University of Cincinnati

Date: 10/31/2019

I, Pritesh Mandal, hereby submit this original work as part of the requirements for the degree of Master of Science in Mechanical Engineering.

It is entitled:

Design and Optimization of Boundary Layer Ingesting Propulsor

Student's name: **Pritesh Mandal**

This work and its defense approved by:

Committee chair: Mark Turner, Sc.D.

Committee member: Michael Alexander-Ramos,
Ph.D.

Committee member: Milind Jog, Ph.D.

Committee member: Paul Orkwis, Ph.D.

36454

Design and Optimization of Boundary Layer Ingesting Propulsor

A thesis submitted to the
Graduate School
of University of Cincinnati
in partial fulfilment of the
requirements for the degree of

Master of Science

in the Department of Mechanical Engineering
College of Engineering and Applied Sciences

Author:

Pritesh Mandal

Committee Chair:

Dr. Mark G. Turner

October 31, 2019

Abstract

The boundary layer ingestion concept has the potential to improve propulsion efficiency that will lead to better fuel economy of commercial aircraft. This design concept has been explored by NASA as the Single-aisle Turboelectric Aircraft with Aft Boundary-Layer propulsor “STARC-ABL.” This thesis discusses the 1D-3D aerodynamic design and optimization process of a propulsor with incoming distortion. It starts off with exploring the subsonic design of a two row propulsor (rotor and OGV) on a 2D basis, with angular momentum (related to work) distribution (rV_θ) manipulation at the rotor TE to get the optimum rV_θ profile with adiabatic efficiency as the objective function. The same concept of varying rV_θ profile on 2D basis was extended to the high speed propulsor design case before moving on to 3D design and optimization. The propulsor is at the tail of the fuselage, and consists of an Inlet Guide Vane (IGV), rotor, and Outlet Guide Vane (OGV). The design process presented accounts for only the large radial distortion and not the circumferential distortion. The thesis also describes a pragmatic way of choosing orthogonal design parameters to proceed with the optimization process using a genetic algorithm.

The final optimized design is highly-dependent on the objective function to be optimized. A Modified Adiabatic Efficiency (MAE) has been defined to account for the loss of kinetic energy at the OGV. This lost kinetic energy is essentially an approximation of the mixing loss of the tangential, axial and radial components of velocity at the OGV exit. The three blade-row adiabatic efficiency and MAE for an intermediate design case is 88.91% and 86.10% respectively. Upon further optimization and a different incoming distortion, using MAE as the objective function improved the MAE to 87.40% with an 88.159% adiabatic efficiency for the final design case presented in the thesis. The rotor-only adiabatic efficiency for this case is 93.85%, the IGV-Rotor two blade-row efficiency is 91.62% and the Rotor-OGV two blade-row efficiency is 90.37%. The IGV is necessary to support

the nacelle in this configuration. It also allows for the rotor work distribution to be optimized. MAE as the objective function leads to a decrease in Kinetic Energy variation at the OGV exit. The final design presented is not an optimum and indicates flow separation near the trailing edge of the hub to mid-span region of the rotor and also at the trailing edge of OGV hub region. Future efforts on the design process are also presented that can lead to improvement in the MAE.

Acknowledgments

The author is grateful to NASA for funding and support through NRA project. The weekly NASA meetings with the rest of the members proved to be very helpful in terms of their feedback. Rob Ogden deserves a special thanks for his system related support and the members of the UC Gas Turbine Simulation Laboratory for their assistance and clearing out certain misconceptions. The author would also like to thank the entire team involved in the NASA BLI project for helping out whenever needed.

The author would like to thank his thesis advisor, Dr. Mark G. Turner, for his insight, guidance and feedback. His methods of tackling an issue and solving it in a pragmatic way is worth the mention. Lastly, a special thanks to my parents and friends for providing me the constant support and encouragement.

Contents

1	Introduction	13
1.1	Motivation	13
1.2	Literature Review	14
2	Low-Speed Design and Analysis	18
2.1	T-AXI design and optimization	18
2.1.1	Design optimization using T-AXI	19
3	High-Speed Design and Optimization of Axi-symmetric Ducted Propulsor	25
3.1	NASA input conditions	25
3.2	T-AXI design and optimization (2D analysis)	26
3.3	FINE/Turbo design and optimization (3D analysis)	35
3.3.1	FINE/Turbo initial 3D design	35
3.3.2	Design variables for optimization	36
3.3.3	Intermediate results discussion	40
3.3.4	Line plots at different stations for Case EXPL-AXI-3IRO-6-4	41
4	Modified Adiabatic Efficiency	45
5	Optimizations with Revised Objective Function	51
5.1	Revised Operating Conditions	51
5.2	Design variables for optimization (using modified adiabatic efficiency as the objective function)	53
5.3	Comparison of line plots, pre and post using MAE at different stations	57

5.4	Comparison of B2B contour plots, pre and post using MAE at 10% span	60
5.5	Comparison of B2B contour plots, pre and post using MAE at 90% span	67
6	Performance Runs	72
7	Conclusion and Future work	74
7.1	Conclusion	74
7.2	Future work	75
	Bibliography	76
8	Appendix	78

List of Figures

2.1	Motor Speed vs Efficiency	19
2.2	Motor Speed vs Power	19
2.3	Sample stack file	20
2.4	Optimization flow chart	20
2.5	Pre-optz	21
2.6	Post-optz	21
2.7	Work distribution along rotor radius	23
2.8	Post-optimization rV_θ distribution along rotor radius	24
3.1	3D representation of propulsor ingesting boundary layer coming from fuselage Taken from A. Arntz, O. Atinault, D. Destarac, A. Merlen. Exergy-based Aircraft Aeropropulsive Performance Assessment: CFD Application to Boundary Layer Ingestion. AIAA AVIATION 2014, Jun 2014,ATLANTA, United States. <hal-01068957> . . .	25
3.2	From Flynn, Marie-Claire and Jones, Catherine E. and Norman, Patrick J. and Galloway, Stuart J. (2017) Establishing viable fault management strategies for distributed electrical propulsion aircraft. In: International Society of Air Breathing Engines, 2017-09-03 - 2017-09-08, Manchester Central.	26
3.3	Pareto Front of optimization run	31
3.4	Kinetic energy distribution @ OGV exit plane	32
3.5	rV_θ distribution of IGV	32
3.6	rV_θ distribution of rotor	33
3.7	Meridional view of Propulsor	34
3.8	Total Pressure @ IGV inlet plane	36

3.9	Radial angle @ IGV inlet plane	36
3.10	Case 6-4: PT @ rotor outlet plane	41
3.11	Case 6-4: PT @ OGV outlet plane	41
3.12	Case 6-4: TT @ rotor inlet plane	42
3.13	Case 6-4: TT @ rotor outlet plane	42
3.14	Case 6-4: Rvtheta @ IGV outlet plane	42
3.15	Case 6-4: Rvtheta @ rotor outlet plane	42
3.16	Case 6-4: Flow angle @ OGV inlet	43
3.17	Case 6-4: Flow angle @ OGV outlet	43
3.18	Case 6-4: Axial velocity @ OGV outlet	43
4.1	Meridional view of Propulsor	48
4.2	Propulsor Total Enthalpy-Entropy, H-s, diagram for Case EXPL-AXI-3IRO-7-15 . .	48
4.3	Zoom in view of Fig. 4.2	49
5.1	Radial profile of the Total Pressure Inlet Boundary Condition @ IGV inlet plane . .	52
5.2	Radial profile of the Total Temperature Inlet Boundary Condition @ IGV inlet plane	52
5.3	Radial profile of Radial Angles Inlet Boundary Condition @ IGV inlet plane	53
5.4	Case 6-4: PT @ OGV outlet: Pre MAE	57
5.5	Case 7-15: PT @ OGV outlet: Post MAE	57
5.6	Case 6-4: TT @ OGV outlet: Pre MAE	57
5.7	Case 7-15: TT @ OGV outlet: Post MAE	57
5.8	Case 6-4: rV_θ @ OGV outlet: Pre MAE	58
5.9	Case 7-15: rV_θ @ OGV outlet: Post MAE	58
5.10	Case 6-4: Flow angle @ OGV outlet: Pre MAE	58
5.11	Case 7-15: Flow angle @ OGV outlet: Post MAE	58
5.12	Case 6-4: Axial velocity @ OGV outlet: Pre MAE	59
5.13	Case 7-15: Axial velocity @ OGV outlet: Post MAE	59
5.14	Case 6-4(Pre MAE): Static Pressure @ 10% span	60
5.15	Case 7-15(Post MAE): Static Pressure @ 10% span	61
5.16	Case 6-4(Pre MAE): Iso-surface for $V_z=-0.5\text{m/s}$	62

5.17	Case 7-15(Post MAE): Iso-surface for $V_z=-0.5\text{m/s}$	62
5.18	Case 6-4(Pre MAE): Mrel @ 10% span	63
5.19	Case 7-15(Post MAE): Mrel @ 10% span	64
5.20	Case 6-4(Pre MAE): Entropy @ 10% span	65
5.21	Case 7-15(Post MAE): Entropy @ 10% span	66
5.22	Case 6-4(Pre MAE): Mrel @ 90% span	67
5.23	Case 7-15(Post MAE): Mrel @ 90% span	68
5.24	Case 6-4(Pre MAE): PT @ 90% span	69
5.25	Case 7-15(Post MAE): PT @ 90% span	69
5.26	Case 6-4(Pre MAE): Entropy @ 90% span	70
5.27	Case 7-15(Post MAE): Entropy @ 90% span	71
6.1	Adiabatic Efficiency vs Mass flow	72
6.2	Pressure ratio vs Mass flow	72
6.3	Comparison of MAE on changing objective function	73
6.4	Comparison of AE and MAE	73
8.1	Plot of Velocity profile vs y	115
8.2	Plot of Velocity squared profile vs y	116

List of Tables

2.1	Flow parameters and geometry for preliminary case	19
2.2	Optimization parameters	21
2.3	Parameters pre and post optimization	22
3.1	IGV input parameters	27
3.2	Stage input parameters	27
3.3	Input parameters	28
3.4	Base case properties for T-AXI simulation	29
3.5	Optimization parameters	30
3.6	Design details: Optimized Case 56	34
3.7	Summary of optimization runs with Adiabatic Efficiency as the objective function . .	39
3.8	Summary of optimization parameters used for runs with Adiabatic Efficiency as ob- jective function	40
5.1	Summary of optimization runs with Modified Adiabatic Efficiency as the objective function	55
5.2	Summary of optimization parameters used for runs with Modified Adiabatic Effi- ciency as objective function	55

Nomenclature

C	Curvature
R	Radius of curvature
C_p	Constant pressure specific heat, J/(kg·K)
h	Total specific enthalpy (J/kg)
$\dot{m}$	Mass flow rate, kg/s
p	Pressure, Pa
R	Propulsor tip radius, m
r	Propulsor hub radius, m
s	Entropy, J/(kg·K)
T	Temperature, K
V	Mass averaged resultant velocity, m/s
k	Ratio of heat capacity for Air, gama

Greek

Δ	Change in
γ	Specific heat ratio
n	normal vector
ρ	Density, kg/m ³
θ	tangential direction

Superscripts

A	Area average
-----	--------------

Subscripts

r	Radial direction
-----	------------------

z	Axial direction
t	Total condition
s	Static condition
in	Inlet condition (Station 1)
out	Outlet condition (Station 2)

Abbreviations

BLI	Boundary layer ingestion
NASA	National Aeronautics and Space Administration
BVF	Boundary vorticity flux
EWBL	End wall boundary loss
CFD	Computational fluid dynamics
UC	University of Cincinnati
LE	Leading edge
TE	Trailing edge
LHS	Left hand side, m^2/s^2
SOGA	Single objective genetic algorithm
MOGA	Multiple objective genetic algorithm
IGV	Inlet guide vane
OGV	Outlet guide vane
KE	Kinetic energy
AoA	Angle of attack
LE	Leading edge
TE	Trailing edge
T-AXI	Turbomachinery AXIsymmetric solver
T-CDES	Turbomachinery Compressor DESign
MP	Mixing Plane
KE	Kinetic Energy
KE_r	Kinetic Energy in radial direction
AE	Adiabatic efficiency

MAE	Modified adiabatic efficiency
PT	Absolute total pressure
TT	Absolute total temperature
B2B	Blade to blade
NLH	Non Linear Harmonics
Mrel	Relative Mach Number

Chapter 1

Introduction

1.1 Motivation

The benefits of the design concept of BLI has been explored by NASA as the Single-aisle Turboelectric Aircraft with Aft Boundary-Layer propulsor “STARC-ABL”. But the concept of BLI was never implemented in an commercial air plane unlike in the case of torpedoes, submarines and ships. There have been no tests carried out to figure out the effectiveness of the BLI concept when a propulsor is installed on the fuselage of an airplane. The propulsor is supposed to be operating when the air plane is in cruise mode, which is supposed to hugely impact the total fuel consumption for the entire flight time. This is, provided we see some positive results from testing done on a propulsor in a wind tunnel with effects of radial and circumferential distortion and other flow parameters simulated to best resemble the actual flow conditions prevailing during an actual flight. If the concept of BLI works equally good for passenger airplanes it could lead to a staggering reduction in the amount of fuel consumption every day. And this can be a breakthrough development in the field of commercial air planes. NASA has this vision of implementing the concept of BLI on air planes and it needs to design and test a propulsor for the same.

My role in the project is to generate steady-state design of a propulsor (excluding the flow path and the nacelle) from 0D-3D and optimize it using low and high fidelity flow solvers and optimizers. The distortion coming in to the fan IGV is highly loaded at the tip. This tip loaded profile coming in needs to be looked into to make it more even span wise for better efficiency of the propulsor. Making the rotor design forced vortex instead of free vortex seems to be helpful for this case and

should be looked into critically. Also, the kinetic energy profile that exits the OGV would mix out into the free stream and be lost. In that regards optimizing the design of the propulsor such that there is least spanwise KE variation with the best possible adiabatic efficiency is something that should be given due consideration too.

Its fortunate that NASA Glenn Research Center considered Gas Turbine Simulation Lab of Aerospace department of UC, run by Prof. Mark G. Turner and Prof. Paul Orkwis, to be capable enough to carry out the design and analysis of such a critical project. I am humbled to be a part of the team that is involved in making this project a success.

1.2 Literature Review

There have been many papers written on design and optimization of axial compressor with different methods used to handle geometric parameters of the blade and flow properties of the complete stage. One such paper by Hidetaka Okui et al.[1] presents a 3D optimization of a moderately loaded transonic compressor rotor by means of a multi-objective optimization system. The latter makes use of a differential evolutionary algorithm in combination with an Artificial Neural Network and a 3D Navier-Stokes solver. Operating it on a cluster of 30 processors enabled the evaluation of the off-design performance and the exploration of a large design space composed of the camber line and spanwise distribution of sweep and chord length.

Tblade3 is a parametric 3D blade geometry modelling tool for turbomachinery systems which was developed by Kiran Siddappaji et al.[2]. This geometry tool was used for generating the blade sections of the propulsor. Syed Moez Hussain Mahmood [3] worked on the "Optimization Capabilities for Axial Compressor Blades and Seal Teeth Cavity" for his Master's thesis. His method of 3D design and optimization was referred for the propulsor design.

AR Wadia et al.[1] describes the trend in using aerodynamic sweep to improve the performance of transonic blading. His paper reports on the experimental and analytical assessment of the pay-off derived from both aft and forward sweep technology with respect to aerodynamic performance and stability. The single stage experimental investigation includes two aft-swept rotors with varying degree and type of aerodynamic sweep and one swept forward rotor. A three-dimensional viscous flow analysis was used to assist in the interpretation of the data. The reduced shock/boundary

layer interaction, resulting from reduced axial flow diffusion and less accumulation of centrifuged blade surface boundary layer at the tip, was identified as the prime contributor to the enhanced performance with forward sweep. The impact of tip clearance on the performance and stability for one of the aft-swept rotors was also assessed. AR Wadia et al.[2] extends the assessment of the payoff derived from forward sweep with respect to aerodynamic performance and stability to multistage configurations. In the two-stage evaluations, the testing was extended to include the effect on inlet flow distortion. In the case of the 180-degree one-per-rev distortion pattern, the two-stage configuration was evaluated both with and without inlet guide vanes (IGVs). The presence of the inlet guide vanes had a profound impact on lowering the two stage fan's sensitivity with inlet distortion.

