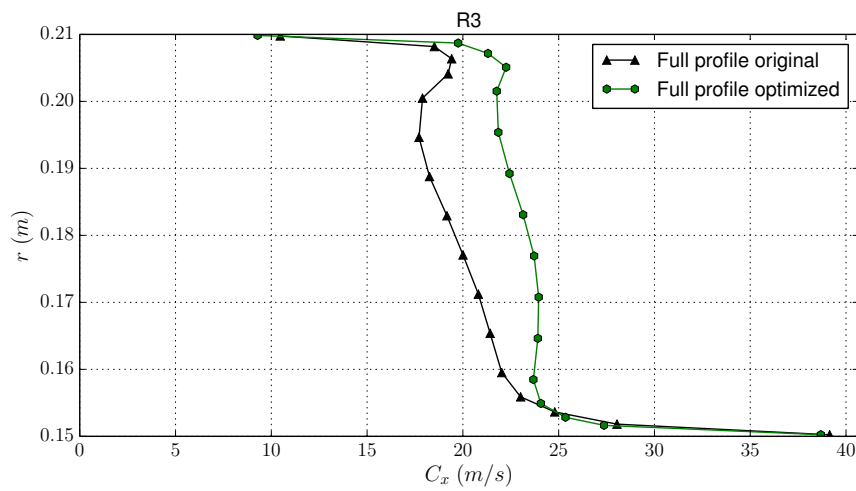
Figure 6.3: In this figure, the axial velocity predictions for the original and optimized blade designs are plotted as a function of radius.



(a) R1 tangential velocity measurements



(b) R2 tangential velocity measurements



(c) R3 tangential velocity measurements

Figure 6.4: In this figure, the tangential velocity predictions for the original and optimized blade designs are plotted as a function of radius.

be calculated and is shown in Figure 6.5b, along with the upper and lower incidence limits of $i_{min} = -5^\circ$ and $i_{max} = 5^\circ$. It is clear that these limits are not adhered to in the region of the hub and shroud.

The statement of the optimization problem in Equation 5.2.15 mentions that incidence is controlled at five equally distributed actual locations between $r = 0.16$ m and $r = 0.20$ m. Inside this range, incidence angle is within the mandated limits, but outside the range incidence is greater than i_{max} , reaching up to $i = 20^\circ$. Incidence this high is very likely to cause flow separation near the root and tip of the blade, leading to reduced performance in those areas and explaining the blockage that lead to more peaked axial velocity profiles in Section 6.3.3.

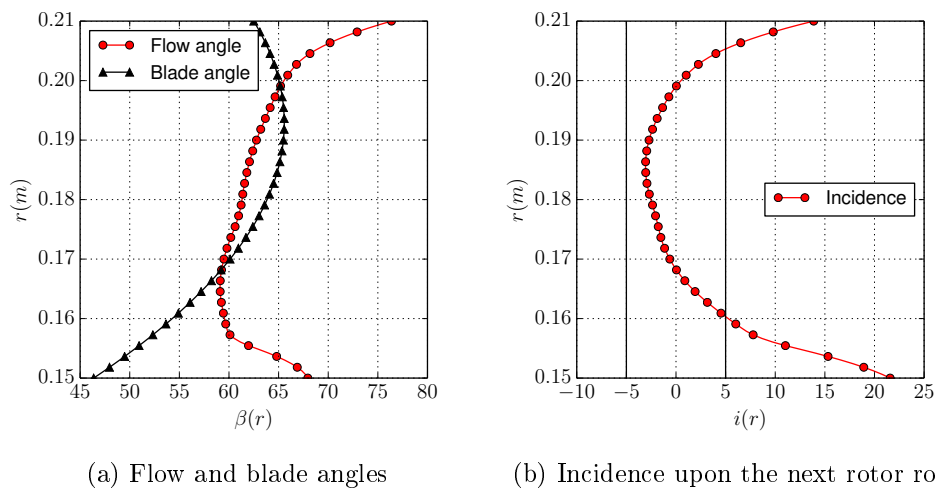


Figure 6.5: In this figure, the blade and flow angles are compared, in order to determine the incidence upon the next rotor row. Note how incidence is greater than the allowable limit near the hub and shroud, where incidence is not constrained. This is likely the cause of the blockage in axial velocity visible in Figure 6.3.

6.4 Adjusting the optimized blade design to lower the incidence angle

6.4.1 Overview

From the analysis in Section 6.3.4 and the blade angles in Table 6.1, it is clear that excessive incidence is worsening the performance of downstream stages. There are three ways to solve this problem. The first of these is to restart the optimization process, using the full profile model instead of the single profile

model. This approach implies extreme computational cost; the full profile model was initially ruled out for reasons relating to computational cost.

The second approach is to reformulate the optimization problem of Equation 5.2.15 in order to include incidence constraints that extend right the way to the blade hub and shroud. This new optimization problem would have to be solved again, but the process could be greatly sped up by bootstrapping the model with existing data, since calculating the incidence angle is merely a post-processing operation. A concern with this approach is that the design space is already highly constrained and adding additional incidence constraints will worsen this problem. Adding additional constraints will necessitate a more sophisticated approach to handling incidence than that is outlined in Section 4.7.4.

The third approach to solving the problem of excessive incidence is to manually adjust the blade angles to lower the incidence upon the second and third rotor stages. It is obvious that this is a substantially simpler task than restarting the optimization process. Even were one to choose to restart the optimization process with either the single profile or full profile models, seeding it with incidence adjusted designs would be sensible. Accordingly, in this section the optimum blades are adjusted to decrease the incidence angle. The new set of blades are referred to as the “adjusted” blade set.

6.4.2 Calculating new blade angles

In order to lower the incidence angle, i , upon the first rotor, the stator outlet flow angle, α_3 , should be adjusted so that relative rotor inlet angle, β_1 , implied by α_3 , is close to the rotor inlet blade angle, χ_1 . From Figure 6.5, it may be observed that incidence angles are bounded by $i(0.15 \text{ m}) \approx 25^\circ$ and $i(0.21 \text{ m}) \approx 15^\circ$. Accordingly, incidence at $r = 0.15 \text{ m}$ must be reduced by 20° and incidence at $r = 0.21 \text{ m}$ must be reduced by 10° .

The necessary decreases in incidence could be achieved by either increasing the inlet blade angle or decreasing the outlet flow angle. Five simulations were run with the intention of decreasing the incidence upon the downstream rotor. Two of these achieved the decreased incidence by only adjusting the rotor inlet angle or stator outlet angle respectively. The remaining three simulations linearly interpolated between these extremes. The most successful design was that which only adjusted the stator outlet angle.

6.4.3 Adjusted blade angle results

The blade angles for the best adjusted design are shown in Table 6.3. In Figure 6.6, the pressure coefficient as a function of mass flow is shown for the best adjusted design, along with the original optimized design. The performance for the original, optimized and adjusted blade designs is summarized in Table 6.4. The adjusted design slightly exceeds the 8.55% improvement shown by the

Table 6.3: The current design, optimum design, adjusted design and upper and lower bounds.

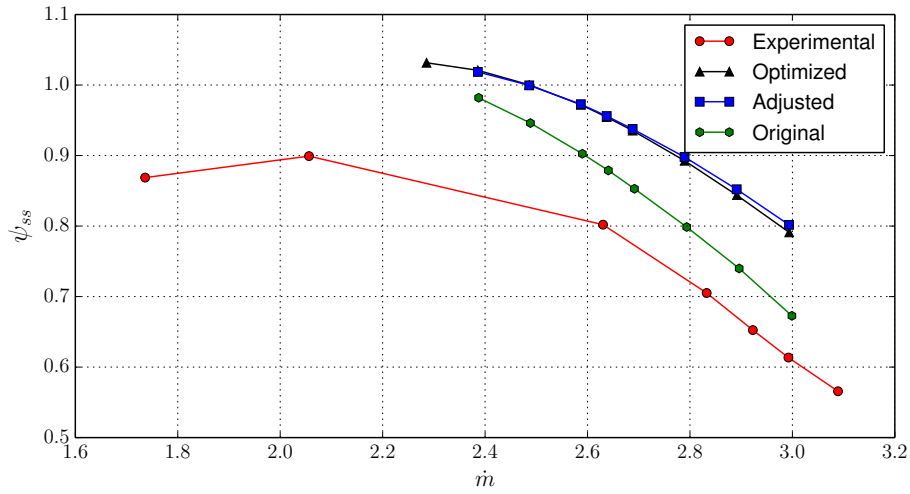
	\mathbf{x}_l	$\bar{\mathbf{x}}$	\mathbf{x}^*	\mathbf{x}^{adj}	\mathbf{x}_u
χ_1^1	43.52°	53.52°	46.37°	46.37°	63.52°
χ_2^1	12.48°	22.48°	29.48°	29.48°	32.48°
κ_1^1	33.52°	43.52°	33.52°	33.52°	53.52°
κ_2^1	-12.76°	-2.76°	6.47°	-5.68°	7.24°
χ_1^2	48.37°	58.37°	63.84°	63.84°	68.37°
χ_2^2	30.44°	40.44°	30.44°	30.44°	50.44°
κ_1^2	27.14°	37.14°	31.97°	31.97°	47.14°
κ_2^2	-13.92°	-3.92°	4.54°	4.54°	6.09°
χ_1^3	51.55°	61.55°	62.50°	62.50°	71.55°
χ_2^3	40.65°	50.65°	55.41°	55.41°	60.65°
κ_1^3	24.32°	34.32°	44.32°	44.32°	44.32°
κ_2^3	-15.68°	-5.68°	4.32°	-13.53°	4.32°

Table 6.4: The various designs discussed in this work and their relative improvements over the original design

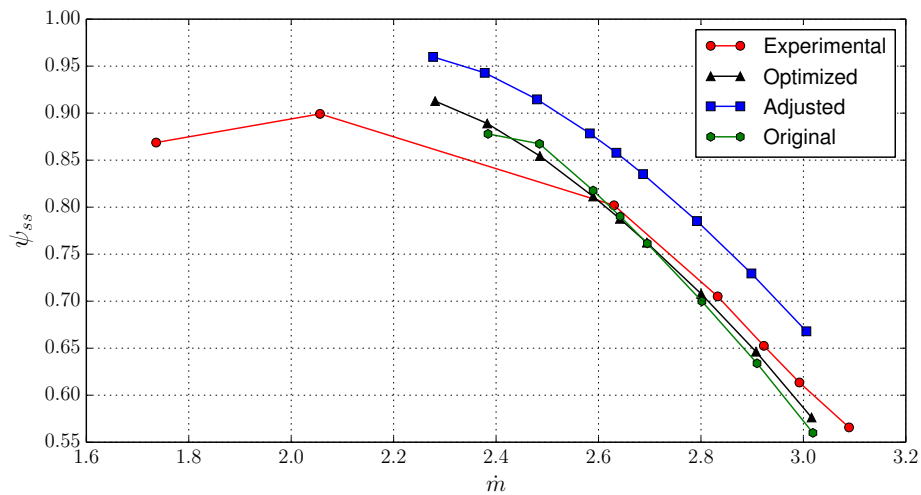
Design	Single stage model		Full model	
	ψ_{ss}	Improvement	ψ_{ss}	Improvement
Original	0.304	-	0.790	-
Optimized	0.330	8.55%	0.791	0.13%
Adjusted	0.331	8.88%	0.858	8.61%

optimized design when evaluated by the single stage model and importantly, maintains a 8.61% improvement when evaluated by the full model.

When evaluated by the single stage model, the adjusted blade design does not maintain the maximum stalling mass flow of 2.3 kg/s; it stalls at 2.4 kg/s. It should be noted that this requirement is conservative, as the current blade set stalls at 2.4 kg/s. Further, when evaluated by the full model, stalling mass flow is found to be 2.3 kg/s, meeting this objective. For these reasons, the adjusted design is deemed acceptable from a stall perspective.



(a) Single stage model



(b) Full model

Figure 6.6: The mass flow-pressure coefficient curves for the adjusted design. Note how the adjusted design maintains the increase in pressure coefficient when evaluated with the full model. The adjusted design achieves a 8.88% improvement over the current design when evaluated with the single stage model and a 8.61% improvement when evaluated with the full model.

6.5 Conclusions

In this chapter, the optimized blade design of Chapter 5 was subjected to detailed numerical analysis. The significant increase found during the optimization process using the single stage model, did not translate to an equivalent increase in the full model. This was investigated in Section 6.3.4 and found to likely be related to high incidence caused by the omission of an incidence constraint at the hub and shroud. A new design was created by manually adjusting the blade angles of the optimum design, in order to lower the incidence angle upon the downstream rotor. This adjusted design delivered an increase of 8.61% over the current design, when evaluated with the full model. For the adjusted design, several blade angles are limited by the upper and lower bounds. This indicates that there might be further opportunity for improvement if these bounds are widened.

Chapter 7

Conclusion

7.1 Introduction

Over the previous five chapters, the problem of designing a new set of compressor blades for the Rofanco compressor has been addressed. Over the course of this chapter, the contents of this work are critically analyzed. Section 7.2 to Section 7.6 discuss some conclusions relating to specific topics discussed in this work. Section 7.7 evaluates the objectives introduced in Section 1.4.

7.2 The modeling of axial compressors for design purposes

In Chapter 3, four models were considered for design purposes. These models were constructed from the combinations of two boundary conditions and two model domains. The first set of boundary conditions specified pressure and temperature at the rotor inlet and mass flow at the stator outlet. Due to poor agreement between the velocity profiles predicted by these models and experimental data, they were discarded from consideration.

The second set of boundary conditions used in this work specified a velocity profile at rotor inlet and temperature and pressure at the stator outlet. Two domains were simulated using these boundary conditions: a single compressor stage and the full compressor. Despite the full compressor model showing better agreement with experimental data, the single stage model was used in the design process for reasons of computational cost. Equation 2.3.14 was used to convert the pressure rise over a single stage to that over the whole machine.

7.2.1 Single stage approximation to the full compressor model

In order to make the computational problem tractable, a simplified single stage compressor was used in the design process. Incidence constraints were imposed on this model, in order to ensure that flow out of the stator aligned with the rotor of the next stage. These constraints were deliberately not applied at the hub and shroud radii, because it was felt that tip gap effects might distort flow fields in this area and lead to erroneous constraint violations (a severe problem, given the highly constrained nature of the design space). Unfortunately, omitting these constraints at the hub and shroud resulted in the optimizer driving high incidence angles near the rotor root and tip, which in turn lead to blockage of axial flow and a peaked axial velocity profile. This velocity profile lead to decreased performance over the full model, for reasons outlined in Section 2.2.4.

In Chapter 6, it was found that the optimizer had exploited the difference between the low fidelity and high fidelity model and driven the optimum toward an area where the behaviors of these models diverged. The relationship between the pressure coefficient reported by the single stage and full model was investigated early in this work. An initial set of ten geometries were evaluated using both the single stage and full models. The pressure coefficients reported by both models was found to vary proportionally, with no clear bias. Unfortunately, as data was generated very early on and not felt to be particularly important, it is not available today.

Had more time been spent categorizing the relationship between the single stage model and full model, it might have been possible to deduce that incidence constraints were also necessary at the hub and shroud. Part of the reasoning behind not studying this relationship in great detail before commencing the optimization process, was that a generally good correspondence between the two models might not rule out problems of the type observed in Chapter 6. The reason for this is that what mattered for the optimization process was not that the models showed general agreement; instead they needed to agree in the region of the optima. Without knowing the location of the optima, correspondence in the region of interest could not be established.

7.2.2 Automatic analysis of axial compressors

An wholly automated system for analyzing arbitrary compressor geometries was created. Designs could be specified using geometry files, created from input design vectors. Python scripts were developed to import these geometry files into ANSYS Workbench, mesh them, analyze them using CFX and post-process them with CFD-Post. These results were further processed using custom Python scripts. This flexible, automated system enabled hundreds of

CFD simulations to be run, for various parameters and arbitrary geometries, without any human intervention.

7.2.3 Evaluation of stall

In Section 3.6, a method for evaluating the stall margin of a compressor using three static simulations was developed. During the course of the optimization process it was observed that some stalled designs were incorrectly evaluated as unstalled.

The reason for this misclassification was an inability of the non-transient model to properly capture the oscillatory flow behavior during a stalled simulation. When stalled, the flow field through the machine is inherently transient, causing a fluctuating pressure rise over the machine. When modeling this process with a static simulation, these oscillations can be observed in the pressure monitors. If all convergence criteria happen to be met, while pressure is near the peak of an oscillation, an erroneously high pressure rise is reported. If this process happens for both simulations at $m_{stall,1}$ and $m_{stall,2}$, it might lead to the data point being marked as unstalled, according to the relation in Equation 3.6.1.

By far the most common behavior for stalled simulations in this work was to not converge, resulting in failures of the stall criteria being extremely rare. Accordingly, this failure in stall prediction did not impact the results of this work. The use of this criteria is recommended for future work due to its robustness and simplicity, but this shortcoming should be noted.

7.3 Meta-modeling and efficient global optimization

7.3.1 Constraint handling in EGO problems

As discussed in Section 4.7.4, infill criteria typically display a very high degree of multi-modality and therefore favor the use of a population based optimizer, such as PSO. PSO does not require the smoothness or even existence of constraint gradients. Because of this, infill constraints could be handled simply by masking the infill criteria in areas where design constraints were violated.

This crude approach to constraint handling had a drawback. When constraint values are predicted using meta-models, errors in these predictions can lead to incorrect predictions of feasibility or infeasibility of a given point. If the gradient of the constraint is flat near the constraint boundary, small prediction errors could induce a large error in the prediction of the constraint boundary. This could lead to poor performance, as the optimizer would have to creep toward the real constraint boundary, by placing infill samples up against the incorrectly predicted constraint boundary.

A better constraint handling strategy is constrained expected improvement. Constrained expected improvement is a modified form of expected improvement which incorporates probabilistic information about the feasibility of a given point. More information regarding expected improvement can be obtained from Forrester *et al.* (2008).

