CRANFIELD UNIVERSITY

YOUSEF AZABI

MODELLING AND AERODYNAMIC DESIGN OPTIMISATION OF THE TWIN-BOOM AEGIS UAV

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

AUTONOMOUS AND CYBER-PHYSICAL SYSTEMS

PhD Thesis
Academic Year: 2018 - 2019

Supervisor: Dr. AL SAVVARIS
Supervisor: Dr. T KIPOUROS
January 2019

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

AUTONOMOUS AND CYBER-PHYSICAL SYSTEMS

PhD Thesis

Academic Year 2018 - 2019

YOUSEF AZABI

MODELLING AND AERODYNAMIC DESIGN OPTIMISATION OF THE TWIN-BOOM AEGIS UAV

Supervisor: Dr. AL SAVVARIS
Supervisor: Dr. T KIPOUROS
January 2019

This thesis is submitted in partial fulfilment of the requirements for the degree of PhD

ABSTRACT

The aircraft industry gives considerable attention to computational optimisation tools in order to enhance the design process and product quality in terms of efficiency and performance, respectively. In reality, most real-world applications contain many complicating factors and constraints that affect system behaviour. Consequently, finding optimal solutions, or even only those viable for a given design problem, in an economical computational time is a difficult task, even with the availability of superfast computers. Thus, it is important to optimise the use of available computational resources.

This research project presents a method for using stochastic multi-objective optimisation approaches combined with Artificial Intelligence and Interactive Design techniques to support the decision-making process. The improved ability of the developed methods to accelerate the search while retaining all the useful information in the design space was the main area of work. Both the efficiency and reliability of the proposed methodology have been demonstrated through the aerodynamic design of the Aegis-UAV.

Initially, the optimisation platform Nimrod/O was deployed to enable the designer to manipulate and better understand different design scenarios. This happened before any commitment to a specific design architecture to allow for a wider exploration of the design space before a decision was made for a more detailed study of the problem. This had the potential to improve the quality of the product and reduce the design cycle time. The optimisation was performed using the Multi-Objective Tabu Search (MOTS) algorithm, chosen for its suitability for this type of complex aerodynamic design problem.

Prior to the optimisation process, a parametric study was performed using the Sweep Method (SM) to explore the design space and identify design limitations. Analysis and investigation of the SM results were used to help determine the formulation of the design problem. SM was chosen because it has been proven to be reliable, effective, and able to provide a large amount of structured information about the design problem to the decision maker (DM) at this stage.

Next, since most decisions of a DM in practical applications concern regions of the Pareto front, an interactive optimisation framework was proposed where the DM was involved with the optimisation process in real time. The framework used the Multi-Objective Particle Swarm Optimisation (MOPSO) algorithm for its suitability to this type of design problem. The results obtained confirmed the ability of the DM to use its preferences effectively, to steer the search to the Region of Interest (ROI) without degrading the aerodynamic performance of the optimised configurations. Even using only half the evaluations, the DM was able to obtain results similar to, or better than those obtained by the non-interactive use of MOTS and MOPSO. Furthermore, it was possible for the DM to stop the search at any iteration, which is not possible in non-interactive approaches even though the solutions do not converge or may be infeasible.

Finally an Artificial Neural Network (ANN) was introduced to guide the MOPSO algorithm in deciding whether the trial solution was worthy of full evaluation, or not. The results obtained showed the success of the ANN in recognising non-valid particles. Consequently, the solver avoided wasting computational efforts on non-worthwhile particles. The optimisation process provides particles that are more valid for almost the same computational time. Demonstrating the algorithm’s effectiveness was done by comparing results of the ANN-MOPSO solutions with those obtained by the other approaches for the same design problems.

In conclusion, future avenues of research have been identified and presented in the final chapter of the thesis.

Keywords:

Multi-objective optimisation, Nimrod/O, Interactive optimisation, Artificial Neural Network, Particle Swarm Optimisation, Tabu Search, Parallel Coordinates

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisors Dr Al Savvaris and Dr Timos Kipouros, for their guidance and support throughout this research programme. Their comprehensive knowledge, experience, helpful suggestions and comments have all contributed significantly to the success of this research project.

To my brothers and sisters, you have always supported me and have given me so much guidance throughout the years. I cannot thank you enough for helping me through this tough time.

Sincere thanks to my father, Mohmed Azabi who motivated me to become involved in academia, and to aim as high as possible in my career, and to my mother, Sada Ashour; my lovely wife Naima Abdalla, and my beautiful children, Ahmed, Arwa, Mayar, and Asil, for their prayers and support.

Also, I want to thank all the staff in the Autonomous Systems Research Group for everything they have done to make things easier for me throughout my academic years.

Lastly, heartfelt appreciation goes to my colleagues for their support throughout this project. You helped me overcome the many obstacles I faced and provided academic help when needed. Thank you for your motivation and patience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Acronyms

AIRC	= Aerospace Integration Research Center
AOA	= Angle of attack
ANN	= Artificial Neural Network
AVL	= Athena Vortex Lattice
BFGS	= Broyden Fletcher Goldfarb Shanno
BD	= Base design
CFD	= Computional Fluid Dynamics
CPU	= Central Processing Unit
DM	= Decision Maker
EA	= Evolutionary Algorithms
ES	= Evolutionary Strategies
ESDU	= Engineering Sciences Data Unit
FEM	= Finite Element Methods
GA	= Genetic Algorithmic
GN	= Gauss Newton
GPU	= Graphics Processing Unit
HPC	= High Performance Computing
LTM	= Long Term Momories
MALE	= Medium Altitude Long Endurance
MAVT	= Multi-Attribute Value Theory
MCDM	= Multi-Criteria Decision Making
MDO	= Multidisciplinary Optimisation
ML	= Machine Learning
MOO	= Multi-Objective Optimisation

MOPSO	= Multi-Objective Particle Swarm Optimisation
MOTS	= Multi-Objective Tabu Search
MTM	= Medium Term Momories
n/a	= Not applicable
NN	= Neural Network
NSGA-II	= Non-Dominated Sorting Genetic Algorithm II
PG	= Prandtl Glauert
ROI	= Region of Interest
SA	= Simulated Annealing
SM	= Sweep Method
S.M	= Static margin
SNOPT	= Sparse Nonlinear OPTimiser
STM	= Short Term Momories
TS	= Tabu Search
UAV	= Unmanned Aerial Vehicle
VLM	= Vortex Lattice Method

Greek Symbols

λ_f	= Inverted V-tail taper ratio
λ_{vt}	= Vertical tail taper ratio
λ_w	= Wing taper ratio
ϕ_t	= Inverted V-tail angle
Γ_tail	= Dihedral angle

Roman Symbols

AR_f	= Inverted V-tail aspect ratio
AR_{ht}	= Horizontal tail aspect ratio
AR_{vt}	= Vertical tail aspect ratio

b_w	= Wing span
c_1	= Attraction of the particle towards its own success
c_2	= Attraction of the particle towards the global best
C_{r_w}	= Wing root chord
C_{t_w}	= Wing tip chord
bvt	= Vertical tail span
bht	= Horizontal tail span
b_tail	= Inverted V-tail span
b_hp	= Span for the horizontal projection area for the inverted V-tail
b_vp	= Span for the vertical projection area for the inverted V-tail
C_D	= Drag coefficient
C_L	= Lift coefficient
Cm	= Pitch moment coefficient
Cht	= Horizontal tail chord
Cr_vt	= Vertical tail root
Ct_vt	= Vertical tail tip
Ct_tail	= Inverted V-tail tip
Cr_tail	= Inverted V-tail root
Cm_α	= Pitching moment slope
Cl_b	= Variation of rolling force coefficient with sideslip angle
Cn_b	= Variation of yawing force coefficient with sideslip angle
Cm_q	= Variation of pitching moment coefficient with pitch rate
Cn_r	= Variation of yawing force coefficient with yaw rate
$C^*_{L\_BD}$	= Base design lift coefficient
Cm^*_{BD}	= Base design pitching moment coefficient
E	= Endurance ratio

F_{TV} = Inverted V-tail volume

ht_angle = Horizontal tail rotation angle

L_{boom_BD} = Length of the boom for the base design

$L_{boom-optimised}$ = Length of the boom for the optimised UAV

L_t = Moment arm (from UAV xc.g to tail aerodynamic centre)

MAC_w = Mean aerodynamic chord of the wing

M_{boom_BD} = Mass of the boom for the base design

M_{f_BD} = Mass of the inverted V-tail for the base design

M_{ht_BD} = Mass of the horizontal tail for the base design

M_{unit} = Mass per unit area

M_{new_wing} = Mass for the optimised wing

M_{new_ht} = Mass for the optimised horizontal tail

M_{vt_BD} = Mass of the vertical tail for the base design

M_{w-BD} = Wing mass of the base design

p_i = Particle

$\vec{p}_{pbest}$ = Personal best solution of a given particle

$\vec{p}_{gbest}$ = solution of the best particle of the entire swarm

r_r = Uniform random weight for the personal best solution

r_2 = Uniform random weight for the global best solution

S_{f_BD} = Surface area of the inverted V-tail for the base design

$S_{f-optimised}$ = Surface area of the inverted V-tail for the optimised UAV

S_{ht} = Horizontal tail surface area

S_{ht_BD} = Surface area of the horizontal tail for the base design

$S_{ht-optimised}$ = Surface area of the horizontal tail for the optimised UAV

S_{tail} = Inverted V-tail surface area

S_{vt} = Vertical tail surface area

$\mathrm{S_{vt\_BD}}$ = Surface area of the vertical tail for the base design

$\mathrm{S_{vt-optimised}}$ = Surface area of the vertical tail for the optimised UAV

$\mathrm{S_{w}}$ = Wing surface area

$\mathrm{S_{W-BD}}$ = Wing surface area for the base design

$\mathrm{S_{w-optimised}}$ = Wing surface area for the optimised UAV

$\mathrm{UAV\ mass}$ = Optimised UAV total mass

$\mathrm{V_{ht}}$ = Horizontal tail volume

$\mathrm{V_{vt}}$ = Vertical tail volume

v = Cruise speed

$\mathrm{V_{st}}$ = Optimised UAV stall velocity

$\mathrm{V_{max}}$ = Optimised UAV maximum velocity

$\mathrm{V_{st}^{*}}$ = Base design stall velocity

$\mathrm{V_{max}^{*}}$ = Base design maximum velocity

x = Design variable

$\mathrm{x_{l}}$ = Design variable lower bound

$\mathrm{x_{u}}$ = Design variable upper bound

w = Inertial weight

1 Chapter - Introduction

1.1 Introduction

In this introductory chapter, the scope of the research and its aim are presented. The outcome of the research is identified by the significant contribution it makes to knowledge, and the findings which relate to the concepts presented.

The use of computational optimisation techniques in aerodynamic shape design processes has become conventional because it makes the design and development of a complex vehicle more flexible, quicker, and easier to understand [1]. These techniques have the potential to improve the efficiency of the design process and the quality of the design [2]. In addition, the rapid growth in computing technology offers design engineers the opportunity to use optimisation in design of real-world applications [3]. Thus, optimisation had started to become a key technology in the development of innovations for real-world applications. However, an increase of model complexity and the limitations of even supercomputers makes it challenging to find solutions in an acceptable computation time, computations could take weeks or months to produce a solution that may not even be feasible [4,5]. This poses a serious problem within multi-objective, multi-discipline, optimisation problems where a number of conflicting objectives must be optimised simultaneously to obtain highly optimal solutions [1,6].

Therefore, this research has paid particular attention to the development of an optimisation framework that can be used to accelerate the optimisation search while retaining the useful information contained in the design space for a multi-objective aerodynamic design problem. The main idea is to focus all the computational efforts on the region of interest and worthwhile solutions, rather than exploring and evaluating all possible design solutions in the design space.

1.2 Motivation

Even though, optimisation is the key to develop the best solution for a given design problem, the complexity of the optimisation problems in today's industry make it very challenging to produce efficient solutions in terms of high quality within a reasonable length of time.

Researchers in engineering design aim to address optimisation problems by trade-off; balancing the efficiency of the design process against the fidelity of the numerical model. To resolve this possible conflict, problem approximation and function approximation techniques can be used [7]. Problem approximation attempts to substitute the original problem with a less demanding computationally solvable problem. Due to the fluid nature of the conceptual design process, it is not recommended to use high fidelity analysis design tools at this stage of design as they can be costly [5,8,9]. In contrast, at the end of the preliminary design phase, it is possible to utilise more expensive and time-consuming software, since by then only a few designs are being studied [10]. On the other hand, the use of function approximation techniques through the use of surrogate objective functions may degrade the accuracy of the results [11]**.**

However, all real-world design problems require a high degree of accuracy, which is obtained only by evaluation of its objective function [11], which can be a lengthy process. This problem becomes worse when the number of unfeasible trial solutions are more than the feasible, which is often the case in aerodynamic shape design optimisation problems [12,13].

For real-world problems, the Decision Maker (DM) is interested only in a sub-region of the objective space because the benefits from trade-off solutions that lie outside the region of interest is very small compared with the computational cost and the efforts required of the DM in analysing unnecessary information [14]. The solution to such a problem is to incorporate the designer within the optimisation loop [15–18]. Furthermore, it is not suitable to spend a long time evaluating non-worthwhile solutions. Engineering design problems invariably require a combination process that accelerates the optimisation process while retaining all the useful information of the design space. This process should produce more optimal solutions in less computational time. However, the accuracy of such solutions could not be achieved without using the real objective function [11].

For these reasons, both the interactive optimisation - where the DM is involved in the optimisation process - and an Artificial Neural Network (ANN) is used to improve the efficiency of the design process in terms of computational time and quality of the solutions. Before that, significant work is performed by automating the optimisation for the design process with greatly increased computing power by getting access to the Cluster/Grid using the Nimrod/O tool. Furthermore, the existing Aegis UAV platform was used with the motivation of developing an approach that can be used for any aircraft, to provide support during the development of the methodology using the available database.

1.3 Research scope

The aircraft industry has given considerable attention to computational optimisation techniques which improve both the efficiency of the design process and product quality in terms of efficiency and performance respectively. However, due to the increasing complexity of real-world problems it is becoming more difficult to find feasible optimisation solutions in an acceptable computer time. This research investigates the implementation of advanced optimisation techniques to shorten the path to optimal solutions by adding machine learning and the human feature of decision making to the optimisation process.

1.4 Research aim and objectives

The aim of this work is modelling and aerodynamic design optimisation of the Twin-boom Aegis UAV using a low fidelity method and multi-objective optimisation.

In support of the aim of this study, the following objectives are proposed:

- Carry out an intensive literature review of previous research and work in this area.

- Define and assess a strategy to formulate and construct, including constraints, the multi-objective aerodynamic shape design optimisation problem.

- Propose a methodology to enhance design optimisation by deriving an automatic optimisation process that takes into consideration the reduction of the computational time, and quickly provides information regarding the design problem.

- Investigate and demonstrate the benefits of using the interactive approach in the aerodynamic shape design optimisation process by incorporating the decision maker in that process to accelerate the optimisation search without degrading the solutions.

- Introduce an ANN within the optimisation process to increase convergence of the optimiser and obtain highly optimal solutions.

1.5 The main contribution of the present work

The effective use of optimisation methods for aerodynamic design has become essential [19] and supercomputers and powerful optimisation techniques have been developed to handle various real-world design problems. However, increasing model complexity and the limitation of even supercomputers makes it challenging to find solutions in an acceptable computation time; computations could take weeks or months to produce a solution that may not even be feasible [4,5]. The majority of current optimisations search the whole design space, which requires a very large number of evaluations to explore the design space evenly [20], and that leads to slow convergence on the required solutions. In fact, the DM in most practical application is interested only in a sub-region of the objective space [21,22]. Moreover, because aerodynamic shape design problems tend to be highly constrained [12,13] and there are many more invalid solutions than valid [13] it is not efficient to spend a long time evaluating non-worthwhile solutions [11]. Thus, particular attention has been paid in this work to using an advanced optimisation technique to accelerate the optimisation search while retaining all the useful information of the design space.

The research makes novel contributions in the field of aerodynamic shape design optimisation by developing a methodology that uses stochastic multi-objective optimisation combined with Artificial Intelligence and Interactive techniques to support decision-making in the optimisation process.

- The work commenced with the development of a new framework that combined the Nimrod/O tool with the flow solver, Athena Vortex Lattice (AVL). Within such a framework, it was easy to derive an automatic optimisation process taking into consideration all the combinations of design variables using the computational power made available by gaining access to the Cluster/Grid. This framework allowed efficient exploration of the design space, and the discovery of existing relations between the objectives and the design variables. Then ranges of efficient scenarios that could be used in the aerodynamic shape design problem were performed to provide a better understanding of the design problem. Knowing the relations that exist between the objectives and design variables gives a better understanding of the design problem so the scenarios investigated could be limited, which made the process much more efficient.
- The research also made a significant contribution by investigating the use of interactive optimisation in the aerodynamic shape design process for the whole aircraft, focusing on improving the efficiency of the solutions, in terms of computational cost and optimality. This makes it possible to concentrate computations on exploring only the interesting areas of the design space to identify just the preferred designs for the intended applications.
- An active area of research in aeronautical engineering is the development of screening criteria to determine which candidate solutions are worthy of full evaluation. The motivation is to improve the efficiency of the design process with high optimality solutions. In engineering the ANN is an attractive tool due to its remarkable characteristics of learning to cope with imprecise and uncertain information. In this research, the neural network

is used to guide the optimisation algorithm by deciding whether the candidate solutions are worthy of full evaluation, or not, in the case of aerodynamic shape design problems. In fact, such design problems require a high degree of accuracy, which is obtained only by evaluation of the objective functions. In addition, the research has made a significant contribution in its comparison of solving the same problem by interactive and non-interactive optimisation approaches.

To date, a review of published work has shown that to reduce the computational time there have been several attempts to use the neural network as a surrogate model instead of the objective functions [23,24]. However, using surrogate objective functions may degrade the accuracy of the results [11]. The literature review found only one publication, where the neural network was used to guide the optimisation algorithm - concerning an airfoil optimisation problem - by deciding whether the trial solution is worthy of full evaluation or not, rather than using the neural network to model the objective function [11]. On the other hand, attempts to use interactive optimisation include either testing of benchmarking functions [16–18] or optimisation of wing airfoil [15].

Finally, this research provides support for decision-making for a design optimisation process that will benefit the entire community of aircraft researchers and designers.

1.6 Publications

Peer reviewed journals:

1. Yousef Azabi, A. Savvaris, T. Kipouros, Initial Investigation of Aerodynamic Shape Design Optimisation for the Aegis UAV, Transportation Research Procedia, Elsevier, 29 (2018) 12–22.
2. Yousef Azabi, AL Savvaris, Timoleon Kipouros, The Interactive Design Approach for Aerodynamic Shape Design Optimisation of the Aegis UAV. Aerospace 6(4), April 2019

3. Yousef Azabi, AL Savvaris, Timoleon Kipouros, Artificial Intelligence to Enhance Aerodynamic Shape Optimisation of the Aegis UAV. Machine Learning and Knowledge Extraction 1(2), April 2019

Conference paper:

1. Yousef Azabi, A. Savvaris, T. Kipouros, Initial Investigation of Aerodynamic Shape Design Optimisation for the Aegis UAV, CEAS 2017 Conference, 16-20 October, Bucharest

1.7 Thesis structure

The rest of the thesis is organised as follows. Chapter 2 reviews relevant literature, including the design principles at different phases of the design process, and how various elements within the optimisation influence the efficiency of the design, especially for complex design problem. A particular focus is given to Parallel Coordinates as a necessary visualisation technique to better understand the distribution and achieved trends for large amounts of data. Chapter 3 analyses and describes the application of advanced optimisation techniques in the field of aerodynamic shape optimisation. Three types of optimisation techniques, namely automated optimisation using the Nimrod/O tool, interactive optimisation, and a machine learning algorithm. The advantages and disadvantages of using these well-developed algorithms are discussed. The research methodology with details of each technique are presented in Chapter 4. Chapter 5 introduces the problem formulation for the case study used in this research to demonstrate the effectiveness of the proposed methodology. The formulation considers six design scenarios for the Aegis UAV. The results, discussion, and conclusions for the automated optimisation, interactive optimisation, ANN algorithm, respectively are presented in Chapters 6 to 8. Finally, the conclusions and recommendations for the future work are presented in Chapter 9.

2 Chapter - Elements for Design and Optimisation

This chapter provides the necessary background needed to understand the technical terms, theory and discussions presented in this thesis. It introduces the design principles of unmanned aerial vehicles (UAVs) and differences in the requirements for different design phases. According to [25], the level of fidelity that the models should provide is mainly determined by the development phase which the Multidisciplinary Optimisation (MDO) process aims to enhance. In addition, a powerful framework that supports and speeds up the development process is an essential part of the design optimisation process. While these elements influence the efficiency of the optimisation process, the optimiser algorithm itself and the tool used to improve the visualization of data to provide the necessary support for decision-making are essential too.

2.1 UAV design

The design of UAVs has expanded substantially over the past two decades [26]. Although UAVs were initially introduced for military applications, they have now become vital for many civilian uses [27]. Different UAV applications required different UAV configurations, see Figure 2-1. Regardless of the specific UAV configuration, engineers are required to design UAVs that can successfully withstand a wide range of flight conditions, are suitable for long survey periods and have the advantage of low cost. This is made possible by using low fidelity code during the conceptual design phase, which can accelerate the design procedure and enable the design engineers to manipulate large numbers of parameters. Due to the fluid nature of the conceptual design process, it is not recommended to use high fidelity analysis design tools such as Computational Fluid Dynamics (CFD) and Finite Element Methods (FEM) at this stage as they can be unnecessarily costly [5,8]. What is required is a tool that strikes a balance between sufficient accuracy and computational cost. This tool should contain considerable data concerning basic aircraft geometry to minimise the time required for the tens of thousands of necessary computations [9]. At the end of the preliminary design phase, it is possible to utilise more costly and time-consuming software since by then only one design case is being studied [10]. Ideally, an efficient design configuration of limited cost and less computation time would be achieved by coupling an aerodynamic design code with an optimisation algorithm [28,29].

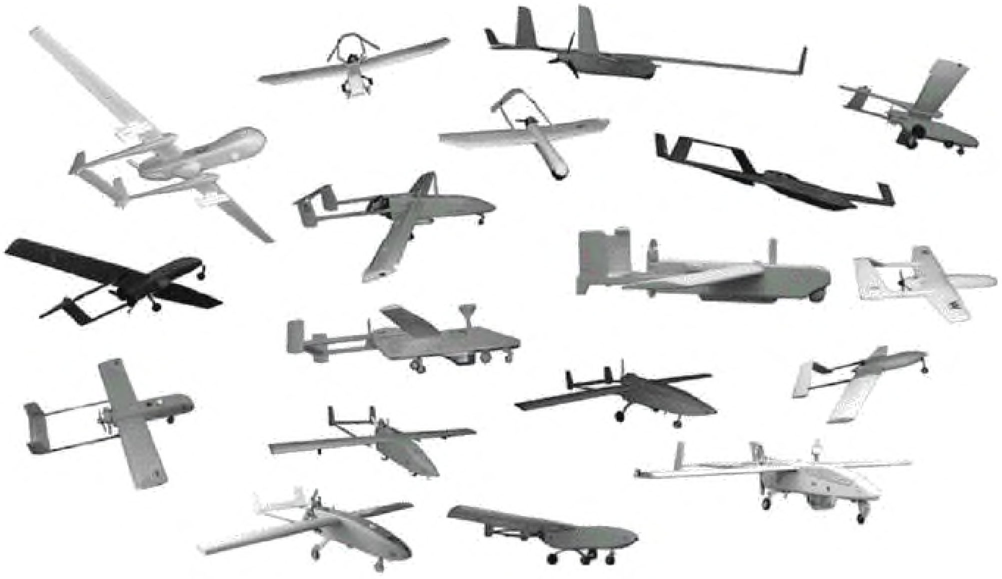


Figure 2-1: Conventional UAV; twin-boom pusher UAV with different tail arrangements [30]

On the other hand, even though the aircraft optimisation process is a function of several disciplines including; aerodynamics, structural engineering, control theory and aeroelasticity, the design process invariably starts with a shape to satisfy the aerodynamic constraints, and this is followed by adapting the shape to meet the requirements of the other disciplines [31]. Typically, each discipline contains more than one objective, and these objectives commonly conflict with each other. The solution to such a problem is complex and requires a slightly different approach to single objective optimisation problems; there is no longer a single best solution, rather a set of trade-off solutions [32]. It is a process of optimising simultaneously and systematically a collection of objective functions [33], and this is the focus of the current research. This requires the formulation and solution of multi-objective optimisation problems.

A consequence of this is that computational time becomes a significant factor with the final design a trade-off. It is generally recommended that a Pareto front should be used to find the optimal compromise solution [34–36]. Such an approach requires a large number of evaluations to achieve a well-distributed Pareto front that explores the whole design space efficiently [3]. For such an optimisation, computational time and post-data-analysis complexity are the main issues.

2.1.1 Conceptual and preliminary design

In general, the aircraft design process can be separated into three stages; the conceptual design phase, the preliminary design phase, and detail design phase [37]. The initial idea is usually established in the conceptual design phase, such as mission, sizing, cruise speed, gross weight, and wing-tail arrangements. The basic equations of flight performance and configuration arrangement to achieve the recommended mission are then determined. It is the design stage where every detail in the design space is investigated. Since it is possible very little is known a priori about the performance of the proposed aircraft, the design is usually re-analysed and sized many times.

Once the aircraft configuration and propulsion systems have been estimated, the next step is the preliminary design phase. The main activity in this stage is to ensure a proper fit between different aircraft parts and disciplines, i.e., the different specialists will analysis the aircraft from their own points of view. At the end of this stage, a prototype may be developed. In [5], Jameson stated that due to the fluid nature of the conceptual design process and, as explained above, it is not recommended to use a sophisticated (high fidelity) analysis design tools such as CFD and FEM at this stage. For example, an analysis using a Navier-Stokes code might be very accurate [38] but takes a long time to analyse even a single case. Thus, this kind of analysis tool might not be suitable for the conceptual design phase, where the design process is repeated many times with several configurations being evaluated in order to reach a compromise on the final design requirements. On the other hand, at the end of the preliminary design phase when, also as described above, only one design case is being studied, it is possible to utilize more costly and time-consuming software such as CFD and FEM [10].

In [39] several computational methods are compared to check their ability to compute the aerodynamic forces and moments on a small UAV. The majority of tools that have been used depends either on potential flow theory or Navier-Stokes (N-S) equations [5,8,38]. It was found that the N-S solver required around eight hours to run per single case while the potential flow solver needed only seconds to run. In addition, the potential flow solver was better at providing a large amount of data in a short time. Thus, if time (i.e., cost) is not an important consideration N-S solvers can provide data that are more accurate.

2.1.2 The effect of wing and tail sizing on UAV performance

Nowadays, UAVs are readily available with various configurations, such as conventional wing-tail, canard, tandem wings, and flying wings [40]. Several studies have compared their aerodynamic efficiency, structural sizing, and stability requirements, and it was found that the conventional wing-tail arrangement always had better performance [41].

Figure 2-2 shows the different configuration for conventional wing-tail arrangements where the wing and tail both play critical roles in the UAV performance. The wing is designed primarily to produce lift while the tail is mainly to stabilize the UAV but also to generate lift [42]. Consequently, optimisation of the UAV with a conventional configuration should account for the performance of both wing and tail to obtain optimal shape, since they both directly affect the aerodynamic and stability performance [29,30]. For the design of long-endurance UAVs, a significant task for the designer is to reduce the fuel-burn, which is achieved mainly by keeping the aerodynamic drag of the UAV during flight as low as possible [40]. Thus, optimal UAV configurations will result in maximum UAV endurance. Gudmundsson defined endurance as the time period that a UAV will remain airborne under a particular flight condition while consuming a certain amount of fuel [43].

The endurance is defined mathematically using Breguet equations [30,43]. It shows that for propeller-driven aircraft, endurance depends on $\left(C_L^{1.5}/C_D\right)$ [27,44], which is improved by maximizing lift and minimizing drag. The total drag of a UAV flying at low speed (around M= 0.13) results from both induced and parasite drag [42]. Parasite drag (profile drag and friction drag) is usually a function of Reynold number and flying speed. On the other side, induced drag is purely a pressure drag caused by the wing tip vortices, which generate a perturbation in the flow field over the wing which, in turn, perturbs the pressure distribution over the wing yielding an increase in drag [26,38].

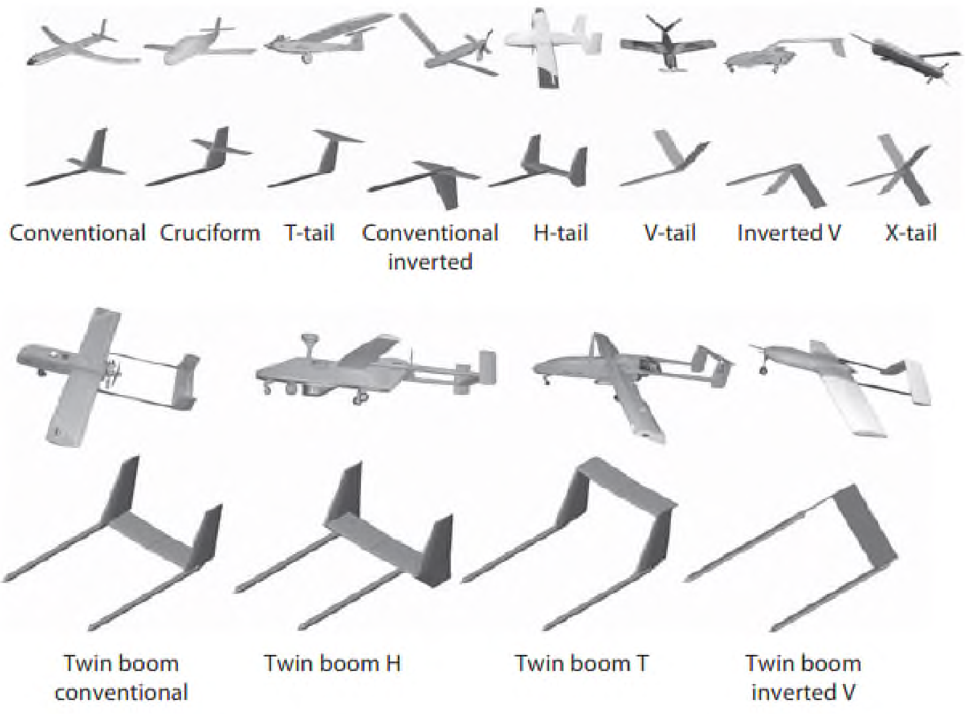


Figure 2-2: Conventional wing-tail arrangements: single attach point tail configurations and twin-boom tail configurations

Obviously, to obtain a feasible solution for the design of the aerodynamic shape of the aircraft, the design process should include wing and tail parameters simultaneously [29]. The reason is the interaction that exists between the aerodynamic efficiency and stability characteristics [43,45,46].

2.2 Computational design optimisation

Creating the best aircraft is the ambition of the aircraft designer, which means using the tools best capable of finding the optimal combination of design features. Using computational simulations has enabled designers to scan many alternative designs relatively quickly and has proved to be good practice. However, the possibility of successfully performing complete evaluations for all possible design scenarios is limited, which means the possibility of finding true optimal solutions is weak. One way to ensure the true optimal solution is by automating the optimisation process using efficient computational design optimisation methods [47], executing intensive optimisation processes to produce discrete design scenarios for various combinations of the design variable. Hence, the aircraft industry has given considerable attention to computational optimisation tools to enhance the efficiency of the design process and product quality and performance

Using trade-off techniques in engineering and scientific research has a long history. The Wright Brothers in their laboratory work used a kind of balance of technical solutions when performing a wing geometry parametric study. “*There is a lively branch of applied mathematics, which will here be called optimisation and which attempts to choose the variables in a design process so as formally to achieve the best value of some performance index while not violating any of the associated conditions or constraints*” [48]. Figure 2-3 shows a general flowchart describing the flow of an optimisation design process.

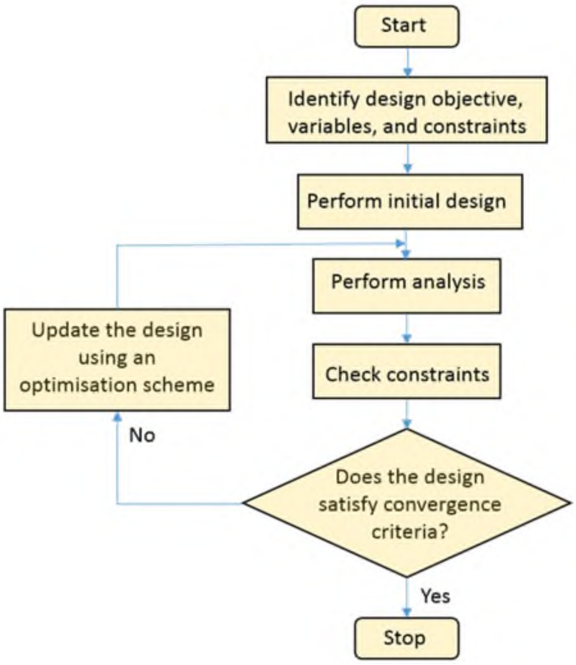

Start
Identify design objective, variables, and constraints
Perform initial design
Perform analysis
Check constraints
Does the design satisfy convergence criteria?
Update the design using an optimisation scheme
No
Yes
Stop

Figure 2-3: Design optimisation process to describe the general procedure of any optimisation design problem [49]

The main purpose of using optimisation techniques is to help designers to achieve the best design under certain constraints [50]. Aircraft design is a complex process since it involves thousands of parameters that must be addressed simultaneously during the design process. The only way to consider all the requirements efficiently is by using optimisation techniques to perform the necessary repetitive calculations to achieve an acceptable compromise design [10].

2.2.1 Requirements for optimisation algorithms

One of the essential characteristics for a design framework is the ability to provide information to the decision maker (DM) at the early stages of the design, so he/she can make informed decisions with regard to the design. However, evaluations of the objectives of a design problem are invariably computationally challenging, especially with increased model complexity and there are limits, even with supercomputers, to finding solutions in an acceptable computational time. Thus, it is necessary to select a computational framework that has some characteristics which lead to simplifying the design optimisation problem [51], such as;

- Choose an optimisation algorithm that requires fewer function evaluations to find an optimum answer.
- Choose an optimisation algorithm that can perform the required computations in parallel to minimise the computational time needed to find an optimum answer.
- Choose an optimisation algorithm from tried and tested packages such as Nimrod/O. This tool greatly simplifies the procedure of defining and integrating many codes together using a text plan file.

The above goals may conflict, but it is essential to have an optimisation framework that is able to handle a wide variety of real-world problems. The optimisation algorithms which are in use today can be classified into two different categories; deterministic and stochastic [52].

Deterministic algorithms, such as the Gradient Method have definite rules for moving from one solution to another, and strongly depend on the initial point. The classical gradient methods have been used for a relatively long time and have been effectively applied to many engineering design problems. It is considered as a local method in that a locally optimal solution will generally be obtained at convergence. The gradient algorithms depend mainly on the evaluation of either first order or first and second order gradient information for the objective and constraints functions with respect to the design variables [53], where the local optimal solution is found by moving with respect to the function gradient values [54].

The work of Hicks et al. [55] to optimise the airfoil section by coupling a numerical optimisation method based on the method of feasible directions with an aerodynamic analysis code is considered as the first practical application for aerodynamic shape optimisation [4]. Later, the work of Hicks et al. was extended to the design of three-dimensional wing geometry, combining an aerodynamic code capable of fully simulating potential inviscid flow with a conjugate gradient optimisation algorithm based on the methods of feasible direction. Since then, aerodynamic shape optimisation has been extensively explored using deterministic methods.

In [29], Chen et al. proposed using the Sparse Nonlinear OPTimiser algorithm (SNOPT) to increase efficiency when solving the aerodynamic shape optimisation problem for an aircraft using wing and tail design variables. The SNOPT uses a gradient-based optimiser combined with adjoint gradient evolution. The authors argued that because of the computational cost of CFD solutions, it is essential to use an algorithm that requires fewer evaluations to explore the design space while obtaining highly optimal solutions. The study aimed at showing the benefits of considering the wing and tail surfaces simultaneously, including trim constraints in aerodynamic shape design optimisation without any restrictions on the structure. It is a single objective optimisation problem. For example, the optimised wing when applying trim constraint using the tail rotation angle achieved a drag value 0.78% higher than the untrimmed optimised wing, but still 2.79% lower than the drag value of the base design. The optimised wing with trim constraints still has around a 4.1% lower drag value than the trimmed base design using tail rotation only. On the other hand, the optimised configuration obtained by including wing and tail design variables without tail rotation, achieved drag reductions of 0.29% and 3.82%, respectively compared to the optimised configurations using wing design variables only, and the base design. However, there was an increase in the absolute pitching moment coefficents from 0.041 (base design) to 0.078 (using wing only) and 0.133 (using wing-tail). Finally, comparing the optimisation results achieved using the wing and tail design variables with optimisation results obtained using wing design variables only (both cases trimmed by including tail rotation angle), showed a drag reduction of 0.11%.

Unfortunately, gradient-based methods are not always possible to use. According to [56], classical gradient-based algorithms can be used only for continuous, differentiable objective functions. However, if the optimisation problem includes non-differentiable functions or is a multi-objective optimisation problem as the case in the real-world design problems, a different optimisation technique that has stochastic characteristics is required. Stochastic algorithms, such as Evolutionary Strategies (ES) methods, Tabu Search (TS), Simulated Annealing (SA), and Multi-Objective Particle Swarm Optimisation (MOPSO), depend on probabilistic transition rules. These algorithms are known also as Meta-Heuristic Methods, since they use procedures which attempt to avoid being stopped or trapped in a local optimum [57]. Stochastic algorithms are considered new when compared with deterministic methods, but have been used successfully in diverse types of engineering optimisation problems.

Evolutionary Algorithms (EA) is one of a set of optimisation techniques that have been used successfully in many complex applications in the last decades. It is a population-based meta-heuristic, originally inspired by aspects of natural evolution [54]. The Genetic Algorithmic (GA) is a well-known EA [58]. In these methods, the design variables, which represent a population of individuals progress over the search space and create offspring by use of mutation, reproduction, and selection mechanisms [54]. Each iteration of the algorithm includes a competitive selection that excludes poor solutions.

A disadvantage of EAs is that they involve a large number of functions when evaluating an optimal solution and that results in slow convergence [29], a particular problem when EAs are used for small design spaces [59]. Nevertheless EAs have a good reputation regarding the estimation of the global minimum [36,60], and hence, many researchers use EAs [59,61–63].

A heuristic approach related to GA is Simulated Annealing (SA). In SA, each point in the search space is analogous to a state of some physical process, where the fitness function is analogous to an “internal energy” of the system. During this process, the aim is to minimise the internal energy of the system [64][65]. The SA is usually implemented to search optimal solution in small design space [66]. in [67], the SA performance is compared to the GA to solve 8-Queens chess problems. It has been found that the SA was better than GA in terms in less number of steps that required to find the solution.

Another typical heuristic approach is the Tabu Search (TS) algorithm [64], which is considered as one of the most efficient heuristic methods in the sense of its ability to find a good quality solution in an appropriate computational time [64][68]. TS has rapidly expanded in the last years as one of the efficient meta-heuristic techniques dealing with complex real-world problems [69]. Ghisu el al. have argued that researchers previously paid little attention towards the TS algorithm compared to other multi-objective meta-heuristics, but recently TS has gained attention for its implementation in aerodynamic shape design problems [70]. Motivated by the need for implementation of more advanced optimisation techniques in the field of aerodynamic shape optimisation, Jaeggi et al. presented a strategy to develop a single objective TS algorithm to cope with a multi-objective design problem [68]. The published paper included details of the new algorithm, such as search procedure and memory techniques used for intensification and diversification. The new approach, Multi-Objective Tabu Search (MOTS), proved to be very efficient when compared with the well-known GA NSGA-II.

Another technique that has emerged is Particle Swarm Optimisation (PSO). PSO is a stochastic population-based algorithm introduced in 1995 by Kennedy and Eberhart [71]. PSO is similar to GA in a sense but employs different strategies and computational efforts [72,73]. It is inspired by the collaborative behaviour of a school of fish or a flock of birds in search of food [74–76]. Even though, the PSO was initially implemented by its authors for neural network training [71], it has become a very popular global optimiser. Since the PSO is a relatively simple concept and computational inexpensive in terms of computer memory requirements and speed relative to other population techniques, many researchers have extended the algorithm to handle multi-objective optimisation problems [13,75]. The majority of MOPSO algorithms share the same basic approach. A swarm of a certain number is initialized randomly and that number remains constant until the end of the run.

2.2.2 The concepts of Pareto equivalence and domination

Design optimisation continues to be a broad research topic, and many of its applications find comprehensive use in engineering and science, both in industry and academia [77]. In real-world problems, optimisation will include more than one local solution as well as the global minimum or maximum. Engineers are usually interested in obtaining the globally optimal solution since it corresponds to the maximum or minimum values of the objective function. On the other hand, when the optimisation design problem contains more than one objective function, designers want more than just one global optimum that corresponds to only one objective function, since maximising the design for only one objective situation may not be best for the other objectives. This kind of design problem results in many non-dominated solutions known as Pareto optimal solutions, which present different trade-offs between the objectives. A Franco-Italian economist, V. Pareto (1848-1923) developed the theory of Pareto optimality for multi-objective optimisation problems [49]. Originally, the term Pareto goes back to text written by Pareto [78];

"We will begin by defining a term which is desirable to use in order to avoid prolixity. We will say that the members of a collectivity enjoy maximum ophelimity in a certain position when it is impossible to find a way of moving from that position very slightly in such a manner that the ophelimity enjoyed by each of the individuals of that collectivity increases or decreases. That is to say, any small displacement in departing from that position necessarily has the effect of increasing the ophelimity which certain individuals enjoy, and decreasing that which others enjoy, of being agreeable to some and disagreeable to others."

When considering Pareto optimal solutions, all the objectives are taken into account. Figure 2-4 represents a comparison between dominated and non-dominated (Pareto-optimal) solutions, for a minimisation problem.

As mentioned previously, a multi-objective design problem will be solved by considering a set of solutions rather than a single solution. The Pareto front solutions shown in Figure 2-4 are optimal because the values of an one object cannot be improved without worsening the condition of the other [65].

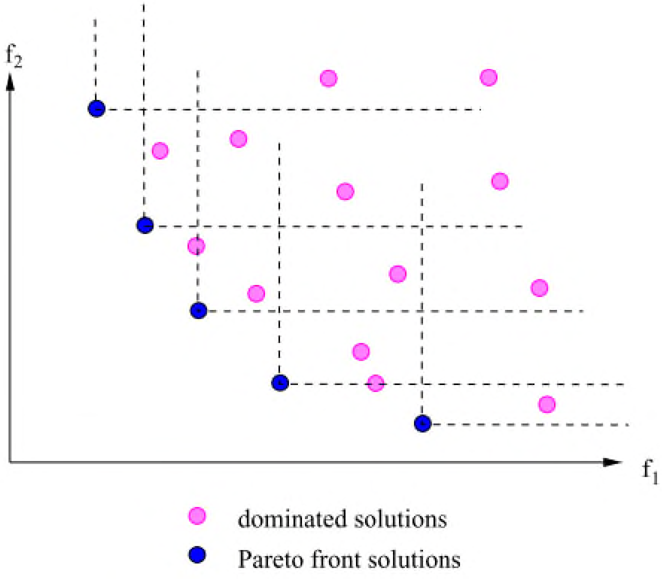


Figure 2-4: Pareto front compared to the dominated solutions with respect to two objective functions [75]

However, the Pareto front may include strong, inferior, and weak Pareto solutions. Figure 2-5 shows an example of Pareto optimal solutions for two objective functions where there are optimal compromise solutions and inferior Pareto solutions, which are ideal for one objective. Thus, the inferior solutions can be ignored and obtain the front consisting of optimal compromise solutions [79,80].

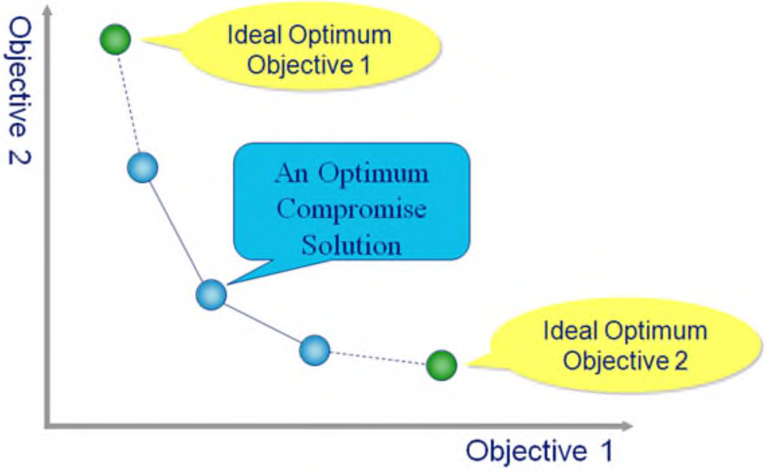

Objective 2
Ideal Optimum Objective 1
An Optimum Compromise Solution
Ideal Optimum Objective 2
Objective 1

Figure 2-5: Pareto equivalence and domination, optimal compromise solution and inferior Pareto solutions

2.2.3 Requirements of multi-objective optimisation for engineering design

Multi-Objective Optimisation (MOO) is the area within Multi-Criteria Decision Making (MCDM), which simultaneously optimises more than one objective function. MOO is used with problems where no single solution exists that simultaneously optimises all of the objectives, where objectives are conflicting and there are a number of non-dominated solutions. The main task of the optimiser in multi-objective design problems is to provide an efficient presentation of the non-dominated solutions to the DM. Figure 2-6 illustrates different aspects of solution set quality that can be recognised through the optimisation results for a MOO problem [14]. In real-world problems, the DM is usually interested only in a set or sub-set of solutions of the objective space. Focusing on this region will improve the efficiency of the optimisation process and reduce the amount of unnecessary information for the DM to analyse [20,81].

