Optimization of a Bi-dimensional Intake for a Supersonic Aircraft using a Genetic Algorithm

(Versão Final após Defesa)

Diogo Rafael Ribeiro Pinto Simão

Dissertação para obtenção do Grau de Mestre em
Engenharia Aeronáutica
(mestrado integrado)

Orientador: Prof. Doutor Francisco Miguel Ribeiro Proença Brójo

Janeiro de 2023

Folha em branco

Declaração de Integridade

Eu, Diogo Rafael Ribeiro Pinto Simão, que abaixo assino, estudante com o número de inscrição 33620 de/o Engenharia Aeronáutica da Faculdade das Engenharias, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 5 / 1 / 2023

Diogo Rafael Ribeiro Pinto Simão

(assinatura conforme Cartão de Cidadão ou preferencialmente assinatura digital no documento original se naquele mesmo formato)

Folha em branco

Acknowledgements

First, I would like to thank my parents for supporting me throughout the years.

I would also like to thank my professor, Francisco Brójo, for helping whenever it was needed.

I would like to thank Matilde as well, who has encouraged me these years to keep going.

Folha em branco

Resumo

Desde as primeiras aeronaves supersónicas, as admissões foram objeto de muitos estudos devido à sua importância. Desde a sua posição nas asas, à sua passagem para as raízes destas, nariz e finalmente a montagem na sua fuselagem, foram necessárias décadas no estudo das admissões e resolução das suas restrições. Estas restrições continuam a estar presentes no projeto, quer sejam em relação ao espaço, peso, arrasto, centro de gravidade, visibilidade no radar, entre outras.

Embora admissões com um número infinito de choques fracos sejam teoricamente preferíveis, na prática, uma admissão com um número pequeno de choques é superior, pois é mais leve, devido a menores dimensões, e traz consigo um melhor desempenho em condições off-design.

No presente trabalho é projetada uma admissão bi-dimensional com compressão mista, tendo os requisitos de motor e outros constrangimentos do avião Panavia Tornado em mente.

Os resultados obtidos são corroborados com resultados obtidos em artigos relacionados e posteriormente comparados com os resultados de uma otimização multiobjectivos para minimizar o comprimento e maximizar a recuperação de pressão total. Deste modo, é possível alcançar um desempenho alto enquanto se mantém o peso baixo. Estes objetivos são alcançados ao usar um intervalo de velocidade próximo da operação do Tornado. De acordo com o esperado, uma admissão com menores dimensões tem uma recuperação total de pressão inferior em condições on-design. No entanto, também é sinónimo de baixo peso estrutural. Outra ligação encontrada é o aumento de recuperação de pressão total em admissões projetadas para menores velocidades, além de um comprimento inferior. Este resultado, além da redução na importância da velocidade máxima da aeronave na maioria dos perfis de missão, sugere que uma aeronave com uma velocidade máxima inferior é vantajosa, pois a sua agilidade e desempenho será superior.

Palavras-chave

Admissão supersónica; otimização; design; projeto; bidimensional;2-D; conduta;choques oblíquos; choques normais

Abstract

Since the beginning of jet flight, the intake has been subject to many studies due to its importance. From its position on the wings, in the wing root, nose and being mounted on the fuselage, many decades were spent on the research and solution of constraints. These are still present in the form of space and weight saving, lower drag, centre of gravity, lower radar visibility, etc.

Although theoretically intakes with an infinite number of weak shocks are better, these fail to produce better results. An intake with small number of shocks is preferred as it is smaller in size, thus saving weight, and has better off-design performance.

In the present work, a bi-dimensional mixed compression intake is designed having the Panavia's Tornado engine requirements in mind and the aircrafts performance values. The designed intake results were in line with other studies.

A comparison is made between the parametrical result and the results of a genetic optimization algorithm to minimize length and maximize total pressure recovery (thus saving weight while having high performance) by using a variation on Mach numbers around the designated speed for the Tornado, while using a mixed compression intake. Accordingly, a longer intake had better performance, but a shorter intake comes with lower weight. With lower speeds, a higher total pressure recovery and lower length is reached. This, along with the loss of importance of high speeds for most mission profiles, suggests that an aircraft with a lower top speed is advantageous as its weight lowers and performance increases.

Keywords

Supersonic admission;optimization;design;project;bi-dimensional;2-D; duct; oblique shocks;normal shocks

Folha em branco

Index

List of figures ... xiii
List of Tables ... xv
Nomenclature ... xvii
List of Acronyms ... xix
1. Introduction ... 1
1.1 Motivation ... 1
1.2 Objectives ... 1
1.3 Thesis Structure ... 1
2. Literature Review ... 3
2.1 Supersonic intake types ... 5
2.1.1 Bi-dimensional Admissions ... 5
2.1.2 Axisymmetric Admissions ... 5
2.1.3 Three-dimensional Admissions ... 6
2.2 Intake Compression types ... 7
2.2.1 External Compression Intakes ... 7
2.2.2 Internal Compression Intakes ... 7
2.2.3 Mixed Compression Intakes ... 7
2.3 Evolutionary Optimization ... 8
2.3.1 Concept of evolution ... 8
2.3.2 Genetic evolution algorithm ... 8
2.3.3 Chromosome representation ... 9
2.3.4 Initial population ... 9
2.3.5 Fitness function ... 9
2.3.6 Selection Operators ... 10
2.3.7 Reproduction operators ... 11
2.3.8 Stoppage Conditions ... 11
2.3.9 Genetic Algorithm ... 12
2.3.10 Multiobjective optimization problems ... 12
2.3.11 Pareto's optimization method ... 13
3. Parametric Study ... 14
3.1 Engine choice-RB199 Mk 103 ... 14
3.1.1 RB199 Mk 103 ... 14
3.1.2 Technical data on the RB199 Mk 103 ... 15
3.2 Intake dimensioning process flowchart ... 16
3.3 Admission dimensioning ... 17
3.3.1 Free-flow air properties ... 17
3.3.2 Bi-dimensional admission dimensioning script/code using MATLAB ... 17
3.3.3 Total pressure recovery ... 19
3.4 4 Parametric results ... 21
4. Optimization Study ... 24
4.1 Configuration of the multiobjective optimization with speeds ranging from Mach 1.8 to Mach 2.2 ... 24
4.2 Optimization Results with speeds ranging from Mach 1,8 to Mach 2,2 ... 26
4.3 Configuration of the multiobjective optimization for Mach 2.2 ... 29

4.4 Optimization Results for Mach 2.2 30
5. Final Considerations 34
5.1 Conclusion 34
5.2 Future studies 35
6. Bibliography 36
6.1 References 36
6.2 Links used 37
7. Appendix 38

List of figures

Figure 1	Panavia Tornado (left) [4] and F-15 Eagle (right) [5]
Figure 2	Mig-21 (left) [7] and Mirage 2000 [8]
Figure 3	J-10 (left) [10] and F-35 (right) [11]
Figure 4	Types of Compression [12]
Figure 5	RB199 Mk103 engine [13]
Flowchart 1	Engine Intake design procedure
Figure 6	Normalized Inlet Plot with 2x2 ramp arrangement using MATLAB
Figure 7	Visual Representation of the normalized intake with its formed shocks using CATIA V5
Figure 8	Pareto's Front for the multiobjective optimization with Mach ranging between 1.8 and 2.2 using MATLAB
Figure 9	Dispersion Graph of the results of the multiobjective optimization
Figure 10	Filtered Results of Multiobjective Optimization
Figure 11	Pareto's Front for the multiobjective optimization for Mach 2.2 using MATLAB
Figure 12	Dispersion graph of results of the multiobjective optimization for Mach 2.2
Figure 13	Filtered results of multiobjective optimization for Mach 2.2

Folha em branco

List of Tables

Table 1	Comparison of values with Omni F. et al for validation of the script
Table 2	Station Points Coordinates of Normalized Intake
Table 3	Airflow and Shock Properties along the intake
Table 4	MATLAB configuration of the multiobjective optimization
Table 5	Genetic Algorithm's results after removal of impossibilities
Table 6	Genetic Algorithm's results complying with military requirements
Table 7	MATLAB configuration of the multiobjective optimization for Mach 2.2
Table 8	Genetic Algorithm's Optimization Results for Mach 2.2 after removal of impossibilities
Table 9	Genetic Algorithm's results complying with military requirements for Mach 2.2

Folha em branco

Nomenclature

h	Altitude	m
T_i	Temperature at a stage	K
p_i	Pressure at a stage	Pa
g_0	Gravitational Acceleration	m/s^2
R	Air Gas Constant	$J/_{kg\,K}$
ρ_i	Density at a stage	$kg/_{m^3}$
a	Speed of Sound	$m/_s$
M_i	Mach number at a stage	
p_{t_i}	Stagnation Pressure at a stage	Pa
θ	Flow Deviation Angle	Deg
β	Oblique Shock Angle	Deg
η_{spec}	Minimum Pressure Recovery at a given Mach number (MIL-E-5008B)	
PR_i	Pressure Recovery at a stage	
PR_{sub}	Pressure Recovery of subsonic duct	
TPR	Total Pressure Recovery	
μ	Parents	
λ	Offspring	
f_k	Objective	
x_i	Decision Vector	
$f^*(x)$	Pareto's Vector	
$P\mathcal{F}$	Paretos Front	

Folha em branco

List of Acronyms

DSI	Diverterless Supersonic Intake
JSF	Joint Strike Fighter
ECI	External Compression Intake
ICI	Internal Compression Intake
MCI	Mixed Compression Intake
IDS	Interdictor/Strike
ECR	Electronic combat/Reconnaisance
ADV	Air Defence Variant
SFC	Specific Fuel Consumption
TPR	Total Pressure Recovery

Folha em branco

Chapter 1

Introduction

Motivation

With the rise of supersonic aircraft and the pressure to increase aircraft performance due to ongoing conflicts as well as the increasing projects of civilian aircraft with supersonic capabilities, an increase of efficiency in these aircraft is required without compromising performance. To address this issue, intakes are an important component worth looking at in order to achieve these objectives. The best way to increase efficiency on these is to increase the total pressure recovery obtained and to minimize drag and weight. Of course, it is hard to design an intake that produces all these results without having some sort of compromise and that usually comes in the form of a decrease in off-design performance. This thesis focuses on producing an algorithm that can design an intake that performs well on-design and off-design, depending on the constraints. For that a multiobjective optimization was introduced in order to find intakes which could be more suitable. A comparison will be drawn between the two results, while also explaining the differences between the two.

Objectives

The main goal of this work is to create a MATLAB script which can design and optimize a bi-dimensional fixed geometry admission for a supersonic capable engine from a set of requirements and compare the two. The resulting algorithm will, therefore, ease future designing processes.

Thesis Structure

This thesis will be covering different subjects in order to better understand the subject in question; therefore, it is organized into the following manner:

- The first chapter will be an explanation of the structure of the thesis, objectives and the motivations behind the research;
- In the second chapter the state of the art will be covered which will review recent or past studies that enhance the knowledge needed for a better understanding of

intake technology and functioning, and a explanation of all variants of intakes and compression types.

- The third chapter covers the parametrical study, which concerns the engine choice and information about its application and technical data, an explanation of the dimensioning algorithm along with its study and results.
- The multiobjective optimization will be the subject of the fourth chapter with an explanation of the theory behind it, an explanation of the genetic algorithm, and the usage of it to produce an optimized intake within a certain range of speed;
- The last chapter (chapter 6) is the conclusion of this thesis with suggestions of possible future works in the study of this area.

Chapter 2

Literature Review

The first time a jet engine was designed and built, it was intended to be placed in the nose of the aircraft or be wing mounted due to having a lower amount of total pressure loss caused by the friction on the wet surface by the flow which enters the intake. For an intake to be efficient it must achieve a high total pressure recovery while minimizing drag, pressure distortion at the first engine component face, weight, complexity, and cost, all this while the constraints of the project are considered. These can either be, space saving constraints, dimensioning, radar susceptibility, flight envelope, ground clearance, etc.

To address the occupied space issue of the intakes two solutions were developed, one was the scoop intake and the other the submerged flush mounted intake: the scoop intake offers good pressure recovery but suffers from a drag penalty while the flush mounted intake had lower pressure recovery but minimized drag. Using the flush mounted intake, it was observed that the boundary layer losses were significantly higher in the ramp compared against the boundary layer losses on the lateral wall. Further investigation by Seddon and Goldsmith [1] confirmed that an adverse gradient behaviour of the boundary layer was responsible for the biggest issues when designing intakes.

Another potential solution that stopped being used was the wing root mounted intake. Sóbester, A. [2] explains that these suffered from high spillage drag when it was used with wings with high angles of sweep, this was due to an excess flow that spilled from the rearward most area of the admission caused by the wing sweep. Wing sweep also caused losses of pressure at low speeds, because of the higher amounts of air entering the rearward most section. Additionally, lower pressure recovery values and uniformity were achieved, due to shock losses and overspeed caused by having most of the airflow turning done by the more aft lip surfaces of the intake.

When approaching supersonic speeds other types of intakes were developed, these being the axisymmetric and bi-dimensional intakes, which used oblique shocks to slow the airflow before entering the engine. These intakes will be explained in more detail in the next section.

While an infinite number of small oblique shocks is theoretically better, it is preferable to have a small number of stronger shocks in order to obtain a simpler and more practical design process. Drag also comes into account in the decision-making process of the shock system structure. If an intake has a high capture to throat area ratio (optimal for low speeds), the spillage of the excess airflow will separate and contribute to the overall drag. If the capture area is sized for high speeds, a lower performance at low speeds will be observed, due to insufficient air feeding the engine. This problem can be fixed in two ways: increasing the capture area by means of blow-in doors, for example, or by having a variable intake geometry which enables a higher capture are at lower speeds and smaller capture area at higher speeds.

The variable intake geometry can also be used to adapt to variations of angles of attack. Lip position changing, shock positioning and variation of the boundary layer bleeding mechanism are ways that one can increase the performance of the intake, however, it can significantly increase costs and complexity, so much so that it can become unviable.

When talking of drag, it usually means operational drag, but for safety reasons, drag must also be calculated for other situations, for when the engine isn't functioning at all. This means that while the engine might have an optimum operational drag to pressure recovery ratio when the engine is running, it might also perform badly in a glide/lift to drag ratio, or create excessive yaw when it is not operating, these issues could exist in a high enough level that the intake must be scrapped.

The ducts are also an important component in the study of admissions. In uniform, high pressure recovery cases, the ducts should be maintained as close as possible to a straight line, therefore reducing flow separation. However, for some designs this may not be the case.