7.3.2 Generating multiple infill samples per iteration

In order to reduce the ratio of time spent training meta-models to time spent simulating new infill samples, it is desirable to produce multiple infill samples per iteration. In this work, two methods for achieving this were outlined. The first uses an exclusion radius to impose a minimum distance between infill samples, based on the work of Chaudhuri *et al.* (2012). The second corrects estimates of model variance to account for the infill samples added in the same iteration, by inserting fake data points at the location of the previous infill samples from this iteration. The value associated with these fake data points are the value predicted by the meta-model at that location. An interesting avenue for future work would be to compare this infill update strategy to others proposed in the literature.

7.3.3 Response surfaces and stepwise forward regression

In this work, two variants of quadratic response surfaces were used. The first variant was full polynomials with cross-terms and coefficients determined via stepwise forward regression. The second variant included no cross-terms, but preprocessed data using principle component analysis (PCA). PCA, described in Section 4.5.4, is a linear mapping between the current design space and a space that maximizes model variance in each direction. Applying PCA aligns the variance in the data with the main model directions. During the course of this work it was observed that the second variant, using PCA, created models of comparable quality to the first variant, at significantly lower computational cost.

7.3.4 SVM hyper-parameter bounds

In Equations 4.6.17 and Equations 4.6.19, bounds for the Gaussian kernel radius were derived from the support vector machine decision function. These bounds are simple to compute, yet provide a tighter bounding for the range of allowable kernel radii than is recommended by common guides, such as Chang and Lin (2011). Similar bounds can be computed for other kernels, such as the polynomial or sigmoid kernel, following the approach of Section 4.6.4. To the knowledge of the author, these simple bounds based on an understanding of the underlying SVM model have not been proposed before.

7.4 The compressor blade design problem formulation

7.4.1 Correcting the single stage blade design problem

There are several problems with the formulation of the optimization problem as outlined in Equation 5.2.15. The first of these is that the incidence angle is not controlled at the hub and shroud. From the perspective of problem formulation, the fix is straightforward: extend the incidence constraint out to the hub and shroud. A concern with this approach is that the design space is already highly constrained; adding additional constraints will worsen this. If this route is followed, a more sophisticated approach to handling constraints is recommended, such as that outlined in Section 7.3.1.

A second problem with the problem as stated in Equation 5.2.15, is that provided the incidence constraints are met, the stator outlet angle, κ_2 is largely a free parameter. Because the effects of misalignment of the stator outlet and rotor inlet are not materialized in the single stage model, κ_2 can be adjusted arbitrarily, provided it is not so grossly incorrect as to cause blade separation. This can be corrected via the observation that rotor and stator inlet and outlet angles are related via the geometry of Figure 2.1. If κ_1' is the zero incidence stator inlet angle implied by χ_2 and κ_2' is the zero incidence stator outlet angle implied by χ_2 , then an alternative parameterization for a blade station of Equation 5.2.2 is given below. The parameters a and b are offset parameters, which can be tightly bounded.

$$\mathbf{b}_i = [\chi_1^i \quad \chi_2^i \quad \kappa_1'^i + a \quad \kappa_2'^i + b] \quad (7.4.1)$$

A third problem is that several blade angles for the optimum design are pushing up against the problem boundary, as is visible in Table 6.3. In any future problem, the problem bounds should be widened. The unexpected accuracy ($R^2 \approx 0.9$) of the objective and constraint meta-models indicates that widening the bounds should not pose difficulties from the perspective of creating accurate meta-models.

7.4.2 An alternative to constraining the incidence angle

While incidence constraints are likely a helpful component of evaluating a multi-stage model using a single stage simulation, they are only an indirect control over the applicability of the single stage simulation. A more direct way to determine the applicability of a single stage model would be to compare the flow properties as a function of radius at the single stage inlet and outlet. If the outlet flow differs by more than some allowable tolerance from the inlet flow, the validity of the assumption becomes questionable.

7.4.3 Using multiple models of different fidelity

There is no particular reason to only use a single numerical model as part of the optimization process. Instead, a hierarchy of predictors can be trained to approximate the output of a series of increasingly accurate and computationally expensive models. Each predictor could be trained to approximate the difference between the sum of the previous predictors and the current data source. Since earlier models hopefully capture much of the underlying trend, the computationally expensive models higher in the hierarchy can be sampled less densely than those lower in hierarchy. This approach would help to resolve the discrepancy outlined in Section 7.4.1, while keeping computational costs in check.

One could apply this idea to the Rofanco compressor as follows. Let $\psi_{ss}^{analytic}$ be the pressure coefficient estimated via semi-analytic methods, ψ_{ss}^{single} be the pressure coefficient estimated via the single stage model and ψ_{ss}^{full} be the pressure coefficient estimated via full model. The first predictor, \hat{y}_1 would be trained on the data densely sampled from the (presumably) least accurate data, $\psi_{ss}^{analytic}$. The predictor \hat{y}_2 , would be trained on the difference between \hat{y}_1 and the predictions of the single stage compressor model i.e. $\psi_{ss}^{single} - \hat{y}_1$. The final predictor would be trained on the difference between the \hat{y}_2 and the full compressor model i.e. $\psi_{ss}^{full} - \hat{y}_2$. Predictions for the performance of the machine as a whole would be given by $\hat{y}_1 + \hat{y}_2 + \hat{y}_3$. The model \hat{y}_1 would be mostly densely sampled, while \hat{y}_2 would be less densely sampled and \hat{y}_3 would be least densely sampled.

7.4.4 Making the full model optimization problem tractable

The most reliable way to correct the design problem is to use the full profile model to evaluate the objective function. This requires making the optimization problem tractable from a perspective of computational cost. One approach is that shown in Section 7.4.3, where multiple models can be used to construct a computationally inexpensive, but more accurate model. An alternative approach is to recognize that software licenses are the bottleneck preventing large scale parallel computation. It is possible to re-architecture the EGO process to keep the available licenses fully utilized at all times by converting the meta-model training and infill optimization components of EGO to asynchronous background processes. Every time a license was freed by the completion of a simulation, an infill sample could be proposed using the current state of the meta-models and the current best infill sample. This would minimize the time where licenses are available but not being used.

7.5 Engineering software and computation

7.5.1 Software licenses as the bottleneck to parallel computation

Over the last decade, the cost of powerful computational hardware has plummeted. Moreover, with the rise of cloud computing services such as Amazon EC2, large amounts of this hardware can be rented in increments of an hour or less, enabling data-parallel simulations to be executed cost-effectively in a massively parallel manner. It is possible to construct a cloud based computational cluster that would feature at number 63 on the Top 100 Supercomputer list, for less than 6,000\$ (Trader (2014)).

Unfortunately, there has not been a similar change in pricing models of simulation software. Because it is not possible to purchase licenses on an hourly basis, it is not possible to take advantage of the available hardware. This forces simulations to be executed in a serial or near-serial manner, despite the availability of sufficient hardware for a massively parallel computation.

Were an hourly licensing model available for this work, it would not have been necessary to use the reduced fidelity single stage model. Further, it is likely that the optimization process itself would have been much shorter, making it easier to experiment and innovate therein.

7.5.2 Retraining of predictor models

Although the cost of an EGO iteration is dominated by the time spent running simulations, the cost of training predictor models is not negligible. Most of the models trained in a given iteration are constraint models. It can be observed that it is redundant to retrain a constraint when little extra information is added to the meta-model. It is likely possible to develop a criteria to determine whether meta-models should be retrained based on the meta-model error at the infill points. If all infill points are predicted accurately by a given meta-model, it is likely not necessary to retrain that meta-model, as those points add little information. Similarly, when applying a group of recipes to a constraint dataset, it is probably not necessary to train any more models when a model of greater than a certain threshold of accuracy is trained. Since the most expensive part of the meta-model training process is typically the search for meta-model hyper-parameters, it might be advantageous to simply train a new meta-model using the hyper-parameters of the old. If this meta-model is of acceptable quality, there is no need to undertake an expensive parameter search.

7.5.3 Idempotency in optimization software

Idempotency is the property that $f(f(\dots f(x))) = f(x)$. This implies that when a function is applied, subsequent applications of that function do not have any effect. In the context of software engineering, idempotent means that running the same subroutine multiple times has no effect. This is extremely useful when writing optimization software; idempotent software can be restarted at any time and can recover from failure.

7.6 The new compressor blades

In Chapter 6, the optimal compressor blades are subjected to detailed numerical analysis. It was found that the increase in pressure coefficient obtained on the single stage model could not be realized when the blade was evaluated with the full model. This was attributed to excessive incidence near the blade hub and shroud, caused by the lack of an incidence constraint in those areas. Incidence constraints were intentionally omitted in those areas, in order to avoid feasible designs being excluded due to incorrect reports of flow angle, as discussed in Section 5.2.5. The optimized blade design was manually adjusted, in order to lower the incidence angle. The adjusted blades realized an increase of 8.61% over the original blades when evaluated via the single-stage model and 8.55% over the original blades when evaluated via the full model. Since several blade angles in the adjusted design are limited by the problem bounds, it is possible that there is further opportunity for improvement if these bounds are widened.

7.7 Evaluating the objectives

In this section, the objectives from Section 1.4 are evaluated, in order to determine if they have been met.

1. The exploration of the techniques and processes involved in the application of shape optimization to airfoils.

A variety of different techniques and processes were used in the course of this work. Chapter 2 provided a brief overview of some of the analytic relationships important to understanding axial compressors, along with a description of some past experimental data. A relationship between the pressure rise over a single stage and pressure rise over multiple stages was developed, which was essential in making the computational process tractable. A computational model was developed and validated in Chapter 3. This computational model was combined with the optimization techniques of Chapter 4, in order to produce an optimized blade design. Due to the breadth and scope of techniques used, it is clear that this goal has been achieved.

2. Investigation into several different meta-modeling techniques, their training procedures, strengths and weakness.

In Chapter 4, three types of meta-models were introduced. Section 4.6.2 to Section 4.6.4 discussed response surfaces, kriging and support vector regression respectively. These models were combined with the data transforms of Section 4.5 in order to generate meta-model “recipes”. An automated training function was developed to fit each recipe to arbitrary data. In Chapter 5, these models were used as part of the EGO optimization process, in order to produce optimized blades for the Rofanco compressor. Accordingly, it is reasonable to say that this goal has been achieved.

3. The practical application of efficient global optimization techniques, discussed in Chapter 5.

A powerful and flexible general purpose EGO code was developed. This code was applied to the Rofanco compressor blade design problem in Chapter 5 and used to generate the compressor blades discussed in Chapter 6. The existence of these blades confirms that this objective has been met, despite difficulties in problem formulation unrelated to the optimization process.

4. The design of a new set of blades for the Rofanco compressor.

A new set of blades were designed in Chapter 5 and analyzed in Chapter 6. While a substantial pressure rise was achieved when evaluating these blades using the single stage model, this pressure rise was not sustained when evaluated using the full compressor model. After investigation, this was found to be related to excessive incidence near the hub and shroud.

The design was adjusted manually, in order to lower the incidence angle. The adjusted blades delivered a performance increase of 8.61% over the original blades, when analyzed with the full compressor model. Therefore, it is fair to say that this goal has been achieved.

7.8 Conclusion

In this work, meta-model based design optimization was explored through the medium of designing new compressor blades for the Rofanco compressor, a low speed axial compressor at the Department of Mechanical and Mechatronic Engineering at Stellenbosch University. The blades produced directly by the optimization process did not improve upon those of the Rofanco compressor, due to a discrepancy between the full fidelity model and the approximate model that had to be used in the design process for reasons of computational cost. A small adjustment to the optimum blades produced a new set of compressor blades with a pressure coefficient 8.88% greater than that of the original Rofanco compressor.

Appendices

Appendix A

Analytic calculations relating to the Rofanco compressor

A.1 Introduction

In this section, some analytic calculations relating to the Rofanco compressor are presented. All calculations are performed at mid-span ($r = 0.18$ m), design mass flow rate ($m = 2.7$ kg/s), design speed ($N = 3000$ rpm) and atmospheric inlet conditions.

A.2 Solving the velocity triangles

At the rotor inlet, axial flow is assumed due to the lack of inlet guide vanes. This means that axial velocity and rotational velocity specify the inlet velocity triangle. Axial velocity can be calculated from mass flow:

$$\begin{aligned} C_a &= \frac{m}{\rho A} = \frac{m}{\rho \pi (r_o^2 - r_i^2)} \\ &= \frac{2.7 \text{ kg/s}}{(1.184 \text{ kg/m}^3) \pi ((0.21 \text{ m})^2 - (0.15 \text{ m})^2)} \\ &= 33.61 \text{ m/s} \end{aligned} \quad (\text{A.2.1})$$

Rotational velocity can be calculated from the compressor geometry:

$$U = \omega r = (3000 \text{ rev/min}) \left(\frac{2\pi}{\text{rev}} \right) \left(\frac{\text{min}}{60 \text{ s}} \right) (0.18 \text{ m}) = 56.55 \text{ m/s} \quad (\text{A.2.2})$$

Since the inlet flow angle is zero i.e. $\alpha_1 = 0^\circ$, the rotor inlet flow angle can be calculated from these quantities:

$$\beta_1 = \arctan \left(\frac{U}{C_a} \right) = \arctan \left(\frac{56.55 \text{ m/s}}{33.61 \text{ m/s}} \right) = 59.276^\circ \quad (\text{A.2.3})$$

Solving the rotor outlet/stator inlet velocity triangle requires the estimation of the relative rotor outlet flow angle (β_2). From Table 2.2, the Rofanco compressor blade angles at mid-span are:

$$\zeta_{rotor} = 49.40^\circ \quad \theta_{rotor} = 17.93^\circ \quad \zeta_{stator} = 16.16^\circ \quad \theta_{stator} = 41.05^\circ \quad (\text{A.2.4})$$

Using geometry, it is possible to compute the rotor blade angles from this data:

$$\chi_1 = \zeta_{rotor} + \frac{\theta_{rotor}}{2} = 49.40^\circ + \frac{17.93^\circ}{2} = 58.37^\circ \quad (\text{A.2.5})$$

$$\chi_2 = \zeta_{rotor} - \frac{\theta_{rotor}}{2} = 49.40^\circ - \frac{17.93^\circ}{2} = 40.44^\circ \quad (\text{A.2.6})$$

Since flow does not exactly follow the compressor blades, it is necessary to calculate the rotor deviation angle, from Equation 2.3.3:

$$\delta = \left(0.23 \left(\frac{2a}{l} \right)^2 + 0.1 \left(\frac{\chi_2}{50^\circ} \right) \right) \theta \left(\frac{s}{l} \right)^{\frac{1}{2}} \quad (\text{A.2.7})$$

$$= \left(0.23 + 0.1 \left(\frac{40.44^\circ}{50^\circ} \right) \right) (17.93^\circ) \left(\frac{1}{1.088} \right)^{\frac{1}{2}} \quad (\text{A.2.8})$$

$$= 5.34^\circ \quad (\text{A.2.9})$$

With the rotor outlet blade angle and deviation known, the rotor outlet flow angle may be obtained:

$$\beta_2 = \chi_2 + \delta = 40.44^\circ + 5.34^\circ = 45.78^\circ \quad (\text{A.2.10})$$

With β_2 known, it is possible to calculate the tangential components of the absolute ($C_{2,t}$) and relative ($W_{2,t}$) rotor outlet velocity:

$$W_{2,t} = C_a \tan \beta_2 = (33.61 \text{ m/s}) \tan 45.78^\circ = 34.54 \text{ m/s} \quad (\text{A.2.11})$$

$$C_{2,t} = U - W_{2,t} = 56.55 \text{ m/s} - 34.54 \text{ m/s} = 22.01 \text{ m/s} \quad (\text{A.2.12})$$

With the tangential component of the absolute rotor velocity known, it is possible to calculate the absolute rotor outlet angle:

$$\alpha_2 = \arctan \left(\frac{C_{2,t}}{C_a} \right) = \arctan \left(\frac{22.01 \text{ m/s}}{33.61 \text{ m/s}} \right) = 33.23^\circ \quad (\text{A.2.13})$$

A.3 Calculating flow quantities of interest

Using the basic quantities solved in the previous section, the flow coefficient may be calculated from its definition:

$$\phi = \frac{C_a}{U} = \frac{33.61 \text{ m/s}}{56.55 \text{ m/s}} = 0.594 \quad (\text{A.3.1})$$

The reaction ratio may be calculated from Equation 2.2.5:

$$\begin{aligned} R &= \frac{1}{2} (1 + \phi(\tan \beta_2 - \tan \alpha_1)) \\ &= \frac{1}{2} (1 + (0.594) (\tan (45.78^\circ) - \tan (0^\circ))) \quad (\text{A.3.2}) \\ &= 0.805 \end{aligned}$$

It can be shown that the stage loading coefficient is given by the relation below (Dixon (2005)).

$$\begin{aligned} \Psi &= \lambda \phi (\tan \alpha_2 - \tan \alpha_1) \\ &= (0.92) (0.594) (\tan 33.23^\circ - \tan 0^\circ) \quad (\text{A.3.3}) \\ &= 0.358 \end{aligned}$$

Appendix B

Introduction to optimization and meta-modeling

B.1 Introduction

In this section, several meta-modeling concepts relevant to the work of Chapter 4 are discussed.