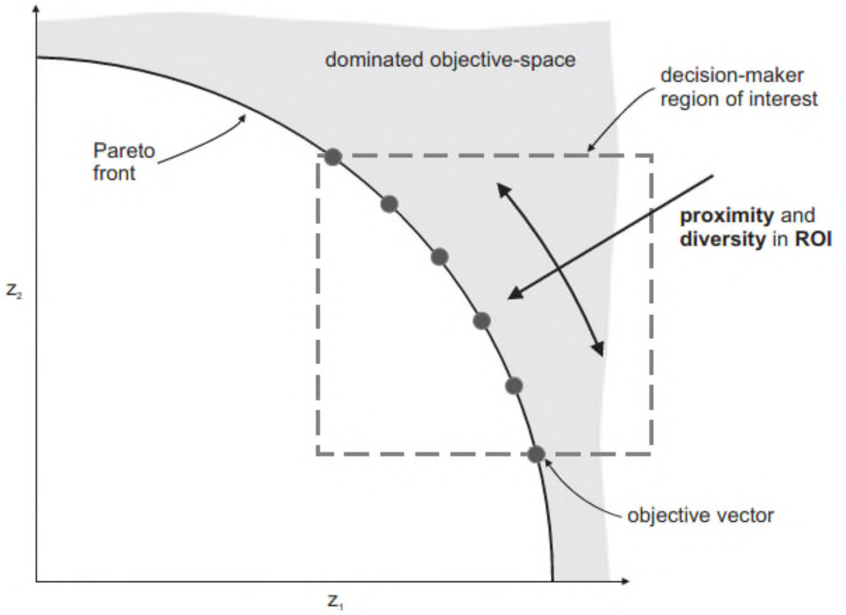

dominated objective-space
decision-maker region of interest
Pareto front
proximity and diversity in ROI
objective vector
z_2
z_1

Figure 2-6: The ideal solution to a multi-objective optimisation design problem and the ROI when there are compromise requirements for two objectives [14]**.**

2.2.4 Efficiency and robustness in multi-objective optimisation

Optimisation algorithms use the results from numerical analyses and simulations, herein called “evaluations”, to guide the search for an optimal design. In conventional parameter optimisation, an algorithm's efficiency is measured in terms of the total number of evaluations required to find the optimal design or a design at a specified performance level. In Pareto optimisation, efficiency is similarly judged by the number of evaluations needed to find a suitably accurate approximation of the Pareto front [35].

Using fewer evaluations to find the Pareto front is very important because often each evaluation can require a significant amount of CPU time. For example, in the case of the aerodynamic optimisation problem, the optimiser has to call the aerodynamic code each time to evaluate the objective function. So, reducing the total number of evaluations needed has a significant impact on the time required to find an optimised design or Pareto front [35]. Therefore, an intensive effort has been made in this research to reduce the number of evaluations while obtaining good optimality solutions. Two algorithms with different characteristics have been used in this research; Multi-Objective Tabu Search (MOTS) and Multi-Objective Particle Swarm Optimisation (MOPSO). Each has a different procedure and setting to achieve high performance.

The number of regions and evaluations are the main parameters used to identify the efficiency of the MOTS algorithm while the best combination of particles and iteration are the main parameters in the case of the MOPSO algorithm. To select an optimal setting for the MOTS algorithm, several runs with different sets of regions and evaluations were investigated. The number of evaluations had a significant impact on the computational time [35], while increasing the number of regions forced the algorithm to explore more areas of the design space [82]. On the other hand, the MOPSO settings were investigated by using different combinations of iterations and numbers of particles.

2.3 Data analysis

Aerospace design optimisation is a complex process, where the DM is required to consider numerous and multi-disciplinary criteria. Simulations regularly produce multi-dimensional data. To take the correct decision, it is essential to understand the data distribution and the achieved trends, which present the correlations that exist between the different design parameters.

Conventional visualisation techniques such as scatter plots are limited to a maximum of three dimensional data. In [14,83] the authors stated that data visualized in low-dimensional space using scatter plots is applicable only when the objective space involved trade-off for two or three parameters, where each parameter can be represented directly on an axis. In addition, it was argued that Parallel Coordinate techniques are an efficient way for representing relations when there are more than three parameters in the design space, as is the case with multi-objective design optimisation problems which often require simultaneous optimisation against many conflicting objectives. For such optimisation design problems, no single solution exists that can satisfy the objectives, instead non-dominated solutions are presented which required external trade-offs to select the preferred solution [6,21]. In such a design problem, the decision maker requires an additional tool to simplify the trade-offs and help the designer take the right decision within the computational optimisation process [84], such as Parallel Coordinates.

Parallel Coordinates are not limited to optimisation problems, and have proved to have strong capabilities as a visualization aid and have been used successfully in different applications to assist with the visualization of large amounts of data. For example, in [85] Parallel Coordinates were used to provide insight into multi-dimensional data, where the data size was tens of millions of points for each time set considered, and each different attribute. This work intends to enhance the visualization process where the interactive exploration of large sets of data becomes infeasible. The obtained results in terms of the quality of the visualization and interactive response time were highly encouraging.

2.3.1 Visualisation in computational design

Real-world design problems require the designer to invest in many computational tools to improve and speed-up the engineering design process [86,87]. The aerodynamic design problem is one of the fields where high performance design tools can have a significant impact on the design process. Successful application of these tools can enhance the techniques used by the DM and improve the optimising process - known as MCDM [88].

Naturally, many real-world design problems consist of multiple conflicting requirements, and a DM will struggle to find or improve on a compromise solution. The process needs a physical tool that gives a better explanation for the characteristic of the optimisation process and assists with the post-optimisation data analysis. To overcome these issues, Parallel Coordinate techniques for viewing multi-dimensional data are used [84,89]. This presents the solution in a multi-dimensional parallel coordinate system, where parallel coordinates represent each variable. It allows observation of relations between design variables and trade-offs between objectives, and monitors the evolutionary process [90–92].

Several visualization techniques are available in the literature [91,93,94], however Parallel Coordinates is the most popular [95] because it is easy to use and enables the DM to perform tasks interactively and efficiently. One of the earliest reports of work done using Parallel Coordinates as a static user interface is in [96]. Visualization of the population in a high-dimensional objective space presented significant information allowing the DM to trade-off between objectives, assessing the quality of the Pareto front, and helping the DM to express his/her preferences [97].

Given the success of the Parallel Coordinate approach as a visualization tool for exploring data analysis [95], researchers have integrated the Parallel Coordinate visualization features into the design optimisation process to provide support for optimal decision making. Multi-objective optimisation methods are categorized into three categories; a priori, posterior and interactive, which depends on when the DM provides his/her preference. To increase the effectiveness of the multi-objective optimisation problem, the DM should interact while the process is running to build an understanding of the problem domain by gathering more information that will help to gradually steer the process to the Region of Interest (ROI).

One of the earliest interactive approaches is presented in [18]. This approach allowed the DM to explore the dominated solutions by using a heat-map-visualization-based user interface, as shown in Figure 2-7. Using this approach, the DM is not required to review every individual solution in order to make a choice, rather it is necessary only to focus on small particles of interest to guide the solutions. The shortcoming of this approach is a high frequency of interactions is required, which was the inspiration for selecting Parallel Coordinate visualisation based on work carried out by Kipouros et al. [98,99] for further works.

Subsequent work produced a more flexible approach using the idea of a visualisation-based user interface that implemented the Parallel Coordinates technique and a 2D scatter graph to formulate the DM's preferences [18,100]. Figure 2-8 shows the user interface overview. The visualization tools enabled the DM to reach the right decision by recognizing the existing trend correlations between design variables as well as objective functions [15]. By interacting with the Parallel Coordinates and 2D scatter graph, it is possible to steer the optimisation search according to the DM's expertise and preferences, to focus on the ROI.

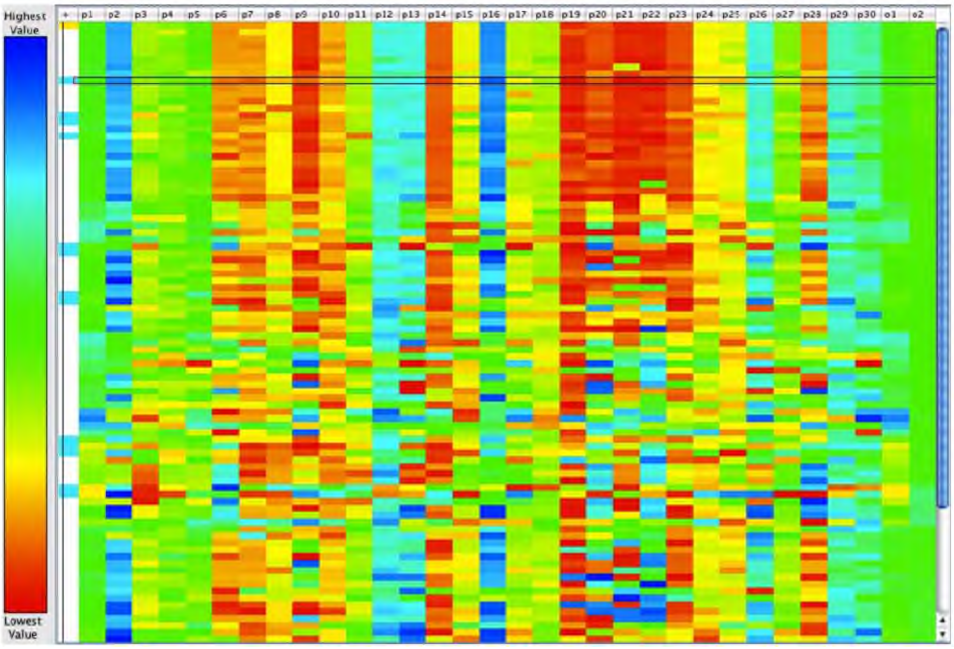


Figure 2-7: Heatmap visualisation base: The rows represent individual solutions while the columns either represent objectives or design variables [18]

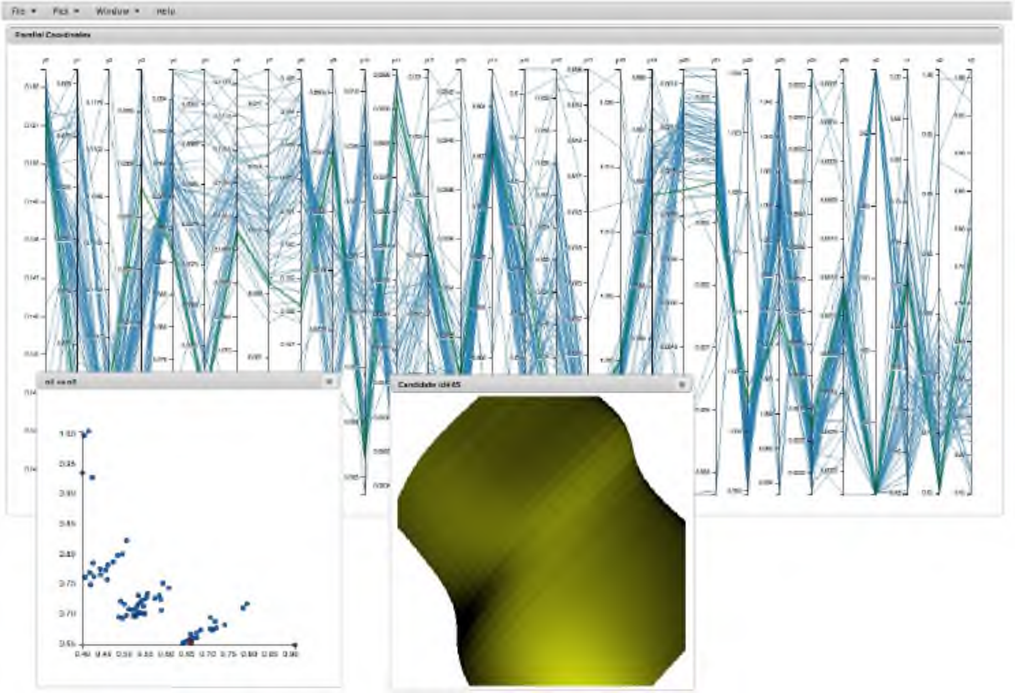


Figure 2-8: **Visualisation-based user interface showing implementation of the Parallel Coordinates technique, 2D scatter graph, and window showing the candidate solution** [100]

2.4 Summary

The procedures used to optimise aerodynamic considerations related to this research have been introduced in this chapter. This includes discussion on the characteristics of the optimisation techniques, of global and local solutions, the concept of Pareto equivalence, and dominated and non-dominated solutions. This is followed by a comparison between the two main optimisation techniques; deterministic and stochastic. The focus is on stochastic algorithms such as MOTS and MOPSO since these will be the algorithms applied in the proposed design framework.

The next chapter introduces the applications of three advanced optimisation techniques; Automatic design tool (Nimrod/O), interactive optimisation and an Artificial Neural Network in the field of aircraft aerodynamic shape design. The advantages and disadvantages of each technique are explained.

3 Chapter - Optimisation Applications and Decision-Making

Aeronautics has benefitted substantially from the development of optimisation techniques, which have the potential to improve the efficiency of the design process and the quality of the new products. In practice, real-world problems consist of multiple conflicting requirements that a designer struggles to meet and find an efficient compromise solution in a suitable computational time. Hence, improving optimisation techniques is an attractive field for many researchers. This chapter presents a discussion on the use of advanced computational tools and techniques in the field of aircraft shape design. It also covers some important considerations used to classify each technique; computational resources required, efficiency, simplicity, applications, and advantages and disadvantages. These considerations include the benefits of using an automatic design optimisation process, interactive optimisation, and machine learning to accelerate the design cycle and obtain optimal or near optimal results. The focus is on using methodologies and computational tools to provide correct and effective support decisions, making for a design optimisation process.

Increasingly, nowadays, computational optimisation methods are essential components of engineering design and manufacturing systems. These applications have changed the conventional design procedure that depended directly on the design and builds a prototype for the test with the design process that depended on producing a mathematical model that simulated the design problem and performed repetitive evaluations until a near optimal solution was obtained, after which a prototype was built and tested [101]. Selecting the most desirable solution (optimal compromise solution) for any design problem requires a human decision. Such a decision can be either occur a priori, a posteriori or interactively [15].

3.1 A priori approach

In this approach, the DM is required to specify his/her preferences before the solution process starts. This is usually done by using the weighting, or ϵ-

constraint, method where the design optimisation problem is formulated as a single objective instead of a multi-objective problem. Once the design problem is formulated, the optimiser will try to find a single optimal solution satisfying the articulated preferences of the DM [6].

- In the weights methods, the DM is asked to specify the weight ($\mathrm{w_i}$) for each of the objective functions. The problem is then formulated as [82]:

$$\mathbf{min} \sum_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{w_i}\, \mathbf{f_i}(\mathbf{x}) \qquad (3\text{-}1)$$

Subject to

$$\mathrm{x} \in \mathrm{S}$$

Where $w_i \geq 0$ for all $i = 1, \ldots, k$ and the summation of the total weights is equal to unity. The method is simple and direct, but usually the DM does not have adequate information to accurately define the problem. Thus, wrong weightings for the different objectives may be used [33]. In case of convex problems, changing the weights of the objectives is necessary to test that the optimal solution has been found [102].

The weighting method can be used as an a posteriori approach by performing multiple runs using different weights to obtain multiple trade-off solutions. In this case, the DM is asked to select a single solution as the trade-off [6].

- On the other hand, in the ϵ-constraint method one objective can be selected to be optimised while the others are converted into constraints. The problem is then defined as:

$$\mathbf{min}\ \mathbf{f}_{\boldsymbol{l}}(\mathbf{x}) \qquad (3\text{-}2)$$

Subject to

$$\mathrm{f_j}(\mathrm{x}) \leq \epsilon_{\mathrm{j}} \quad \text{for all} \quad \mathrm{j} = 1, \ldots, \mathrm{k}, \quad \text{and } \mathrm{j} \neq l$$

Where the ϵ_j is the upper bound for the objectives and $l \in \{1, \ldots, k\}$.

The solution obtained using the ϵ-constraint method is always weakly optimal for two reasons, only one objective is considered in the optimisation process, and there is invariably a lack of information required to specify the upper bounds for the constraint objectives [102]. Thus, several runs must be performed to prove that the obtained solution is optimal, which is usually computationally expensive.

Many studies have investigated multi-objective optimisation and have emphasised the possible benefits compared to single-objective optimisation with a composite objective function. For example, Gaiddon et al. performed a study to compute the aerodynamic balance of a missile by finding the best inlet shape for the missile during acceleration, cruising, and manoeuvring. The work compared a number of optimisation algorithms using both mono-objective and multi-objective optimisation approaches. The results showed that only by using multi-objective optimisation and Pareto solutions could the best set of designs be found to satisfy the several performance criteria [103].

3.2 A posteriori approach

On the other hand, in a posteriori optimisation, the DM is required to specify his/her preferences after the optimisation process is over. In a multi-objective optimisation problem, several objective functions are to be optimised simultaneously to find a well-distributed Pareto front. The results will be presented to the DM who is assumed to know the problem being considered sufficiently well as to be able to perform a trade-off between the different objectives and select a single solution. The main benefits of a posteriori approaches are their ability to provide Pareto optimal (non-dominated) solutions for multi-objective design optimisation problems in a single run, which is an advantage over the a priori approach [6]. In this case, the DM is required to select the most preferred solution after having had an overview of the different Pareto solutions [104].

Generally, the multi-objective optimisation problem is similar to a single optimisation problem in containing a number of constraints that need to be satisfied by the optimiser to find feasible solutions. The multi-objective optimisation problem in its general form is either a minimisation or maximisation and may defined as [105]:

$$\max/\min \ f_m(x) \qquad m = 1,2,\dots,M \qquad (3\text{-}3)$$

Subject to

$$h_j(x) = 0, \qquad j = 1,2,\dots,J$$

$$g_k(x) \leq 0, \qquad k = 1,2,\dots,K$$

$$x_i^l \leq x_i \leq x_i^u, \qquad i = 1,2,\dots,n$$

The solution of the multi-objective optimisation problem is expressed mathematically in form of non-dominated solutions (Pareto order). By definition, a decision variables vector $x = (x_1, x_2, \dots, x_n)^T$ dominates another decision variables vector if it is not worse in any objective and is strictly better in at least one objective, see Figure 3-1.

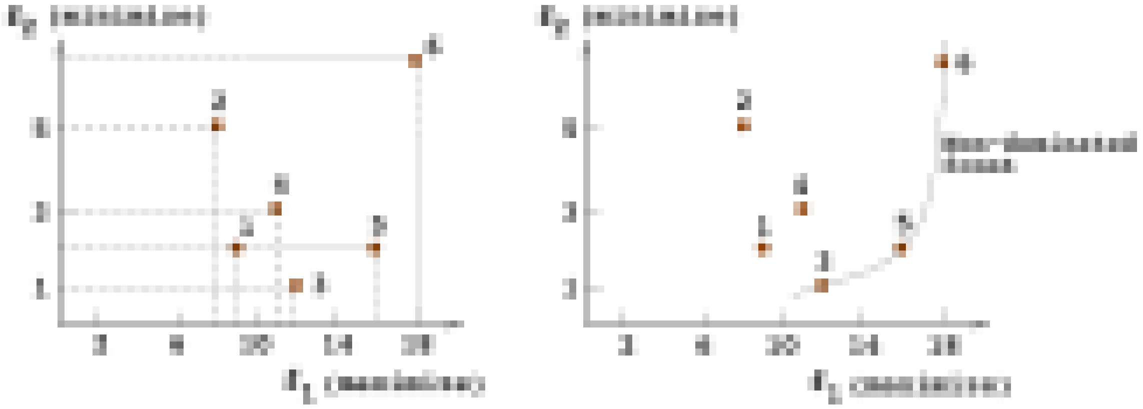


Figure 3-1: Left panel; set of solutions on the objective space. Right panel; first non-dominated solutions (Pareto front) [105]

By pair-wise comparison of the solutions shown in the left panel of Figure 3-1, using Equation 3-3, the non-dominated solutions can be established. The right panel of Figure 3-1 shows that only three points from the set of six solutions are considered as non-dominated solutions. The non-dominated solutions share the property that neither of two solutions can improve any more with respect to the other, without sacrificing one objective in the other solution [65].

The use of this approach led to the development of many popularly used optimisation tools [77]. The aim is to provide better results regarding computational time, quality of the solutions, and the labour required for setting up the test experiments.

3.2.1 Automatic design optimisation tools

Several optimisation tools have been developed to assist the designer in performing a demanding design experiment using an arbitrary computational model. Nimrod [106], modeFRONTIER [107] and iSIGHT [108] are examples of tools to improve the design process while reducing the time and cost. These computational frameworks can assist a designer to verify different design scenarios before committing to a specific design which reduces computational requirements and duration of the design cycle, while improving the quality of the product [109]. Moreover, such an approach allows the designer to focus on the technical issues of the design problem instead of programming details and operating system requirements.

This work used Nimrod/O, a tool that has previously and successfully been used with the High-Performance Computer (HPC) at Cranfield University [77]. The HPC system allows 250-300 users to carry out complex research across many processors simultaneously [110]. Nimrod/O has been applied to a varied range of problems since it was launched. For example, in [111], Nimrod/O tool was used successfully with various design problems that require intensive computations, including modelling of air quality, airfoils, plate fatigue life and quantum chemistry. The evaluations were performed on parallel batches, which is limited only by the number of processors available. Other projects showing the efficiency of Nimrod/O can be found in [4], where different test functions were optimised.

Furthermore, Nimrod/O is easily to adapt to new optimisation algorithms simulating real-world design problems. According to [77] the first multi-objective optimisation algorithm interfaced with Nimrod/O was DEMO (Differential Evolution for Multi-objective Optimisation). The framework was tested on two different optimisation problems; the first was a standard mathematical function with two objectives, while the second was shape optimisation of rib-reinforcement that consisted of three objectives. The success achieved in interfacing DEMO with the Nimrod/O package has encouraged researchers to further enhance Nimrod/O with more powerful multi-objective algorithms, such as Multi-Objective Tabu Search (MOTS) [64] and Non-dominated Sorting Genetic Algorithm (NSGA) [112].

Abramson et al. studied an airfoil design problem using NSGA and Simulated Annealing (SA) algorithms implemented in Nimrod/O using an algorithm written especially for the problem. The conclusion shows that using Nimrod/O provided greater flexibility with a shorter computational time and less work required for setting the experiments [113]. In [114] the Nimrod/O optimiser was used to determine the geometry of a hole in a thin plate under load, which gave optimal life. Multiple searches from various start points were used to overcome the problem of multiple local optimums.

Using a Sweep Method (SM), Nimrod/O can perform a full parametric sweep across all possible design variable combinations to explore the entire design space [115]. Such a process examines the range of design scenarios and provide a clear idea for the DM of both how the design variables and objective functions are related, and the limitations of the design space [116]. This will lead to a robust optimisation process in a shorter time.

Several researchers have investigated the design optimisation of aircraft using various levels of fidelity to reduce the length of the design cycle and computational cost, while improving the quality of the design [117]. However, Piperni et al. [25] and Zhang et al. [118] have argued that the level of fidelity that should be delivered by the models is mostly determined at the development stage, which the design process aims to enhance. Leifsson et al. [4] found that optimisation results for both expensive high-fidelity models and low-fidelity models show the same trends, but the optimised results from a low fidelity model have the advantages of being efficient in terms of computational costs. The low-fidelity computations are around 320 times faster than high fidelity computations using Navier-Stokes equations.

The first part of this thesis discusses the application of Nimrod/O to the optimal design of aircraft shape, which is considered as a compromise between flight conditions and shape parameters. To deliver a sufficient level of fidelity with very short computational times to the DM, the low fidelity flow solver Athena Vortex Lattice (AVL) has been used to capture the physics of the problem at this stage of the design process [9,119–121]. It is unnecessary to use high fidelity computational tools in this stage of design when the AVL can give reliable results of acceptable accuracy. It is better to use high fidelity tools at the detail stages of the design or at higher Mach numbers where low fidelity tools fail to achieve acceptable results [118]. To develop an approach that can be used for any aircraft, the existing Aegis UAV platform - with its available and extensive database - is used for developing the methodology.

Actually, the idea of using optimisation for the design of the aircraft's aerodynamic shape is not new and several researchers have performed such investigations using a Gradient-based Method (GM) [55,122–125] or Evolutionary Algorithms (EA) [59,61,62,126,127]. The GMs usually suffer from the difficulty of getting gradient information for the objective function, and the optimiser often becomes trapped in local minima. In contrast, EAs have the important ability to compute the global minimum [60]. However the convergence is slow since they requires a large number of evaluations [29].

To investigate the implementation of more advanced optimisation techniques, initially this work used Multi-Objective Tabu Search (MOTS) as the search algorithm [64,128]. TS has rapidly expanded in recent years as one of the more efficient meta-heuristic techniques able to cope with complex real-world problems [69]. For example, in [128] the MOTS algorithm is integrated into a design system for the optimisation of turbomachinery blades. The aim was to improve the performance of the engine and give an insight into the nature of the design problem which, if successful, could significantly reduce the work load on the designer. The obtained results showed that using a robust optimisation algorithm for solving a real-world problem can successfully provide a set of non-dominated solutions, which provide good support for the DM in determining the trade-off between the objectives under consideration. In similar work, the TS algorithm was integrated with a multi-fidelity flow solver for the test case of the aerodynamic design of two different airfoils provided by industry [129]. The aim was to accelerate the convergence to the global optimum in less computational time. The results show the advantages of using a low fidelity solver to reduce the high fidelity function evaluations required for each design case.

3.3 Interactive approach

In the a priori or a posteriori approaches, the designer formulates the problem and allows the optimisation loop to run automatically until solutions are identified. Early optimisation was usually considered as a single objective optimisation problem, but the need to simulate real-world design problems required the capability of solving multi-objective design problems where there was no single objective solution, but a set of non-dominated solutions which required a human input to the analysis to select an appropriate compromise solution. Even though this technique is appropriate since it requires the least interaction with the designer, it may require additional computational effort since it explores the whole design space intensively [130], which can lead to post-data analysis problems. Also, it is not a particularly good approach for design problems that contain more than two objectives [20]. Thus, new procedures and ways of integrating and applying existing technology must be explored and developed [131].

3.3.1 Human in the loop

For real-world problems, the DM is interested only in a sub-region of the objective space. The benefits from trade-off solutions outside the region of interest are very small compared with the computational cost and effort made by the DM in analysing unnecessary information [14]. Researchers have investigated different optimisation techniques to reduce the length of the design cycle, reduce computational cost, and improve the quality of the design [117]. The solution to such a problem can be to incorporate the designer within the optimisation loop [15–18]. Figure 3-2 shows a comparison between non-interactive and interactive optimisation where the DM is involved within the design loop. This makes it possible to focus all the computational resources on exploring only the areas of interest within the search space to identify just the preferred designs for the intended applications. Actually, by applying guidance to the search and ignoring regions which are not interesting to the designer, the computational cost is also reduced [15].

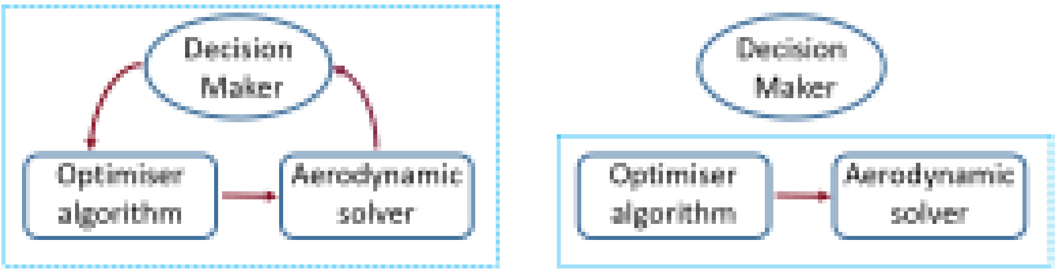


Figure 3-2: **Left panel; the preference base design loop (interactive). Right panel the conventional design loop (non-interactive)**

Over the last three decades, many researchers have been focused on developing interactive multi-objective optimisation methods. The majority of these methods are based on Multi-Attribute Value theory (MAVT) [132], where the performance is captured by asking the DM to compare the non-dominated solutions either in pairs or in groups to articulate the required preference. Based on the DM preference, the optimisation problem will be formulated as finding a suitable value for the objective function satisfying the preferences of the DM [18,100]. The majority of these approaches are based on an Evolutionary Algorithm (EA). For example, in [20] an evolutionary multi-objective optimisation algorithm was developed to lead the DM to regions of interest depending on his/her preference. The progress towards the preferred solution was developed by interaction with the DM after a certain number of iterations. The preference was used to model a value function, which was used until the next interaction.

Ke Li et al. [81] derived an approximation value function that modelled the DM preference by scoring the candidate solutions and then employed an EMO algorithm to lead the DM to the region of interest (ROI) of his/her choice. An interesting work by Muberra et al. developed an interactive evolutionary algorithm for a multi-objective feature selection problem that depended on a preference-based approach. The DM guides the search to the regions that contain the interesting solutions. A supervised learning algorithm is used to assist with measuring the feature selection performance classification, eliminating irrelevant information, improving the time efficiency and simplifying the optimisation search as a whole [22]. More such work including evolutionary algorithms are in [16,20,133,134].

In contrast, there are not yet many studies adapted to use the Particle Swarm Optimisation (PSO) algorithm to optimise interactively [18]. Agrawal et al. proposed an interactive particle swarm approach for solving multi-objective optimisation problems by including DM preference. The interaction starts only after a finitely large archive of non-dominated solutions is reached. The role of the DM is to select a preferable solution from among the non-dominated Pareto solution by making a pairwise comparison. The approach used an adaptive-grid mechanism for incubation of the particle swarm for the multi-objective optimisation [135]. A good summary of interactive Multi-objective Optimisation methods based on MAVT can be found in [20]. However, such methods are still not yet applicable to real-world problems. Methods in this category are limited to show information or plots for selected points in the objective space. Whereas, the concept of using interactive optimisation is to provide enough information about the design space for the DM to be able to steer the optimisation search to the ROI efficiently [100].

The first interactive optimisation approach based on a non-MAVT-base was presented in [18]. It used a Multi-Objective PSO algorithm to guide the particles to the ROI using DM preferences by means of a heat-map-visualization-based user interface. In this method, the DM is not required to review every individual solution to make a choice, it was necessary only to focus on a small number of particles of interest to guide the solutions. Even though this method allowed the DM to focus only on the ROI, it required a large number of interactions [100]. Later, a new version of the interactive Multi-Objective Particle Swarm Optimisation using DM interaction was developed [15]. It uses the idea of a visualisation-based user interface that implements the Parallel Coordinates technique, and a 2D scatter graph to articulate the DM’s preferences. The approach was used successfully on a 2D aerofoil benchmark problem used by Kipouros et al. in [136].

The literature on using an interactive approach for aerodynamic shape optimisation is modest. To date, most published work using interactive techniques concerns either standard test problems or optimisation of wing airfoils, as explained previously. This work introduces a preference base framework, which incorporates the flow solver Athena Vortex Lattice (AVL) [137] with stochastic Multi-Objective Particle Swarm Optimisation (MOPSO) [15] to lead a DM to the solution of his/her choice using a visualisation-based user interface. To demonstrate the effectiveness and superiority of this approach, we compare the interactive optimisation results with the non-interactive results using MOPSO and MOTS algorithms for the aerodynamic shape design optimisation problem of the Aegis UAV. The framework used was flexible and able to provide optimal or near optimal solutions in much less computational time when compared to the a posteriori approach.

However, due to the stochastic characteristics of the algorithms used and early steering of the optimisation search, the DM may miss some important information, which may be the only penalty for performing the optimisation interactively.

3.4 Machine learning

Machine learning is a field of computer science that gives the computer the ability to “learn” without being explicitly programmed. The term machine learning was first introduced in 1959 by Arthur Samuel, one of the pioneers in the field of computer gaming and artificial intelligence [138].

Increasingly, machine learning (ML) is being widely used for classification [139], numerical prediction [140], and pattern recognition [141]. With ML the computer can “learn” the complex and multifaceted relationships between dependent and independent variables via a “black box” (or neural net), which processes the data. Such applications have been used extensively in, for example; biology [142], engineering [143–145], environmental analysis [146], information technology

[147] and medicine [148]. Such applications reveal the extent to which ML has been used to boost research and development.

The development of big data analytics and data-mining [149–151], has dramatically extended the practical application of knowledge-base machine learning models. One such is the Artificial Neural Network (ANN), a non-linear fitting algorithm, which is now a very popular ML technique thanks to its ease of training, adaptive structure, and tunable training parameters [152,153]. ANNs. can cope with non-linear problems and, when trained, are able to carry out predictions and generalisations very fast. They are also able to cope with ill-defined and complex problems, noisy and incomplete data and are fault tolerant [153].

With so many advantages the ANN has been developed extensively, and currently widely available are the back-propagation NN [154], extreme learning machine [155], general regression NN [156], and multilayer perceptron NN [157,158]. Previous research studies have shown that ANNs can be advantageously tailored to different practical applications [159,160].

3.4.1 Main architecture of ANNs

Figure 3-3 shows a structure of a typical ANN. It consists of at least three different layers: the input, hidden, and output layers. Each layer consists of a certain number of neurons. Each neuron inter-connects with all the neurons in the following and the previous layer. Each connection represents a weight that contributes to the fitting. With a proper activation function $\mathrm{f(NET)}$, a combination of optimised weights can generate the prediction of the dependent variable.

$$\mathrm{NET} = \sum_{\mathrm{i,j}}^{\mathrm{n}} \mathrm{w_{ij}}\,\mathrm{x_i} + \mathrm{b} \qquad (3\text{-}4)$$

Where $\mathrm{w_{ij}}$ represents the weight value of a connection, $\mathrm{x_i}$ represents an input independent variable, and b represents a bias. The training of an ANN is essentially the optimisation of each weight contribution based on the data groups in the training set. There are various techniques and optimisation methods that can introduce smooth non-linear fitting to the training of an ANN. Though there are some difference in the weight training and algoritmic structures, the basic principles, as well as the training and testing processes, are very similar [161].

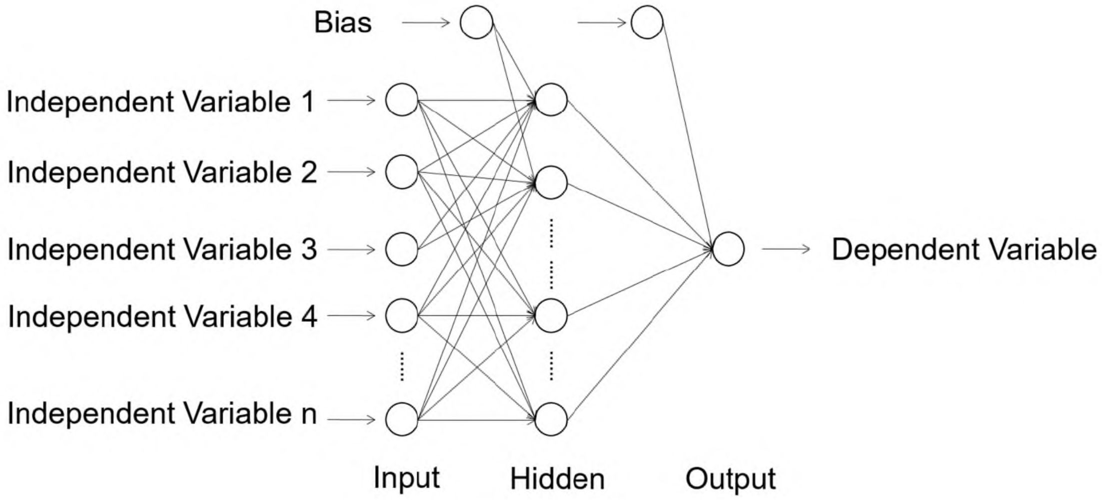


Figure 3-3: General structure of a typical artificial neural network, where each circle represents a neuron in the algorithm. The neurons are interconnected with all other neurons in the adjacent layers [162]

Machine learning is attractive to engineers because of its remarkable characteristic of learning to handle imprecise and uncertain information. Furthermore, it is able to achieve excellent generalized solutions through the use of powerful training algorithms, which produce solutions that are reliable inside and outside the region of design space used for the training data [163], and perform massive parallel computations, which has a significant impact on the computational time [164]. These factors have attracted researchers to the application of ANN with advanced optimisation algorithms for various aircraft design problem enhancements [165–167]. Actually, ANN is used mostly to model the objective functions (surrogate model) for the design problem, where the computing of the objective function is time-consuming and computationally expensive [168,169]. However, using surrogate objective functions instead of the real objective functions for complex industrial problems may degrade the accuracy of the results [11].

3.4.2 Properties of learning

In order to use the ANN to obtain the required results, it has to learn the structure of the domain. Generally, any NN has to be trained using a learning algorithm and known date set. The learning algorithms for NNs can be classified into two main categories: supervised learning and unsupervised learning [146]. The unsupervised learning is usually used to find the structure of unlabelled data. The aim is to learn more about the trained data to discover more of its features for identifying subsets. The cluster analysis of data is considered the most common use of unsupervised learning. For unsupervised learning, there is no need to know the outputs values.

In contrast, the output values are required when supervised learning is used. In other words, the training dataset or the desired outputs for a given set of inputs must be provided for the algorithm to adjust the network's weights. These weights will be used for processing any subsequent new set of data. These weights will be adjusted until the desired goal is reached. Generalized solutions mean the NN can produce an output that is very close to the expected outputs for any given input data, and is one of the main applications of supervised learning [170].

3.4.3 Application of the Artificial Neural Network

ANN's are a computational modelling tool able to process massive data sets concerning real-world problems [161]. The ANN models may be used as an alternative method in engineering analysis and predictions. ANNs are claimed to mimic the learning process of a human brain. They operate like a "black box"' model, requiring no detailed information about the internal workings of the system. Instead, they learn the relationship between the input parameters and the controlled and uncontrolled variables by studying previously recorded data, similar to the way a nonlinear regression might perform. Another advantage of using ANNs is their ability to handle large and complex systems with many interrelated parameters. They seem simply to ignore excess data that are of minimal significance and concentrate instead on the more important inputs.

Regardless of the machine learning in use, obtaining high optimality requires a deep understanding of the numerical optimisation technique, training procedure, specification of the machine in use, and full knowledge of the design problem. For that reason, researchers have investigated various traditional optimisation algorithms to obtain the best combinations of the process parameters. Although these traditional optimisation algorithms had performed well in many particles cases, they did have some restrictions related to their search procedure [6]. The solutions obtained by many ANN algorithims can be far from the optimal solutions that are expected by a DM if it becomes trapped in local minimum [23]. To overcome such a problem, bio-inspired algorithms, which are based on natural behaviour, such as PSO algorithms and EAs, have been proposed [24].

Keeping in view the success of EAs [171], ANNs had been used in many projects to construct a surrogate model to reduce the overall computational cost and to obtain greater optimality in the solutions [171,172]. Because of its stochastic characteristics, Genetic Algorithms (GA) have been introduced as a powerful tool for the aerodynamic shape design optimisation problem. The stochastic characteristics prevent the algorithm from being trapped in a local minimum [173]. Hari et al. used ANN to model a function for non-linear mapping of the quasi-steady stall information of an aircraft, and then used a Gauss-Newton (GN) optimisation technique to estimate the aerodynamic coefficients [174]. The obtained results were compared to the results obtained using the Equation Error Method (EEM), which required expensive information gathered from wind tunnel and computational fluid dynamics. It is found that neural-network base GN method provides better results. In [171] an efficient methodology was introduced to accelerate the optimisation process of design of the wing section. The expensive Nevier Stokes flow solver was supplemented by a low-cost ANN for the evaluation of the objective functions. To increase the computation speed and convergence of the solutions, a GA with parallel computation features was also used.

In a different optimisation problem Giannakoglou et al. investigated the use of multi-layer perceptron NNs with radial basis functions, by implementing an optimisation method based on an evolutionary algorithm. The application included several 2D and 3D aerodynamic shapes, respectively airfoils and turbomachinery blades. The ANN was trained to model the objective functions. The obtained results conclusively demonstrated the superiority of using ANN in reducing computational time [144].

In a similar work, Magrini and Benini used an ANN to create a surrogate model to approximate the highly nonlinear relationship for optimisation when morphing a leading edge airfoil [175]. The applied methodology was based on the GA. The results show that the ANN can provide a significant reduction of the convergence effort. In another interesting project a surrogate model was constructed to evaluate the aerodynamic characteristics for NACA2411 [168]. The work used different approaches for construction of the surrogate model with the goal of reducing the requirement for using high fidelity computational tools in the aerodynamic shape design.

On the other hand, considerable interest has been shown in setting the problem using PSO algorithms. The aim was to obtain high-quality products in terms of computational cost and performance. Since the algorithm has a simple mechanism it was computationally inexpensive in terms of memory requirements, and faster relative to other population techniques [74,75]. In [11], the authors extensively reviewed PSO algorithms used to evolve ANNs. However, most of the research was focused on using of ANNs to model the objective function, or the PSO was used to help with the training of the ANNs. For example, in [176], the authors proposed a method that used MOPSO to train the ANN to optimise the architecture and connections of the network simultaneously. The obtained results showed the effects of the proposed framework compared with other findings available in the literature. In [154] the Backpropagation Algorithm (BA) and PSO algorithm were used as training methods for an ANN. The obtained results were compared in terms of convergence and computational time. The PSO showed significant advantages over the BA in terms of robustness and efficiency when obtaining optimal solutions even with reduction of swarm number.

Similarly, in [177] the MOPSO was used to choose the fuzzy rules for the Pareto optimum, which were used to classify electrical loads. Another interesting work, where a single point optimisation problem for an airfoil was examined using Gradient-Base and PSO algorithms. The PSO solutions showed the limitations of using gradient methods with such a problem. In addition, to improve the computational time when using PSO, an ANN was developed and a surrogate model used instead of the flow solver software. Computational time was reduced and good optimality solutions obtained relative to the one obtained using the computational flow solver [178].

However, real-world design problems require a high degree of accuracy, which can be obtained only by evaluation of the objective function [11]. The concern here is that the evaluation of the objective functions for such design problems may take weeks or even months. Furthermore, the problem becomes worse when infeasible trial solutions are more numerous than the feasible, which is the case in aerodynamic shape design optimisation problems [12,13]. Therefore, it is not useful to spend a long time evaluating those worthwhile solutions [11].

A significant contribution of this research has been the use of a ANN to guide the optimisation algorithim by deciding whether the trial solution is worthy of full evaluation or not, as applied to aerodynamic shape design optimisation for the Aegis UAV. The ANN was used to perform deep space exploration, retaining all the useful information of the design space in an adequate computational time. In addtion, the research has made a significant contribution by comparing solving the problem by interactive and non-interactive optimisation approaches.

3.5 Summary

This chapter has introduced different architectures for aerodynamic shape optimisation, and provided an insight into the use of advanced optimisation techniques to obtain optimal or near optimal solutions. It is evident that the rapid growth in computing technology offers design engineers the potential of using design optimisation in real-world applications. Since real world problems are highly challenging, continued exploration and development of existing computational tools and methodologies are essential to identify and suggest practical solutions. It has been established that computational automatic design optimisation tools, interactive optimisation techniques, and machine learning algorithms are essential tools to improve the quality of new products in terms of performance, while reducing computational cost and time, the latter are important given the limitations of even supercomputers to find solutions in adequate computational times.

In the next chapter the detailed methodology for each technique used in this work is introduced: automatic optimisation, interactive optimisation, and Artificial Neural Network.

4 Chapter – Research Methodology

The aim of this chapter is to demonstrate the possibility of using advanced optimisation approaches to accelerate the optimisation of the design process while retaining all the useful information in the design space. Applicability of the methodology is presented through the aerodynamic shape design optimisation of the Aegis UAV.

The research methodology adopted used three different approaches; a posteriori, interactive, and machine learning, see Figure 4-1.

Figure 4-1: Schematic of the methodology used in this research, which consists of three different approaches; automated optimisation (Nimrod/O), interactive optimisation, and machine learning.

Figure 4-1 summarises the procedure used in this research. Details of the structure of each methodology, linking of the codes together and flow of the data are explained below. Furthermore, this chapter introduces the flow solver used to capture the physics of the design problem.

4.1 Computational flow solver

Design of the aerodynamic shape of aircraft has benefited greatly from the development of optimisation techniques, which have been linked with the flow solver software to accelerate the process of trade-off between the requirements of different disciplines [28,122]. However, if the flow solver is not accurate enough, the optimisation process will converge to solutions which violate the flow characteristics or display numerical errors.

Describing a candidate UAV shape through the optimisation process is achieved by satisfying the defined objective and constraint functions using the flow solver. In addition, repeated evaluations of the objective functions are required for each candidate shape, which makes the flow solver the most computationally expensive component within the optimisation framework. For these reasons, the efficiency of the framework is largely dependent on the selection of the flow solver. It is necessary to the keep the correct balance between solution accuracy and computational expense, which are dictated by the design stage and flow regime [25,118].

For example, due to the fluid nature of the conceptual design process, it is not recommended to use high fidelity analysis design tools such as CFD and FEM at this stage as they can be costly [5,8]. As explained in Section 2.1.1, what is required is a tool that strikes a better balance between accuracy and computational cost.

4.1.1 Athena Vortex Lattice (AVL)

There are several difficulties involved when considering the use of analysis tools in the earlier phases of design for optimisation of aircraft aerodynamic shape. A suitable selection and validation of the analysis tool is required before integrating it into the optimisation process. In addition, the code solver must meet some important requirements regarding the accuracy of the results and computation time.

“AVL code is a program for the aerodynamic and flight-dynamic analysis of rigid aircraft of arbitrary configuration. The code was developed by Mark Drela and Harold Youngren and is published under the GNU-General Public License” [137]. AVL is based on the Vortex Lattice Method (VLM), which is a numerical method. It is a well-known code for aerodynamic and flight-dynamic analysis of rigid body aircraft of arbitrary configuration. This code uses an extended Vortex Lattice Method for generating wings and tails (lifting surfaces) and slender-body models for fuselages and nacelles [179]. In this method, the lifting surfaces are modelled with horseshoe vortices distributed along span and chord. Effects of thickness and viscosity are neglected [38].

Many codes utilize VLM for aerodynamic characteristics calculations, but AVL is the most well-known and provides the most accurate and efficient results when compared with other aerodynamic analysis software employing the same method [179]. In addition, AVL code is easy to use and capable of manipulating a large number of design parameter within a short computational time and limited cost. It has the capability to simulate many surfaces at the same time, i.e., the wing downwash effect on the tail, and fuselage wing-tail effect, and to simulate complicated configurations (no limitation on wing sweep and dihedral angles). AVL is able to provide the geometry image of the lifting surfaces, fuselage and complete configuration.

AVL is most appropriate for UAV configurations, which are comprised of thin lifting surfaces with a small angle of attack [137]. However, AVL code is not capable of calculating non-lifting drag (friction drag), but can compute induced drag (lift related drag) with high accuracy [180,181].

The AVL code utilizes the Prandtl-Glauert (PG) transformation, which solves compressible flow problems using the Laplace Equation. The PG transformation can be used successfully for either freestream Mach numbers up to 0.5, or before transonic flow started [182].