Chen et al.[3] presents two optimized rotors. The first rotor is a part of a 3-blade row optimization (IGV-rotor-stator) of a high-pressure compressor. It is based on modifying blade angles and advanced control of curvature of the airfoil camber line. The effects of these advanced blade techniques on the performance of the transonic 1.5-stage compressor were calculated using a 3D Navier-Stokes solver combined with a vortex/vorticity dynamics diagnosis method. The flow diagnostics of the rotor blade based on vortex/vorticity dynamics indicated that the boundary-layer separation behind the shock is verified by on-wall signatures of vorticity and skin friction vector lines. In addition, azimuthal vorticity and boundary vorticity flux (BVF) are shown to be two vital flow parameters of compressor aerodynamic performance that directly relate to the improved performance of the optimized transonic compressor blade. A second rotor-only optimization is also presented for a 2.9 pressure ratio transonic fan. The objective function is the axial moment based on the BVF.

H.D.Weingold et al.[4] in their paper on bowed stators discusses improvement of the efficiency of a multistage compressor. His analysis of multistage compressor stator surface static pressure data reveals that the radial growth of suction surface corner separation prematurely separates core flow stator sections, limiting their pressure rise capability and generating endwall loss. Modelling of the stator flow field, using a three-dimensional Euler analysis, has led to the development of "bowed" stator shapes, which generate radial forces that reduce diffusion rates in the suction surface corners, in order to delay the onset of corner separation. The strong interaction effects of the bowed stator with the viscous endwall flowfield are shown to be predictable using a three-dimensional multistage

Navier-Stokes analysis. This permits matching of the rotors to the altered stator exit profiles, in order to avoid potential stability limiting interactions. Application of bowed stators to a high bypass ratio engine eleven-stage high-pressure compressor has resulted in substantial improvement in efficiency, with no stability penalty.

Mialn Banjac's et al.[5] paper describes a methodology and a fully-tested and calibrated mathematical model for the treatment of endwall effects in axial compressor aerodynamic calculations. Additional losses and deviations caused by the clearance and secondary flows are analyzed. These effects are coupled with endwall boundary layer losses (EWBL) and blockage development. Stall/surge detection is included, and mutual interaction of different loss mechanisms is considered. Individual mathematical correlations for different effects have been created or adopted from earlier papers with the aim of forming one integral model that is completely described in this paper. Separate mathematical correlations and calibration measures are discussed in detail in the first part of the paper. The developed overall model is suitable for application in two-dimensional (2D) or mean-line compressor flow calculations. During the development, it was tested, calibrated, and validated using throughflow calculations comparing numerical results with experimental data for a large number of test cases. These test cases include compressors with very different configurations and operating ranges.

Another paper by Simon J Gallimore et al.[6] on effect of sweep and dihedral on 3D blade design of rotor into the core compressors for the Rolls-Royce Trent engine. It follows the development of the basic ideas in a university research project, through multistage low-speed model testing, to the application to high pressure engine compressors. An essential element of the project was the use of multistage CFD, and some of the development of the method to allow the designs to take place is also discussed. The first part of the paper concentrates on the university-based research and the methods development. The second part describes additional low-speed multistage design and testing and the high-speed engine compressor design and test.

The above papers have explored the geometrical aspects of blade design and improved the efficiency. This paper by Aniwat Tiralap et al.[7] has worked on the loading of the rotor blades to investigate its effect on the efficiency. Changes in loss generation associated with altering rotor tip blade loading of an embedded rotor-stator compressor stage was assessed with unsteady three-dimensional computations, complemented by control volume analyses. Tip-fore-loaded and tip-aft-

loaded rotor blades are designed to provide variation in rotor tip blade loading distributions for determining a compressor design hypothesis that aft-loading a rotor blade tip yields a reduction in loss generation in a stage environment. Aft-loading a rotor blade tip delays the formation of tip leakage flow, resulting in a relatively less mixed-out tip leakage flow at the rotor outlet and a reduction in overall tip leakage mass flow, hence a lower loss generation. Tip leakage flow, which is not mixed-out at the rotor outlet, is recovered in the downstream stator. The tip leakage flow recovery process yields a higher benefit for a relatively less mixed-out tip leakage flow in the tip-aft-loaded rotor blades on a time-averaged basis. These characterizing parameters together determine the attendant overall loss associated with rotor tip leakage flow in a compressor stage environment.

Chapter 2

Low-Speed Design and Analysis

The design and analysis of a boundary layer ingesting high speed fan can become very overwhelming very quickly for anyone who is new to fan design and optimization. 3D blade design parameters such as incidence, deviation, solidity, curvature, vortexing and its impact on the aerodynamic design of the fan stage coupled with boundary layer ingestion can become a bit too much to handle for a beginner in this field. One way of going about it is to break up in parts and learn the process before jumping into the problem and trying to tackle all at once. With the same rationale the low-speed design with clean flow at the inlet and forced vortex conditions was explored first to get going with things. E. J. Gunn and C. A. Hall [8] from Whittle laboratory, University of Cambridge have designed a low-speed fan rig running with a representative BLI inlet velocity profile. This fan design turned out to be a good starting point and I essentially ended up using certain design parameters from their fan design such as rotor inlet hub-to-tip radii ratio, flow coefficient, stage loading coefficient, number of rotors and stators. Note that the low-speed design and analysis also serves as an exercise to explore various methods of design improvement before implementing them for the actual high-speed design.

2.1 T-AXI design and optimization

T-AXI suite is a highly robust low fidelity axisymmetric flow solver. This suite was used for preliminary turbomachinery design starting from 1D/ Meanline inputs. As mentioned earlier the preliminary design uses similar key design parameters as the design of E. J. Gunn and C. A. Hall

Flow coefficient	0.52
Stage loading coefficient	0.502
Rotor inlet hub-to-tip radii ratio	0.3
Rotor inlet tip diameter	0.6933
Rotor and stator aspect ratio	3
Number of rotor blades	20
Number of stator blades	30
Rotor rpm	2350

Table 2.1: Flow parameters and geometry for preliminary case

from Whittle laboratory. Optimization of the axisymmetric design is then performed to improve the design. The starting flow and geometrical parameters are shown in Table 2.1:

Parameters such as the rotor hub diameter, rotor speed and stage pressure ratio were decided based on Emrax 207 motor model. The long-term plan for the low-speed design was to run it in a test rig and the motor purchased for the test was Emrax 207. It was desired that the motor should operate at 96%. Accordingly the rotor rpm was selected to be 2350 rpm as per Fig. 2.1 . The continuous power available at this speed is 16 kw, which in turn sets the pressure ratio of the fan stage to 1.0153 and the temperature ratio to 1.0048 for a given inlet mass flow rate of 11.43 kg/s. The outlet Mach number was selected in such a way so as to maintain a constant flow area throughout the passage. This is a free vortex design and the adiabatic efficiency for the fan stage came out to be 90.0%.

Figure 2.1: Motor Speed vs Efficiency

Figure 2.2: Motor Speed vs Power

2.1.1 Design optimization using T-AXI

Now that there is a design to work with, it was optimized using open source optimizer DAKOTA. Various fluid and geometrical properties such as solidity, aspect ratio and vortexing were looked into

to improve the design. The inputs needed for running T-AXI are the stack file, wall file and bgp file. A sample of the stack file is shown in Fig.2.3. The stack file has data such as the leading edge, trailing edge coordinates, number of blades on each row, rV_θ etc. This makes it easy to manipulate the data and link it with an optimizer to figure out the optimum combination of various parameters.

```

T-C_DES output, extension      Sample
      gamma,TT_in(deg R),type(0-comp,1-turb),ma, Re,BL-switch,visc-flag,refl
      1.400000    527.670000    0    0.230166  0.7900622E+07    1    1    0.346650
The overall information for each blade row is:
      Number_of_blades  rotation_speed  Hub_cl  Tip_cl
Geometry, loss, blockage and angular momentum are input starting at the hub
      r_le  r_te  x_le  x_te  Loss  Blk  rV_theta
20      0.248547  0.000000  0.004500
0.295518  0.296683  0.000000  0.365672  0.000000  0.037000  0.048760
1.004000  1.004000  0.115228  0.250635  0.000000  0.037000  0.048760
999.0000000000000000
30      0.000000  0.000000  0.000000
0.296992  0.297719  0.462779  0.690782  0.000000 -0.000000  0.000000
1.004000  1.004000  0.453325  0.700227  0.000000 -0.000000  0.000000

```

Figure 2.3: Sample stack file

The optimization process for various geometrical or flow parameters involves DAKOTA as the optimizer and T-AXI as the flow solver. The process flow can be seen in Fig 2.4.

Figure 2.4: Optimization flow chart

Aspect ratio

Aspect ratio of the blades can have a significant impact on the performance of the fan stage. This feature is captured in the form of axial location of leading edges (LE) and trailing edges (TE) in

the stack file. Aspect ratio has a direct impact on the solidity of the blade which in turn affects the diffusion factor of the blade and hence the performance of the blade. The TE of the rotor and stator were set as variables for the optimization run, using T-AXI as the solver and Dakota as the optimizer. The LE were varied by $\pm 10\%$ from their base value for the optimization run. Note that these optimization runs are carried out as test runs to see the impact of different geometrical and flow parameters on the design. The various parameters for the single objective genetic algorithm (SOGA) optimization run are listed as below:

Population size	600
Number of generation	7
Maximum function evaluations	3000
Total continuous design variables	4
Objective function	efficiency

Table 2.2: Optimization parameters

The pre-optimization T-AXI efficiency for the case was 90.0%. Post optimization the efficiency increased to 91.44%. The stack file and other major parameters that changed pre and post optimization are shown in Fig 2.6, Fig 2.7 and Table 2.3.

```

T-C_DES output, extension Sample
      gamma, TI_in(deg R), type(0-comp, 1-turb), ma, Re, BL-switch, visc-flag, refl
      1.400000      527.670000      0      0.230166 0.7900622E+07      0      1      0.346650
The overall information for each blade row is:
  Number_of_blades  rotation_speed  Hub_cl  Tip_cl
Geometry, loss, blockage and angular momentum are input starting at the hub
  r_le  r_te  x_le  x_te  Loss  Blk  rV_theta
16      0.248547  0.000000  0.004500
0.295518  0.296683  0.000000  0.365672  0.000000  0.015000  0.048760
1.004000  1.004000  0.115228  0.250635  0.000000  0.015000  0.048760
999.0000000000000000
24      0.000000  0.000000  0.000000
0.296992  0.297719  0.462779  0.690782  0.000000  0.030000  0.000000
1.004000  1.004000  0.453325  0.700227  0.000000  0.030000  0.000000

```

Figure 2.5: Pre-optz

```

T-C_DES output, extension APc
      gamma, TI_in(deg R), type(0-comp, 1-turb), ma, Re, BL-switch, visc-flag, refl
      1.400000      527.670000      0      0.230166 0.7900622E+07      0      1      0.346650
The overall information for each blade row is:
  Number_of_blades  rotation_speed  Hub_cl  Tip_cl
Geometry, loss, blockage and angular momentum are input starting at the hub
  r_le  r_te  x_le  x_te  Loss  Blk  rV_theta
16      0.248547  0.000000  0.004500
0.295518  0.296683  0.035257  0.330414446592  0.000000  0.015000  0.048760
1.004000  1.004000  0.140008  0.225855014916  0.000000  0.015000  0.048760
999.0000000000000000
24      0.000000  0.000000  0.000000
0.296992  0.297719  0.397739867  0.755821132117  0.000000  0.0300  0.000000
1.004000  1.004000  0.364478726  0.789073273482  0.000000  0.0300  0.000000

```

Figure 2.6: Post-optz

Parameter	Pre-optz	Post-optz
Aspect ratio rotor	1.58	2.10
Aspect ratio OGV	2.58	1.57
Average Chord rotor (non-dimen.)	0.4208	0.3155
Average Chord OGV (non-dimen.)	0.2566	0.4216
Solidity rotor	1.65	1.237
Solidity stator	1.506	2.475

Table 2.3: Parameters pre and post optimization

Vortexing

Vortexing can have a very big impact on the performance of a rotor blade. It basically has a say on the way the work is distributed along the rotor span. In other words, the pressure distribution along the rotor span can be manipulated. Vortexing can also turn out to be a very interesting way of manipulating the work distribution when there is a distortion profile imposed at the inlet of the fan stage. The profile of rV_θ distribution can be based on the use of an optimizer to vary the pattern of rV_θ along the span and use adiabatic efficiency as the objective function. This can be done by using a B-spline to represent the rV_θ distribution at the rotor trailing edge. The control points of the B-spline can be set as the design variables and the most efficient rV_θ distribution can be obtained. The area averaged rV_θ is kept the same as that of the free vortex case and only its distribution is varied span-wise to see what best fits, to get the most efficient design.

Figure 2.7: Work distribution along rotor radius

Fig. 2.9 shows a typical distribution of angular momentum (rV_θ) at different radius location of rotor. $rV_{\theta 1}, rV_{\theta 2}, rV_{\theta n}$ are set as design variables which changes its values for each iteration of the optimization. The area average rV_θ is calculated for each such distribution with the help of the below formula:

$$\overline{rV_\theta^A} = \frac{1}{\pi(R^2 - r^2)} \int_r^R (rV_\theta)_n 2r_n dr_n \quad (2.1)$$

This area average $\overline{rV_\theta^A}$ on the LHS of the above equation is made equal to the desired overall rV_θ based on the desired work (total temperature rise) and wheel speed. The work done or the enthalpy change at each rotor section is different and the optimizer (DAKOTA) which is looped with the flow solver T-AXI iterates to find out the best suitable work distribution for highest efficiency. One such optimization run is discussed below:

Pre-optimization

The two row (rotor-stator) adiabatic efficiency for the case is 92.5%. The rV_θ value is constant ($0.005859 \text{ m}^2/\text{s}$) along the rotor span. The pressure ratio and temperature ratio are 1.0157 and

1.0048 respectively.

Post-optimization

The adiabatic efficiency increased to 92.85% on an axisymmetric basis. The rV_θ distribution along the rotor span is not uniform though the area average $\overline{rV_\theta}$ is same as before. The pressure ratio and temperature ratio are 1.0156 and 1.0048 respectively. The pressure ratio and temperature ratio remains almost constant since the area average $\overline{rV_\theta}$ is maintained same. The rV_θ distribution post optimization is shown in Fig 2.10:

Figure 2.8: Post-optimization rV_θ distribution along rotor radius

It can be distinctly observed that the tip and hub are unloaded and with 80% span of the rotor seeing the maximum work. The results are somewhat as expected. The optimizer together with the flow solver guides the design where less work is extracted out of the hub and tip where the boundary layer losses are the highest. And it makes sense to do that for the low speed case with clean flow inlet.

Chapter 3

High-Speed Design and Optimization of Axi-symmetric Ducted Propulsor

3.1 NASA input conditions

The propulsor inlet will see a significant amount of distortion. It is not a clean flow with uniform velocity and pressure that comes in the propulsor. The airplane fuselage, tail and the wings create boundary layers and wakes which goes into the propulsor and the propulsor must be designed to handle it. Figures 3.1 and 3.2 give a better understanding of the flow field at the propulsor inlet.

Figure 3.1: 3D representation of propulsor ingesting boundary layer coming from fuselage
Taken from A. Arntz, O. Atinault, D. Destarac, A. Merlen. Exergy-based Aircraft Aeropropulsive Performance Assessment: CFD Application to Boundary Layer Ingestion. AIAA AVIATION 2014, Jun 2014, ATLANTA, United States. <hal-01068957>

Figure 3.2: From Flynn, Marie-Claire and Jones, Catherine E. and Norman, Patrick J. and Galloway, Stuart J. (2017) Establishing viable fault management strategies for distributed electrical propulsion aircraft. In: International Society of Air Breathing Engines, 2017-09-03 - 2017-09-08, Manchester Central.