One such case is discretion. By having a straight duct, the engine compressor face will have a bigger radar signature thus harming its stealth capabilities, making the aircraft an easier target. One way of reducing the intake's visibility is to coat it in ferrites in order to absorb the waves, this will in turn increase the weight of the intake. If no steps towards discretion are taken, the electromagnetic waves of the radar will enter the air intake, and, after reflecting inside it multiple times, be re-transmitted out of the entrance plane in multiple directions, and finally, electric charges formed on surface due to the interaction with the radar waves can travel through the skin of the structure and concentrate on the leading edge (point effect) and transmit from there, but the cowl can be designed with broken lines and electrically insulated elements to minimize this issue. Another valuable way of reducing the radar signature of the intake is adjusting the entrance cross-section or gridding it, as with smaller openings comes a decrease in the re-transmitted waves.

Another option would be placing blades in the diffuser duct and/or curving the diffuser so that the radar doesn't identify the engine of the aircraft.

Supersonic intake types

Bi-dimensional Admissions

Used in aircraft such as the McDonnell Douglas F-15 or the Panavia Tornado (shown in Fig. 1), 2D admissions are generally used above Mach 1.5 (same with axisymmetric) and can be annular or partially rolled and rectangular, as mentioned by Laruelle, G. [3]. These intakes are normally of mixed compression with an external compression ramp and a cowl, followed by an internal boundary layer bleed. This intake normally operates with variable geometry normally being hydraulically actuated ramps which control the compression and/or bleed. In comparison to an axisymmetric intake, the rectangular intake has less distortion at high angles of attack, better geometry variation possibilities and a lower risk of surge in asymmetric flow conditions.

Figure 1 Panavia Tornado (left) [4] and F-15 Eagle (right) [5]

Axisymmetric Admissions

Examples of axisymmetric admissions can be found on the Mig-21 or even the Mirage (as in Fig. 2), these intakes are similar to 2D intakes in function and share most of its elements with it, with the difference that these intakes revolve around an axis. Laruelle, G. [3] also points that these intakes can be either split in half (Mirage), partial (F111) or full (Mig-21). For a variable geometry, a translating spike (seen on the Sr-71 Blackbird) as reported by Anderson, J. T. [6] or collapsing conical bullet is the norm. This type of intake is normally much more efficient structure wise, which in turn saves weight, when compared to a bi-dimensional intake.

Figure 2 Mig-21 (left) [7] and Mirage 2000 (right) [8]

Three-dimensional Admissions

Three dimensional admissions vary from the latter intakes in their functioning. Examples of these intakes may be found in the Lockheed Martin F-35, or even the Chinese J-10 (Fig.3). These intakes make use of a bump which prevents the boundary layer from entering the intake. The bump also acts as an external compression ramp, being complemented by a cowl as in the other cases above. These intakes in the cases of the aircraft in Fig.3 are called Diverterless Supersonic Intakes, or DSI for short. The main advantages of these are in weight and complexity as the Joint Strike Fighter Program (JSF) [9] declares, as they have no moving parts in comparison to other intake variants, and in stealth since the "bump" hides the engine face as well as not having a diverter from which radar can reflect of between it and the aircrafts skin.

Figure 3 J-10 (left) [10] and F-35 (right) [11]

Intake Compression types

All above-described intakes can be designed to operate using three types of compression, external, internal, or mixed compression, these being achieved by the use of ramps at a pre-determined angle, which will in turn form the intended oblique shocks. These are assessed in the following sub-section, with an explanatory figure at the end (Fig.4).

External Compression Intakes

An external compression intake functions by forming multiple oblique shocks at the exterior through the usage of a ramp and a final normal shock at the lip. The advantages of this intake are dissipated upon reaching Mach 2,5, as the cowl drag increases. This intake solution is simple, yet a major drawback of it is the length of the ramp which in turn increases the structural weight of the intake. This makes it heavier than an MCI (Mixed Compression Intake). An example is found in Fig.4 (top right).

Internal Compression Intakes

The internal compression system intake also makes use of oblique shocks and in the end a normal shock, the first of these is formed at the lip, while the normal shock forms inside the duct of the intake, after the last oblique shock, as seen in Fig. 4 (top left).

Mixed Compression Intakes

Mixed compression intakes are a mixture between the above two types, having part of its compression made on the external ramps and the rest by internal ramps, followed then by a normal shock (bottom of Fig.4). This type of intake is the most frequently used because it allows for a smaller intake length, thus, saving weight. This type of admission is the most used for supersonic airplanes and it will be covered on this dissertation.

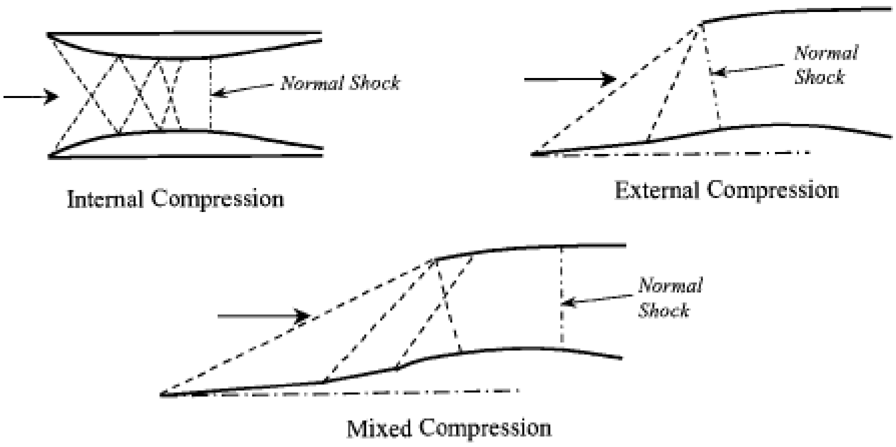


Figure 4 Types of Compression [12]

Evolutionary Optimization

Concept of evolution

Evolution is a process in which an organism adapts to changing competition and/or environment through reproduction, mutation, symbiosis, survival and competition, thus changing the populations characteristics of the same organism, improving the chances of survival.
Computational evolution follows the basis of biological evolution by using evolutionary process algorithms in order to solve systems.

Genetic evolution algorithm

The evolutionary algorithm iterates towards an optimal solution through a stochastic search, and it is altered by its components:

- Representation of the problems solution;
- An objective function evaluating function (fitness) or the individual's capability of survival;
- Establishment of initial population;
- Selection operators;
- Reproduction operators.

Considering an initial instant (t=0s) with an initial population, the algorithm's solving process is iterative and ends when it complies with a finishing condition. An iteration can be thought of as a generation.

Chromosome representation

Chromosomes are a structure comprised of long chains of DNA molecules which can be thought of as information, these set the characteristics of an individual. In evolutionary computation, each individual, comprised of the set of its chromosomes, is a potential optimized solution of the problem, and its chromosomes are the variables that need optimization in order to comply with the objective function.

In the designing phase of the evolutionary algorithm, an important step is to find an appropriate representation of the chromosomes (possible solutions). How we represent data and the chromosomes determines the efficiency and speed in finding the optimized solutions for the system.

Initial population

Since evolutionary algorithms are stochastic (they represent random events with random variables), each one of them has a population which is candidate to the optimal solution. The first step in an evolutionary algorithm is to find an initial population that represents the intended variables interval; therefore, a random value of the dominion of variables must be attributed to each chromosome gene.

To obtain more diverse results, a bigger initial population size is required. This increase in size however comes at a cost of higher computation time for each iteration, as there is more complexity in searching for a solution. Having a smaller initial population is also possible, which lowers the amount of time taken between each iteration and its complexity, but more generations will be needed to find an optimal solution. One way to offset a smaller initial population is by increasing the mutation factor of the individuals.

Fitness function

The fitness function evaluates the survival capacity of an individual to survive according to the evolutionary algorithm, by quantifying how optimal the solution is.

The fitness function is the objective function which describes the problem's optimization, but it isn't quite correct to point that the chromosome corresponds to the hoped-for representation of the objective function.

Usually, the fitness function can be used to evaluate solutions, represented by chromosomes, which are in turn evaluated by the objective function. In cases where the fitness function isn't defined explicitly, a relative fitness function is defined which quantifies the performance and compares with the other individuals of the population. According to the optimization problem type, the fitness function varies as such:

- No constraints, the fitness function is defined as the objective function;
- With constraints, some evolutionary algorithms possess a fitness function with two objectives. One is the original objective, and the other is a constraint in the form of a penalizing function;
- Multiobjective, can be solved by an approximation by ponderation, where the fitness function is the pondered sum of all sub-objectives. These can also be solved using a Pareto front optimization algorithm;
- Noisy and dynamic problems, where the value of the solution function change alongside time.

So, it is possible to conclude that the operators that solve the algorithm use it to evaluate the chromosomes value. In this way, the fittest individuals are chosen by the crossover operators while the less fit are chosen by the mutation operators.

Selection Operators

The selection operator's main role is to choose the best solutions and can be achieved using two methods:

- Choosing a new population: at the end of each generation a new population of potential solutions is chosen to serve as the next generation. The new population must be comprised of progenitors and offspring. This operator must ensure that the individuals with the best characteristics survive for the next generation;
- Reproduction: the offspring is produced by the application of crossover and/or mutation factors. The crossover factors must select the individuals with the best characteristics to ensure the offspring contains the genetic material of the fittest, while mutation must be used on the less fit individuals in order to introduce a random change that possibly improves the individual's characteristics.

And the most common operators are the following:

- Selective pressure;
- Random selection;
- Proportional selection;
- Tournament selection;

- Ranking based selection;
- Boltzmann method selection;
- (μ^{+}, λ) method selection;
- Elitism;
- Hall of fame.

Reproduction operators

The reproduction operators are a process of offspring production from selected progenitors by applying crossover or mutation criteria, and these are:

- Crossover, which is creates one or more individuals from the combination of randomly selected genetic material from two or more progenitors (usually between less fit and a fitter individuals).
- Mutation, which randomly alters to the information present in the genes of the chromosomes with the goal of introducing new genetic material in the population in order to augment genetic diversity. Only the individuals with the lowest value of the objective function should become subject to mutation.

Reproduction can be also used by substitution, where offspring with a higher objective function value than the progenitors replace them entirely.

Stoppage Conditions

A stoppage condition must be given in which when met, communicates with the evolution operators of the algorithm to stop producing new generations, so computational time and resources are saved. The simplest way to introduce a stoppage condition is to limit the number of generations which the algorithm can produce. Nonetheless, other convergence criteria can be used when any type of alteration to the genotype or phenotype of the population is unaltered:

- When no alterations are recorded in the optimization of the objective function through consecutive generations;
- When there are no changes in the population;
- When an acceptable solution is found;
- When the deviation of the solution associated with the objective function is approximately zero.

Genetic Algorithm

A genetic algorithm is better used when classical optimization can't be applied, specifically where the objective function is discontinuous, non-differentiable, stochastic, or non-linear, and functions the following way:

1. An initial population is generated according to the selected variables;
2. Within the population some individuals are chosen and attributed a ranking in relation to the objective function;
3. Individuals with a higher score are selected according to the objective function to produce offspring;
4. The next generation will be composed by lower scored individuals and by the offspring of higher scored progenitors. The offspring is constituted by a mutation factor in relation to the progenitors and by combining the characteristics of the progenitors determined by a crossover factor;
5. The algorithm ends according to selected criteria, can be a limit on the number of iterations or any other stoppage condition.

Multiobjective optimization problems

Before beginning the multiobjective optimization configuration it is necessary to define the sample size and Pareto's front. On single objective optimization it is possible to find a local optimal point which corresponds to the problem's global solution. However, for optimization problems with two or more objectives simultaneously, finding the optimal solution can be an issue, since sometimes the objectives are conflicting, where one objective deteriorates the other. Sometimes it is possible to find an intermediate point in the optimization algorithm where a balance between the minimization of one objective and the maximization of the other exists.
For example, maximizing the Mach number possible is contradictory to maximizing the total pressure recovery. This equilibrium can be defined for a multiobjective problem as a set of unnominated solutions or an optimal Pareto set. For this reason, the purpose of a multiobjective optimization is to obtain a set of viable solutions which constitute multiple results which can be considered optimal.

There are two methods to solve optimization problems using genetic algorithms:

- Pondered aggregation method, where the aggregated objective is defined as the pondered sum of all sub-objectives;

- Pareto's optimization method, in which a dominance relation is applied to obtain an approximate Pareto's front.

Pareto's optimization method

To understand the Pareto's optimization method, it is necessary to understand some concepts such as dominance, Pareto's optimization, Pareto's front and others.

As Tavares, G. [19] stated, a decision vector x_1 is dominant over another decision vector x_2, if, and only if:

- x_1 isn't worst than x_2 in all its objectives, for example $f_k(x_1) \leq f_k(x_2), \forall k = 1, \dots, n_k$;
- x_1 is strictly better than x_2 in at least one objective, for example $\exists k = 1, \dots, n_k : f_k(x_1) < f_k(x_2)$.

Pareto's optimization can be defined as: a decision vector, $x^* \in \mathcal{F}$ is considered optimized by Pareto's method if a decision vector which dominates it doesn't exist, $x \neq x^* \in \mathcal{F}$.

A Pareto's vector, $f^*(x)$ is optimal by the Pareto's method if x is itself optimized by Pareto. Therefore, for multiobjective problems it is necessary to define a set which contains all the Pareto's optimal solution's: $P^* = \{x^* \in \mathcal{F} | \nexists x \in \mathcal{F}: x < x^*\}$.

As such, the Pareto's front is given as a set between an objective vector, and the set of optimal Pareto's solutions, P^* where the optimal Pareto's front, $P\boldsymbol{\mathcal{F}}^*$ is defined as: $P\boldsymbol{\mathcal{F}}^* = \{f = (f_1(x^*), f_2(x^*), \dots, f_k(x^*)) | x^* \in P\}$. This set represents only the solutions which tend to be the ideal solutions considering each objective.

Chapter 3

Parametric Study

Engine choice-RB199 Mk 103

RB199 Mk 103

To design an air intake an engine must be provided to be coupled with. As requirements such as altitude, engine width and minimum airflow change between engines and respective aircraft. As such, the RB199 Mk 103 engine used in the Panavia Tornado GR.1/ GR.1A/ GR. 1B/ GR.4(IDS) will be the subject of this work as the engine data required for this work is available.