B.2 R^2 in the context of response surfaces

The most basic measure of model quality is the sum-of-squares of the residuals, shown below (Myers *et al.* (2009)). The original data is denoted by y and \hat{y} is the approximation.

$$SS_E = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n e_i^2 = \mathbf{y}^T \mathbf{y} - \mathbf{b}^T \mathbf{X}^T \mathbf{y} \quad (\text{B.2.1})$$

Related to the error sum-of-squares are the regression sum-of-squares, SS_R , and the total sum-of-squares, SS_T . SS_R is a measure of how much variance is explained by a model and SS_T is a measure of the variance in the data. According to Myers *et al.* (2009), these quantities are given by the following relations for response surfaces:

$$SS_R = \mathbf{b}^T \mathbf{X}^T \mathbf{y} - \frac{\left(\sum_{i=1}^n y_i \right)^2}{n} \quad (\text{B.2.2})$$

$$SS_T = SS_R + SS_E = \mathbf{y}^T \mathbf{y} - \frac{\left(\sum_{i=1}^n y_i \right)^2}{n} \quad (\text{B.2.3})$$

The R^2 coefficient is a normalized measure of the ratio of variance in the model to the variance explained by the model:

$$R^2 = 1 - \frac{SS_E}{SS_T} \quad (\text{B.2.4})$$

While R^2 provides useful evidence as to how well the model predicts the current data, it is more important to evaluate how well the approximation will generalize, i.e. how well it will predict new data. This may be evaluated using the technique of cross-validation, explained in Section 4.4.2. The cross-validation error sum of squares is referred to as PRESS and is analogous to SS_E .

$$PRESS = \sum_{i=1}^n (e_{-i})^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (\text{B.2.5})$$

Fortunately, for response surfaces the model does not have to be repeatedly retrained for each reduced data set in order to calculate the cross-validation error. It may be shown that the cross validation error may be related to the residual at a particular data-point using the hat matrix, $H = X(X^T X)^{-1} X^T$; see Myers *et al.* (2009) for more details.

$$PRESS = \sum_{i=1}^n \left(\frac{e_i}{1 - h_{ii}} \right)^2 \quad (\text{B.2.6})$$

The $PRESS$ value may be used to compute the prediction R^2 coefficient (R_{pred}^2). This quantity is analogous to R^2 , but based on cross-validation error and provides a measure of the generalizability of the model.

$$R_{pred}^2 = 1 - \frac{PRESS}{SS_T} \quad (\text{B.2.7})$$

B.3 Kriging correlation and regression models

The Python library used to implement kriging is `scikit-learn` (Pedregosa *et al.* (2011)). The implementation of kriging in this library is based on that of the MATLAB[®] toolbox DACE. Accordingly, the reader is referred to the technical report on DACE by Lophaven *et al.* (2002). In this section, the regression and correlation models used in this work are discussed.

B.3.1 Regression models

The regression model provides f_i , the components of the kriging model in Equation 4.6.6. In this work, all three regression models available in `scikit-learn` are used. The models used include the constant model, the linear model and the quadratic model. They are summarized in Table B.1.

Table B.1: In this table, the kriging regression models from `scikit-learn` are summarized. The formulation is from Lophaven *et al.* (2002), upon which the implementation of `scikit-learn` is based.

Model	Number of terms	Equations
Constant	$p = 1$	$f_1(x) = 1$
Linear	$p = n + 1$	$f_1(x) = 1$ $f_2(x) = x_1 \dots f_{n+1}(x) = x_n$
Quadratic	$p = \frac{1}{2}(n + 1)(n + 2)$	$f_1(x) = 1$ $f_2(x) = x_1 \dots f_{n+1}(x) = x_n$ $f_{n+2}(x) = x_1^2 \dots f_{2n+1}(x) = x_1 x_n$ $f_{2n+2}(x) = x_2^2 \dots f_{3n}(x) = x_2 x_n$ \dots $f_p(x) = x_n^2$

B.3.2 Correlation models

The elements R_{ij} of the correlation matrix \mathbf{R} are of the form below, where m is the number of training samples :

$$R_{ij} = \mathcal{R}(\theta, \mathbf{s}_i, \mathbf{s}_j), i, j = 1 \dots m \quad (\text{B.3.1})$$

The Python package, `scikit-learn`, uses correlations of the form given below, where n is the number of dimensions in the design space:

$$\mathcal{R}(\theta, \mathbf{a}, \mathbf{b}) = \prod_{k=1}^n \mathcal{R}_k(\theta, a_k, b_k) \quad (\text{B.3.2})$$

In this work, two choices of correlation function, \mathcal{R} , are investigated. They are the squared exponential function:

$$\mathcal{R}_k(\theta, a_k, b_k) = \exp(-\theta_k \|a_k - b_k\|^2) \quad (\text{B.3.3})$$

As well as the cubic function, with $\xi = \min\{1, \theta_k \|a_k - b_k\|\}$:

$$\mathcal{R}_k(\theta, a_k, b_k) = 1 - 3\xi^2 + 2\xi^3 \quad (\text{B.3.4})$$

These functions are the recommended correlation functions in `scikit-learn` for smooth data.

B.4 Some efficient global optimization infill criteria

In this section, some efficient global optimization infill criteria are discussed, based on the discussion of Sasena *et al.* (2002). The first and most well known

infill criteria is expected improvement, which is discussed in Section 4.7.2 and will not be discussed here. Expected improvement was generalized by Schonlau (1998), who introducing a parameter, g , which controls the trade off between a local and global search. Generalized expected improvement is given by:

$$E(I^g) = s^g \sum_{k=0}^g (-1)^k \left(\frac{g!}{k!(g-k)!} \right) (f'_{min})^{g-k} T_k \quad (\text{B.4.1})$$

Where f'_{min} and T_k are given by:

$$f'_{min} = (f_{min}^n - \hat{y}) / \hat{s} \quad (\text{B.4.2})$$

$$T_k = -\phi(f'_{min}) (f'_{min})^{k-1} + (k-1) T_{k-2} \quad (\text{B.4.3})$$

The bases of the recursion in T_k are $T_0 = \Phi(f'_{min})$ and $T_1 = -\phi(f'_{min})$.

Watson and Barnes (1995) proposed two infill criteria with the aim of finding extreme values. The first of these is the threshold bounded extreme criterion, which was developed with the objective of maximizing the probability that the sample in question exceeded some threshold. The threshold bounded extreme (TBE) criterion is given by:

$$\text{TBE} = \Phi\left(\frac{f_{min}^n - \hat{y}}{\hat{s}}\right) \quad (\text{B.4.4})$$

The second criterion from Watson and Barnes (1995) is the regional extreme (RE) criterion, which aims to minimize the expected value of the smallest observation.

$$\text{RE} = \begin{cases} \hat{y} + (f_{min}^n - \hat{y}) \Phi(f'_{min}) + \hat{x} \phi(f'_{min}) & \text{if } \hat{s} > 0 \\ 0 & \text{if } \hat{s} = 0 \end{cases} \quad (\text{B.4.5})$$

The last criteria used in this work is the maximum variance (MV) criteria, which places samples in the areas of highest uncertainty:

$$\text{MV} = \max \{ \text{Var}(x) \} \quad (\text{B.4.6})$$

B.5 Optimization algorithms used in this work

B.5.1 Multi-objective GA

The latin hypercubes used in this work are generated by a genetic algorithm (GA) implemented as part of this work. The implementation of the GA used in this process is uninteresting, since it only incorporates the straight forward effects of mutation, crossover and elitist selection. It is however worth discussing how candidate latin-hypercubes are compared, since it strongly influences the performance of the optimizer.

When comparing two latin hypercubes, the following three rules are used to determine which is better:

- Any solution that is a valid latin hypercube is better than a solution that is not.
- A latin hypercube with a greater minimum distance between samples is better than a latin hypercube with a lesser minimum distance between samples.
- If two latin hypercubes have equal minimum distances between samples, then the latin hypercube with greater average distance between samples is better.

The first rule is straightforward and the second rule is implicit in the definition of an optimal latin hypercube. The third rule is a tie-breaking heuristic, which ranks latin hypercubes with a higher average distance between samples higher than those with a lower average distance. The reasoning behind this is that members of the population with a higher average distance between samples have a greater probability of having a higher minimum distance when perturbed via mutation or crossover. This third rule substantially improves the performance of the GA, when attempting to find optimal latin hypercubes.

B.5.2 PSO

Particle swarm optimization (PSO) is used to solve the infill optimization problem, as it excels in solving highly multi-modal problems. Since the infill optimization problem is central to implementing efficient global optimization, it is worth taking some time to discuss the PSO algorithm implemented as part of this work.

The PSO algorithm was originally proposed by Eberhart and Kennedy (1995). It is based on a swarm of n particles, with position \mathbf{x} , moving around the design space with velocity, \mathbf{v} , constrained to be between \mathbf{v}_{min} and \mathbf{v}_{max} . The position and velocity in the k -th iteration are given by the equations below, where \mathbf{p}_k^{local} is the best point yet found by the current particle at iteration k and \mathbf{p}_k^{global} is the best point yet found by all samples at iteration k .

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1} \quad (\text{B.5.1})$$

$$\mathbf{v}_{k+1} = \omega \mathbf{v}_k + c_1 r_1 (\mathbf{p}_k^{local} - \mathbf{x}_k) + c_2 r_2 (\mathbf{p}_k^{global} - \mathbf{x}_k) \quad (\text{B.5.2})$$

Several enhancements are made to PSO. The first of these is the combination of the elite particle and elite velocity heuristics proposed by Fourie and Groenwold (2002). The elite particle heuristic replaces the worst particle with a particle centered on the global optima. The elite velocity heuristic proposes that when a new optima is found, the velocity in the next iteration be set to

some multiple of that previous velocity for that particle i.e. $\mathbf{v}_{k+1} = a\mathbf{v}_k$ where $a \in [0, 1]$.

Other heuristics used in this work are center particle, proposed by Liu *et al.* (2007) and craziness, dynamic velocity reduction and dynamic inertia reduction, proposed by Fourie and Groenwold (2000). The center particle heuristic adds a special particle, called the center particle, whose position is set to the average of all the particles. The craziness heuristic adds a random velocity between v_{min} and v_{max} to a given particle, with probability p . The dynamic velocity and inertia reduction heuristics reduce the velocity and inertia of particles when an improvement is not found within h iterations. Putting this into mathematical form, with $\alpha, \beta \in [0, 1]$:

$$\text{if } f(p_k^{global}) \geq f(p_{k+h}^{global}) \quad \text{then } \omega_{k+1} = \alpha\omega_k, v_{k+1}^{max} = \beta v_k^{max} \quad (\text{B.5.3})$$

Appendix C

An efficient global optimization example problem

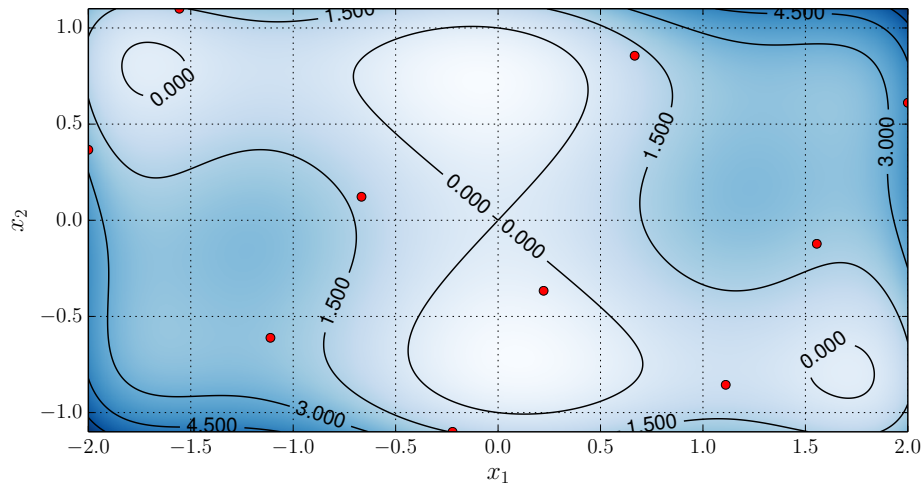
C.1 Introduction

In this section, an EGO example problem is used to demonstrate the meta-modeling and EGO concepts presented in Chapter 4. The aim of this example is to demonstrate the application of the EGO concepts discussed and to validate the developed optimization tools. This problem is also used to introduce the standard meta-model report which contains information relating to the quality of fit and hyper-parameter choice of each of the meta-models.

C.2 Problem description

The example problem is the constrained optimization of the six-humped camel back function, described in Equation C.2.1. In the domain of interest, this function possess four local minima. The global minima is repeated at $(a, -b)$ and $(-a, b)$ where $a = 0.089842$ and $b = 0.712656$ (Molga and Smutnicki (2005)). Constraint g_1 is active at the global optimum. The contours of the objective and constraint functions are shown in Figure C.1a, Figure C.1b and Figure C.1c respectively.

$$\begin{aligned}
 \min \quad & f(x_1, x_2) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + 4x_2^2(-1 + x_2^2) \\
 \text{Subject to: } & g_1(x_1, x_2) = \text{sgn}\left(\frac{x_1b}{2a}\left(\frac{x_1^2}{a^2} - 3\right) - x_2\right) \leq 0 \\
 & g_2(x_1, x_2) = \leq 0 \\
 & -2 \leq x_1 \leq 2 \\
 & -1.1 \leq x_2 \leq 1.1
 \end{aligned} \tag{C.2.1}$$



(a) The six-hump camelback objective function

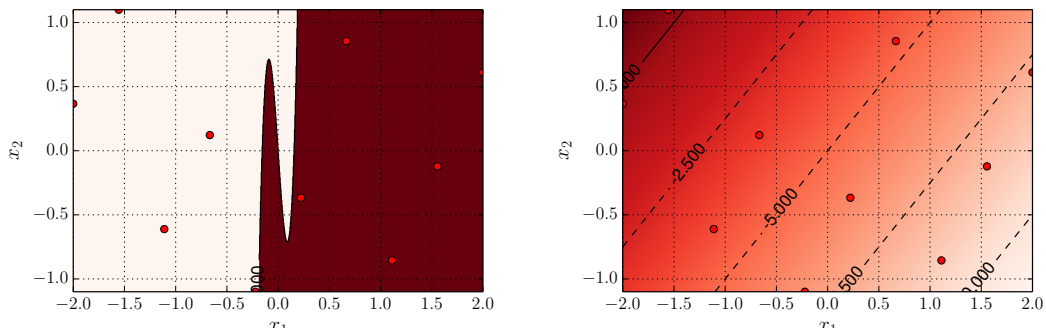
(b) The classification constraint function (g_1). (c) The regression constraint function (g_2).

Figure C.1: The example problem objective and constraint functions. The samples used to train the initial meta-models are also shown.

C.3 The optimization process

A 15-level latin hypercube was used to generate the initial sample set. A software library was created which completely automates the process of fitting meta-models to arbitrary data. The central concept of this library is model “recipes”, which consist of a chain of predefined preprocessing operations and meta-model. As an example, the recipe “powerscale_3 scale svr-rbf” specifies that data be transformed using the Box-Cox transform with $\lambda = 3$, scaled and then fitted using Gaussian kernel support vector regression. Some transforms and meta-models are listed in Tables 4.1 and 4.2.

Aside from training meta-models, the software library also produces a number of plots and tables for each model it trains, so that it is possible to access the quality of fits. This information is used by another software component developed as part of this work to produce a report containing information about

APPENDIX C. AN EFFICIENT GLOBAL OPTIMIZATION EXAMPLE PROBLEM

99

Table C.1: The number and variety of infill samples used in the example problem.

Technique	Count	Aim
Maximum variance	1	Improve model quality by reducing variance
Expected improvement	2	Maximize the probability of improving the optimum
Meta-model optima	2	Sample the predicted optimum, in order to realize performance gains
Total	5	

Table C.2: The choice of models used in the example problem.

Response surfaces	SVM	Kriging
scaled rs	scaled svr	scaled kriging
scaled quad-rs	scaled pca svr	scaled pca kriging
scaled pca quad-rs		

the best variant of each of the meta-models described in Section 4.6.

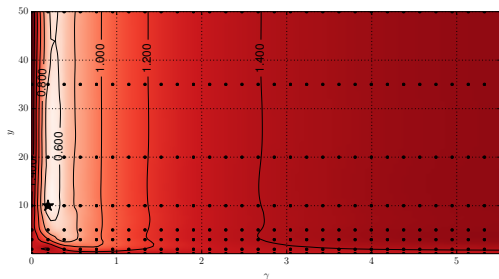
A report for the objective function for the first EGO iteration of the example problem is shown in Figure C.2. Several useful metrics and plots are shown in this report. Table a) lists the best recipe for each type of meta-model as well as its estimated error. Figures b) and c) are contour plots of MSE as a function of γ and C for the two levels of support vector regression parameter grid search. The selected values of C and γ are indicated by a star. Figures e), f) and g) are plots of the actual versus predicted value for each meta-model variant. Data points closer to the line $y = \hat{y}$ have lower error. Figure h) plots the standardized residuals of the best kriging variant against the model prediction. Clustering of samples in this plot typically indicates some sort of systemic over- or under-prediction of the data. The histogram should appear to be Gaussian distribution and centered around zero. Finally, Figure i) is a plot of cross-validation error as a function of θ for each combination of kriging regression and correlation model studied. The best combination is indicated by bold font in the legend.

The meta-model recipes used in this example problem are shown in Table C.1. The number and type of infill samples used are shown in Table C.2. The infill criteria are optimized using the PSO algorithm described in Appendix B.5.2.