NASA furnished the inlet flow field conditions such as pressure, velocity, flow angle, temperatures that the propulsor is likely to see during the wind tunnel testing for a 2 degree AoA case. These inputs given by NASA were converted to radial profiles of tangentially mass averaged quantities by a fellow project member Justin Holder, and used as inlet conditions for designing the propulsor.

3.2 T-AXI design and optimization (2D analysis)

The T-CDES (Turbomachinery Compressor DESign) code is used to get the meanline axial flow compressor design, the first step of T-AXI, an axisymmetric method for a complete turbomachinery geometry design. All the inputs that are used to run T-CDES are shown in Tables 3.1 - 3.3 .

Solidity	0.4776
Aspect ratio	2.832
Loss coefficient	0.045
Inlet angle	0
Inlet mach number	0.644
Lambda	0.97
Row space coefficient	1.5
Tip radius	0.145m

Table 3.1: IGV input parameters

The geometrical information and thermo-fluid dynamic data for each stage are in Table 3.2.

Mach1(rotor inlet)	0.55
Total temperature rise	29.2
Rotor loss coefficient	0.04
Stator loss coefficient	0.04
Rotor aspect ratio	2.3
Stator aspect ratio	2.3
Rotor solidity	1.5
Stator solidity	1.9
Rotor space coefficient	0.85
Stator space coefficient	0.85
Inlet angle	0
Inlet mach number	0.644
Stage Lambda	0.963
Row space coefficient	1.5
Stage tip radius	0.1524m

Table 3.2: Stage input parameters

Number of stages	1
Inlet mass flow rate (kg/s)	10.5
Revolutions per minute (rpm)	22138
Inlet Total Pressure (Pa)	88809.3
Inlet Total Temperature (K)	298.65
Alpha 3 - Last Stage (deg)	0.0
Mach 3 - Last Stage	0.49
Ratio of clearance/(tip radius)	0.001

Table 3.3: Input parameters

The T-CDES run output produces input files for running T-AXI, which in turn gives input files for T-Blade3. One of the output files created after running T-CDES is a stack file which contains the geometrical and fluid data. A couple of changes were made to the stack file before using it to run T-AXI. The inter-row distance between the rotor and stator were increased to accommodate the instrumentation that is supposed to be used for wind tunnel testing. Also, keeping the rotor and stator too close to each other would create acoustic problems. As a rule of thumb by NASA the distance between them was increased to the average chord length of the rotor. The incoming total pressure profile given by NASA Fig 3.8 was accounted for by changing the loss profile coming into the IGV. Also, the boundary layer switch was turned off and blockage value used for each row was sufficient to account for the boundary layer. The flow path used for the case was furnished by NASA and that was used to modify the walls file used by T-AXI. The stator and the IGV were axially swept backwards. With these changes the performance and geometrical parameters of the fan are shown in Table 3.4 based on loss models in T-AXI.

Efficiency	86.86
Pressure ratio	1.33
Temperature ratio	1.097
Mass flow (kg/s)	10.5
IGV aspect ratio	2.66
Rotor aspect ratio	1.54
Stator aspect ratio	2.321
IGV solidity	1.057
Rotor solidity	1.328
Stator solidity	1.477
Number of IGV, rotor, stator	20, 16, 33

Table 3.4: Base case properties for T-AXI simulation

This case optimized the rV_θ distribution of the IGV and rotor. The rV_θ distribution at the IGV exit is employed to unload the rotor tip and load up the rotor hub. This helps in bringing down the inlet rotor tip Mach number which in turn brings down the tip shock losses of the rotor. The rV_θ distribution at the IGV exit is set to be linear. The rotor rV_θ distribution was modified to get the best non-linear distribution in terms of efficiency and kinetic energy distribution at the OGV exit. The objective functions of the optimization scheme were 1-adiabatic efficiency and summation of kinetic energy of each streamline. The lesser the variation of kinetic energy along the rotor span, the lower the mixing loss with the free stream air at the OGV exit plane. A detailed explanation of the same is included in Chapter 4. The 2D flow solver used for the optimization is T-AXI and DAKOTA is used as the optimizer.

The number of variables used for the optimization were 9, which consisted of area average IGV rV_θ , area average rotor rV_θ , delta IGV rV_θ and 6 spanwise rotor rV_θ (@ 0%,20%,40%,60%,80%,100%). The six spanwise rotor rV_θ are manipulated to have an area average rV_θ equal to chosen area average rotor rV_θ . There is a range of values given (0.9-1.1 times free vortex rV_θ) from which DAKOTA can choose 'area average rotor rV_θ ' instead of taking the free vortex rotor rV_θ . The angular velocity is kept the same. A B-spline is fit between the six spanwise rotor rV_θ variables giving a non-linear

distribution of work at the rotor TE. The swirl distribution at the IGV exit is linear. The variable 'delta IGV rV_θ ' which can be negative or positive is an adder to the tip IGV rV_θ and hub IGV rV_θ . Again these IGV rV_θ and rotor rV_θ lines are transposed numerically to the right or left to ensure the area average rV_θ is equal to the variable chosen by DAKOTA for the rotor and IGV.i.e. area average IGV rV_θ and area average rotor rV_θ . This is a multi objective genetic algorithm optimization with 1-efficiency and summation of spanwise kinetic energy (KE) at the OGV exit as the two objectives for the minimization. These two objectives contradict each other and it would be helpful to get a scatter plot of the optimum cases to decide on the best case to be chosen. The various parameters used for the optimization are tabulated in Table 3.5.

Method	MOGA
Max function evaluations	9000
Population size	300
Number of generations	30
Design variables	9
Non-linear inequality constraints	4

Table 3.5: Optimization parameters

The MOGA optimization scheme iterated for 9387 cases before giving a Pareto front consisting of 60 optimum cases. A scatter plot of all the cases gives a better idea of the most optimum choice as shown in Figure 3.3. The extreme cases are: Case1- lowest on 1-efficiency and highest on summation of KE and Case60- highest on 1-efficiency and lowest on summation of KE. The goal is to come up with a case with sufficiently high efficiency with as lowest summation of KE as possible. The Pareto front is not completely smooth suggesting further iterations are required in the MOGA optimization. Since this is a low-fidelity simulation, and higher fidelity optimization is planned, this level of optimization convergence is adequate. Case 56 has been chosen in which to proceed to 3D. The total KE was included as the objective function since it is convenient, and also it contains both useful KE and lost KE and post-processing the optimum case 56 shows that the rV_θ curve for case 56 is flatter than case 60.

Figure 3.3: Pareto Front of optimization run

Radial profiles comparing the Kinetic Energy (KE) at the OGV exit comparing the baseline and optimal Case 56 is shown in Fig 3.4. It can be seen that there is less variation relative to the mean for the optimized case. Figures 3.5 and 3.6 show the radial profiles of rV_θ at the IGV and rotor exit respectively for several cases along the Pareto front.

Figure 3.4: Kinetic energy distribution @ OGV exit plane

Figure 3.5: rV_{θ} distribution of IGV

Figure 3.6: rV_θ distribution of rotor

Fig. 3.6 shows the rV_θ radial profile at the rotor exit of case 1, case 56 and case 60 (optimum cases) of the optimization run. Case 1 is the one with highest efficiency but also highest summation of KE. If the sole objective of the optimization process is to increase efficiency then the optimizer guides us to unload the tip and hub completely and increase the work added between (50-70)% span. The unloading of the tip can be attributed to tip strong inlet profile coming in, and any further increase of additional work would make the rotor even more tip strong. Also, the end wall and shock losses are high in the tip region making it not so preferable to add more work in those region. The same argument applies to the hub being unloaded as the losses at hub are also significant. So it makes more sense when the mid-span region is loaded to come out with a more uniform work distribution along the rotor span. But unfortunately since the objective is not only to get a good efficiency but also to get a uniform KE distribution at the OGV exit, case 1 is not the ideal case. Case 60 of the optimum cases is the one with least efficiency but with least summation of KE at the same time. It can be seen from the figure that the tip and hub are not unloaded unlike case 1. In fact the region between (60-90)% span is unloaded which comes up again at the tip and hub, but overall along the span the variation is not that significant, directing the design towards a more uniformly distributed work. But neither of the above two cases serves both the objectives that we

have for the optimization. Looking at the scatter plots Fig. 3.3 case 56 is the best combination of good efficiency and reduced KE variation, so its taken up for further 3D analysis and optimization.

Figure 3.7: Meridional view of Propulsor

Efficiency	87.122
Pressure ratio	1.344
Temperature ratio	1.101
IGV aspect ratio	2.659
Rotor aspect ratio	1.559
Stator aspect ratio	2.387
IGV solidity	1.059
Rotor solidity	1.320
Stator solidity	1.451

Table 3.6: Design details: Optimized Case 56

Fig 3.7 shows the meridional view of the propulsor and Table 3.6 gives the design details of Case 56 (chosen for 3D design and optimization).

3.3 FINE/Turbo design and optimization (3D analysis)

3.3.1 FINE/Turbo initial 3D design

T-AXI creates initial input files for T-Blade3 based on the axisymmetric flow angles. The design obtained after T-AXI optimization (case 56) is used as a starting point for 3D analysis to get a more realistic view of the actual performance of the propulsor. This case is the initial design or the base design to carry out further optimizations. Multigrid is used by Fine/Turbo so it is easy to set up fine grids that can easily be run with coarser grids. The nomenclature used is 000 for the finest mesh followed by 111 (medium) and 222 (coarse). Simulations for optimization were performed using the 111 grid level. A previous study showed that the 111 grid can produce similar trends in optimization in an order of magnitude less CPU time.

Inlet and Outlet Boundary conditions:

The Reynolds number at which the propulsor is operating is 2973520. The rpm of the rotor is 22138 rpm, inlet mass averaged total temperature is 298.639 K and total pressure and radial angle profiles are imposed at the IGV inlet. The incoming tangential flow angle is zero. The propulsor is run with exit Mach number of 0.51 with radial equilibrium conditions. The exit Mach number is an optional exit boundary condition available in Fine/Turbo. It allows for a design to proceed in a similar fashion to an operating line. As the efficiency changes, the static pressure is a result. With a lot of different designs explored, holding the exit Mach number is similar to a flow function and is better when comparisons of designs are considered. For initial and optimization runs, a mixing plane interface condition was used. The fluid is modelled as an ideal perfect gas (air) with Sutherland's law used for the viscosity. The Spalart-Allmaras turbulence model was used with the turbulent Prandtl number set to the default of 1. All surfaces are adiabatic. This is a steady-state run for the axisymmetric design. The total pressure and radial angle profiles are shown in Figs 3.8 and 3.9.

Figure 3.8: Total Pressure @ IGV inlet plane

Figure 3.9: Radial angle @ IGV inlet plane

The adiabatic efficiency for the base 3D case with the conditions as mentioned in 'FINE/Turbo initial 3D design' and medium grid (111) was 84.97%, pressure ratio 1.357, temperature ratio 1.107 and mass flow rate 10.13 kg/s.

3.3.2 Design variables for optimization

The choice of geometric variables and their sequence is essential for the design improvement to proceed in the right direction. The objective function for all the optimizations presented in this section is the overall adiabatic efficiency of all three rows. The best way to iterate on the geometric design variables would be to change them for all rows simultaneously, but it is too many variables for the the optimizer (genetic algorithm) to deal with. Hence care is taken to ensure that a few number of design variables are chosen in an optimization run which are orthogonal to the other design parameters of the propulsor. The list shows the order in which these were carried out:

Case:EXPL-AXI-3IRO-6-2 Optimization Run 1- Rotor leading edge (LE) and trailing edge (TE) metal angles : In this the rotor LE and TE metal angles are set up as design variables with a range of -7 deg to 7 deg and the optimum metal angels are obtained at the end of the optimization run. The reason for this choice of variable would be to align the air flow with the rotor metal angles and let the optimizer produce the optimum incidence and deviation at the LE and TE respectively. These angles are iterated upon at 5 spanwise locations of the rotor : 0, 0.25, 0.5, 0.75 and 1.0.

Case:EXPL-AXI-3IRO-6-2 Optimization 2- Rotor curvature : The curvature (second derivative) of the blade changes the loading of the blade. There is an option to change the meanline second derivative of the blade in tblade3. It varies the camber distribution of the blade. In a way it controls the way the pressure gradients are set on the pressure and suction side of the blade. The dependency of pressure gradient to curvature is

$$\frac{1}{\rho} \frac{\partial p}{\partial n} = \frac{V^2}{R} = CV^2 \quad (3.1)$$

Where: ρ = density

p = pressure

n = normal vector

V = velocity

R = radius of curvature

C = curvature

This variable is set for 4 spanwise location (0.0, 0.35, 0.65 and 1.0) at 6 chordwise location of the blade (0.15, 0.25, 0.5, 0.75 and 0.95). At the tip of the rotor, it changes the shock structure (oblique/normal and attached/detached) and the optimization process produces the type of shock for the best efficiency.

Case:EXPL-AXI-3IRO-6-2 Optimization 3- Inlet guide vane (IGV) chord : The IGV chord is an important geometric variable for optimization , wherein the chord of the blade was changed along its span (0.0, 0.25, 0.50, 0.75, 1.0). This became a very important variable since there was significant pre-swirl to the rotor going in. This pre-swirl was generated because of the turning of the IGV (about -5 deg at the tip and 15 deg at the hub) which helps in reducing the tip mach numbers of the rotor and thereby reducing the rotor tip shock losses. In order to accomplish the desired turning, the chord at each span also needs to be as per the turning requirement apart from changing the TE metal angles. This optimization run sets up the correct chord needed at each span location to generate the needed swirl going into the rotor, and effectively changes the solidity at a given span.

Case:EXPL-AXI-3IRO-6-2 Optimization 4- Outlet guide vane (OGV) LE metal angles : In this

the OGV LE metal angles are set up as variable with a range of -7 deg to 7 deg and the optimum metal angles are obtained at the end of the optimization run. The reason for this choice of variables was to align the air flow with the OGV metal angles and let the optimizer produce the optimum incidence at the LE. These angles are modified at 5 spanwise locations of the rotor : 0, 0.25, 0.5, 0.75 and 1.0.

Case:EXPL-AXI-3IRO-6-2 Optimization 5- OGV lean and chord : The next set of optimization was to optimize the tangential lean and chord at the same time for the OGV. Both these variables were changed at 5 spanwise locations of OGV (0.0, 0.25, 0.50, 0.75, 1.0). From the post processing of the optimum case of optimization 4, separation was observed at the hub region of OGV. To mitigate this high entropy region the tangential lean was optimized along with the chord. The air flow needs to be turned back to zero, with no swirling flow downstream of the OGV. To accomplish this the chord at various sections of the OGV had to be modified once again to get the desired turning that needs to happen across the OGV.

Case:EXPL-AXI-3IRO-6-2 Optimization 6- OGV curvature : The OGV curvature (second derivative) was changed for 4 spanwise location (0.0, 0.35, 0.65 and 1.0) at 6 chordwise location of the blade (0.15, 0.25, 0.5, 0.75 and 0.95). This was done to obtain the best pressure loading of the blade.

Case:EXPL-AXI-3IRO-6-2 Optimization 7- OGV trailing edge (TE) metal angles : The OGV TE metal angles are set up as variable with a range of -7 deg to 7 deg and the optimum metal angles were obtained at the end of the optimization run. These angles were modified at 5 spanwise locations of the rotor : 0, 0.25, 0.5, 0.75 and 1.0.

Here the case nomenclature is as follows:

Case:EXPL-AXI-3IRO-6-2 : Exploratory Design - Axisymmetric Blading - 3(Blade rows) IGV Rotor OGV - Case Version # - Sub Version #

Table 3.7 summarizes the adiabatic efficiency, pressure ratio and mass flow rates of the optimum case at the end of each optimization run and Table 3.8 summarizes the optimization parameters used for each optimization run.