The Panavia Tornado is a family of variable sweep wing multirole twin-engined combat aircraft that was produced between 1979 and 1998. It was jointly developed and manufactured between the U.K., West Germany and Italy. Three primary variations were made, the Tornado IDS fighter-bomber, the Tornado ECR capable of suppressing enemy air defences and the Tornado ADV interceptor aircraft. This aircraft has a service ceiling of 15,240m of altitude, is capable of a speed of 2400 km/h at 9000m, corresponding to Mach 2.2.

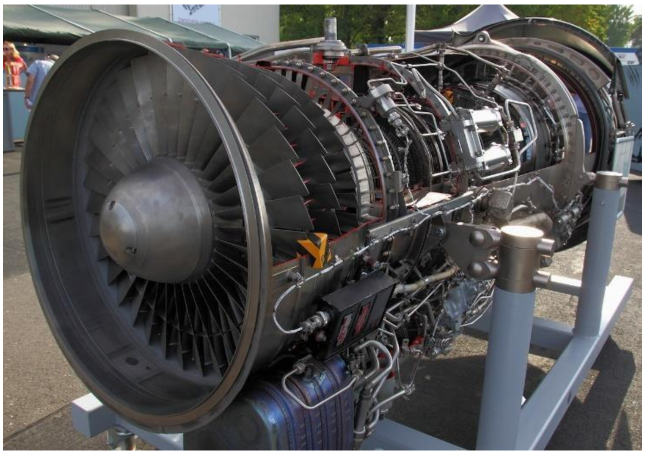


Figure 5 RB199 Mk103 engine [13]

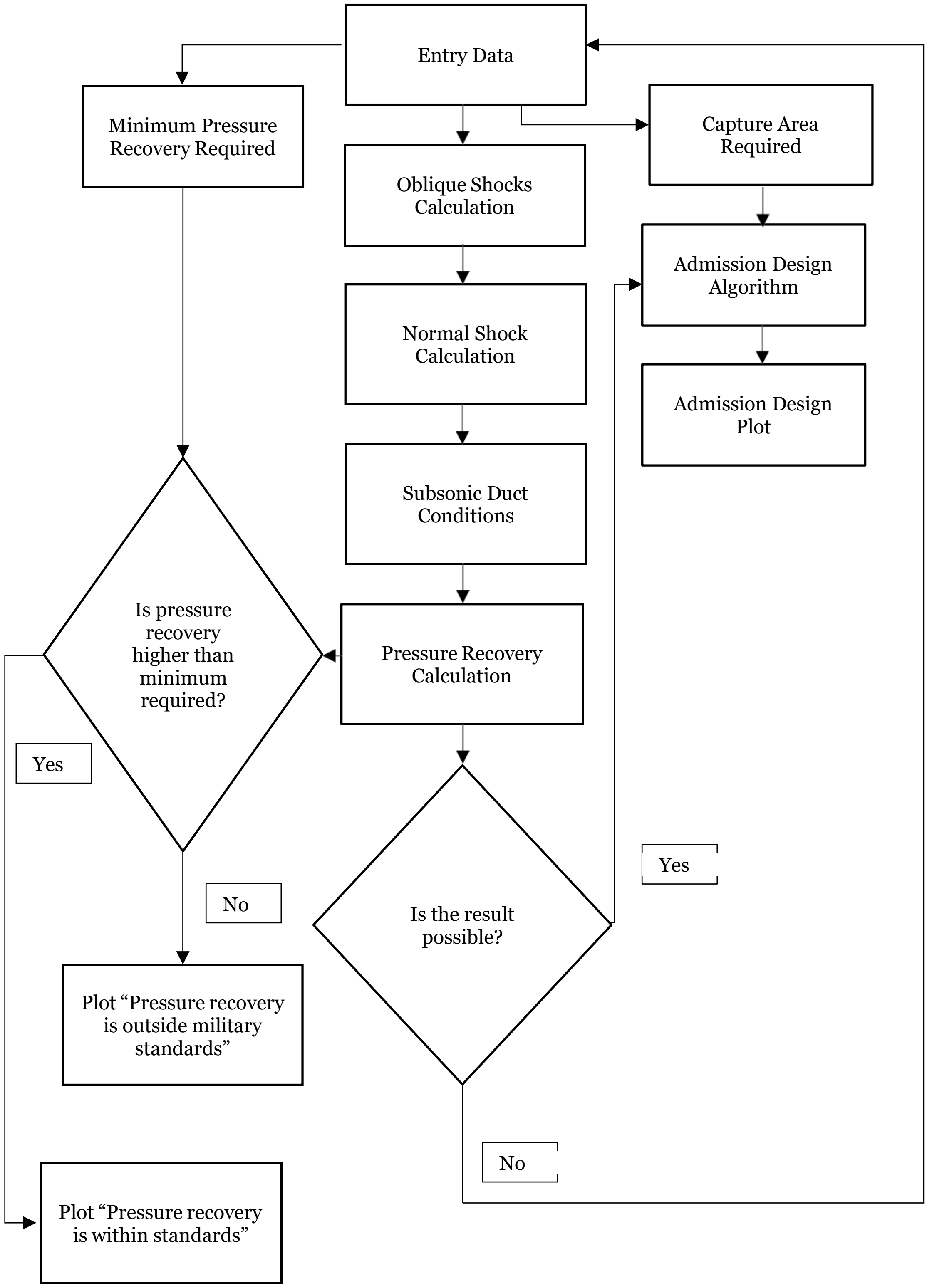
Technical data on the RB199 Mk 103

The technical data on the engine is as follows:

- Engine type: Turbofan with 3 shafts and low bypass ratio;
- Dry thrust: 40.5 kN;
- Wet thrust: 71.2 kN;
- Dry SFC: 0.6 1/h;
- Wet SFC: 2.2 1/h;
- Static Airflow: 71,21 kg/s;
- Overall pressure ratio: 23,5:1;
- Bypass ratio: 1,25:1;
- Number of spools: 3;
- Number of fan stages: 3;
- Number of low-pressure compressor stages: 3;
- Number of high-pressure compressor stages: 6;
- Number of high-pressure turbine stages: 1;
- Number of intermediate-pressure turbine stages: 1;
- Number of low-pressure turbine stages: 2;
- Engine length: 3251,2 mm (3,251m);
- Engine diameter: 720mm (0,7m);
- Engine dry weight: 955,72 kg;
- Turbine inlet temperature: 1600 K;

Intake dimensioning process flowchart

The intake design follows the procedure presented in Flowchart 1:



Flowchart 1 – Engine intake design procedure.

Admission dimensioning

Free-flow air properties

According to the aircraft's altitude and location different air conditions might arise, therefore these must be calculated. In this case the mean values according to altitude will be used.

For:

$$h > 0\,m \ (1)$$

And using the constants:

$$T_0 = 288.15\,K \ (2)$$
$$p_0 = 101325\,Pa \ (3)$$
$$\lambda = -0.0065\,K/m \ (4)$$
$$g_0 = 9.81\,m/s \ (5)$$
$$R = 287\,J/kgK \ (6)$$

One can use the Standard International Atmosphere equations up to 11000m of altitude:
:

$$T = T_0 + \lambda h \ (7)$$
$$\frac{p}{p_0} = \left(\frac{T}{T_0}\right)^{\frac{-g_0}{\lambda R}} \ (8)$$
$$\rho = \frac{p}{RT} \ (9)$$

Bi-dimensional admission dimensioning script/code using MATLAB

Bi-dimensional admissions work based on compressible flow theories, through the usage of shocks. These calculations mostly use the Mach number as this is relative to the speed of sound in the air. The Mach number can be expressed as:

$$M = \frac{v}{a} \ (10)$$

In which the speed of sound, a, is represented by:

$$a = \sqrt{\gamma R T} \ (11)$$

Shocks are formed when the fluid (in this case, air) molecules can no longer transmit information in a wave form upstream because it has exceeded the speed of sound, causing a sudden deceleration of the incoming flow. A shock that is normal to the free-flow can change pressure, temperature, density and Mach number of the fluid. The higher the free-flow speed, the stronger the shock and to work around this, as stronger shocks are bad for pressure recovery, since they are highly irreversible processes, it is possible to produce oblique shocks which are weaker and don't affect the pressure recovery as much, although they do change the flow direction. Shocks can be represented by the following expressions:

- Normal shocks

$$\frac{p_1}{p_0} = \frac{2\gamma M^2-(\gamma-1)}{\gamma+1} \quad (12)\ [14]$$

$$\frac{p_{t1}}{p_{t0}} = \left(\frac{(\gamma+1)M^2}{(\gamma-1)M^2+2}\right)^{\frac{\gamma}{\gamma-1}} \left(\frac{(\gamma+1)}{2\gamma M^2-(\gamma-1)}\right)^{\frac{1}{\gamma-1}} \quad (13)\ [14]$$

$$\frac{T_1}{T_0} = \frac{(2\gamma M^2-(\gamma-1))((\gamma-1)M^2+2)}{(\gamma+1)^2 M^2} \quad (14)\ [14]$$

$$\frac{\rho_1}{\rho_0} = \frac{(\gamma+1)M^2}{(\gamma-1)M^2+2} \quad (15)\ [14]$$

$$M_1^2 = \frac{(\gamma-1)M^2+2}{2\gamma M^2-(\gamma-1)} \quad (16)\ [14]$$

- Oblique shocks

$$\tan\theta = 2\cot\beta \left[\frac{M_1^2 \sin^2\beta - 1}{M_1^2(\gamma+\cos 2\beta)+2}\right] \quad (17)\ [14]$$

From the formulas above the following can be deduced:

$$M_1^2 = \frac{(\gamma+1)^2 M_0^4 \sin^2\beta_1 - 4(M_0^2 \sin^2\beta_1 - 1)(\gamma M_0^2 \sin^2\beta_1 + 1)}{[2\gamma M_0^2 \sin^2\beta_1 - (\gamma+1)][(\gamma-1)M_0^2 \sin^2\beta_1 + 2]} \quad (18)\ [16]$$

$$\frac{p_1}{p_0} = \frac{2\gamma(M\sin\beta)^2-(\gamma-1)}{\gamma+1} \quad (19)\ [14]$$

$$\frac{p_{t1}}{p_{t0}} = \left(\frac{(\gamma+1)(M\sin\beta)^2}{(\gamma-1)(M\sin\beta)^2+2}\right)^{\frac{\gamma}{\gamma-1}} \left(\frac{(\gamma^{+1})}{2\gamma(M\sin\beta)^2-(\gamma-1)}\right)^{\frac{1}{\gamma-1}} \quad (20)\ [14]$$

$$\frac{T_1}{T_0} = \frac{(2\gamma(M\sin\beta)^2-(\gamma-1))((\gamma-1)(M\sin\beta)^2+2)}{(\gamma^{+1})^2(M\sin\beta)^2} \quad (21)\ [14]$$

$$\frac{\rho_1}{\rho_0} = \frac{(\gamma+1)(M\sin\beta)^2}{(\gamma-1)(M\sin\beta)^2+2} \quad (22)\ [14]$$

These equations come accompanied by an optimization criterion. As indicated by M.A. Hutomo and R.O. Bura [14] the Oswatitsch (1944) criterion states that the product between a Mach number before an oblique shock and the sine of the oblique shock angle must equal the product of the subsequent Mach number and the sine of the next oblique shock angle. This criterion ensures the best pressure recovery is obtained and by determining the shock angle we can then determine the ramp angle.

$$M_1 \sin(\beta_1) = M_2 \sin(\beta_2) = \cdots = M_{n-1} \sin(\beta_{n-1}) \quad (23)\ [14]$$

From the equations above the minimum dimensions of the intake can be calculated, of course, more work on the intake will need to be done using simulation data, although a preliminary result is possible.

After the normal shock a small section without slope is used as a transition zone to ensure the reattachment of the boundary layer. The cross-section area is constant, and the length is 2 times the height of the zone.

Using pre-determined dimensions given by engine data, showcased earlier, we can infer that the diameter of the last stage of the duct must have the same size as the engine width. The expansion of the duct to achieve the engine face width cannot be sudden, as it would occur in higher losses, according to Crosthwait, E. L. [15], the expansion angle $2\theta_d$ should be between 6 to 12 degrees to achieve maximum pressure recovery. In this case an expansion angle of 12 degrees is used as it saves length therefore minimizing the much important weight.

Total pressure recovery

A high total pressure recovery is the main goal of an intake design, although not the only one. For every shock that occurs, the total pressure decreases. There are also losses on the subsonic duct, therefore, to design the intake, it is necessary to calculate the total pressure recovery, and to meet its requirements. The used requirements are according to MIL-E-5008B, which provide a minimum total pressure recovery of the intake for every Mach number. This specification is as follows:

$$\eta_{spec} = 1 - 0.075(M - 1)^{1.75}, \text{ for } 1<M<5 \quad (24)\ [14]$$

Then, for the admission in this case, the minimum total pressure recovery required for a Mach 2.2 flight is 89,7%. Intakes that don't reach this value are to be discarded.
Between oblique shocks, the total pressure recovery ratio as assessed by Ran, H., and Mavris, D. [16], can be calculated as the following:

$$PR_i = \frac{p_{t_i}}{p_{t_{i-1}}} = \left[\frac{(\gamma+1)M_{i-1}^2 \sin^2\theta_i}{(\gamma-1)M_{i-1}^2 \sin^2\theta_i + 2}\right]^{\frac{\gamma}{\gamma-1}} \left[\frac{(\gamma+1)}{2\gamma M_{i-1}^2 \sin^2\theta_i - (\gamma-1)}\right]^{\frac{1}{\gamma-1}} \quad (25)\ [16]$$

The following calculates the total pressure recovery ratio between a normal shock:

$$PR_i = \frac{p_{t_i}}{p_{t_{i-1}}} = \left[\frac{(\gamma+1)M^2_{i-1}}{(\gamma-1)M^2_{i-1}+2}\right]^{\frac{\gamma}{\gamma-1}} \left[\frac{(\gamma+1)}{2\gamma M^2_{i-1}-(\gamma-1)}\right]^{\frac{1}{\gamma-1}} \quad (26)\ [16]$$

In the subsonic diffuser, since there aren't any shocks, and assuming the total temperature is constant:

$$PR_{Sub} = \frac{p_{t_i}}{p_{t_{i-1}}} = 1 - K_{Mth} K_d \left(1 - \frac{1}{AR_{i;i-1}}\right)^2 \frac{\frac{\gamma}{2} M^2_{i-1}}{\left(1+\frac{\gamma-1}{2} M^2_{i-1}\right)^{\frac{\gamma}{\gamma-1}}} \quad (27)\ [16]$$

Friction loss is represented by K_{Mth} coefficient and K_d is the expansion loss coefficient. However, these values are determined by experimental data and there were difficulties to access them and/or inability to reproduce these experiments, therefore, for this paper, all losses aren't accounted for. Instead, the stagnation pressure will be calculated as if the airflow was inviscid and incompressible.