Generally, the AVL code works with three input files; geometry input, mass and the run file. The geometry input file defines the model configuration and the properties of the aerodynamic sections. The mass file describes the properties of the mass; the magnitude of the mass itself, the centre of gravity, and inertial properties. The mass file is necessary only if stability analysis is performed. Both of the geometry and mass files are created by the user, using the simple text editor, but in case of the run file, the file is generated by the AVL code itself

4.2 Automated optimisation – a posteriori approach

Generally, real-world problems consist of multiple conflicting requirements to which a designer struggles to find or improve a compromise solution in a suitable computational time. Dealing with such complex design problems requires the designer to concentrate on the technical issues rather than programming requirements. Nimrod/O is an optimisation tool developed to simulate real-world design problems efficiently.

4.2.1 Methodology used

The aim of this section to demonstrate how the possibility of using advanced computational tools can accelerate the optimisation process and effectively retain the useful information contained in the design space. Applicability of the methodology is demonstrated through aerodynamic shape design optimisation of the Aegis UAV. Figure 4.2 summarise the components of the automated framework used in the various stages of development. It consists of four main parts: Nimrod/O design optimiser tool [77,111,113,183], optimiser algorithm (MOTS) [36,64,68], Sweep Method (SM) [115], and flow solver. The flow solver used here is the Athena Vortex Lattice (AVL) [137]. It is a low fidelity flow solver using an extended Vortex Lattice Method for simulating the flow around wings and tails and using slender-body theory for fuselage and nacelle modelling [179].

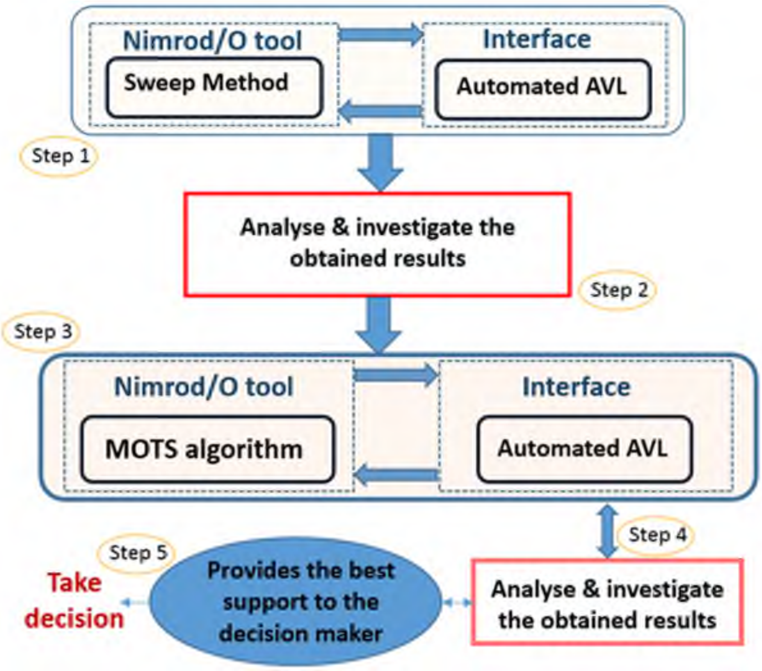


Figure 4-2: Schematic of the optimisation methodology used in the a posteriori approach

To define the new geometry for each candidate solution automatically, a code (Interface-AVL) to control the interaction was developed. The optimisation process required a repetitive evaluation of the objective functions for each candidate, which made the flow solver the most computationally expensive component within the optimisation framework. Thus, if the flow solver is not accurate enough and not able to capture the physical processes of this stage, the optimisation process will converge to solutions that violate the flow characteristics or display numerical errors. The Interface-AVL is developed to (i) generate the configuration needed to run AVL, (ii) satisfy certain constraints that are required to complete the flight mission, and (iii) compute other quantities such as parasite drag [42] and mass properties, that are required to perform the sweep and evaluate the objective function (see Appendix A for the parasite drag calculation). Before applying the framework to the Aegis UAV configuration, three different test cases from the literature were used to validate and verify the flow solver and the results from the Interface-AVL. These are presented in detail in Appendix B.

The Nimrod/O tool offers a declarative file to allow users to focus on the technical issues of the design problem instead of being concerned with programming details. By using such a file, the user can simply define the design variable characteristics, tasks to be executed, and the algorithm to be used. Firstly, the SM is used to explore the design space and identify the existing relations between design variables and objective functions. The analysis of these results was then used as an effective feedback to correct any dysfunctions that existed in the developed Interface-AVL, and then to formulate the design problem [83]. Next, MOTS was defined in the Nimrod/O declarative file to perform the optimisation. It took less than one minute to switch from the SM to MOTS. The optimisation started by finding the optimal number of regions and evaluations used to define the efficiency of the algorithm. Once the number of regions and evaluations were selected, the optimisation process started by performing various design scenarios to explore the UAV characteristics and provide the correct support for the DM to make the best decision.

4.2.2 Nimrod/O tool

Nimrod is a tool that provides a range of optimisation algorithms to assist engineers to evaluate the best design in a short time. It is unique in its ability to solve diverse kinds of optimisation problems without asking the user to write and develop his/her own optimiser code. Figure 4-3 shows a schematic of the Nimrod/O construction. The tasks to be executed by the optimisation methods are passed to Nimrod/G or EnFuzion to run in the HPC. To reduce the computational time, a cache is placed between the Nimrod and Nimrod/O to prevent recalculation of parameters already calculated. In addition, a permanent database is linked to the cache in case the Nimrod/O is terminated prematurely [130].

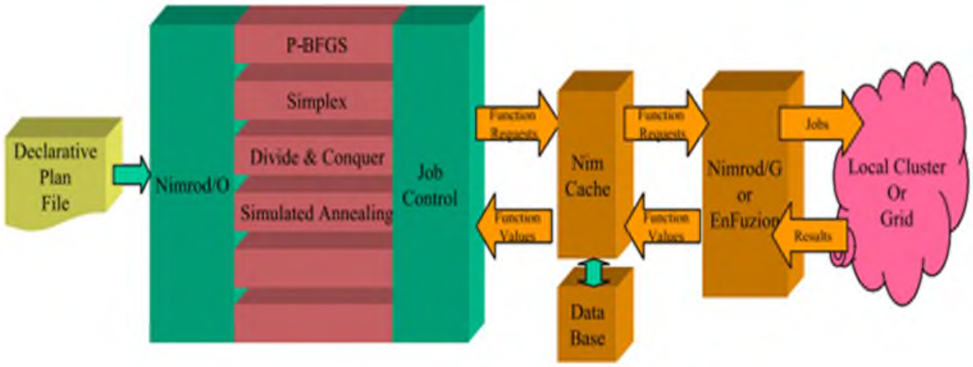
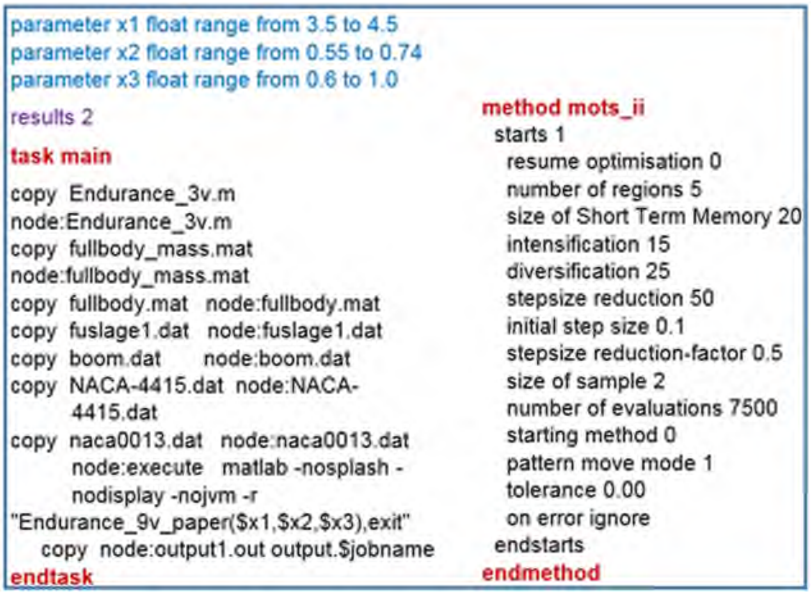


Figure 4-3: Schematic of the Nimrod/O tool optimiser construction [130]

Using such a tool, the designer was enabled to execute a computational model over various design variables utilising the resources on a global computational grid. The tool computed the value of the objective function corresponding to different combinations of parameters, and the resulting jobs were sent to a cluster of processors [77,114,136]. The design space was explored exhaustively over a range of pre-set design variables to minimise or maximise a selected objective function. The current Nimrod/O package that runs on the HPC at Cranfield University supports the following algorithms: Simplex, Simulated Annealing, Broyden–Fletcher–Goldfarb–Shanno (BFGS), Tabu Search (MOTS-II), NSGA-II, and Sweep Method (SM).

In order to define the design problem, Nimrod/O comes with a friendly interface text file (schedule). Figure 4-4 shows an example of a Nimrod/O declarative file, which can be written by any editor. It consists of three main sections; each section defines a particular part of the optimisation process. The first part defines the design variables; integer, float, and text. The second section defines tasks required to be executed. The final section of the schedule file contains the optimisation method. In addition, the schedule file may include constraints, which are located after the first part. The constraints can be either hard or soft: hard cannot be violated whereas soft can. In the case of soft constraints, penalty values should be added to the objective functions. More sections may appear in the schedule file for named results. If such sections exist, they will be after the constraints section and linked to the multi-objective optimisation algorithms only. Such sections will specify the number of objective values that will be computed during the run. If such sections are missing, then the number of objective functions is one [183].



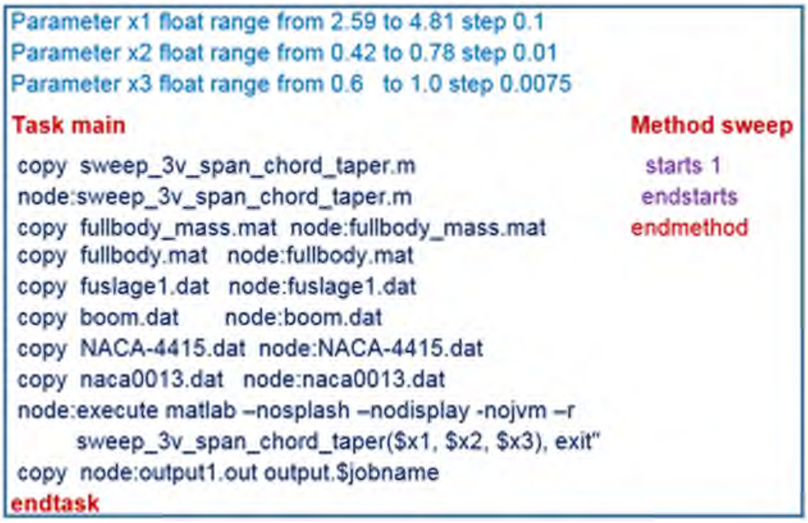
```
parameter x1 float range from 3.5 to 4.5
parameter x2 float range from 0.55 to 0.74
parameter x3 float range from 0.6 to 1.0
results 2
task main
copy  Endurance_3v.m
node:Endurance_3v.m
copy  fullbody_mass.mat
node:fullbody_mass.mat
copy  fullbody.mat   node:fullbody.mat
copy  fuslage1.dat   node:fuslage1.dat
copy  boom.dat       node:boom.dat
copy  NACA-4415.dat  node:NACA-
      4415.dat
copy  naca0013.dat   node:naca0013.dat
      node:execute   matlab -nosplash -
      nodisplay -nojvm -r
"Endurance_9v_paper($x1,$x2,$x3),exit"
   copy  node:output1.out output.$jobname
endtask
```

```
method mots_ii
 starts 1
  resume optimisation 0
  number of regions 5
  size of Short Term Memory 20
  intensification 15
  diversification 25
  stepsize reduction 50
  initial step size 0.1
  stepsize reduction-factor 0.5
  size of sample 2
  number of evaluations 7500
  starting method 0
  pattern move mode 1
  tolerance 0.00
  on error ignore
 endstarts
endmethod
```

Figure 4-4: A simple Nimrod/O declarative file, which consists of four main sections; design variables, results, task, and the optimisation algorithm used

4.2.3 Sweep method

The SM or parametric sweep is a method of exploring the parameter space of a design problem with high accuracy. A parametric sweep allows the designer to explore the design space by performing a sweep over the input parameters [115]. The parametric sweep takes all combinations of the input parameters which explore a range of design scenarios and provides a clear idea for the DM of how the design variables and objective functions are related [116]. The model may be computationally expensive, but the SM is executed in Nimrod/O parallel computing that can considerably accelerate the execution and allow complex design problem to be explored efficiently. Figure 4-5 shows an example of this file, which consists of three main sections; parameter descriptions, the main task, and sweep method characteristics.



```
Parameter x1 float range from 2.59 to 4.81 step 0.1
Parameter x2 float range from 0.42 to 0.78 step 0.01
Parameter x3 float range from 0.6   to 1.0 step 0.0075
Task main                                                Method sweep
 copy  sweep_3v_span_chord_taper.m                         starts 1
 node:sweep_3v_span_chord_taper.m                          endstarts
 copy  fullbody_mass.mat  node:fullbody_mass.mat          endmethod
 copy  fullbody.mat   node:fullbody.mat
 copy  fuslage1.dat   node:fuslage1.dat
 copy  boom.dat       node:boom.dat
 copy  NACA-4415.dat  node:NACA-4415.dat
 copy  naca0013.dat   node:naca0013.dat
 node:execute matlab –nosplash –nodisplay -nojvm –r
        sweep_3v_span_chord_taper($x1, $x2, $x3), exit"
 copy  node:output1.out output.$jobname
endtask
```

Figure 4-5: Parametric sweep declarative file, the sweep is performed over three design variables x1 (wing span), x2 (wing root chord), and x3 (wing taper ratio)

4.2.4 Optimisation algorithm

This work used the Multi-Objective Tabu Search (MOTS) from Cambridge University [68]. This is a meta-heuristic algorithm that searches for the global minimum and does not stop at local minimums. It is relatively simple to use and has already shown potential in the field of aerodynamics [128,136]. The Single Objective Tabu (TS) implementation of Connor and Tilley was used as a starting point for the multi-objective search [184]. The TS depends on Hook and Jeeves (H&J) local search algorithm that was coupled with three different memories to prevent being trapped in a local minimum.

Short-term memories (STM) are used to store recently visited points which are considered as “Tabu”. The search is not allowed to revisit those points which help it to climb away from a local minimum. In contrast, medium-term memory (MTM) is used to store the optimum, or the solution that is close to the optimum. The solutions stored in the MTM are used for an intensification strategy, i.e. restart the pattern search once no new optima are found. The long-term memory (LTM) has two functions; to record the regions which have already been explored, and to direct the search algorithm to the regions which are still need to be explored (diversification).

In general, algorithm search techniques use the local iteration point until a new candidate point is found. The new candidate point will be added to the MTM and replace the old one. This process will continue until the local iteration point reaches the user-specified value by diversifying or intensifying the search [36].

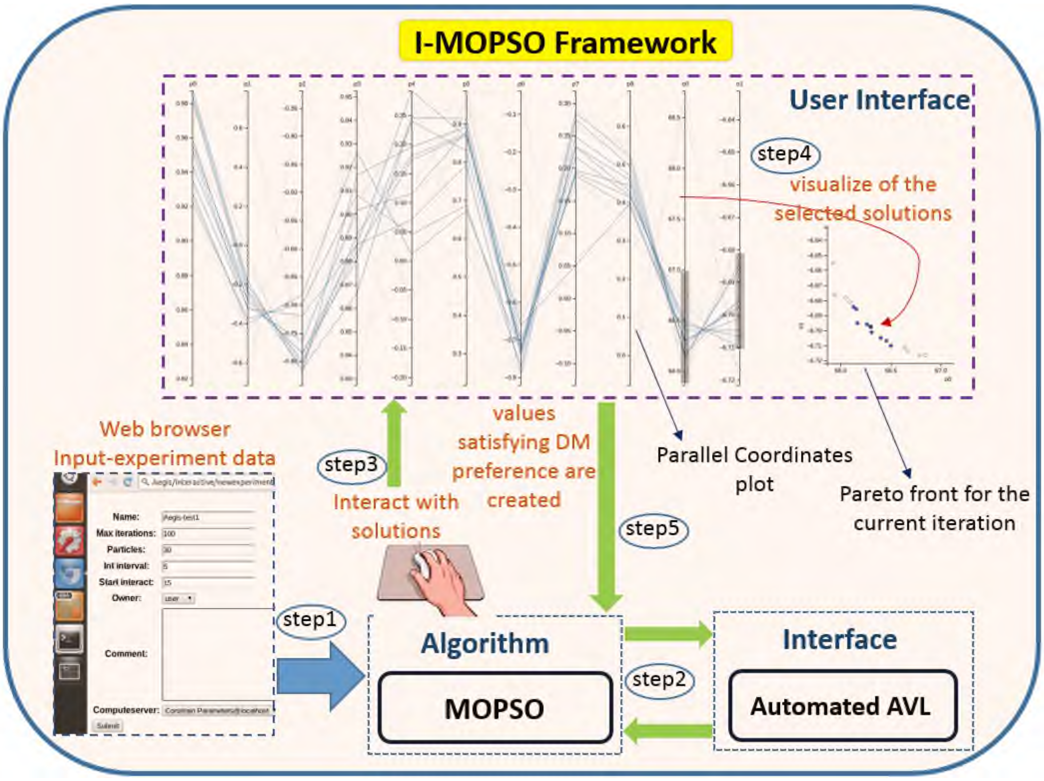
4.3 Interactive approach – human in the loop

The DM in most practical applications is interested only in a part of the Pareto front, the subset of good solutions. To come up with such results, the DM must be part of the optimisation process rather than waiting until the process finishes. The main idea is to focus only on the ROI rather than exploring all the design space [21,22]. Involving the DM within the optimisation process periodically to provide significant information usually helps with focusing on a sub-part of the design space. Consequentially, computational time is reduced and an optimal solution that satisfies the DM's preferences is more readily achieved [18,81]. In such an approach, the DM will provide his/her preference while the iteration is underway.

4.3.1 Methodology used

Figure 4-6 presents the main components of the interactive optimisation framework used in this work. The primary goal is to accelerate the optimisation search, and to achieve the best optimality level, by involving the DM in the overall optimisation process. The decision support framework consists of four main parts; the web browser input data, the main programme (developed in Python), Interface-AVL, and the interactive user interface.

The user-input interface is used to input parameters required for performing interactive optimisation. It is accessible from a common web browser, which is driven by a Python-based web application. This window allows the user to input the number of iterations, number of particles, and the required interactive intervals. The underlying algorithm used in this work was developed and tested by Hettenhausen et al. [15], which was a development of previous work [18], where the user interaction was based on heatmap visualisation. The framework used the a posteriori optimisation approach with MOPSO [74], as the method for searching and exploring the design space to obtain a set of optimal solutions.



/Aegis/interactive/newexperiment

Name: Aegis-test1
Max iterations: 100
Particles: 30
Int interval: 5
Start interact: 15
Owner: user
Comment:
Computeserver: Condman Parameters@localhost
Submit

Figure 4-6: Interactive optimisation approach methodology; incorporates decision maker preference within the optimisation process

The flow solver used for modelling and simulation of the new UAV configurations is the Athena Vortex Lattice (AVL) [137,179]. However, for the automatic reading and filing of this data into new files, an Interface-AVL code was developed [83]. To integrate the external model (Interface-AVL) with the I-MOPSO framework, the framework was amended following Hettenhausen et al. [185]. Once the optimisation run is started, the DM interacts with the process periodically. The visualisation-based user-interface in this framework used a Parallel Coordinates technique [84,89,90], to articulate the DM's preferences and provide a 2D scatter graph. Both visualisation tools enable the DM to make the best decision by recognizing the existing trend and correlations between design variables as well as objective functions [15]. By interacting with the parallel coordinates and 2D scatter graph, it is possible to steer the optimisation search according to expertise and preferences, to focus on the most interesting set of solutions or even a single solution.

4.3.2 Multi-Objective Particles Swarm Optimisation (MOPSO)

The MOPSO algorithm used in this framework was initially developed by Coello Coello and Lechuga [74], then further developed and tested by Hettenhausen et al. [15]. The majority of MOPSO algorithms share the same basic approach: a swarm of a certain number will be initialized randomly and that number will remain constant until the end of the run. The swarm behaviour is bounded by the velocity equation, which is updated continuously dependent on both the previous weighted velocity and known good solutions $\mathrm{p}_{\vec{\mathrm{pbest}}}$ and $\mathrm{p}_{\vec{\mathrm{gbest}}}$ representing the particles and the whole swarm, respectively [15].

The velocity equation represents the information exchanged between the particles in the swarm, and is responsible for updating the position of the particles at each iteration $\mathrm{t}+1$, and can be written as:

$$\vec{\mathrm{v}}_{\mathrm{t}+1} = \mathrm{w} * \vec{\mathrm{v}}_{\mathrm{i}} + \mathrm{c}_1 * \mathrm{r}_{\mathrm{r}}\left(\mathrm{p}_{\vec{\mathrm{gbest}}} - \vec{\mathrm{x}}_{\mathrm{t}}\right) + \mathrm{c}_2 * \mathrm{r}_2\left(\mathrm{p}_{\vec{\mathrm{pbest}}} - \vec{\mathrm{x}}_{\mathrm{t}}\right) \qquad (4\text{-}1)$$

Where the $\vec{\mathrm{x}}_t$ denotes the position of a particle p_{i} in iteration t, and it is updated using the equation:

$$\vec{\mathrm{x}}_{\mathrm{t}+1}(\mathrm{t}) = \mathrm{x}_{\mathrm{t}} + \vec{\mathrm{v}}_{\mathrm{t}+1} \qquad (4\text{-}2)$$

Where c_1 and c_2 are positive constants representing the attraction of the particle towards its own success or to its neighbours, and are called acceleration constants, r_{r} and r_2 are uniform random weights in the range $[0,1]$, and w is the inertial weight which takes a value of 0.4 and is used to control or define the impact of the previous particle velocity on the current particle velocity [74]. $\mathrm{p}_{\vec{\mathrm{pbest}}}$ and $\mathrm{p}_{\vec{\mathrm{gbest}}}$ commonly represent the archive of non-dominating solutions discovered by a specific particle during the iterations and by the whole swarm, respectively.

In order to select a leader, the search space is divided into a hypercube grid (10x10). The non-dominated particles are stored in the archive at each iteration and each is defined according to its objective function value. The hypercube grid containing more than one particle will be assigned a fitness equal to the number of particles divided by ten to reduce the fitness of that hypercube. Next, the leader will be selected by applying roulette-wheel selection. This process will continue until the end of the iterations [15,75].

The major difference between the algorithm presented in a previous study [74] and the one used here is the addition of further guidance to $\mathrm{p}_{\vec{\mathrm{pbest}}}$ and $\mathrm{p}_{\vec{\mathrm{gbest}}}$. This is actioned by the DM and executed via “virtual guide particles”. The “virtual guide particles” enhance exploration and diversity in regions and intervals selected by the DM. The DM chooses the ranges of parameters toward which it wishes to guide the optimisation, leaving MOPSO to guide the particle in the direction of the specific interval. Values that satisfy the DM constraint are generated and listed for every dimension in the domain; these originate from identified solutions in the archive and are used to generate the “virtual guide particles”, in accordance with the following conditions.

- When at most one point satisfying the constraints is found, a random particle is generated via a Gaussian distribution which is centred at the mid-point of the upper and lower boundaries with a standard deviation of about 10% of the separation between the upper and lower boundaries.
- When more than one point satisfying the constraints is found, but no other boundary limits have been set by the DM, a single value is chosen at random and a small turbulence value is applied to it.
- When more than one point satisfying the constraints is found and a specific boundary limit has been defined by the user, the value of the parameter is selected as determined by the convergence of the range:
    - If the selected range has less than 80% coverage by established points, a single point is randomly selected from within the largest gap.
    - If the selected range has more than 80% coverage by established points, an existing point is randomly chosen from within the range, and a small turbulence value applied to it.

.

4.3.3 Particles selection schema on I-MOPSO interface

The user-interface is focused on anticipating what the DM might need to do. It ensures an interface that is easy to access, understand, and use to facilitate the necessary actions. Steering the optimisation process towards the desired solution starts once the DM submits the number of iterations, number of particles, and the interactivity interval. Once the pre-set interval is reached, the DM can start to interact with the optimisation process. In the user interface, each parallel coordinate axis enables the DM to select a range for the design variable and objective functions that are assigned to that axis. This is performed by scrolling a grey bar within the desired range along the axis, see Figure 4-6.

Applying such constraint to multiple axes enables the DM to visualise the correlation that exists between the design variables and objective functions. By selecting a range of the desired solution along the objective axis, particles that contribute to this solution from all the design variables will be highlighted. The analysis and investigation of the highlighted and non-highlighted solutions provide an insight into the correlations that exist between the design variables and objective functions. It is stressed here that MOPSO in this framework generates design variables in the interval [-1, 1]. Thus, the design variables' values displayed on the parallel coordinates also range in the interval [-1, 1]. However, scale and translation factors are applied to each design variable after the initialisation to deliver those particles with the appropriate values to the external model (Interface-AVL) [185]. Figure 4-7 shows an example of DM selections that can be made with parallel coordinates and the corresponding observations that may be obtained from it, as locations on the Pareto front. Note that the DM can select a single particle, which is shown by a green colour on the parallel coordinates and red on the scatter plot.

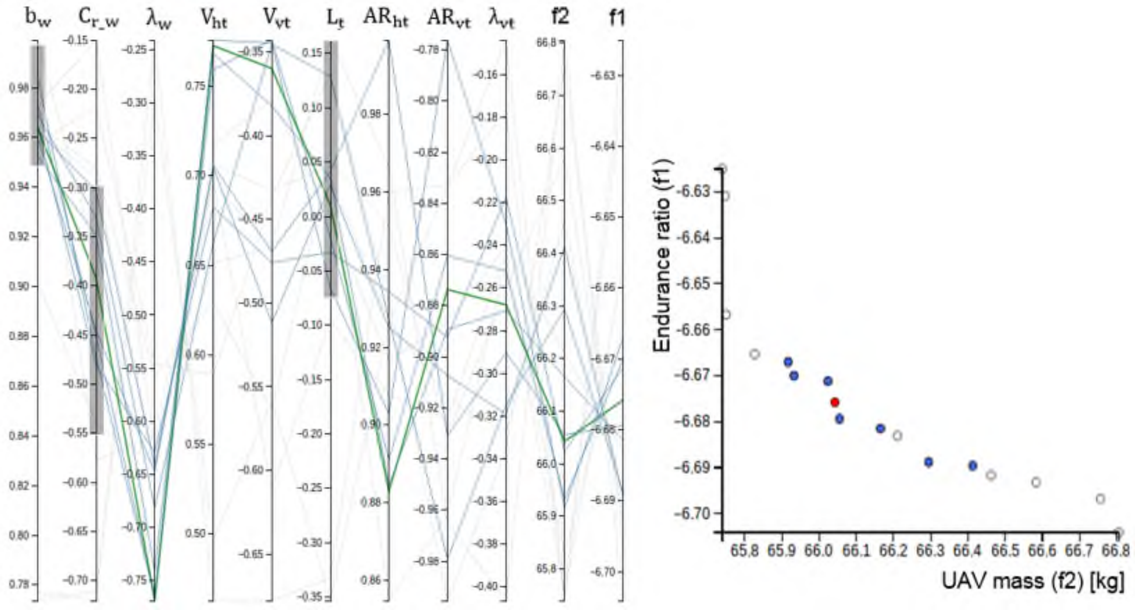


Figure 4-7: Particle selection schema, selection of objective functions on parallel coordinates and its location highlighted on the Pareto front using a scatter plot

Experience of the above has shown that the DM is able to interact with the optimisation process and guide the search in the desired direction. Furthermore, the location of the potential solutions within the Pareto front scatter graph provides a clear idea of the direction in which the solutions must be steered. The scatter graph is used to display the correlation existing between any two parameters in the design space. More details about using “virtual guide particle” schema to promote those regions considered desirable by the DM can be found in [15].

4.4 Machine learning approach

In this research, the ANN is used to improve the performance of the optimisation process by including the real objective function. To overcome the time-consumption problem, only algorithm trial solutions classified as valid by the ANN will be evaluated.

4.4.1 Methodology used

Figure 4-8 shows the schematic diagram of the proposed ANN framework for the aerodynamic shape design optimisation of the Aegis UAV, obtained by considering continuous live training and a certain level of scepticism. The scepticism parameter demonstrates the level of doubt towards the invalid particles that result by the ANN. The learning machine framework consists of three main parts: The MOPSO algorithm, Interface-AVL, and the ANN. The optimiser algorithm used in this work was developed and tested by Coello Coello [76] to handle multi-objective optimisation design problems, and was inspired by [71]. The ANN is used here as a fast approximation evaluator to decide whether the trial solution by the optimiser (MOPSO) is worth full evaluation or not. On the other hand, the flow solver used to evaluate worthwhile solutions is the AVL [137,179]. To continuously read the new design variables and then construct the required files for the AVL code, the Interface-AVL code was developed for automatic reading and filing [83]. However, only the particles signed by the ANN as valid are modelled for the evaluation. The Neural Network (NN) used in this framework is a multilayer perceptron that focuses on simplicity through scripting and component-based design [11].

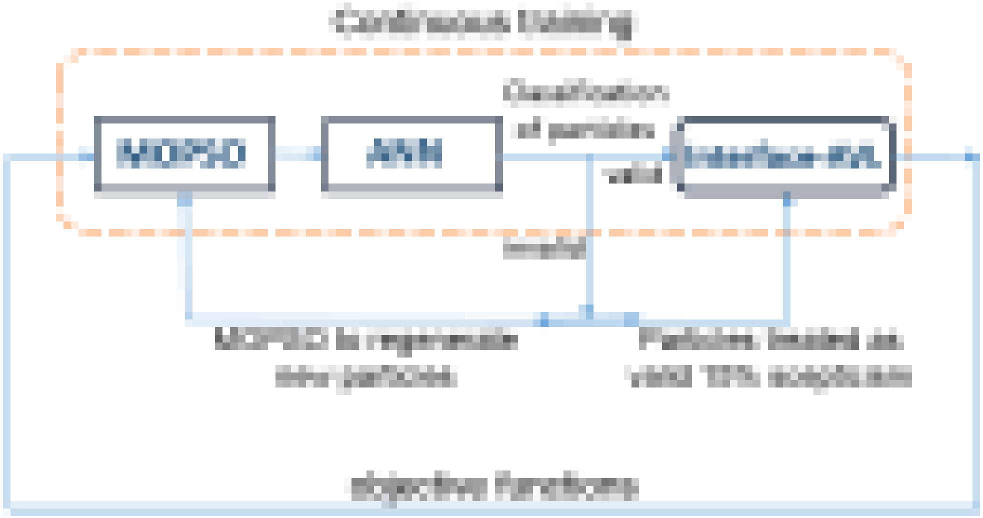

Continuous training
Classification of particles
MOPSO
ANN
Interface-AVL
valid
invalid
MOPSO to regenerate new particles
Particles treated as valid 15% scepticism
objective functions

Figure 4-8: ANN-MOPSO optimisation framework for continuous live training with 15% scepticism

When using a continuous live training approach for the ANN, the trial solutions suggested by the optimiser will be classified by the ANN as valid or invalid. The continuous live training of the ANN was performed in parallel with the evaluation of the objective function by the Interface-AVL. The minimum size of the training set to be provided for the ANN before starting classification was defined as 500.

Then, as the run continued, the ANN classified the trial solutions as valid or invalid. If the ANN classified a trial solution as valid (satisfy the constraints), the corresponding objective function was calculated. When the trial solutions were classified as invalid (does not satisfy the constraints) by the ANN two different actions were performed. First action, 15% (scepticism) of the invalid particles are treated as a valid particle and sent by the MOPSO to the Interface-AVL to be evaluated. The second action is the recalculations of position and velocity for the rest of invalid particles which are sent back to the ANN for testing before evaluation. By doing this, the computational time will be used only for evaluating the accepted trial solutions by the ANN, whereas the solutions rejected by the ANN are regenerated by the optimiser as a new trial solution with new velocity and position.

4.4.2 Overview of the used Artificial Neural Network

This work uses the ANN as described in [11], and which runs in parallel with the MOPSO. It is a ANN with multilayer perceptron composed of 100 hidden neural layers, each with 10 hidden nodes. The code was originally written in C++, which used the C++ library to help with the computationally intensive tasks [186]. However, to simplify the development of the tool, the interfacing layers have been written by Python. Most developments for the tool were for Python script layers, to avoid the core layer which is written in C++ [187]. Python is one of the most popular high-level languages, it contains a wide-ranging library that can be used in machine learning, data mining, and other scientific applications [188]. The ANN remains under active development by its authors, though many researchers have been able to improve the tool with new functions to suit their work [187].

For the learning process, the ANN used a limited memory Broyden Fletcher Goldfarb Shanno (L-BFGS) algorithm [11]. This is a straightforward self-optimisation method than can accelerate the pretraining process using a line search method. The L-BFGS algorithm used parallelism for computing the gradients on CPUs, GPUs, and computer clusters [189]. Generally, L-BFGS methods do not require manual tuning for the optimiser parameters to find a good convergence rate and so are considered stable and easy to use for training purposes.

4.4.3 Training of the ANN

The capability of the ANN to learn from a presentation of data that expresses the model behaviour is one of its best features. Once the ANN has learnt the existing relation between the inputs and outputs, it can generalise the solution which, in our case, means that the ANN can decide whether the trial solution by the optimiser (MOPSO) is worthy of full evaluation or not.

Various training processes and other issues regarding the ANN that have a significant effect on the optimisation performance are discussed below.

4.4.3.1 Initial training (from the archive) versus live training

Initial training means that the ANN will be trained from existing data. The archive should be prepared in advance so it can be used from an early stage of the optimisation. Using a well-distributed training archive may lead to discerning the bad particles very early, but compiling such a training archive is not always easy and is often expensive, especially for complex industrial problems. In contrast, with live training, as used in this work, the training data can be gathered and archived by the optimiser during the run in progress. When using live training, the decision must be made whether to use single or continuous training.

4.4.3.2 Continuous training versus single training

When doing live training, a consideration that has a significant influence on the optimisation performance is whether to spend all the optimisation time for retraining the ANN or to be satisfied with a shorter period of training and then use this archive of trained data to guide the optimisation until the end. Rawlins et al. [11] argued that time used to train the ANN after each iteration is negligible when is compared to the time required for the evaluation of the objective function for the problem. However, continuous training may lead to overtraining that may not improve the predictive ability of an ANN any more. Thus, it is desirable, it possible, to stop training once an acceptable size for the training archive has been reached.

Real-world problems and especially aerodynamic shape optimisation problems are highly constrained, and that is reflected in their being difficult to solve [12,13]. Thus, aerodynamic shape optimisation usually has many more invalid solutions than valid. For such design problems, using machine learning with continuous training is more efficient than using single training, especially with the knowledge that the time used to train the ANN after each iteration in continuous training is negligible when it is compared to the cost of the objective functions [11].

4.4.3.3 Size of the training set

The definition of the initial training set depends on the training approach adopted by the optimiser. If the training uses a pre-existing archive, the initial training set is the size of sample data drawn from the archive to train the ANN. On the other hand, for single live training, it is the acceptable size of the training set acquired by the optimiser.

In contrast, the size of the training set in the case of continuous training is considered to be the minimum size from the training archive considered to be a reliable source for training. The minimum size of the training set to be gained by the ANN before commencing classification is defined as 500 in this research [11].

4.4.3.4 Level of scepticism

The level of scepticism represents the level of suspicion that we have towards the particles that have been classified as invalid by the ANN. It depends mainly on the size of the archive and whether the training is single or continuous. For this work, 15% was selected, which is appropriate when using a large training set or continuous training approach [11]. Table 4-1 summarize the parameters used in the experiments carried out in this research programme.

Table 4-1: Summary of the parameter values used to perform the experiments

Parameter	Values
Archive used	Live
Training approach	Continuous
Initial training set	500
Scepticism	15%

4.5 Summary

Three advanced optimisation techniques that can be used to achieve good optimality solutions for the design of the aerodynamic shape of an aircraft have been investigated. First, an automated optimisation framework driven by Nimrod/O was introduced. In computing, Nimrod/O is considered an advanced tool for science and engineering modelling, a tool that allows a designer to explore many design scenarios since it is both fast and computationally inexpensive. In particular, the designer can allow the tool to search in combinations of design variables to find the optimal solution.

Second, the interactive optimisation approach - where the designer interacts with the optimisation process to steer the search to the ROI - was explored. By using such a technique, the designer can focus the search process in a particular area that satisfies the design requirements. Hence, this is a less time consuming and more flexible technique than an automated optimisation framework.

Finally, an ANN was used to guide the optimisation algorithm by deciding whether the trial solution is worthy of full evaluation or not, this increased the efficiency of the computing by achieving high optimality solutions in terms of the quality of the solutions and computational time required.

5 Chapter - Case Study Aegis UAV

The previous chapter describes in detail the methodology, and procedures used to perform aerodynamic shape design optimisation for an aircraft for each optimisation technique: automatic design tool, interactive optimisation, and Artificial Neural Network. In order to demonstrate the application and efficiency of the developed methodology for the aerodynamic shape design process, a case study is proposed. This chapter gives a brief introduction to the Aegis UAV, which has been used as a case study in this research. It is an unmanned aircraft still under development by a team from Cranfield University. The chapter begins by providing background information about the characteristics and previous attempts to develop the Aegis UAV. This is followed by investigation of the aerodynamic performance of the Aegis UAV base design including both U and inverted-V tail configurations. The chapter ends by formulating the design problem, defining the design space and the objective functions that will be used to improve the performance of the Aegis UAV using the developed framework. In order to maintain applicability, relevant design constraints will be applied.

The existing aerodynamic and mass data for the Aegis UAV platform presently in the Aerospace Integration Research Center (AIRC) Laboratory at Cranfield University, see Figure 5-1, is used as a continuous feedback to assist in developing the current methodology. The previously obtained in-flight data were computed using the Engineering Sciences Data Unit (ESDU) [190].

5.1 Description of the base design (Aegis UAV)

As stated above, the design test case used to demonstrate the effectiveness of the new methodology is the Aegis UAV. It is a project launched by Cranfield University four years ago to develop medium altitude and long endurance UAV for surveillance missions, and an initial base design configuration was built [190,191]. The design is not yet fully optimised, and there is room for further improvement and this will be done using optimisation techniques. This project is an integral part of that development.

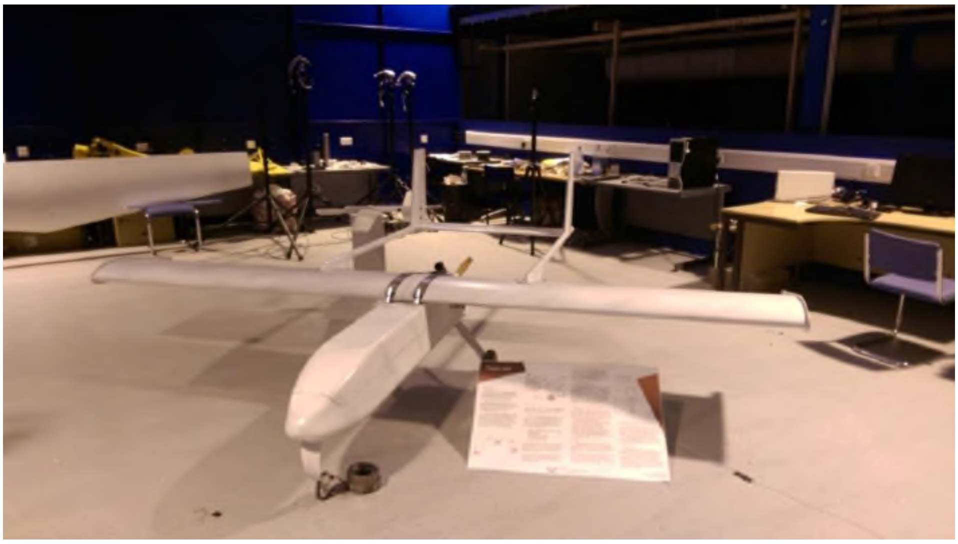


Figure 5-1: Photograph of model Aegis UAV with U-tail configuration at Cranfield University, the A1 sheet of paper gives an indication of model size

Figure 5-1 shows a photograph of the actual model of the Aegis UAV with U-tail configuration. It is design based upon the AAI RQ-2 Pioneer aircraft used by the USA and Israeli forces. The Aegis UAV has a twin boom pusher configuration, where the tail is mounted on two booms that extend from the wing upper surface. The aircraft utilises rectangular wings with one of two different tail configurations; U-tail and inverted V-tail, as shown in Figure 5-2. NACA 4415 and NACA 0013 airfoil sections were used for the wing and tail lifting surfaces, respectively. The Aegis UAV with U-tail configuration has a maximum take-off mass of 66 kg, and 64.5 kg for the Aegis UAV with inverted V-tail configuration. Both configurations are capable of carrying 10 to 15 kg payload.

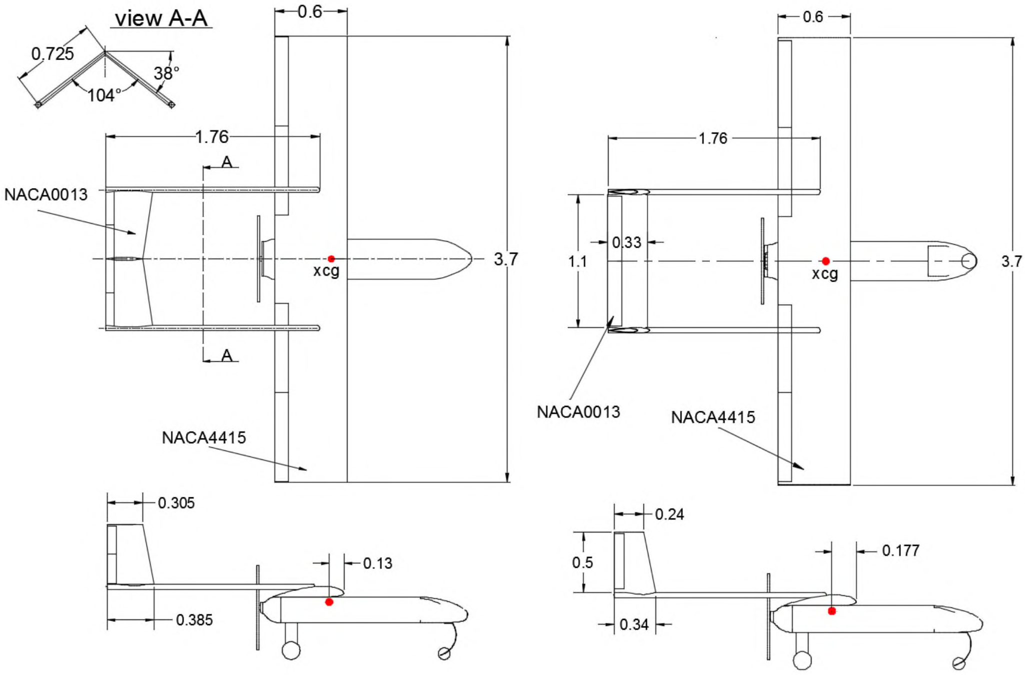


Figure 5-2: 2D Schematic diagram for Aegis UAV presenting detail dimensions of the wing and tail; left Inverted V-tail, and right U-tail. Dimensions are in meters

The operational altitude for the UAV is around 2000 m where the Reynolds number is around 1.5×10^6 based on the chord and Mach number around 0.13. Using AVL, the lift coefficients for level flight were found to be 0.3044 ($C_{L\ \alpha=0} = 0.3044$) at cruise speed equal to 43.60 m/s for the Aegis UAV with U-tail and 0.3047 ($C_{L\ \alpha=0} = 0.3047$) at cruise speed equal to 43.07 m/s for Aegis UAV with inverted V-tail.

5.2 The project ATHENA and previous attempts to develop the Aegis UAV

The first attempt to design the Aegis UAV began at Cranfield University with the launch of a project called ATHENA. The project aimed to use a different technological approach to designing a MALE (medium-altitude long-endurance) UAV called Ages. The first target of the project was to copy the Israeli MALE UAV, IAI Scout, an H-tail UAV with a rectangular wing. The rectangular wing used a Wortmann FX63-137 airfoil of span 4.96 m and a chord of 0.73 m. The study gave more attention to the tail shape; either the H-tail used by IAI Scout or an inverted V-tail. The operational point was defined by a velocity of 50 m/s where the chord-based Reynold number was 2.32x10^6 at zero angle of attack. The maximum take-off mass for this version was 150 kg [191]

A study was conducted to explore the aerodynamic characteristics of the flow around the UAV based on the geometry of the Scout. The UAV geometry was designed using the CATIA toolbox, and the flow around the UAV was studied using the commercial package ANSYS-FLUENT 12.0. Due to constraints on the available computational resources, and time constraints on the labour required to carry through the experimental procedures, only certain angles for the vertical tail inclination and V-tail angle were studied.

The studies found that there was no significant influence on the drag and lift of the Aegis UAV for different tail angles with either tail configuration. However, comparison of two specific geometries for the Aegis UAV, with H-tail and inverted V-tail, showed that the inverted V-tail angle was slightly more efficient by 10%. The studies showed that the wing is responsible for around 70% of the drag, with 20% generated by the fuselage. On the other hand, the wing is the main source of lift, contributing 90% to the lift, while the fuselage generates 9%.

The studies showed clearly that the overall drag for the UAV was higher than expected. The study concluded that an improvement should be made to the wing to reduce overall drag. To reduce the drag while keeping enough lift to maintain the Aegis UAV in level flight, two different airfoil geometries (Wortmann Fx63-133 and Selig S1210) and a limited number of wing planform parameters were investigated in an attempt to improve the aerodynamic characteristics [192]. The work was performed using the CFD software FLUENT 13.0.

The study investigated only the following wing planforms to reduce overall drag; reducing the chord length from 0.73 m to 0.60 m, and reducing wingspan from 4.960 m to 4.830 m, 4.464 m, and 3.830 m. The results showed that reduction in wing span and chord has a significant effect on reduction of overall drag. For example, reducing the wingspan by 10% and 20% caused a 5.7 N and 8.3 N reductions in the overall drag, respectively. Moreover, the drag force reduced by 25% when the chord length was reduced to 0.6 m. The work also looked at tapering the wing to provide better aerodynamic performance. Due to the design resource limitations, only a wing taper ratio of 0.6 was investigated. However, that study showed that the nose design of the body makes a significant contribution to the total drag, and has been identified as an area for future improvement.