Optz Runs	Parameters changed	Adiabatic Effi- ciency(%) (Objective function)	Pressure ratio	Mass flow (kg/s)
1	Rotor leading edge (LE) and trailing edge (TE) metal angles 5 spanwise locations	85.983	1.316	9.99
2	Rotor curvature 5 chordwise locations 4 spans	86.922	1.303	9.927
3	IGV chord 5 spanwise locations	87.287	1.306	9.944
4	OGV LE metal angles 5 spanwise locations	87.331	1.3063	9.9445
5	OGV tangential lean and chord 5 spanwise locations	87.47	1.3073	9.9415
6	OGV curvature 7 chordwise locations 4 spans	87.582	1.3098	9.8907
7	OGV TE metal angles 5 spanwise locations	87.633	1.3178	9.8628

Table 3.7: Summary of optimization runs with Adiabatic Efficiency as the objective function

Optz Runs	Number of Design Variables	Population Size	Number of generations	Max function evaluations	Evaluations executed	Constraint: Pressure ratio
1	10	50	8	400	400	1.3 - 1.348
2	20	60	9	540	540	1.3 - 1.348
3	5	50	8	400	319	1.3 - 1.348
4	5	50	8	400	327	1.3 - 1.348
5	10	60	8	480	423	1.3 - 1.348
6	28	70	8	560	560	1.3 - 1.348
7	5	50	8	400	320	1.3 - 1.348

Table 3.8: Summary of optimization parameters used for runs with Adiabatic Efficiency as objective function

After a total of 7 overall 3D design optimization runs with 400, 400, 319, 327, 423, 560, 320 individual design iterations, a total of 2749 designs and 90 design parameters and structural analysis and appropriate changes made on the rotor by Justin Holder (Case:EXPL-AXI-3IRO-6-4), the 3 blade row efficiency (Fine grid, Non-Linear Harmonic unsteady model) was 88.15%, Rotor-only: 93.16%, IGV-rotor : 89.33%. The Pressure Ratios: 3 blade row: 1.326, rotor-only: 1.347, IGV-rotor: 1.342 rotor-OGV: 1.331, Grid Size 3,487,843 nodes. The power consumption was 282.4kW with atmospheric inlet conditions. The Structural Analysis was performed by Justin Holder and design modified to obtain Safety Factor of 1 at 120% speed. Note that during optimization runs medium grid mixing plane (MP) was used with structured grids.

3.3.3 Intermediate results discussion

Since there were already 7 optimization runs carried out, and structural changes made to the optimum case after iteration Case:EXPL-AXI-3IRO-6-2 Optimization 7 which gave us Case:EXPL-AXI-3IRO-6-4, it's best to discuss the design available to figure out the future course of optimization runs.

3.3.4 Line plots at different stations for Case EXPL-AXI-3IRO-6-4

Several plots are significant from the point of view of the direction in which future optimizations should be carried out.

Total Pressure coming in and out of the Propulsor:

Figure 3.10: Case 6-4: PT @ rotor outlet plane

Figure 3.11: Case 6-4: PT @ OGV outlet plane

The average total pressure profile coming out of the rotor (Fig. 3.10) is not as tip strong as compared to the one at the IGV inlet (Fig. 3.8). The objective of the design process is to reduce to effect of the distortion coming into the nacelle. It was partially achieved on a 2D basis by forced vortex condition employed at the rotor outlet. This helped in loading the hub region as well and extracting more work out of the rotor hub. When Fig 3.10 and Fig 3.11 are compared, one can easily observe loss of total pressure at the OGV hub region: spans (0.0 - 0.2). The possible reason for this loss could be separation happening at the hub of OGV trailing edge. We can confirm that from contour plots, and with the separation further, optimizations can be carried out to reduce the separation region.

Total Temperature coming in and out of the Rotor:

Figure 3.12: Case 6-4: TT @ rotor inlet plane

Figure 3.13: Case 6-4: TT @ rotor outlet plane

Rvtheta coming out of IGV and Rotor:

Figure 3.14: Case 6-4: Rvtheta @ IGV outlet plane

Figure 3.15: Case 6-4: Rvtheta @ rotor outlet plane

Fig 3.12 shows the temperature profile coming into the rotor and Fig 3.13 shows how the profile has been changed to extract more work out of the hub region and unloading the tip, thereby making the work distribution more even throughout the rotor span. This was achieved by optimizations carried out using T-AXI on rV_θ distribution along the IGV TE and rotor TE. Forced vortex condition was used and the optimum rV_θ distribution was achieved from highest efficiency and lowest axial kinetic energy point of view. Fig 3.15 shows the loaded hub of the rotor and the unloaded tip.

Flow angles at OGV inlet and outlet:

Figure 3.16: Case 6-4: Flow angle @ OGV inlet

Figure 3.17: Case 6-4: Flow angle @ OGV outlet

Fig. 3.16 shows the air flow angles coming into the OGV. These flow angles need to be turned back to zero degrees at the outlet so as to ensure that there is least amount of swirl coming out of the OGV. Any swirl flow coming out of the OGV will mix out with the free stream and will be a loss for the entire propulsor and downstream system. The flow angle coming into the OGV hub is around -55 deg, which is very high to be turned back to zero. And it's clearly seen in Fig. 3.17 that the flow angles at the hub and tip are considerably high. Secondary flows near the hub and casing contribute to the swirl variation at the OGV exit. The optimization only accounts for the adiabatic efficiency of the propulsor, and does not take into account the mixing loss downstream of the propulsor. In order to minimize the mixing of swirl loss, there has to be a way of accounting for the swirl kinetic energy in the adiabatic efficiency itself. Optimizations need to be carried out to take care of swirl flow as well.

Figure 3.18: Case 6-4: Axial velocity @ OGV outlet

Fig. 3.18 shows the spanwise variation of axial velocity at the OGV outlet. This variation of axial velocity will also increase the loss due to mixing out with the free stream. The ideal condition to keep the mixing losses as low as possible would be to have a constant axial velocity profile at the OGV exit. There must be a way to account for least variation of axial velocity as well in the optimizations.

Chapter 4

Modified Adiabatic Efficiency

The objective function for the 3D optimization runs presented in Chapter 3 had been adiabatic efficiency using a Single Objective Genetic Algorithm (SOGA). From the discussion of the results of Case:EXPL-AXI-3IRO-6-4 kinetic energy is lost at the OGV exit in the form of tangential swirl. Also, there is sufficient variation of axial velocity along the OGV span which would also get mixed out and the resulting Kinetic Energy would be lost. With optimizations that use adiabatic efficiency as the objective function, the kinetic energy lost at the OGV exit does not impact the stage adiabatic efficiency, and is not captured when it is the objective function. The flow field coming out of the propulsor is going to mix out with the free stream and if the propulsive efficiency is considered which includes the propulsor and the free stream, then the overall efficiency should be impacted by the lost kinetic energy. Mass averaged radial profiles of axial, tangential and radial velocity at the OGV exit plane are utilized during optimization. Among these three velocity profiles tangential velocity is the one which cannot be recovered and all of it is lost due to mixing with the free stream. The radial velocity can be recovered to a certain extent by changing the flow path and the tail cone slope. The axial velocity also can be recovered to a large extent by reducing its variation along the span @ OGV exit plane. Hence, there is a need to switch the objective function from adiabatic efficiency to a new - "Modified Adiabatic Efficiency" or MAE.

Ideally the tangential velocity coming out of the OGV should be nearly zero, the axial and radial velocity variation along the span at the OGV exit should be small, which is not the case for the design (Case:EXPL-AXI-3IRO-6-4) when the objective function was set to be the adiabatic efficiency. The flow angles are very high at the hub and tip at the OGV exit even though the OGV

exit metal angles are set to zero in the design. Secondary flows as well as large deviation angles near the hub and casing contribute to the swirl variation at the OGV exit and consequently the high flow angles. The axial velocity also has a significant variation along the OGV exit. This is largely due to the tip loaded total pressure profile coming into the propulsor. This is taken care of to a certain extent by extracting more work out of the rotor hub region in the 2D optimizations with T-AXI and DAKOTA as mentioned in Chapter 3 Section 3.2. But the tip is still doing more work as compared to the hub, which is ideally the best work distribution for an efficient rotor if the incoming flow field is undistorted. But since there is distortion coming into the propulsor, spanwise work variation is very important. Making the axial velocity profile coming out of the OGV as uniform as possible tends to reduce the adiabatic efficiency but also reduces the losses of the overall propulsor due to mixing with free stream. It hence becomes very important to balance between losing out on adiabatic efficiency and gaining on the overall propulsor farfield efficiency.

There is a need to make the mass averaged radial, tangential and axial velocity at OGV exit plane as a part of the optimization process. One way to go about it would be to include all these parameters as objective functions along with adiabatic efficiency and run a Multi-Objective Genetic Algorithm (MOGA) optimization. Finding the best combination of 4 objective functions would take a lot of additional CPU time. A better way is to include all these terms in the efficiency equation itself i.e. modify the adiabatic efficiency to account for all the kinetic energy terms associated with the three velocity components and have a single modified adiabatic efficiency as the new objective function. In that case a SOGA would definitely serve the purpose and getting an optimal case would be easier.

Adiabatic efficiency for a compression process is the ratio of the ideal change of Total Enthalpy to the actual change of Total Enthalpy.

$$\eta = \frac{\Delta H_{ideal}}{\Delta H_{actual}} \quad (4.1)$$

Ideal Total Enthalpy rise which is the numerator of the adiabatic efficiency needs to be deducted by the amount of kinetic energy lost due to mixing out to the free stream. The ideal total enthalpy rise is adjusted since in an ideal situation there would not be any mixing of kinetic energy except

for the “ uniform jet velocity ” to free-stream jet velocity. The modified adiabatic efficiency is then

$$\eta_{mod} = \frac{\Delta H_{ideal-mod}}{\Delta H_{actual}} \quad (4.2)$$

Mass-averaged value of any function f

$$\bar{f} = \frac{\int_A (\rho f \vec{v} \cdot d\vec{A})}{\int_A (\rho \vec{v} \cdot d\vec{A})} \quad (4.3)$$

$$\Delta H_{ideal-mod} = \Delta H_{ideal} - \frac{\overline{V_\theta^2}}{2} - \frac{\overline{V_z^2} - \bar{V}_z^2}{2} - \frac{\overline{V_r^2} - \bar{V}_r^2}{2} \quad (4.4)$$

Fig. 4.1 is the meridional view of a propulsor to show Station 1 (IGV inlet) and Station 2 (OGV outlet). Points 1 and 2 in Fig 4.2 correspond to Stations 1 and 2 respectively.

Fig. 4.2 is the H-s diagram for Case EXPL-AXI-3IRO-7-15 (the final optimized case with modified adiabatic efficiency as the objective function described in Chapter 5) which gives a thermodynamic perspective of the concept of lost kinetic energy. The "lost" Kinetic Energy is,

$$KE_{lost} = \frac{\overline{V_\theta^2}}{2} + \frac{\overline{V_z^2} - \bar{V}_z^2}{2} + \frac{\overline{V_r^2} - \bar{V}_r^2}{2} \quad (4.5)$$

which accounts for all the kinetic energy of swirl and the variation of kinetic energy of V_r and V_z . This is because the tail cone could potentially convert the V_r to an axial component, and thrust is due to V_z being larger than the freestream. Line 1-2' in Fig. 4.2 is the ideal path for the compression process in the propulsor and line 1-2 is the actual path followed by the propulsor when compression is happening from pressure P_{T1} to P_{T2} . KE_{lost} (kinetic energy lost) depicted by 3'-2' in Fig. 4.3 is the amount of input energy which is completely lost due to mixing with the free stream air flow. Due to this loss actual compression of air happens from total pressure P_{T1} to $P_{T3'}$. Hence, a suitable amount is debited from the ideal ΔH , to account for the kinetic energy lost and it reflects in turn in the modified adiabatic efficiency. This helps in avoiding the illusion of the overly optimistic

adiabatic efficiency and gives us the realistic modified adiabatic efficiency as given by Eq 4.3 and 4.4.

Figure 4.1: Meridional view of Propulsor

Figure 4.2: Propulsor Total Enthalpy-Entropy, H-s, diagram for Case EXPL-AXI-3IRO-7-15

Figure 4.3: Zoom in view of Fig. 4.2

Assuming a perfect gas, total specific kinetic energy at OGV exit plane

$$\overline{T}_T - \overline{T}_S = \left(\frac{\overline{V}_z^2 + \overline{V}_\theta^2 + \overline{V}_r^2}{2 * C_p} \right) \quad (4.6)$$

Modified adiabatic efficiency

$$\eta_{modified} = \frac{\Delta H_{ideal-mod}}{\Delta H_{actual}} = \frac{\frac{\Delta H_{ideal}}{C_p} - (\overline{T}_T - \overline{T}_S) + \frac{\overline{V}_z^2}{2C_p} + \frac{\overline{V}_r^2}{2C_p}}{\frac{\Delta H_{actual}}{C_p}} \quad (4.7)$$

Actual change in enthalpy

$$\Delta H_{actual} = C_p(\overline{T}_{T2} - \overline{T}_{T1}) \quad (4.8)$$

Ideal change in enthalpy

$$\Delta H_{ideal} = \overline{T}_{T1} C_p \left(\left(\frac{\overline{P}_{T2}}{\overline{P}_{T1}} \right)^{\frac{k-1}{k}} - 1 \right) \quad (4.9)$$

Where

$\bar{f}$ = Mass averaged value of function f

ρ = density kg/s

$\vec{v}$ = velocity vector

$\overline{V_\theta}$ = mass averaged tangential velocity @ any plane (m/s)

$\overline{V_r}$ = mass averaged radial velocity @ any plane (m/s)

$\overline{V_z}$ = mass averaged axial velocity @ any plane (m/s)

Cp = specific heat capacity of air at constant pressure ($J/kg.K$)

$\overline{T_T}$ = mass averaged total temperature @ any plane (K)

$\overline{T_S}$ = mass averaged absolute static temperature @ any plane (K)

H = Total enthalpy (J/kg)

Explanation of lost KE in the axial direction: Axial velocity V_z is a summation of average axial velocity and delta V_z from its average which is defined as V_z''

$$V_z = \overline{V_z} + V_z'' \quad (4.10)$$

$$\overline{V_z^2} = (\overline{V_z})^2 + \overline{(V_z'')^2} + 2 * \overline{V_z} * \overline{V_z''} \quad (4.11)$$

The third term of Eq. 4.11 turns out to be zero since $\overline{V_z''}$ equals zero. The lost axial KE is then

$$\overline{(V_z'')^2} = \overline{V_z^2} - (\overline{V_z})^2 \quad (4.12)$$

Refer Appendix for more details. Similar analogy is also applicable for KE lost in the radial direction as well.

$$\overline{(V_r'')^2} = \overline{V_r^2} - (\overline{V_r})^2 \quad (4.13)$$

Chapter 5

Optimizations with Revised Objective Function

5.1 Revised Operating Conditions

The operating conditions of the propulsor were changed significantly by NASA and this led to a significant change in the performance of the propulsor. The rpm of the rotor changed to 21008 rpm (1100 ft/s rotor tip speed) from the 22348 rpm used in the optimizations shown in Chapter 3. The rotor tip clearance ratio (ratio of clearance to tip radius at the leading edge) increased from 0.09% to 0.15%. Also, the IGV LE was swept forward and at the same time the rotor LE and TE was swept backward. One reason for this geometrical change was to increase the axial distance between IGV and rotor tips so that the influence of the rotor shock on the IGV is less. Also, sweeping the rotor LE reduces the effective shock strength in the transonic regions of the rotor. From the previous post processing it could easily be seen that the relative Mach numbers were above sonic from 70% span to the rotor tip. The flow path at hub region of the rotor LE was slightly increased so as to avoid the low velocity pocket in that region. NASA ran a propulsor-fuselage-vertical tail simulation. From that, the fine-grid 1 deg AoA, inlet (IGV upstream) total temperature, total pressure and radial angle profiles were also updated by NASA on June 4, 2019. This represents the Aerodynamic Interface Plane (AIP). The case is now run with exit Mach number of 0.51. The rest of the design and operating conditions are kept unchanged. This case is the new base case for the next set of optimizations. The radial profile of the axisymmetric average total pressure, total temperature and radial angles are shown in Figs 5.1 - 5.3.