Then, the total pressure recovery is calculated as:

$$TPR = \prod PR_i \times PR_{Sub} \quad (28)\ [16]$$

Parametric results

Having finished the MATLAB script for a mixed compression intake design, it was then tested for validation. To compare the total pressure recovery of an intake at Mach 2 with the same oblique shock angle (34.33°) and number of ramps (2x2 arrangement) to what Ommi, F., et al [17] experimented with, Table 1 was made.

Table 1 Comparison of values with Omni F. et al for validation of the script

	Parametrical Result for validation				Omni F. et al [17]			
Shock Number	Mach Number	Deviation Angle of Flow (deg)	Oblique Shock Angle (deg)	Pressure Recovery	Mach Number	Deviation Angle of Flow (deg)	Oblique Shock Angle (deg)	Pressure Recovery
0	2	5.03	34.33	0.998	2	5	34.33	0.997
1	1.820	5.318	38.29	0.998	1.821	5.3	38.24	0.997
2	1.637	5.556	43.57	0.998	1.638	5.556	43.51	0.997
3	1.447	5.576	51.24	0.998	1.448	5.558	51.11	0.997
4	1.245	Normal Shock	-	0.9877	1.127	Normal Shock	-	0.977
5	0.815	-	-	-	0.835	-	-	-

From Table 1 the total pressure result from this work and Omni F. et all is respectively TPR=0.980 and TPR=0.985, which although not equal, probably due to different calculating method, still proves the algorithm is valid for its intended purpose.

Having a proven algorithm, it was then possible to plot the intake geometry, having therefore, a visual representation of the admission. Looking back in the present work, the engine that was chosen was the RB199 MK103 which equipped the Panavia Tornado, so to have a better comprehension of the results, the intake is to be simulated at the operating conditions of this aircraft. These were obtained for the Mach number of 2.2 and at an altitude of 9000m.

A visual representation of the intake (Fig. 6 and Fig. 7) along with pressure recovery values, Mach numbers, oblique shock angles and flow deviation angles (Fig. 8) is provided below.

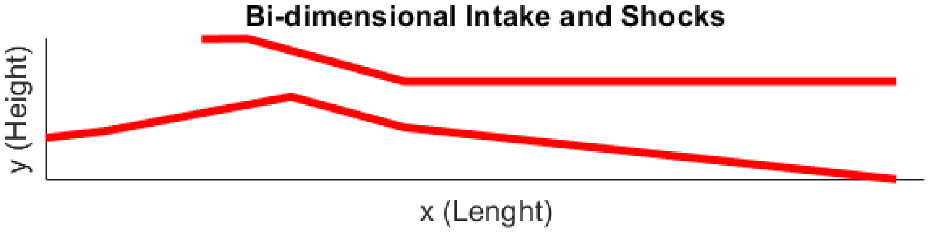


Figure 6 Normalized Inlet Plot with 2x2 ramp arrangement using MATLAB

Figure 6 was obtained with the developed MATLAB program, which automatically plots the intake design according to specified requirements and conditions. Figure 7 is a visual representation of what the shocks formed should look like on the designed intake.

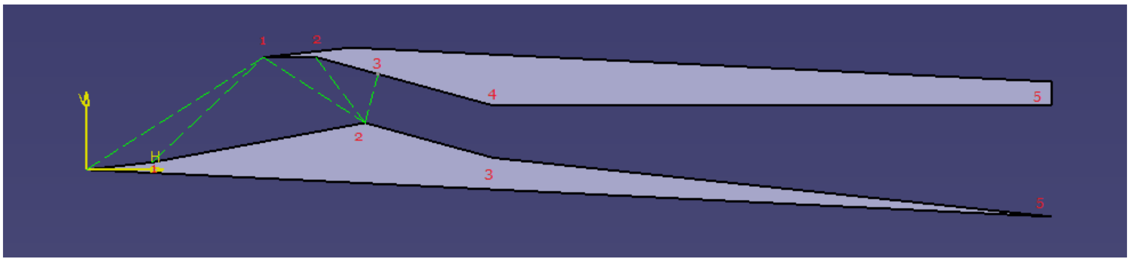


Figure 7 Visual Representation of the normalized intake with its formed shocks using CATIA V5

The obtained intake would have the following coordinates (Table 2):

Table 2 Station Points Coordinates of Normalized Intake

Station Points	0	1	2	3	4	5
Upper Wall (x)	-	1.5973	2.0675	2.7571	3.6680	8.7060
Upper Wall (y)	-	1.0098	1.0130	0.8252	0.5772	0.5772
Lower Wall (x)	0	0.5921	2.5130	3.6680	-	8.7060
Lower Wall (y)	0	0.0659	0.4212	0.1068	-	-0.4227

Using equation 29 and the results of Table 3, the total pressure recovery of this intake is 0.973539 with a normalized length relative to the engine face diameter (according to Table 2) of 8.7062. These values, considering there are no losses due to friction or expansion on the subsonic diffuser accounted for, are within a realistic spectrum of results. A by-product of the intake being slightly offset from the subsonic duct, as confirmed by Hasselrot, A., and Björn, M. [18], is a smaller radar signature of the engine compressor, although the expansion angle is set to 12°, simulations must be performed in computer fluid dynamics software in order to confirm if this deviation won't cause separation of the flow.

Table 3 Airflow and Shock Properties along the intake

	1	2	3	4	5	6
Mach	2.20	1.96	1.72	1.46	1.19	0.85
Oblique Shock Angles	32.30	36.84	43.21	53.41	80.57	90
Pressure	30723.157	44414.618	64207.540	92820.976	134185.697	199952.838
Stagnation Pressure	328513.834	326845.346	325185.332	323533.749	321890.555	319820.905
Density	0.466	0.606	0.787	1.022	1.328	1.763
Temperature	229.65	255.52	284.31	316.34	351.98	456.32
Flow Deviation Angle	6.354	6.872	7.342	7.376	2.747	0
Pressure recovery factor	0.995	0.995	0.995	0.995	0.994	1

Chapter 4

Optimization Study

Although good results and conclusions may come from a parametrical study, there are ways to achieve better results. Using optimization tools, it is possible to achieve an optimal result. To achieve this an evolutionary computation method is used to increase the performance of the intake, by varying each variable within a given interval.

Configuration of the multiobjective optimization with speeds ranging from Mach 1.8 to Mach 2.2

The multiobjective optimization was utilized following the results of the parametric study to find a viable intake across a range of Mach numbers. As such, different independent variables were used to create an initial population and bring forth an optimization of the following generations to create an optimized Pareto's front.

The objectives considered were the maximization of the total pressure recovery and the minimization of the relative intake length(in relation to engine face diameter).

The algorithm was created using MATLAB optimization toolbox [20] for the parametrical study algorithm mentioned earlier. To use this, the following independent variables were chosen:

- Initial Mach: from 1.8 to 2.2;
- First oblique shock angle: from 1 to 90 degrees;
- Number of external ramps: from 1 to 3;
- Number of internal ramps: from 1 to 3;

With the variables chosen, one must also characterise the operators, stoppage conditions and functions. An explanation to these parameters is found below:

- Crossover fraction: percentage of individuals present in the population that are generated by the crossover function;
- Distance measure function: maintains the diversity on the Pareto front, favouring the closest individuals to it. It takes an additional argument to calculate the distance either through a space function (phenotype) or by the sizing of the space (genotype);

- Pareto fraction: percentage of individuals in the population present in the Pareto front;
- Migration direction: movement of individuals between subpopulations, of which the algorithm creates when the size of an existing population in a vector is superior to 1. Therefore, the worst individuals of a subpopulation are replaced by the best individuals of another subpopulation;
- Migration interval: number of generations under which the migration is done;
- Limit of generations without evolution: if the average of changes over the fitness function for a determined number of generations is lower than the tolerance function, the algorithm stops;
- Tolerance function: used over the fitness function to evaluate the distance between populations;
- Generations: maximum number of generations to be generated by the algorithm;
- Creation function: creates an initial random population inside the boundaries set beforehand;
- Selection function: chooses individuals for the next generation based on their value inside the fitness function. The tournament parameter picks two random "progenitors" and picks the best suited one.
- Crossover function: Combines two individuals to form a new individual for the next generation. The heuristic type creates offspring which spawn randomly from two progenitors, with characteristics closer to the progenitor with the best fitness function value and further away from the progenitor with the worst fitness function value;
- Mutation function: it introduces small changes in individuals within each population, which provides genetical diversity and allows the optimization function to test a higher range of values within the set boundaries. With the Adaptive Feasible, a random direction is picked which is adapts depending on the results of the last generation obtained. An integration step is also defined so that with each direction of displacement of the population, the set boundaries are respected.

The parameters of which the optimization was made, are presented in Table 4:

Table 4 MATLAB configuration of the multiobjective optimization

Parameters	Configuration Values
Population Initial Range	[1.8 1 1 1] up to [2.2 90 3.9 3.9]
Population Size	300

Crossover Fraction	0.2
Distance Measure Function	@distancecrowding, 'genotype'
Pareto Fraction	0.4
Migration Direction	'both'
Migration Interval	5
Stall Generation Limit	60
Tolerance Function	1×10^{-5}
Generations	10000
Creation Function	Linear Feasible
Selection Function	'Tournament'
Crossover Function	Heuristic:0.6
Mutation Function	Adaptive Feasible

Optimization Results with speeds ranging from Mach 1,8 to Mach 2,2

After having executed the genetic algorithm's, the following results (Table 5) and Pareto's Front (Figure 8) were obtained:

Table 5 Genetic Algorithm results after removal of impossibilities

Relative lenght	Total Pressure Recovery	Mach number	1st Oblique shock angle	Nº of Int. Ramps	Nº of Ext. Ramps
8.5230	0.9807	2.00	38.66	1	1
7.8856	0.8806	1.90	33.82	2	1
7.2451	0.7157	2.07	29.12	2	3
7.6833	0.8589	1.97	32.59	1	2
8.1500	0.9833	1.99	34.45	2	2
8.5387	0.9865	2.03	35.61	2	1
7.5111	0.7992	1.99	31.09	2	2
8.1454	0.9849	1.99	34.59	2	2
7.2142	0.7748	1.94	31.67	1	1
7.0630	0.7600	1.93	31.37	1	1
7.2110	0.8023	1.89	32.74	1	1
8.0732	0.9205	1.97	33.00	2	2
7.9262	0.9415	1.92	34.20	2	2
8.0656	0.9572	1.95	34.06	2	2
7.3007	0.8173	1.92	32.39	1	2
7.5212	0.8252	1.98	31.94	1	2
8.1258	0.9784	1.98	34.37	2	2
7.9260	0.9414	1.92	34.20	2	2
7.9260	0.9415	1.92	34.20	2	2
7.7895	0.8890	1.96	33.31	1	2
7.7466	0.8029	2.04	30.54	2	2

8.5243	0.9807	2.00	38.66	1	1
8.0660	0.9573	1.95	34.06	2	2

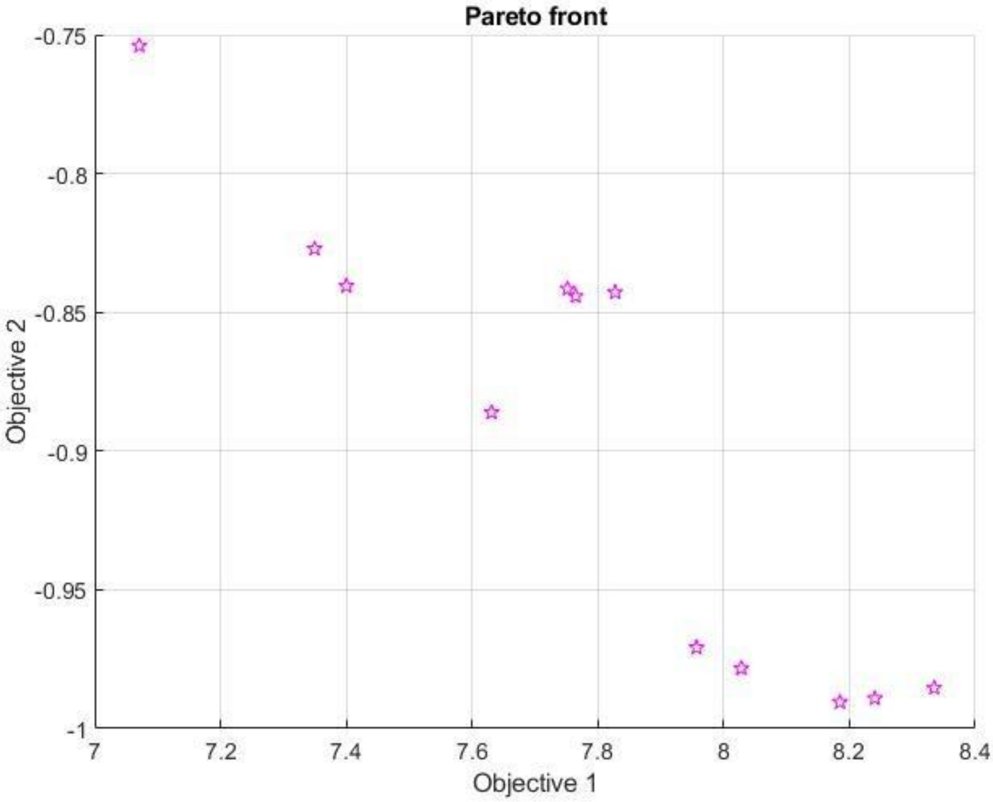


Figure 8 Pareto's Front for the multiobjective optimization with Mach ranging between 1.8 and 2.2 using MATLAB

We can represent these results in another way, and present a tendency line (Fig. 9), which helps better understand the results at hand:

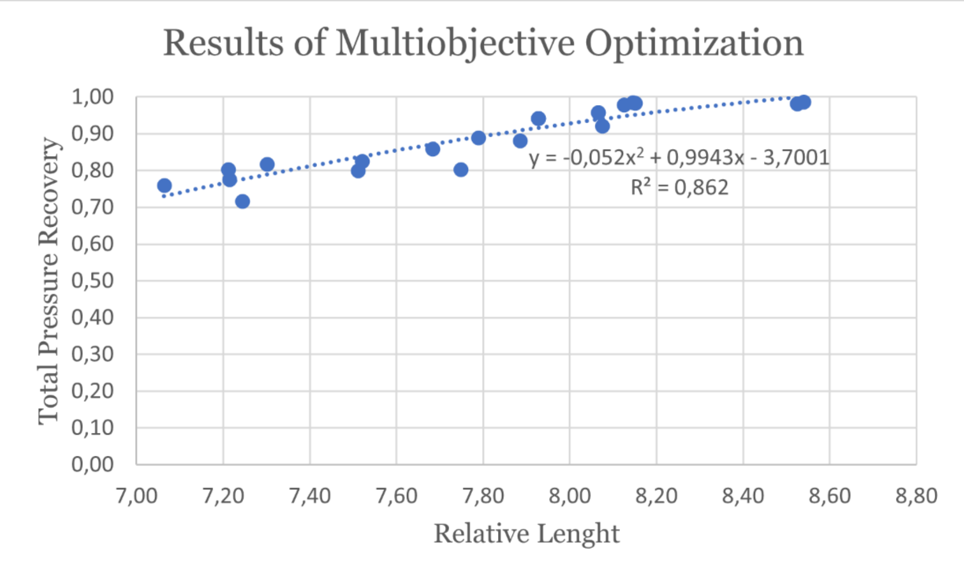


Figure 9 Dispersion Graph of the results of the multiobjective optimization

Out of these results, some were removed because they do not comply with the military requirements of total pressure recovery in equation 24. Therefore, the only valid solutions are in Table 6:

Table 6 Genetic Algorithm results complying with military requirements

Relative Length	Total Pressure Recovery	Mach Number	1st Oblique Shock Angle	Nº of Internal Ramps	Nº of External Ramps
8.5230	0.9807	2.00	38.66	1	1
8.1500	0.9833	1.99	34.45	2	2
8.5387	0.9865	2.03	35.61	2	1
8.1454	0.9849	1.99	34.59	2	2
7.9262	0.9415	1.92	34.20	2	2
8.0656	0.9572	1.95	34.06	2	2
8.1258	0.9784	1.98	34.37	2	2
7.9260	0.9414	1.92	34.20	2	2
7.9260	0.9415	1.92	34.20	2	2
8.5243	0.9807	2.00	38.66	1	1
8.0660	0.9573	1.95	34.06	2	2

With these results it is possible to make an evaluation through a dispersion graph. In Figure 10, the best results can be found when moving closer to the top left corner of the graph, maximizing the total pressure recovery while maintaining a shorter intake. If we search for these points in Table 6, it is possible to observe that the best results come typically when the Mach number is lower, this is the reason why after multiple iterations the Mach values are closer to the lower bound (tend to get closer to Mach 1,8). If the Mach number increases, it is still possible to maintain a high total pressure recovery, although the intake length will usually increase as a result due to the necessity of using more ramps to create weaker shocks. This would in turn harm the intake's off-design operation, as less ramps are usually better in this regard.

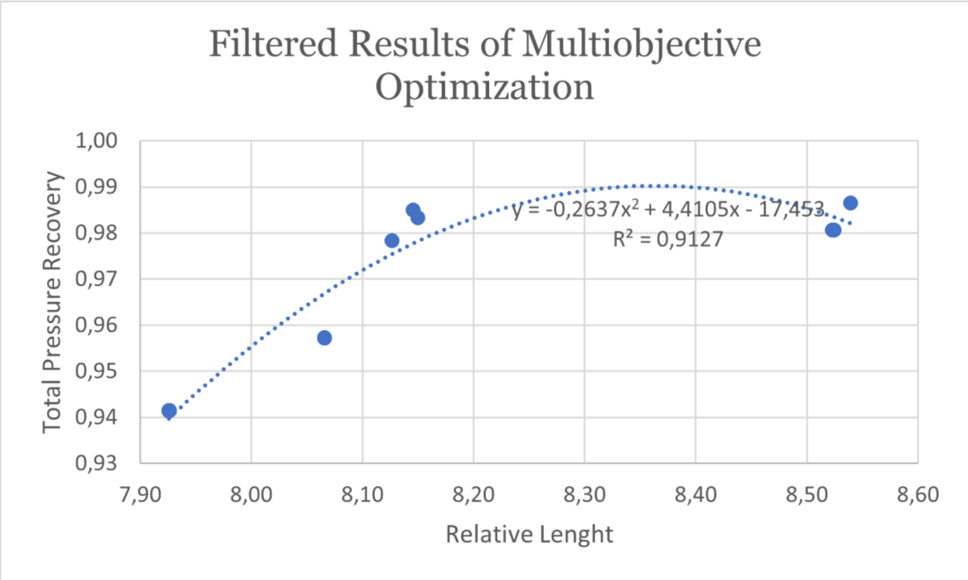


Figure 10 Filtered Results of Multiobjective Optimization

To better evaluate the results obtained, a multiobjective optimization for a single Mach number must be made, this will allow for a better evaluation of how the Mach number affects total pressure recovery and intake length.

Configuration of the multiobjective optimization for Mach 2.2

As with the configuration chapter of the multiobjective optimization for speeds ranging from Mach 1.8 to Mach 2.2, a parameter configuration table is presented below for an optimization for speeds of Mach 2.2.

It was needed to change the lower bound of the possible range of population in the Mach variable in comparison to the optimization made earlier. This parameter was increased from 1.8 to 2.2 in order to coincide with the upper bound (Table 7). This will force the algorithm to find the best solutions only for Mach 2.2.

Table 7 MATLAB configuration of the multiobjective optimization for Mach 2.2

Parameters	Configuration Values
Population Initial Range	[2.2 1 1 1] up to [2.2 90 3.9 3.9]
Population Size	300
Crossover Fraction	0.3
Distance Measure Function	@distancecrowding, 'genotype'
Pareto Fraction	0.4
Migration Direction	'both'
Migration Interval	5
Stall Generation Limit	60
Tolerance Function	1×10^{-5}
Generations	2500
Creation Function	Linear Feasible
Selection Function	'Tournament'
Crossover Function	Heuristic:0.6
Mutation Function	Adaptive Feasible

Optimization Results for Mach 2.2

As with the earlier optimization, a table of the optimization results are presented in Table 8:

Table 8 Genetic Algorithm's Optimization Results for Mach 2.2 after removal of impossibilities

Relative Lenght	Total Pressure Recovery	Mach number	Oblique Shock Angle	Nº of Internal Ramps	Nº of External Ramps
8.5605	0.9572	2.2	30.33	2	3
8.1251	0.7595	2.2	28.94	1	2
8.6389	0.8605	2.2	28.76	3	3
7.3836	0.6486	2.2	27.47	1	1
7.8204	0.6843	2.2	27.84	2	1
7.8494	0.6886	2.2	28.33	1	1
8.0910	0.8192	2.2	29.14	1	3
8.3795	0.8044	2.2	28.31	3	3
7.8971	0.7305	2.2	28.51	1	2
8.3870	0.8961	2.2	30.14	1	3
8.5077	0.9835	2.2	31.04	2	3
9.0025	0.9773	2.2	31.79	2	2
7.2961	0.6391	2.2	27.19	2	1
9.1933	0.9800	2.2	33.81	2	1

8.3618	0.9773	2.2	31.79	1	3
8.3870	0.8961	2.2	30.14	1	3
7.8494	0.6886	2.2	28.33	1	1
8.2174	0.8490	2.2	29.51	1	3
8.3487	0.9827	2.2	32.04	1	3
7.7057	0.6770	2.2	28.08	1	1
7.5015	0.6763	2.2	27.72	1	2
8.1251	0.7595	2.2	28.94	1	2
7.2669	0.6473	2.2	27.24	1	3
8.2342	0.7728	2.2	29.14	1	2
7.8204	0.6843	2.2	27.84	2	1
8.5658	0.9825	2.2	34.00	2	1

Of which the Pareto's Front is displayed in Figure. 11:

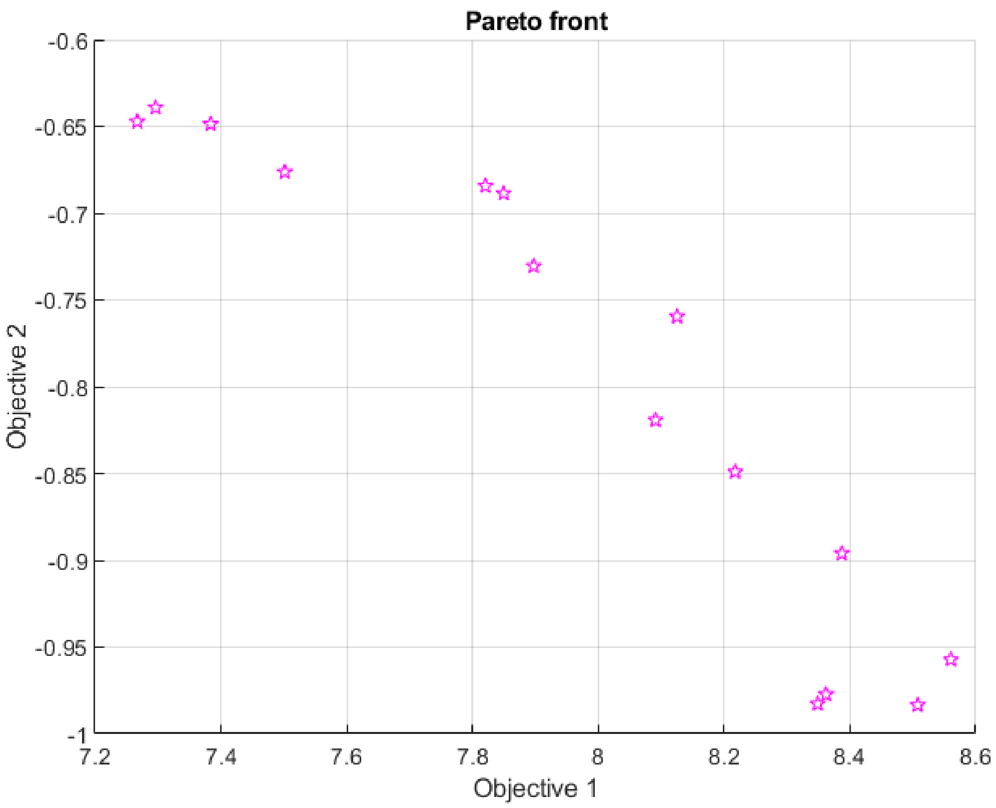


Figure 11 Paretos Front for the multiobjective optimization for Mach 2.2 using MATLAB

The results on Table 8 show possible solutions to the intake and can be presented in a dispersion graph with tendency line (Figure 12). Note that the tendency line rises above a total pressure recovery of one ($TPR > 1$) at about a relative length of 9, but any result above this mark is impossible.

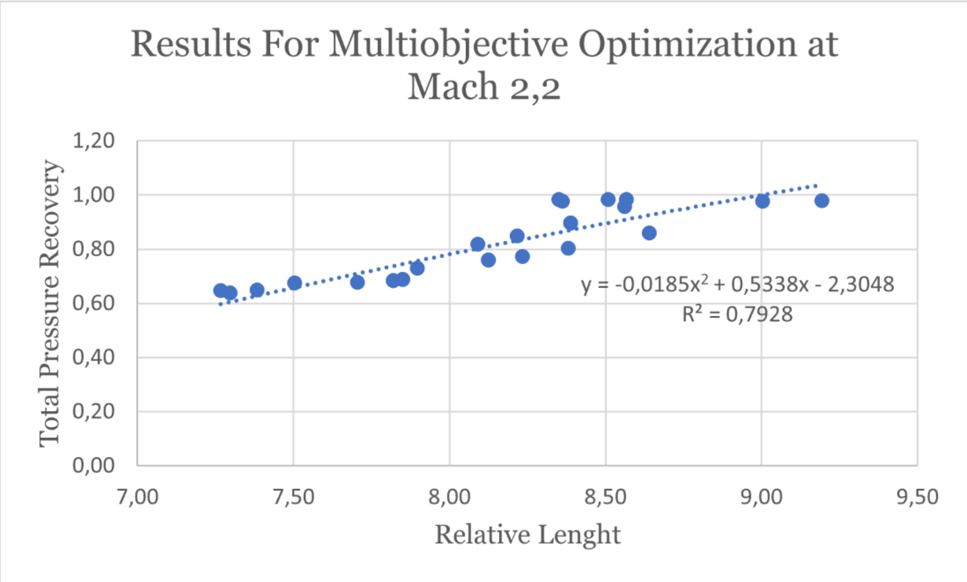


Figure 12 Dispersion graph of results of the multiobjective optimization for Mach 2.2

Not all the results comply with military requirement, as the minimum total pressure recovery of the admission should be of 89,68%. As such, the complying results are shown in Table 9:

Table 9 Genetic Algorithm's results complying with military requirements for Mach 2.2

Relative Lenght	Total Pressure Recovery	Mach number	Oblique Shock Angle	Nº of Internal Ramps	Nº of External Ramps
8.5605	0.9572	2.2	30.33	2	3
8.5077	0.9835	2.2	31.04	2	3
9.0025	0.9773	2.2	31.79	2	2
9.1933	0.9800	2.2	33.81	2	1
8.3618	0.9773	2.2	31.79	1	3
8.3487	0.9827	2.2	32.04	1	3
8.5658	0.9825	2.2	34.00	2	1

With these results a dispersion graph is presented with its tendency line (Fig. 13):

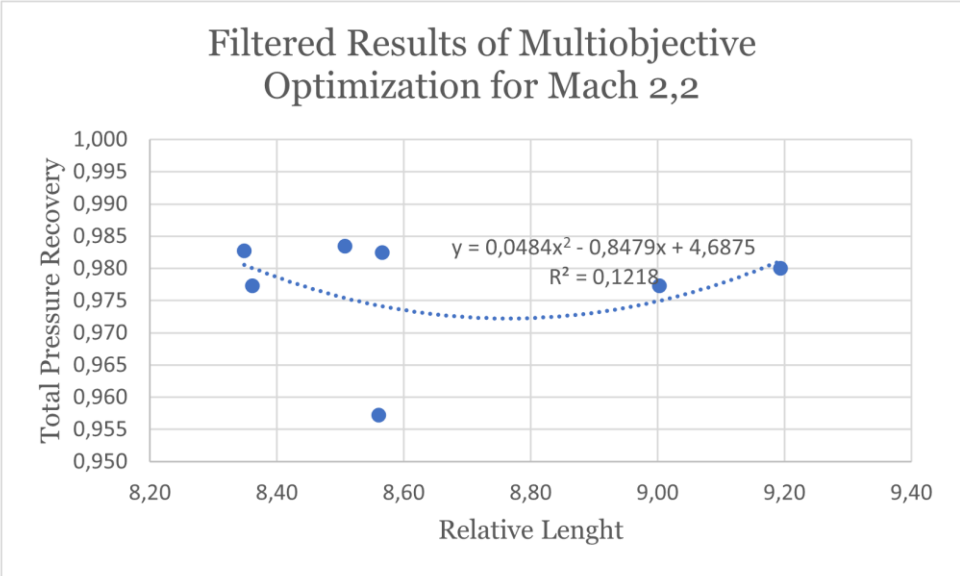


Figure 13 Filtered results of multiobjective optimization for Mach 2,2

As one can observe while comparing Figure 10 with Figure 13, the higher Mach frequently implies a longer intake, which in turn increases the amount of weight of the aircraft, sacrificing some performance. In Figure 13 we can also observe a result (8.5605;0.9572) which skews the tendency line.