It was evident that the procedure followed in studying and developing the first version of the Aegis UAV (MTOM 150 kg) used a process that limited the designer to scanning only a certain number of alternative designs. Within that procedure, the possibility of performing a complete evaluation of all the possible design scenarios necessary to achieve an optimal design, was not possible. Arora [47] argued that the only way to ensure a true optimal solution was by automating the optimisation process using efficient computational design optimisation methods, able to execute intensive optimisation processes and produce discrete design scenarios for various combinations of the design variables through the coupling of an aerodynamic design code with an optimisation algorithm [28][29]. However, in the conceptual design phases it is not recommended to use high fidelity analysis design tools such as CFD packages as they can be unnecessarily costly [5][8]. On the other hand, once the aircraft configuration and propulsion systems have been assessed within the conceptual design phase, the use of high fidelity software is recommended to ensure a proper fit between the performances of different systems [10].

The next stage was an attempt to design and develop the Aegis UAV through the ATHENA project based upon the AAI RQ-2 Pioneer aircraft. A prototype was built and a six-degree-of-freedom (6-DOF) model of the Aegis UAV was constructed based on MATLAB and Simulink. The work included the break-down of the masses of the major components in the Aegis UAV, and flight data evaluation using ESDU [190]. To overcome the shortcomings of the design procedure used with the first version of the Aegis UAV, the methodology developed here used advanced optimisation techniques to improve aerodynamic performance by providing correct support decisions for the design optimisation process.

5.3 Problem formulation

To formulate the design problem, the design space and object functions are defined. The design parameters and their upper and lower bounds are shown in Table 5-1. Two different objectives are included, maximise the endurance ratio $\left(C_L^{1.5}/C_D\right)$ and minimize the structural mass. However, since all the solvers attempt to minimise the objective functions, the endurance ratio is redefined as $-\left(C_L^{1.5}/C_D\right)$. Ideally, the endurance ratio will be an optimum, i.e. maximise the term $\left(C_L^{1.5}/C_D\right)$, by using suitable flight conditions and configurations [30,43,193]. This is a single-point optimisation under nominal flight conditions, where the optimiser is seeking to minimise the drag coefficient by varying the shape design variables, subject to $C_L = C_{L\_BD}^{*}$. However, the aim in future is to extend the work to perform multipoint aerodynamic shape optimisation [122,194].

On the other hand, the total mass of the UAV is the sum of the masses of all the subsystems, including the frame structure, propulsion system and payloads, and is parameterized in terms of aircraft wing, boom, and tail design variables, see Section 5.3.1.

Table 5-1: Design variables and their upper and lower bounds for Aegis UAV configurations

Parameters		Lower Bounds		Base design		Upper Bounds	
		U-tail	Inverted V-tail	U-tail	Inverted V-tail	U-tail	Inverted V-tail
b_w	[m]	2.59	2.59	3.7	3.7	4.81	4.81
C_{r_w}	[m]	0.42	0.42	0.6	0.6	0.78	0.78
λ_w	[-]	0.6	0.6	1.0	1.0	1.0	1.0
V_{ht}	[-]	0.35	n/a	0.43	n/a	0.55	n/a
V_{vt}	[-]	0.02	n/a	0.029	n/a	0.035	n/a
L_t	[m]	1.45	1.45	1.58	1.58	2.0	2.0
AR_{ht}	[-]	3.0	n/a	3.33	n/a	4.0	n/a
AR_{vt}	[-]	1.5	n/a	1.69	n/a	2.5	n/a
λ_{vt}	[-]	0.5	n/a	0.68	n/a	1.0	n/a
F_{TV}	[-]	n/a	0.13	n/a	0.19	n/a	0.25
AR_f	[-]	n/a	1.5	n/a	2.1	n/a	2.5
λ_f	[-]	n/a	0.65	n/a	0.79	n/a	1.0
ϕ_t	[deg]	n/a	95.0	n/a	104	n/a	120.0
ht_angle	[deg]	-6.0	n/a	0.0	n/a	6.0	n/a

5.3.1 Mass model

During the aerodynamic optimisation process the dimensions of the wings, booms, and tails will change according to the new values of the design variables. This will affect the weight of the component as well as overall UAV weight. Since one of our objectives is to reduce the overall weight, a method or procedure must be available to recalculate the new UAV weight depending on its design. The airframe and lifting surfaces are a carbon fibre monocoque [195]. The mass for the wing and tail surfaces are parameterized as a function of surface area unit mass while the mass of the boom is parameterized as a function of mass per unit length.

5.3.1.1 Wing mass

For the wing, the mass per unit area ($\mathrm{M_{unit}}$) for the base design was calculated as:

$$\mathrm{M_{unit}} = \frac{\mathrm{M_{w-BD}}}{\mathrm{S_{W-BD}}} \tag{5-1}$$

Where:

$\mathrm{M_{w-BD}}$: Wing mass of the base design.

$\mathrm{S_{W-BD}}$: Wing surface area for the base design.

The mass for the optimised wing ($\mathrm{M_{new\_wing}}$) was calculated as:

$$\mathrm{M_{new\_wing}} = \mathrm{M_{unit}} * \mathrm{S_{w-optimised}} \tag{5-2}$$

Where $\mathrm{S_{w-optimised}}$ is the wing surface area for the optimised UAV, and defined as:

$$\mathrm{S_{w-optimised}} = \left[\frac{\mathrm{b_w}}{2}\left(\mathrm{C_{r\_w}} + \mathrm{C_{t\_w}}\right)\right] \tag{5-3}$$

5.3.1.2 U-tail mass

The U-tail shape has two components, a horizontal tail and vertical tail. The horizontal tail mass per unit area ($\mathrm{M_{unit}}$) was defined as:

$$\mathrm{M_{unit}} = \frac{\mathrm{M_{ht\_BD}}}{\mathrm{S_{ht\_BD}}} \tag{5-4}$$

Where:

$\mathrm{M_{ht\_BD}}$: Mass of the horizontal tail for the base design.

$\mathrm{S_{ht\_BD}}$: Surface area of the horizontal tail for the base design.

Then the mass for the optimised horizontal tail was computed using:

$$\mathrm{M_{new\_ht} = M_{unit} * S_{ht-optimised}} \quad (5\text{-}5)$$

Where $\mathrm{S_{ht-optimised}}$ was the surface area of the horizontal tail for the optimised UAV, and defined as:

$$\mathrm{S_{ht-optimised} = bht * Cht} \quad (5\text{-}6)$$

On the other hand, the unit mass of the vertical tail was calculated as:

$$\mathrm{M_{unit} = \frac{M_{vt\_BD}}{S_{vt\_BD}}} \quad (5\text{-}7)$$

Where:

$\mathrm{M_{vt\_BD}}$: Mass of the vertical tail for the base design.

$\mathrm{S_{vt\_BD}}$: Surface area of the vertical tail for the base design.

Then the mass for the optimised horizontal tail was computed using:

$$\mathrm{M_{new\_vt} = \left(M_{unit} * S_{vt-optimised}\right) * 2} \quad (5\text{-}8)$$

Where $\mathrm{S_{vt-optimised}}$ is the surface area of the vertical tail for the optimised UAV, and defined as:

$$\mathrm{S_{vt-optimised} = \left[\frac{bvt}{2}(Ct\_vt + Cr\_vt)\right]} \quad (5\text{-}9)$$

The vertical tail was optimised with the possibility of obtaining a rectangular or trapezoidal shape.

5.3.1.3 Inverted V-tail mass

The inverted V-tail shape has two symmetrical fins. The tail fin planform was optimised with the possibility of obtaining a rectangular or trapezoidal shape.

$$\mathrm{M_{unit} = \frac{M_{f\_BD}}{S_{f\_BD}}} \quad (5\text{-}10)$$

Where:

$\mathrm{M_{f\_BD}}$: Mass of the inverted V-tail for the base design.

$\mathrm{S_{f\_BD}}$: Surface area of the inverted V-tail for the base design.

Then the mass for the optimised fin was calculated using:

$$\mathrm{M_{new\_f}} = \left(\mathrm{M_{unit}} * \mathrm{S_{f-optimised}}\right) * 2 \qquad (5\text{-}11)$$

Where $\mathrm{S_{f-optimised}}$ is the surface area of the inverted V-tail for the optimised UAV, and defined as:

$$\mathrm{S_{f-optimised}} = \left[\frac{\mathrm{b\_tail}}{2}(\mathrm{Ct\_tail} + \mathrm{Cr\_tail})\right] \qquad (5\text{-}12)$$

5.3.1.4 Boom mass

The booms are long hollow tubes and the only the length was changed during the optimisation process. Therefore, the optimised mass could be defined as a function of mass per unit length

$$\mathrm{M_{unit}} = \frac{\mathrm{M_{boom\_BD}}}{\mathrm{L_{boom\_BD}}} \qquad (5\text{-}13)$$

Where:

$\mathrm{M_{boom\_BD}}$: Mass of the boom for the base design.

$\mathrm{L_{boom\_BD}}$: Length of the boom for the base design.

The mass for the optimised boom was calculated as:

$$\mathrm{M_{new\_boom}} = \left(\mathrm{M_{unit}} * \mathrm{L_{boom-optimised}}\right) * 2 \qquad (5\text{-}14)$$

Where $\mathrm{L_{boom-optimised}}$ is the length of the boom for the optimised UAV

Finally, the mass ($\mathrm{UAV\ mass}$) of the new optimised configurations was defined as:

$$\begin{aligned}\mathrm{UAV\ mass} = {} & \left[\mathrm{M_{base\_UAV}} - \left(\mathrm{M}_{base\_wing} + \mathrm{M_{base\_vt}} + \mathrm{M_{base\_ht}} + \right.\right. \\ & \left.\left.\mathrm{M_{base\_f}} + \mathrm{M_{base\_boom}}\right)\right] + \left(\mathrm{M_{new\_wing}} + \mathrm{M_{new\_ht}} + \right. \\ & \left.\mathrm{M_{new\_vt}} + \mathrm{M_{new\_f}} + \mathrm{M_{new\_boom}}\right) \qquad (5\text{-}15)\end{aligned}$$

5.3.2 Formulation procedure

Prior to the formulation of the design problem, a parametric study was performed using the SM to explore the design space, as explained in Section 4.1.3. On the other hand, the MOTS algorithm setting was investigated by performing several runs with different numbers of regions and evaluations [83], since using as few evaluations as possible to find the optimal solutions is very important, because each evaluation could require a significant amount of CPU time. As expected the number of evaluations had a significant impact on the computational time [35], while increasing the number of regions forced the algorithm to explore more areas of the design space [82]. A compromise between number of evaluations and number of regions was made: five regions and evaluations ranging from 1300 to 5500, the compromise depended on the number of design variables.

Two main procedures were used to optimise the Aegis UAV configuration; using only wing design variables, and then using wing-tail design variables simultaneously. However, because the UAV needed to be trimmed during flight, the horizontal tail rotation angle is added as an additional design variable in the case of the UAV with the U-tail shape, see Table 5-2. Thus, initially six-design cases were performed to demonstrate the effectiveness of the methodology and gain a better understanding of the design requirements for the Aegis UAV.

In case 1, the UAV with U-tail shape was optimised using wing design variables subject to pitching moment constraints. In case 2, the horizontal-tail rotation angle and trimming constraints instead of pitching moment constraints, are added to case 1. In case 3, the UAV with U-tail is optimised with the simultaneous inclusion of wing and tail design variables, while considering pitching moment constraints. In case 4, the horizontal-tail rotation angle design variable is added to case 3 and trimming constraint instead of pitching moment constraints. In case 5, the UAV with inverted V-tail shape was optimised using wing design variables subject to pitching moment constraints. In case 6, the tail design variables were added to design case 5.

Table 5-2: Summary of all design optimisation cases for Aegis UAV with U-tail and Inverted V-tail shapes

Case	Geometry design variables	Pitching constraints	UAV
1	wing	base design pitching moment	U-tail
2	wing & horizontal-tail rotation angle	trimming constraint Cm=0	U-tail
3	Wing & tail	base design pitching moment	U-tail
4	wing, tail & horizontal-tail rotation angle	trimming constraint Cm=0	U-tail
5	wing	base design pitching moment	Inverted V-tail
6	wing & tail	base design pitching moment	Inverted V-tail

Actually, before formulating the design optimisation problem, a study was performed to explore the Aegis UAV design space and its characteristics for level flight. Simulation of each configuration as a function of the angle of attack was performed. The study found that the Aegis UAV base design was not balanced: i.e., the pitching moment (Cm) about the centre of gravity of the UAV should be equal to zero for level flight. A study was performed to explore the possibility of changing the centre of gravity so Cm was equal to zero.

The common methods to achieve a trimmed UAV during level flight are: (1) change location of centre of gravity by mass redistribution, (2) adjust the horizontal tail incident angle, (3) adjust horizontal tail volume, and (4) adjust elevator deflection. Each of these is presented in detail in Appendix C. and it was concluded that, Cm=0 can be satisfied by the optimiser only by rotation of the horizontal tail (case 2 and case 4) or by the elvevator deflection during flight, see Appendix C.

5.3.3 Using wing design variables (Cases 1 and 5)

In this case, only wing planform parameters were used to obtain a set of optimal wing configurations for the Aegis UAV with both U-tail and inverted V-tail for steady flight (cases 1 and 5). The formulation of the design problem was as follows:

$$\mathrm{min.\,f(\,x\,)} \qquad = \qquad \min \begin{Bmatrix} -\frac{C_L{}^{1.5}}{C_D} \\ \mathrm{UAV\ mass} \end{Bmatrix} \tag{5-16}$$

$$\text{subject to} \qquad V_{st}(x) - V_{st}^{*} \leq 0$$

$$Cm(x) \geq Cm_{BD}^{*}$$

$$-V_{max}(x) + V_{max}^{*} \leq 0$$

$$x_l \leq \quad x \quad \leq x_u$$

$$\text{where} \quad x = \begin{bmatrix} b_w & C_{r\_w} & \lambda_w \end{bmatrix}^T$$

(Case 2):

To account for the fact that the UAV must be trimmed during flight, the horizontal tail rotation angle was used as an additional design variable for the Aegis UAV with U-tail shape **(case 2)**. The trimming was achieved by applying a pitching moment constraint, $Cm = 0$, instead of $Cm(x) \geq Cm_{BD}^{*}$ in Equation (5-16), and the design variables for the new case were defined as:

$$x = \begin{bmatrix} b_w & C_{r\_w} & \lambda_w & \mathrm{ht\_angle} \end{bmatrix}^T \qquad \text{(in case of U-tail only)} \tag{5-17}$$

5.3.4 Using wing and tail design variables simultaneously for Aegis UAV with U-tail shape (Case 3)

In order to obtain a feasible solution for the full UAV configuration by the aerodynamic shape optimisation process, the optimisation problem should include wing and tail design variables simultaneously (case 3), since an interaction exists between aerodynamic efficiency and stability characteristics [43,45,46]. The trade-off in any wing-tail design is between producing the desired stability and control moments versus weight and parasitic drag. To address these issues, an optimisation was performed in which wing-tail design variables were used simultaneously to obtain a set of optimised configurations. Nine design variables were used to optimise the UAV with U-tail shape; wingspan, wing root chord, wing taper ratio, horizontal tail volume, vertical tail volume, tail arm, horizontal tail aspect ratio, vertical tail aspect ratio, and vertical tail taper ratio, see Figure 5-3. The horizontal tail geometry is parameterized as a rectangular shape. The optimiser allows a horizontal tail with various aspect ratios. Since the horizontal tail should not stall before the wing [41], the horizontal tail aspect ratio was limited to being lower than the wing's aspect ratio. This technique allowed for a horizontal tail with a limited range of span and chord values which were calculated as a function of the aspect ratio using:

$$\mathrm{Cht} = \frac{\mathrm{bht}}{\mathrm{AR_{ht}}} \qquad (5\text{-}18)$$

$$\text{where} \qquad \mathrm{bht} = \sqrt{\mathrm{AR_{ht}} * \mathrm{S_{ht}}} \qquad (5\text{-}19)$$

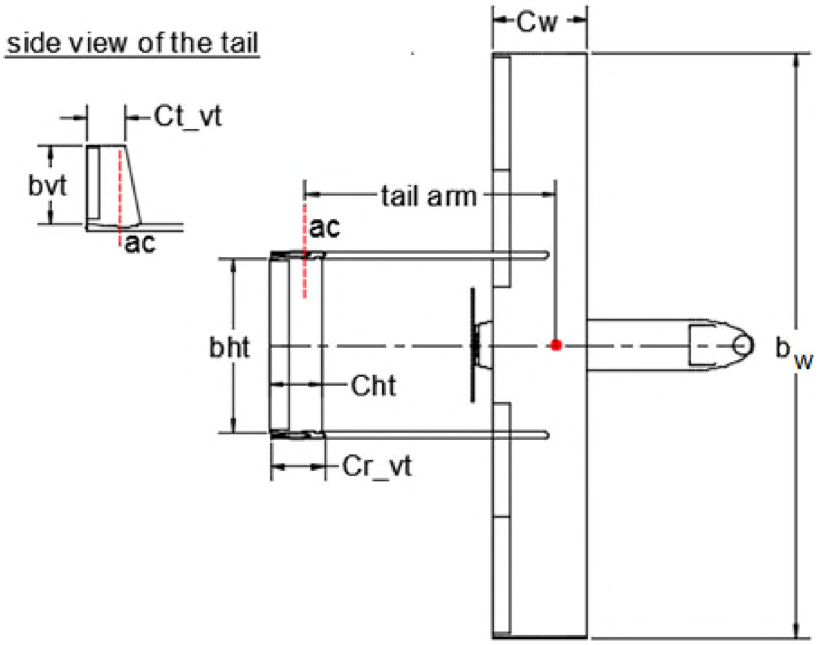


Figure 5-3: Definition of design variables for Aegis UAV with the U-tail shape, see also Figure 5-1

In order to calculate horizontal tail surface area, the horizontal tail volume and tail arm were used as design variables. It was a trade-off between obtaining the required lift, minimizing the drag, and satisfying stability requirements. Since the propeller diameter was 711 mm (28.0 inch), the minimum horizontal tail span value 1000 **mm** was used as a geometrical constraint to preserve enough clearance for the propeller.

$$\mathrm{S_{ht}} = \frac{\mathrm{V_{ht}*S_w*MAC_w}}{\mathrm{L_t}} \tag{5-20}$$

On the other hand, the vertical tail geometry was parameterized with the possibility of there being either a rectangular or trapezoidal configuration. This formulation allowed various degrees of design freedom for the optimiser. The vertical tail span and the surface area were defined by:

$$\mathrm{bvt} = \sqrt{\mathrm{AR_{vt}} * \mathrm{S_{vt}}} \tag{5-21}$$

$$\text{where} \qquad \mathrm{S_{vt}} = \frac{\mathrm{V_{vt}*S_w*b_w}}{\mathrm{L_t}} \tag{5-22}$$

The vertical tail taper ratio and aspect ratio were used as design variables to calculate the resultant root and tip chord of the vertical tail. To prevent the possibility of a significant reduction in lift coefficient that could lead to tip stall, the tip chord is assigned a minimum value of 240 mm [196,197]. Also, defining the minimum value of the vertical tail tip chord encouraged the optimiser towards either decreasing or increasing the taper ratio. Increasing vertical tail taper ratio (minimizing root chord) allowed for more efficient aerodynamic characteristics, while reduction in vertical tail taper ratio would result in a reduction of structural mass per unit area.

$$\mathrm{Ct\_vt} = \lambda_{\mathrm{vt}} * \mathrm{Cr\_vt} \tag{5-23}$$

$$\text{where} \qquad \mathrm{Cr\_vt} = \frac{2*\mathrm{bvt}}{\mathrm{AR_{vt}}(1+\lambda_{\mathrm{vt}})} \tag{5-24}$$

The formulation of the design problem was as follows:

$$\mathrm{min.\,f(\,x\,)} \qquad = \qquad \min\left\{\begin{matrix} -\frac{\mathrm{C_L}^{1.5}}{\mathrm{C_D}} \\ \text{UAV mass} \end{matrix}\right\}, \tag{5-25}$$

$$\text{subject to} \qquad\qquad \mathrm{Cm(x)} \geq \mathrm{Cm^{*}_{BD}}$$

$$V_{st}(x) - V_{st}^{*} \le 0$$

$$-V_{max}(x) + V_{max}^{*} \le 0$$

$$bht \ge 1.0 \text{ (geometric constraint)}$$

$$Ct\_vt \ge 0.24 \text{ (geometric constraint)}$$

$$Cm_{\alpha} \le 0, \quad Cn_{b} \ge 0, \quad Cl_{b} \le 0, \quad Cm_{q} \le 0, \quad Cn_{r} \le 0, \text{ and}$$

$$x_{l} \le \quad x \quad \le x_{u}$$

$$x = \left[b_{w} \quad C_{r\_w} \quad \lambda_{w} \quad V_{ht} \quad V_{vt} \quad L_{t} \quad AR_{ht} \quad AR_{vt} \quad \lambda_{vt}\right]^{T}$$

(Case 4):

To consider the trim drag generated by the tail, one more design case is considered (case 4) where one more design variable was added; the horizontal tail rotation angle. This was to consider optimising the configurations using Cm=0 instead of $Cm(x) \ge Cm_{BD}^{*}$ in Equation (5-25), with the following design variables:

$$x = \left[b_{w} \quad C_{r\_w} \quad \lambda_{w} \quad V_{ht} \quad V_{vt} \quad L_{t} \quad AR_{ht} \quad AR_{vt} \quad \lambda_{vt} \quad ht\_angle\right]^{T} \qquad (5\text{-}26)$$

5.3.5 Using wing and tail design variables simultaneously for Aegis UAV with inverted V-tail shape (Case 6)

An optimisation was performed on the Aegis UAV with inverted V-tail shape in which the following wing-tail design variables were used simultaneously; wingspan, wing root chord, wing taper ratio, tail volume, tail arm, inverted tail aspect ratio, inverted tail taper ratio, and inverted V-tail angle. Figure 5-4 shows the inverted V-tail configuration with the design variables. Since the lift generated by the fin of the V-tail was not directly upwards or downwards, the lift forces were reduced. The vertical component of the lift was cos(dihedral angle) multiplied by total lift. To compensate for this reduction, the inverted V-tail area must be 1/cos(dihedral angle) larger. Furthermore, the inverted V-tail area must be enlarged by a second 1/cos(dihedral angle) factor to compensate for the reduction in response to the angle of attack [198]. The effective fin area was defined as:

$$\mathrm{Effective}_{\mathrm{area}} = \mathrm{S}_{\mathrm{tail}} * \cos^2(\Gamma\_\mathrm{tail}) \qquad (5\text{-}27)$$

$$\mathrm{S}_{\mathrm{tail}} = \frac{\mathrm{F}_{\mathrm{TV}} * \mathrm{Cw} * \mathrm{S}_{\mathrm{W}}}{\mathrm{L}_{\mathrm{t}} * \cos^2(\Gamma_{\_\mathrm{tail}})} \qquad (5\text{-}28)$$

$$\text{where} \quad \Gamma\_\mathrm{tail} = \frac{180 - \phi_{\mathrm{t}}}{2} \qquad (5\text{-}29)$$

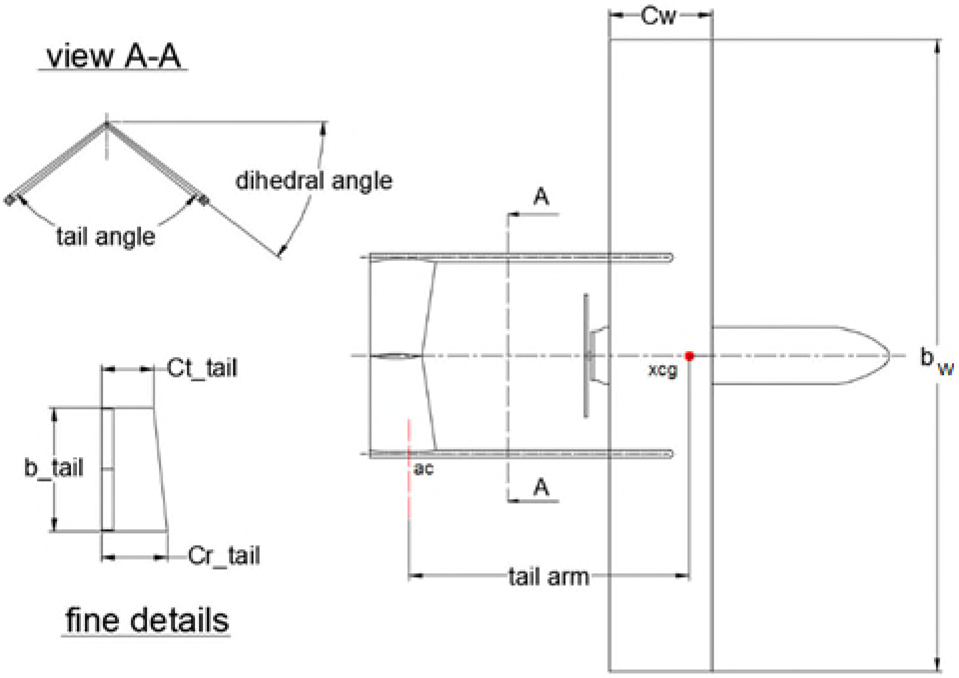


Figure 5-4: Definition of design variables for Aegis UAV with the Inverted V-tail shape, the Γ_tail is the dihedral angle in the figure

By using the aspect ratio and the taper ratio as design variables, the fin planform area was optimised with the possibility of obtaining a rectangular or trapezoidal shape. The resulting tip chord and fin span were defined by:

$$\mathrm{Ct\_tail} = \frac{2 * \mathrm{b\_tail}}{\mathrm{AR}_{\mathrm{f}} * \left(\left(\frac{1}{\lambda_{\mathrm{f}}}\right) + 1\right)} \qquad (5\text{-}30)$$

$$\text{where} \quad \mathrm{b\_tail} = \sqrt{(\mathrm{AR}_{\mathrm{f}} * \mathrm{S}_{\mathrm{tail}})} \qquad (5\text{-}31)$$

As previously explained, the fin tip chord is limited to a specific value to preserve the cruise Reynolds number. The fin root chord was calculated as a function of taper ratio:

$$\mathrm{Cr\_tail} = \frac{\mathrm{Ct\_tail}}{\lambda_f} \tag{5-32}$$

To ensure optimising an appropriate inverted V-tail shape, a number of geometrical constraints were placed on the design variables. These concerned mainly; (i) downflow from the propeller onto the tail surface, (ii) allowing sufficient clearance between the propeller and the booms, and (iii) satisfying the stability requirements [199]. In order to fulfil constraints applied to the fin shape, the optimiser continuously generated horizontal and vertical projection surfaces for the current fin to calculate its projection span. The problem was formulated as:

$$\min.\mathrm{f}(\,\mathrm{x}\,) \quad = \quad \min\left\{\begin{matrix} -\frac{{C_L}^{1.5}}{C_D} \\ \text{UAV mass} \end{matrix}\right\}, \tag{5-33}$$

$$\text{subject to} \qquad \mathrm{Cm(x)} \geq \mathrm{Cm}^{*}_{\mathrm{BD}}$$

$$\mathrm{V_{st}(x)} - \mathrm{V^{*}_{st}} \leq 0$$

$$-\mathrm{V_{max}(x)} + \mathrm{V^{*}_{max}} \leq 0$$

$$\mathrm{b\_hp} \geq 0.5 \text{ (geometric constraint)}$$

$$\mathrm{b\_vp} \geq 0.257 \text{ (geometric constraint)}$$

$$\mathrm{b\_vp} \leq 0.445 \text{ (geometric constraint)}$$

$$\mathrm{Cm}_{\alpha} \leq 0, \quad \mathrm{Cn_b} \geq 0, \quad \mathrm{Cl_b} \leq 0, \quad \mathrm{Cm_q} \leq 0, \quad \mathrm{Cn_r} \leq 0$$

$$\mathrm{x_l} \leq \mathrm{x} \leq \mathrm{x_u}$$

$$\mathrm{x} = \left[\mathrm{b_w} \quad \mathrm{C_{r\_w}} \quad \lambda_{\mathrm{w}} \quad \mathrm{F_{TV}} \quad \mathrm{L_t} \quad \mathrm{AR_f} \quad \lambda_{\mathrm{f}} \quad \phi_{\mathrm{t}}\right]^{\mathrm{T}}$$

5.4 Summary

This chapter has reported the case studies used to demonstrate the effectiveness of the proposed methodology for the design of the Aegis UAV twin-boom pusher. This chapter included a description of the base design, problem formulation procedure, and problem formulation. The Aegis UAV configurations were investigated and analysed to provide the required support for the formulation of the design problem. Two different objectives were included; to maximise the endurance and minimize the structural mass. These were obtained by varying wing and tail design parameters. The formulation considered six design scenarios. In order to maintain the applicability of the design problem, the design space and various constraints were defined.

The next chapter will provide results obtained from an optimisation investigation of the Aegis UAV configurations using the automated optimisation approach.

6 Chapter - Results, Observations and Discussion of the Automated Approach

This chapter presents results, observations, and discussion of the strategy used to enhance the design optimisation process through the use of the computational tool, Nimrod/O. Prior to the optimisation process, a parametric study was performed using SM to explore the design space and identify any design limitations. The analysis and investigation of the SM results were used as feedback to better formulate the design problem, as explained above.

6.1 Parametric sweep study and results

Prior to the formulation of the design problem, a parametric study was performed to (i) explore the design space, (ii) provide an insight into how the objective functions behave with respect to the design variables, and (iii) validate the Interface-AVL results. It takes all combinations of the variables to explore the entire design space. Whether the design variables have positive or conflicting effects on the objective functions becomes clear after performing the parametric study. If the objectives do not conflict with each other as the design variables change, a single optimal solution exists that can be a solution for the design optimisation problem. In contrast, if the objectives conflict, no single solution exists that simultaneously optimises each objective [200]. The study started by using wing design variables only, see Equation (5-16).

The sweep steps were in steps of 0.1 m, 0.01 m, and 0.0075 for the design variables; span $(\mathrm{b_w})$, root chord $\left(\mathrm{C_{r\_w}}\right)$, and taper ratio (λ_w), respectively. The sweep process was performed seven times to cover all design variable combinations. Firstly, the sweep study began by sweeping each design variable independently to display any dysfunction that may exist in the framework, and then sweeping different design variables in combinations. The first result of the parametric sweep for level flight subject to a constant lift coefficient ($\mathrm{C_L} = \mathrm{C^*_{L\_BD}}$, forced within the flow solver) showed a slight oscillation existed in the endurance ratio. By improving the quality of panel discretization for the lifting surfaces, design variables sweep steps, and precision of some variables in the code, the results were improved and the graph became smoother, see Figures 6-1 to 6-3.

It is obvious from Figure 6-1 that increasing wingspan (wing root chord and taper ratio fixed at 0.6 m and 1.0, respectively) will reduce the endurance ratio (i.e. improvement).

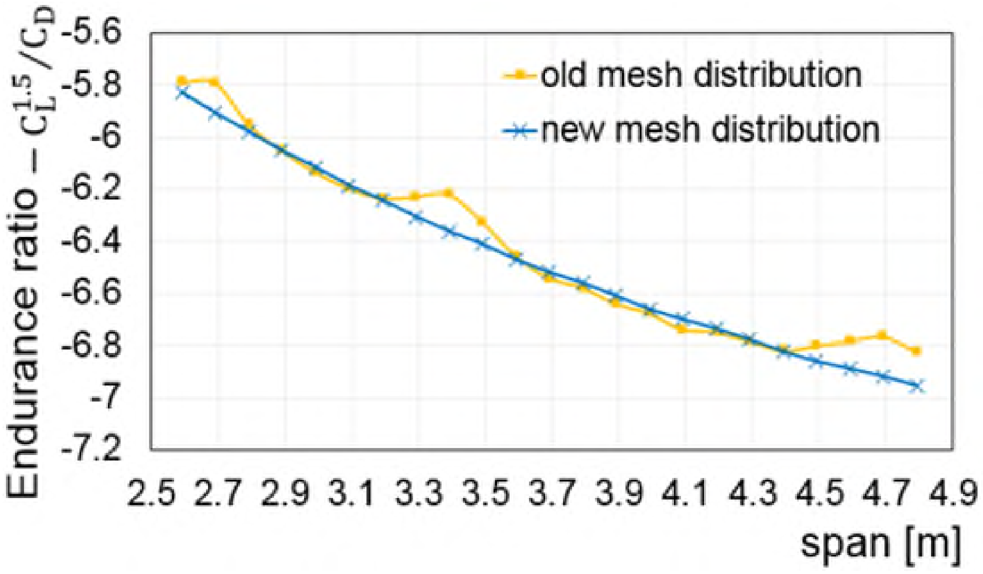


Figure 6-1: Sweep results before and after panel discretisation and accuracy improvements for wingspan design variable

On the other hand, Figure 6-2 shows that increasing the root chord length (wingspan and taper ratio fixed at 3.7 m and 1.0, respectively) caused the endurance ratio to pass through a minimum value at a chord length of about 0.65 m, and then increase. Similarly, Figure 6-3 shows that increasing the taper ratio (wingspan and root chord fixed at 3.7 m, and 0.6, respectively) the endurance ratio passed through a minimum value at a taper ratio of about 0.75 but then increases.

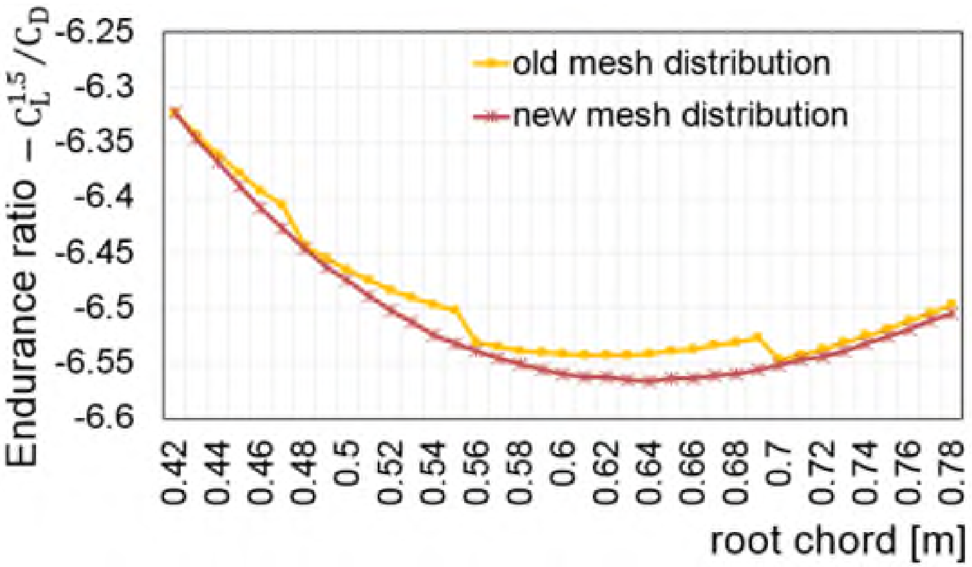


Figure 6-2: Sweep results before and after panel discretisation and accuracy improvements for wing root-chord design variable

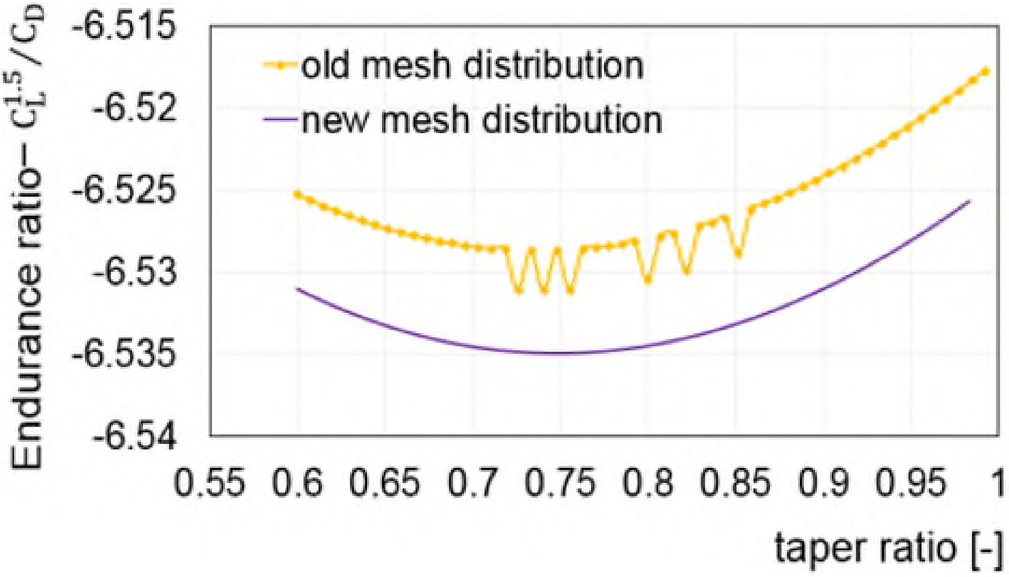


Figure 6-3: Sweep results before and after panel discretisation and accuracy improvements for wing taper-ratio design variable

To check whether the design parameters have conflicting effects on the objective functions or not, each design variable is allowed to vary as a function of both objectives as shown in Figures 6-4 to 6-6. It is clear that there is a conflict when optimising the wingspan design variable as a function of both of the objectives. This can be seen, for example in Figure 6-4. As the span is increased one of the objectives (endurance ratio) decreased, the other (UAV mass) increased. However, the aim is to minimise both of these objectives. Thus, a conflict exists, and Pareto optima should be used to optimise the wingspan.

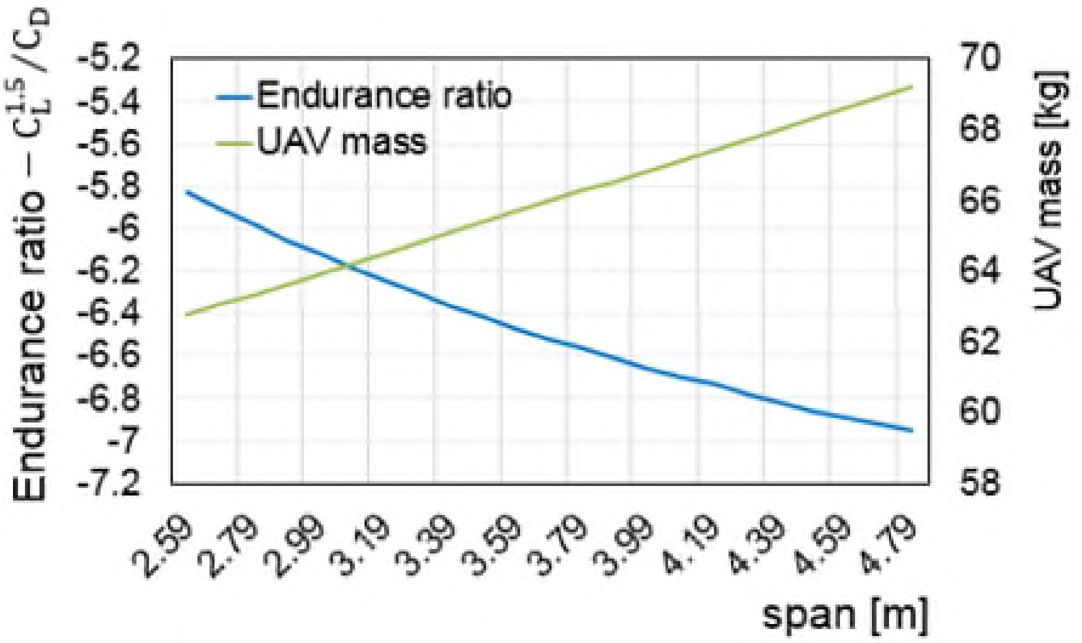


Figure 6-4: Design variables varying with both objectives for wingspan

On the other hand, as seen in Figures 6-5 and 6-6, increasing either root chord or taper ratio will cause UAV total mass to increase monotonically, but will have a varying impact on the endurance ratio. Sweeping the root chord in steps of 0.01 m, gradually improved the endurance ratio until the wing root chord equalled 0.64 m, after which it slowly declines. Sweeping the taper ratio in steps of 0.0075 produced a maximum absolute value of endurance ratio when taper ratio was about 0.76. Thus, optimising the wing geometry as a function of these three design variables is a trade-off problem. Such a problem requires the correct support for the DM to perform the trade-off process accurately.

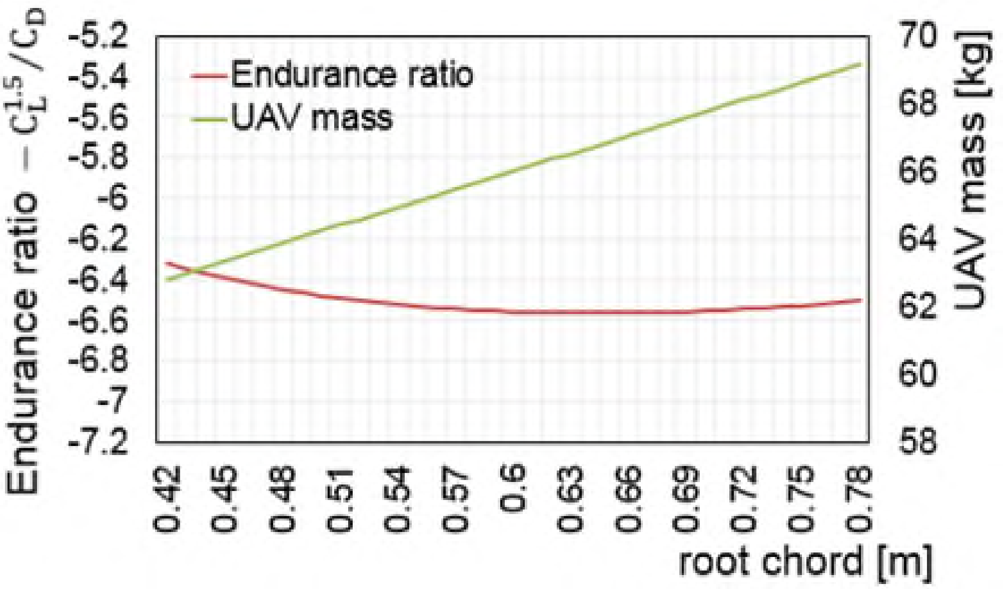


Figure 6-5: Design variables varying with both objectives for wing root-chord

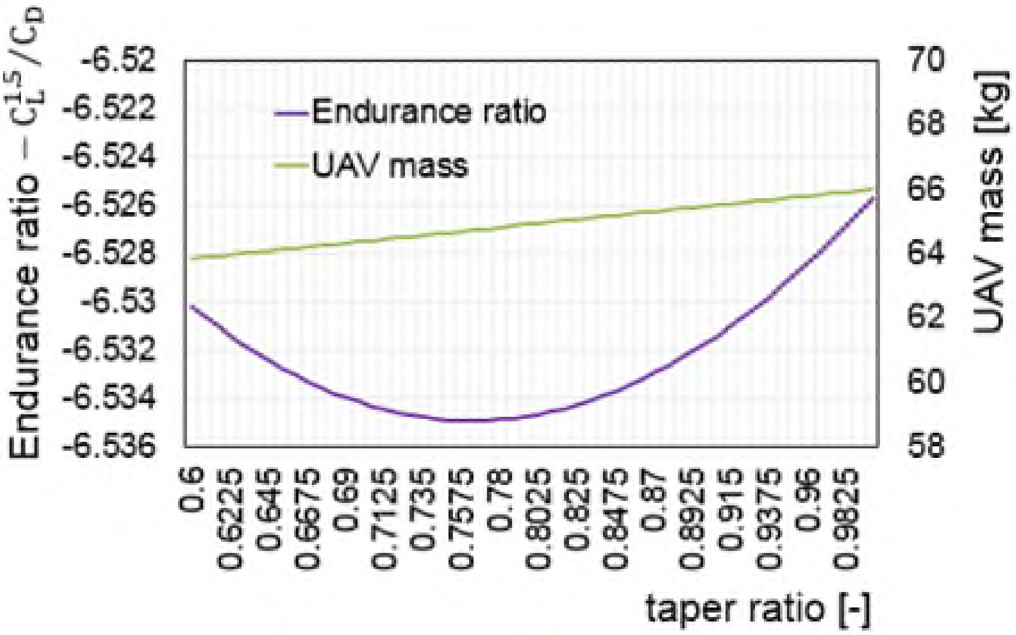


Figure 6-6: Design variables varying with both objectives for wing taper-ratio

The sweep process was performed by the simultaneous sweeping of the design variables in various combinations (two and three design variables together). Visualization of these results was achieved using parallel coordinates. This kind of technique is very effective at highlighting the effects of using different combinations of the design variables on the objective functions. Figures 6-7 to 6-10 show that a trade-off solution is required to provide an overall improvement in the values of the objective functions, and that only a compromise in the values of the design variables can lead to an optimum result. It is clear that the objective functions are more sensitive to wingspan and less sensitive to taper ratio (Figure 6-7 and Figure 6-8). Also, it can be seen that the objective functions are highly sensitive to span-root chord sweep but less so to root chord-taper ratio sweep (Figure 6-7 and Figure 6-9). Actually, sweeping the design variables either individually or in combinations provided a better understanding of the design space, design variables, and objective function relations.