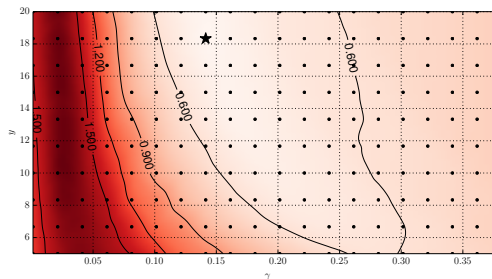
Regression report: f (iteration 1)

(a) Summary of meta-model error

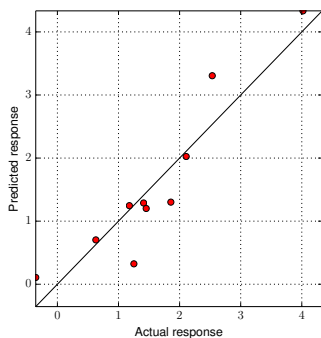
Recipe	ϵ_{CV}	R^2
krig	0.4337	0.8455
scaled svr-rbf	0.6765	0.6239
rs	0.4661	0.8215



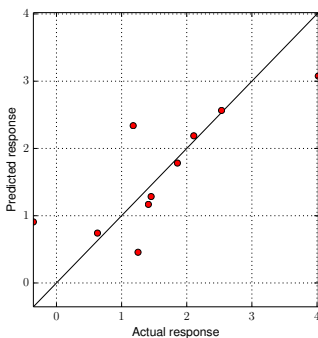
(c) SVR parameter search level 1



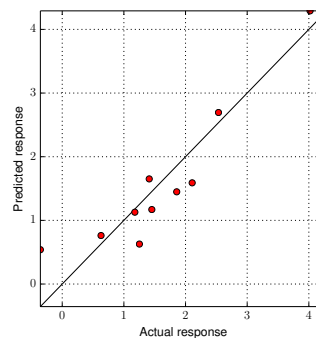
(d) SVR parameter search level 2



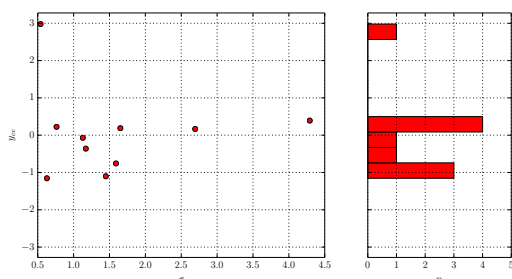
(e) RS vs. actual value



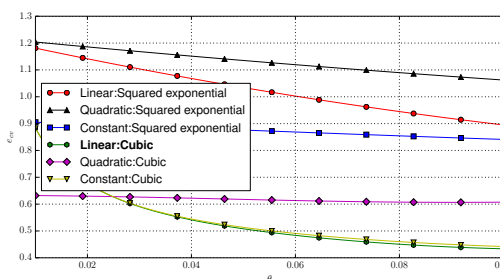
(f) SVR vs. actual value



(g) Kriging vs. actual value



(h) Kriging standardized residuals



(i) Kriging error for various parameters

Figure C.2: An example report for the objective function during the first EGO iteration.

C.4 Results

C.4.1 The kriging model and expected improvement

The kriging approximation for the objective function for the first EGO iteration is shown in Figure C.3. Comparing Figure C.3 to Figure C.1a shows that the kriging meta-model is a reasonable approximation of the original function in Equation C.2.1. The expected improvement and infill samples are plotted in Figure C.4. When optimizing infill, the meta-model variant with the highest R^2 is used to predict each constraint model. PSO is used to optimize the infill criteria and is discussed in Appendix B.

This accounts for the fact that the infill samples are slightly shifted from the peaks of expected improvement - the peaks fall into constrained regions. The two infill samples generated by optimizing expected improvement are combined with the two generated by optimizing maximum variance and the optimum of the best predictor of the objective function to form the set of infill samples that are added to the training samples of the next iteration.

C.4.2 Comparing actual and predicted model error

The cross-validation predicted model R^2 and the actual R^2 , calculated by comparing the prediction against the known objective function, are shown in Figure C.5. The actual accuracy of the kriging and support vector regression models tend to increase as iterations are completed, while the actual accuracy of the response surface model decreases. This is likely due to the inability of the response surface model to capture the complex behavior of Equation C.2.1. Overall, the kriging meta-model predicts the underlying function most accurately, while the response surface meta-model is least accurate. It is interesting

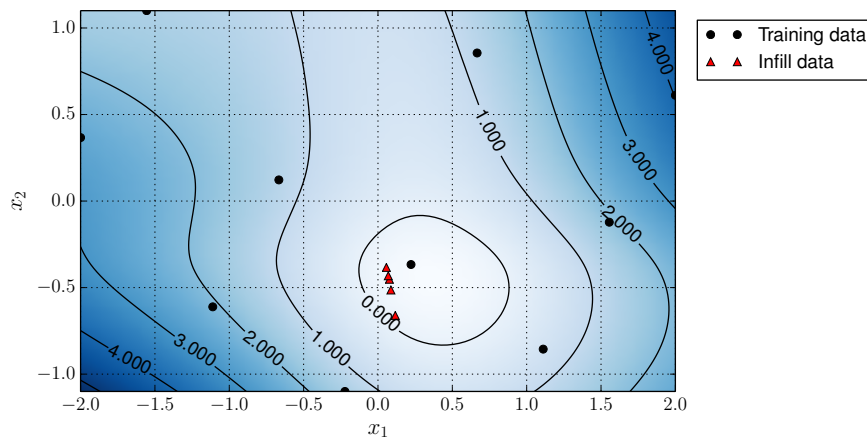


Figure C.3: The kriging approximation to the objective function given in Equation C.2.1, during the first iteration.

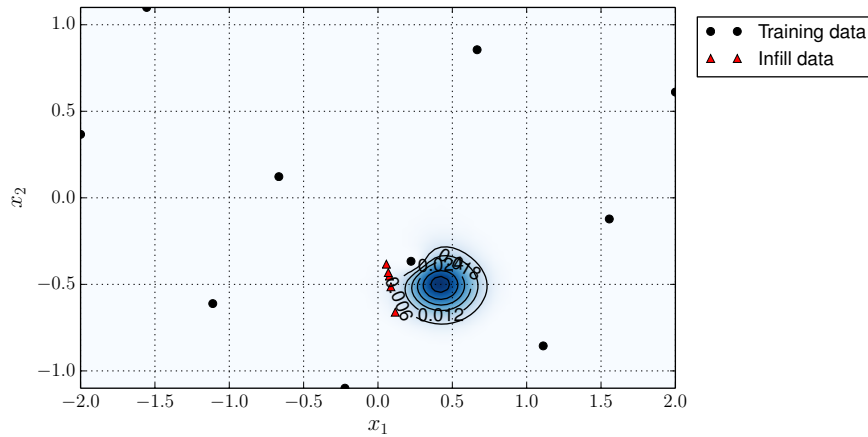


Figure C.4: The expected improvement contours for the first iteration, calculated from the kriging model in Figure C.3. The red samples are the initial latin hypercube set, while the black samples are infill samples. Note how close the infill samples are to the model optima, shown in Figure C.1a.

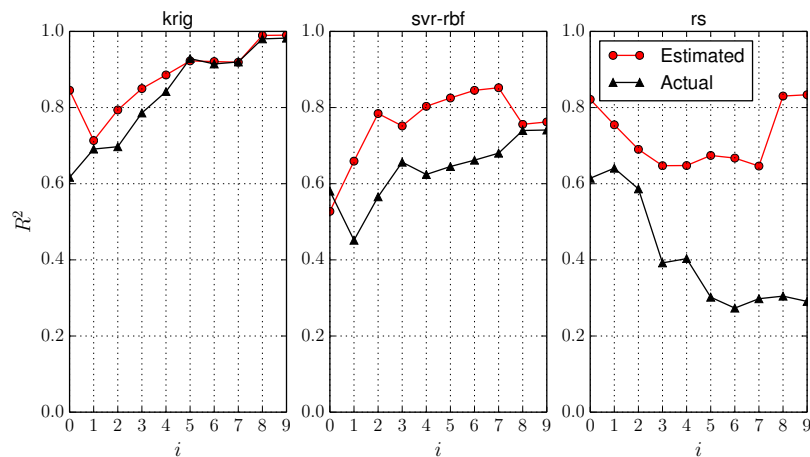
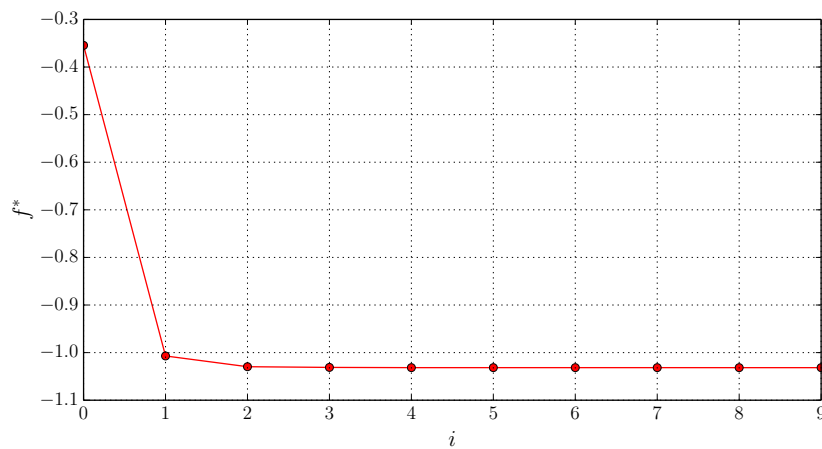
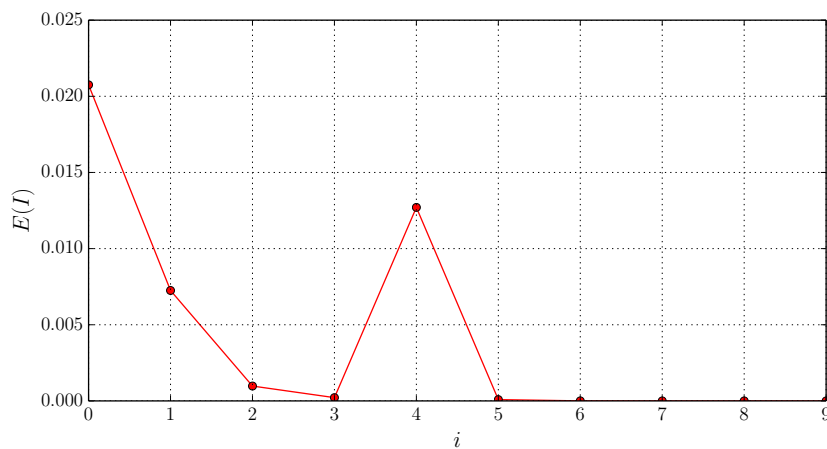


Figure C.5: The actual and predicted cross-validation R^2 , for each of the three basic meta-models. It is interesting to note that the actual R^2 value observed for the kriging and support vector regression models increases during the process, while that of the response surfaces decreases. This is likely due to the inability of the response surface to capture the underlying function.

to observe that aside from the first iteration of support vector regression, no meta-models substantially over predicts their capacity to model the underlying function.



(a) The best solution as a function of iterations



(b) Expected improvement as a function of iterations

Figure C.6: The best sample found and expected improvement, as a function of the optimization iterations completed. Note the correspondence between high expected improvement and a new model optima, for all iterations but iteration four.

C.4.3 Expected improvement and optima during the optimization process

Figures C.6a and C.6b plot the optimum value found and expected improvement as a function of the number of EGO iterations. These quantities decrease proportionally, aside from an unexpected spike in expected improvement in iteration four. This spike does not correspond to an improvement in the model optima.

Appendix D

Optimization problem detailed results

Table D.1: The blade angles for the sampled design points and their identity numbers. The objective and constraint values are found in Table D.2.