Chapter 5

Final Considerations

Conclusion

This thesis objective was to design a bi-dimensional admission for a supersonic jet aircraft, this was better achieved through the usage of an mixed compression intake. The engine used as basis for the admission design was the Panavia Tornado's RB199 Mk103, when the aircraft would fly at its maximum Mach. Some values like the airflow velocity at the engine face were assumed, and some calculations were simplified, i.e., the subsonic duct pressure recovery, due to the lack of experimental data. After a design for the intake was calculated, a comparison was drawn with data from other source, the values of Mach, oblique shock angles, flow deviation angles, and pressure recovery between different stages were found to be acceptable, thus theoretically proving the intake is viable and validating the formulation.

From the parametrical result, an intake was obtained that complied with military requirements while having a simple design, the admission was slightly offset which should theoretically reduce radar visibility, although not all of it, since it is still possible to draw a straight line from the intakes entrance to the compressor's face without coming into contact with the inlet's walls.

Since the introduction of the Tornado in the 1970's, the requirements for supersonic airplanes changed, and the aircrafts that replace it in its role usually do not share the same maximum speed. Therefore, a decision was made to also introduce an optimization of the intake by means of a genetic algorithm, to provide with multiple solutions of intakes, while maintaining the speeds that match the Tornado's successors (i.e., Eurofighter Typhoon, F/A-18 Super Hornet or F-35). This optimization was carried with two objectives, minimizing the length, and maximizing the total pressure recovery. The four independent variables considered were the flight Mach, the first oblique shock angle, and the number of intake ramps (external and internal). This optimization was made for an altitude of 9000m, which is the same altitude that the Tornado achieves its highest Mach. The algorithm optimized the intake design, through a series of generations, producing a Pareto front.

Another multiobjective optimization was made, and this time the speed was set to a constant of Mach 2.2. This was to compare the results between the two optimizations.

The results of these optimization were the expected. A higher Mach signified a longer intake, which would in turn harm the aircrafts performance by having increased weight, but a high total pressure recovery was still possible. It is possible to conclude that if a high maximum speed is not absolutely required, a smaller intake accompanied by slower speed is preferable, as it achieves high total pressure recovery (becoming efficient) and saves weight.

Some issues were found mainly on the multiobjective optimization although a workaround was found. These issues meant a higher computational time and sometimes crashes while running the optimization, as MATLAB became unresponsive.

Future studies

In a future study, it would be best to run simulations on the design, along with building a Mach 2.0 experimental setup of the designed intake to evaluate its performance. It would then be possible to evaluate pressure recovery, friction, expansion and drag losses. With these values a further refinement of the script could be made, further enhancing it. It could be of interest to test the intake with blow-in doors and/or bleed to enhance the performance. A study on the off-design conditions of the designed intake (Fig.7) could provide valuable information.

With access to this data and studies, a better understanding can be reached allowing for future design processes to be more accurate in their results, providing better performance.

Bibliography

References

[1] Seddon, J., and Goldsmith, E. L., "Intake Aerodynamics", Collins Professional and Technical Books, London, 1985.

[2] Sóbester. A., "Tradeoffs in Jet Inlet Design: A Historical Perspective", Journal of Aircraft Vol. 44, No. 3, May-June 2007.

[3] Laruelle, G., "Air Intakes: Role, Constraints and Design", ICAS 2002 Congress, 8-13 September, 2002, Toronto, Canada.

[6] Anderson, J. T., "How Supersonic Inlets Work: Details of the Geometry and Operation of the SR-71 Mixed Compression Inlet", Technical Fellow Emeritus Lockheed Martin Skunk Works.

[14] Hutomo M. A., and Bura. R. O., "Conceptual Design of Supersonic Mixed-Compression Intake For Ramjet Powered Target Drone", Technical Report November 2015, Bandung Institute of Technology, Ganesha 10, Bandung, Indonesia.

[15] Crosthwait, E. L., Kennon, I.G., Jr., et al, "Preliminary design methodology for Air-Induction Systems", Technical Report SEG-TR-67-1, Systems Engineering Group, 1967.

[16] Ran, H., and Mavris, D., "Preliminary Design Of A 2D Supersonic Inlet To Maximize Total Pressure Recovery", AIAA 5th Aviation, Technology, Integration, and Operations Conference, 26-28 September 2005, Arlington, Virginia.

[17] Ommi, F., Khanaposhtani, V. F., et al, "A New Approach for Supersonic Diffuser Design", Journal of Applied Sciences Research, 6(5): 401-414, 2010.

[18] Hasselrot, A., and Björn, M., "An Overview of Propulsion Systems for Flying Vehicles", Technical Report FOI-R—1563-SE, ISSN 1650-1942, June 2005.

[19] Tavares, G., "Otimização de um Motor Turbofan com Combustão Sequencial e Elevada Razão de Bypass", Universidade da Beira Interior (Master's Thesis).

Links used

[4]-https://www.flight-manuals-online.com/wp-content/uploads/2020/04/Tornado-photo-scaled.jpg - Date of access: 21-07-2022

[5]-https://d1a2ot8agkqe8w.cloudfront.net/web/2011/11/f-15c-pair-us-air-force_43181.jpg - Date of access: 21-07-2022

[7]-https://aeromagazine.uol.com.br/media/mig-21_romenia.jpg - Date of access: 21-07-2022

[8]-https://live.staticflickr.com/1742/40828117580_d975889df6_b.jpg - Date of access: 21-07-2022

[9]-https://www.jsf.mil/diverterless_inlet - Date of access: 01-08-2022

[10]-https://sinodefence.files.wordpress.com/2017/06/133743xridcyl98rlirirq.jpg - Date of access: 21-07-2022

[11]-https://wallpaper-house.com/data/out/6/wallpaper2you_106982.jpg - Date of access: 21-07-2022

[12]-https://rocketgirl2017.files.wordpress.com/2018/05/internal-external-mixed-comp.png - Date of access: 22-07-2022

[13]-https://www.flugzeuglexikon.com/ILA%20-%20Luftfahrtausstellung/Kampfflugzeuge/Tornado%20-%20Panavia%20Aircraft/Triebwerk-RB-199---MK-103.jpg - Date of access: 21-07-2022

[20]-https://www.mathworks.com/help/gads/gamultiobj.html - Date of access: 01-08-2022

Appendix

press_recov_simples.m

```
function [Lenght,tpr]=press_recov_simples(M,beta,Int,Ext)
%clear all
clc
sumH=0;
%INSERT ENGINE REQUIREMENTS HERE
%Insert Airflow in kg/s
AirFlow=71.21;
%Insert initial requirements here
R=287;
%Ext=2;
%Int=1;
Ext_R=fix(Ext);
Int_R=fix(Int);
gamma=1.4;
table=zeros(10,(Ext_R)+2+(Int_R));
%Insert angle of attack
alpha=0;
%Insert Mach speed
%M=2.2;
table(1,1)=M;
%Insert angle of first shock in degrees
%beta=33.8;
%Insert exterior ramp lenght
D=1.3435;
x=0;
table(2,1)=beta;
%Insert altitude in meters
h=9000;
%Free Flow Conditions
T0 = Temp_H (h);
p0 = pressureH(T0,h);
RHO0 = RHO_h(p0,T0);
table(3,1)=p0;
pt0 = pt_h(gamma,p0,M);
table(4,1)=pt0;
table(5,1)=RHO0;
table(6,1)=T0;
a0 = SpSound(gamma,R,T0);
V0=M*a0;
%Capture Area Required for the engine
Cap_Area=AirFlow/(RHO0*V0);
%Minimum Pressure Recovery according to Military Standards, MIL-E-5008B
r_p_rec = ref_press_recov(M);
%Initial requirements end here
%program for formulating a table of information of each step in the ramp
for x=0:(Ext_R+Int_R)
    x=x+1;
    [M1,MratioOB] = MachRatioOB(gamma,table(1,x),table(2,x));
    table(1,(x+1))=M1;
    [beta1]=Oswatitsch(table(1,x),table(2,x),table(1,(x+1)));
    table(2,(x+1))=beta1;
    [pratio,p1] = p_ratioOB(table(3,x),table(1,x),gamma,table(2,x));
    table(3,(x+1))=p1;
    [pt_rat,pt1] = p_t_ratioOB(table(4,x),gamma,table(1,x),table(2,x));
    table(4,(x+1))=pt1;
    [RHOrat,RHO1] = Rho_ratioOB(table(5,x),gamma,table(1,x),table(2,x));
    table(5,(x+1))=RHO1;
    [T1, Trat] = T_ratioOB(gamma,table(1,x),table(6,x),table(2,x));
    table(6,(x+1))=T1;
```

```
    [theta,tan_theta] = Oblique_shock(table(1,x),gamma,table(2,x));
    table(7,(x))=theta;
end
for x=0:(Ext_R+Int_R-1)
    x=x+1;
    [pr] = precovery(gamma,table(1,x),table(2,x));
    table(8,x)=pr;
end
    [M1,Mrat] = Mach_ratioN(table(1,(Ext_R+Int_R+1)),gamma);
    table(1,(Ext_R+Int_R+2))=M1;
    table(2,Ext_R+Int_R+2)=90;
    [pratio,p1] =
p_ratioN(table(3,Ext_R+Int_R+1),table(1,(Ext_R+Int_R+1)),gamma);
    table(3,(Ext_R+Int_R+2))=p1;
    [pt_rat,pt1] =
p_t_ratioN(table(4,Ext_R+Int_R+1),gamma,table(1,Ext_R+Int_R+1));
    table(4,(Ext_R+Int_R+2))=pt1;
    [RHOrat,RHO1] =
Rho_ratioN(table(5,Ext_R+Int_R+1),gamma,table(1,Ext_R+Int_R+1));
    table(5,(Ext_R+Int_R+2))=RHO1;
    [T1, Trat] = T_ratioN(gamma,table(1,Ext_R+Int_R),table(6,Ext_R+Int_R+1));
    table(6,(Ext_R+Int_R+2))=T1;
    [pr] = precoveryN(gamma,table(1,(Ext_R+Int_R+1)));
    table(8,Ext_R+Int_R+1)=pr;
%VERIFY PRESSURE RECOVERY FUNCTION MAY BE BADLY IMPLEMENTED OR EVEN THE


%intake capture area
%FUNÇÃO PARA CALCULAR POSIÇÕES DE RAMPA E LIP
sumLip=0;
yLip=D*tand(table(2,1));
ramp_angle=0;
d_sum=0;
h_sum=0;
dlip_sum=0;
hlip_sum=0;
Dint_sum=0;
if (Int_R > 0 && Ext_R > 0) %Mixed compression intake
    for x=0:(Ext_R+Int_R)
        x=x+1;
        if x<(Ext_R)
        ramp_angle=ramp_angle+table(7,x);
        %ramp_angle
        table(9,x)=(yLip-h_sum-
D*tand(table(2,x+1)+ramp_angle)+d_sum*tand(table(2,x+1)+ramp_angle))/(tand(ram
p_angle)-tand(table(2,x+1)+ramp_angle));
        %table(9,x)
        d_sum=d_sum+table(9,x);
        %d_sum
        h_sum=h_sum+(table(9,x)*tand(ramp_angle));
        %h_sum
        table(10,x)=h_sum;
        elseif x == Ext_R
        ramp_angle=ramp_angle+table(7,x);
        table(9,x)=(yLip-(sum(table(9,1:Ext_R-1))-D)*tand(table(2,Ext_R+1)-
ramp_angle)-h_sum)/(tand(table(2,Ext_R+1)-ramp_angle)+tand(ramp_angle));
        d_sum=d_sum+table(9,x);
        h_sum=h_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=h_sum;
        elseif x>Ext_R && x<Ext_R+Int_R
        D_int=d_sum-D;
        ramp_angle=ramp_angle-table(7,x);
        table(9,x)=(yLip+hlip_sum+(D_int-dlip_sum)*tand(ramp_angle-
table(2,x+1))-h_sum)/(tand(ramp_angle-table(2,x+1))-tand(ramp_angle));
        dlip_sum=dlip_sum+table(9,x);
        %ramp_angle=ramp_angle-table(7,x);
        hlip_sum=hlip_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=yLip+hlip_sum;
```

```
        elseif x==(Ext_R+Int_R)
        if Int_R==1
        D_int=d_sum-D;
        table(9,x)=D_int;
        else
        table(9,x)=D_int-dlip_sum;
        end
        ramp_angle=ramp_angle-table(7,x);
        hlip_sum=hlip_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=yLip+hlip_sum;
        end
        ramp_angle=ramp_angle-table(7,Ext_R+Int_R+1);
    end
elseif (Int_R==0 && Ext_R>0) %External Compression Intake
    for x=0:(Ext_R-1)
        x=x+1;
        if Ext_R==1
        ramp_angle=ramp_angle+table(7,x);
        table(9,x)=D;
        table(10,x)=D*tand(table(7,x));
        end
        if x<(Ext_R)
        ramp_angle=ramp_angle+table(7,x);
        table(9,x)=(yLip-(D-d_sum)*tand(table(2,x+1)+ramp_angle)-
h_sum)/(tand(ramp_angle)-tand(table(2,x+1)+ramp_angle));
        d_sum=d_sum+table(9,x);
        h_sum=h_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=h_sum;
        elseif x==(Ext_R)
        ramp_angle=ramp_angle+table(7,x);
        table(9,x)=D-d_sum;
        h_sum=h_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=h_sum;
        end
    end
    else
        end
    W=Cap_Area/yLip;
    %calculation of the necessary area at engine face to deliver air at
    %M=0.4
    a_throat=SpSound(gamma,R,table(6,Ext_R+Int_R+2));
    %Calculation of static pressure and stagnation pressure at engine face

p_engine=table(3,Ext_R+Int_R+2)+((table(5,Ext_R+Int_R+2)*((table(1,Ext_R+Int_R
+2)*a_throat)^2))/2)-((table(5,Ext_R+Int_R+2)*((0.4*a_throat)^2))/2);
    pt5 = pt_h(gamma,p_engine,0.4);
    p_rec_sub=pt5/table(4,Ext_R+Int_R+2);
    table(8,Ext_R+Int_R+2)=p_rec_sub;
    %PRESSURE RECOVERY FUNCTION
tpr=1;
    for x=0:(Ext_R+Int_R+1)
        x=x+1;
        tpr=table(8,x)*tpr;
    end
    %Removal of physically impossible data
    possible=isreal(table);
        if possible == 0
        table=NaN;
        tpr=NaN
        Lenght=NaN
    end

    y=0;
    if possible== 1
    while y<((Ext_R)+(Int_R)+1)
        y=y+1;
        if table(1,y+1)>table(1,y)
            table=NaN;
```

```
            tpr=NaN
            Lenght=NaN
            break
        end
    end
    end

    if tpr>=r_p_rec
        fprintf('The intake total pressure recovery (%f) is within military 
stantards (%f)',tpr,r_p_rec)
    elseif tpr<r_p_rec
        fprintf('The intake total pressure recovery (%f) is outside military 
standards (%f)',tpr,r_p_rec)
    elseif isnan(tpr)== 1
        fprintf('The chosen intake is not valid ')

    end

if isnan(tpr)==1
      Lenght=NaN;
      tpr=NaN;
      return
else
  %INSERT SIZING ALGORITHM HERE QUICKLY PLZ, REARRANGE IN ORDER TO FIX
  dimension_ext=zeros(2,Ext_R+3);
  dimension_ext(1,1)=0;
  dimension_ext(2,1)=0;
    for x=1:Ext_R
    dimension_ext(2,x+1)=table(10,x);
    dimension_ext(1,x+1)=sum(table(9,1:x));
    end
  dimension_int=zeros(2,Int_R+1);
  dimension_int(1,1)=D;
  dimension_int(2,1)=yLip;
  for x=2:Int_R
      dimension_int(1,x)=sum(table(9,Ext_R+1:x+Ext_R-1))+D;
      dimension_int(2,x)=table(10,x+Ext_R-1);
  end
  l=(sind(ramp_angle+90))*(sqrt(((dimension_ext(1,Ext_R+1)-
dimension_int(1,Int_R))^2)+((dimension_ext(2,Ext_R+1)-
dimension_int(2,Int_R))^2)));
  dimension_int(2,Int_R+1)=dimension_int(2,Int_R)+(l*sind(ramp_angle));
  dimension_int(1,Int_R+1)=dimension_int(1,Int_R)+(l*cosd(ramp_angle));
  %transition zone plotting dimensions
  d_trans=2*sqrt(((dimension_int(2,Int_R+1)-
dimension_ext(2,Ext_R+1))^2)+((dimension_int(1,Int_R+1)-
dimension_ext(1,Ext_R+1))^2));
  %Interruption for calculating dimension at engine face

A_engine=((table(1,Ext_R+Int_R+2)*a_throat)*(sqrt(((dimension_int(2,Int_R+1)-
dimension_ext(2,Ext_R+1))^2)+((dimension_int(1,Int_R+1)-
dimension_ext(1,Ext_R+1))^2))))/(0.4*a_throat);
  %Continuation of transition zone plotting dimensions
  dimension_ext(1,Ext_R+2)=d_trans*cosd(ramp_angle)+dimension_ext(1,Ext_R+1);
  dimension_ext(2,Ext_R+2)=d_trans*sind(ramp_angle)+dimension_ext(2,Ext_R+1);
  dimension_int(1,Int_R+2)=dimension_int(1,Int_R+1)+d_trans*cosd(ramp_angle);
  dimension_int(2,Int_R+2)=dimension_int(2,Int_R+1)+d_trans*sind(ramp_angle);
  dimension_ext(1,Ext_R+3)=dimension_int(1,Int_R+2);

dimension_ext(2,Ext_R+3)=dimension_ext(2,Ext_R+2)+((dimension_ext(1,Ext_R+3)-
dimension_ext(1,Ext_R+2))*tand(ramp_angle));
  %plotting dimensions of subsonic duct
  h_exp=A_engine-(dimension_int(2,Int_R+2)-dimension_ext(2,Ext_R+3));
  d_exp=h_exp/tand(6);
  dimension_ext(1,Ext_R+4)=dimension_ext(1,Ext_R+3)+d_exp;
  dimension_ext(2,Ext_R+4)=dimension_ext(2,Ext_R+3)-h_exp;
  dimension_int(1,Int_R+3)=dimension_ext(1,Ext_R+4);
  dimension_int(2,Int_R+3)=dimension_int(2,Int_R+2);
```

```
  %Relative Lenght function
  Lenght=dimension_ext(1,Ext_R+4)/A_engine
  %Relative Height
  H=max(dimension_int,[],2);
  Hmax=H(2,1);
  L=min(dimension_ext,[],2);
  Hmin=L(2,1);
  HDiff=(Hmax-Hmin)/A_engine;
%   %plotting function
%   figure
%   hold on
%   plot(dimension_ext(1,1:Ext_R+4),dimension_ext(2,1:Ext_R+4),'r',
'LineWidth',3)
%   %introduction to plotting internal ramps here
%
plot(dimension_int(1,1:Int_R+3),dimension_int(2,1:Int_R+3),'r','LineWidth',3)
%   %introduction to plotting shocks
%   %plot(,'b--','LineWidth',2)
%   xlabel('x (Lenght)')
%   ylabel('y (Height)')
%   grid on
%   title('Bi-dimensional Intake and Shocks')
%   daspect([1 1 1])
%   set(gca,'Xtick',[],'YTick',[])

  end

  end
```