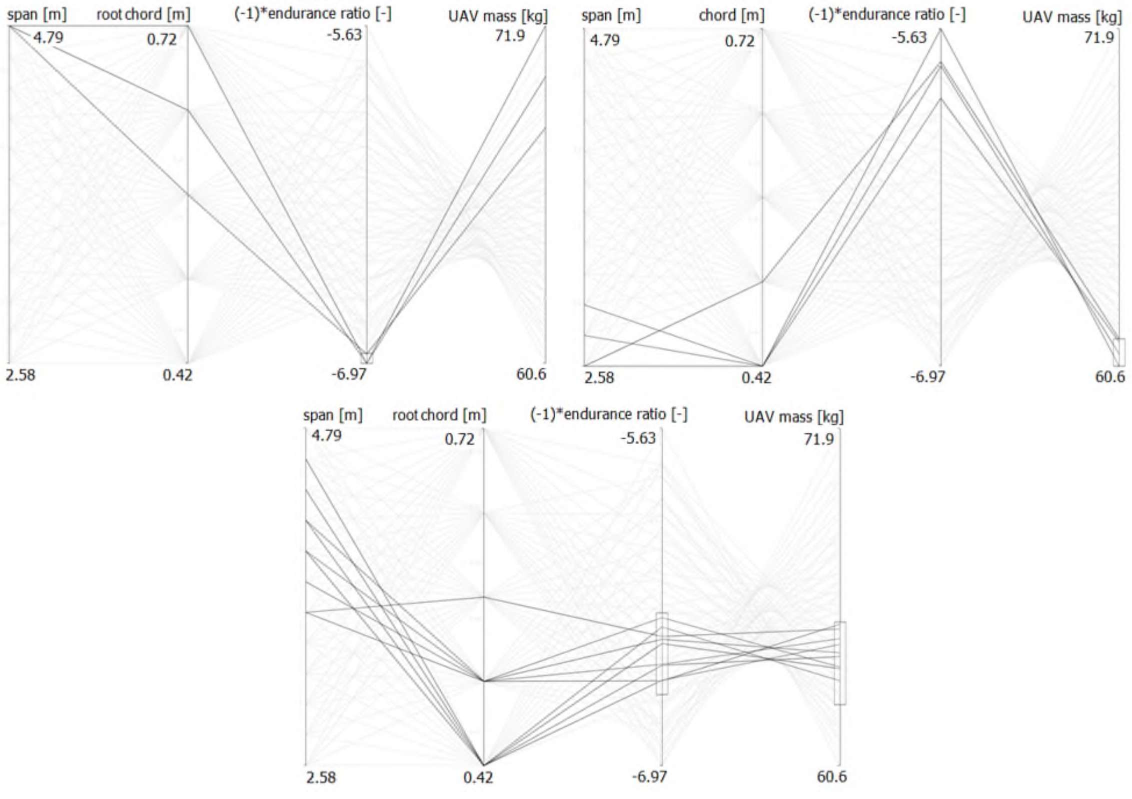


Figure 6-7: Span and root chord sweep results

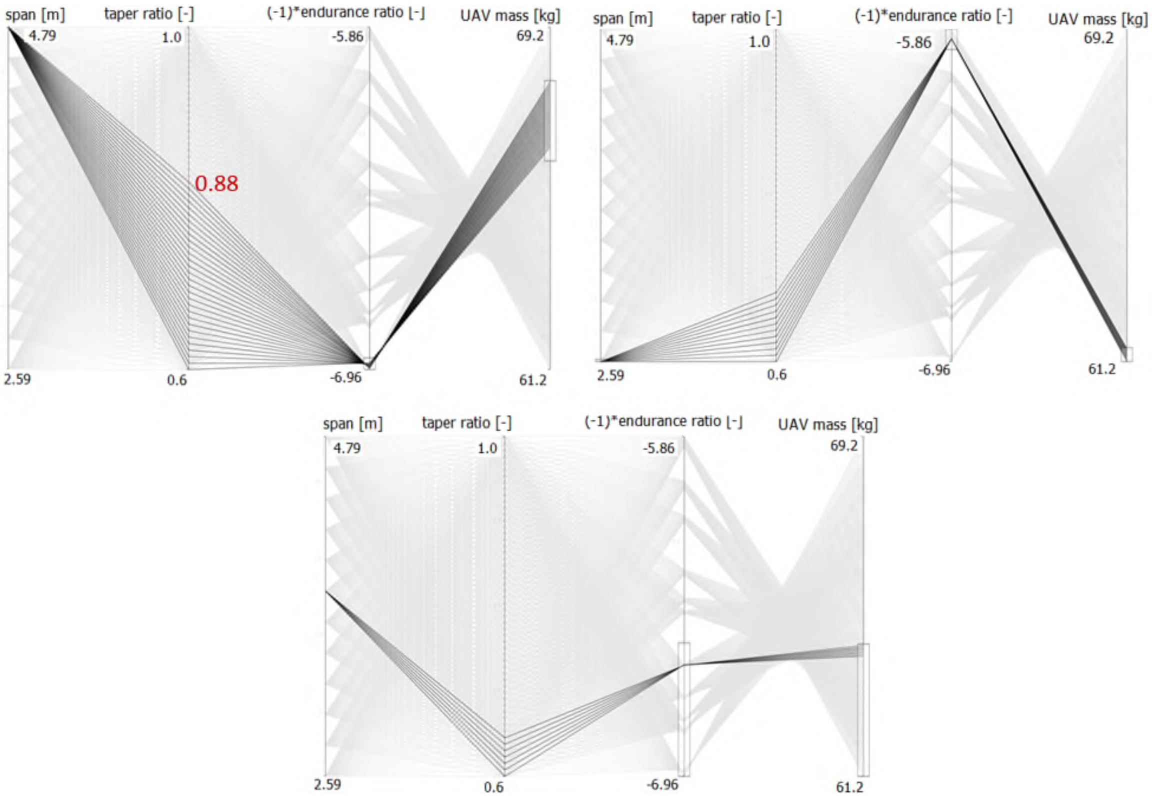


Figure 6-8: Span and taper ratio sweep results

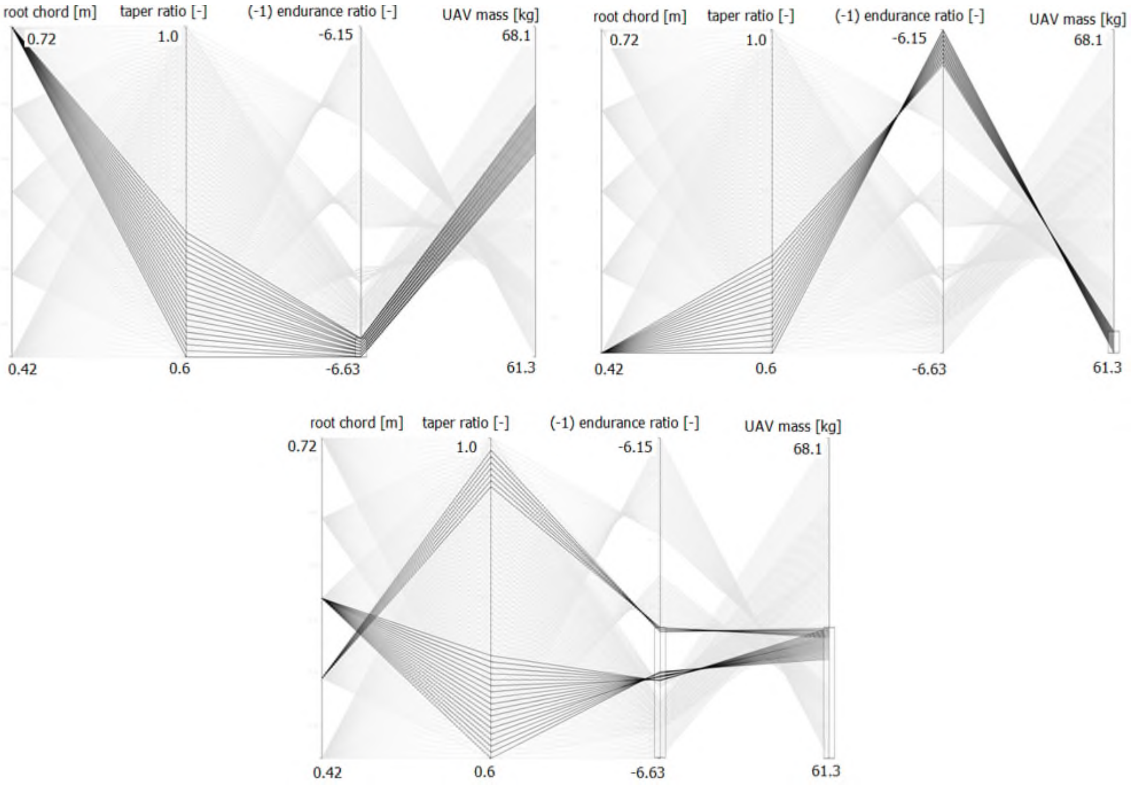


Figure 6-9: Root chord and taper ratio sweep results

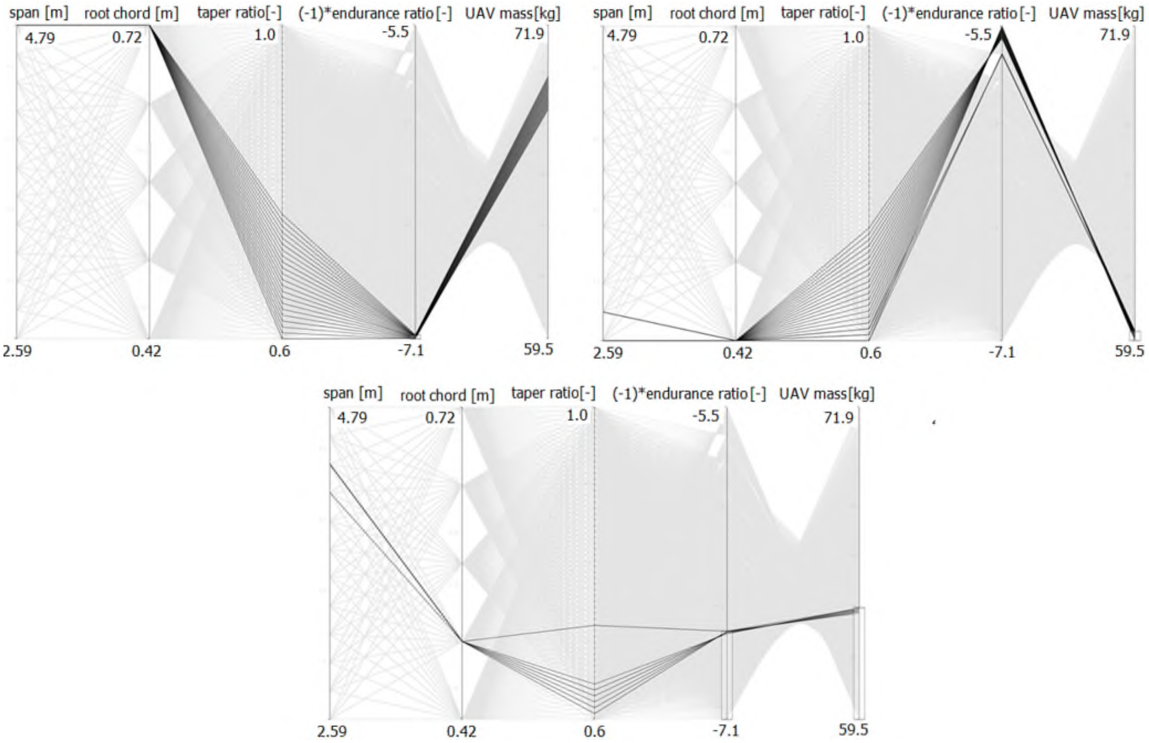
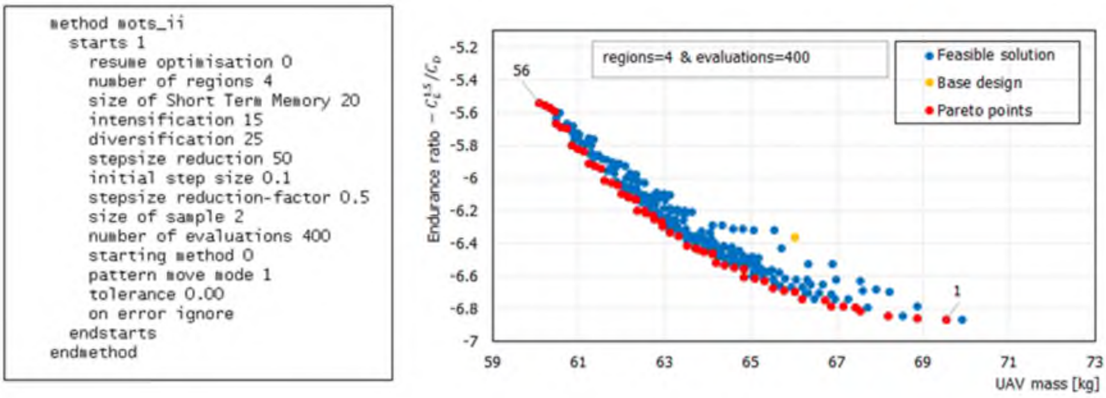


Figure 6-10: Span, root chord, and taper ratio sweep results

6.2 Optimisation process and results

Exploring the design parameter space without applying any constraints is an important stage in checking the implementation of the interface using the AVL code, MOTS algorithm, and Nimrod/O tool optimiser. The left panel in Figure 6-11 shows a section of the Nimrod/O schedule file, where the user can identify the MOTS parameters. The optimisation process first starts without applying any constraints and then under constraints (subject to all constraint in Equation 5-16, except pitching moment) using wing design variables. Figure 6-11, right panel, shows the feasible solutions without applying a constraint. It shows 56 different Pareto points that have different design variable combinations and dissimilar aerodynamic characteristics. The optimiser satisfied the maximum requirements of both objectives in point-1 and point-56. Point-56 gives minimum UAV weight, whereas point-1 gives the best endurance ratio. As Pareto optima do not accept the domination of a single objective, different combinations of wing design variables are available for the remaining Pareto points, where there is no absolute domination of one objective over the other.



```
method mots_ii
  starts 1
    resume optimisation 0
    number of regions 4
    size of Short Term Memory 20
    intensification 15
    diversification 25
    stepsize reduction 50
    initial step size 0.1
    stepsize reduction-factor 0.5
    size of sample 2
    number of evaluations 400
    starting method 0
    pattern move mode 1
    tolerance 0.00
    on error ignore
  endstarts
endmethod
```

Figure 6-11: left MOTS setting and right optimisation results without constraints

Finally, the optimiser is re-executed under constraints to get more feasible solutions. Figure 6-12 shows optimisation results using five regions, and different numbers of evaluations. It is clear that the Pareto front results become smoother as the number of evaluations is increased.

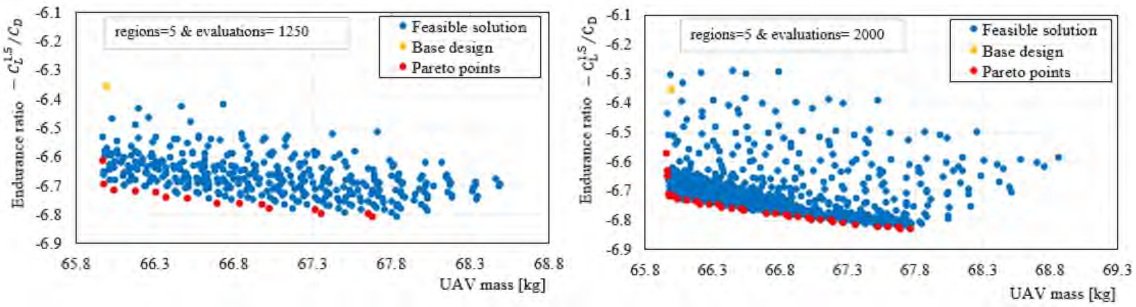


Figure 6-12: Initial optimisation results with using constraints; left using 5 regions and 1250 evaluations, and right using 5 regions and 2000 evaluations

Since the design variable bounds needed to be redefined, 14 runs were performed using different regions and evaluations. Then, by directly observing the Pareto front of each solution, using the parallel coordinates visualisation technique, the design variables' bounds were efficiently redefined, see Figure 6-13. It obvious that when the wingspan is a maximum (4.8 m) and taper ratio is a minimum (0.6), the endurance ratio has its best value (-7.0), and the mass values can vary from minimum to maximum. This variation in mass values is due to differences in wing root values. It is clear that wing root values between 0.55 m and 0.725 m are more efficient, whereas chord root values below 0.55 contribute neither to best endurance nor minimum weight. Using maximum span value and proper selection of taper ratio and chord root will result simultaneously in a reasonable endurance ratio and UAV mass.

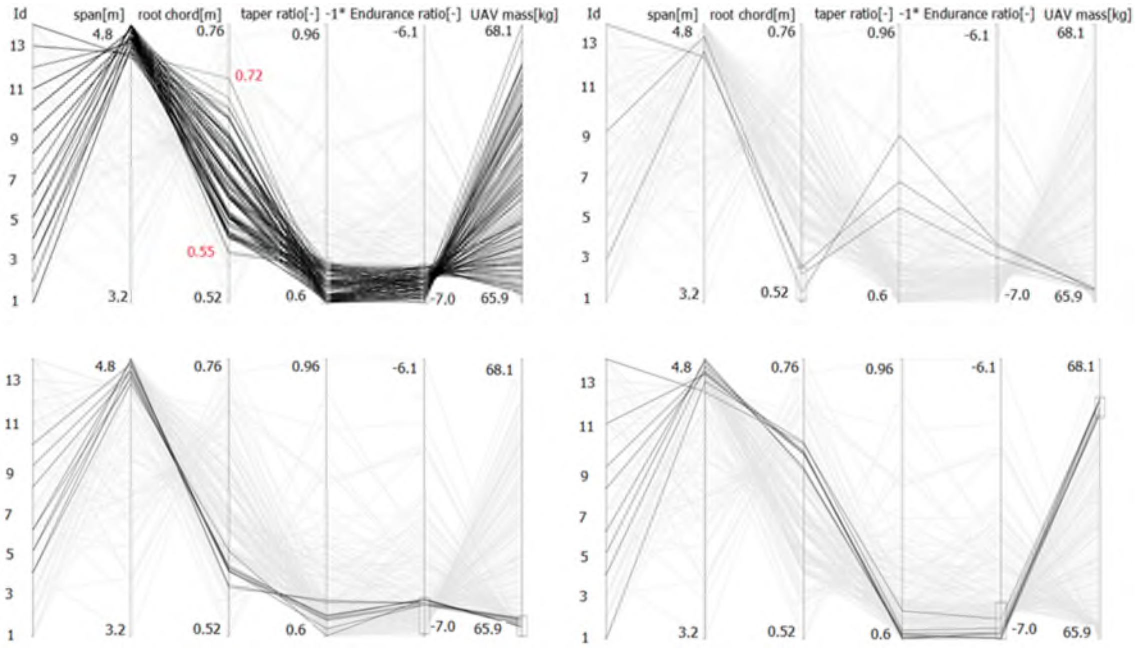


Figure 6-13: Pareto front points analysis using parallel coordinates

To investigate the effects of lower values of taper ratio, another case study was performed by changing its lower bound from 0.6 to 0.4. It was found that the minimum value of taper ratio that can be assigned to any new configuration was 0.5. The optimiser could not go below this value of taper ratio because of the constraint on stall velocity. However, a minimum taper ratio of 0.6 was selected as a lower bound at this stage of the design, to consider the wing surface area required for control surfaces in the next step of the design process. On the other hand, endurance ratio is almost directly proportional to wingspan, so that as wingspan increases, the endurance ratio is also improved. However, the maximum wingspan is restricted to 4.5 m due to constraints imposed by material stiffness, the ground effect at landing, and handling requirements.

To select a sufficient number of suitable regions and evaluations necessary to obtain an efficient and optimal design, the MOTS setting was investigated by performing several runs with different sets of regions and evaluations as a convergence study. The study was performed using the wing and wing-tail design variables simultaneously. Consequently, five regions and evaluations ranging from 1300 to 5500 were used, which depended on the number of design variables. To demonstrate the effect of the above changes, Nimrod/O was again used to obtain a set of optimal wing shapes for the Aegis UAV using wing design variables at level flight. Figure 6-14 shows the feasible solutions. It shows 64 different Pareto optima, and there is no absolute domination of one objective over the other.

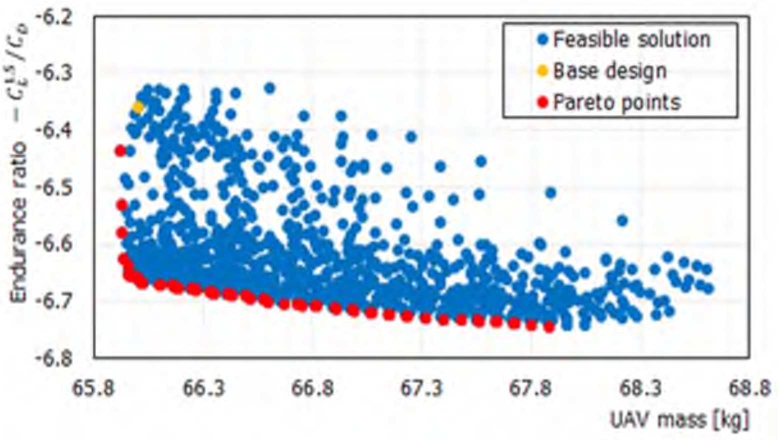


Figure 6-14: Optimisation results using 5 regions and 1300 evaluations

6.3 Exploring the Aegis UAV configurations under various design scenarios

To demonstrate the efficient use of the preceding formulation and study, we solved a series of cases of aerodynamic shape optimisation problems for the Aegis UAV under different design requirements to gain a better understanding of the design requirements to achieve optimal solutions

6.3.1 Case 1: optimisation of Aegis UAV with U-tail shape, subject to base design pitching moment, by using wing design variables

Initially, the optimisation was performed without using the pitching moment constraint. Figure 6-15 shows the aerodynamic shape design optimisation results, in which only wing design variables were varied to obtain the optimum Pareto set. Two compromise Pareto solutions were selected for this study, and their performance compared with the base design, see Figure 6-16. Pareto P1 achieved an improvement of 5.98% in the endurance ratio with only 2.12% mass penalty, whereas Pareto P2 with 0.15% mass penalty achieved 5.04%

improvement in endurance ratio. Since no pitching moment constraint was applied through the optimisation process, the absolute value of the pitching moment increased from 0.0470 to 0.0790 and from 0.0470 to 0.0590 for Pareto P1 and Pareto P2, respectively. This increase in pitching moments will lead to an increase in drag when the UAV is trimmed during flight.

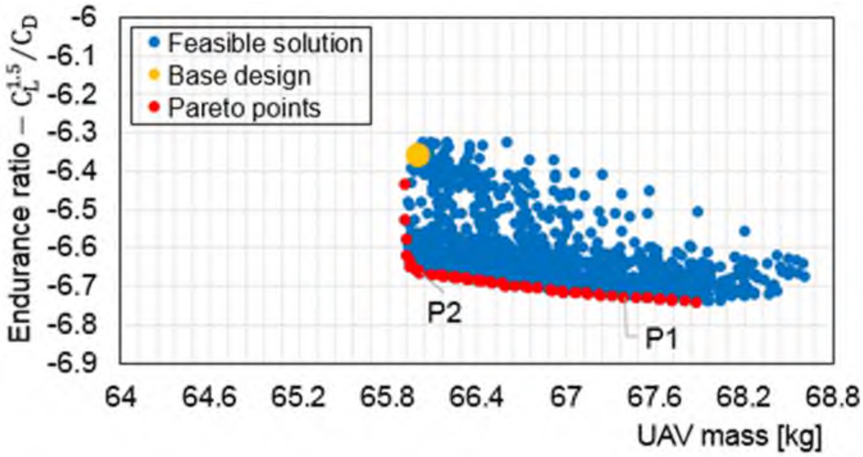


Figure 6-15: Optimisation results using 5 regions and 1300 evaluations; Two Pareto points are selected to be a comparison with base design (simulation time is around 26 minutes)

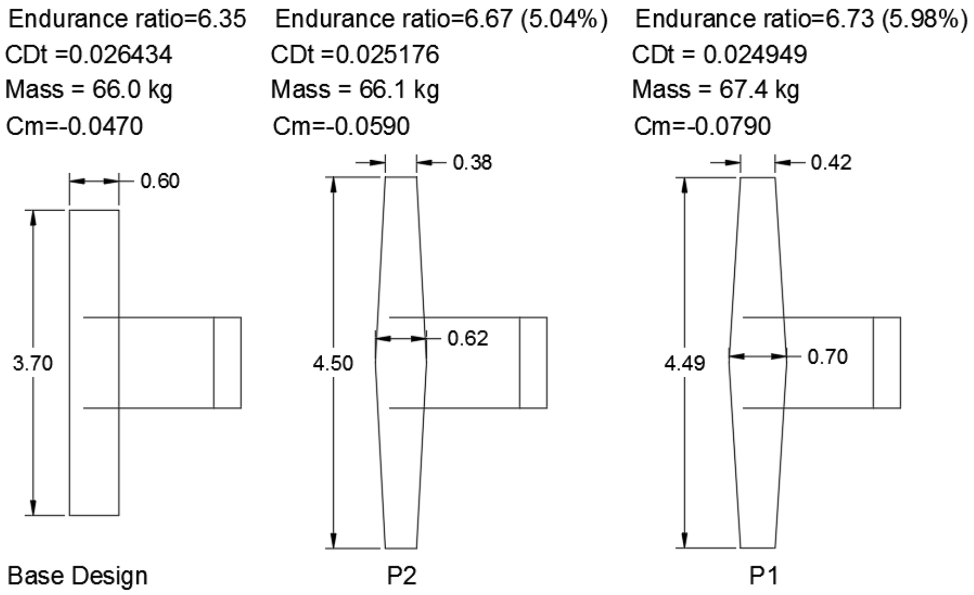


Figure 6-16: Comparison of the configurations for the selected compromise solutions with base design configuration

To account for lower drag and increment in power consumption at the trimming condition, the optimisation process was re-executed by adding the base design pitching moment constraint (Cm ≥ -0.0470) which freed the elevator server from applying higher loads when the UAV was trimmed during level flight. This moment constraint was enforced without implanting the tail geometry as a design variable into the optimisation process. Considering the stochastic characteristics of the MOTS, five independent runs using the same settings (five regions and 2400 evaluations) were performed to estimate the performance of the optimisation for this case. The average run time was 39 minutes. Figure 6-17 shows the distribution of feasible solutions for one of the runs.

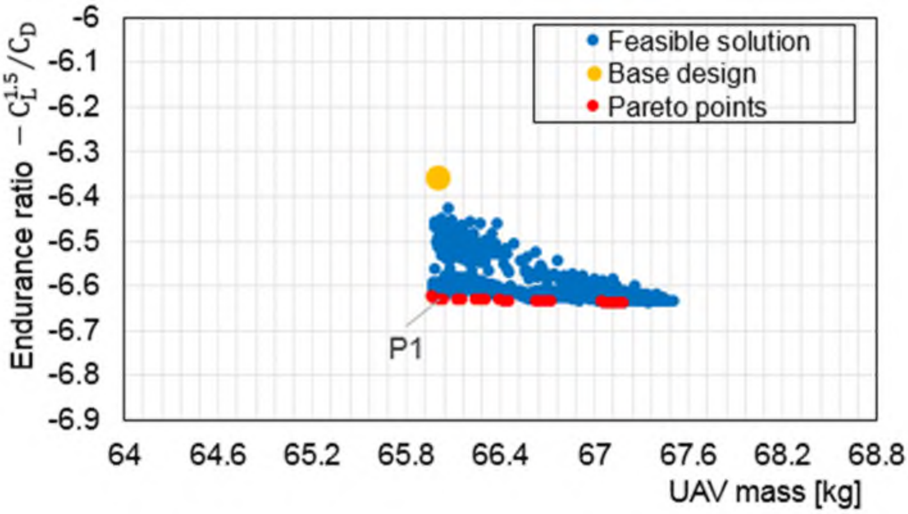


Figure 6-17: Optimisation results by using 5 regions and 2400 evaluations under pitching moment constraint

Finding a set of typical Pareto optimal solutions using the MOTS algorithm was only half the task; choosing a single preferred solution from the obtained set was equally important. A visual inspection of the Pareto points can provide useful information to DMs about the possible trade-offs between objectives. Pareto P1 was selected as an optimal compromise solution for further study, see Figure 6-17. Figure 6-18 shows a comparison of Pareto P1 with the base design. The endurance ratio increased from 6.35 to 6.63 with zero mass penalty. It corresponded to 4.15% reduction in drag. Since the pitching moment constraint was applied to the optimised geometry, the negative pitching moment decreased from -0.0470 to -0.0467.

On the other hand, the endurance ratio of the optimised configurations subjected to pitching moment constraint was reduced compared to the one optimised without applying pitching moment constraint, which is what would be expected. However, the optimised configuration with pitching moment constraint had a lower negative moment value. This led to a lower increment in drag when trimmed later, in flight.

Figure 6-19 shows a comparison of the aerodynamic performance for the compromise solution P1 and the base design. It was evident that Pareto P1 had a better performance than the base design at most angles of attack. These results indicate that both configurations are longitudinally stable, but Pareto P1 has a lower pitching moment at the design point, C_L=0.3044

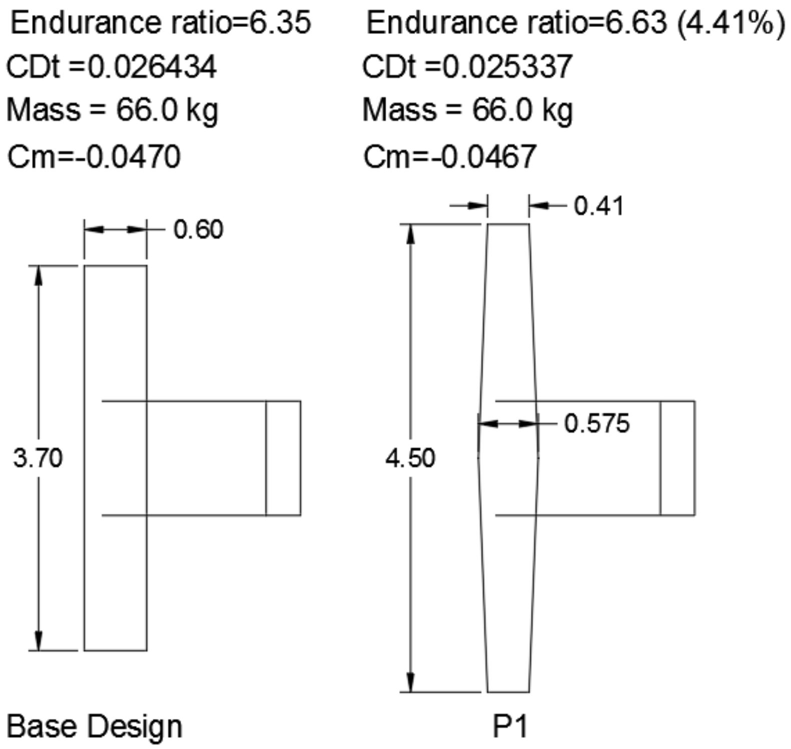


Figure 6-18: Comparison of the optimal compromise configuration Pareto P1 with the base design

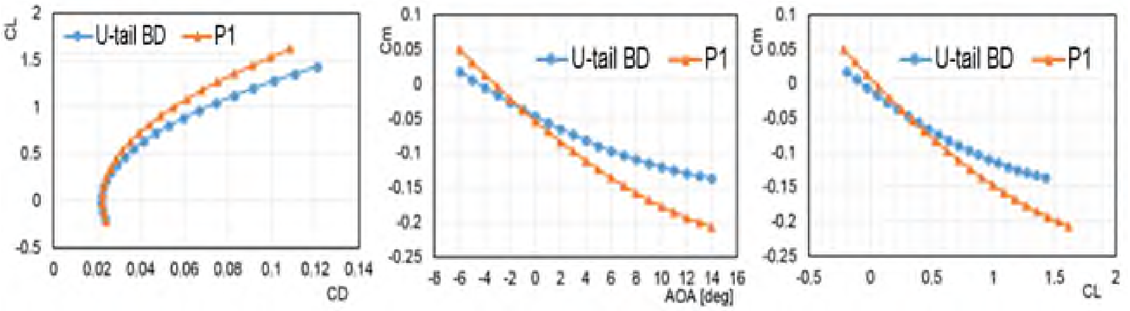


Figure 6-19: Comparison of the aerodynamic performance for Pareto P1 and base design

6.3.2 Case 2: Optimisation of Aegis UAV with U-tail shape, subject to trim constraint, by using wing and horizontal-tail rotation angle design variables

In this case, the optimisation process was one in which the trim (Cm=0) was achieved by adding the horizontal tail rotation angle as an additional design variable (∓ 6 deg), see Equation 5-2. Initially, the optimisation process commenced using only the horizontal tail rotation angle to trim the base design. The optimised tail rotation angle for the trimmed base design is -2.0 deg, and the drag increased by 2.39% relative to base design. Next, the optimisation process was performed by varying wing design variables and tail rotation angle simultaneously, using Cm=0 (Equation 5-17). Figure 6-20 shows the feasible solutions of selected simulations. Pareto P1 was selected as the optimal compromise solution for further investigation. To investigate the drag increment, we compared Pareto P1 with the base design and trimmed base design, trimmed by rotating only the horizontal tail angle. The compromise solution Pareto P1 configuration exhibited a drag reduction of 2.56% compared with the base design and 4.84% less than the trimmed base design, see Figure 6-21. It is evident that it was inappropriate to trim the base design by optimising horizontal tail rotation angle only, and it was better to use the wing and horizontal tail rotation angle jointly as design variables to optimise the base design. Figure 6-22 shows the comparison of the aerodynamic performance of Pareto P1 with base design and trimmed base design.

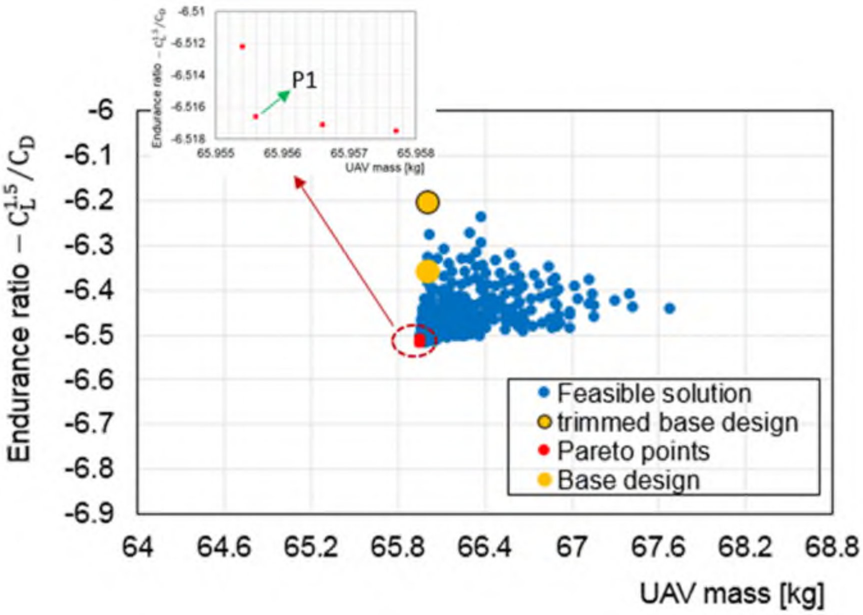


Figure 6-20: Trimming optimisation results by using 5 regions and 3000 evaluations, Pareto solutions are in the zoomed graph for magnification and clarity

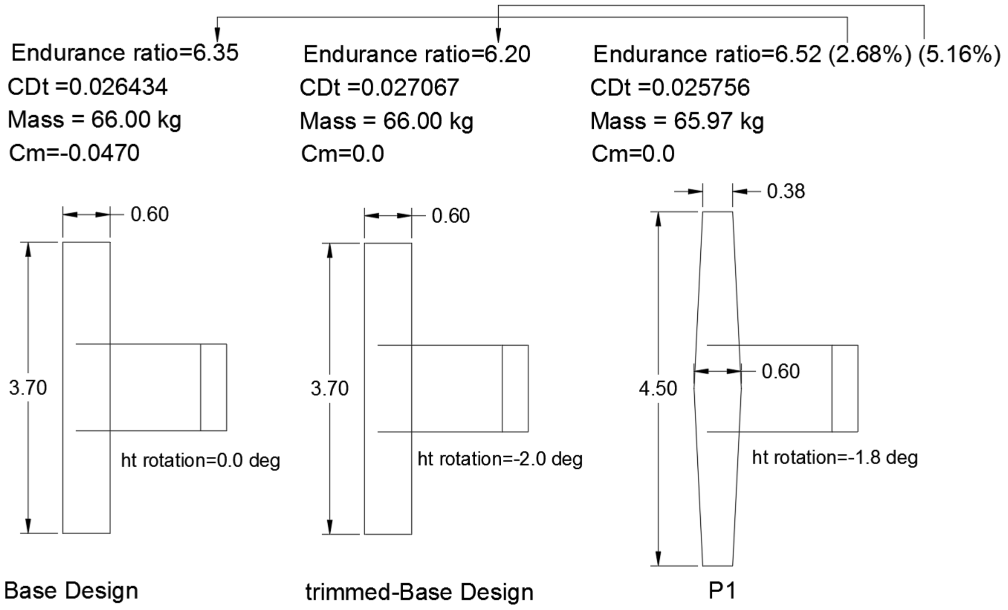


Figure 6-21: Comparison of base design, trimmed-base design (using tail rotation angle only), and optimum compromise configuration (using wing and tail rotation angle)

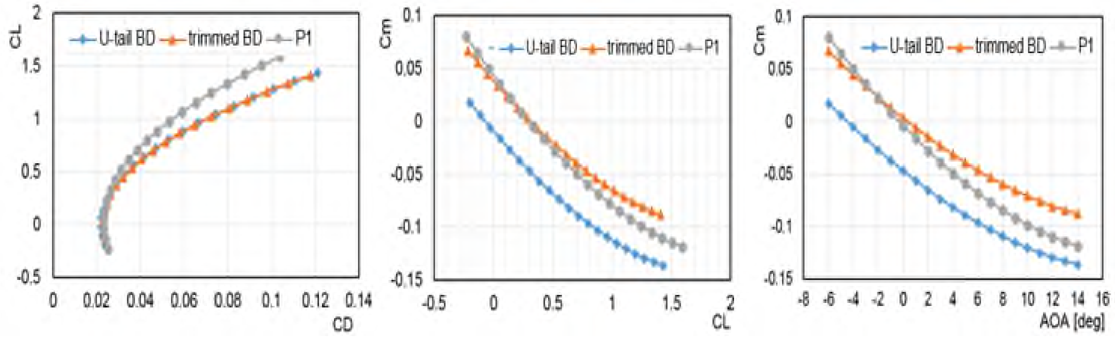


Figure 6-22: Comparison of the aerodynamic performance for the base design, trimmed base design, and Pareto P1 configurations

6.3.3 Case 3: Optimisation of Aegis UAV with U-tail shape, subject to base design pitching moment, by using wing and tail design variables

The following study was to optimise the wing and tail components simultaneously and observe the gain that can be obtained in UAV characteristics compared to optimising the wing only. Because our algorithm, MOTS, is stochastic, five runs were performed using the same setting (five regions and 5500 evaluations). The average run time was 135 minutes. Figure 6-23 shows the optimisation result for one of the runs. Pareto points P1 and P2 were selected as compromise solutions for further investigation. Pareto P1 may be considered as the best compromise configuration, whereas Pareto P2 shares with three other Pareto solutions almost the same highest endurance ratio, but Pareto P2 has the advantage of the lowest mass.

Figure 6-24 shows the selected configurations with the base design. Since the pitching moment constraint was imposed on the design variables, the Pareto P1 and P2 pitching moments do not exceed the base design pitching moment. The endurance ratio of Pareto P2 increased to 6.73 while that of Pareto P1 rose to 6.72. Which means the endurance ratios of Pareto P2 and P1 were 5.98% and 5.83% higher than base design, respectively. The improvements in endurance ratio and mass were obtained without violation of any static or dynamic stability requirements, see Table 6-1. Figure 6-25 compares the aerodynamic and stability performance of the selected optimised configurations with the base design. It is evident that all the stability derivatives are within the design constraints, see Equation 5-25.

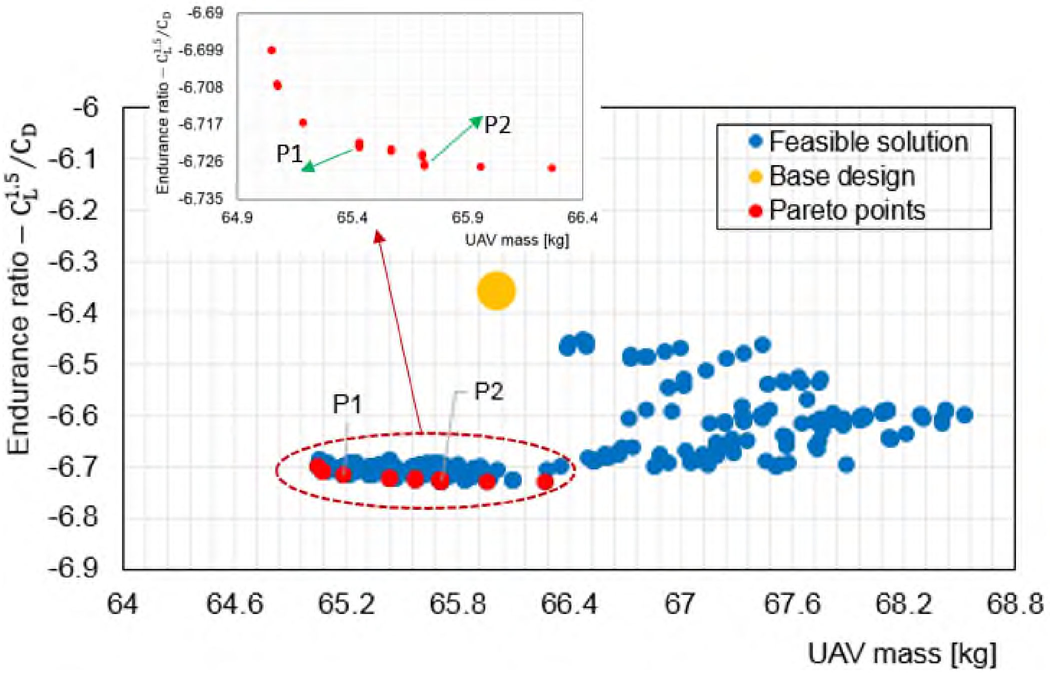


Figure 6-23: Feasible solutions obtained by using 5 regions and 5500 evaluations, Pareto solutions are in the zoomed graph for magnification and clarity

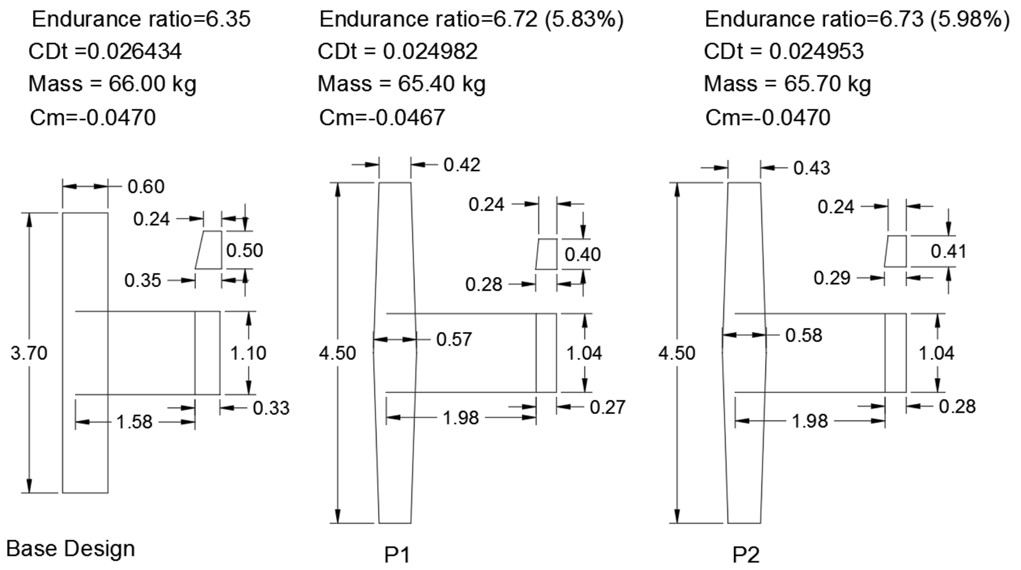


Figure 6-24: Comparison of selected optimised configurations for the UAV with U-tail shape using wing-tail design variables with the base design

Table 6-1: Comparison for the stability derivatives of selected optimised configurations with base design

Pareto solution	AOA [deg]	V [m/s]	Cm_{α} [1/deg]	Cn_{b} [1/deg]	Cl_{b} [1/deg]	Cm_{q} [1/deg]	Cn_{r} [1/deg]
P2	-0.48	43.0	-0.895	0.079	-0.0216	-15.13	-0.088
P1	-0.5	43.3	-0.921	0.077	-0.0214	-15.55	-0.085
BD	0.0	43.60	-0.549	0.118	-0.046	-8.91	-0.131

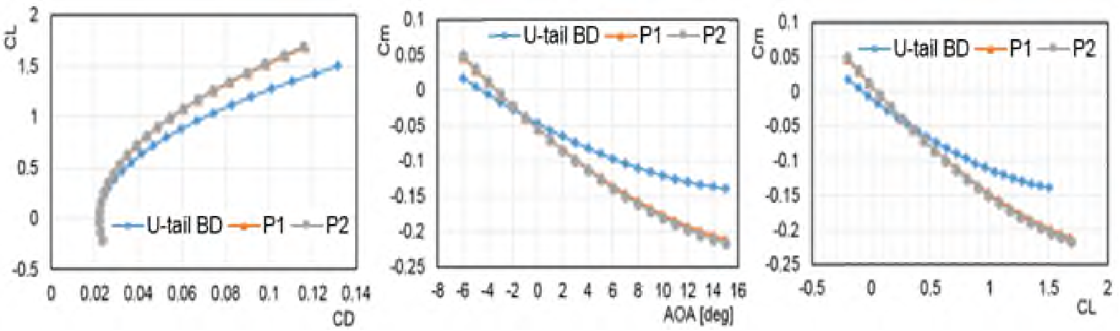


Figure 6-25: Comparison of aerodynamic performance of the selected configurations with the base design

6.3.4 Case 4: Optimisation of Aegis UAV with U-tail shape, subject to the trim constraint (Cm=0), by using the wing, tail, and horizontal-tail rotation angle design variables

In this case, the optimisation process was performed with the trim (Cm=0) achieved by adding a horizontal tail rotation angle as an additional design variable in order to improve flight performance (see Equation 5-26). The left panel of Figure 6-26 shows the distribution of the feasible solution, while the right panel shows only the Pareto front. The optimiser formed nine Pareto points within a computational time equal to 145 minutes by using five regions and 5500 evaluations. Figure 6-27 compares the optimised trimmed configurations with base design. Since the reduction in mass between the highest and lowest Pareto points did not exceed 0.90%, Pareto point P1 was selected for further study. Pareto P1 may be considered as a compromise point since it has the second best endurance ratio with 0.17% mass penalty, when compared to Pareto P2. The Pareto P1 configuration achieved a 4.38% drag reduction compared to base design and 6.62% reduction in drag when compared to trimmed base design (see Figure 6-21). In addition, no drag will be added at level flight for Pareto P1 since it is trimmed, whereas extra drag will be added for the base design at level flight because of trimming, as shown in Figure 6-28. Table 6-2 shows that all the configurations are statically and dynamically stable since the derivatives were satisfied by the constraints.