Figure 5.1: Radial profile of the Total Pressure Inlet Boundary Condition @ IGV inlet plane

Figure 5.2: Radial profile of the Total Temperature Inlet Boundary Condition @ IGV inlet plane

Figure 5.3: Radial profile of Radial Angles Inlet Boundary Condition @ IGV inlet plane

The adiabatic efficiency for the baseline case came out to be 86.679%, pressure ratio is 1.2554 and temperature ratio is 1.0775 with a mass flow of 9.49 kg/s.

5.2 Design variables for optimization (using modified adiabatic efficiency as the objective function)

Case EXPL-AXI-3IRO-7-5 - IGV TE, Rotor LE-TE, OGV LE-TE metal angles : In this set of optimization the IGV TE, Rotor LE-TE and OGV LE-TE metal angles are set as variable for 0% and 25% span. The objective of this optimization run is to eliminate the high entropy region at the region near the hub. Aligning the flow by changing the metal angles simultaneously for all three blade rows is best method for finding the optimum metal angles. This rational was not followed in previous optimization runs since doing all the three rows simultaneously for 5-6 span locations is too many variables to handle by the optimizer. The design space becomes too large to handle by Dakota and previous experience of optimizing just two rows simultaneously did not produce encouraging results.

Case EXPL-AXI-3IRO-7-6-1 - Rotor LE metal angle, chord wise curvature and location of maximum thickness : The post processing of the optimization showed marginal improvement in the flow

near the hub. One more significant thing that came up in the post processing is the shock structure at the rotor tip. The nature of the shock is near-normal and it is detached. Certain design variables can be iterated upon to change this shock to more oblique with reduced loss at rotor blade tip. The rotor LE metal angle, curvature of the blade at 7 chord wise location (including the LE and TE curvature points) and the location of the maximum thickness along the blade chord were defined as variables and optimization was run.

Case EXPL-AXI-3IRO-7-7 - OGV TE metal angle : In this optimization run the OGV TE metal angles are defined as the variable @ 5 spanwise locations (0.0, 0.25, 0.5, 0.75, 1.0). The objective of this optimization was to reduce the swirl coming out of the OGV as much as possible so that the kinetic energy lost due to mixing out of the tangential velocity with free stream is reduced. The modified adiabatic efficiency has kinetic energy term associated with tangential velocity at OGV exit which drives the optimizer towards an OGV with a small amount of swirl coming out of it.

Case EXPL-AXI-3IRO-7-8 - OGV LE metal angle, curvature and chord length : The OGV hub region is still showing some significant amount of entropy rise which would be best if gotten rid of. In an effort towards that OGV LE metal angles, curvatures (including LE and TE points) and chord length are varied at 0% and 25% span and the optimization is run.

Case EXPL-AXI-3IRO-7-14/15 - Rotor LE and TE metal angles : Ever since the rotation speed was decreased to 21008 rpm the pressure ratio came down from about 1.3 to around 1.25. This was not acceptable design for NASA. Hence, in an effort to ramp up its pressure ratio, the rotor speed was again increased to 21581 rpm. This corresponds to a rotor tip velocity of 1130 ft/s. Now since the rotor rpm has been changed the rotor LE and TE metal angles needs to be changed accordingly. Hence, rotor LE and TE metal angles were set as variables at 5 spanwise location (0.0, 0.25, 0.5, 0.75, 1.0). Note that the optimization was separately run for LE and TE.

Table 5.1 summarizes the adiabatic efficiency, pressure ratio and mass flow rates of the optimum cases at the end of each optimization run and Table 5.2 summarizes the optimization parameters used for each optimization run.

Optz Runs	Parameters changed	MAE(%) (Objective function)	AE(%)	Rpm	Pressure ratio	Mass flow (kg/s)
EXPL-AXI-3IRO-7-5	IGV TE, Rotor LE and TE, OGV LE and TE metal angles 2 span locations (0,0.25)	83.83	86	21008	1.269	9.413
EXPL-AXI-3IRO-7-6-1	Rotor LE metal angle, curvature, location of maximum thickness 7 chordwise locations 2 span locations (0.7,1.0)	83.87	86.03	21008	1.266	9.399
EXPL-AXI-3IRO-7-7	OGV TE metal angle 5 spanwise locations	85.026	85.72	21008	1.262	9.473
EXPL-AXI-3IRO-7-8	OGV LE metal angles, curvature and chord $\times$ 7 chordwise locations 2 spans (0,0.25)	88.058	85.687	21008	1.261	9.473
EXPL-AXI-3IRO-7-14/15	Rotor LE and TE metal angles 5 spanwise locations Note: Both optimizations were performed separately	85.362	86.18	21581	1.306	9.715

Table 5.1: Summary of optimization runs with Modified Adiabatic Efficiency as the objective function

Optz Runs	Number of Design Variables	Population Size	Number of generations	Max function evaluations	Evaluations executed	Constraint
EXPL-AXI-3IRO-7-5	10	45	5	225	206	–
EXPL-AXI-3IRO-7-6-1	18	80	5	400	400	–
EXPL-AXI-3IRO-7-7	5	50	5	250	211	–
EXPL-AXI-3IRO-7-8	18	60	5	300	300	–
EXPL-AXI-3IRO-7-14	5	20	5	100	85	–
EXPL-AXI-3IRO-7-15	5	20	5	100	78	–

Table 5.2: Summary of optimization parameters used for runs with Modified Adiabatic Efficiency as objective function

After a total of 6 more 3D design optimization runs with 206, 400, 212, 300, 85, 78 individual

design iterations, total of 1281 designs and 64 design parameters (Case:EXPL-AXI-3IRO-7-15), the 3 blade row adiabatic efficiency (Fine grid, Non-Linear Harmonic unsteady model) was 88.159%, Rotor-only: 93.85%, IGV-Rotor : 91.626% and Rotor-OGV : 90.375%. The modified adiabatic efficiency was 87.40%. The Pressure Ratios: 3 blade row: 1.309, rotor-only: 1.331, IGV-Rotor: 1.322 Rotor-OGV: 1.317, Grid Size 3,725,907 nodes. The power consumption was 266.25kW with atmospheric inlet conditions.

Based on post processing the results of Case EXPL-AXI-3IRO-7-15, the effects of the inclusion of modified adiabatic efficiency could be easily noticed. At this point I would like to compare the results pre and post using modified adiabatic efficiency (MAE).

5.3 Comparison of line plots, pre and post using MAE at different stations

Total Pressure profiles at the exit of the Propulsor are shown in Fig. 5.4 and 5.5.

Figure 5.4: Case 6-4: PT @ OGV outlet: Pre MAE

Figure 5.5: Case 7-15: PT @ OGV outlet: Post MAE

The Total pressure profile coming out of the propulsor shows that the radial Total pressure gradient is lower post MAE. This is an indication of lower swirl component of the air velocity. This is beneficial for the overall efficiency of the propulsor and downstream system.

Total Temperature profiles at the exit of the Propulsor are shown in Fig. 5.6 and 5.7.

Figure 5.6: Case 6-4: TT @ OGV outlet: Pre MAE

Figure 5.7: Case 7-15: TT @ OGV outlet: Post MAE

Post MAE it can be seen that the total temperature increases even more at the hub region (span

0 - 0.1). Also, the work extracted from 0.6 to 0.9 span locations drops off only to pick up after 0.9 span. This essentially states the fact that unlike in conventional axial flow fan designs, the objective in this design process was always to unload the tip and try to get more work done from the mid section and hub region of the rotor blade. This is because of the tip strong distortion coming into the propulsor. Extracting a lot of work near the hub region is not always a good idea because of the rotor tangential velocity. This area of the hub in the future will need to be revisited from work distribution point of view. Overall the mid section to above hub region looks good.

rV_θ profiles at the exit of the Propulsor are shown in Fig. 5.8 and 5.9.

Figure 5.8: Case 6-4: rV_θ @ OGV outlet: Pre MAE

Figure 5.9: Case 7-15: rV_θ @ OGV outlet: Post MAE

Flow angle profiles at the exit of the Propulsor are shown in Fig. 5.10 and 5.11.

Figure 5.10: Case 6-4: Flow angle @ OGV outlet: Pre MAE

Figure 5.11: Case 7-15: Flow angle @ OGV outlet: Post MAE

Axial velocity, V_z profiles at the exit of the Propulsor are shown in Fig. 5.12 and 5.13.

Figure 5.12: Case 6-4: Axial velocity @ OGV outlet: Pre MAE

Figure 5.13: Case 7-15: Axial velocity @ OGV outlet: Post MAE

Comparisons of flow angles and Axial velocity, V_z pre and post usage of MAE indicates how useful the objective function turned out to be. The flow angles coming out of the OGV have been reduced drastically, if not zero. The profile of V_z coming out of the OGV is also straightened out to a great extent. This essentially indicates lower mixing out losses at the OGV downstream.

5.4 Comparison of B2B contour plots, pre and post using MAE at 10% span

Static pressure contours are shown in Fig. 5.14 and 5.15 (000, NLH - 3 Harmonics)

Figure 5.14: Case 6-4(Pre MAE): Static Pressure @ 10% span

Figure 5.15: Case 7-15(Post MAE): Static Pressure @ 10% span

Contours of static pressure show that the loading has increased on the OGV in Fig. 5.15. This is not a very good sign and generally indicates probable flow separation. This region of the OGV needs to be revisited from the aspect of blade loading. An iso-surface of negative velocity will actually bring out the flow separation regions.

Iso-surface contours of Axial velocity -0.5 m/s are shown in Fig. 5.16 and 5.17 (000, NLH - 3 Harmonics)

Figure 5.16: Case 6-4(Pre MAE): Iso-surface for $V_z = -0.5 \text{ m/s}$

Figure 5.17: Case 7-15(Post MAE): Iso-surface for $V_z = -0.5 \text{ m/s}$

The Iso-surface contours clearly show that the flow separation regions have significantly reduced over design improvements from Case 6-4 to case 7-15. The OGV hub, tip and Rotor hub pressure side are the regions where most improvement is observed in terms of reduction of flow separation region.

Mrel contours are shown in Fig. 5.18 and 5.19 (000, NLH - 3 Harmonics)

Figure 5.18: Case 6-4(Pre MAE): Mrel @ 10% span

Figure 5.19: Case 7-15(Post MAE): Mrel @ 10% span

The amount of diffusion taking place in the OGV of Fig 5.19 is high as compared to the one in case of Fig. 5.18. Also the amount of turning in the OGV is also very high in case of Fig. 5.19 as compared to Fig. 5.18. Also, the low Mach number region in Case 6-4 ,Fig 5.18 is significantly reduced in Case 7-15, Fig 5.19 indicating lesser flow separation region.

Entropy contours are shown in Fig. 5.20 and 5.21 (000, NLH - 3 Harmonics)

Figure 5.20: Case 6-4(Pre MAE): Entropy @ 10% span

Figure 5.21: Case 7-15(Post MAE): Entropy @ 10% span

The reason the incoming entropy is different for Case 7-15 as compared to Case 6-4 is that the inlet profiles of PT are different. The entropy contours of pre and post MAE shows some encouraging results. Overall the entropy associated with the flow at the hub region has reduced on each row. The high entropy region at the TE of OGV for Case 6-4, Fig. 5.20 significantly reduced in Case 7-15, Fig. 5.21. The wakes at the OGV downstream are also less in Case 6-4, Fig. 5.20 as compared to Case 7-15, Fig. 5.21.

5.5 Comparison of B2B contour plots, pre and post using MAE at 90% span

Relative Mach Number contour are shown in Fig. 5.22 and 5.23 (000, NLH - 3 Harmonics)

Figure 5.22: Case 6-4(Pre MAE): Mrel @ 90% span

Figure 5.23: Case 7-15(Post MAE): M_{rel} @ 90% span

The Relative Mach number at 90% span contour plots show the shock structure near the rotor tip. It can be seen from Fig. 5.23 that the strength of the shock has reduced considerably. Also, the shock has moved a bit towards the TE of the rotor. Also, the shock in Fig 5.22 is more of a normal shock and the one in Fig. 5.23 seems to be getting converted to an oblique shock.

Absolute Total Pressure contour are shown in Fig. 5.24 and 5.25 (000, NLH - 3 Harmonics)

Figure 5.24: Case 6-4(Pre MAE): PT @ 90% span

Figure 5.25: Case 7-15(Post MAE): PT @ 90% span

Entropy contour are shown in Fig. 5.26 and 5.27 (000, NLH - 3 Harmonics)

Figure 5.26: Case 6-4(Pre MAE): Entropy @ 90% span

Figure 5.27: Case 7-15(Post MAE): Entropy @ 90% span

The entropy contours shows some encouraging results as the overall entropy of the flow in IGV and Rotor seems to have lowered in post MAE case as compared to pre MAE case.

Chapter 6

Performance Runs

The analysis till now has been concentrating on the design point adiabatic efficiency and modified adiabatic efficiency. But an important aspect of any turbomachinery design is also its off design performance characteristics, its stall margin and range of operation. This thesis does not include any study on operating range but it is something which should be worked on for the future. The 100% speed lines are compared in Figs 6.1-6.3 for the pre-MAE case 6-4 and the post-MAE case 7-15. Recall that the tip clearance was increased dramatically, and the rpm reduced for Case 7-15. Also the inlet profiles are different. The points on the performance curves Fig 6.1- 6.4 are evaluated using exit Mach number as the exit boundary condition. It allows for a design to proceed in a similar fashion to an operating line. As the efficiency changes, the static pressure is a result. The grid size used for all the evaluations needed for performance curve was 111.

Figure 6.1: Adiabatic Efficiency vs Mass flow

Figure 6.2: Pressure ratio vs Mass flow

Figure 6.3: Comparison of MAE on changing objective function

Figure 6.4: Comparison of AE and MAE

Fig 6.1 shows the adiabatic efficiency at different mass flow rates, comparing it for pre and post usage of MAE as the objective function. It appears that the usage of MAE as the objective function has been detrimental to the design process. But it's not, MAE is a better metric for assessing the propulsor. Fig 6.2 compares the pressure ratio to the mass flow pre and post usage of MAE as objective function. The operating range is similar for both designs. Fig 6.3 shows how the MAE compares pre and post usage of MAE as the objective function. This graph shows that the peak MAE is obtained post usage of MAE as the objective function. This is an indication that the design has progressively improved over successively optimizations. Again, these are not the same rpm and tip clearance. Fig 6.4 shows a comparison of the MAE and AE of the latest design. It should be noted that the MAE will always be less than the AE, but the only objective should be to bring them as close as possible.

Chapter 7

Conclusion and Future work

7.1 Conclusion

The final optimized design is highly dependent on the objective function to be optimized. A Modified Adiabatic Efficiency (MAE) has been defined to account for the loss of kinetic energy at the OGV. This lost kinetic energy is essentially an approximation of the mixing loss of the tangential, axial and radial components of velocity at the OGV exit. The three blade-row adiabatic efficiency and MAE for an intermediate design case is 88.91% and 86.10% respectively. Upon further optimization and a different incoming distortion, using MAE as the objective function improved the MAE to 87.40% with an 88.159% adiabatic efficiency for the final design case presented in the thesis. The rotor-only adiabatic efficiency for this case is 93.85%, the IGV-Rotor two blade-row efficiency is 91.62% and the Rotor-OGV two blade-row efficiency is 90.37%. The IGV is necessary to support the nacelle in this configuration. It also allows for the rotor work distribution to be optimized. MAE as the objective function leads to a decrease in Kinetic Energy variation at the OGV exit. The final design presented is not an optimum and indicates flow separation near the trailing edge of the hub to mid-span region of the rotor and also at the trailing edge of OGV hub region. Future efforts on the design process are also presented that can lead to improvement in the MAE.

7.2 Future work

There are many things that can be pursued as further work in this design process. A few major ones include.