Mach.m

```
function [M] = Mach(v,a)
%Mach in fuction of speed and speed of sound function
M=v/a;
end
```

Mach_ratioN.m

```
function [M1,Mrat] = Mach_ratioN(M,gamma)
%Mach after shock funtion
M1=sqrt(((gamma-1)*(M^2)+2)/(2*gamma*(M^2)-(gamma-1)));
Mrat=M1/M;
end
```

MachRatioOB.m

```
function [M1,MratioOB] = MachRatioOB(gamma,M,beta)
%Mach Ratio function using oblique shocks
M1=sqrt((((gamma+1)^2)*(M^4)*((sind(beta))^2)-4*((M^2)*((sind(beta))^2)-
1)*(gamma*(M^2)*((sind(beta))^2)+1))/((2*gamma*(M^2)*((sind(beta))^2)-(gamma-
1))*((gamma-1)*(M^2)*((sind(beta))^2)+2)));
MratioOB=M1/M;
end
```

Mass_FP.m

```
function [MFP] = Mass_FP(M1,gamma,g,R)
%Mass Flow Parameter
MFP =M1*sqrt((gamma*g)/R)*(1+((gamma-1)/2)*(M1^2));
end
```

Oblique_shock.m

```
function [theta,tan_theta] = Oblique_shock(M1,gamma,beta)
%oblique shock angles
tan_theta=2*cotd(beta)*(((M1^2)*((sind(beta))^2)-
1)/((M1^2)*(gamma+cosd(2*beta))+2));
theta=atand(tan_theta);
end
```

Oswatitsch.m

```
function [beta1] = Oswatitsch(M,beta,M1)
%Oswatitsch (1944) optimization
sin_beta1=(M*sind(beta))/M1;
beta1=asind(sin_beta1);
end
```

p_ratioN.m

```
function [pratio,p1] = p_ratioN(p0,M,gamma)
%this function describes the pressure ratio between shocks
p1=p0*((2*gamma*(M^2))-(gamma-1))/(gamma+1);
pratio=p1/p0;
end
```

p_ratioOB.m

```
function [pratio,p1] = p_ratioOB(p0,M,gamma,beta)
%this function describes the pressure ratio between oblique shocks
p1=p0*((2*gamma*(M*sind(beta))^2)-(gamma-1))/(gamma+1);
pratio=p1/p0;
end
```

p_t_ratioN.m

```
function [pt_rat,pt1] = p_t_ratioN(pt0,gamma,M)
%Stagnation pressure ratio function
pt_rat=((((gamma+1)*(M^2))/((gamma-1)*(M^2)+2))^(gamma/(gamma-
1)))*(((gamma+1)/(2*gamma*(M^2)-(gamma-1)))^(1/(gamma-1)));
pt1=pt0*pt_rat;
end
```

p_t_ratioOB.m

```
function [pt_rat,pt1] = p_t_ratioOB(pt0,gamma,M,beta)
%oblique shock Stagnation pressure ratio function
pt_rat=((((gamma+1)*((M*sind(beta))^2))/((gamma-
1)*((M*sind(beta))^2)+2))^(gamma/(gamma-
1)))*(((gamma+1)/(2*gamma*((M*sind(beta))^2)-(gamma-1)))^(1/(gamma-1)));
pt1=pt0*pt_rat;
end
```

precovery.m

```
function [pr] = precovery(gamma,M,beta)
%Function used to calculate pressure recovery
pr=((((gamma+1)*(M^2)*((sind(beta))^2)/((gamma-
1)*(M^2)*((sind(beta))^2)+2)))^(gamma/(gamma-
1)))*(((gamma+1)/(2*gamma*(M^2)*((sind(beta))^2)-(gamma-1)))^(1/(gamma-1)));

end
```

Rho_ratioN.m

```
function [RHOrat,RHO1] = Rho_ratioN(RHO0,gamma,M)
%density ratio function
RHOrat=((gamma+1)*(M^2))/((gamma-1)*(M^2)+2);
RHO1=RHOrat*RHO0;
end
```

Rho_ratioOB.m

```
function [RHOrat,RHO1] = Rho_ratioOB(RHO0,gamma,M,beta)
%oblique shock density ratio function
RHOrat=((gamma+1)*((M*sind(beta))^2))/((gamma-1)*((M*sind(beta))^2)+2);
RHO1=RHOrat*RHO0;
end
```

speed_sound.m

```
function [a] = speed_sound(gamma,R,T)
%Speed of sound function
a=sqrt(gamma*R*T);
end
```

T_ratioN.m

```
function [T1, Trat] = T_ratioN(gamma,M,T0)
%Temperature ratio function
Trat=((2*gamma*(M^2)-(gamma-1))*((gamma-1)*(M^2)+2))/(((gamma+1)^2)*(M^2));
T1=T0*Trat;
end
```

T_ratioOB.m

```
function [T1, Trat] = T_ratioOB(gamma,M,T0,beta)
%oblique shock Temperature ratio function
Trat=((2*gamma*((M*sind(beta))^2)-(gamma-1))*((gamma-
1)*((M*sind(beta))^2)+2))/(((gamma+1)^2)*((M*sind(beta))^2));
T1=T0*Trat;
end
```

pressure.m

```
function [p0] = pressureH(T0,h)
%Pressure calculation at given altitude
p0=101325*((T0/288.15)^((-9.81)/(-0.0065*287)));
end
```

Temp_H.m

```
function T0 = Temp_H (h)
%ambient temperature in function of altitude
T0=288.15+(-0.0065*h);
end
```

RHO_h.m

```
function RHO0 = RHO_h(p0,T0)
%air density acording to pressure at given altitude
RHO0=p0/(287*T0);
end
```

pt_h.m

```
function pt0 = pt_h(gamma,p0,M)
%Stagnation Pressure function
pt0=p0*((1+((gamma-1)/2)*(M^2))^(gamma/(gamma-1)));
end
```

codigo.m

Problema e Limites

```
FitnessFunction = @optimizacao; %Indica a Funccao Fitness
numberOfVariables = 4;          %Numero de variáveis
lb = [2.2 1 1 1];         %Limite inferior dos inputs
ub = [2.2 90 3.9 3.9];            %Limite superior dos inputs
A = [];
b= [];
Aeq = [];
beq = [];
```

Opcoes de Optimizacao

```
options = gaoptimset('PopInitRange', [lb;ub]);
%Limites da Populção Incial
options = gaoptimset(options,'PopulationSize', 300);
%Dimensão de cada População
options = gaoptimset(options,'CrossoverFraction', 0.3);
%Percentagem de individuos presentes na populacao feitos por Crossover
options =
gaoptimset(options,'DistanceMeasureFcn',{@distancecrowding,'genotype'});
%Ajuda a manter a diversidade e tipo de distancia entre individuos na...

...Frente de Pareto: "genotype": design space; "phenotype": function space
options = gaoptimset(options,'ParetoFraction', 0.4);
%Percentagem de indivíduos na Frente de Pareto
options = gaoptimset(options,'MigrationDirection', 'both');
%Direccao de migracao: os melhores individuos de uma subpopulacao...

...substituem os piores os piores individuos em outra subpopulacao...

...Nao sao removidos da fonte sao copiados
options = gaoptimset(options,'MigrationInterval', 5);
%Numero de geracoes que executa migracao
options = gaoptimset(options,'StallGenLimit', 60);
%Numero de geracoes de paragem
options = gaoptimset(options,'TolFun', 1e-5);
%Tolerancia de Paragem: se a media ponderada de mudanca dos resultados ao
longo...

...da StallGenLimit for menor que a Tolerancia de Paragem, o algoritmo para.
```

```
options = gaoptimset(options,'Generations', 2500);
%Numero de geracoes
options = gaoptimset(options,'CreationFcn', @gacreationlinearfeasible);
%Cria populacoes dentro dos limites
options = gaoptimset(options,'SelectionFcn', {  @selectiontournament [] });
%Selecciona cada parente escolhendo 2 inidividuos aleatoriamente e escolhe o
melhor para ser parente
options = gaoptimset(options,'CrossoverFcn', {  @crossoverheuristic 0.6 });
%Combina 2 individuos para formar um novo. Heuristica: cria a uma distancia do
melhor parente
options = gaoptimset(options,'MutationFcn', @mutationadaptfeasible);
%Mudancas aleatorias nos individuos, conferem diversidade. Adaptive Feasible:
considera as bem sucedidas

%options = gaoptimset(options,'HybridFcn', {  @fgoalattain [] });
%Funcao que ajuda na minimizacao
options = gaoptimset(options,'HybridFcn', []);                    %Funcao que
ajuda na minimizacao

options = gaoptimset(options,'Display', 'diagnose');
%Tipo de visualizacao
options = gaoptimset(options,'PlotFcns', { @gaplotdistance
@gaplotscorediversity @gaplotselection @gaplotpareto @gaplotparetodistance
@gaplotrankhist @gaplotspread });

%options = gaoptimset('PlotFcns',@gaplotpareto);

%options = gaoptimset(options,'HybridFcn',@fgoalattain);
%set(RandStream.getGlobalStream,'State',Output.rngstate.state);
```