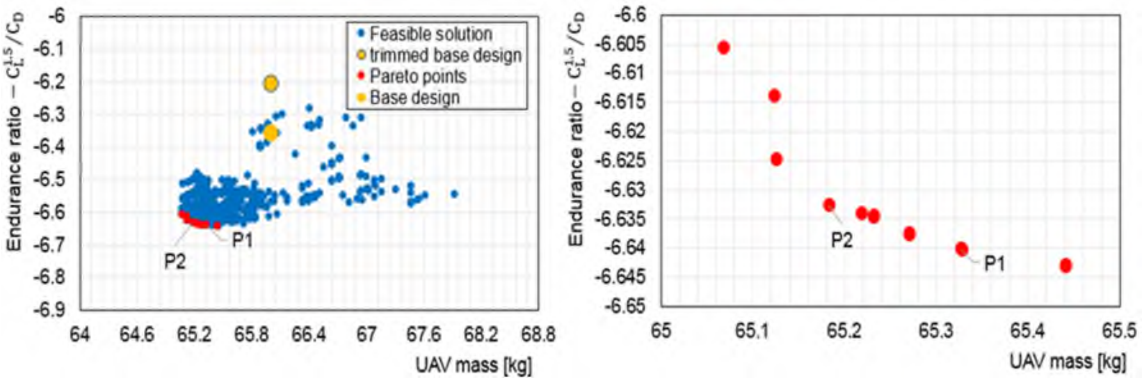


Figure 6-26: Left panel; optimisation results under the trim condition: right panel the Pareto front with two different compromise solutions

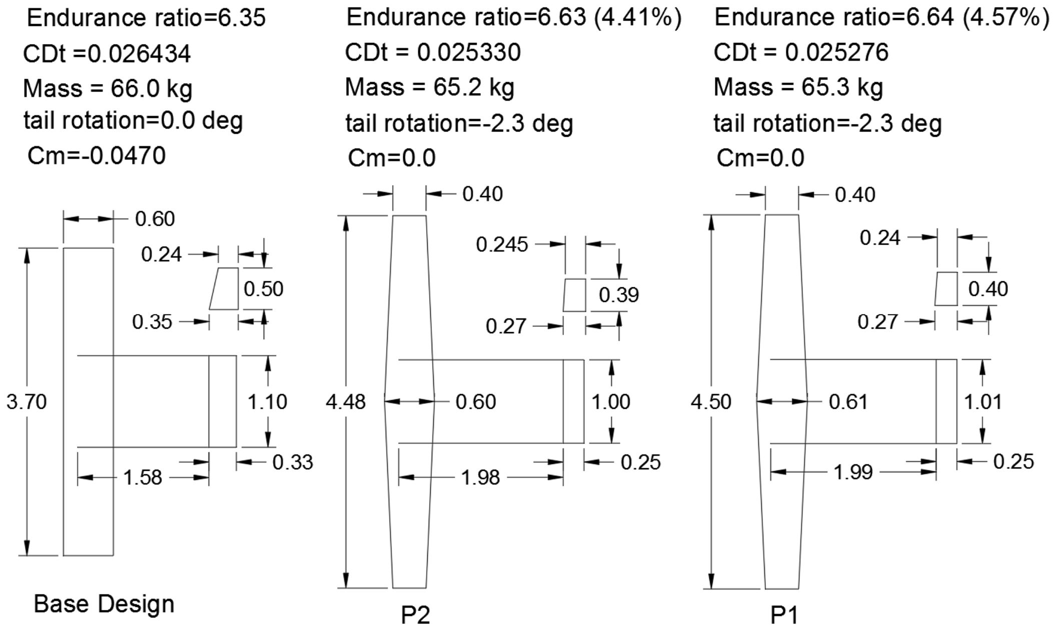


Figure 6-27: Comparison of the optimised configurations for the UAV with U-tail shape using the wing, tail, and horizontal-tail rotation angle design variables with the base design

Table 6-2: Comparison for the stability derivatives of selected optimised configurations using wing, tail, and horizontal-tail rotation angle with base design

Pareto solution	AOA [deg]	V [m/s]	Cm_{α} [1/deg]	Cn_{b} [1/deg]	Cl_{b} [1/deg]	Cm_{q} [1/deg]	Cn_{r} [1/deg]
P1	-0.29	43.0	-0.963	0.072	-0.021	-13.96	-0.082
P2	-0.30	43.2	-0.963	0.072	-0.022	-14.08	-0.082
BD	0.0	43.60	-0.549	0.118	-0.046	-8.91	-0.131

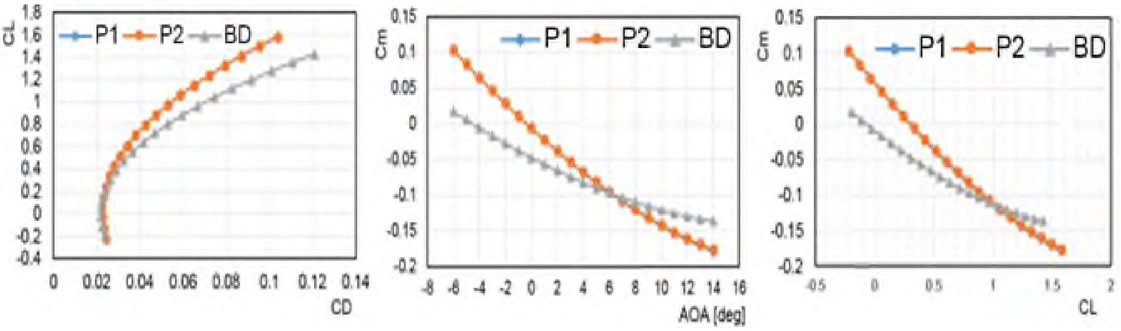


Figure 6-28: Comparison of the aerodynamic performance of the base design and the optimised configuration under trim condition using the wing, tail, and horizontal tail-rotation angle design variables

6.3.5 Case 5: Optimisation of Aegis UAV with inverted V-tail shape, subject to base design pitching moment, by using wing design variables

Figure 6-29 shows the results obtained from the optimisation of the endurance ratio and mass without pitching moment constraint. The results were obtained using five regions and 1300 evaluations. Pareto point P1 is a highly compromised solution since it is close to the centre of the conflicting objectives. The endurance ratio at Pareto P1 was increased by 5.12% with respect to the base design. Since no pitching moment constraint was applied to the design variables, the negative pitching moment increased from -0.0720 to -0.0860, see Figure 6-30. The optimiser used the angle of attack to maintain the equilibrium of the Aegis UAV in level flight. The angle of attack for Pareto P1 decreased from zero to -0.44 deg. It is clear that the optimiser using the angle of attack as a tool to obtain low drag and maintain the required lift for the UAV.

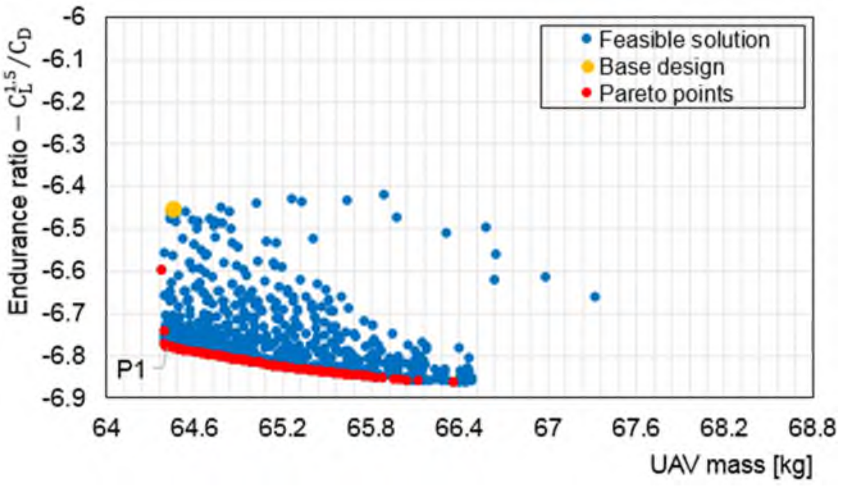


Figure 6-29: The optimisation results without pitching moment constraint by using 5 regions and 1300 evaluations (simulation time was around 23 minutes)

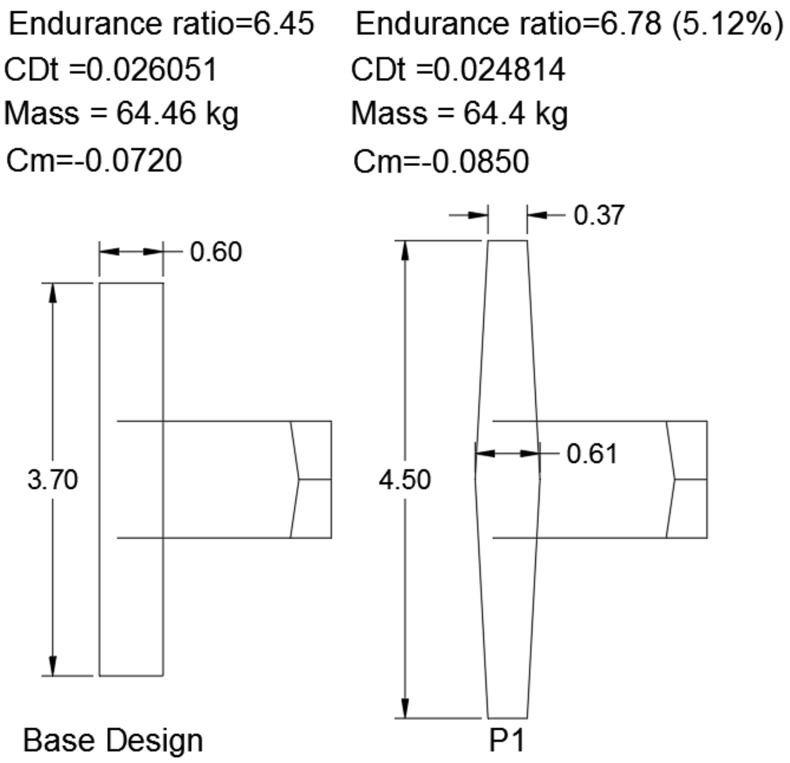


Figure 6-30: Comparison of the configurations for the selected compromise solution with base design configuration

The optimiser was again executed with wing design variables under the constraint (Cm≥-0.0720). Figure 6-31 shows the solution obtained using five regions and 2400 evaluations. Pareto P1 was considered as a good compromise solution for the two objectives, whereas Pareto P2 has endurance ratio 0.15% higher than

Pareto P1 with a mass penalty of 0.40 kg. The endurance ratio of Pareto P1 increased by 3.89%, while Pareto P2 increased 4.03% with respect to base design. However, the increase in Pareto P1 endurance ratio was obtained with zero mass penalty, whereas Pareto P2 endurance ratio was obtained with 0.68% mass penalty. Since the pitching moment was constrained to the value of base design, the Pareto points pitching moment do not exceed the absolute value of the base design, see Figure 6-32.

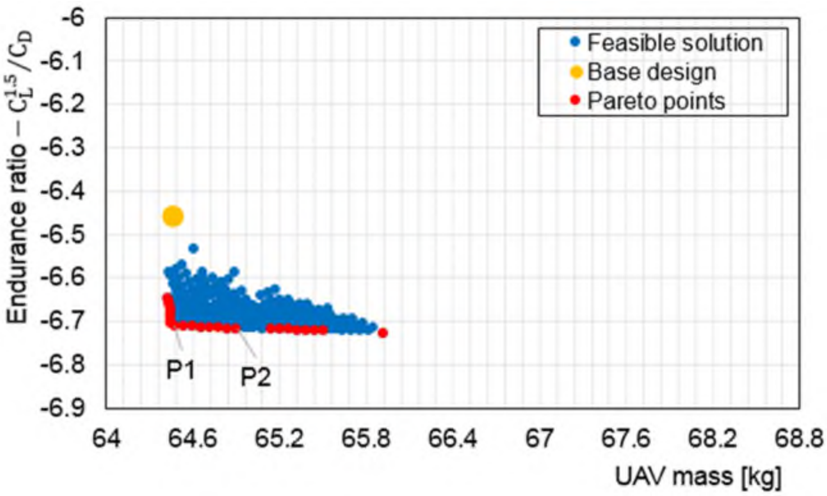


Figure 6-31: The optimisation results with pitching moment constraint by using 5 regions and 2400 evaluations (simulation time is around 38 minutes)

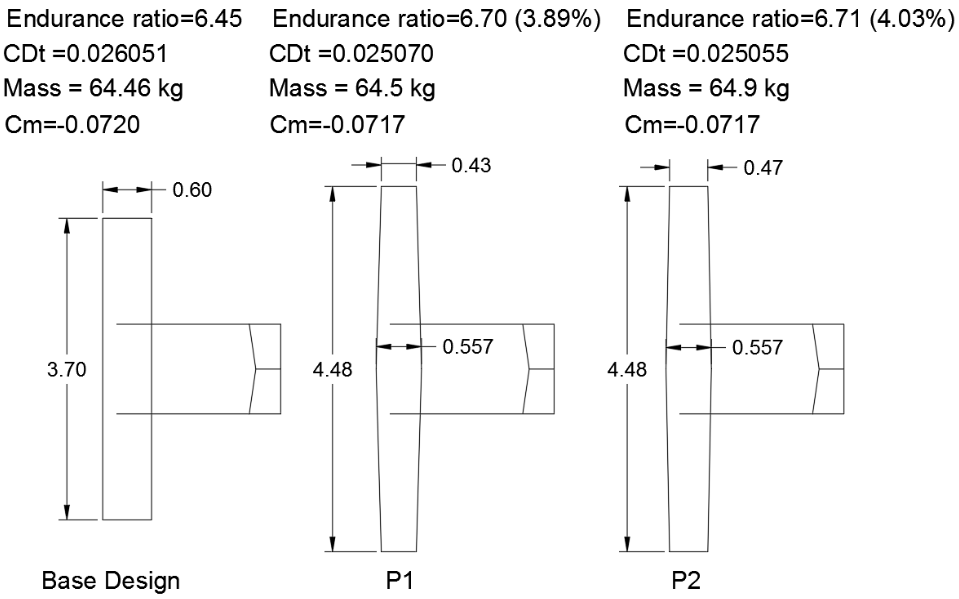


Figure 6-32: Compromise of two selected solutions optimised using wing design variables with the base design

Figure 6-33 compares the aerodynamic performance of the selected configurations with a base design. It is clear that Pareto P1 shows slightly better performance than Pareto P2 for the positive angle of attack. This shows that the Pareto P2 configuration has better performance at the design point CL=0.3047, whereas the Pareto P1 configuration has better performance off-design. In fact, all the Pareto solutions can be considered as optimised solutions for the base design, and any one could be chosen for further study. However, only one compromise solution will be considered for the final design. This is the choice of the DM and will depend on other considerations concerning the design and the mission requirements. In this work, Pareto P1 was selected for further study and investigation.

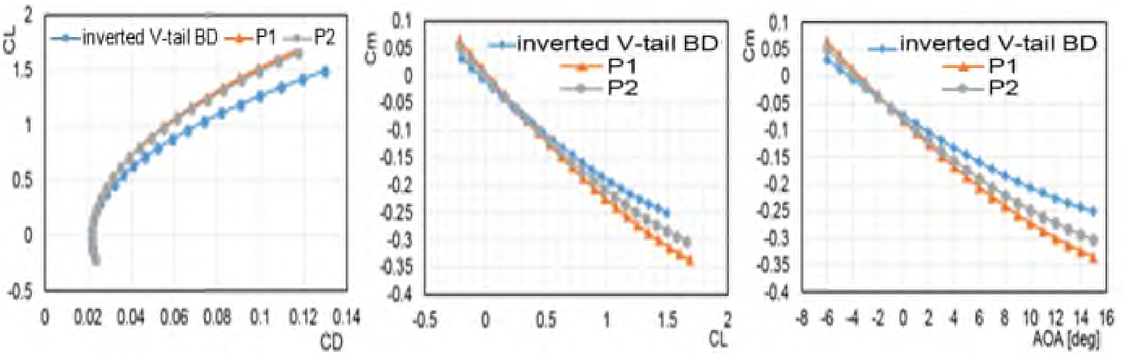


Figure 6-33: Comparison of the aerodynamic performance of the selected configurations with base design

6.3.6 Case 6: Optimisation of Aegis UAV with inverted V-tail, subject to base design pitching moment, by using wing and tail design variables

An optimisation was performed for the inverted V-tail UAV combined with wing-tail design variables. The results were compared with those obtained from optimising the wing design variables only. The design variables were subject to the constraint of the pitching moment ≥-0.720, as explained in the problem formulation. Figure 6-34 shows the feasible solutions of the optimisation process performed using five regions and 5500 evaluations. The optimiser applied some 4990 penalty functions on solutions that did not satisfy the constraints. It is evident that aerodynamic design optimisation is highly constrained as the number of infeasible solutions is much more than feasible solutions. The optimiser spent around 126 minutes to satisfy the constraints and obtain a feasible solution. It is clear that Pareto points 1, 2, and 3 have almost the same endurance ratio. However, Pareto P3 has a greater mass reduction than Pareto P1 or P2, ranging from 300 g to 600 g. Therefore, Pareto P3 was chosen for further investigation as an optimal compromise solution. Figure 6-35 shows a comparison of the selected configuration with the base design. The results show an improvement in the endurance ratio for the selected point equal to 5.43% and the reduction in mass of 0.55% while maintaining a lower value of the pitching moment. It is also observed that the optimiser used the angle of attack as a trade-off to keep the UAV in level flight (CL=0.3047) with minimum drag, see Table 6-3. Figure 6-36 shows a comparison of the aerodynamic performances of the Pareto P3 and base design. The optimised configuration, Pareto P3, shows better performance than the base design configuration for all angles of attack. Both configurations are longitudinally stable with pitching moment value around 0.720.

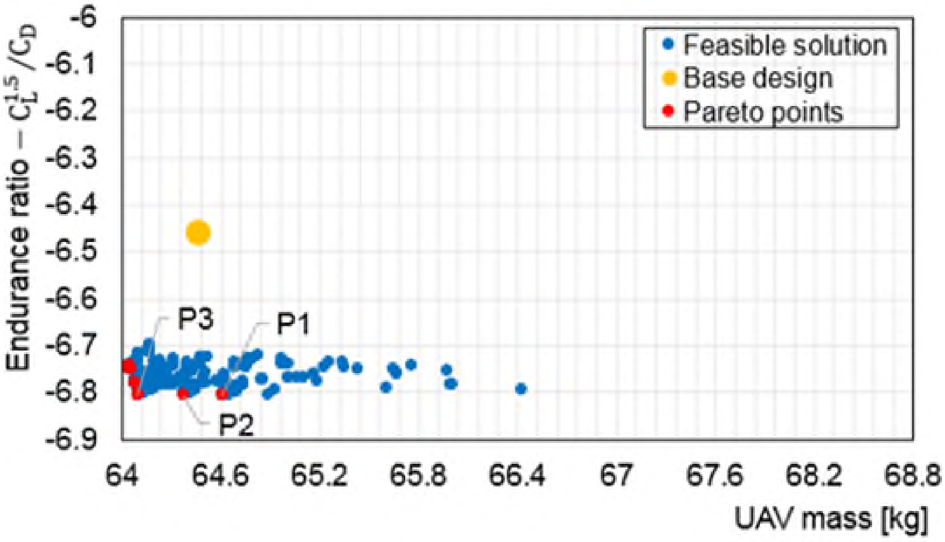


Figure 6-34: Feasible solutions obtained by optimising the UAV with inverted V-tail using wing and tail design variables simultaneously

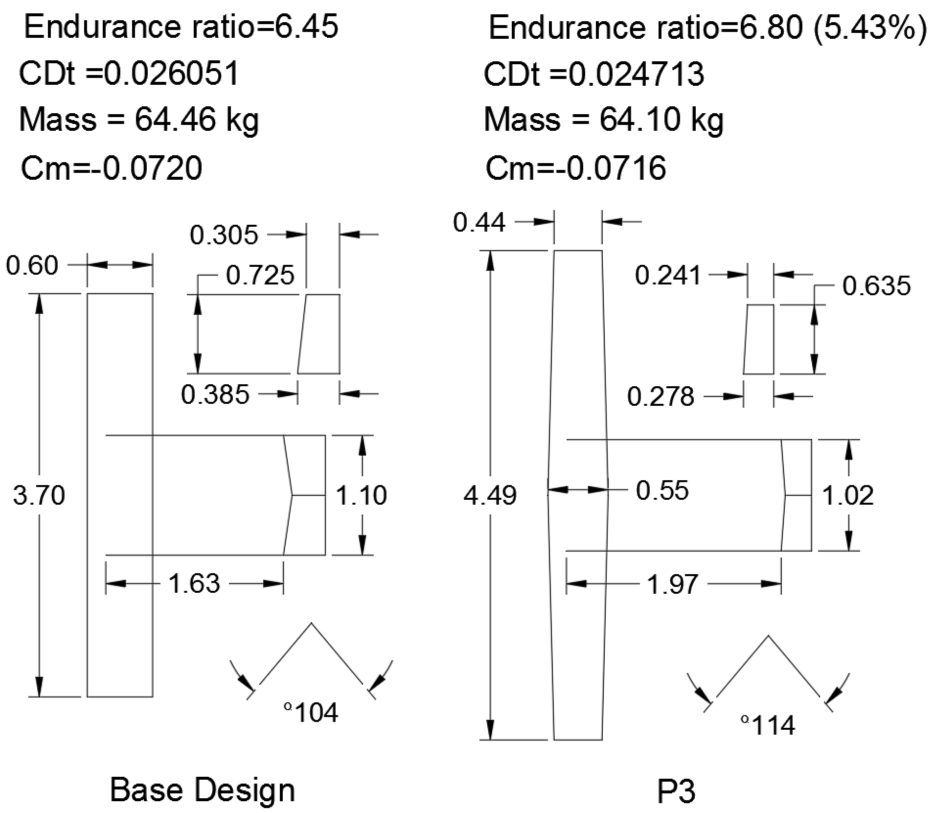


Figure 6-35: Comparison of the optimised configuration for the UAV with inverted V-tail shape using wing-tail design variables with base design

Table 6-3: Compares for the stability derivatives of selected optimised configurations using wing-tail design variables with base design

Pareto solution	AOA [deg]	V [m/s]	Cm_{α} [1/deg]	Cn_{b} [1/deg]	Cl_{b} [1/deg]	Cm_{q} [1/deg]	Cn_{r} [1/deg]
P3	-0.5	42.83	-1.20	0.038	-0.01	-14.94	-0.048
BD	0.0	43.60	-0.915	0.072	-0.017	-9.54	-0.087

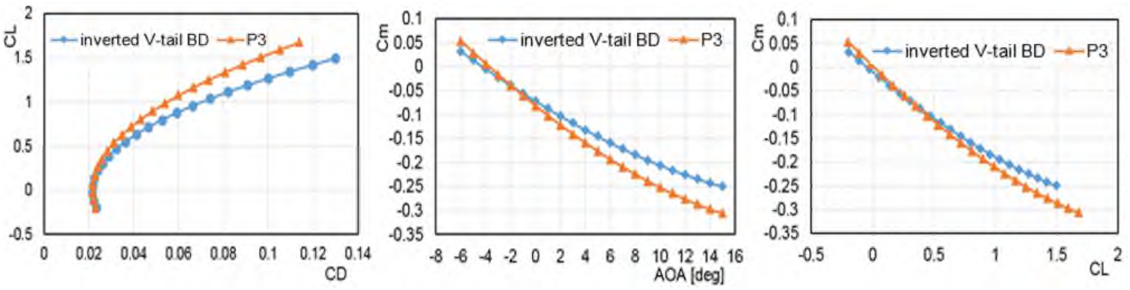


Figure 6-36: Comparison of the aerodynamic performance of the base design and the optimised configuration using wing and tail design variables

6.4 Discussion of results

6.4.1 Comparison of the optimised configurations for the UAV with U-tail and inverted V-tail shapes using wing design variables

The left panel in Figure 6-37 compares the optimised configurations for the Aegis UAV with U-tail and inverted V-tail shapes by varying wing design variables subject to pitching moment constraint. The endurance ratio of the optimised configurations for the UAV with inverted V-tail shape is 1.06% higher than that of the optimised configurations for the UAV with U-tail shape. The endurance ratio for the UAV with U-tail shape increased from 6.35 to 6.63 (4.41%) with zero mass penalty, while the endurance ratio for the UAV with inverted V-tail increased from 6.45 to 6.70 (3.88%) with 0.06% mass penalty, which is negligible. Even though the optimised configuration for the UAV with inverted V-tail shape still has a higher endurance ratio and lower mass than the optimised UAV with U-tail shape, the improvements in endurance ratio and mass with respect to the base design are better in the case of the UAV with U-tail shape

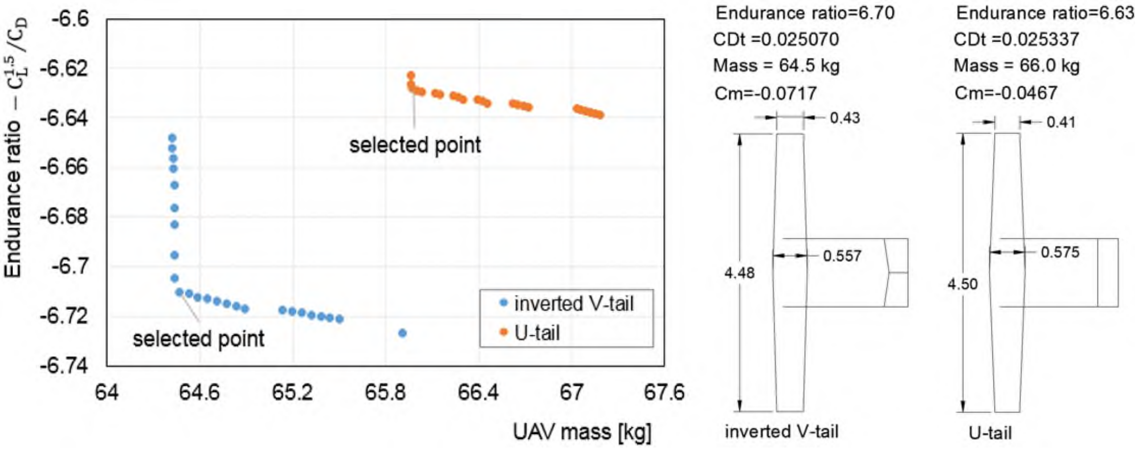


Figure 6-37: Left panel; comparison of the obtained Pareto front for UAV with U-tail and UAV with inverted V-tail shapes optimised using wing design variables: the right panel; comparison of configurations for two selected solutions

6.4.2 Comparison of the optimised configurations for the UAV with U-tail shape using wing, wing-tail, and wing-tail-horizontal tail rotation angle

In Figure 6-38 it can be seen that the optimised configuration for the UAV using wing-tail simultaneously has an endurance ratio 1.36% higher than that for the optimised configurations using only wing design variables. The optimised configuration for the UAV using wing-tail design variables has a 0.91% reduction in mass whereas the optimised configuration using wing design variables only has zero reduction in mass when compared to the UAV base design.

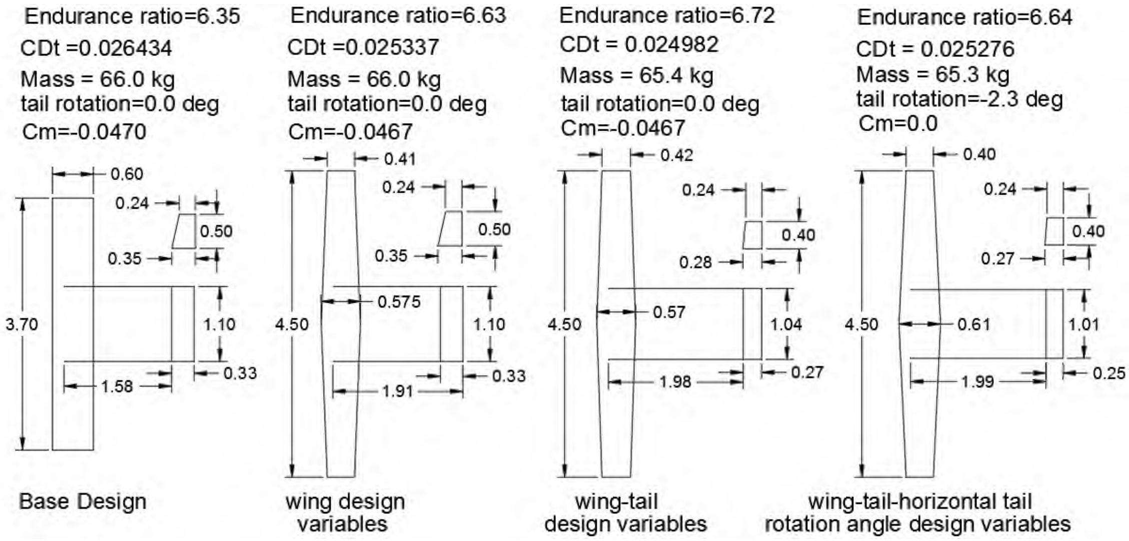


Figure 6-38: Comparison of the optimised configurations for the UAV with U-tail shape using wing, wing-tail, and wing-tail-horizontal tail rotation angle with base design

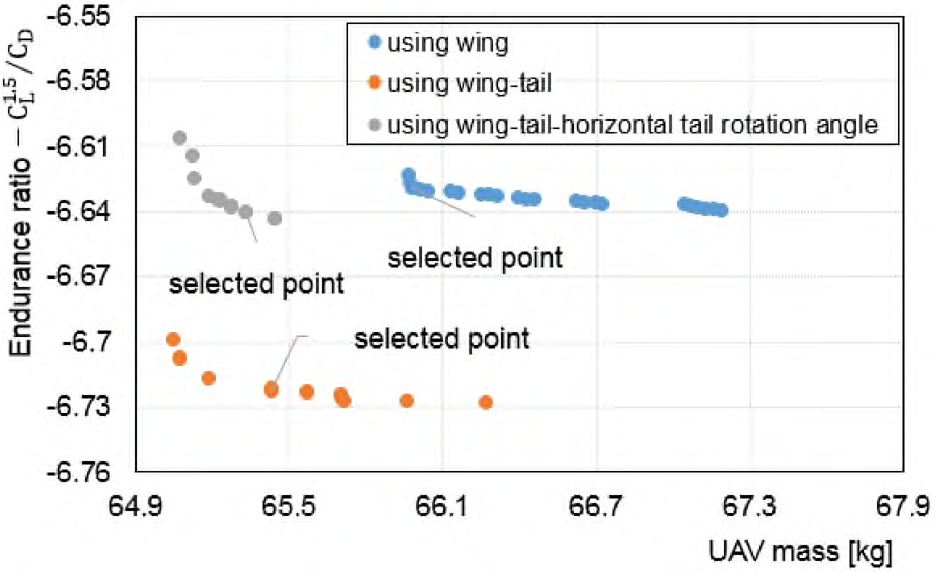


Figure 6-39: Pareto front for the optimised configurations for the UAV with U-tail shape using wing, wing-tail, and wing-tail-horizontal tail rotation angle

Figure 6-39 compares the Pareto fronts for these configurations. It is obvious that by using wing-tail design variables simultaneously, the optimiser has a greater capacity to reduce drag and satisfy both geometrical and aerodynamic constraints simultaneously. For example, the optimum compromise solution obtained by including the wing and tail design variables has a drag value 1.40% and 5.49% lower than the optimum compromise solution obtained by including wing design variables only and base design, respectively. When the trim constraint is included as a design variable, as horizontal tail rotation, the optimised configuration including wing design variables exhibited a drag of 2.56% lower than the base design and 4.84% lower than the trimmed base design (base design optimised using horizontal tail rotation angle only, Cm=0). With real flights, it is inappropriate to trim the base design by rotation of the horizontal tail only. Furthermore, the optimum compromise configuration for the UAV using wing-tail-horizontal tail rotation angle has a drag 1.18% higher than the optimum compromise configuration using wing-tail design variables, but with zero pitching moment. In fact, the obtained results for the optimised UAV with U-tail are consistent with the results obtained in [29], even though the optimisation in that research was performed for a differing flight regime and included optimisation of single objective.

6.4.3 Comparison of the optimised configurations for the UAV with inverted V-tail shape using wing and wing-tail

As expected, the optimised configurations for the UAV using wing-tail design variables achieved better performance than the optimised configuration using wing design variables only. Figure 6-40 shows these configurations.

The endurance ratio for the optimised configuration for the UAV with respect to wing-tail design variables simultaneously, improved by 1.49% compared to the optimised configuration using only the wing variables, and by 5.43% compared to the base design. It has a lower mass than the UAV base design and the optimised configuration for the UAV using only wing design variables. Also, the mass for the optimised configuration for the UAV by wing design variables only is slightly higher than the mass of the base design. It is clear from Figure 6-41 that combining wing and tail in the optimisation process provides better performance in both endurance ratio and UAV mass. In conclusion, using wing-tail design variables in an aerodynamic optimisation provided better UAV performance than using wing design variables only.

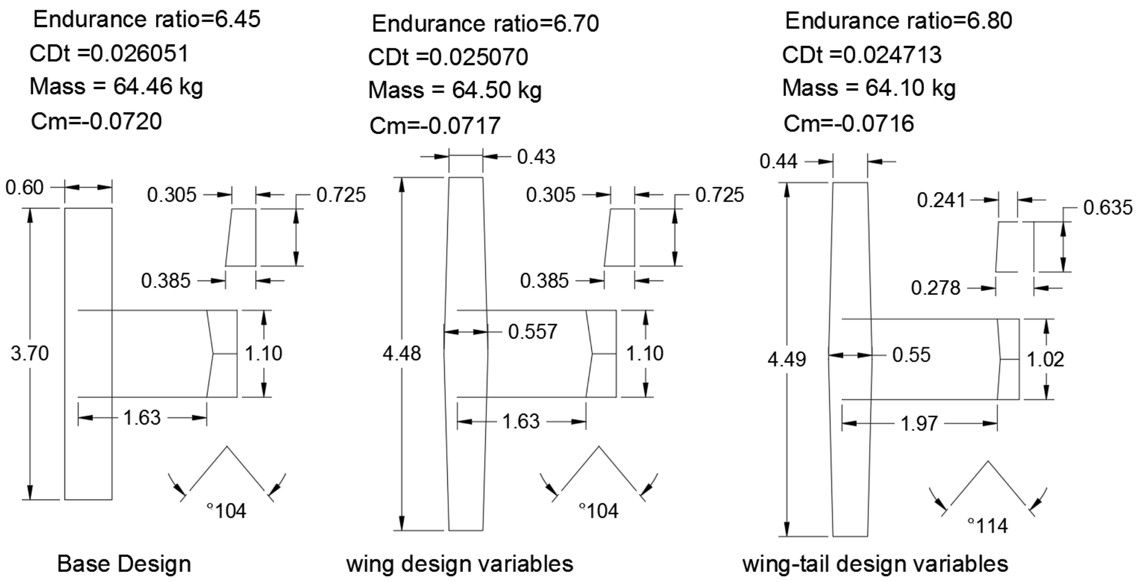


Figure 6-40: Comparison of optimised configurations for the UAV with inverted V-tail shape: base design (left), wing only (middle) and wing-tail (right)

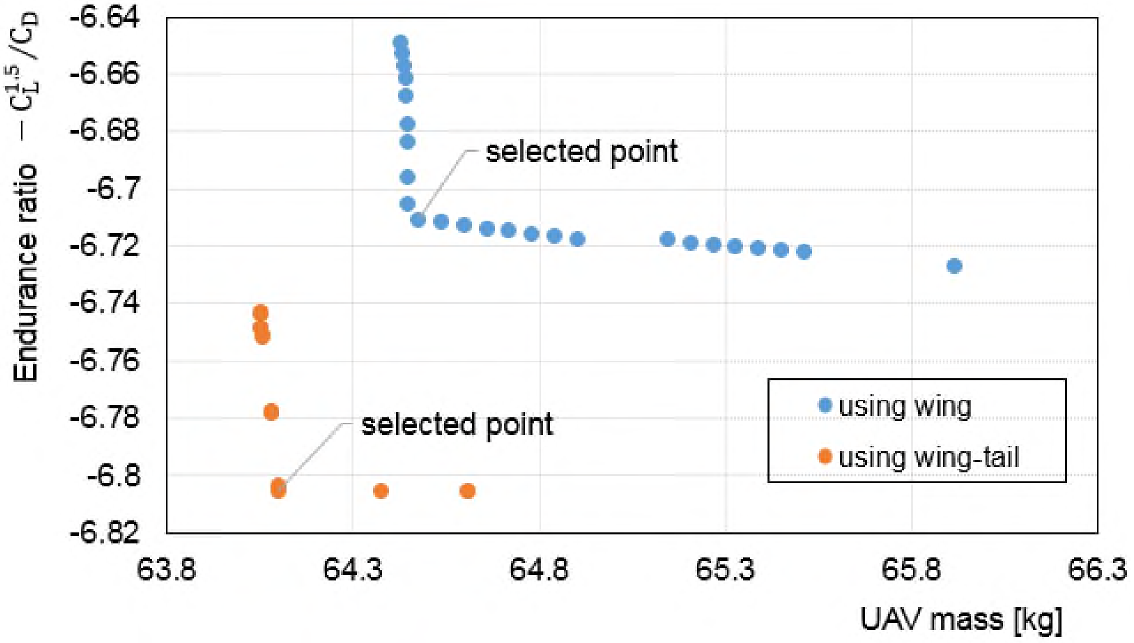


Figure 6-41: Pareto front of the optimised configurations for the UAV with inverted V-tail shape using wing and wing-tail

6.4.4 Comparison of the optimised configurations for the UAV with U-tail and inverted V-tail shapes using wing-tail design variables simultaneously

In Figure 6-42, we compare the optimised configurations for the UAV with U-tail and inverted V-tail shapes, where the optimised configurations were obtained by varying wing-tail design variables simultaneously under pitching moment constraint. The optimised configuration for the UAV with U-tail shape achieved higher improvement in the endurance ratio than the optimised configuration for the UAV with inverted V-tail, when compared to the base design. However, the optimised UAV with U-tail shape has an endurance ratio 1.18% lower than that for the optimised UAV with inverted V-tail shape

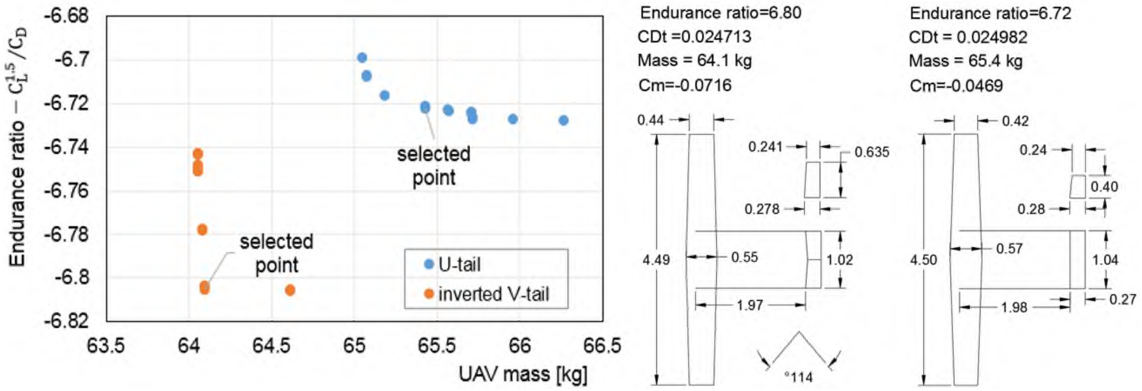


Figure 6-42: Left panel; obtained Pareto front for optimised UAV with U-tail and inverted V-tail using wing-tail design variables: right panel comparison of configurations for two selected solutions

6.5 Conclusions

In this chapter, a strategy for the aerodynamic shape design optimisation of the aircraft utilizing the Nimrod/O tool has been used effectively to explore the design space and quickly provide useful information regarding the design problem. The efficiency and reliability of the proposed framework have been demonstrated through the design of the Aegis-UAV. Six design scenarios were examined to optimise both endurance ratio and total mass by including wing planform, tail components, and horizontal tail rotation angle either separately or simultaneously.

On the other hand, the analysis and visualisation of sweep results that were performed before the optimisation process provided an insight into how the objective functions behave with respect to the design variables, which was very helpful with formulating the optimisation problem. Furthermore, by direct observation of the Pareto front for each solution, using parallel coordinates visualisation, the design variables' bounds were redefined efficiently.

The optimisation was performed using five regions and evaluations ranging from 1300 to 5500. Multiple runs were performed for most of the design cases and the results were mostly similar for each design case, which is considered to be supporting evidence for the robustness and the suitability of using MOTS for aerodynamic shape design optimisation of aircraft in general (more figures can be found in Appendix D.1). However, for confirmation, the authors selected optimisation results for one case (case 4) to be compared with the leading multi-objective GA, NSGA-II. Comparison supports the effectiveness of using the MOTS algorithm in such a design problem (see Appendix D.2).

Generally, the optimisation results show that the optimised UAV with inverted V-tail shapes has a higher endurance ratio and lower mass than the optimised UAV with the U-tail shape. However, the improvements in endurance ratio and mass with respect to the base design are better in the case of the UAV with the U-tail shape. The endurance ratio of the optimised UAV with U-tail is improved by 4.41% and 5.83% whereas for UAV with inverted V-tail improved by 3.89% and 5.43%, respectively, when using wing and wing-tail design variables. It is evident that combining wing and tail in the optimisation process provides better performance in both endurance ratio and UAV mass.

The improvement in the endurance ratio for the compromise solutions for both configurations of the Aegis UAV was achieved with zero reduction in the mass when the wing design variables are used. However, a reduction in mass of 0.91% and 0.56%, was obtained with the compromise solutions for UAV with U-tail and inverted V-tail respectively, when the wing-tail design variables were used. In fact, the ability of the optimiser to improve the endurance without any penalty on the UAV total mass reflects an efficient algorithm. Generally, the methodology is reliable, effective, and able to provide a substantial amount of information concerning the design problem to the DM.

7 Chapter - Results, Observations and Discussion for the Interactive Approach

Even though optimisation is the key to developing the optimum solution for a given design problem, the increase in model complexity and limitations of even supercomputers make finding those solutions difficult within realisable computation times, especially when high fidelity computation is used. In such optimisation techniques, the whole design space is explored; however, the DM in most practical applications is interested only in a part of the Pareto front. That can be achieved by steering the search to focus on the ROI rather than explore the whole design space. Using such a technique, the computational time is reduced and the complexity of data analysis that usually exists with a posteriori approaches is avoided. This chapter introduces the results of interactive optimisation including observation and discussion. The benefits and superiority of the interactive approach have been demonstrated by comparing the optimisation results obtained in the previous chapter using the MOTS algorithm (case 3) with the interactive approach results for the same design case obtained using the I-MOPSO algorithm.

7.1 Experiment configuration and setting

By using interactive optimisation, adding the human element to the decision making process, we aim to shorten the path for finding the optimal solution. Here, we were investigating the Aegis UAV configuration with a U-tail shape (case 3, in Chapter 6). The design optimisation problem contained two objectives and nine design variables. The objectives were to minimise the endurance ratio $\left(-\mathrm{C}_{\mathrm{L}}^{3/2}/\mathrm{C}_{\mathrm{D}}\right)$ and structural mass (UAV$_{\text{mass}}$). The optimisation process included simultaneous consideration of wing and tail surfaces to obtain the non-dominated solutions. The design variables are wingspan (b_{w}), wing root ($\mathrm{C}_{\mathrm{r\_w}}$), wing taper ratio ($\lambda_{\mathrm{w}}$), horizontal tail volume (V_{ht}), vertical tail volume (V_{vt}), tail arm (L_{t}), horizontal tail aspect ratio ($\mathrm{AR_{ht}}$), vertical tail aspect ratio ($\mathrm{AR_{vt}}$), and vertical tail taper ratio (λ_{vt}).

This case was selected as it had the greatest improvement and the highest number of design variables. Also, we have a relatively large pool of existing evaluations from prior experiments using the Multi-Objective Tabu Search (MOTS) algorithm.

The work started by integrating the Interface-AVL with the I-MOPSO framework. Two files were amended: the input text file and runner.py script file. The input file was used to define the design space parameters and the algorithm characteristics, and the runner.py script was used to call the Interface-AVL. The input file was equipped with an instruction to simplify the insertion of new data. To ensure a fair comparison, the experiments had the same, previously used, object function, design variables, upper and lower bounds, and constraints (see Appendix E.1).

For MOPSO setting, an inertia weighting of $\mathrm{w} = 0.4$, and constant weightings of $\mathrm{c}_1 = \mathrm{c}_2 = 2$ were used. A Gaussian distribution was selected to generate the initial population around the base geometry with a mean of $\mu = 0.0$ and a standard deviation of $\sigma = 0.2$ [15,76].

Once the I-MOPSO was executed, the code gathered the data from the input file into the various script files within the I-MOPSO framework. Next, the runner.py script called the Interface-AVL to provide the design variables after scaling and translation, and to receive the values of the objective functions. The code, by default, is expected to read a list of the objectives as below.

$FinalResults$:

$[flag \qquad obj1, \qquad obj2]$

Where *flag* can have the value either 0 or 1 and *obj1*, *obj2* are the values of the objective function. *flag* will equal unity when the results are valid and equal zero when the results are invalid. An outcome is considered invalid when it does not satisfy any of the constraints. To separate the invalid results from the valid results of the objective functions, a value of 100 was assigned to both of the objectives when the *flag* was equal to zero.

7.2 Optimisation process and results

To demonstrate the benefit and superiority of the interactive approach, it was compared to the a posteriori approach. We performed aerodynamic shape optimisation of the Aegis UAV interactively by including wing and tail design variables simultaneously, as stated in Section 7.1. Figure 7-1 shows the optimisation results obtained using MOTS with 5500 evaluations. From our previous results [83], we know that the problem is highly constrained and the number of infeasible solutions is much more than the feasible solutions. Because our algorithm, MOTS, is stochastic, we performed five runs to evaluate the optimisation performance, see case 3, Chapter 6. Of these five runs, on average, only 597 trial solutions were valid, and 4903 were invalid.