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
0	53.52°	22.48°	43.52°	-2.76°	58.36°	40.44°	37.14°	-3.91°	61.55°	50.65°	34.32°	-5.68°
1	61.84°	13.49°	47.3°	0.01°	55.26°	48.08°	40.75°	-12.74°	63.48°	46.53°	42.98°	-4.76°
2	48.73°	24.24°	52.18°	1.86°	64.67°	48.75°	40.58°	-12.57°	67.18°	43.34°	25.66°	-7.61°
3	56.63°	15.34°	44.95°	2.87°	54.58°	34.8°	39.07°	-6.35°	67.85°	54.43°	32.72°	-10.97°
4	52.76°	17.69°	35.7°	1.69°	49.04°	46.91°	46.8°	1.88°	53.23°	60.31°	44.15°	-4.25°
5	59.32°	21.22°	45.96°	-10.24°	64.16°	45.39°	29.15°	-11.23°	65.5°	52.08°	24.99°	-8.45°
6	52.26°	26.09°	40.07°	-2.17°	66.35°	49.26°	44.78°	4.74°	59.11°	47.04°	34.4°	-9.63°
7	58.98°	20.04°	48.48°	-12.42°	61.98°	47.75°	41.25°	5.41°	63.15°	57.96°	28.19°	-3.07°
8	49.57°	18.03°	49.82°	-5.53°	57.44°	34.64°	44.11°	-4.67°	64.32°	57.79°	30.87°	-9.8°
9	60.49°	23.4°	34.19°	-5.7°	64.33°	32.96°	40.08°	-0.47°	70.21°	43.0°	28.35°	-13.16°
10	53.94°	16.35°	43.77°	-5.87°	63.15°	31.28°	37.05°	4.4°	70.54°	42.33°	27.18°	-11.48°
11	56.97°	25.09°	41.92°	-6.71°	51.73°	44.05°	36.55°	-0.64°	57.6°	53.09°	34.57°	-15.18°
12	46.88°	19.54°	37.72°	-2.68°	62.15°	35.31°	47.14°	-2.99°	65.16°	47.71°	31.71°	0.29°
13	49.74°	24.75°	45.79°	-10.74°	55.76°	32.12°	38.73°	3.23°	62.81°	53.93°	24.49°	-9.97°
14	61.17°	28.95°	48.65°	7.07°	51.05°	33.46°	43.44°	-9.04°	62.64°	51.57°	33.06°	-6.44°
15	47.55°	22.4°	40.75°	4.21°	62.65°	36.99°	37.89°	1.38°	71.05°	48.21°	42.64°	-11.31°
16	47.39°	21.39°	42.6°	5.39°	54.42°	50.1°	45.96°	-8.2°	59.79°	59.31°	27.68°	-0.89°
17	54.61°	13.15°	34.36°	2.2°	59.12°	48.25°	36.21°	-7.02°	56.26°	51.74°	41.8°	-10.64°
18	47.89°	12.82°	38.23°	-12.26°	67.02°	46.23°	35.37°	-3.33°	69.7°	52.75°	30.71°	-15.51°
19	52.93°	28.78°	48.98°	6.06°	66.85°	36.15°	31.17°	3.73°	66.17°	56.78°	40.12°	-0.22°
20	63.18°	16.51°	53.02°	-3.35°	59.29°	38.0°	42.77°	-1.14°	71.55°	49.73°	34.24°	-14.84°
21	44.53°	16.68°	46.8°	-6.21°	56.43°	40.01°	27.47°	-0.13°	55.25°	59.64°	29.53°	-4.92°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
22	57.81°	14.5°	45.45°	-0.66°	54.25°	47.07°	38.56°	-11.06°	65.67°	40.65°	28.86°	2.47°
23	62.18°	16.18°	35.2°	-2.84°	65.51°	49.59°	33.02°	-1.98°	66.34°	46.87°	24.82°	-13.83°
24	58.48°	24.58°	42.26°	-10.41°	48.87°	39.51°	42.43°	-2.82°	70.04°	57.62°	40.79°	2.98°
25	62.01°	21.72°	49.65°	-7.55°	63.83°	39.85°	45.45°	4.07°	52.05°	44.18°	37.26°	0.45°
26	61.0°	28.11°	46.46°	-11.92°	65.34°	47.58°	29.32°	-7.36°	54.58°	40.82°	36.76°	-4.59°
27	51.08°	18.36°	44.28°	-1.16°	56.26°	32.28°	32.35°	2.72°	53.9°	51.24°	35.24°	-12.65°
28	50.75°	30.8°	51.34°	-11.25°	59.96°	40.18°	44.95°	1.04°	63.31°	43.68°	32.39°	-6.6°
29	58.14°	12.98°	39.07°	-2.51°	49.54°	34.47°	27.14°	-13.91°	55.42°	44.35°	31.21°	0.96°
30	48.56°	17.35°	47.64°	-1.67°	58.28°	34.3°	28.98°	0.37°	68.86°	58.3°	26.0°	-12.99°
31	54.78°	17.02°	43.94°	3.88°	67.19°	40.86°	34.87°	-13.58°	69.37°	41.15°	37.43°	-1.39°
32	63.02°	19.87°	36.04°	-4.86°	53.24°	50.27°	30.5°	-8.03°	58.61°	50.4°	38.44°	-5.26°
33	63.52°	30.46°	51.5°	-1.84°	48.7°	41.7°	35.54°	-0.3°	64.66°	47.2°	31.38°	-12.49°
34	56.46°	22.73°	39.23°	-6.37°	52.4°	39.17°	30.33°	-1.81°	59.62°	49.22°	44.32°	-2.91°
35	51.92°	14.66°	45.12°	4.72°	60.13°	49.76°	39.4°	1.21°	55.75°	41.99°	36.08°	-10.13°
36	63.35°	14.83°	41.76°	1.53°	67.86°	31.61°	39.57°	-10.89°	66.84°	56.62°	38.77°	-1.56°
37	48.39°	31.3°	46.63°	-7.89°	68.03°	44.72°	36.04°	3.4°	70.71°	46.36°	37.77°	-8.29°
38	59.65°	27.27°	46.29°	0.18°	58.95°	33.29°	36.88°	-4.5°	52.89°	45.69°	43.98°	-3.92°
39	45.7°	31.81°	33.86°	-6.04°	53.41°	45.06°	30.66°	-10.55°	59.28°	57.46°	26.67°	-2.07°
40	53.44°	17.52°	50.16°	4.55°	59.79°	46.4°	31.34°	-0.81°	63.65°	42.16°	28.52°	1.97°
41	45.37°	15.84°	48.81°	2.7°	66.68°	35.65°	33.86°	-4.17°	53.73°	59.14°	37.6°	-1.9°
42	55.12°	24.08°	50.83°	2.53°	64.84°	45.22°	37.39°	5.58°	54.41°	55.78°	27.85°	-6.27°
43	57.3°	23.07°	36.38°	4.38°	53.58°	46.57°	41.08°	5.24°	55.08°	58.8°	26.17°	-5.09°
44	62.85°	24.41°	50.33°	-3.68°	61.31°	32.45°	34.03°	5.75°	60.96°	49.89°	43.65°	-6.94°
45	51.42°	23.57°	33.52°	-3.01°	58.79°	44.22°	33.19°	-5.51°	69.2°	56.11°	33.73°	3.82°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
46	53.6°	19.37°	50.66°	-8.56°	60.97°	42.87°	44.61°	-9.55°	68.52°	45.52°	26.84°	-11.81°
47	45.2°	15.67°	50.49°	-6.54°	49.71°	31.95°	32.01°	-8.87°	60.29°	57.12°	29.7°	-3.24°
48	58.81°	23.74°	43.6°	-3.18°	52.57°	36.49°	33.35°	0.71°	57.77°	40.99°	37.09°	-7.95°
49	50.07°	13.66°	51.67°	-7.05°	49.37°	48.42°	27.81°	-7.86°	62.31°	46.2°	42.3°	-5.76°
50	58.31°	28.45°	34.53°	-8.73°	65.17°	49.09°	41.76°	0.87°	71.21°	49.05°	37.93°	-3.75°
51	60.33°	32.31°	44.78°	-10.58°	52.06°	33.8°	40.41°	4.91°	53.57°	57.29°	43.82°	-7.28°
52	51.76°	21.89°	38.73°	6.57°	61.14°	40.69°	31.5°	3.56°	57.94°	42.67°	41.63°	-9.29°
53	48.23°	15.51°	36.21°	0.52°	48.36°	46.74°	32.68°	-6.02°	68.19°	41.66°	40.62°	2.3°
54	44.36°	12.48°	44.44°	2.03°	60.8°	35.48°	45.29°	1.72°	55.92°	45.86°	32.22°	3.31°
55	56.8°	14.16°	42.09°	-8.39°	50.05°	31.78°	43.77°	-6.86°	66.51°	44.52°	29.19°	-13.5°
56	61.34°	25.25°	41.42°	-12.76°	57.94°	37.16°	46.63°	4.57°	65.0°	48.04°	41.29°	4.15°
57	59.15°	22.23°	53.35°	3.04°	63.49°	44.55°	46.97°	-7.19°	60.46°	50.06°	33.9°	1.63°
58	53.1°	19.03°	36.88°	-10.07°	54.08°	47.24°	46.46°	-12.91°	69.53°	59.81°	32.89°	-14.34°
59	47.72°	32.14°	40.91°	-7.38°	62.99°	33.63°	46.13°	-6.18°	60.79°	59.47°	39.45°	-7.44°
60	52.43°	18.7°	49.49°	2.37°	60.47°	45.9°	31.0°	-8.7°	53.4°	41.83°	33.23°	-5.6°
61	47.05°	14.33°	52.85°	-1.0°	52.73°	40.35°	36.71°	-10.22°	64.83°	56.28°	25.16°	-0.05°
62	52.6°	20.72°	41.08°	5.73°	65.84°	36.65°	44.28°	-3.49°	60.63°	55.27°	31.04°	-15.68°
63	57.64°	25.42°	47.81°	7.24°	63.66°	43.71°	29.49°	2.56°	66.0°	53.42°	41.13°	-14.5°
64	56.13°	20.88°	34.7°	0.35°	50.72°	43.21°	38.23°	2.89°	64.16°	43.84°	41.46°	4.32°
65	43.69°	25.93°	38.56°	-7.72°	50.89°	43.88°	38.4°	-10.39°	52.39°	45.36°	41.97°	3.14°
66	44.02°	27.44°	40.58°	3.37°	57.27°	41.02°	33.69°	-5.34°	54.07°	44.68°	26.5°	-7.78°
67	57.47°	32.48°	52.51°	-1.33°	49.21°	39.34°	35.03°	-9.88°	67.68°	50.9°	39.78°	-15.34°
68	49.23°	15.0°	47.97°	-5.2°	56.6°	33.96°	43.27°	-5.68°	53.06°	44.01°	36.42°	3.65°
69	44.19°	16.01°	47.47°	4.05°	67.69°	37.66°	34.19°	-11.56°	56.76°	48.72°	38.1°	-11.14°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
70	58.65°	26.26°	40.41°	-9.57°	62.82°	43.04°	29.66°	2.05°	60.12°	55.1°	33.56°	-13.66°
71	60.83°	19.2°	39.4°	6.4°	65.68°	37.83°	33.52°	-3.83°	56.42°	54.77°	38.61°	-2.57°
72	62.51°	30.63°	37.55°	0.85°	68.2°	41.86°	40.24°	-5.85°	61.8°	41.49°	25.33°	2.64°
73	46.04°	18.87°	49.32°	-0.49°	52.9°	33.12°	44.45°	-8.54°	52.56°	55.44°	34.07°	-5.43°
74	54.44°	26.93°	52.01°	-1.5°	59.63°	39.01°	31.67°	-10.05°	52.22°	58.97°	30.2°	-2.74°
75	57.97°	17.19°	42.43°	-6.88°	56.94°	41.53°	45.79°	-5.01°	69.03°	52.92°	29.03°	-0.39°
76	46.71°	23.91°	46.97°	-4.19°	67.36°	47.91°	41.42°	-9.21°	63.82°	47.88°	40.96°	1.46°
77	56.29°	16.85°	37.22°	-4.69°	53.91°	43.38°	42.09°	-11.73°	58.94°	54.94°	35.08°	-9.13°
78	55.28°	19.71°	49.15°	5.56°	58.11°	38.33°	35.2°	-2.49°	67.52°	42.83°	31.88°	-12.15°
79	50.91°	30.97°	51.84°	6.23°	52.23°	41.19°	39.91°	-11.39°	55.58°	53.26°	27.51°	-0.55°
80	55.79°	27.1°	52.34°	-7.21°	58.62°	50.44°	32.51°	-1.31°	71.38°	43.17°	42.81°	0.62°
81	49.91°	18.19°	44.11°	3.21°	54.75°	31.44°	42.6°	1.55°	59.95°	56.45°	38.94°	-2.23°
82	46.55°	23.24°	35.87°	-4.02°	53.07°	36.82°	30.83°	-3.16°	65.84°	56.95°	27.01°	-12.32°
83	51.25°	27.94°	33.69°	-11.75°	51.89°	30.6°	43.1°	-12.4°	61.97°	48.55°	42.47°	-10.47°
84	45.87°	20.21°	41.25°	-3.85°	57.61°	41.36°	29.82°	5.08°	65.33°	60.15°	35.58°	-8.12°
85	50.58°	30.3°	34.02°	-9.9°	55.59°	42.54°	28.65°	-13.41°	70.37°	43.51°	39.11°	-5.93°
86	43.52°	27.77°	37.39°	-9.23°	64.5°	49.43°	31.84°	-10.72°	61.47°	53.76°	28.02°	-8.96°
87	54.95°	12.65°	35.37°	6.74°	57.78°	39.68°	37.72°	3.06°	58.27°	44.85°	30.54°	1.29°
88	57.13°	22.06°	52.68°	-8.89°	58.45°	37.49°	35.71°	-2.65°	61.3°	51.07°	43.14°	1.8°
89	45.03°	31.47°	45.62°	-4.52°	66.52°	47.41°	40.92°	0.03°	64.49°	52.41°	32.56°	2.81°
90	60.16°	13.32°	51.0°	-5.03°	48.53°	31.11°	42.26°	0.2°	63.99°	45.19°	27.35°	-3.58°
91	49.07°	21.05°	39.91°	-3.52°	68.36°	32.79°	32.85°	-7.7°	62.47°	50.23°	36.59°	-14.0°
92	46.21°	29.45°	49.99°	0.69°	60.3°	48.92°	37.22°	-7.53°	58.1°	52.25°	31.55°	-2.4°
93	48.06°	29.29°	53.18°	-9.4°	50.55°	49.93°	39.74°	-11.9°	57.43°	46.7°	33.4°	-4.42°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
94	55.62°	29.96°	45.28°	-8.05°	62.31°	35.98°	34.36°	-3.66°	57.1°	53.59°	40.29°	-14.17°
95	44.86°	22.56°	39.74°	1.36°	55.42°	35.81°	27.64°	-13.07°	61.63°	51.41°	28.69°	-11.65°
96	54.28°	28.28°	35.03°	5.9°	51.22°	42.03°	28.31°	-1.48°	67.35°	47.37°	29.87°	-12.82°
97	59.49°	28.61°	35.54°	6.9°	51.56°	42.37°	45.12°	6.09°	56.09°	45.02°	25.83°	-11.98°
98	46.38°	31.64°	48.14°	-11.58°	66.18°	45.56°	28.48°	0.54°	58.44°	55.94°	24.32°	-1.06°
99	55.45°	20.38°	43.1°	-10.91°	55.93°	38.67°	34.53°	-12.07°	51.55°	47.54°	39.95°	-10.81°
100	45.54°	15.17°	42.76°	-2.34°	55.09°	45.73°	39.24°	-8.37°	56.93°	42.5°	26.34°	-10.3°
101	54.11°	25.76°	44.61°	-9.06°	63.32°	35.14°	41.92°	-6.52°	70.88°	48.38°	39.28°	-3.41°
102	62.68°	31.98°	38.06°	1.02°	61.81°	36.32°	41.59°	5.92°	54.24°	60.65°	34.74°	-4.08°
103	51.59°	13.82°	42.93°	5.06°	61.47°	42.2°	46.29°	2.22°	66.68°	50.57°	36.93°	-6.77°
104	43.86°	26.77°	43.27°	1.19°	49.88°	37.33°	38.9°	-2.15°	58.78°	52.58°	35.92°	3.48°
105	52.09°	27.61°	37.89°	-0.32°	57.1°	32.62°	37.56°	3.9°	68.36°	41.32°	43.48°	-13.33°
106	60.66°	18.53°	43.44°	-11.08°	50.21°	40.52°	27.98°	-9.71°	57.26°	59.98°	29.36°	0.79°
107	59.99°	20.55°	48.31°	3.54°	56.77°	38.84°	29.99°	2.39°	51.89°	54.26°	24.66°	-7.11°
108	47.22°	29.12°	38.39°	5.22°	64.0°	38.5°	28.14°	-4.34°	59.45°	49.56°	35.75°	-6.1°
109	50.41°	13.99°	38.9°	-12.09°	62.48°	43.54°	28.82°	4.24°	54.91°	50.73°	34.91°	-8.62°
110	61.5°	21.56°	47.13°	3.71°	50.38°	46.07°	43.94°	-0.97°	68.69°	58.63°	40.45°	-1.23°
111	62.34°	29.79°	34.86°	-9.73°	67.52°	30.44°	35.87°	-5.18°	56.59°	48.89°	36.25°	-14.67°
112	59.82°	22.9°	40.24°	4.89°	54.92°	30.94°	38.06°	-12.23°	52.73°	54.6°	35.41°	-0.72°
113	53.77°	26.43°	46.13°	-11.42°	61.64°	34.13°	45.62°	-4.0°	61.13°	58.47°	43.31°	1.13°
114	44.7°	26.6°	51.17°	-0.15°	60.63°	44.89°	30.16°	-13.75°	68.02°	46.03°	30.03°	-9.46°
115	48.9°	24.92°	53.52°	-2.0°	65.0°	48.59°	36.38°	-1.65°	51.72°	54.1°	39.61°	-15.01°
116	61.67°	17.86°	37.05°	-12.59°	51.39°	44.38°	42.93°	-2.32°	69.87°	49.39°	42.14°	-1.73°
117	55.96°	29.62°	41.59°	-8.22°	59.46°	38.17°	32.18°	-13.24°	62.14°	60.48°	30.37°	-8.79°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
118	50.24°	31.14°	36.71°	-4.36°	66.01°	30.77°	27.3°	-4.84°	62.98°	55.61°	25.5°	0.12°
119	49.4°	25.59°	36.55°	-5.37°	53.74°	42.7°	43.61°	-9.38°	67.01°	51.91°	32.05°	2.14°
120	53.27°	30.13°	39.57°	-0.83°	56.1°	34.97°	34.7°	-6.69°	54.74°	58.13°	38.27°	3.98°
121	63.52°	32.48°	53.52°	7.24°	68.36°	50.44°	47.14°	6.09°	60.66°	60.65°	44.32°	4.32°
122	50.66°	12.48°	53.52°	7.24°	66.17°	50.44°	47.14°	6.09°	71.55°	60.65°	44.32°	4.32°
123	49.71°	31.38°	50.36°	4.04°	64.05°	32.44°	33.8°	4.63°	62.17°	50.69°	39.41°	-5.68°
124	50.37°	28.28°	49.38°	7.24°	60.13°	35.09°	29.67°	4.25°	67.73°	50.03°	37.51°	3.89°
125	49.98°	32.48°	38.75°	7.24°	63.83°	32.85°	33.85°	6.09°	66.96°	52.74°	43.22°	-0.93°
126	60.24°	23.52°	50.55°	2.51°	63.47°	38.41°	36.44°	-0.9°	59.26°	43.59°	24.8°	-2.89°
127	58.43°	32.48°	48.2°	2.79°	60.92°	38.37°	27.14°	6.09°	70.14°	47.41°	31.15°	2.71°
128	47.77°	32.48°	42.24°	5.48°	61.61°	34.27°	27.14°	-0.41°	67.2°	52.66°	44.32°	-3.67°
129	45.75°	30.62°	44.84°	7.24°	62.83°	35.13°	28.56°	-0.18°	61.29°	52.42°	40.0°	2.51°
130	49.36°	32.48°	53.52°	5.93°	62.63°	35.41°	31.73°	-2.46°	61.64°	52.92°	44.32°	-0.53°
131	63.52°	32.48°	53.52°	7.24°	62.36°	50.44°	47.14°	-11.3°	71.55°	60.65°	24.32°	4.32°
132	43.52°	32.48°	53.52°	-12.76°	67.74°	50.44°	47.14°	6.09°	63.27°	60.65°	44.32°	-8.75°
133	59.84°	32.48°	33.52°	7.04°	61.79°	30.44°	27.14°	6.09°	71.55°	45.39°	44.32°	4.32°
134	47.76°	25.56°	53.52°	7.24°	65.1°	30.44°	27.14°	6.09°	59.94°	40.65°	44.32°	4.32°
135	59.47°	32.48°	53.52°	7.24°	58.08°	30.44°	27.14°	6.09°	68.76°	40.65°	44.32°	4.32°
136	53.61°	32.48°	51.49°	7.24°	56.89°	30.44°	27.14°	-8.51°	71.55°	60.65°	44.32°	4.32°
137	62.23°	32.48°	52.74°	7.24°	55.52°	30.44°	27.14°	6.09°	71.55°	58.81°	39.73°	4.32°
138	51.76°	32.48°	45.61°	7.24°	58.26°	30.44°	32.32°	-7.32°	61.08°	46.52°	44.32°	4.32°
139	47.4°	32.07°	40.02°	4.27°	62.57°	30.44°	27.14°	-8.02°	55.65°	48.24°	44.32°	4.32°
140	58.51°	32.48°	38.29°	7.24°	57.21°	30.44°	27.14°	-0.88°	71.55°	46.6°	44.32°	4.32°
141	43.7°	32.48°	33.52°	5.9°	58.51°	50.44°	47.14°	-11.53°	71.55°	60.25°	44.32°	-15.68°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
142	43.52°	12.48°	33.52°	7.24°	63.78°	50.44°	47.14°	6.09°	71.55°	60.65°	24.32°	-15.68°
143	49.71°	21.75°	35.14°	7.24°	62.78°	30.44°	37.33°	5.95°	64.3°	51.42°	44.32°	-15.68°
144	49.24°	25.76°	37.89°	7.24°	59.71°	30.44°	32.21°	-1.05°	63.11°	60.65°	44.32°	-6.72°
145	52.93°	31.97°	44.05°	7.24°	60.78°	30.44°	40.36°	6.09°	69.37°	50.66°	44.32°	4.32°
146	58.05°	25.27°	53.52°	-8.39°	62.05°	39.09°	27.14°	6.09°	71.55°	43.99°	25.4°	4.32°
147	43.52°	12.48°	33.52°	7.24°	57.34°	41.62°	38.71°	-11.68°	71.55°	58.02°	44.32°	-15.68°
148	49.27°	28.97°	46.54°	2.47°	63.46°	30.95°	35.48°	6.09°	64.67°	49.87°	39.75°	4.32°
149	53.52°	32.48°	37.36°	7.24°	58.19°	31.14°	34.04°	4.15°	65.15°	57.03°	44.32°	-15.68°
150	51.26°	32.48°	33.7°	7.24°	60.25°	32.17°	28.42°	-3.25°	68.69°	52.66°	44.32°	-15.68°
151	63.52°	12.48°	53.52°	7.24°	62.5°	50.44°	47.14°	-13.91°	60.85°	60.65°	24.32°	-15.68°
152	43.52°	32.48°	33.52°	-12.76°	66.02°	50.44°	27.14°	6.09°	71.55°	60.65°	44.32°	4.32°
153	49.66°	21.18°	41.57°	5.83°	61.64°	35.75°	30.18°	4.74°	71.55°	55.34°	36.68°	0.01°
154	46.72°	12.48°	33.52°	7.24°	60.48°	36.45°	47.14°	3.3°	67.71°	60.65°	24.32°	-13.59°
155	50.33°	30.26°	53.52°	7.24°	63.62°	41.48°	27.14°	-6.83°	52.62°	51.67°	28.91°	4.32°
156	63.52°	12.48°	50.7°	7.23°	60.28°	40.07°	45.64°	6.09°	66.69°	52.32°	25.43°	4.32°
157	47.96°	31.42°	43.92°	7.24°	63.11°	34.8°	41.68°	-1.08°	65.39°	60.65°	29.75°	0.35°
158	48.68°	29.07°	47.2°	5.25°	63.8°	37.72°	27.14°	-2.48°	55.67°	51.14°	31.3°	1.47°
159	51.7°	28.86°	49.54°	2.41°	60.02°	37.87°	30.85°	6.09°	61.32°	51.64°	34.09°	-4.1°
160	50.01°	27.48°	44.81°	6.55°	59.85°	37.17°	36.09°	-2.14°	69.45°	52.92°	32.24°	-1.06°
161	43.52°	32.48°	33.52°	7.24°	68.36°	50.44°	47.14°	6.09°	51.55°	50.61°	44.32°	4.32°
162	43.52°	12.48°	53.52°	-12.76°	68.36°	50.44°	47.14°	6.09°	56.09°	55.18°	24.32°	-15.68°
163	49.29°	12.48°	34.09°	7.24°	59.5°	30.72°	47.14°	-0.56°	60.75°	59.18°	24.49°	-15.65°
164	59.18°	12.48°	46.59°	7.24°	57.58°	30.44°	47.14°	6.09°	71.55°	51.13°	24.32°	4.32°
165	63.52°	12.48°	53.52°	7.24°	60.61°	30.44°	47.14°	6.09°	63.39°	52.4°	24.32°	4.32°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
166	63.52°	12.48°	38.6°	7.24°	58.32°	30.44°	47.14°	6.09°	66.57°	60.65°	24.32°	-8.12°
167	57.5°	13.49°	33.52°	7.24°	57.34°	30.44°	47.14°	6.09°	67.98°	60.65°	24.32°	4.32°
168	49.64°	12.48°	38.84°	4.99°	60.71°	30.44°	47.14°	2.89°	62.52°	56.92°	24.32°	-15.41°
169	55.23°	12.48°	42.24°	7.24°	58.44°	30.44°	45.37°	6.09°	67.87°	54.71°	24.32°	4.32°
170	52.3°	16.83°	42.93°	7.24°	60.88°	30.44°	35.49°	6.02°	71.55°	51.19°	28.27°	4.32°
171	43.52°	32.48°	53.52°	-12.76°	68.36°	50.44°	47.14°	6.09°	51.55°	45.71°	24.32°	4.32°
172	63.52°	32.48°	53.52°	7.24°	68.36°	50.44°	47.14°	6.09°	51.55°	40.65°	44.32°	4.32°
173	49.29°	32.48°	36.26°	4.42°	63.28°	32.22°	30.11°	6.09°	66.02°	60.65°	24.66°	4.32°
174	63.52°	13.73°	34.75°	7.24°	58.53°	45.27°	47.14°	-9.57°	71.03°	48.4°	24.32°	-15.68°
175	53.11°	12.54°	49.15°	3.82°	64.34°	41.66°	27.14°	6.0°	67.63°	46.57°	40.54°	4.32°
176	43.52°	23.42°	40.87°	-7.18°	63.52°	37.73°	27.14°	6.09°	70.34°	54.21°	24.32°	4.32°
177	56.73°	27.42°	33.52°	7.24°	65.45°	37.31°	27.33°	5.77°	66.32°	48.5°	27.92°	3.83°
178	50.22°	24.2°	47.99°	7.24°	62.53°	32.92°	29.66°	0.38°	64.45°	53.17°	37.48°	0.91°
179	49.13°	32.48°	50.71°	6.29°	62.35°	31.33°	29.52°	-2.59°	62.61°	54.78°	34.83°	2.93°
180	51.1°	32.48°	51.2°	7.24°	62.8°	30.44°	28.23°	-0.6°	65.58°	55.29°	32.09°	4.32°
181	43.52°	12.48°	53.52°	-12.76°	68.36°	50.44°	47.14°	6.09°	51.55°	40.65°	44.32°	4.32°
182	63.52°	12.48°	53.52°	7.24°	53.75°	50.44°	27.14°	-13.91°	71.55°	60.65°	24.32°	4.32°
183	63.52°	12.48°	33.52°	7.24°	55.13°	37.49°	47.14°	-13.91°	71.55°	46.74°	24.32°	-15.68°
184	50.59°	31.04°	35.37°	7.24°	65.66°	30.44°	27.14°	6.09°	63.34°	40.65°	24.32°	0.68°
185	43.52°	21.44°	52.39°	-12.76°	68.36°	35.61°	47.14°	6.09°	51.55°	41.27°	24.32°	4.32°
186	54.99°	12.48°	53.52°	-12.76°	68.36°	39.85°	47.14°	5.59°	51.55°	40.65°	24.32°	4.32°
187	43.52°	12.48°	53.52°	-8.51°	62.32°	43.03°	27.14°	6.09°	71.55°	40.65°	44.32°	4.32°
188	44.56°	32.24°	48.93°	3.6°	66.39°	30.44°	27.14°	6.09°	52.92°	48.11°	24.32°	0.29°
189	53.29°	24.22°	41.6°	7.24°	59.52°	30.85°	27.14°	6.09°	71.55°	46.97°	24.32°	4.32°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
190	43.55°	32.48°	52.04°	-12.76°	62.94°	30.44°	27.14°	6.09°	62.59°	53.69°	26.3°	4.32°
191	63.52°	32.48°	53.52°	7.24°	66.4°	50.44°	47.14°	-13.91°	53.31°	40.65°	44.32°	-15.68°
192	43.52°	32.48°	33.52°	7.24°	68.36°	50.44°	47.14°	-3.18°	51.55°	40.65°	44.32°	-15.68°
193	43.52°	12.48°	53.52°	-12.76°	62.87°	35.98°	27.14°	6.09°	66.39°	40.65°	44.32°	4.32°
194	43.52°	32.48°	33.52°	7.24°	68.36°	40.92°	47.14°	-3.15°	51.55°	40.65°	44.32°	4.32°
195	55.87°	12.48°	33.52°	7.24°	55.62°	42.36°	27.14°	-12.5°	66.83°	60.65°	24.32°	4.32°
196	58.19°	18.4°	33.52°	-12.76°	59.37°	37.31°	27.14°	6.09°	71.55°	60.65°	24.32°	4.32°
197	48.68°	25.27°	33.56°	4.85°	61.46°	30.48°	27.68°	6.04°	69.69°	60.64°	44.32°	4.32°
198	43.52°	12.48°	53.52°	-2.76°	64.48°	33.29°	27.14°	6.09°	69.5°	40.65°	44.32°	4.32°
199	46.98°	12.48°	46.86°	-1.09°	59.24°	39.9°	30.2°	2.59°	67.97°	40.65°	44.32°	3.64°
200	51.5°	25.46°	45.53°	7.24°	61.61°	39.17°	27.52°	4.98°	66.83°	40.65°	28.81°	4.32°
201	63.52°	32.48°	53.52°	-12.76°	67.05°	50.44°	47.14°	6.09°	51.55°	40.65°	25.71°	-15.68°
202	63.52°	32.48°	33.52°	-12.76°	56.17°	50.44°	27.14°	6.09°	71.55°	60.65°	24.32°	-15.68°
203	46.37°	29.48°	33.52°	6.47°	63.84°	30.44°	31.97°	4.54°	62.5°	55.41°	44.32°	4.32°
204	50.14°	20.17°	33.52°	-12.76°	57.46°	30.44°	27.14°	6.09°	71.55°	60.65°	26.72°	4.32°
205	52.83°	32.48°	33.52°	5.07°	58.87°	30.44°	27.14°	0.51°	68.5°	57.78°	24.4°	4.32°
206	63.52°	21.5°	35.39°	7.24°	55.15°	38.98°	27.14°	-1.92°	71.55°	51.95°	24.32°	4.32°
207	61.31°	32.48°	53.52°	7.24°	59.01°	38.41°	47.14°	-13.91°	51.55°	40.65°	44.32°	-15.68°
208	52.8°	25.78°	41.02°	7.24°	57.66°	30.44°	27.14°	6.09°	71.55°	60.65°	24.32°	4.32°
209	43.69°	21.2°	33.52°	4.14°	62.05°	30.44°	27.14°	-1.74°	69.15°	60.65°	44.31°	4.32°
210	43.52°	12.48°	33.52°	-12.76°	60.72°	30.44°	27.14°	6.09°	71.55°	60.65°	24.32°	4.32°
211	43.52°	12.48°	33.52°	7.24°	68.36°	50.44°	47.14°	-13.91°	51.55°	50.48°	24.32°	2.13°
212	63.52°	32.48°	33.52°	7.24°	64.96°	50.44°	27.14°	6.09°	51.55°	45.36°	24.32°	-15.68°
213	62.44°	32.34°	53.44°	-5.56°	64.29°	37.62°	27.24°	6.08°	59.89°	53.51°	27.97°	-15.63°