Execucao da Optimizacao e Extraccao dos Resultados

```
[x,Fval,exitFlag,Output,population,score] =
gamultiobj(FitnessFunction,numberOfVariables,A,b,Aeq,beq,lb,ub,options);
```

optimizacao.m

```
function f = optimizacao(x)

[Lenght,tpr] = press_recov_simples(x(1),x(2),x(3),x(4));
f(1) = Lenght;
f(2) = -tpr;

end
```

ref_press_recov.m

```
function r_p_rec = ref_press_recov(M)
%Military total pressure recovery reference MIL-E-5008B
%Intake pressure recovery must be atleast this value or above
r_p_rec=1-0.075*(M-1)^1.75;
%this is used for mach speed between 1 and 5
end
```

TESTINGFUNCTION.m

```
clear all
clc
sumH=0;
%INSERT ENGINE REQUIREMENTS HERE
%Insert Airflow in kg/s
AirFlow=71.21;
%Insert initial requirements here
R=287;
Ext=2;
Int=2;
Ext_R=fix(Ext);
Int_R=fix(Int);
gamma=1.4;
table=zeros(10,(Ext_R)+2+(Int_R));
%Insert angle of attack
alpha=0;
%Insert Mach speed
M=2;
table(1,1)=M;
%Insert angle of first shock in degrees
beta=34.33;
%Insert exterior ramp lenght
D=2;
x=0;
table(2,1)=beta;
%Insert altitude in meters
h=9000;
%Free Flow Conditions
T0 = Temp_H (h);
p0 = pressureH(T0,h);
RHO0 = RHO_h(p0,T0);
table(3,1)=p0;
pt0 = pt_h(gamma,p0,M);
table(4,1)=pt0;
table(5,1)=RHO0;
table(6,1)=T0;
a0 = SpSound(gamma,R,T0);
V0=M*a0;
%Capture Area Required for the engine
Cap_Area=AirFlow/(RHO0*V0);
%Minimum Pressure Recovery according to Military Standards, MIL-E-5008B
r_p_rec = ref_press_recov(M);
%Initial requirements end here
%program for formulating a table of information of each step in the ramp
for x=0:(Ext_R+Int_R)
    x=x+1;
    [M1,MratioOB] = MachRatioOB(gamma,table(1,x),table(2,x));
    table(1,(x+1))=M1;
    [beta1]=Oswatitsch(table(1,x),table(2,x),table(1,(x+1)));
    table(2,(x+1))=beta1;
    [pratio,p1] = p_ratioOB(table(3,x),table(1,x),gamma,table(2,x));
    table(3,(x+1))=p1;
    [pt_rat,pt1] = p_t_ratioOB(table(4,x),gamma,table(1,x),table(2,x));
    table(4,(x+1))=pt1;
    [RHOrat,RHO1] = Rho_ratioOB(table(5,x),gamma,table(1,x),table(2,x));
    table(5,(x+1))=RHO1;
    [T1, Trat] = T_ratioOB(gamma,table(1,x),table(6,x),table(2,x));
    table(6,(x+1))=T1;
    [theta,tan_theta] = Oblique_shock(table(1,x),gamma,table(2,x));
    table(7,(x))=theta;
end
for x=0:(Ext_R+Int_R-1)
    x=x+1;
```

```
    [pr] = precovery(gamma,table(1,x),table(2,x));
    table(8,x)=pr;
end
    [M1,Mrat] = Mach_ratioN(table(1,(Ext_R+Int_R+1)),gamma);
    table(1,(Ext_R+Int_R+2))=M1;
    table(2,Ext_R+Int_R+2)=90;
    [pratio,p1] =
p_ratioN(table(3,Ext_R+Int_R+1),table(1,(Ext_R+Int_R+1)),gamma);
    table(3,(Ext_R+Int_R+2))=p1;
    [pt_rat,pt1] =
p_t_ratioN(table(4,Ext_R+Int_R+1),gamma,table(1,Ext_R+Int_R+1));
    table(4,(Ext_R+Int_R+2))=pt1;
    [RHOrat,RHO1] =
Rho_ratioN(table(5,Ext_R+Int_R+1),gamma,table(1,Ext_R+Int_R+1));
    table(5,(Ext_R+Int_R+2))=RHO1;
    [T1, Trat] = T_ratioN(gamma,table(1,Ext_R+Int_R),table(6,Ext_R+Int_R+1));
    table(6,(Ext_R+Int_R+2))=T1;
    [pr] = precoveryN(gamma,table(1,(Ext_R+Int_R+1)));
    table(8,Ext_R+Int_R+1)=pr;


%intake capture area

%FUNÇÃO PARA CALCULAR POSIÇÕES DE RAMPA E LIP
sumLip=0;
yLip=D*tand(table(2,1));
ramp_angle=0;
d_sum=0;
h_sum=0;
dlip_sum=0;
hlip_sum=0;
Dint_sum=0;
if (Int_R > 0 && Ext_R > 0) %Mixed compression intake
    for x=0:(Ext_R+Int_R)
        x=x+1;
        if x<(Ext_R)
        ramp_angle=ramp_angle+table(7,x);
        %ramp_angle
        table(9,x)=(yLip-h_sum-
D*tand(table(2,x+1)+ramp_angle)+d_sum*tand(table(2,x+1)+ramp_angle))/(tand(ram
p_angle)-tand(table(2,x+1)+ramp_angle));
        %table(9,x)
        d_sum=d_sum+table(9,x);
        %d_sum
        h_sum=h_sum+(table(9,x)*tand(ramp_angle));
        %h_sum
        table(10,x)=h_sum;
        elseif x == Ext_R
        ramp_angle=ramp_angle+table(7,x);
        table(9,x)=(yLip-(sum(table(9,1:Ext_R-1))-D)*tand(table(2,Ext_R+1)-
ramp_angle)-h_sum)/(tand(table(2,Ext_R+1)-ramp_angle)+tand(ramp_angle));
        d_sum=d_sum+table(9,x);
        h_sum=h_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=h_sum;
        elseif x>Ext_R && x<Ext_R+Int_R
        D_int=d_sum-D;
        ramp_angle=ramp_angle-table(7,x);
        table(9,x)=(yLip+hlip_sum+(D_int-dlip_sum)*tand(ramp_angle-
table(2,x+1))-h_sum)/(tand(ramp_angle-table(2,x+1))-tand(ramp_angle));
        dlip_sum=dlip_sum+table(9,x);
        %ramp_angle=ramp_angle-table(7,x);
        hlip_sum=hlip_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=yLip+hlip_sum;
        elseif x==(Ext_R+Int_R)
        if Int_R==1
        D_int=d_sum-D;
        table(9,x)=D_int;
        else
```

```
        table(9,x)=D_int-dlip_sum;
        end
        ramp_angle=ramp_angle-table(7,x);
        hlip_sum=hlip_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=yLip+hlip_sum;
        end
        ramp_angle=ramp_angle-table(7,Ext_R+Int_R+1);
    end
elseif (Int_R==0 && Ext_R>0) %External Compression Intake
    for x=0:(Ext_R-1)
        x=x+1;
        if Ext_R==1
        ramp_angle=ramp_angle+table(7,x);
        table(9,x)=D;
        table(10,x)=D*tand(table(7,x));
        end
        if x<(Ext_R)
        ramp_angle=ramp_angle+table(7,x);
        table(9,x)=(yLip-(D-d_sum)*tand(table(2,x+1)+ramp_angle)-
h_sum)/(tand(ramp_angle)-tand(table(2,x+1)+ramp_angle));
        d_sum=d_sum+table(9,x);
        h_sum=h_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=h_sum;
        elseif x==(Ext_R)
        ramp_angle=ramp_angle+table(7,x);
        table(9,x)=D-d_sum;
        h_sum=h_sum+(table(9,x)*tand(ramp_angle));
        table(10,x)=h_sum;
        end
    end
    else
        end
    W=Cap_Area/yLip;
    %calculation of the necessary area at engine face to deliver air at
    %M=0.4
    a_throat=SpSound(gamma,R,table(6,Ext_R+Int_R+2));
    %Calculation of static pressure and stagnation pressure at engine face

p_engine=table(3,Ext_R+Int_R+2)+((table(5,Ext_R+Int_R+2)*((table(1,Ext_R+Int_R
+2)*a_throat)^2))/2)-((table(5,Ext_R+Int_R+2)*((0.4*a_throat)^2))/2);
    pt5 = pt_h(gamma,p_engine,0.4);
    p_rec_sub=pt5/table(4,Ext_R+Int_R+2);
    table(8,Ext_R+Int_R+2)=p_rec_sub;
    %PRESSURE RECOVERY FUNCTION
tpr=1;
    for x=0:(Ext_R+Int_R+1)
        x=x+1;
        tpr=table(8,x)*tpr;
    end
    %Removal of physically impossible data
    possible=isreal(table);
        if possible == 0
        table=NaN;
        tpr=NaN
        Lenght=NaN
    end

    y=0;
    if possible== 1
    while y<((Ext_R)+(Int_R)+1)
        y=y+1;
        if table(1,y+1)>table(1,y)
            table=NaN;
            tpr=NaN
            Lenght=NaN
            break
        end
    end
```

```
    end

    if tpr>=r_p_rec
        fprintf('The intake total pressure recovery (%f) is within military
stantards (%f)',tpr,r_p_rec)
    elseif tpr<r_p_rec
        fprintf('The intake total pressure recovery (%f) is outside military
standards (%f)',tpr,r_p_rec)
    elseif isnan(tpr)== 1
        fprintf('The chosen intake is not valid ')

    end

if isnan(tpr)==1
      Lenght=NaN;
      tpr=NaN;
      return
else
  %INSERT SIZING ALGORITHM HERE QUICKLY PLZ, REARRANGE IN ORDER TO FIX
  dimension_ext=zeros(2,Ext_R+3);
  dimension_ext(1,1)=0;
  dimension_ext(2,1)=0;
    for x=1:Ext_R
    dimension_ext(2,x+1)=table(10,x);
    dimension_ext(1,x+1)=sum(table(9,1:x));
    end
  dimension_int=zeros(2,Int_R+1);
  dimension_int(1,1)=D;
  dimension_int(2,1)=yLip;
  for x=2:Int_R
      dimension_int(1,x)=sum(table(9,Ext_R+1:x+Ext_R-1))+D;
      dimension_int(2,x)=table(10,x+Ext_R-1);
  end
  l=(sind(ramp_angle+90))*(sqrt(((dimension_ext(1,Ext_R+1)-
dimension_int(1,Int_R))^2)+((dimension_ext(2,Ext_R+1)-
dimension_int(2,Int_R))^2)));
  dimension_int(2,Int_R+1)=dimension_int(2,Int_R)+(l*sind(ramp_angle));
  dimension_int(1,Int_R+1)=dimension_int(1,Int_R)+(l*cosd(ramp_angle));
  %transition zone plotting dimensions
  d_trans=2*sqrt(((dimension_int(2,Int_R+1)-
dimension_ext(2,Ext_R+1))^2)+((dimension_int(1,Int_R+1)-
dimension_ext(1,Ext_R+1))^2));
  %Interruption for calculating dimension at engine face

A_engine=((table(1,Ext_R+Int_R+2)*a_throat)*(sqrt(((dimension_int(2,Int_R+1)-
dimension_ext(2,Ext_R+1))^2)+((dimension_int(1,Int_R+1)-
dimension_ext(1,Ext_R+1))^2))))/(0.4*a_throat);
  %Continuation of transition zone plotting dimensions
  dimension_ext(1,Ext_R+2)=d_trans*cosd(ramp_angle)+dimension_ext(1,Ext_R+1);
  dimension_ext(2,Ext_R+2)=d_trans*sind(ramp_angle)+dimension_ext(2,Ext_R+1);
  dimension_int(1,Int_R+2)=dimension_int(1,Int_R+1)+d_trans*cosd(ramp_angle);
  dimension_int(2,Int_R+2)=dimension_int(2,Int_R+1)+d_trans*sind(ramp_angle);
  dimension_ext(1,Ext_R+3)=dimension_int(1,Int_R+2);

dimension_ext(2,Ext_R+3)=dimension_ext(2,Ext_R+2)+((dimension_ext(1,Ext_R+3)-
dimension_ext(1,Ext_R+2))*tand(ramp_angle));
  %plotting dimensions of subsonic duct
  h_exp=A_engine-(dimension_int(2,Int_R+2)-dimension_ext(2,Ext_R+3));
  d_exp=h_exp/tand(6);
  dimension_ext(1,Ext_R+4)=dimension_ext(1,Ext_R+3)+d_exp;
  dimension_ext(2,Ext_R+4)=dimension_ext(2,Ext_R+3)-h_exp;
  dimension_int(1,Int_R+3)=dimension_ext(1,Ext_R+4);
  dimension_int(2,Int_R+3)=dimension_int(2,Int_R+2);

  %Relative Lenght function
  Lenght=dimension_ext(1,Ext_R+4)/A_engine
  %Relative Height
  H=max(dimension_int,[],2);
```

```
  Hmax=H(2,1);
  L=min(dimension_ext,[],2);
  Hmin=L(2,1);
  HDiff=(Hmax-Hmin)/A_engine;
  %plotting function
  figure
  hold on
  plot(dimension_ext(1,1:Ext_R+4),dimension_ext(2,1:Ext_R+4),'r',
'LineWidth',3)
  %introduction to plotting internal ramps here

plot(dimension_int(1,1:Int_R+3),dimension_int(2,1:Int_R+3),'r','LineWidth',3)
  %introduction to plotting shocks
  %plot(,'b--','LineWidth',2)
  xlabel('x (Lenght)')
  ylabel('y (Height)')
  grid on
  title('Bi-dimensional Intake and Shocks')
  daspect([1 1 1])
  set(gca,'Xtick',[],'YTick',[])

  end
```