1. The design of the IGV and OGV needs significant improvement and the flow separation region needs to be reduced if not eliminated completely.
2. The sampling method used for all the optimizations was random which is not the best option. Latin Hypercube Sampling method is a better way of exploring the complete design space and should be explored as future work.
3. The operating range was never explicitly part of an objective function, and no optimizations were carried out to improve the propulsor's stall margin. Operating range is a very important aspect of any turbomachine and should be part of future studies.
4. Another way of handling the propulsor downstream system would be to include the propulsor and the free stream as the complete computational domain and conduct the simulations accordingly. This would require a nacelle design integration with correct boundary layer state on the nacelle as well as a tail cone design capability. It would be more computationally intensive, but then adiabatic efficiency could be retained as the objective function for design optimizations.
5. The optimization process needs improvement. Genetic algorithm may require thousands of function evaluations to get a local minimum and with the limited number of iterations performed for each optimization run, the solution may not be the absolute optimum design. The number of function evaluations needs to be increased or switch to a different algorithm such as Adjoint method of optimization. This method is as part of a gradient-based optimization approach better suited at finding absolute local minimum or maximum for a large number of design variables.

Bibliography

- [1] Hidetaka Okui, Tom Verstraete, R. A. Van den Braembussche, Zuheyr Alsalihi “Three-Dimensional Design and Optimization of a Transonic Rotor in Axial Flow Compressors,” Journal of Turbomachinery, May 2013, Vol. 135.
- [2] Siddappaji, K., Turner, M. G., and Merchant, A., 2012. “General capability of parametric 3d blade design tool for turbomachinery”. In Proceedings of ASME Turbo Expo 2012.
- [3] Syed Moez Hussain Mahmood “Optimization Capabilities for Axial Compressor Blades and Seal Teeth Cavity,” Master’s thesis, University of Cincinnati.
- [4] A.R.Wadia, P.N.Szucs, D.W.Crall “Inner Workings of Aerodynamic Sweep,” International Gas Turbine & Aeroengine Congress & Exhibition Paper, (97-GT-401).
- [5] A.R.Wadia, P.N.Szucs, D.W.Crall “Forward Swept Rotor Studies in Multistage Fans with Inlet Distortion ,” ASME Turbo Expo Paper, (GT-2002-30326).
- [6] Chen, H.L., Turner, M.G., Siddappaji, K. and Mahmood, S.M.H. (2017) "Flow Diagnosis and Optimization Based on Vorticity Dynamics for Transonic Compressor/Fan Rotor." Open Journal of Fluid Dynamics , 7, 40-71
- [7] H.D.Weingold, R.J.Nuebert, R.F.Behlke, G.E.Potter "Bowed Stators: An Example of CFD Applied to Improve Multistage Compressor Efficiency." ASME Paper, Journal of Turbomachinery, 1997, Vol. 119
- [8] Milan Banjac, Milan V. Petrovic, Alexander Wiedermann “Secondary Flows, Endwall Effects, and Stall Detection in Axial Compressor Design ,” ASME Paper, Journal of Turbomachinery, 2015, Vol. 137

- [9] Simon J. Gallimore, John J. Bolger, Nicholas A. Cumpsty, Mark J. Taylor, Peter I. Wright, James M. M. Place “The Use of Sweep and Dihedral in Multistage Axial Flow Compressor Blading—Part I: University Research and Methods Development ,” ASME Paper, Journal of Turbomachinery, 2002, Vol. 124
- [10] Aniwat Tiralap, Choon S. Tan, Eric Donahoo, Matthew Montgomery, Christian Cornelius “Effects of Rotor Tip Blade Loading Variation on Compressor Stage Performance,” ASME Paper, Journal of Turbomachinery, 2017, Vol. 139
- [11] E. J. Gunn, C. A. Hall “Non-Axisymmetric Stator Design for Boundary Layer Ingesting Fans,” ASME Turbo Expo Paper, (GT2017-63082)

Chapter 8

Appendix

A1: Scripts and input files for Optimization using T-AXI and DAKOTA for optimizing adiabatic efficiency by varying aspect ratio (SOGA)

DAKOTA input file: dakota_TAXI_SOGA.in

```
environment
  tabular_graphics_data
  tabular_graphics_file='dakota_TAXI_SOGA.dat'
```

```
method
  output verbose
  max_iterations = 3200
  max_function_evaluations = 4000
  soja
  seed = 10983
  population_size = 400
  initialization_type unique_random
  crossover_type shuffle_random
  num_offspring = 2
  num_parents = 2
  crossover_rate = 0.8
  mutation_type replace_uniform
  mutation_rate = 0.1
```

```
%SET HERE PARAMETERS FOR SOGA
fitness_type merit_function
replacement_type favor_feasible
convergence_type best_fitness_tracker
percent_change = 0.05
num_generations = 10
```

```
variables
```

```

continuous_design = 4
cdv_lower_bounds  0.490281  0.565591  1.00504  1.001349
cdv_upper_bounds  0.663321  0.765211  1.35975  1.354766
cdv_descriptors   'RhxFE'    'RtxFE'    'ShxFE'    'StxFE'

interface

fork
  analysis_driver = './runTAXI'
  parameters_file = 'params.in'
  results_file = 'results.out'
  failure_capture = recover 1.0000 0.9000 0.0
  aprepro

responses

descriptors = '1_minus_Eff' 'PR' 'Alfa_ogv' 'M_ogv' 'eff'
num_objective_functions = 1

num_nonlinear_inequality_constraints = 4
nonlinear_inequality_upper_bounds = 1.35 45.00 0.80 1.00
nonlinear_inequality_lower_bounds = 1.00 0.00 0.00 0.87
no_gradients
no_hessians

```

T-AXI bash script:

```
#!/bin/sh
# TAXI simulation Dakota system call script
# See Advanced Simulation Code Interfaces chapter in Users Manual

# They are passed to the script from dakota_TAXI_SOGA.in as:
# $1 is params.in FROM Dakota
# $2 is results.out returned to Dakota
# -----
# PRE-PROCESSING
# -----
# Incorporate the parameters from DAKOTA into the template, writing ros.in
# Use the following line if SNL's APREPRO utility is used instead of DPrePro.
# ../aprepro -c '*' -q --nowarning ros.template ros.in

# Free vortex design creates equal rVt at hub and tip
# Reading below the delta rVt value which is the variable parameter to keep
the work distribution constant
# This delta is added to the hub rVt and subtracted from the tip rVt.

dprepro $1 T-Axi.param.template T-Axi.param

# Adding delta to hub rVt and subtracting from tip rVt

python update_stackfile.py Case6
# -----
# ANALYSIS
# -----
# Running TAXI,
#Future change the address to environmental variable
taxi_driver -ga Case6_new 11 >taxi_run.log
# -----
# POST-PROCESSING
# -----f
# Simulation driven error detection and mitigation
#Dakota Use manual 6.4 Chapter 19
#
err_flag=$(grep -i "Aborting T-Axi" taxi_run.log | wc -l)
if [ $err_flag -gt 0 ];
then
    echo "FAIL" > $2
exit
else
#
eff=$(grep -i -m 1 'ADBEFF' taxi_run.log | wc -l)
if [ $eff -le 0 ];
```

```

then      # need to set values out of limits with fail capture for *.in file
echo "FAIL" > $2
exit
else # ADBEFF tag is found
    eff_tag=$(grep -i 'ADBEFF' taxi_run.log | cut -d ":" -f2 | bc -l)
    if [ $eff_tag -le 0 ];
    then      # need to set values out of limits with fail capture for *.in file
        echo "FAIL" > $2
        exit
    else      # efficiency is greater than 0
        eff=$(grep -i -m 1 'ADBEFF' taxi_run.log | cut -d ":" -f2)
        PR=$(sed -n '365p' < bladedata-Case6_new.dat | cut -d "=" -f2)
        Alfa_ogv=$(sed -n '251p' < bladedata-Case6_new.dat | awk '{print $10}')
        M_ogv=$(sed -n '251p' < bladedata-Case6_new.dat | awk '{print $8}')
    fi
fi
fi
fi
echo $((sed 's/[E]/*10~/g' <<< $(echo 1-$eff)) | bc -l) >> $2      # added
when eff is exponential number, and Dakota minimizes solutions
echo $PR >> $2
echo $Alfa_ogv >> $2
echo $M_ogv >> $2
echo $eff >> $2

```

Python script update_sackfile.py

```
# Python script to read the stack file and T-Axi.param file and update the rVt info
# run> python update_stackfile.py casename
# Original script by Kiran Siddappaji
# Further edition by Pritesh
# changes for Rotor and Stator inclusion
# University of Cincinnati
import sys, os, errno, subprocess
import math, time, csv
import numpy as np

#-----
def file_len(fname):
    with open(fname) as f:
        for i, l in enumerate(f):
            pass
        return i + 1
#-----

# Variable initialization
stacklines = []

# Reading the stack file
casename = sys.argv[1]
filename = str(casename)+'.tstk'
file_length = file_len(filename)
f = open(filename, "r")
lineread = f.readlines()
# Parsing the file
#change for reading no of rotors and stators
for num, line in enumerate(lineread, 1):
    linelist = line.split()

# Reading the T-Axi.param file
filename2 = 'T-Axi.param'
f2 = open(filename2, "r")
lineread2 = f2.readlines()

RhxFE = float(lineread2[0])
RtxFE = float(lineread2[1])
ShxFE = float(lineread2[2])
StxFE = float(lineread2[3])
f2.close()

# Write the updated stack file for T-Axi run
f3 = open("Case6_new.tstk", "w")
for i in range(0,37): #file_length):
```

```

f3.write(lineread[i])

j=37
f3.write(" ")
f3.write(lineread[j].split()[0] + " ")
f3.write(lineread[j].split()[1] + " ")
f3.write(('%.6f'%RhLE) + " ")
f3.write(lineread[j].split()[3] + " ")
f3.write(lineread[j].split()[4] + " ")
f3.write(lineread[j].split()[5] + " ")
f3.write((lineread[j].split()[6] + " ")+"\\n")
k=38
f3.write(" ")
f3.write(lineread[k].split()[0] + " ")
f3.write(lineread[k].split()[1] + " ")
f3.write(('%.6f'%RtLE) + " ")
f3.write(lineread[k].split()[3] + " ")
f3.write(lineread[k].split()[4] + " ")
f3.write(lineread[k].split()[5] + " ")
f3.write((lineread[k].split()[6] + " ")+"\\n")
# print default value for span 999
n=39
f3.write(" ")
f3.write((lineread[n].split()[0] + " ") + "\\n" )
#Stator
m=40
f3.write(" ")
f3.write(lineread[m].split()[0] + " ")
f3.write(lineread[m].split()[1] + " ")
f3.write(lineread[m].split()[2] + " ")
f3.write((lineread[m].split()[3] + " ") + "\\n" )

p=41
f3.write(" ")
f3.write(lineread[p].split()[0] + " ")
f3.write(lineread[p].split()[1] + " ")
f3.write(('%.6f'%ShLE) + " ")
f3.write(lineread[p].split()[3] + " ")
f3.write(lineread[p].split()[4] + " ")
f3.write(lineread[p].split()[5] + " ")
f3.write((lineread[p].split()[6] + " ") + "\\n")
q=42
f3.write(" ")
f3.write(lineread[q].split()[0] + " ")
f3.write(lineread[q].split()[1] + " ")
f3.write(('%.6f'%StLE) + " ")
f3.write(lineread[q].split()[3] + " ")
f3.write(lineread[q].split()[4] + " ")

```

```
f3.write(lineread[q].split()[5] + "  ")
f3.write(lineread[q].split()[6] + "  ")

f3.close
```

A2: Scripts and input files for Optimization using T-AXI and DAKOTA for optimizing Kinetic energy at OGV exit and adiabatic efficiency by varying rV_θ of rotor and IGV (MOGA)

DAKOTA input file: dakota_TAXI_MOGA.in

```
# Dakota Input File: dakota_TAXI_MOGA.in
# Usage:
# Original file by Kiran,Futher changes by Pritesh
# dakota -i dakota_TAXI_MOGA.in -o dakota_TAXI_MOGA.out > dakota_TAXI_MOGA.stdout
environment
#
#graphics
  tabular_graphics_data
    tabular_graphics_file='dakota_TAXI_MOGA.dat'
  #single_method

method
output verbose
max_iterations = 7500
max_function_evaluations = 9000
  moga # single objective genetic algorithm
seed = 10983
population_size = 300
initialization_type unique_random
crossover_type shuffle_random
num_offspring = 2
num_parents = 2
crossover_rate = 0.8
mutation_type replace_uniform
mutation_rate = 0.1
#SET HERE PARAMETERS FOR MOGA
  fitness_type domination_count
  replacement_type below_limit = 24 # elitist
  shrinkage_percentage = 0.8
  convergence_type metric_tracker
    percent_change = 0.05
    num_generations = 30
  final_solutions = 10

variables
#change the design variables and their bounds
continuous_design = 9
cdv_lower_bounds      -0.1000 -0.1    0.215694   0.10 0.10   0.1      0.1    0.1 0.1
cdv_upper_bounds      0.1000 0.1      0.263626   1.00 1.00  1.15 1.15 1.0 1.0
```



```

cdv_descriptors 'I_rvt_req' 'del_rvt' 'req_rvt_rotor' 'rVt_zero' 'rVt_one' 'rVt_two' 'rVt_three'
'rVt_four' 'rVt_five'

interface
    fork                                     #asynchronous evaluation_concurrency = 45
        analysis_driver = './runTAXI'
        parameters_file = 'params.in'
        results_file = 'results.out'
        failure_capture = recover 1.000 20.00 1.400 50.000 1.000 0.0
        aprepro
responses
# Objective function: Maximize Efficiency: minimize 1-eta
# Change : no value for 1-eff
descriptors = '1_minus_Eff' 'Sum_KE' 'PR' 'Alfa_ogv' 'M_ogv' 'eff'
    num_objective_functions = 2
# Constraint: No constraint
    num_nonlinear_inequality_constraints = 4
    nonlinear_inequality_upper_bounds = 1.35 45.00 0.80 1.00
    nonlinear_inequality_lower_bounds = 1.30 0.00 0.00 0.87
    no_gradients
    no_hessians

```

T-AXI bash script:

```
#!/bin/sh
# TAXI simulation Dakota system call script
# See Advanced Simulation Code Interfaces chapter in Users Manual
# They are passed to the script from dakota_TAXI_MOGA.in as:
# $1 is params.in FROM Dakota
# $2 is results.out returned to Dakota

# PRE-PROCESSING
# Incorporate the parameters from DAKOTA into the template, writing ros.in
# Use the following line if SNL's APREPRO utility is used instead of DPrePro.
# ../aprepro -c '*' -q --nowarning ros.template ros.in

# Free vortex design creates equal rVt at hub and tip
# Reading below the delta rVt value which is the variable parameter to keep the work distribution
# This delta is added to the hub rVt and subtracted from the tip rVt.

dprepro \"$1 T-Axi.param.template T-Axi.param

# Adding delta to hub rVt and subtracting from tip rVt

python script1.py Case6

# ANALYSIS

# Running TAXI,
#Future change the address to environmental variable
/home/mandalpi/T-AXI_new/taxi_driver -ga Case6_new 11 >taxi_run.log

python script2.py bladedata-Case6_new.dat

# POST-PROCESSING
# Simulation driven error detection and mitigation
#Dakota Use manual 6.4 Chapter 19

err_flag=$(grep -i "Aborting T-Axi" taxi_run.log | wc -l)
if [ $err_flag -gt 0 ];
then
    echo "FAIL" > $2
exit
else
eff=$(grep -i -m 1 'ADBEFF' taxi_run.log | wc -l)
if [ $eff -le 0 ];
then
    # need to set values out of limits with fail capture for *.in file
echo "FAIL" > $2
```

```

exit
else # ADBEFF tag is found

    eff_tag=$(sed -n '367p' < bladedata-Case6_new.dat | cut -d "=" -f2)
    if [ $eff_tag -le 0 ];
    then          # need to set values out of limits with fail capture for *.in file
        echo "FAIL" > $2
        exit
    else          # efficiency is greater than 0
        eff=$(sed -n '367p' < bladedata-Case6_new.dat | cut -d "=" -f2)
        PR=$(sed -n '365p' < bladedata-Case6_new.dat | cut -d "=" -f2)
        Alfa_ogv=$(sed -n '251p' < bladedata-Case6_new.dat | awk '{print $10}')
        M_ogv=$(sed -n '251p' < bladedata-Case6_new.dat | awk '{print $8}')
        Sum_KE=$(sed -n '1p' < Values.dat | awk '{print $0}')
    fi
fi
fi
echo $((sed 's/[E]/*10~/g' <<< $(echo 1-$eff)) | bc -l) >> $2
echo $Sum_KE >> $2
echo $PR >> $2
echo $Alfa_ogv >> $2
echo $M_ogv >> $2
echo $eff >> $2