Continued on next page

Table D.1 – continued from previous page

No.	χ_1^0	χ_2^0	κ_1^0	κ_2^0	χ_1^1	χ_2^1	κ_1^1	κ_2^1	χ_1^2	χ_2^2	κ_1^2	κ_2^2
214	43.52°	12.48°	33.52°	7.24°	65.55°	40.11°	27.14°	6.09°	65.17°	50.55°	24.32°	1.94°
215	47.93°	21.06°	50.55°	4.26°	65.31°	38.94°	47.14°	-0.58°	51.55°	48.57°	43.84°	-11.36°
216	43.52°	32.48°	33.52°	-3.78°	64.86°	35.04°	41.55°	6.06°	57.73°	56.05°	35.84°	3.5°
217	63.52°	12.48°	53.52°	7.24°	59.91°	50.44°	27.14°	-13.91°	62.17°	40.65°	24.32°	-15.68°
218	49.9°	30.38°	33.52°	7.24°	62.86°	30.44°	29.99°	6.09°	61.76°	60.65°	44.32°	4.32°
219	46.89°	25.54°	33.52°	5.39°	64.39°	33.53°	32.07°	6.09°	71.05°	51.03°	39.84°	4.32°
220	51.77°	25.6°	50.4°	1.55°	64.93°	35.2°	32.66°	5.74°	66.15°	49.68°	36.44°	-6.54°

Table D.2: The objective and constraint function values for each design point. The blade angles are shown in Table D.1.