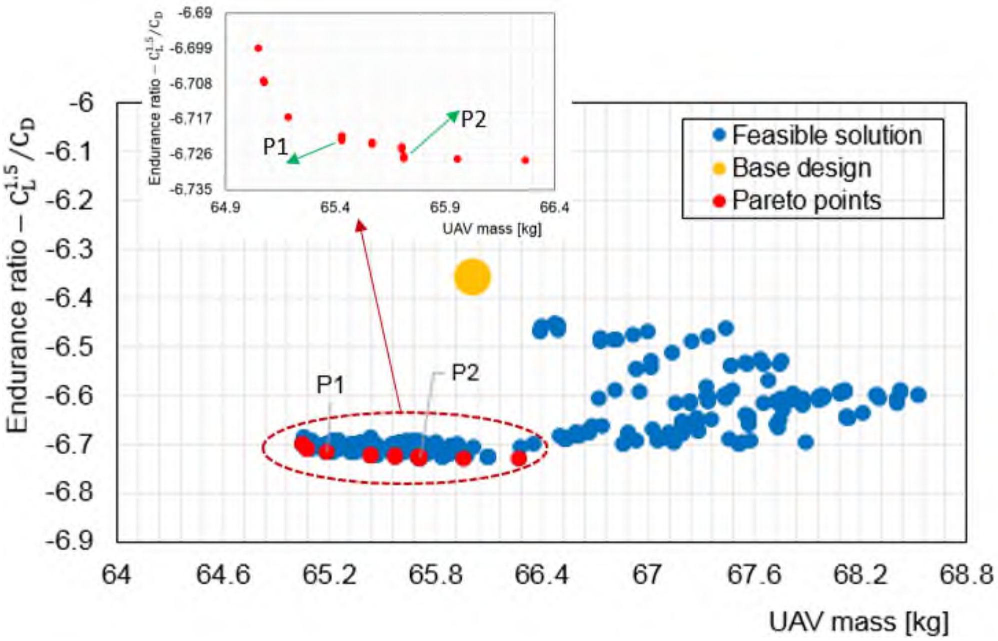


Figure 7-1: Feasible solutions obtained by using 5 regions and 5500 evaluations by MOTS (case 3, Chapter 6)

For the interactive runs, the DM is interested in steering the solutions to the defined ROI with a significantly lower number of evaluations than used in the a posteriori approach (MOTS), Figure 7-1. The region of interest was established based on our experience of optimising the Aegis UAV for different cases (Chapter

6). However, obtaining similar results to those presented in Figure 7-1 with around half of the evaluations will be a good result in our case. The limitations for the ROI were defined as:

$$f_1 = \frac{\mathrm{C}_\mathrm{L}^{1.5}}{\mathrm{C}_\mathrm{D}} \leq -6.7 \quad (Endurance\ ratio\ objective) \tag{7-1}$$

$$f_2 = UAVmass \leq 66.0\ kg \quad (UAV\ mass\ objective) \tag{7-2}$$

To discover and investigate the visualisation tools implemented in the obtained framework, see Figure 4-6, several runs were executed, both interactively and non-interactively, using various combinations of iterations and initial numbers of particles. The aim was to understand all the features and controls that appeared on the interactive screen, to obtain an idea of an adequate number of particles and iterations, and whether it is better to increase the number of particles or iterations during an evaluation. Even though this comparison is not the goal of this work, it was necessary to make this assessment for efficient optimisation whether interactive or non-interactive

Once the preliminary study concerning the I-MOPSO-AVL performance was completed, interactive and non-interactive optimisation results were compared, in order to highlight the reduction of computational time while achieving similar or even better results. As discussed in Section 7.1, the DM by interactive optimisation attempted to deliver solutions identical to the one obtained using MOTS for 5500 evaluations, with fewer evaluations. Firstly, a comparison of non-interactive MOPSO and MOTS for 5500 evaluations was performed, see Figure 7-2. Considering the stochastic characteristics of the MOPSO algorithm, ten independent runs using the same algorithm setting were carried out. The non-interactive MOPSO evaluations (5500) were divided into 110 iterations with a swarm population of 50 (particles). Next, the best case was compared with the previous results from MOTS. To some extent, the MOTS algorithm showed better performance than the non-interactive MOPSO, and it was obvious that using I-MOPSO to achieve solutions similar to the one obtained by MOTS with fewer evaluations would be a challenging task for the DM. In Figure 7-2 the red dashed box represents the ROI when performing the optimisation interactively.

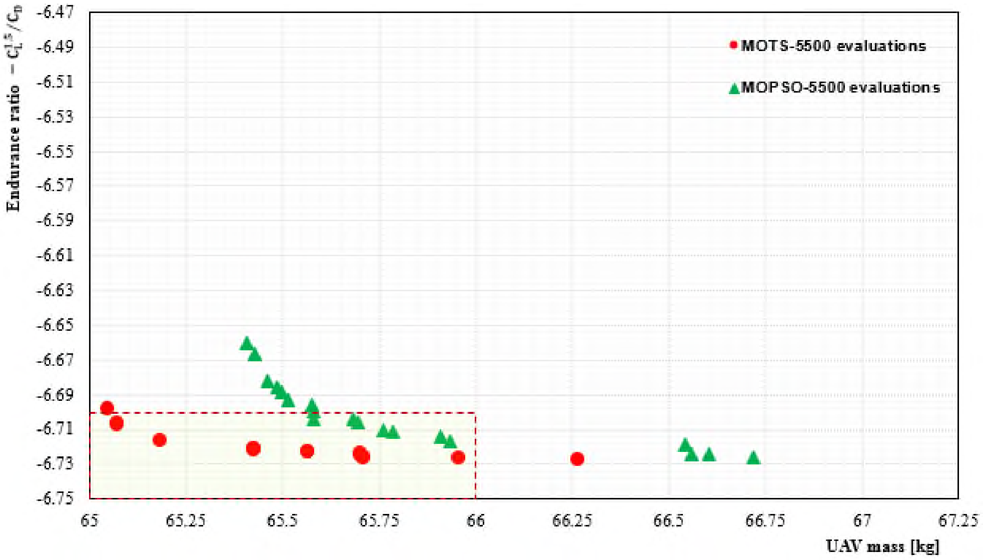


Figure 7-2: Pareto front using MOTS and MOPSO for 5500 evaluations. The red dashed box represents the ROI when performing the optimisation interactively

The next step was to begin interactive optimisation to obtain solutions within the ROI as described in Sections 4.3.2 and 4.3.3. The interaction with the optimiser began after 15 iterations since the non-interactive results showed that only a few particles were able to satisfy the applied constraints within the first 10 iterations. By means of adding the human element to the interaction, we aimed to shorten the path for finding the optimal solution, i.e., less time consumed and a significant reduction in project costs. The obtained results showed that when the number of evaluations reached 2000, the DM began to produce results inside the ROI. To obtain more solutions within the ROI, the number of evaluations was increased to 3000. Figure 7-3 compares the optimisation results using MOTS and MOPSO for 5500 evaluations with I-MOPSO for 3000 evaluations. With the latter, the DM was able to guide the optimisation process to within the region of interest using only 3000 evaluations. Steering the optimisation process to the region of interest prevents the algorithm wasting time exploring the whole design space. This approach consumes less time while providing a more flexible approach for any design problem.

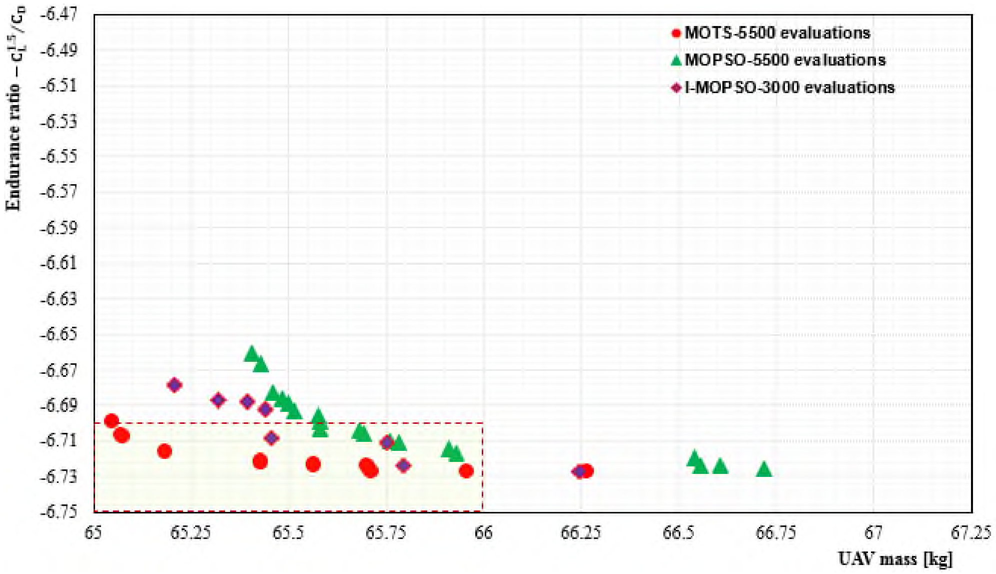


Figure 7-3: Comparison of Pareto fronts for MOTS and MOPSO using 5500 evaluations with I-MOPSO using 3000 evaluations.

An additional demonstration of the superiority of undertaking optimisation interactively, is presented in Figure 7-4 which shows a comparison of interactive and non-interactive results for MOTS and MOPSO with I-MOPSO for 3000 evaluations. The superiority of interactive optimisation is evident. The non-interactive results from MOTS and MOPSO were not able to generate solutions inside the ROI, whereas including the DM in the optimisation process allows the search to focus on the area of the Pareto front that is of interest.

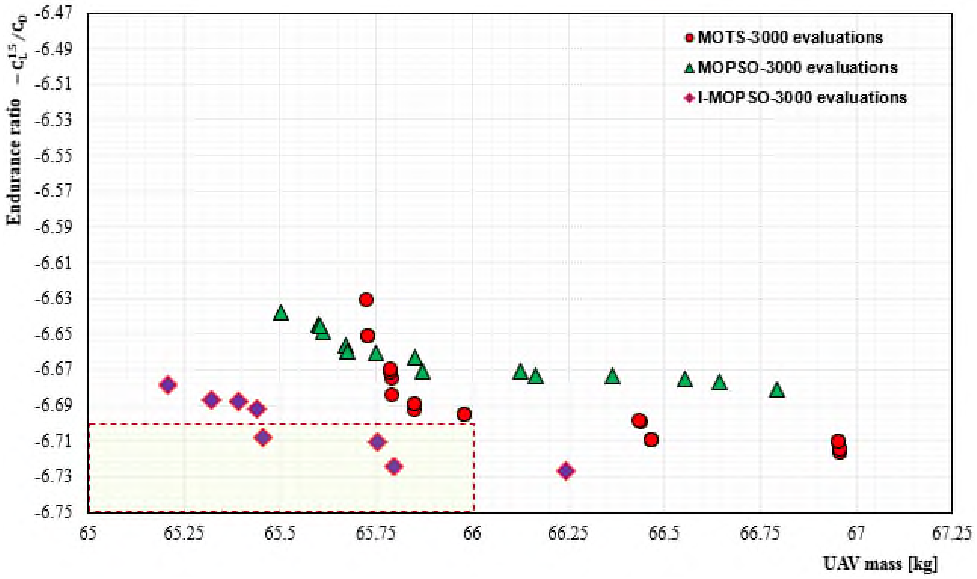


Figure 7-4: Comparison of Pareto front using MOTS and MOPSO with I-MOPSO, each using 3000 evaluations. Note the superiority of using interactive optimisation

In fact, with interactive optimisation, the DM does not need to continue the optimisation process to the end, since s/he is able to stop the simulation at any iteration that gives a solution that satisfies the design requirements. Figures 7-5, 7-6 and 7-7 demonstrate the dynamic behaviour of the Pareto front at 1000, 2000, and 2500 evaluations.

It was evident that at 1000 evaluations the Pareto solutions cover a vast design space, then as the iterations increased, the solutions accelerated quickly to cover less design space. This is because the DM pushed the optimiser to focus on the solutions that appear within the correlation, and ignore other points that are not of interest. Of course, the DM benefitted from both visualisations tools available on the interface screen, Parallel and Scatter Cartesian coordinates, to guide the simulation to the affected region for the interactive solutions (see Section 4.3.3). Furthermore, Figure 7-7 shows clearly that the simulation can benefit from interactive optimisation since the solutions were within the ROI with only 2500 evaluations. It is evident that interactive optimisation accelerated the optimisation process and achieved optimal results. Thus, integrating the DM with the optimisation search enabled faster convergence to the optimal solution.

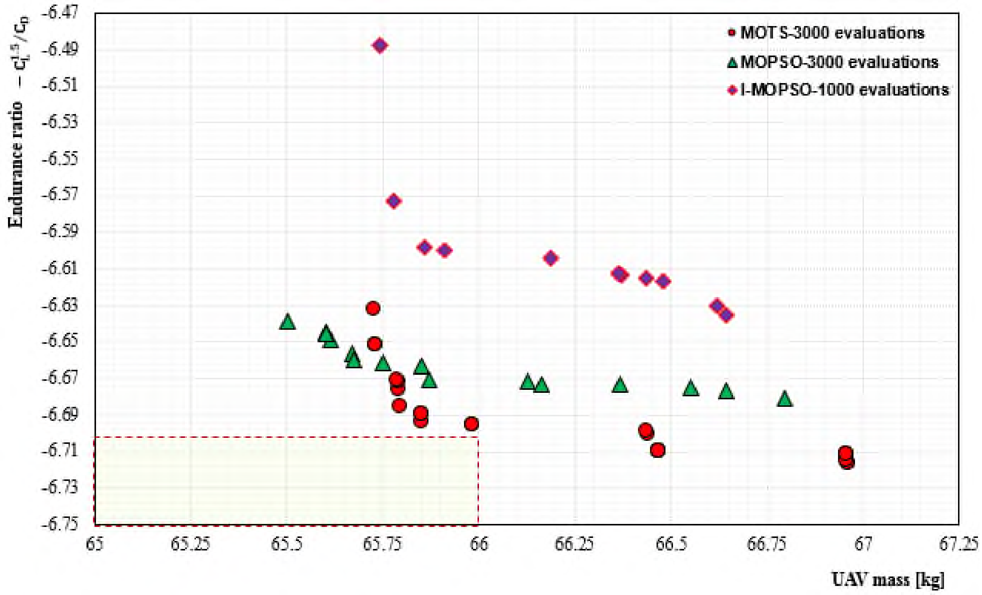


Figure 7-5: Comparison of Pareto front using MOTS and MOPSO for 3000 evaluations with I-MOPSO for 1000 evaluations

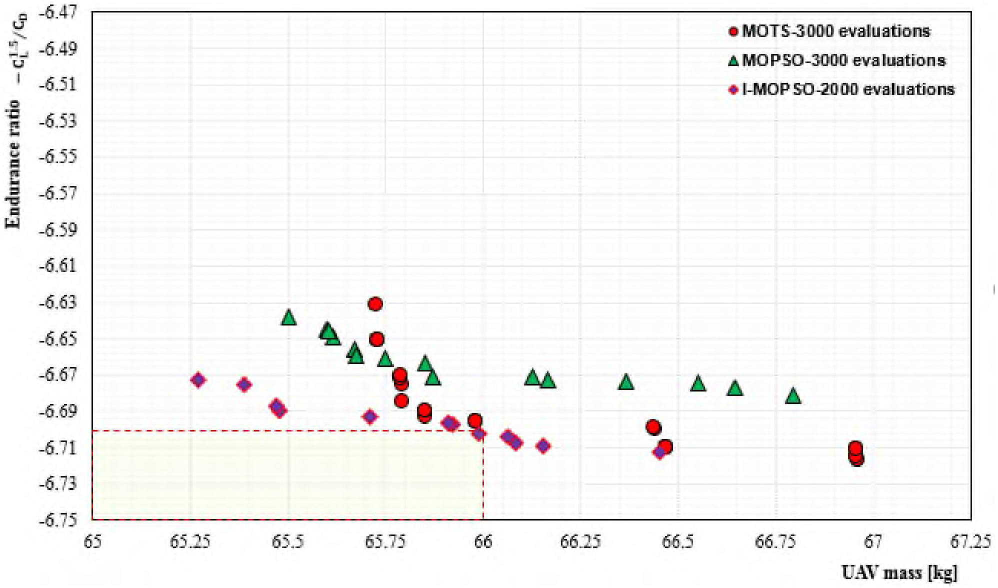


Figure 7-6: Comparison of Pareto front using MOTS and MOPSO for 3000 evaluations with I-MOPSO for 2000 evaluations; it is possible to view the solution at any number of iteration and assess whether or not it is converging

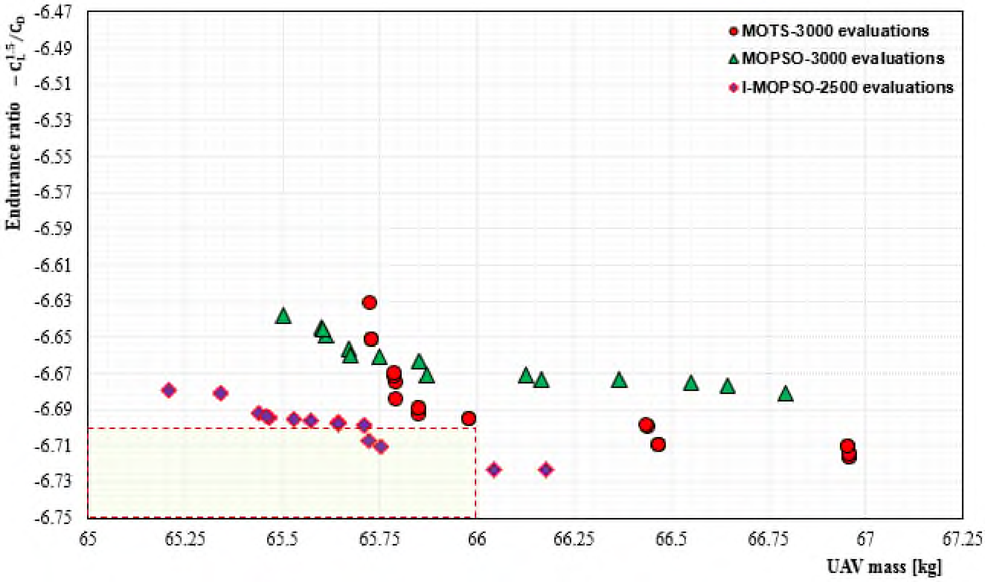


Figure 7-7: Comparison of Pareto front using MOTS and MOPSO for 3000 evaluations with I-MOPSO for 2500 evaluations; with the interactive approach it is possible to stop the run as and when the required solutions are obtained

7.3 Visualisation of results using Parallel Coordinates

Parallel Coordinates has been shown to be a good technique for visualisation, analysis and the study of large amounts of data [83]. This approach allows observation of the relations between design variables, trade-offs between objective functions, and can monitor the evaluation process [90–92]. Furthermore, it is stressed that such a visualisation tool can be used to display the characteristics of a trend and correlations that exist among and between the design space parameters [201]. Figure 7-8 shows a comparison of the results for various optimisation scenarios used in this work; MOTS for 5500 and 3000 evaluations, non-interactive MOPSO also for 5500 and 3000 evaluation, and Interactive Multi-Objective Particle Swarm Optimisation (I-MOPSO) for 1000, 2000, 2500, and 3000 evaluations. It is obvious that by using the Parallel Coordinates techniques, the DM can highlight the number of runs that satisfies the constraints (ROI) which are highlighted when the ROI is selected, see the two red rectangular boxes on Figure 7-8 that represent the objective functions.

Generally, the trends achieved display the correlation that exists between the design variables and the objectives. Besides, such an approach helps the understanding of the influence of the design variables in combination on the solutions within the ROI. Thus, the DM can now relatively easily identify the effective range for each design variable. Also, it becomes clearer which are the design variables with limited range or a range wide enough to satisfy the design requirements. In fact, it is a responsibility of the DM to select which solutions best fit the requirements for the next stage of the design.

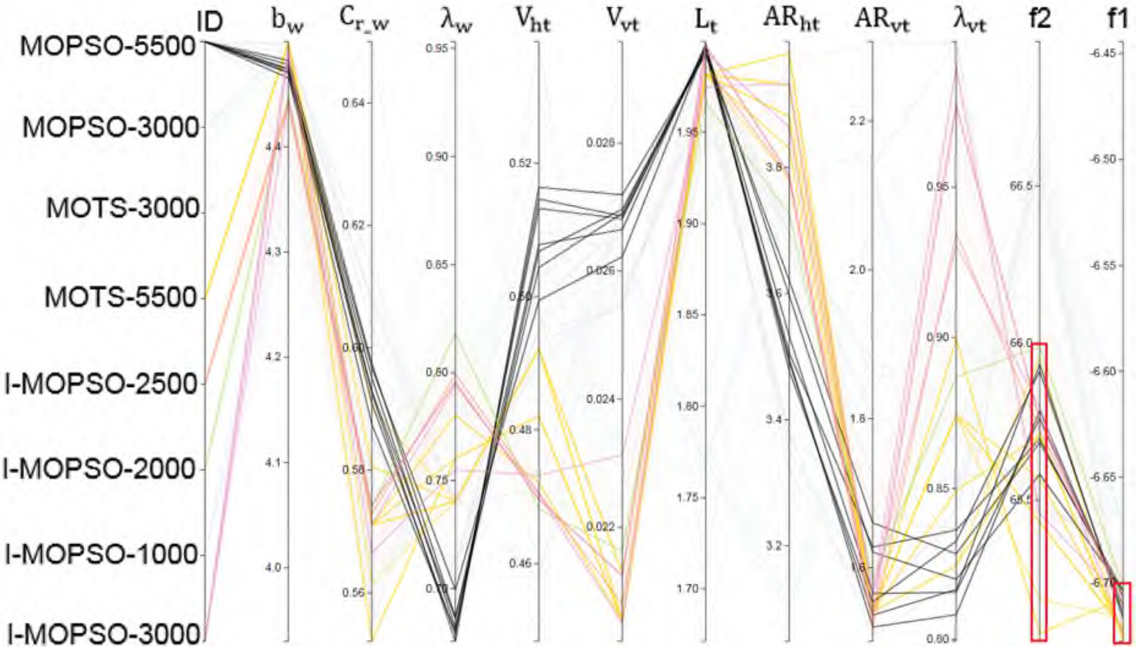


Figure 7-8: Optimisation case able to satisfy the DM's interest; for the region of interest see the red rectangles on f1 and f2

For example, visualization and analysis of the results displayed using Parallel Coordinates give more details about values of the design variables which participate in achieving high-performance solutions. It was clear that the maximum values of span (b_w) and boom length (L_t) always participated in provision of high performance (results within the ROI). However, there was more scope to vary the values of the design variables; vertical tail taper ratio (λ_{vt}), horizontal tail volume (V_{ht}) and vertical tail volume (V_{vt}) and still maintain high performance. In fact, this is the idea behind multi-objective optimisation, several conflicting objectives are optimised simultaneously as a function of various design variables and constraints, to identify non-dominated solutions (since there is no single optimal solution). It is a DM's job to analyse the non-dominated solutions to select the most appropriate solution satisfying his/her requirements.

To obtain optimal geometry for the UAV with respect to wing and tail design variables simultaneously, the optimiser had to find compromise solutions that included both wing surface area and tail volume [8,122]. Since the wing sizing is related to tail sizing, which has as its primary function to simultaneously counter the moments that will be produced by the wing, and also satisfy stability requirements.

Visualisation of the results shows the difference in the ability of the optimisation algorithms to meet these requirements. For example, when the MOPSO for 5500 evaluations selected higher values of vertical and horizontal tail volume (higher tail volume, i.e. surface tail area multiplied by arm length), the obtained total mass (f_2) was higher than that obtained by MOTS for 5500 and I-MOPSO for 3000 evaluations. However, the optimiser in this case (MOPSO-5500) was trying to reduce the weight and improve the endurance ratio because low values for wing taper ratio (λ_{w}) and vertical tail taper ratio (λ_{vt}) had been selected. On the other hand, MOTS for 5500 and I-MOPSO for 3000 evaluations selected almost identical values of the design variables that satisfy the requirements of optimum objective functions. Note, however, that the vertical tail taper ratio (λ_{vt}) was obviously not identical for both algorithms. The lower UAV mass (f_2) obtained by MOTS for 5500 evaluations is a direct confirmation of the previous results. However, it is very evident that interactive optimisation with 3000 evaluations provided very competitive results when compared to the results obtained for 5500 evaluations using either MOPSP or MOTS.

In spite of the successes obtained using interactive optimisation, we noticed that the interactive solutions (Pareto solutions) are concentrated around certain values of each design variable, whereas non-interactive solutions were more widely spread. As a result, each Pareto solution for the non-interactive process consist of diverse values for the design variables, which is preferred by the designer [21]. In addition, the non-interactive solutions were obtained by exploring a wide range of the design space whereas, in the interactive approach, the search is steered by the DM once the run is started, providing solutions obtained by less exploration of the design space, see Figures 7-9 and 7-10. In fact, this is the major penalty for interactive optimisation, we may miss information in the design space despite the fact that we are achieving high optimality corresponding to our design requirements in much less computational time - which is the primary focus of the DM in this case.

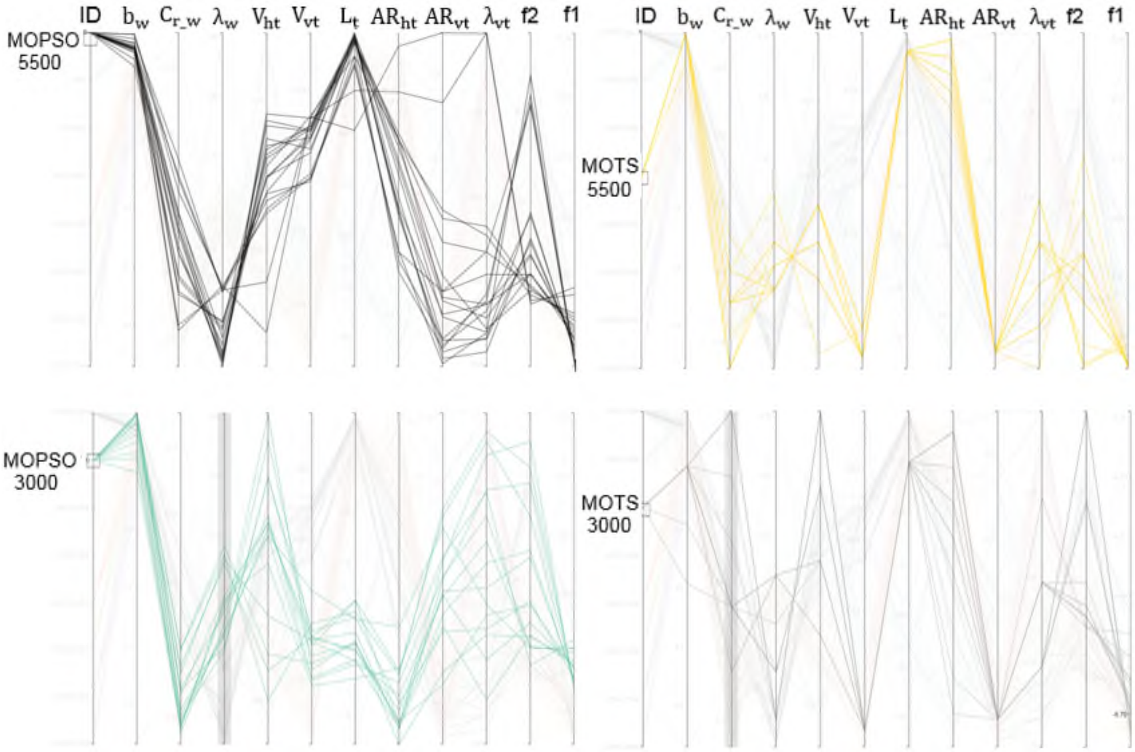


Figure 7-9: Objective functions as a result of particles spread widely across the entire design space in case of non-interactive optimisation

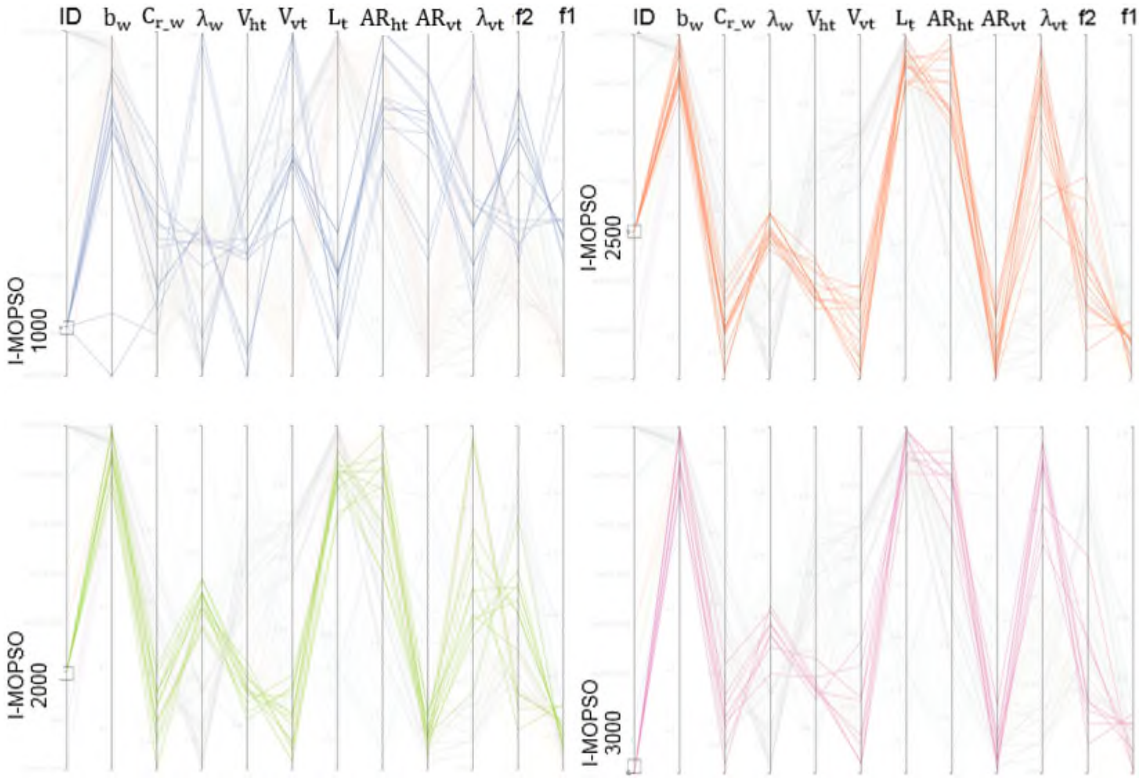


Figure 7-10: Objective functions as a result of particles with limited variation in the case of interactive optimisation

7.4 Investigation of selected configurations

Finally, selected configurations from non-dominated solutions were considered for further analysis. Figure 7-11 compares the full Pareto Front, obtained using MOTS and MOPSO for 5500 evaluations and I-MOPSO for 3000 evaluations. A visual inspection of the Pareto front for each run can provide useful information to DMs about the possible trade-offs between objectives. Configurations P1, P2, and P3 were selected as a compromise solutions since we have the same interest in minimising both objectives. It is obvious that P3 provided better improvements than P2 in endurance ratio and UAV mass. On the other hand, P1 showed a slightly better improvement in the endurance ratio and UAV mass than P3. But it must be born in mind that P3 was obtained with only 3000 evaluations compared to 5500 evaluations for P1 and P2.

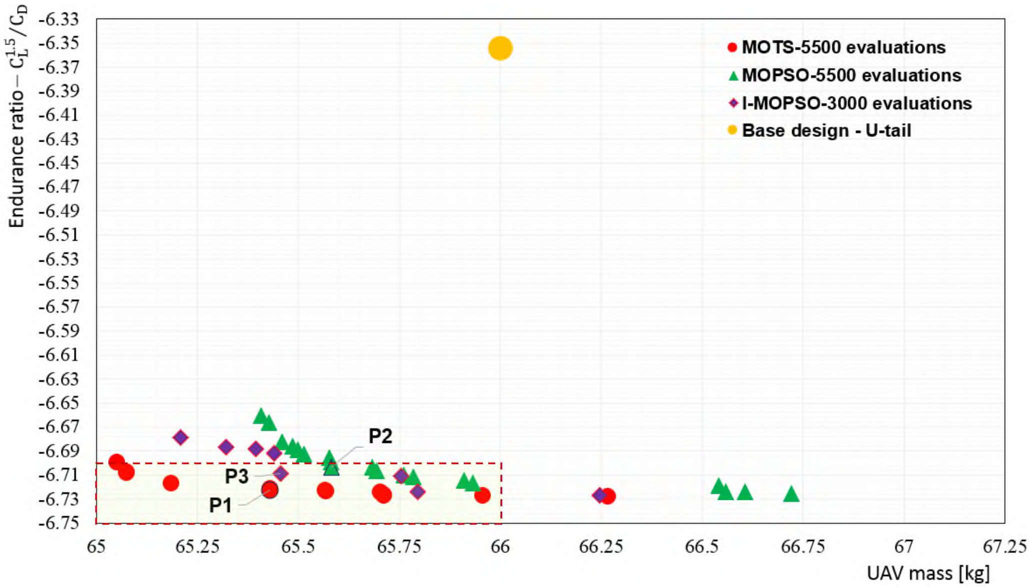


Figure 7-11: Pareto fronts for all runs and selected configurations compared with base design, it is evident that interactive optimisation provided very competitive results using only 3000 evaluations.

Figure 7-12 shows the geometry of the selected compromise solutions and the performance of each one numerically. In fact, all the optimised configurations have a similar performance. The interactive compromise solution P3 has a drag 5.34% lower than that for the base design geometry while P1 and P2, respectively, have a drag that was 5.49% and 5.27% lower than that for the base design. The drag decreased from 264.3 counts for the base design to 249.8, 250.2, and 250.4 counts, respectively for P1, P3, and P2 at level flight. Furthermore, the optimised configurations using either I-MOPSO or MOPSO have a lower absolute value of pitching moment than the P1 configuration and so will gain a lower increment of drag when better trimmed for level flight, which should lead to better performance.

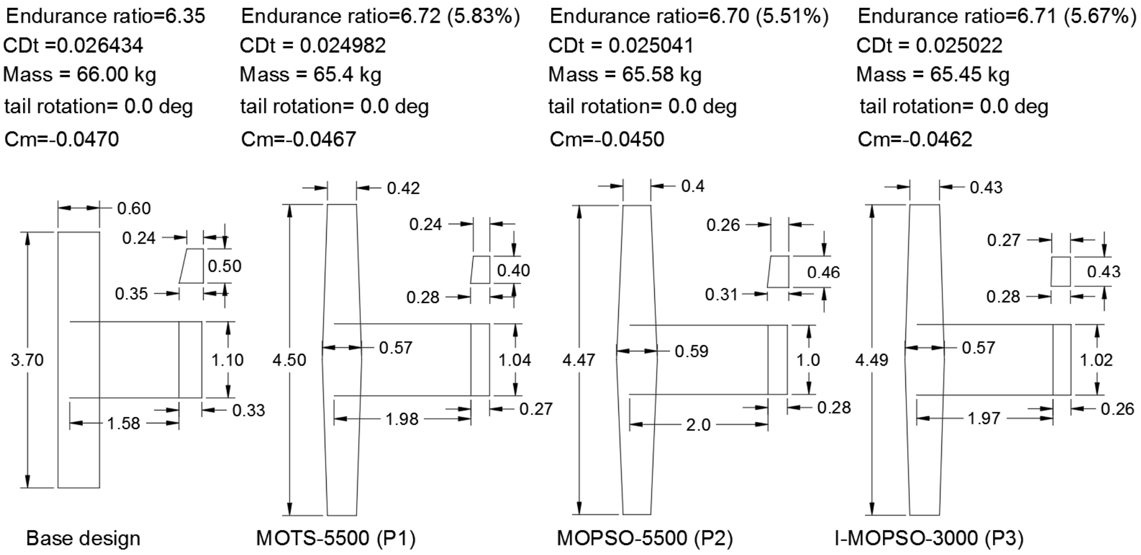


Figure 7-12: Comparison of detail configurations for optimum compromise solutions obtained using MOTS-5500, MOPSO-5500 and I-MOPSO-3000 with the Aegis UAV base design

As a further study, the aerodynamic characteristics for each of the optimised configurations and base design were simulated for different angles of attack, see Figure 7-13. It is obvious the configuration obtained by I-MOPSO has a high lift to drag ratio for all angles of attack, which is coincident with the other optimised configurations. All configurations are stable longitudinally, laterally, and directionally. It is evident that interactive optimisation requires less computational time and does not degrade the aerodynamic performance of the optimised configurations.

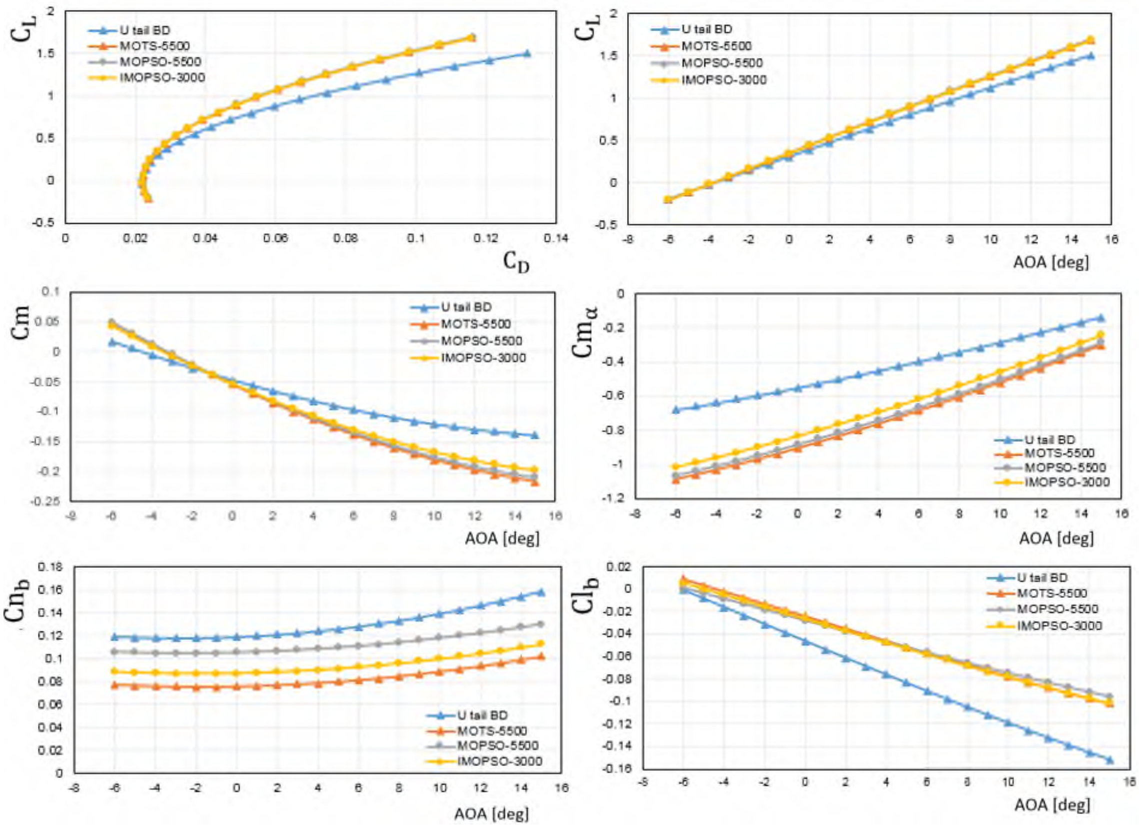


Figure 7-13: Comparison of the aerodynamic performance of the I-MOPSO-3000, MOTS-5500, and MOPSO-5500 configurations with the Aegis UAV base design

7.5 Conclusion

This chapter has investigated the efficiency and effectiveness of interactive optimisation in the design of the aerodynamic shape of the Aegis UAV with U-tail. The proposed framework is flexible and able to obtain high optimality solutions in shorter computational times. It combines interactive Multi-Objective Particle Swarm Optimisation with a low fidelity flow solver, AVL. We compared the interactive results for I-MOPSO with non-interactive results for MOPSO and MOTS algorithms for the whole UAV with U-tail shape. The obtained results using interactive optimisation show the ability of the DM to use his/her preferences effectively, to steer the search to the ROI without degrading the aerodynamic performance of the optimised configurations. Even using only half of the evaluations, the DM was able to obtain results similar to those obtained by a posteriori approaches. An advantage was that it was possible for the DM to stop the search at any iteration, which is not possible in a posteriori approaches even when solutions do not converge or may be infeasible.

This remarkable increase in performance is accompanied by a significant reduction in computational time. For example, comparing the interactive and non-interactive results obtained using the MOPSO algorithm shows that by almost halving the number of total evaluations, the computational time has been cut by a third (33.4%). Each 5500-evaluation run (non-interactive) required about 195 minutes to be completed, whereas each 3000-evaluation run (interactive) took around 130 minutes. The computational time used by the interactive approach includes the time required by the DM to interact with the optimisation search. The fall in computational time is, in fact, remarkable. In this experiment, a Desktop computer with an i7-6700 CPU processor was used to install the Virtual Machine framework.

In spite of the successes obtained using interactive optimisation, interactive solutions were concentrated within a limited range for each design variable whereas non-interactive optimisation solutions were widely spread. Thus, interactive optimisation may miss some important information regarding the design space despite achieving high optimality corresponding to the design requirements in substantially less computational time - which was a primary aim of the DM in this case. In the next chapter the work on optimisation of results will be enhanced and extended using an Artificial Neural Network performing deep space exploration to retain all the useful information contained in the design space in an adequate computational time with high optimality.

8 Chapter - Results, Observations and Discussion on the ANN Approach

We have achieved optimal solutions using interactive optimisation, requiring much less computational time than the non-interactive optimisation approach. However, we have shown in Chapter 7 that we could miss some important information contained in the design space, since the interactive method starts to focus on certain areas early in the search. Moreover, the DM does not always succeed in guiding the search to the region of interest because of the stochastic characteristics of the algorithm. To overcome these problems, an ANN is used to increase the performance of the optimisation.

This research has made a contribution to knowledge by using the ANN to guide the optimisation algorithm, by deciding whether a trial solution is, or is not, worth a full evaluation. It was demonstrated that the aerodynamic shape optimisation problem is highly constrained, and there are many more invalid solutions than valid [13]. Thus, it would require a considerable time to evaluate the non-worthwhile solutions. However, by using an ANN in parallel with the optimiser, any trial solution could be checked by the ANN before computing. To make a fair and worthwhile comparison for the same design problem, the results obtained using the ANN were compared with results obtained previously using MOTS, MOPSO and I-MOPSO. This comparison of the methods is based on the efficiency, quality of the solutions, and flexibility of each technique.

8.1 Experiment setting and process

In order to perform fair comparisons, the same design variables and their bounds were used; design case 3 for the Aegis UAV. The structure of the input file used by ANN-MOPSO to define the information required by the external mode is shown in Appendix E.2. The input file contains the following information; the number of design variables, number of objectives, location and name of the external mode, upper and lower bounds of the design variable and scaling of the objective functions.

Once the input file is prepared, the experiment is starting by typing a command in the command line. It is permitted to select the number of particles, number of iterations, training procedure, and scepticism percentage. Table 8-1 summarise the parameters used. The continuous live training approach was used for the ANN, and the trial solutions suggested by the optimiser were classified as valid or invalid. The continuous live training of the ANN was performed in parallel with the evaluation of the objective function by the Interface-AVL. The minimum size of the training set for the ANN before starting classification was defined as 500. However, several experiments were performed using training sets of 300, 400, and 500. Then, as the run started, the ANN classified the trial solutions as valid or invalid, as explained in Section 4.4.1.

The time used by the continuous training of the ANN after each iteration was negligible when compared to the cost of the objective function evaluations [11]. Several experiments were performed to compare single and continuous training. The results showed that continuous training is a better approach for the aerodynamic shape design optimisation problem, see Appendix E.3.

Table 8-1: Summary of the ANN-MOPSO parameters used to perform the optimisation

Parameter	Evaluations= 3000	Evaluations= 5500
particles	30	50
Iteration	100	110
Training approach	continuous	continuous
Initial training set size	500	500
Scepticism	15%	15%
Archive	Live	Live

8.2 Results and analysis

The experiments started by allowing for 5500 objective function evaluations. The evaluations were divided into sets of 100 iterations and 55 particles. The continuous training approach was selected with a minimum size of training set of 500. Because the algorithm used possesses stochastic characteristics, five-runs were performed. Figure 8-1 shows a comparison of MOTS, MOPSO, and I-MOPSO with ANN-MOPSO for 5500 evaluations. The results using ANN-MOPSO shows a slightly better improvement by moving towards the centre of the Pareto front. However, some of the MOTS and I-MOPSO solutions obtained a higher reduction in mass than ANN-MOPSO solutions. More analysis was performed using parallel coordinate technique and simulations, and is discussed below.

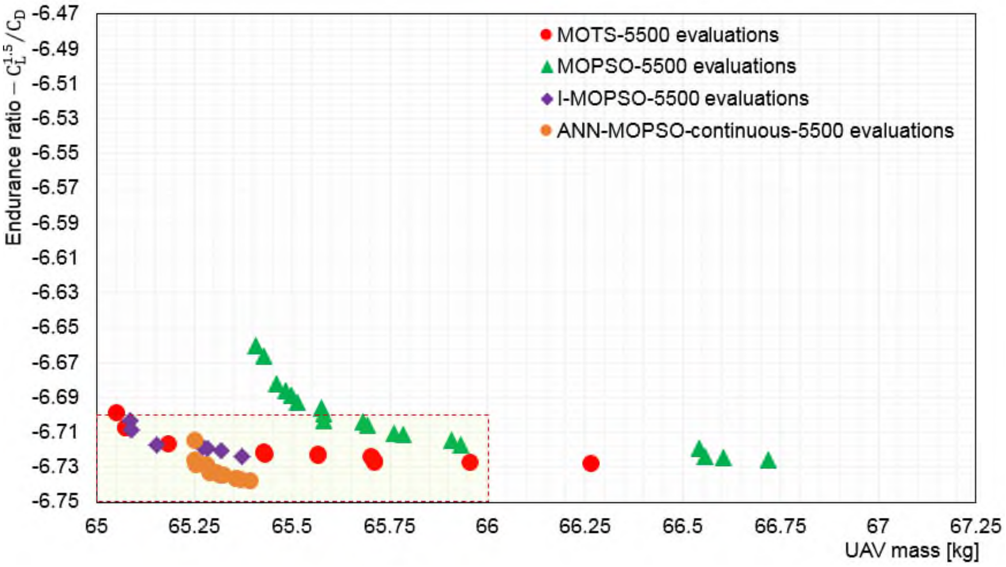


Figure 8-1: Comparison of Pareto fronts using MOTS, MOPSO, and I-MOPSO with ANN-MOPSO all for 5500 evaluations

In order to identify the gains in efficiency obtained by using an ANN in the aerodynamic shape design problem, a comparison of ANN-MOPSO results obtained using 3000 evaluations with MOTS, MOPSO, I-MOPSO results obtained for 5500 evaluations is shown in Figure 8-2. The ANN-MOPSO for 3000 evaluations was able to provide results within the ROI, similar to the optimisation results obtained by MOTS, MOPSO, and I-MOPSO which were for 5500 evaluations. It is evident that by using of the ANN the computational time required to obtain results within the ROI is reduced.