```

Python script script1.py to update the stack file

```
# Python script to read the stack file and T-Axi.param file and update the rVt info
# run> python update_stackfile.py casename
# By Pritesh Mandal
# University of Cincinnati

import sys, os, errno, subprocess
import math, time, csv
import numpy as np
from scipy.interpolate import interp1d
import scipy.interpolate as si
import matplotlib.pyplot as plt

def file_len(fname):
    with open(fname) as f:
        for i, l in enumerate(f):
            pass
        return i + 1
f5 = open("Info.txt", "r")
lineread5 = f5.readlines()
#print(lineread5)

span=[]
for num, line in enumerate(lineread5, 1):
    linelist5 = line.split()
    if num == 1:
        ns = int(linelist5[0])
    if num == 2:
        for i in range(ns):
            span.append(float(linelist5[i]))
    if num == 3:
        nc = int(linelist5[0])
    if num == 4:
        RBlk = float(linelist5[0])
    if num == 5:
        SBlk = float(linelist5[0])
    if num == 6:
        IBlk = float(linelist5[0])

#print(n)

#STORE THE ABOVE INPUTS IN A SEPARATE TEXT FILE TO USE IT IN THE NEXT ITERATION

stacklines = []
```

```

# Reading the stack file
casename = sys.argv[1]
filename = str(casename)+'.tstk'
file_length = file_len(filename)
f = open(filename, "r")
lineread = f.readlines()

# Parsing the file
#change for reading no of rotors and stators
for num, line in enumerate(lineread, 1):
    linelist = line.split()
        #print linelist
    if num == 9:
        IhrLE = float(linelist[0])
    if num == 9:
        IhrTE = float(linelist[1])
    if num == 9:
        IhxLE = float(linelist[2])
    if num == 9:
        IhxTE = float(linelist[3])
    if num == 19:
        ItrLE = float(linelist[0])
    if num == 19:
        ItrTE = float(linelist[1])
    if num == 19:
        ItxLE = float(linelist[2])
    if num == 19:
        ItxTE = float(linelist[3])
        if num == 19:
            Iloss = float(linelist[4])
            if num == 19:
                rvt_i = float(linelist[6])

    if num == 22:
        RhrLE = float(linelist[0])
    if num == 22:
        RhrTE = float(linelist[1])
    if num == 22:
        RhxLE = float(linelist[2])
    if num == 22:
        RhxTE = float(linelist[3])
    if num == 32:
        RtrLE = float(linelist[0])
    if num == 32:
        RtrTE = float(linelist[1])
    if num == 32:
        RtxLE = float(linelist[2])
    if num == 32:

```

```

RtxTE = float(linelist[3])
    if num == 32:
Rloss = float(linelist[4])
    if num == 32:
rvt_r = float(linelist[6])

if num == 34:
Nstator = int(linelist[0])
if num == 35:
ShrLE = float(linelist[0])
if num == 35:
ShrTE = float(linelist[1])
if num == 35:
ShxLE = float(linelist[2])
if num == 35:
ShxTE = float(linelist[3])
if num == 45:
StrLE = float(linelist[0])
if num == 45:
StrTE = float(linelist[1])
if num == 45:
StxLE = float(linelist[2])
if num == 45:
StxTE = float(linelist[3])
    if num == 45:
Sloss = float(linelist[4])
    if num == 45:
rvt_s = float(linelist[6])
#print(req_rvt)

l=8

ImrTE=[]
for i in range(nc):
    ImrTE.append(lineread[i+1].split()[1])
ImrTE=np.asfarray(ImrTE,float)
#print(ImrTE)

RmrTE=[]
for i in range(nc):
    RmrTE.append(lineread[i+1+nc+2].split()[1])
RmrTE=np.asfarray(RmrTE,float)
#print(RmrTE)

# Reading the T-Axi.param file
filename2 = 'T-Axi.param'
f2 = open(filename2, "r")
lineread2 = f2.readlines()

```

```

req_rvt_igv= float(lineread2[0])

req_rvt_rotor= float(lineread2[1])
Upd_rVt_igv=[]
for i in range(ns+2):
    Upd_rVt_igv.append(float(lineread2[i+2])*req_rvt_igv)
#print(Upd_rVt_igv)
Upd_rVt_rotor=[]
for i in range(ns+2):
    Upd_rVt_rotor.append(float(lineread2[i+13])*req_rvt_rotor)
#print(Upd_rVt_rotor)

#req_rvt_h= req_rvt_igv - del_rvt
#req_rvt_t= req_rvt_igv + del_rvt

# IGV

Radius_CP = lmrTE      #CP: control points
#print(Radius_CP)

Radius_CP_array = np.asarray(Radius_CP)
Upd_rVt_array = np.asarray(Upd_rVt_igv)
#print(Upd_rVt_array)

# GENERATING THE X,Y COORDIANTES OF THE CUBIC SPLINE
t= range(len(Radius_CP_array))
#print(t)
ipl_t= np.linspace(0.0, len(Radius_CP_array) - 1, nc)

x_tup = si.splrep(t, Radius_CP_array, k=3)
y_tup = si.splrep(t, Upd_rVt_array, k=3)

x_list = list(x_tup)
xl = Radius_CP_array.tolist()
x_list[1] = xl + [0.0, 0.0, 0.0, 0.0]
#print(xl)

y_list = list(y_tup)
yl = Upd_rVt_array.tolist()
y_list[1] = yl + [0.0, 0.0, 0.0, 0.0]

X_Spline_igv = si.splev(ipl_t, x_list)
Y_Spline_igv = si.splev(ipl_t, y_list)

#print(X_Spline_igv)
#print(Y_Spline_igv)

```

```

m=[]
c=[]
for i in range(nc-1):
    m.append((Y_Spline_igv[i+1] - Y_Spline_igv[i])/(ImrTE[i+1] - ImrTE[i]))
    c.append(Y_Spline_igv[i] - (m[i]*ImrTE[i]))

Int=[]
for i in range(nc-1):
    Int.append(((m[i]/3)*((ImrTE[i+1] ** 3)-(ImrTE[i] ** 3)))
    + ((c[i]/2)*((ImrTE[i+1] ** 2)-(ImrTE[i] ** 2))))

sum(Int)
rvt_avg = (sum(Int)*2)/((ItrTE ** 2)-(IhrTE ** 2))
#print(rvt_avg)

delta_adder = req_rvt_igv - rvt_avg
#print(delta_adder)

# UPDATING RV_THETAS WITH THE ADDER

f_rvtheta_igv=[]
for i in range(nc):
    f_rvtheta_igv.append(Y_Spline_igv[i] + delta_adder)
#print(f_rvtheta_igv)

# FIXING UP ROTOR RVT

Radius_CP = RmrTE      #CP: control points
#print(Radius_CP)

Radius_CP_array = np.asarray(Radius_CP)
Upd_rVt_array = np.asarray(Upd_rVt_rotor)
#print(Upd_rVt_array)

# GENERATING THE X,Y COORDIANTES OF THE CUBIC SPLINE
t= range(len(Radius_CP_array))
#print(t)
ipl_t= np.linspace(0.0, len(Radius_CP_array) - 1, nc)

x_tup = si.splrep(t, Radius_CP_array, k=3)
y_tup = si.splrep(t, Upd_rVt_array, k=3)

x_list = list(x_tup)
xl = Radius_CP_array.tolist()
x_list[1] = xl + [0.0, 0.0, 0.0, 0.0]

y_list = list(y_tup)
yl = Upd_rVt_array.tolist()

```



```

y_list[1] = y1 + [0.0, 0.0, 0.0, 0.0]

X_Spline_rotor = si.splev(ipl_t, x_list)
Y_Spline_rotor = si.splev(ipl_t, y_list)

#print(X_Spline_rotor)
#print(Y_Spline_rotor)

m=[]
c=[]
for i in range(nc-1):
    m.append((Y_Spline_rotor[i+1] - Y_Spline_rotor[i])/(RmrTE[i+1] - RmrTE[i]))
    c.append(Y_Spline_rotor[i] - (m[i]*RmrTE[i]))
Int=[]
for i in range(nc-1):
    Int.append(((m[i]/3)*((RmrTE[i+1] ** 3)-(RmrTE[i] ** 3)))
    + ((c[i]/2)*((RmrTE[i+1] ** 2)-(RmrTE[i] ** 2))))

sum(Int)
rvt_avg = (sum(Int)*2)/((RtrTE ** 2)-(RhrTE ** 2))
#print(rvt_avg)

delta_adder = req_rvt_rotor - rvt_avg
#print(delta_adder)

# UPDATING RV_THETAS WITH THE ADDER

f_rvtheta_rotor=[]
for i in range(nc):
    f_rvtheta_rotor.append(Y_Spline_rotor[i] + delta_adder + req_rvt_igv)
#print(f_rvtheta_rotor)

# Write the updated stack file with new span and x coordiantes
f3 = open("Case6_new.tstk", "w")
for i in range(0,8): #file_length):
    f3.write(lineread[i])

# Updating igv hub

f3.write(" ")
f3.write(lineread[1].split()[0] + " ")
f3.write(lineread[1].split()[1] + " ")
f3.write(lineread[1].split()[2] + " ")
f3.write(lineread[1].split()[3] + " ")
f3.write(lineread[1].split()[4] + " ")
f3.write(('%.6f'%IBlk) + " ")
f3.write((((%.6f'%f_rvtheta_igv[0]) + " ") + "\n")

```

```

# Updating intermediate span locations
for i in range(nc-2):
    f3.write(" ")
    f3.write(lineread[i+1+1].split()[0] + " ")
    f3.write(lineread[i+1+1].split()[1] + " ")
    f3.write(lineread[i+1+1].split()[2] + " ")
    f3.write(lineread[i+1+1].split()[3] + " ")
    f3.write(lineread[i+1+1].split()[4] + " ")
    f3.write(('%.6f'%IBlk) + " ")
    f3.write((((%.6f'%f_rvtheta_igv[i+1]) + " ") + "\n")\

\# Updating igv tip
f3.write(" ")
f3.write(('%.6f'%ItrLE) + " ")
f3.write(('%.6f'%ItrTE) + " ")
f3.write(('%.6f'%ItxLE) + " ")
f3.write(('%.6f'%ItxTE) + " ")
f3.write(('%.6f'%Iloss) + " ")
f3.write(('%.6f'%IBlk) + " ")
f3.write((((%.6f'%f_rvtheta_igv[nc-1]) + " ") + "\n")

sv= 999.000000
f3.write(" ")
f3.write((((%.6f'%sv) + " ") + "\n")

# Updating Rotor
a= 1+nc+1
f3.write(" ")
f3.write(lineread[a].split()[0] + " ")
f3.write(lineread[a].split()[1] + " ")
f3.write(lineread[a].split()[2] + " ")
f3.write(lineread[a].split()[3] + "\n")

x=a+1
f3.write(" ")
f3.write(lineread[x].split()[0] + " ")
f3.write(lineread[x].split()[1] + " ")
f3.write(lineread[x].split()[2] + " ")
f3.write(lineread[x].split()[3] + " ")
f3.write(lineread[x].split()[4] + " ")
f3.write(('%.6f'%RBlk) + " ")
f3.write(('%.6f'%f_rvtheta_rotor[0]) + "\n")

\# Updating intermediate span locations
for i in range(nc-2):
    f3.write(" ")

```

```

        f3.write(lineread[i+x+1].split()[0] + " ")
        f3.write(lineread[i+x+1].split()[1] + " ")
        f3.write(lineread[i+x+1].split()[2] + " ")
        f3.write(lineread[i+x+1].split()[3] + " ")
        f3.write(('%.6f'%Rloss) + " ")
        f3.write(('%.6f'%RBlk) + " ")
        f3.write((((%.6f'%f_rvtheta_rotor[i+1]) + " ") + "\n")

# Updating rotor tip
f3.write(" ")
f3.write(('%.6f'%RtrLE) + " ")
f3.write(('%.6f'%RtrTE) + " ")
f3.write(('%.6f'%RtxLE) + " ")
f3.write(('%.6f'%RtxTE) + " ")
f3.write(('%.6f'%Rloss) + " ")
f3.write(('%.6f'%RBlk) + " ")
f3.write((((%.6f'%f_rvtheta_rotor[nc-1]) + " ") + "\n")

sv= 999.00000
f3.write(" ")
f3.write((((%.6f'%sv) + " ") + "\n")

# Updating Stator
m= 0.00
f3.write(" ")
f3.write(('d'%Nstator) + " ")
f3.write(('%.6f'%m) + " ")
f3.write(('%.6f'%m) + " ")
f3.write((((%.6f'%m) + " ") + "\n")

n= 1+(2*nc)+4
for i in range(nc):
    f3.write(" ")
    f3.write(lineread[i+n].split()[0] + " ")
    f3.write(lineread[i+n].split()[1] + " ")
    f3.write(lineread[i+n].split()[2] + " ")
    f3.write(lineread[i+n].split()[3] + " ")
    f3.write(lineread[i+n].split()[4] + " ")
    f3.write(lineread[i+n].split()[5] + " ")
    f3.write(lineread[i+n].split()[6] + "\n")

f3.close

```

Python script script2.py to evaluate sum of kinetic energy at OGV exit

```
# Original script by Ritangshu Giri
# Modifications to the script by Pritesh Mandal
import sys, os, errno, subprocess
from operator import itemgetter
from collections import defaultdict
#import pandas as pd
import numpy as np
from numpy import trapz
import csv
import re
from matplotlib.backends.backend_pdf import PdfPages
import matplotlib.pyplot as plt
from matplotlib import style
import matplotlib.tri
import scipy.io as sio
from scipy.integrate import.simps
from scipy.optimize import curve_fit
from scipy.integrate import quad
import matplotlib as mpl
import sys, os, errno, subprocess
import math, time, csv
from scipy.interpolate import interp1d
import scipy.interpolate as si

mpl.style.use('ggplot')

script = sys.argv[0]
file_name = sys.argv[1]
#print(file_name)

# Opening an user defined blade_data output file (.dat) from TAXI
file_open = open(file_name, 'r')
#plot = False

# Read the lines of the file and store it in a list

contents = file_open.readlines()

#Parsing useful data and storing it as a list

data = []

for line in contents:
```

```

    line = line.strip()
    columns = line.split()
    data.append(columns)

A = [data[273][4], data[274][4], data[275][4], data[276][4], data[277][4], data[278][4],
data[279][4], data[280][4], data[281][4], data[282][4], data[283][4], data[284][4],
data[285][4], data[286][4], data[287][4], data[288][4], data[289][4], data[290][4], data[291][4],
data[292][4]]

B = [data[273][6], data[274][6], data[275][6], data[276][6], data[277][6], data[278][6],
data[279][6], data[280][6], data[281][6], data[282][6], data[283][6], data[284][6],
data[285][6], data[286][6], data[287][6], data[288][6], data[289][6], data[290][6], data[291][6],
data[292][6]]

Total_Temperature_ratio_row_3_outlet = np.array(A).astype(np.float)
Static_Temperature_ratio_row_3_outlet = np.array(B).astype(np.float)

T_tot_row_3_outlet = np.zeros(20)
T_sta_row_3_outlet = np.zeros(20)
KE = np.zeros(20)
RS_KE = np.zeros(20)

k = 0
for k in range(len(Total_Temperature_ratio_row_3_outlet)):
    T_tot_row_3_outlet[k] = 288.15*Total_Temperature_ratio_row_3_outlet[k]
    k += 1

j = 0
for j in range(len(Static_Temperature_ratio_row_3_outlet)):
    T_sta_row_3_outlet[j] = 288.15*Static_Temperature_ratio_row_3_outlet[j]
    j += 1

i = 0
for i in range(len(Static_Temperature_ratio_row_3_outlet)):
    KE[i] = T_tot_row_3_outlet[i] - T_sta_row_3_outlet[i]
    i += 1

Sum_KE = sum(KE)

f1 = open("Values.dat", "w")
f1.write(('%.6f'%Sum_KE) + "newline")