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
0	-0.3	-0.06	-0.06	-0.04	-0.06	-0.04	-0.11	-0.12	-0.13	-0.12	-0.13	-1.0
1	-0.24	-0.14	-0.04	0.04	0.03	-0.02	-0.03	-0.14	-0.21	-0.21	-0.15	-1.0
2	-0.17	0.04	-0.09	-0.15	-0.2	-0.22	-0.21	-0.09	-0.02	0.03	0.04	-1.0
3	-0.34	-0.04	-0.01	0.01	-0.02	-0.04	-0.13	-0.17	-0.18	-0.15	-0.13	1.0
4	-	-	-	-	-	-	-	-	-	-	-	-
5	-0.2	-0.23	-0.21	-0.17	-0.16	-0.14	0.06	0.03	-0.0	-0.01	-0.04	-1.0
6	-0.1	-0.14	-0.05	-0.06	-0.15	-0.14	-0.03	-0.12	-0.11	-0.03	-0.04	-1.0
7	-0.11	-0.19	-0.08	-0.01	-0.06	-0.07	0.02	-0.09	-0.17	-0.11	-0.11	-1.0
8	-0.32	-0.02	-0.03	-0.04	-0.06	-0.04	-0.15	-0.14	-0.14	-0.12	-0.13	1.0
9	-0.32	-0.15	-0.15	-0.14	-0.17	-0.19	-0.03	-0.03	-0.03	-0.0	0.02	1.0
10	-0.35	-0.07	-0.09	-0.11	-0.15	-0.18	-0.1	-0.09	-0.07	-0.02	0.0	1.0
11	-0.28	-0.04	0.03	0.08	0.07	0.06	-0.13	-0.21	-0.26	-0.25	-0.23	1.0
12	-0.33	-0.03	-0.07	-0.09	-0.12	-0.11	-0.15	-0.11	-0.09	-0.06	-0.06	1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
13	-0.35	0.02	0.01	0.0	-0.02	-0.03	-0.19	-0.18	-0.18	-0.15	-0.15	1.0
14	-0.34	-0.0	0.03	0.06	0.04	0.01	-0.17	-0.21	-0.23	-0.21	-0.19	1.0
15	-0.3	-0.0	-0.06	-0.11	-0.16	-0.18	-0.17	-0.11	-0.07	-0.01	0.01	-1.0
16	-0.14	0.01	0.08	0.09	0.03	0.01	-0.19	-0.25	-0.27	-0.21	-0.18	-1.0
17	-0.22	-0.14	-0.1	-0.05	-0.01	0.01	-0.04	-0.08	-0.13	-0.17	-0.19	-1.0
18	-0.17	-0.16	-0.18	-0.18	-0.18	-0.2	-0.01	0.01	0.01	0.01	0.03	-1.0
19	-0.25	-0.1	-0.17	-0.19	-0.2	-0.14	-0.08	-0.01	0.02	0.02	-0.03	-1.0
20	-0.29	-0.1	-0.06	-0.05	-0.09	-0.13	-0.07	-0.11	-0.12	-0.09	-0.04	1.0
21	-	-	-	-	-	-	-	-	-	-	-	-
22	-0.28	-0.07	0.01	0.05	0.01	-0.05	-0.11	-0.18	-0.22	-0.19	-0.12	1.0
23	-0.16	-0.24	-0.19	-0.13	-0.15	-0.16	0.07	0.01	-0.05	-0.03	-0.02	-1.0
24	-0.29	-0.04	0.07	0.12	0.09	0.02	-0.14	-0.24	-0.29	-0.27	-0.2	1.0
25	-0.28	-0.17	-0.09	-0.08	-0.06	0.01	-0.0	-0.09	-0.09	-0.12	-0.18	-1.0
26	-0.21	-0.21	-0.2	-0.16	-0.13	-0.07	0.04	0.03	-0.01	-0.05	-0.1	1.0
27	-0.37	-0.01	-0.01	-0.0	0.01	0.05	-0.17	-0.16	-0.17	-0.18	-0.22	1.0
28	-0.28	0.01	0.02	-0.04	-0.09	-0.11	-0.18	-0.19	-0.14	-0.09	-0.07	1.0
29	-0.38	-0.06	0.02	0.09	0.09	0.1	-0.11	-0.19	-0.26	-0.27	-0.27	1.0
30	-0.31	-0.01	-0.04	-0.06	-0.1	-0.09	-0.16	-0.14	-0.11	-0.08	-0.08	1.0
31	-0.27	-0.16	-0.2	-0.21	-0.23	-0.22	-0.02	0.02	0.03	0.06	0.05	1.0
32	-0.22	-0.16	-0.04	0.05	0.07	0.05	-0.01	-0.14	-0.22	-0.24	-0.22	-1.0
33	-0.3	0.01	0.09	0.13	0.09	0.0	-0.18	-0.26	-0.3	-0.26	-0.18	1.0
34	-0.33	-0.04	0.02	0.06	0.05	0.04	-0.13	-0.19	-0.23	-0.23	-0.21	1.0
35	-0.23	-0.06	-0.02	-0.0	-0.03	-0.03	-0.12	-0.15	-0.17	-0.15	-0.15	1.0
36	-0.28	-0.27	-0.27	-0.24	-0.21	-0.14	0.1	0.09	0.06	0.04	-0.04	1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
37	-0.17	-0.08	-0.12	-0.17	-0.23	-0.25	-0.1	-0.05	-0.0	0.06	0.07	-1.0
38	-0.35	-0.1	-0.09	-0.06	-0.03	0.03	-0.08	-0.08	-0.12	-0.14	-0.2	1.0
39	-0.24	0.01	0.0	0.0	-0.0	0.01	-0.19	-0.18	-0.18	-0.17	-0.18	1.0
40	-0.26	-0.05	-0.05	-0.04	-0.07	-0.09	-0.13	-0.13	-0.14	-0.11	-0.09	-1.0
41	-	-	-	-	-	-	-	-	-	-	-	-
42	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	-	-	-	-	-	-	-	-
44	-0.33	-0.14	-0.11	-0.08	-0.07	-0.04	-0.04	-0.06	-0.1	-0.11	-0.13	1.0
45	-0.22	-0.08	-0.07	-0.07	-0.08	-0.09	-0.09	-0.1	-0.11	-0.09	-0.09	-1.0
46	-0.26	-0.07	-0.04	-0.08	-0.13	-0.16	-0.11	-0.14	-0.09	-0.04	-0.01	-1.0
47	-0.36	0.07	0.07	0.07	0.06	0.06	-0.24	-0.24	-0.25	-0.24	-0.23	1.0
48	-0.36	-0.02	0.03	0.07	0.06	0.03	-0.15	-0.21	-0.25	-0.23	-0.2	1.0
49	-0.28	0.02	0.08	0.12	0.09	0.02	-0.19	-0.26	-0.29	-0.26	-0.2	1.0
50	-0.08	-0.22	-0.16	-0.07	-0.1	-0.2	0.04	-0.02	-0.11	-0.08	0.02	-1.0
51	-	-	-	-	-	-	-	-	-	-	-	-
52	-0.31	-0.03	-0.06	-0.06	-0.08	-0.06	-0.14	-0.12	-0.11	-0.1	-0.12	1.0
53	-0.31	0.08	0.14	0.15	0.08	-0.02	-0.26	-0.31	-0.32	-0.26	-0.15	1.0
54	-0.35	0.02	-0.03	-0.04	-0.06	-0.01	-0.19	-0.15	-0.13	-0.12	-0.16	1.0
55	-0.39	-0.01	0.06	0.09	0.05	-0.0	-0.17	-0.24	-0.26	-0.23	-0.17	1.0
56	-0.3	-0.12	-0.04	0.02	0.0	-0.03	-0.05	-0.13	-0.19	-0.18	-0.14	1.0
57	-0.18	-0.06	-0.08	-0.11	-0.12	-0.09	-0.12	-0.1	-0.07	-0.05	-0.08	-1.0
58	-0.17	-0.1	-0.02	0.07	0.05	-0.07	-0.07	-0.15	-0.24	-0.22	-0.11	-1.0
59	-0.29	-0.07	-0.13	-0.14	-0.13	-0.08	-0.11	-0.05	-0.04	-0.04	-0.1	1.0
60	-0.28	-0.08	-0.09	-0.07	-0.06	-0.02	-0.09	-0.08	-0.11	-0.11	-0.15	-1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
61	-0.3	0.03	0.04	0.03	0.01	-0.01	-0.2	-0.21	-0.21	-0.18	-0.17	1.0
62	-0.27	-0.11	-0.15	-0.16	-0.17	-0.1	-0.06	-0.02	-0.01	-0.0	-0.07	-1.0
63	-0.2	-0.12	-0.13	-0.13	-0.14	-0.13	-0.06	-0.05	-0.04	-0.03	-0.04	-1.0
64	-0.31	0.01	0.09	0.13	0.09	0.03	-0.18	-0.26	-0.3	-0.27	-0.2	1.0
65	-0.31	0.08	0.07	0.09	0.08	0.08	-0.25	-0.25	-0.27	-0.25	-0.26	1.0
66	-0.32	0.06	0.0	-0.03	-0.04	-0.02	-0.24	-0.18	-0.15	-0.13	-0.16	1.0
67	-0.3	0.03	0.07	0.08	0.03	-0.05	-0.21	-0.24	-0.26	-0.21	-0.13	1.0
68	-0.38	-0.01	-0.01	0.0	0.01	0.05	-0.17	-0.16	-0.18	-0.19	-0.23	1.0
69	-0.3	-0.08	-0.19	-0.22	-0.21	-0.12	-0.09	0.01	0.05	0.04	-0.05	1.0
70	-0.22	-0.18	-0.15	-0.12	-0.1	-0.06	0.01	-0.02	-0.05	-0.07	-0.11	-1.0
71	-0.27	-0.21	-0.21	-0.18	-0.13	-0.04	0.04	0.04	0.0	-0.04	-0.14	1.0
72	-0.23	-0.21	-0.19	-0.2	-0.2	-0.16	0.03	0.02	0.03	0.03	-0.01	-1.0
73	-	-	-	-	-	-	-	-	-	-	-	-
74	-	-	-	-	-	-	-	-	-	-	-	-
75	-0.26	-0.11	-0.04	0.02	-0.03	-0.07	-0.06	-0.14	-0.19	-0.14	-0.1	-1.0
76	-0.14	-0.12	-0.15	-0.13	-0.16	-0.17	-0.06	-0.03	-0.05	-0.01	-0.01	-1.0
77	-0.28	-0.1	-0.03	0.02	0.04	0.04	-0.07	-0.14	-0.2	-0.21	-0.22	1.0
78	-0.33	-0.02	-0.02	-0.03	-0.08	-0.12	-0.15	-0.15	-0.14	-0.1	-0.05	1.0
79	-0.29	0.06	0.04	0.04	0.03	0.04	-0.24	-0.21	-0.22	-0.21	-0.22	1.0
80	-0.18	-0.05	-0.03	-0.02	-0.07	-0.15	-0.12	-0.14	-0.15	-0.1	-0.03	-1.0
81	-0.36	0.03	0.02	0.03	0.03	0.04	-0.2	-0.2	-0.2	-0.2	-0.21	1.0
82	-0.32	0.05	0.04	0.02	-0.02	-0.04	-0.22	-0.21	-0.2	-0.16	-0.14	1.0
83	-0.37	0.06	0.01	0.02	-0.0	-0.02	-0.24	-0.18	-0.19	-0.17	-0.16	1.0
84	-0.24	-0.0	-0.01	-0.03	-0.05	-0.04	-0.17	-0.16	-0.14	-0.13	-0.13	1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
85	-0.29	-0.02	-0.02	-0.03	-0.09	-0.15	-0.16	-0.16	-0.15	-0.09	-0.02	1.0
86	-0.14	-0.1	-0.15	-0.16	-0.17	-0.13	-0.07	-0.02	-0.01	-0.01	-0.04	1.0
87	-0.33	-0.05	-0.02	0.0	-0.0	0.01	-0.13	-0.16	-0.18	-0.17	-0.18	1.0
88	-0.31	-0.1	-0.08	-0.04	-0.04	-0.02	-0.07	-0.1	-0.13	-0.13	-0.16	1.0
89	-0.11	-0.07	-0.08	-0.1	-0.18	-0.17	-0.11	-0.09	-0.08	0.01	-0.0	-1.0
90	-0.39	-0.0	0.09	0.14	0.11	0.06	-0.17	-0.27	-0.31	-0.29	-0.24	1.0
91	-0.32	-0.13	-0.22	-0.24	-0.24	-0.17	-0.04	0.04	0.07	0.07	-0.01	-1.0
92	-0.17	0.03	-0.04	-0.07	-0.09	-0.06	-0.2	-0.13	-0.1	-0.09	-0.11	-1.0
93	-0.23	0.11	0.09	0.09	0.06	0.02	-0.29	-0.26	-0.27	-0.23	-0.2	1.0
94	-0.3	-0.12	-0.14	-0.13	-0.11	-0.05	-0.05	-0.03	-0.04	-0.06	-0.12	-1.0
95	-0.34	0.04	-0.01	-0.04	-0.06	-0.05	-0.21	-0.16	-0.14	-0.11	-0.12	1.0
96	-0.3	0.05	0.08	0.08	0.02	-0.06	-0.22	-0.25	-0.25	-0.19	-0.11	1.0
97	-0.3	0.02	0.1	0.13	0.1	0.06	-0.19	-0.28	-0.3	-0.27	-0.23	1.0
98	-0.17	-0.1	-0.17	-0.18	-0.17	-0.09	-0.07	-0.01	0.01	-0.01	-0.08	-1.0
99	-0.34	-0.1	-0.06	-0.02	0.01	0.05	-0.07	-0.11	-0.15	-0.18	-0.23	1.0
100	-0.3	0.03	0.03	0.03	0.0	-0.01	-0.2	-0.2	-0.2	-0.18	-0.16	1.0
101	-0.3	-0.11	-0.13	-0.14	-0.17	-0.18	-0.07	-0.04	-0.04	-0.0	0.01	1.0
102	-	-	-	-	-	-	-	-	-	-	-	-
103	-0.24	-0.05	-0.04	-0.01	-0.08	-0.11	-0.12	-0.13	-0.16	-0.1	-0.07	-1.0
104	-0.34	0.14	0.11	0.11	0.08	0.06	-0.31	-0.29	-0.28	-0.25	-0.24	1.0
105	-0.35	0.04	0.01	-0.01	-0.07	-0.11	-0.21	-0.19	-0.16	-0.1	-0.07	1.0
106	-	-	-	-	-	-	-	-	-	-	-	-
107	-	-	-	-	-	-	-	-	-	-	-	-
108	-0.29	-0.04	-0.12	-0.16	-0.16	-0.11	-0.14	-0.05	-0.02	-0.01	-0.06	-1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
109	-0.27	-0.12	-0.12	-0.09	-0.07	-0.01	-0.05	-0.06	-0.08	-0.11	-0.17	1.0
110	-0.19	-0.05	0.09	0.15	0.11	0.02	-0.13	-0.26	-0.32	-0.29	-0.19	-1.0
111	-0.32	-0.23	-0.26	-0.23	-0.19	-0.1	0.06	0.08	0.05	0.01	-0.07	1.0
112	-	-	-	-	-	-	-	-	-	-	-	-
113	-0.29	-0.12	-0.12	-0.11	-0.08	-0.04	-0.05	-0.05	-0.07	-0.1	-0.14	-1.0
114	-0.25	0.0	-0.07	-0.11	-0.16	-0.18	-0.18	-0.11	-0.07	-0.01	0.01	1.0
115	-	-	-	-	-	-	-	-	-	-	-	-
116	-0.27	-0.11	0.03	0.1	0.09	-0.02	-0.07	-0.2	-0.27	-0.26	-0.16	1.0
117	-0.26	-0.14	-0.14	-0.12	-0.11	-0.05	-0.03	-0.03	-0.05	-0.07	-0.13	1.0
118	-0.31	-0.1	-0.2	-0.22	-0.21	-0.13	-0.07	0.02	0.04	0.03	-0.04	1.0
119	-0.27	0.0	0.03	0.03	-0.0	-0.05	-0.18	-0.2	-0.21	-0.17	-0.12	1.0
120	-	-	-	-	-	-	-	-	-	-	-	-
121	0.21	-0.21	-0.18	-0.15	-0.12	-0.11	0.03	0.0	-0.02	-0.05	-0.07	-1.0
122	0.13	-0.12	-0.13	-0.08	-0.08	-0.15	-0.05	-0.05	-0.09	-0.09	-0.03	-1.0
123	-0.32	-0.01	-0.1	-0.14	-0.15	-0.11	-0.16	-0.07	-0.04	-0.02	-0.06	1.0
124	-0.31	0.02	-0.04	-0.07	-0.1	-0.11	-0.19	-0.14	-0.11	-0.07	-0.06	-1.0
125	-0.3	-0.01	-0.09	-0.13	-0.16	-0.14	-0.16	-0.08	-0.05	-0.02	-0.04	-1.0
126	-0.3	-0.12	-0.13	-0.12	-0.12	-0.08	-0.05	-0.04	-0.06	-0.06	-0.09	1.0
127	-0.28	-0.06	-0.07	-0.08	-0.11	-0.15	-0.11	-0.11	-0.1	-0.06	-0.03	-1.0
128	-0.31	0.01	-0.08	-0.12	-0.15	-0.14	-0.18	-0.1	-0.06	-0.03	-0.03	-1.0
129	-0.31	0.01	-0.09	-0.13	-0.14	-0.09	-0.19	-0.08	-0.04	-0.03	-0.08	-1.0
130	-0.3	-0.01	-0.11	-0.13	-0.14	-0.09	-0.16	-0.07	-0.04	-0.03	-0.08	-1.0
131	0.08	-0.15	-0.1	-0.12	-0.17	-0.21	-0.02	-0.07	-0.05	-0.0	0.04	-1.0
132	0.13	-0.11	-0.09	-0.1	-0.13	-0.18	-0.06	-0.08	-0.07	-0.05	0.01	-1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
133	-0.33	-0.07	-0.09	-0.1	-0.13	-0.15	-0.11	-0.08	-0.07	-0.04	-0.02	1.0
134	-0.36	0.0	-0.1	-0.14	-0.15	-0.1	-0.18	-0.07	-0.03	-0.02	-0.07	1.0
135	-0.35	-0.0	-0.02	-0.03	-0.07	-0.11	-0.17	-0.15	-0.14	-0.1	-0.06	1.0
136	-0.31	-0.0	-0.06	-0.08	-0.1	-0.1	-0.17	-0.12	-0.09	-0.07	-0.07	1.0
137	-0.32	-0.04	-0.02	-0.01	-0.03	-0.07	-0.14	-0.16	-0.17	-0.15	-0.11	1.0
138	-0.36	0.03	-0.05	-0.08	-0.09	-0.06	-0.2	-0.12	-0.1	-0.09	-0.12	1.0
139	-0.36	-0.01	-0.13	-0.15	-0.14	-0.05	-0.17	-0.05	-0.02	-0.04	-0.12	1.0
140	-0.35	-0.01	-0.04	-0.04	-0.09	-0.12	-0.17	-0.14	-0.13	-0.09	-0.05	1.0
141	-0.01	-0.02	-0.02	-0.01	-0.03	-0.12	-0.16	-0.15	-0.16	-0.15	-0.06	-1.0
142	0.02	-0.05	-0.01	0.01	-0.11	-0.22	-0.13	-0.16	-0.18	-0.07	0.04	-1.0
143	-0.33	-0.03	-0.08	-0.11	-0.13	-0.08	-0.15	-0.09	-0.06	-0.04	-0.09	1.0
144	-0.33	-0.01	-0.07	-0.09	-0.1	-0.04	-0.16	-0.1	-0.08	-0.08	-0.13	1.0
145	-0.32	0.01	-0.05	-0.07	-0.11	-0.11	-0.18	-0.13	-0.1	-0.07	-0.06	1.0
146	-0.29	-0.1	-0.09	-0.09	-0.12	-0.16	-0.07	-0.08	-0.09	-0.05	-0.01	1.0
147	-0.24	0.01	-0.03	-0.06	-0.07	-0.1	-0.19	-0.15	-0.11	-0.1	-0.07	1.0
148	-0.33	-0.01	-0.09	-0.12	-0.13	-0.11	-0.16	-0.09	-0.06	-0.04	-0.07	1.0
149	-0.33	0.01	-0.03	-0.05	-0.08	-0.07	-0.18	-0.14	-0.13	-0.1	-0.1	1.0
150	-0.32	0.0	-0.07	-0.11	-0.15	-0.15	-0.18	-0.1	-0.07	-0.02	-0.02	1.0
151	-0.03	-0.15	-0.11	-0.12	-0.13	-0.1	-0.02	-0.06	-0.05	-0.04	-0.08	-1.0
152	0.02	-0.14	-0.18	-0.2	-0.14	-0.09	-0.04	0.01	0.02	-0.03	-0.08	-1.0
153	-0.28	-0.02	-0.06	-0.09	-0.13	-0.14	-0.15	-0.11	-0.08	-0.04	-0.03	-1.0
154	-0.26	-0.01	-0.03	-0.03	-0.1	-0.09	-0.17	-0.14	-0.14	-0.08	-0.08	-1.0
155	-0.26	-0.07	-0.15	-0.16	-0.13	-0.03	-0.11	-0.02	-0.02	-0.05	-0.14	1.0
156	-0.25	-0.1	-0.04	-0.03	-0.05	-0.06	-0.07	-0.14	-0.15	-0.13	-0.11	-1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
157	-0.26	-0.02	-0.1	-0.14	-0.15	-0.11	-0.16	-0.08	-0.04	-0.02	-0.07	-1.0
158	-0.3	-0.05	-0.13	-0.15	-0.14	-0.06	-0.13	-0.04	-0.03	-0.04	-0.12	-1.0
159	-0.3	-0.01	-0.05	-0.05	-0.07	-0.06	-0.16	-0.13	-0.12	-0.1	-0.12	-1.0
160	-0.29	0.0	-0.04	-0.08	-0.12	-0.13	-0.18	-0.13	-0.1	-0.06	-0.04	-1.0
161	-0.01	-0.1	-0.13	-0.08	-0.06	-0.06	-0.08	-0.04	-0.1	-0.12	-0.11	-1.0
162	-0.03	-0.12	-0.04	-0.08	-0.18	-0.15	-0.06	-0.14	-0.09	0.01	-0.03	-1.0
163	-0.34	-0.03	-0.05	-0.06	-0.06	-0.02	-0.15	-0.13	-0.12	-0.11	-0.15	1.0
164	-0.35	-0.05	-0.0	0.0	-0.03	-0.08	-0.13	-0.17	-0.18	-0.14	-0.09	1.0
165	-0.33	-0.1	-0.07	-0.05	-0.05	-0.03	-0.08	-0.1	-0.13	-0.13	-0.14	1.0
166	-0.32	-0.13	-0.05	-0.02	-0.03	-0.02	-0.05	-0.12	-0.16	-0.15	-0.15	1.0
167	-0.32	-0.07	-0.02	-0.0	-0.01	-0.02	-0.11	-0.15	-0.17	-0.16	-0.15	1.0
168	-0.34	-0.03	-0.05	-0.06	-0.08	-0.05	-0.15	-0.12	-0.11	-0.09	-0.12	1.0
169	-0.34	-0.04	-0.02	-0.02	-0.04	-0.05	-0.14	-0.15	-0.16	-0.14	-0.12	1.0
170	-0.34	-0.02	-0.05	-0.07	-0.11	-0.13	-0.15	-0.13	-0.1	-0.07	-0.04	1.0
171	-0.05	-0.05	-0.03	-0.1	-0.16	-0.11	-0.13	-0.14	-0.07	-0.02	-0.06	-1.0
172	0.02	-0.15	-0.11	-0.11	-0.12	-0.08	-0.02	-0.07	-0.06	-0.05	-0.1	-1.0
173	-0.28	-0.04	-0.11	-0.13	-0.15	-0.1	-0.14	-0.07	-0.04	-0.03	-0.07	-1.0
174	-0.23	-0.15	-0.03	-0.0	-0.07	-0.14	-0.02	-0.14	-0.17	-0.1	-0.04	-1.0
175	-0.27	-0.09	-0.11	-0.12	-0.13	-0.13	-0.08	-0.06	-0.06	-0.04	-0.04	-1.0
176	-0.27	-0.02	-0.08	-0.13	-0.16	-0.16	-0.15	-0.09	-0.05	-0.01	-0.01	-1.0
177	-0.28	-0.11	-0.14	-0.15	-0.16	-0.14	-0.06	-0.03	-0.03	-0.01	-0.03	-1.0
178	-0.32	-0.03	-0.1	-0.12	-0.14	-0.1	-0.14	-0.07	-0.05	-0.04	-0.07	1.0
179	-0.32	-0.01	-0.11	-0.14	-0.15	-0.1	-0.17	-0.07	-0.03	-0.03	-0.08	1.0
180	-0.32	-0.02	-0.11	-0.15	-0.16	-0.12	-0.15	-0.06	-0.03	-0.02	-0.06	-1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
181	-0.11	-0.12	-0.12	-0.04	-0.07	-0.07	-0.05	-0.05	-0.13	-0.1	-0.11	-1.0
182	-0.13	-0.17	-0.08	-0.01	0.01	-0.03	-0.0	-0.1	-0.16	-0.18	-0.14	-1.0
183	-0.31	-0.12	-0.06	-0.03	-0.05	-0.05	-0.05	-0.12	-0.15	-0.12	-0.13	1.0
184	-0.34	-0.02	-0.12	-0.16	-0.19	-0.15	-0.15	-0.05	-0.01	0.01	-0.02	1.0
185	-0.31	-0.04	-0.09	-0.16	-0.16	-0.06	-0.13	-0.08	-0.02	-0.01	-0.11	1.0
186	-0.28	-0.17	-0.11	-0.14	-0.13	-0.04	-0.0	-0.07	-0.03	-0.04	-0.13	1.0
187	-0.28	0.0	-0.04	-0.08	-0.13	-0.17	-0.18	-0.14	-0.1	-0.04	-0.0	-1.0
188	-0.34	0.01	-0.14	-0.18	-0.17	-0.06	-0.18	-0.04	0.0	-0.01	-0.11	1.0
189	-0.35	0.0	-0.03	-0.06	-0.11	-0.14	-0.18	-0.14	-0.12	-0.07	-0.04	1.0
190	-0.33	0.01	-0.09	-0.13	-0.14	-0.1	-0.19	-0.08	-0.04	-0.04	-0.08	1.0
191	-0.06	-0.12	-0.13	-0.18	-0.19	-0.15	-0.06	-0.04	0.0	0.02	-0.03	-1.0
192	-0.1	-0.07	-0.06	-0.1	-0.17	-0.14	-0.11	-0.11	-0.07	0.0	-0.04	-1.0
193	-0.34	-0.01	-0.06	-0.09	-0.13	-0.13	-0.17	-0.12	-0.08	-0.05	-0.05	1.0
194	-0.26	-0.04	-0.14	-0.15	-0.15	-0.07	-0.14	-0.04	-0.03	-0.03	-0.1	1.0
195	-0.24	-0.12	-0.08	-0.05	-0.03	-0.01	-0.06	-0.1	-0.13	-0.15	-0.16	-1.0
196	-0.26	-0.16	-0.1	-0.07	-0.07	-0.07	-0.01	-0.08	-0.11	-0.1	-0.1	-1.0
197	-0.3	-0.03	-0.08	-0.1	-0.12	-0.1	-0.15	-0.1	-0.07	-0.05	-0.07	-1.0
198	-0.35	0.01	-0.07	-0.12	-0.17	-0.18	-0.18	-0.1	-0.05	-0.0	0.0	1.0
199	-0.33	0.01	-0.01	-0.03	-0.08	-0.11	-0.19	-0.17	-0.14	-0.1	-0.06	1.0
200	-0.31	-0.0	-0.05	-0.07	-0.12	-0.14	-0.17	-0.13	-0.1	-0.06	-0.04	1.0
201	-0.05	-0.14	-0.06	-0.11	-0.15	-0.12	-0.04	-0.11	-0.07	-0.02	-0.05	-1.0
202	-0.09	-0.17	-0.05	0.0	-0.02	-0.08	-0.0	-0.12	-0.18	-0.15	-0.09	-1.0
203	-0.32	-0.01	-0.11	-0.14	-0.14	-0.08	-0.17	-0.07	-0.04	-0.03	-0.09	-1.0
204	-0.32	-0.05	-0.04	-0.04	-0.07	-0.07	-0.12	-0.13	-0.14	-0.11	-0.1	1.0

Continued on next page

Table D.2 – continued from previous page

No.	ψ_{ss}	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}
205	-0.32	-0.01	-0.06	-0.08	-0.11	-0.09	-0.16	-0.11	-0.09	-0.07	-0.08	1.0
206	-0.3	-0.1	-0.03	-0	-0.03	-0.08	-0.07	-0.14	-0.17	-0.15	-0.1	1.0
207	-0.27	-0.01	-0.09	-0.09	-0.07	0	-0.17	-0.08	-0.09	-0.1	-0.18	1.0
208	-0.32	-0	-0.02	-0.04	-0.08	-0.09	-0.17	-0.15	-0.14	-0.1	-0.09	1.0
209	-0.31	-0.01	-0.1	-0.14	-0.16	-0.11	-0.16	-0.08	-0.03	-0.02	-0.06	1.0
210	-0.31	-0.03	-0.07	-0.09	-0.12	-0.11	-0.14	-0.11	-0.08	-0.05	-0.07	1.0
211	-0.13	-0.14	-0.18	-0.08	-0.16	-0.08	-0.03	0.01	-0.09	-0.02	-0.1	-1.0
212	-0.15	-0.18	-0.15	-0.12	-0.09	-0.05	0.01	-0.02	-0.06	-0.08	-0.13	-1.0
213	-0.26	-0.16	-0.16	-0.14	-0.13	-0.09	-0.01	-0.01	-0.03	-0.04	-0.08	-1.0
214	-0.27	-0.02	-0.1	-0.15	-0.17	-0.14	-0.15	-0.07	-0.03	-0.01	-0.04	-1.0
215	-0.27	-0.05	-0.07	-0.12	-0.13	-0.02	-0.13	-0.1	-0.06	-0.05	-0.15	1.0
216	-0.29	-0.02	-0.11	-0.14	-0.13	-0.06	-0.15	-0.07	-0.04	-0.04	-0.12	-1.0
217	-0.22	-0.18	-0.13	-0.07	-0.07	-0.1	0.01	-0.04	-0.1	-0.1	-0.07	-1.0
218	-0.31	-0.04	-0.11	-0.13	-0.12	-0.05	-0.14	-0.07	-0.05	-0.06	-0.12	-1.0
219	-0.3	-0.01	-0.09	-0.13	-0.17	-0.17	-0.16	-0.08	-0.04	-0.0	-0.0	-1.0
220	-0.3	-0.06	-0.11	-0.14	-0.17	-0.15	-0.12	-0.06	-0.03	-0.01	-0.02	1.0

List of References

- Abbott, I.H., Von Doenhoff, A.E. and Stivers Jr, L. (1945). Summary of airfoil data. Tech. Rep., Langley Memorial Aeronautical Laboratory.
- Andrade, N., Brasileiro, F., Cirne, W. and Mowbray, M. (2007). Automatic grid assembly by promoting collaboration in peer-to-peer grids. *Journal of Parallel and Distributed Computing*, vol. 67, no. 8, pp. 957–966.
- ANSYS CFX Reference Guide (2010 November). *ANSYS CFX Reference Guide*. ANSYS, Inc., Southpointe 275 Technology Drive Canonsburg, PA 15317, release 13.0 edn.
- ANSYS CFX-Solver Modeling Guide (2010 November). *ANSYS CFX-Solver Modeling Guide*. ANSYS, Inc., Southpointe 275 Technology Drive Canonsburg, PA 15317, release 13.0 edn.
- ANSYS TurboGrid User’s Guide (2009 November). *ANSYS TurboGrid User’s Guide*. ANSYS Inc, 275 Technology Drive, Canonsburg, PA 15317, 12th edn.
- Aungier, R. (2003). *Axial-flow compressors: a strategy for aerodynamic design and analysis*. ASME Press. ISBN 9780791801925.
Available at: <http://books.google.co.za/books?id=E91SAAAAMAAJ>
- Benadé, J.G. (1987). *Die herontwerp van die Rofanco kompressor*. Master’s thesis, University of Stellenbosch.
- Box, G. and Draper, N. (1987). *Empirical model-building and response surfaces*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley. ISBN 9780471810339.
Available at: <http://books.google.co.za/books?id=Q02dDRufJEAC>
- Box, G.E. and Wilson, K. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 13, no. 1, pp. 1–45.
- Brandsen, J.D. (2013). *Prediction of Axial Compressor Blade Vibration by Modelling Fluid-Structure Interaction*. Master’s thesis, Stellenbosch University.
- Carregal-Ferreira, J., Holzwarth, A., Menter, F.E.T. and Luu, A. (2002). Advanced CFD analysis of aerodynamics using CFX. *AEA Technology GmbH, Otterfing*.

- Carter, A. (1950). The low speed performance of related aerofoils in cascade. Tech. Rep. CP29, Aeronautical Research Council.
- Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Chaudhuri, A., Haftka, R. and Viana, F. (2012). Efficient global optimization with adaptive target for probability of targeted improvement. In: *8th AIAA Multidisciplinary Design Optimization Specialist Conference, American Institute of Aeronautics and Astronautics, Honolulu, HI*, pp. 1–13.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, vol. 20, pp. 273–297.
- Cumpsty, N. (1989). *Compressor aerodynamics*. Longman Scientific & Technical. ISBN 9780582013643. Available at: <http://books.google.com/books?id=K691QgAACAAJ>
- Dixon, S. (2005). *Fluid mechanics, thermodynamics of turbomachinery*. Elsevier-Butterworth-Heinemann. ISBN 9780750678704. Available at: <http://books.google.com/books?id=QE1RCR4LfyQC>
- Drucker, H., Burges, C.J., Kaufman, L., Smola, A. and Vapnik, V. (1997). Support vector regression machines. In: M.C., M., M.I., J. and T., P. (eds.), *Advances in Neural Information Processing Systems*, vol. 9, p. 155–161. Springer, New York.
- Eberhart, R.C. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In: *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1, pp. 39–43. New York, NY.
- Edwards, A.W. (1974). The history of likelihood. *International Statistical Review/Revue Internationale de Statistique*, pp. 9–15.
- Forrester, A., Sobester, A. and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.
- Forsberg, J. and Nilsson, L. (2005). On polynomial response surfaces and kriging for use in structural optimization of crashworthiness. *Structural and Multidisciplinary Optimization*, vol. 29:, p. 232–243.
- Fourie, P. and Groenwold, A. (2000). Particle swarms in size and shape optimization. In: *Proceedings of the international workshop on multidisciplinary design optimization, Pretoria, South Africa*, pp. 97–106. DTIC Document.
- Fourie, P. and Groenwold, A. (2002). The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization*, vol. 23, no. 4, pp. 259–267.
- Gill, A. (2006). *A Comparison between Stall Prediction Models for Axial Flow Compressors*. Master's thesis, Stellenbosch University.

- Gill, A. (2012). *Four quadrant axial flow compressor performance*. Ph.D. thesis, Stellenbosch: Stellenbosch University.
- Ginsbourger, D., Le Riche, R., Carraro, L. *et al.* (2007). A multi-points criterion for deterministic parallel global optimization based on kriging. In: *Intl. Conf. on Nonconvex Programming, NCP07*.
- Gostelow, J.P. and Pollard, D. (1967). Some experiments at low speed on compressor cascades. *ASME, Transactions, Series A-journal of Engineering for Power*, vol. 89, pp. 427–436.
- Griffith, A. (1926). An aerodynamic theory of turbine design'. *RAE Report*, , no. H1, p. 111.
- Gunn, S.R. (1998). Support vector machines for classification and regression. Tech. Rep., University of Southampton.
- Henkenjohann, N. and Kunert, J. (2007). An efficient sequential optimization approach based on the multivariate expected improvement criterion. *Quality Engineering*, vol. 19, no. 4, pp. 267–280.
- Hill, W.J. and Hunter, W.G. (1966). A review of response surface methodology: a literature survey. *Technometrics*, vol. 8, no. 4, pp. 571–590.
- Horlock, J. (1958). *Axial flow compressors, fluid mechanics and thermodynamics*. Butterworths Scientific Publications.
Available at: <http://books.google.co.za/books?id=3SsIAQAAIAAJ>
- Howell, A. (1942). The present basis of axial flow compressor design: Part 1 - cascade theory and performance. Tech. Rep., Aeronautical Research Council Reports and Memoranda.
- Hoyle, N. (2006). *Automated Multi-Stage Geometry Parameterization of Internal Fluid Flow Applications*. Ph.D. thesis, University of Southampton.
- Jolliffe, I. (2005). *Principal component analysis*. Wiley Online Library.
- Jones, D.R., Schonlau, M. and Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, vol. 13, p. 455–492.
- Kaelbling, L.P., Littman, M.L. and Moore, A.W. (1996). Reinforcement learning: A survey. *arXiv preprint cs/9605103*.
- Kleijnen, J.P. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, vol. 192, p. 707–716.
- Koch, P.N., Simpson, T.W., Allen, J.K. and Mistree, F. (1999). Statistical approximations for multidisciplinary design optimization: the problem of size. *Journal of Aircraft*, vol. 36, no. 1, pp. 275–286.
- Kohavi, R. *et al.* (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *IJCAI*, vol. 14, pp. 1137–1145.

- Kovach, K. and Sandercock, D. (1961). Aerodynamic design and performance of five-stage transonic axial-flow compressor. *Journal of Engineering for Gas Turbines and Power*, vol. 83, no. 3, pp. 303–320.
- Krige, D.G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, vol. 52 (6), p. 119–139.
- Lewis, K.L. (1989). *Prediction of Stall Inception in an Axial Flow Compressor*. Master's thesis, University of Wits.
- Lieblein, S. (1960). Incidence and deviation-angle correlations for compressor cascades. *Journal of Basic Engineering*, vol. 82, no. 3, pp. 575–584.
Available at: <http://link.aip.org/link/?JBE/82/575/1>
- Lin, C.J., Hsu, C. and Chang, C. (2003). A practical guide to support vector classification. *National Taiwan University*.
- Liu, Y., Qin, Z., Shi, Z. and Lu, J. (2007). Center particle swarm optimization. *Neurocomputing*, vol. 70, no. 4, pp. 672–679.
- Lophaven, S., Nielsen, H. and Søndergaard, J. (2002). DACE: A MATLAB kriging toolbox. Tech. Rep., Technical University of Denmark.
- Martin, J.D. and Simpson, T.W. (2004). On the use of kriging models to approximate deterministic computer models. In: *Proceedings of DETC04: ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.
- Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, vol. (58) 8, pp. 1246–1266.
- McKay, M., Beckman, R. and Conover, W. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, vol. 42, no. 1, pp. 55–61.
- Mead, R. and Pike, D. (1975). A biometrics invited paper. A review of response surface methodology from a biometric viewpoint. *Biometrics*, vol. 31, no. 4, pp. 803–851.
- Meher-Homji, C.B. (2000). The historical evolution of turbomachinery.
- Menter, F.R. (1993). Zonal two equation Kappa-Omega turbulence models for aerodynamic flows. *c1993*, vol. 1.
- Menter, F.R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, vol. 32, no. 8, pp. 1598–1605.
- Molga, M. and Smutnicki, C. (2005). Test functions for optimization needs. Available online: <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>.

- Myers, R., Montgomery, D.C. and Anderson-Cook, C. (2009). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley Series in Probability and Statistics. Wiley. ISBN 9780470174463.
Available at: http://books.google.co.za/books?id=89oznEFHF_MC
- Myers, R.H., Khuri, A.I. and Carter, W.H. (1989). Response surface methodology: 1966–1988. *Technometrics*, vol. 31, no. 2, pp. 137–157.
- Myers, R.H., Montgomery, D.C., Vining, G.G., Borror, C.M. and Kowalski, S.M. (2004). Response surface methodology: A retrospective and literature survey. *Journal of Quality Technology*, vol. 36, p. 53.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830.
- Ponweiser, W., Wagner, T. and Vincze, M. (2008). Clustered multiple generalized expected improvement: a novel infill sampling criterion for surrogate models. In: *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*. *IEEE Congress on*, pp. 3515–3522. IEEE.
- Powell, M. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Gómez, S. and Hennart, J. (eds.), *Advances in optimization and numerical analysis: proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico*, Mathematics and its applications, p. 51–67. Kluwer Academic Publishers. ISBN 9780792326731.
Available at: <http://books.google.co.za/books?id=W8BvFyrSxiYC>
- Raubenheimer, G. (2011). *Vibration excitation of axial compressor rotor blades*. Master's thesis, Stellenbosch University.
- Rios, L.M. and Sahinidis, N.V. (2013). Derivative-free optimization: A review of algorithms and comparison of software implementations. *Journal of Global Optimization*, vol. 56, no. 3, pp. 1247–1293.
- Roos, T. (1995). *A prediction method for flow in axial compressors*. Master's thesis, University of Stellenbosch.
- Roos, T.H. (1990). *Reblading and testing of the Rofanco low speed axial flow compressor*. Master's thesis, University of Stellenbosch.
- Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P. (1989 November). Design and Analysis of Computer Experiments. *Statistical Science*, vol. 4, no. 4, pp. 409–423. ISSN 0883-4237.
Available at: <http://dx.doi.org/10.1214/ss/1177012413>
- Sakia, R.M. (1992). The Box-Cox transformation technique: a review. *The Statistician*, vol. 41, pp. 169–178.

- Sasena, M., Papalambros, P. and Goovaerts, P. (2002). Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization*, vol. 34, p. 263–278.
- Sayers, A. (1990). *Hydraulic and compressible flow turbomachines*. McGraw-Hill. ISBN 9780077072193.
Available at: <http://books.google.com/books?id=YNpSAAAAMAAJ>
- Schaller, R.R. (1997). Moore’s law: past, present and future. *Spectrum, IEEE*, vol. 34, no. 6, pp. 52–59.
- Scholkopf, B. and Smola, A.J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.
- Schonlau, M. (1998). *Computer experiments and global optimization*. University of Waterloo.
- Simpson, T.W., Mauery, T.M., Mistree, F. and Korte, J.J. (1998). Comparison of response surface and kriging models for multidisciplinary design optimization. *American Institute of Aeronautics and Astronautics*.
- Simpson, T.W., Poplinski, J.D., Koch, P.N. and Allen, J.K. (2001 July). Meta-models for Computer-based Engineering Design: Survey and recommendations. *Engineering with Computers*, vol. 17, no. 2, pp. 129–150.
Available at: <http://www.springerlink.com/content/74h70gye73ay4amd>
- Smola, A.J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, vol. 14, pp. 199–222.
- Snyman, J. (2005). *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, vol. 97. Springer.
- Spalart, P. and Allmaras, S. (1992). A one-equation turbulence model for aerodynamic flows.
- Thomas, K.D. (2005). *Blade Row and Blockage Modelling in an Axial Compressor Throughflow Code*. Master’s thesis, University of Stellenbosch.
- Timmer, W. (2009). An overview of NACA 6-digit airfoil series characteristics with reference to airfoils for large wind turbine blades. *AIAA Paper*, vol. 268, p. 2009.
- Trader, T. (2014 November). Cycle helps HGST stand up 70,000 core AWS cloud. <http://www.hpcwire.com/2014/11/11/cycle-helps-hgst-stand-70000-core-aws-cloud/>. Accessed November 23, 2014.
- Venkataraman, S. and Haftka, R.T. (2004). Structural optimization complexity: what has Moore’s law done for us? *Structural and Multidisciplinary Optimization*, vol. 28, no. 6, pp. 375–387.

- Versteeg, H. and Malalasekera, W. (1995). *An introduction to computational fluid dynamics*. Pearson Education Limited.
- Viana, F.A., Haftka, R.T. and Watson, L.T. (2010). Why not run the efficient global optimization algorithm with multiple surrogates? In: *Proceedings of the 51th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, AIAA, Orlando, FL, USA. AIAA-2010-3090*.
- Wang, G.G. and Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370–380.
- Watson, A.G. and Barnes, R.J. (1995). Infill sampling criteria to locate extremes. *Mathematical Geology*, vol. 27, no. 5, pp. 589–608.
- Welch, W.J., William, J., Yu, T., Kang, S. and Sacks, J. (1990). Computer experiments for quality control by parameter design. *Journal of Quality Technology*, vol. 22, no. 1, pp. 15–22.
- Wessels, F.J., Venter, G. and Von Backström, T. (2012). An efficient scheme for describing airfoils using non-uniform rational b-splines. In: *ASME Turbo Expo 2012: Turbine Technical Conference and Exposition*, pp. 969–977. American Society of Mechanical Engineers.
- Whitcomb, R.T. (1974). Review of NASA supercritical airfoils. *ICAS paper*, , no. 74-10, pp. 25–30.
- Wise, J. (2008). *Optimization of a Low Speed Wind Turbine using Support Vector Regression*. Master's thesis, University of Stellenbosch.