```

A3: Scripts for 3D optimization using FINE/Turbo and DAKOTA with Modified Adiabatic Efficiency as the Objective function

DAKOTA input file dakota_CFD_gen.in

```
#!/bin/sh
# Dakota Input File: dakota_CFD_gen.in
# Usage:
#   dakota -i dakota_CFD_gen.in -o dakota_CFD_gen.out > dakota_CFD_gen.stdout
# for restart from saved:
#   dakota -i dakota_CFD_gen.in -read_restart -o dakota_CFD_gen.out >> dakota_CFD_gen.stdout
# ---- I will use JEGA library for genetic algorithm:
environment
  #graphics
  tabular_graphics_data
    tabular_graphics_file = 'dakota_CFD.dat'
  # single_method

method
output verbose
  max_iterations = 250
#max_function_evaluations = 375
max_function_evaluations = 210 # = num_generations*population_size
  sogas # multiple objective genetic algorithm
seed = 10983
#population_size = 25
population_size = 35
initialization_type unique_random
crossover_type shuffle_random
num_offspring = 2
num_parents = 2
crossover_rate = 0.8
mutation_type replace_uniform
mutation_rate = 0.1

#SET HERE PARAMETERS FOR SOGA
fitness_type merit_function
replacement_type favor_feasible # elitist
convergence_type best_fitness_tracker
percent_change = 0.05
num_generations = 6

# model # not used in genetic algorithm
# single

variables
```

```

continuous_design = 5
# cdv_initial_point      0.0      0.0      0.0      0.0      0.0
cdv_lower_bounds        -0.349066 -0.349066 -0.349066 -0.349066 -0.349066
cdv_upper_bounds         0.0      0.0      0.0      0.0      0.0
cdv_descriptors          'delta1'   'delta2'   'delta3'   'delta4'   'delta5'

interface
analysis_driver = 'sh CFD_optimization_script.sh'
fork
#asynchronous
#evaluation\_concurrency = 2
#local_evaluation_self_scheduling
parameters_file = 'params.in'
results_file = 'results.out'
failure_capture = recover 1.000
aprepro
# file_tag
# file_save
# when using conmin_frcg (above) with analytic_gradients (below),
# need to turn off the active set vector as rosenbrock_bb does not parse it.

responses
# Objective function: Maximize Efficiency: minimize 1-eta
# Change : no value for 1-eff
descriptors = '1_minus_Eff'
num_objective_functions = 1
#Constraint: No constraint
# num_nonlinear_inequality_constraints = 1
# nonlinear_inequality_upper_bounds = 1.35
# nonlinear_inequality_lower_bounds = 1.30
no_gradients
no_hessians

```

Shell script CFD_optimization_script.sh for optimizing on Modified Adiabatic Efficiency

```
#!/bin/sh
# cfd simulation Dakota system call script
# See Advanced Simulation Code Interfaces chapter in Users Manual
# They are passed to the script from dakota_3dbgb_Ansys_grad.in as:
# $1 is params.in FROM Dakota
# $2 is results.out returned to Dakota

#-----
# Functions:
#-----

# Function to test for bad runs:
#-----
function Running {
    "$@"
    local status=$?
    if [ $status -ne 0 ]; then
        echo "error with $1 .... Script exit" >&2
    fi
    return $status
}

# Function to find wait file:
#-----
function finding {
    if [[ -n $(find . -type f -name \*.mf) ]]; then
        indicator=1;
    #echo $indicator
    else
        indicator=0;
    #echo $indicator
    fi
}

# Function to wait for file existence:
#-----
function waiting {
    echo "Waiting for current process to finish ... "
    n=1
    while [ $indicator -eq 0 ] #
do
```



```

sleep 5
finding
if [ $n -gt $waittime ]; then
echo 'process timeout ... time exceeds' $(( $n * 5)) 'seconds'
break;
fi
#echo 'timer :' $n
n=$(( $n + 1 ))
done
echo 'process time =' $(( $n * 5)) 'seconds'
}

```

Running

```

# -----
# PRE-PROCESSING
# -----
# Incorporate the parameters from DAKOTA into the template, writing ros.in
# Use the following line if SNL's APREPRO utility is used instead of DPrePro.
# ../aprepro -c '*' -q --nowarning ros.template ros.in

```

#Setting up variable to have path for working directories

```

optimization_dir=$(pwd)
cd ..
root_dir=$(pwd)

```

```

cd geometry/
geometry_dir=$(pwd)

```

```

cd row2/
bladerow_dir=$(pwd)

```

```

Threedgbginput_filename=$(find "$bladerow_dir" -name '3dbgbinput.*.dat' -exec basename {} \;)

```

```

Threedgbginput_template_filename=$(find "$bladerow_dir" -name
'3dbgbinput.*.dat.template' -exec\ basename {};)
spancontrolinputs_filename=$(find "$bladerow_dir" -name
'spancontrolinputs_NACA_2.dat' -exec\ basename {} \;)

```

```

spancontrolinputs_template_filename=(find"$bladerow_dir" -name 'spancontrolinputs_NACA_2.dat.template'
-exec basename ;)

```

```

cd ..
cd ..

cd CFD/
CFD_dir=$(pwd)

cd current_evaluation/
eval_dir=$(pwd)

cd ..
cd current_mesh/
current_mesh=$(pwd)

cd ..
cd scripts/
CFD_scripts_dir=$(pwd)

# find lowest static pressure directory
cd ../results/CFView_FinalResults/
CFView_dir=$(pwd)

# find .mf results directory
cd ..
mf_results_dir=$(pwd)

cd $optimization_dir

# update spancontrolinputs file
dprepro $1 $bladerow_dir/$spancontrolinputs_template_filename $bladerow_dir/$spancontrolinputs_filename
# update 3dbgbininput file
dprepro $1 $bladerow_dir$Threedgbginput_template_filename $bladerow_dir/$Threedgbginput_filename

# -----
# ANALYSIS
# -----

# "Running" function to check for each run ...
rm *.mf -f # -f for no error message if file doesn't exist
#rm Ps_lowest.dat -f # -f for no error message if file doesn't exist

# Running 3DBG/T-Blade3,
#-----
echo "$ Running T-Blade3 ... $normal"

```

```

cd $bladerow_dir

Running tblade3 $Threedgbginput_filename > log.dat &
pid=$!
wait $pid

# Creating and moving GeomTurbo file
#-----
if grep -i ' END ' log.dat; then
# Creation of the geomturb file:
echo " T-Blade3 run successful "
echo " creating geometry, Running GeomTurbo "
Running geomturb $Threedgbginput_filename 241 &
pid=$!
wait $pid

cp row.1.geomTurbo $geometry_dir
cp row.2.geomTurbo $geometry_dir
cp row.3.geomTurbo $geometry_dir

cd $geometry_dir

# added for rotor-stator geomturb combination
./combine_geomturb.sh # 2 1

# Change access permissions to read, write and execution before sending to Fan:
# chmod 700 row_wth_splitter.1.geomTurbo

# Copy the file need to run Fineturb on EEE linux box: -r for folders
echo " Geomturb file successfully created. "
geomturb_filename=$(find . -name 'blade_rows.geomTurbo' -exec basename {} \;)

mv $geomturb_filename $current_mesh/$geomturb_filename &
pid=$!
wait $pid

echo " The GeomTurbo file was moved to current_mesh folder "

# go to the folder to run fineturb script on Fan:
cd $CFD_scripts_dir

#Running python code for fine turb
#-----
#Running the create_mesh.py python file to create mesh
#-----
echo " Running Autogrid to create mesh files "
timeout 480 python create_mesh.py # if times out after 8min (480sec), then exit status is 124

```

```

if [ $? -ne 124 ]; then # if mesh doesn't time out; then
# continue like normal

echo " Geometry was successfully meshed "
echo " Searching for negative grid cells "
cd $current_mesh
#searching for negative cells
if ! grep -i -m 2 'Negative Cell Found' mesh.log; then
echo " No negative cells found "
cd $CFD_scripts_dir
#-----
#Running the fineturbo_analysis.py file to run CFD solver:
#-----
echo " Running fineturbo for CFD analysis "
python fineturbo_CFD_analysis.py
echo " fineturbo flow analysis has finished successfully "

cd $eval_dir/

mf_filename=$(find "$eval_dir" -name '*.mf' -exec basename {} \;) #getting filename for the .mf
chmod 700 $eval_dir/$mf_filename #changing its permission to be copied

cp $eval_dir/$mf_filename $optimization_dir/$mf_filename #copying .mf file to the main folder

echo " The .mf output file was successfully copied to optimization folder for grepping data "
#-----
#Running find_lowest_Ps.sh for finding lowest static pressure:
#-----
#cd $CFView_dir/
#./find_lowest_Ps.sh

else
echo " Negative cells detected in the grid. "
fi

else # mesh timed out, so skip fine turbo analysis
echo " Mesh Generation Timed Out "
fi

else
echo " Bad geometry 3DBG rejection. "
fi

cd $optimization_dir/
# -----

```

```

# POST-PROCESSING
# -----

# THE VALUES SHOULD BE IN THE SAME ORDER as in the Response section (input file).
#-----
# extract objective function value (Weight): (Remove second cut if units needed)
#if [ $indicator -eq 1 ]; then
if [ -f *.mf ]; then

adiabatic_eff=$(grep -i 'Isentropic_efficiency' $mf_filename | cut -d "y" -f2 | cut -d "(" -f1)

if (( $(bc <<<"$adiabatic_eff >= 1.0") )); then
    echo '1' > $2
    #echo '0.5' >> $2
    echo " Evaluation failed, set values out of limits  "

else

Vz_outlet=$(sed -n '50p' < $mf_filename | awk '{print $3}')
Vr_outlet=$(sed -n '48p' < $mf_filename | awk '{print $3}')
m_outlet=$(sed -n '84p' < $mf_filename | awk '{print $3}')

PT_in=$(sed -n '75p' < $mf_filename | awk '{print $2}')
TT_in=$(sed -n '76p' < $mf_filename | awk '{print $2}')
m_10=$(sed -n '165p' < $mf_filename | awk '{print $2}')

PT_out=$(sed -n '75p' < $mf_filename | awk '{print $3}')
TT_out=$(sed -n '76p' < $mf_filename | awk '{print $3}')
m_20=$(sed -n '216p' < $mf_filename | awk '{print $2}')
TS_out=$(sed -n '69p' < $mf_filename | awk '{print $3}')
PR=$(sed -n '92p' < $mf_filename | awk '{print $2}')

KE_total=$(echo "($TT_out-$TS_out)-($Vz_outlet*$Vz_outlet*0.5*0.0009940357853)
-($Vr_outlet*$Vr_outlet*0.5*0.0009940357853)" | bc -l)

Delta_Hideal=$(awk "BEGIN {print (((PT_out/PT_in) ^ 0.285714) -1)*$TT_in}")
Delta_Hactual=$(echo "($TT_out-$TT_in)" | bc -l)
eff=$(echo "scale=6;(($Delta_Hideal-$KE_total)/($Delta_Hactual))" | bc -l)
echo "$((sed 's/[E]/*10~/g' <<< $(echo 1-$eff)) |bc -l) > $2

printf "PT_in: %f\t" $PT_in >> track.txt

```

```

printf "PT_out: %f\t" $PT_out >> track.txt
printf "Vr_outlet: %f\t" $Vr_outlet >> track.txt
printf "Vz_outlet: %f\t" $Vz_outlet >> track.txt
printf "TT_out: %f\t" $TT_out >> track.txt
printf "TS_out: %f\t" $TS_out >> track.txt
printf "m_outlet: %f\t" $m_outlet >> track.txt
printf "KE_total: %f\t" $KE_total >> track.txt
printf "Delta_Hideal: %f\t" $Delta_Hideal >> track.txt
printf "Delta_Hactual: %f\t" $Delta_Hactual >> track.txt
printf "modified_eff: %f\t" $eff >> track.txt
printf "adiabatic_eff: %f\n" $adiabatic_eff >> track.txt
printf "PressureRatio: %f\n" $PR >> track.txt

# grep -i 'Absolute_Mass_flow' $mf_filename | cut -d "w" -f2 | cut -c 5-30 >> $2

##### grep 'Power ' $mf_filename | cut -d "r" -f2 | cut -d "[" -f1 >> $2
#grep -i 'Absolute_total_pressure_ratio' $mf_filename | cut -d "o" -f4 >> $2
# echo "((1 - $eff))" | bc >> $2
echo " objective function and constraints data grepped from .mf file"
rm $mf_filename
#rm $Ps_filename
fi
else
    echo '1' > $2
    #echo '0.5' >> $2
    #echo '-3000000' >> $2
    ##### echo '0' >> $2
    #echo '5' >> $2
    echo " Evaluation failed, set values out of limits "

fi

echo ".mf file deleted from the optimization folder"
#echo "lowest Ps file deleted from the optimization folder"
# extract the constraint value (Maximum stress): (Remove second cut if units needed)
# grep -i 'stall margin' run4_computation_1.mf | cut -d"=" -f2 | cut -d"[" -f1 >> $2
#fi
#-----
# -----
# transferring latest .mf file for a successful run with iteration number into results folder:

if [ -f $eval_dir/*.mf ]; then          # if .mf file exists in eval_dir, then

    mf_filename=$(find "$eval_dir" -name '*.mf' -exec basename {} \;)          # get mf_filename
    chmod 700 $eval_dir/$mf_filename #changing its permission to be copied

    dakota_dat=$(find "$optimization_dir" -name 'dakota_CFD.dat' -exec basename {} \;)          #

```

```
    number_of_iteration=$(wc -l < "$optimization_dir/$dakota_dat")
    echo "number of iteration = $number_of_iteration"

cp $eval_dir/$mf_filename $mf_results_dir/"CFD_optimized.$number_of_iteration.mf"
cd $eval_dir
rm $mf_filename      # remove mf file from eval_dir

else # last run was unsuccessful
echo " Last run failed, so .mf file wasn't copied to results folder. "

fi
```

A4: Blade to blade contour plots 10% span for Case:EXPL-AXI-3IRO-7-15, 000,
NLH- 3 Harmonics

A5: Blade to blade contour plots 90% span for Case:EXPL-AXI-3IRO-7-15, 000,
NLH- 3 Harmonics

A6: Graphical explanation of difference between $\overline{\overline{(V_z)^2}}$ and $(\overline{\overline{V_z}})^2$

Quantity V_z , axial velocity is taken for explanation

$$V_z = \overline{\overline{V_z}} + V_z''' \quad (8.1)$$

Where $\overline{\overline{V_z}}$ is the average and V_z''' is the delta change from average of V_z , which is also represented in Fig. 8.1. The blue line in Fig 8.1 is the V_z profile and the red line represents $\overline{\overline{V_z}}$.

$$\overline{\overline{V_z}} = \int (V_z dy) \quad (8.2)$$

Figure 8.1: Plot of Velocity profile vs y

Figure 8.2: Plot of Velocity squared profile vs y

$$\overline{\overline{(V_z)^2}} = \int (V_z^2 dy) \quad (8.3)$$

$(\overline{\overline{V_z}})^2$ is obtained by squaring Eq. 8.2. The difference between $\overline{\overline{(V_z)^2}}$ (orange line in Fig.8.2) and $(\overline{\overline{V_z}})^2$ (red line in Fig. 8.2) represents the lost axial KE $\overline{\overline{(V_z''')^2}}$ (Eq. 4.13 in Chapter 4).

$$\overline{\overline{(V_z''')^2}} = \overline{\overline{(V_z)^2}} - (\overline{\overline{V_z}})^2 \quad (8.4)$$