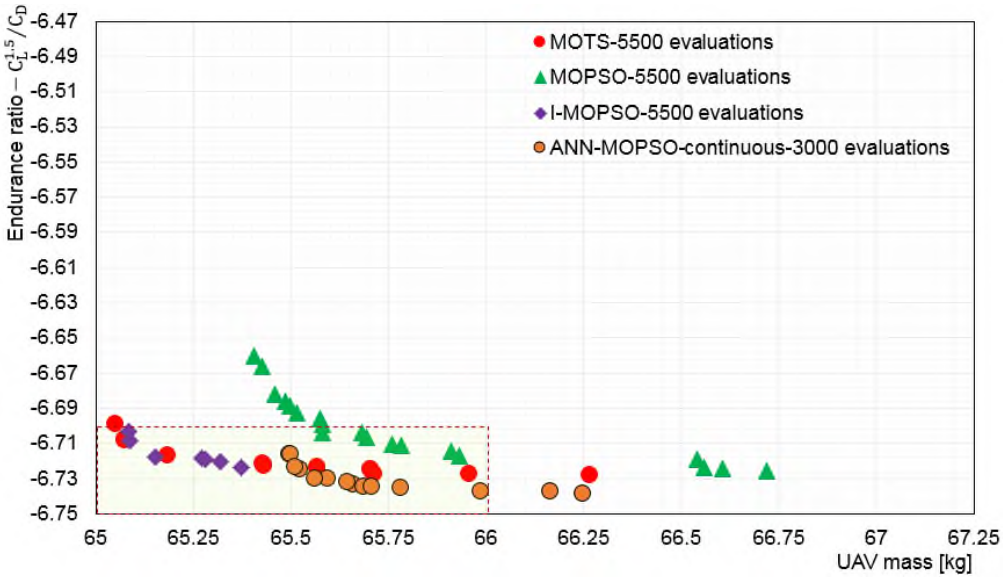


Figure 8-2: Comparison of Pareto front using MOTS, MOPSO, and I-MOPSO for 5500 evaluations with ANN-MOPSO for 3000 evaluations

As an additional demonstration of the superiority of using ANN in the optimisation process, a comparison of the results obtained using ANN-MOPSO, MOTS, MOTSO, and I-MOPSO for 3000 evaluations is presented in Figure 8-3. The superiority of using the ANN to guide the optimisation is obvious. The ANN-MOPSO solutions are better than the interactive results. It is evident that the ANN has efficiently self-trained and succeeded in detecting the invalid particles. This strongly suggests that using an ANN has enabled the optimiser to outperform the other approaches used here. Visual comparison of the Pareto front shows the solutions using an ANN are well inside the ROI and cover both objectives fairly when compared to the solutions obtained using other approaches with the same number of evaluations.

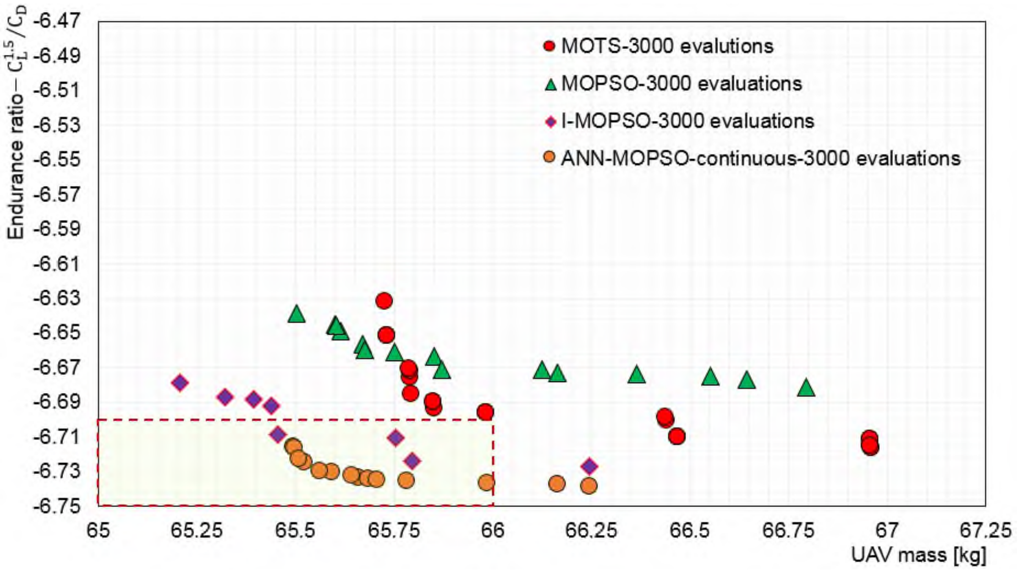


Figure 8-3: Comparison of Pareto fronts using MOTS, MOPSO, IMOPSO and ANN-MOPSO all for 3000 evaluations

Figure 8-4 shows a comparison for the simulations using both ANN-MOPSO and MOTS for 5500 and 3000 evaluations. It is clear that as the simulation continues the search it becomes more tightly constrained, and the ANN-MOPSO began facing difficulties in improving the solutions further, which means it had almost reached an optimal solution for this design case. In fact, since the ANN is under continuous training, the exploration regions change smoothly through the generation of new particles in the areas of interest until highly optimal solutions are obtained. The convergence of the solutions towards the centre of the Pareto front is strong evidence of the success achieved by the optimisation search

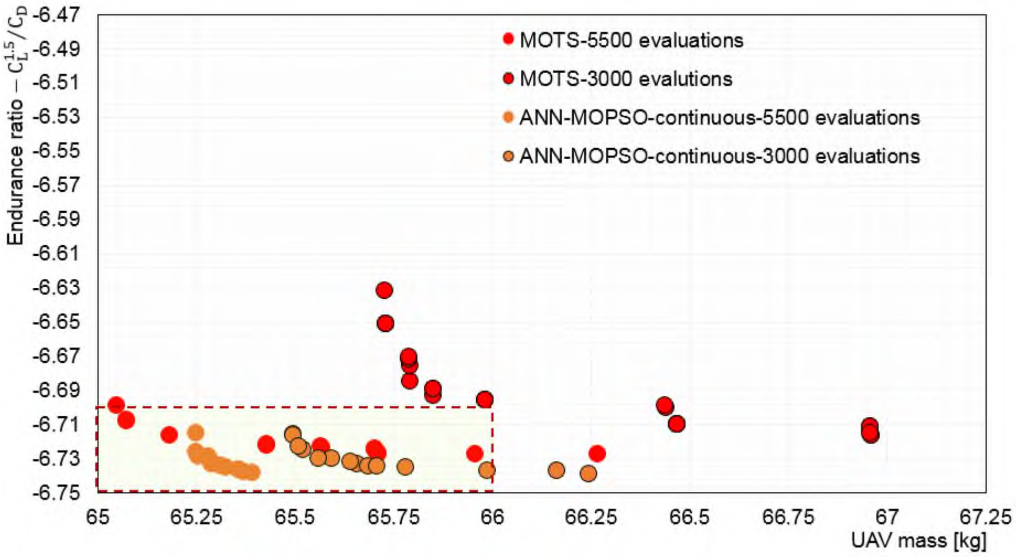


Figure 8-4: Comparison of the Pareto optimal solutions using ANN-MOPSO and MOTS for both 5500 and 3000 evaluations. Note, the Pareto solutions becomes more condensed as the simulation continues

Since the ANN is used to guide the optimisation algorithm by detecting invalid particles, it is of interest to display the differences between valid and invalid particles for each approach. Table 8-2 confirms that the ANN was a significant tool for spotting invalid particles. The results justify the use of an ANN since the DM would have more particles that are valid for almost the same computational time. For example, adding the ANN to MOPSO increased the number of valid particles from 988 to 2238 for 5500 evaluations, and from 529 to 1006 for 3000 evaluations.

Table 8-2: Numbers of valid and invalid particles generated by MOPSO and ANN-MOPSO for 3000 and 5000 evaluations

For 5500 evaluations				For 3000 evaluations		
Code	Valid	Invalid	Time [m]	Valid	Invalid	Time [m]
MOPSO	988	4512	195	529	2471	120
ANN-MOPSO	2238	3258	212	1006	1992	131

It is evident that the ANN is an effective, fast evaluator for deciding if a trial solution passed by the optimiser is worth evaluating, or not. Using the ANN-MOPSO improved the number of the valid particles by 126% and 90% compared to the MOPSO alone, but incurring a penalty of 8.0% and 8.3% in computational time, respectively for the 5500 and 3000 evaluations. This time penalty is a small proportion of the time required to evaluate the trial solutions. The difference in computational times is the penalty for using an ANN to classify whether a trial solution is valid or not, for the evaluation and regeneration of new particles by the optimiser. In addition, it is an interesting observation that the ANN-MOPSO for 3000 evaluations obtained a higher number of valid particles than the MOPSO for 5500 evaluations. This was a strong indication of the positive effect of using ANN for optimising solutions to design problems.

8.3 Data visualisation and analysis using Parallel Coordinates

Figure 8-5 uses Parallel Coordinates to present and analyse the results for the ANN-MOPSO 3000 evaluations, and highly optimal compromise solutions were obtained. The solution obtained using the ANN-MOPSO 5500 evaluations provided an even better compromise for the objective function.

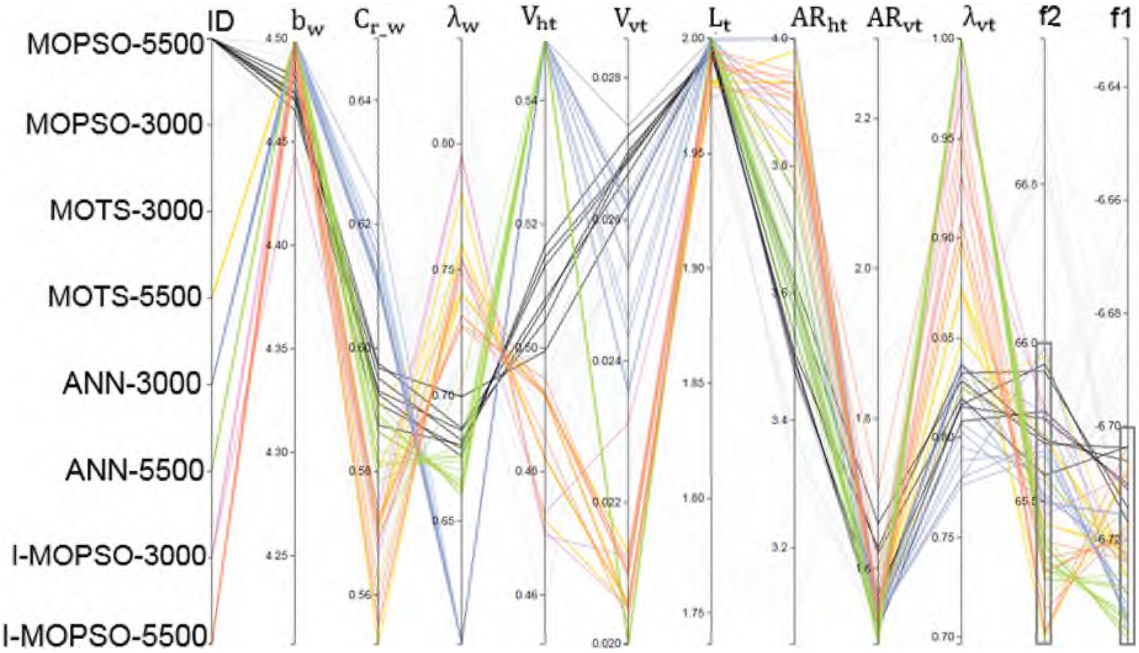


Figure 8-5: Comparison of different optimisation approaches that led to Pareto optimal solution within the ROI

One of the early observations, when visualising the data using parallel coordinate, were the strong correlations obtained when the ANN was used to guide the optimisation process. Study and analysis of these trends revealed correlations between the design variables and objective functions, see Figure 8-6. It brought to light meaningful multivariate patterns and comparisons that helped when used interactively by the DM for analysis and design preferences. Particularly, it helped with identifying which of the design variables, or combination of design variables, particularly participated in obtaining optimal objective functions, i.e. it brought to the attention of the DM more than one design option that could be suitable for the next stage of the design process. For example, highly optimal solutions for the Aegis UAV were achieved by using almost the maximum value for each of the wing span $(\mathrm{b_w})$, horizontal tail volume $(\mathrm{V_{ht}})$, and tail arm $(\mathrm{L_t})$. Whereas, the wing chord $\left(\mathrm{C_{r\_w}}\right)$, wing taper ratio (λ_w), vertical tail volume $(\mathrm{V_{vt}})$, horizontal tail aspect ratio $(\mathrm{AR_{ht}})$ and vertical tail aspect ratio $(\mathrm{AR_{vt}})$ should take values around 0.59 m, 0.67, 0.02, 3.6, and 1.5, respectively.

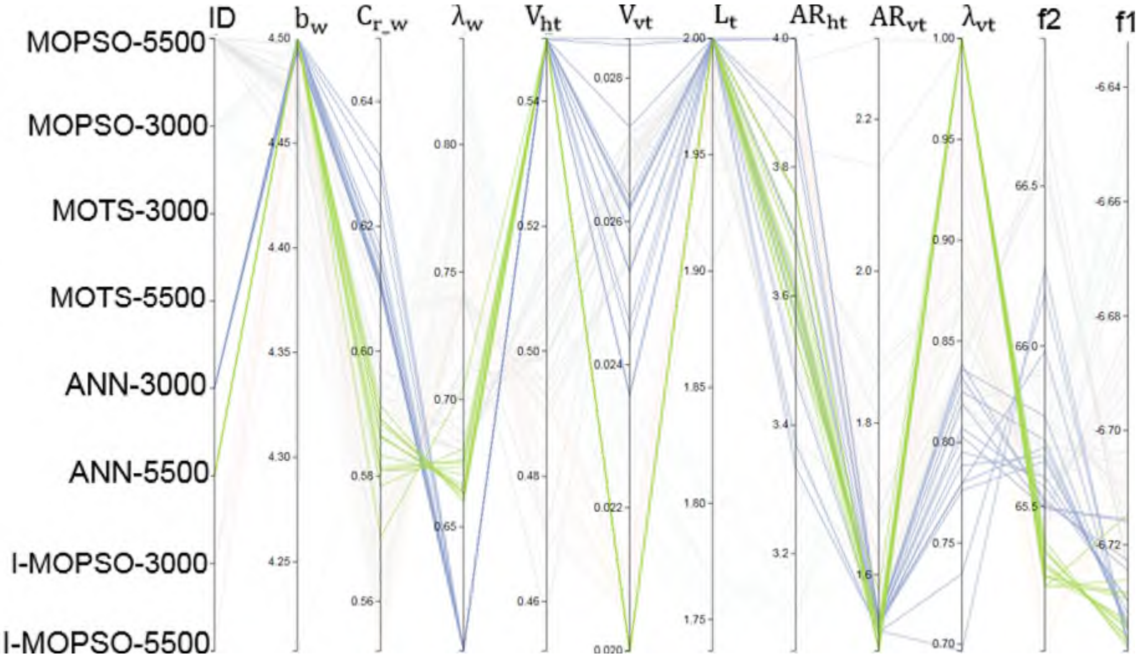


Figure 8-6: Clear trends and strong correlations show the success in the training of the ANN. The ANN-MOPSO for 5500 and 3000 evaluation gave the strongest correlation of all optimisation approaches

In fact, the optimisation is a complicated process where the optimiser is “playing” with nine design variables, seeking to obtain a set of optimal UAV configurations that satisfies both of the objectives but where, as described earlier, the objectives have conflicting requirements regarding the values of the design variables.

Figure 8-7 compares solutions obtained for 5500 evaluations using the ANN-MOPSO, MOTS and I-MOPSO algorithms. The solutions obtained using ANN-MOPSO provide the greatest improvement in the endurance ratio, but part of the solution gave less reduction of mass. It is obvious that the ANN solutions significantly minimised the drag by selecting lower wing taper ratio (λ_{w}) and a higher wingspan (b_{w}). On the other hand, the MOTS and I-MOPSO solutions selected lower values for the wing root chord $\left(C_{r\_w}\right)$, vertical tail taper ratio (λ_{vt}) and slightly higher horizontal tail aspect ratio (AR_{ht}), which resulted in a greater reduction in the mass. However, all solutions satisfied the design requirements, and offer different options for the designer in the next stage. Thus, some design variables can be considered as key for the designer to obtain the required final solution.

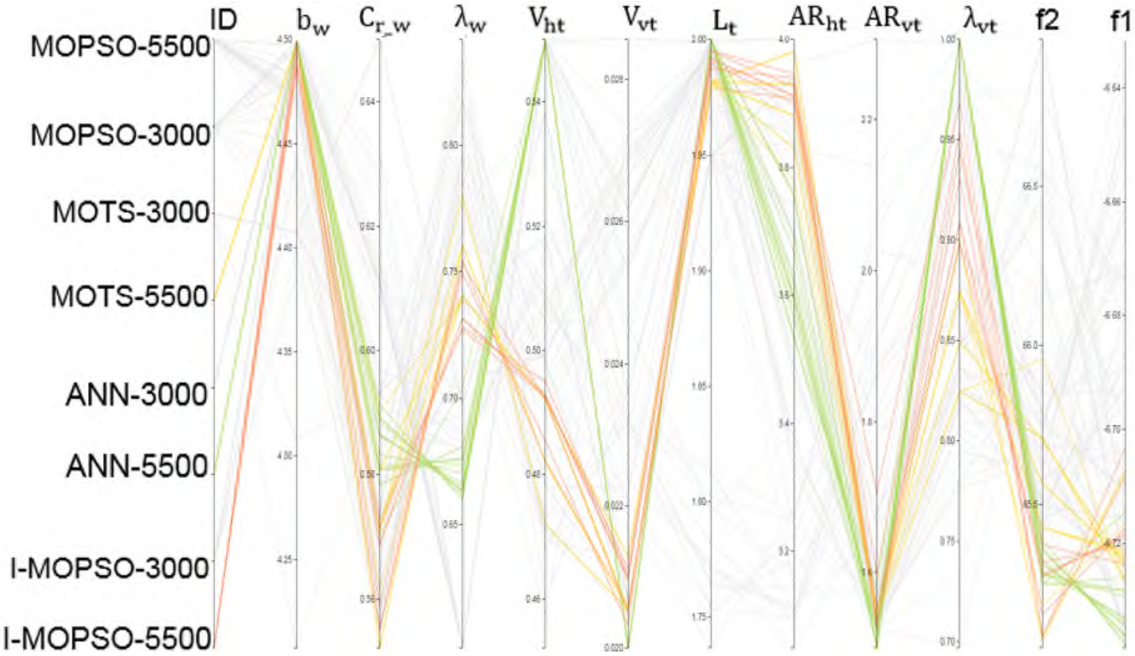


Figure 8-7: High optimality solutions for the multi-objective optimisation design problem can be achieved using various combinations of the design variables.

Note; some design variable can be considered as major whereas others are important in obtaining further improvements.

The patterns and correlations obtained using the ANN have an identical trend to the trend achieved by the DM when performing optimisation interactively. Figure 8-8 compares trends achieved using ANN-MOPSO and I-MOPSO for 5500 evaluations. The similarity in the trends is obvious. These observations highlight the effect of performing optimisation interactively. The correlations achieved by the DM to drive the results are similar to the correlations learned by the ANN to obtain highly optimal solutions.

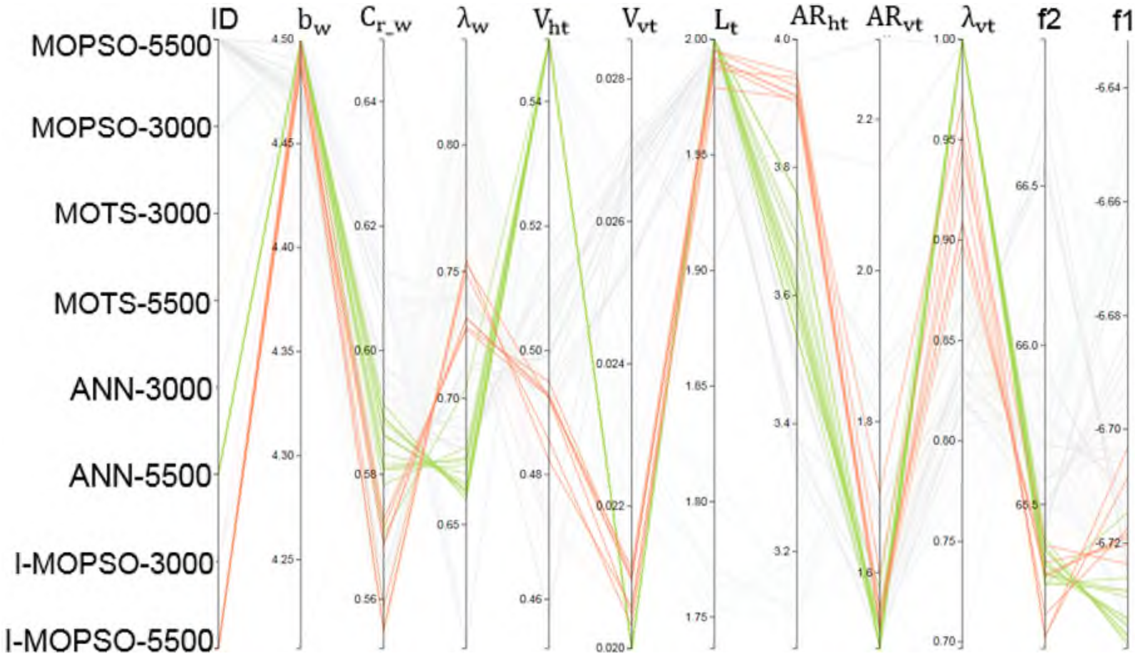


Figure 8-8: Comparison of the trends achieved using ANN-MOPSO and I-MOPSO for 5500 evaluations. The patterns and correlations obtained using ANN have an identical trend to that achieved by the DM when guiding optimisation interactively

8.4 Detailed study for selected solutions

In order to demonstrate the quality of the non-dominated solutions obtained under ANN guidance, a single compromise solution was selected for further simulations. Figure 8-9 compares optimal compromise solutions obtained using ANN-MOPSO for 5500 and 3000 evaluations, with MOTS and MOPSO for 5500 evaluations, and with the base design. The variations of non-dominated solutions with respect to the objective functions are obvious, giving the DM a deeper insight into the design problem to assist with trade-off.

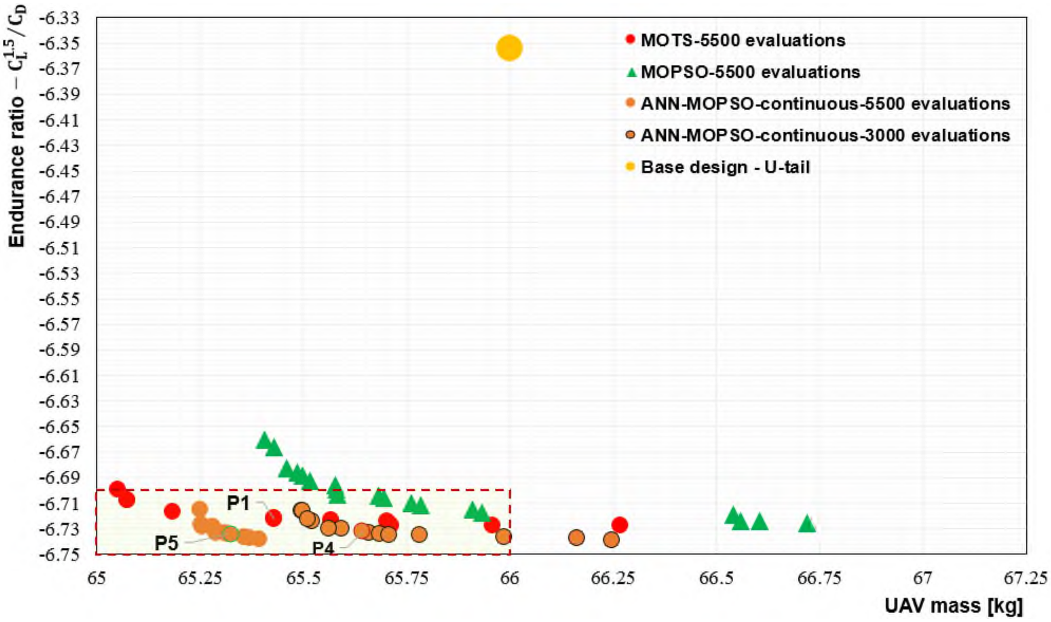


Figure 8-9: Comparison of non-dominated solutions obtained using ANN-MOPSO, MOTS, and MOPSO with U-tail base design

It is obvious that all the compromise solutions are within the ROI. The compromise solution P5 has the best improvements in both of the objectives compared to the compromise solutions P4 and P1. On the other hand, the compromise solution P4 has better improvement in endurance ratio than P1, but slightly less reduction in mass. However, it must be borne in mind that the compromise solution P4 was obtained after only 3000 evaluations whereas the compromise solutions P5 and P1 were obtained after 5500 evaluations.

The compromise solution P4 has an endurance ratio of 5.98% higher than that for the base design with 0.55% reduction in mass, whereas the endurance ratio for the compromise solutions P5 and P1 improved by 6.14% and 5.83% with 0.97% and 0.91% reduction in total mass, respectively compared to base design. The drag decreased from 264.3 counts at level flight (for the base design) to 249.3 counts, 249.5 counts, and 249.8 counts, respectively for P5, P4, and P1. Detailed comparison shows that P5 and P4 are 0.52 and 0.36 drag counts lower than P1, respectively. Thus, P5 has the lowest drag value, and P4 has the second lowest drag value, regardless of increasing the number of evaluations.

The drag reduction for the ANN-MOPSO-3000 optimal compromise solution was gained with a decrease in the negative pitching moment when compared to base design and P1. Which has a slightly positive effect on release the aircraft from drag increments at trimming, see Figure 8-10. From these results, we conclude that using an ANN to guide the optimisation by identifying invalid particles is a strong approach. Using only around half of the evaluations (P4), the ANN was able to obtain better results than those obtained using MOPSO and MOTS for 5500 evaluations. The success was due to its ability to better identify invalid particles and hence provide more valid particles in the same computational time.

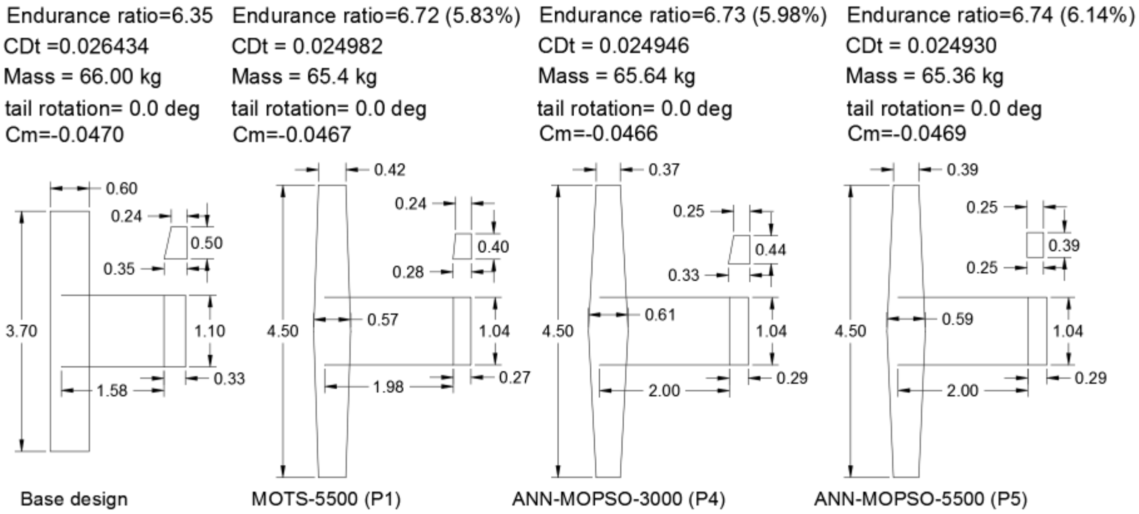


Figure 8-10: Detail configurations for the selected solutions compared with the base design

Further investigation is continuing, performing aerodynamic simulations for each of the above configurations as a function of various angles of attack. Actually, we are most interested in the configuration optimised using the ANN with 3000 evaluations, however, the configuration optimised using the ANN with 5500 evaluation is also included. Figure 8-11 shows a comparison of the aerodynamic characteristics of the selected configurations. It is observed that all the configurations show the same trends. However, as angle of attack increases there will be a positive deviation in lift coefficient, compared to the base design values in the all optimised configurations. Furthermore, the configurations optimised by the guidance of the ANN have a slightly better lift to drag ratio than P1 (optimised using MOTS). On the other hand, there is noticeable variation in stability characteristics for the all optimised configurations even though they satisfy all the design constraints. This behaviour is strongly demonstrated by the dimensions of the optimised configurations. A quick glance at the optimised configurations shows that the optimised configurations had wings with almost the same dimensions whereas there are noticeable differences in the tail dimensions.

It is obvious that the optimiser has reached a stage where no more space is available in the wing design variables to improve the objectives by making changes to the wing planform, whereas there is still more space to play with tail components.

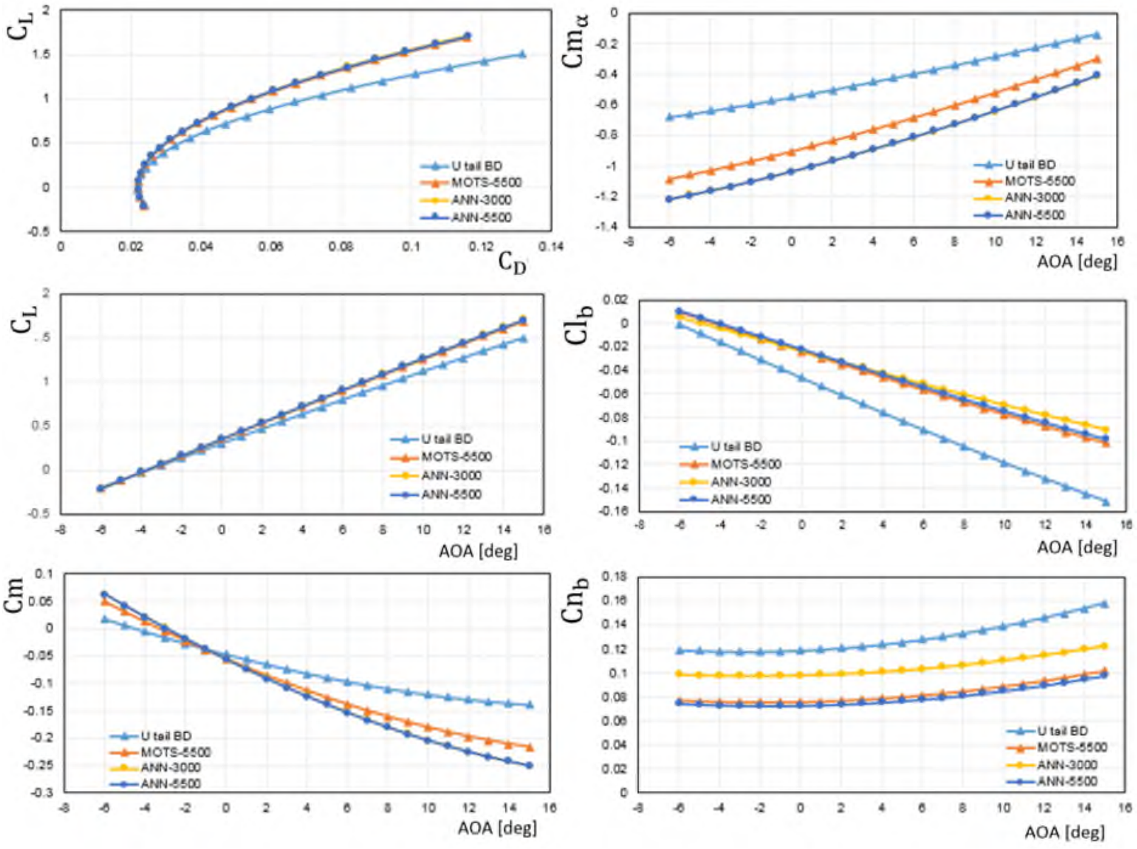


Figure 8-11: Comparison of the aerodynamic performance of the optimised configurations with the base design platform

8.5 Conclusions

This chapter presents the use of an ANN to aid in optimisation of a multi-objective design problem in order to reduce the workload on the designer by achieving effective solutions. The ANN is used here as a fast approximate evaluator to decide whether the trial solution by the optimiser is worth a full evaluation or not. It was considered proof that the aerodynamic shape optimisation problem is highly constrained and that there are many more invalid solutions than valid. It is not sensible to spend a long time evaluating the non-worthwhile solutions.

The study of the optimisation results showed the success of the ANN in increasing the number of worthwhile particles efficiently. The number of valid particles grew from 988 to 2238 (126%) and from 529 to 1006 (90%), respectively for 5500 and 3000 evaluations using ANN-MOPSO. In addition, the results showed that continuous training is a better approach for the aerodynamic shape design optimisation problem. It is proven that the time used to train the ANN and then identify the invalid particles is small compared to the time for the evaluation of the objective functions. Furthermore, it is evident that by adopting the continuous live training approach, the ANN-MOPSO allows for smooth exploration of the search regions by updated the search continuously when it has not succeeded in improving the solutions by replacing invalid particles with valid particles in regions of interest. That means it helps overcome the drawbacks in the stochastic characteristics of the algorithm, and simultaneously makes it less likely that any important data in the design space will be missed.

Demonstrating the effectiveness of the algorithm is achieved by comparing results for the ANN-MOPSO using 3000 evaluations with MOPSO and MOTS for 5500 evaluations and I-MOPSO for 3000 evaluations, subject to the same constraints and design variables. The obtained results strongly indicate that using the ANN to guide the optimisation algorithm is effective at increasing the convergence of the optimiser and obtaining highly optimal solutions. The algorithm incorporating the continuous training approach has been shown to be effective in terms of computational time and solution quality.

9 Chapter - Conclusions and Recommendation for Future Work

9.1 Conclusions

This thesis has presented key developments and applications of an optimisation framework that uses stochastic, multi-objective optimisation, combined with Artificial Intelligence and interactive techniques, for the aerodynamic design of aircraft shapes. The efficiency and reliability of the proposed framework have been demonstrated through the design of the Aegis-UAV. The obtained results show that the proposed techniques provide highly optimal solutions in less computational time than conventional approaches. The main idea was to focus all the computational efforts on worthwhile solutions and areas of interest rather than exploring and evaluating all particles in the design space.

In the beginning the initial strategy, using the sweep method, efficiently identified design space limitations and provided an insight into how the objective functions behaved with respect to the design variables. The study and analysis of the results with the help of visualization techniques were used effectively to eliminate any dysfunctions which existed in the developed framework. Then, the sweep method results were successfully used as feedback to redefine the constraints and re-formulate the design problem.

Once the design space was explored and the sensitivity of the design variables with respect to the objective functions were found, an automatic optimisation process using Nimrod/O was performed. This allowed the efficient exploration and investigation of various design scenarios, which led to the acquisition of a larger amount of information about the limitations and requirements of the design problem before committing to a single design.

The efficiency and reliability of the proposed strategy has been demonstrated through the aerodynamic shape design optimisation of the Aegis UAV with U-tail and inverted V-tail shape. The MOTS algorithm in the form of non-dominated solutions presents the optimised solutions. The optimisation results for the UAV’s configurations showed that the improvements in endurance ratio and mass with respect to the base design were better in the case of the UAV with the U-tail shape even though the optimised configuration for the UAV with inverted V-tail shape had a higher endurance ratio and lower mass than the optimised UAV with the U-tail. Generally, the methodology was reliable, effective, and able to provide a large amount of information about the design problem to the DM at this stage.

Since the improvement in model efficiency in terms of the computational time and design quality are of most interest in this research, the methodology was adapted to interactive-optimisation using the MOPSO algorithm. This method aimed to overcome the disadvantages of the a priori and a posteriori approaches. The interactive optimisation was used to focus the search on a certain area of the design space, since the designer in real life is interested only in a certain area (the ROI) or particular solutions. The superiority of undertaking optimisation interactively was demonstrated by comparison of interactive results obtained using I-MOPSO with non-interactive results obtained using MOPSO and MOTS algorithms for a single design scenario.

With only half of the evaluations used the non-interactive approach, the DM was able to obtain similar or even better results, i.e., it was obvious DM now had the ability to steer the search to the ROI with fewer evaluations. The more significant point is that the DM has the ability to stop the search at any iteration, which is not possible in the non-interactive approaches, even though the solutions may not converge or may not be feasible. Moreover, the complexity of post-data-analysis with the non-interactive approach was avoided, since the DM is involved in the search process. However, it was also obvious that interactive solutions had less variation in the values of the design variables since the interactive search started focusing on a certain search area early in the process. In addition, because our algorithm has stochastic characteristics the designer does not always succeed in guiding the search to the ROI, additionally, by optimising interactively the designer may overlook some important information about the design space.

In order to increase the efficiency of the framework used, an ANN was used to improve the performance of the optimisation algorithm by deciding whether the trial solutions of the optimiser were worth a full evaluation or not. This demonstrated that the ANN had succeeded in identifying invalid particles. Consequently, the optimisation process obtained more particles that were valid for almost the same computational time.

In one experiment, results showed the number of valid particles obtained using ANN-MOPSO compared to MOPSO for 3000 evaluations grew from 529 to 1006 (90% improvement) with a penalty of 8.3% (only 11 minutes) in the computational time. This extra time was used by the ANN to classify whether the trial solutions were valid or not valid for the evaluations and generate new particles by the optimiser. However, this time was negligible when compared to the time taken to evaluate the trial solutions. Another finding that demonstrated the efficiency obtained by using the ANN, was that the ANN-MOPSO for 3000 evaluations obtained a higher number of valid particles compared to the MOPSO for 5500 evaluations with significantly less computational time (a 33% reduction - 64 minutes). The number of valid particle was 1006 out of 3000 evaluations obtained using the ANN-MOPSO. In contrast, the number of valid particles was only 988 out of 5500 evaluations when the MOPSO algorithm was used alone. This is a strong indication of the effect of using the ANN for such design problems.

Finally, comparing the solutions obtained using ANN-MOPSO for 3000 evaluations with the I-MOPSO for 3000 and MOTS for 5500 evaluations shows the superiority of using an ANN within the optimisation process. The algorithm using the ANN obtained highly optimal solutions that are effective in terms of computational time and solution quality for the continuous training approach.

9.2 Recommendations for future work

It is recommended that future work focus on increasing the efficiency and quality of the solutions at the detailed stages of design by using high fidelity flow solvers while focusing on reducing the computational time. The formulation of the design problem could be improved to include multi-fidelity flow solvers. Future work is summarized in three points.

1 Engineers are always in need of computational design tools such as Nimrod/O to improve the design process in terms of execution time, reliability,

and user effort. Nimrod/O allowed us to easily integrate codes written in different languages. It was a matter of editing the text file that came with the tool to specify the optimisation procedures such as design variables, algorithm used, and the task to be executed and, in this research, that included the Interface-AVL. The Nimrod/O allows the designer to focus on the technical issues of the design problem instead of being concerned with programming details and having to manipulate operating system requirements. On the other hand, we faced many difficulties combining the Interface-AVL with the MOPSO algorithm. This was a technical problem related to the operating system. Even with University IT Department help, solving this problem took more than three months and the intervention of an external company, MathWorks. Therefore, future work could include enhancing the Nimrod/O optimisation tool by interfacing I-MOPSO and ANN-MOPSO with Nimrod/O to increase effectiveness when dealing with real world design problems.

2 The results of the investigation using interactive techniques and the ANN for aerodynamic shape optimisation at the conceptual and preliminary design phases of the Aegis UAV were encouraging, and it is hoped that future researchers will use multi-fidelity schemes within the same techniques to capture the flow physics that would enable more detailed designs. Thus, the duration of the search, which is time-consuming, could reduced – where required – by confining it to a narrower area.

3 The UAVs and commercial aircraft, in general, need to operate under multiple cruise conditions because of variability in flight missions, thus confining the optimiser to a single design point can lead to impairment of off-design performance. Thus, future work could focus on improving the ability of the framework to handle multipoint optimisation, to increase the performance of the optimised configurations. On the other hand, even though the formulation used to estimate the mass of the optimised configurations in the case of the Aegis UAV was efficient, it requires improvement in order to be used efficiently for various different aircraft that have spars, ribs, and stringers which support the lifting surfaces structures.

4 Future work could extend and improve the ANN-MOPSO technique, by providing a standard procedure to be followed when selecting the level of scepticism and the minimum size of the data set required by the ANN before starting classification of the particles as either valid or invalid.

References

[1] K. Crombecq, E. Laermans, T. Dhaene, Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling, Eur. J. Oper. Res. 214 (2011) 683–696. doi:10.1016/j.ejor.2011.05.032.

[2] D. Ninian, S. Dakka, Design, Development and Testing of Shape Shifting Wing Model, Aerospace. 4 (2017) 52. doi:10.3390/aerospace4040052.

[3] S. Goertz, C. Ilic, J. Jepsen, M. Leitner, M. Schulze, A. Schuster, J. Scherer, R. Becker, S. Zur, M. Petsch, Multi-Level MDO of a Long-Range Transport Aircraft Using a Distributed Analysis Framework, in: 18th AIAA/ISSMO Multidiscip. Anal. Optim. Conf. Doi10.2514/6.2017-4326., Denver, Colorado, USA, 2017: pp. 1–24. doi:10.2514/6.2017-4326.

[4] L. Leifsson, S. Koziel, A. Bekasiewicz, Fast low-fidelity wing aerodynamics model for surrogate-based shape optimization, Procedia Comput. Sci. 29 (2014) 811–820. doi:10.1016/j.procs.2014.05.073.

[5] A. Jameson, Re-Engineering the Design Process Through Computation, J. Aircr. 36 (1999) 36–50. 97-0641.

[6] J.B. RV Rao, DP Rai, A multi-objective algorithm for optimization of modern machining processes, Eng. Appl. Artif. Intell. Elsevier. 61 (2017) Pages 103-125.

[7] Z. Pourbahman, A. Hamzeh, Reducing the Computational Cost in Multi-objective Evolutionary Algorithms by Filtering Worthless Individuals, ArXiv Prepr. ArXiv1401.5808. (2014).

[8] W.H. Mason, D.L. Knill, A.A. Giunta, B. Grossman, L.T. Watson, W.H. Mason, D.L. Knill, A.A. Giunta, B. Grossman, L.T. Watson, Getting the Full Benefits of CFD in Conceptual Design, in: 16th AIAA Appl. Aerodyn. Conf., American Institute of Aeronautics and Astronautics, Albuquerque, New Mexico, 1998. AIAA 98-2513.

[9] G.E. Nicolai, Leland M.; Carichner, Fundamentals of Aircraft and Airship

Design: Volume 1, American Institute of Aeronautics and Astronautics, United States of America, 2010.

[10] D.P. Raymer, Enhancing Aircraft Conceptual Design Using Multidisciplinary Optimization, 2002.

[11] T. Rawlins, A. Lewis, J. Hettenhausen, T. Kipouros, Enhancing MOPSO through the guidance of ANNs, Proc. Int. Jt. Conf. Neural Networks. (2014) 4003–4010. doi:10.1109/IJCNN.2014.6889853.

[12] G.K. Kenway, A. Mishra, N.R. Secco, K. Duraisamy, J. Martins, An Efficient Parallel Overset Method for Aerodynamic Shape Optimization, 58th AIAA/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. (2017). doi:10.2514/6.2017-0357.

[13] S. Zapotecas Martinez, A. Arias Montano, C.A. Coello Coello, Constrained Multi-objective Aerodynamic Shape Optimization via Swarm Intelligence, in: Proc. 2014 Conf. Genet. Evol. Comput., Vancouver, BC, Canada; July 12–16, 2014, n.d.: pp. 81–88. doi:10.1145/2576768.2598372.

[14] R.J.L. Peter J Fleming, Robin C Purshouse, Many-objective optimization: An engineering design perspective, 2005.

[15] J. Hettenhausen, A. Lewis, M. Randall, T. Kipouros, Interactive Multi-Objective Particle Swarm Optimisation using Decision Space Interaction, IEEE Congr. Evol. Comput. (2013).

[16] K. Deb, A. Kumar, Interactive evolutionary multi-objective optimization and decision-making using reference direction method, in: Proc. 9th Annu. Conf. Genet. Evol. Comput. - GECCO ’07, London, England, United Kingdom, July 7–11,2007, n.d.: p. 781. doi:10.1145/1276958.1277116.

[17] J. Branke, S. Greco, R. Słowiński, P. Zielniewicz, Interactive evolutionary multiobjective optimization driven by robust ordinal regression, Bull. Polish Acad. Sci. Tech. Sci. 58 (2010) 347–358. doi:10.2478/v10175-010-0033-3.

[18] J. Hettenhausen, A. Lewis, S. Mostaghim, Interactive multi-objective

particle swarm optimization with heatmap-visualization-based user interface, Eng. Optim. 42 (2010) 119–139. doi:10.1080/03052150903042632.

[19] S.N. Skinner, H. Zare-Behtash, State-of-the-art in aerodynamic shape optimisation methods, Appl. Soft Comput. J. 62 (2018) 933–962. doi:10.1016/j.asoc.2017.09.030.

[20] K. Deb, A. Sinha, P.J. Korhonen, J. Wallenius, An interactive evolutionary multiobjective optimization method based on progressively approximated value functions, Evol. Comput. IEEE Trans. 14 (2010) 723–739.

[21] A.J. Nebro, A.B. Ruiz, C. Barba-González, J.M. García-Nieto, M. Luque, J.F. Aldana-Montes, InDM2: Interactive Dynamic Multi-Objective Decision Making using evolutionary algorithms, Swarm Evol. Comput. 40 (2018) 184–195. doi:10.1016/j.swevo.2018.02.004.

[22] M. Özmen, G. Karakaya, M. Köksalan, Interactive evolutionary approaches to multiobjective feature selection, Int. Trans. Oper. Res. 25 (2018) 1027–1052. doi:10.1111/itor.12428.

[23] A. Muhtar, I.W. Mustika, The Comparison of ANN-BP and ANN-PSO as Learning Algorithm to Track MPP in PV System, Eng. Semin. (InAES), 2017 7th Int. Annu. IEEE. (2017) 1–6.

[24] B. Sa, P.K., Sahoo, M.N., Murugappan, M., Wu, Y. and Majhi, Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, 2016.

[25] P. Piperni, A. DeBlois, R. Henderson, Development of a Multilevel Multidisciplinary-Optimization Capability for an Industrial Environment, AIAA J. 51 (2013) 2335–2352. doi:10.2514/1.J052180.

[26] P. Panagiotou, G. Ioannidis, I. Tzivinikos, K. Yakinthos, Experimental Investigation of the Wake and the Wingtip Vortices of a UAV Model, Aerospace. 4 (2017) 53. doi:10.3390/aerospace4040053.

[27] P. Panagiotou, P. Kaparos, K. Yakinthos, Winglet design and optimization for a MALE UAV using CFD, Aerosp. Sci. Technol. 39 (2014) 190–205. doi:10.1016/j.ast.2014.09.006.

[28] L. Leifsson and S. Koziel, Simulation-Driven Aerodynamic Design Using Variable-Fidelity Models, World Scientific, London, 2015.

[29] S. Chen, Z. Lyu, G.K.W. Kenway, J.R.R.A. Martins, Aerodynamic Shape Optimization of the Common Research Model Wing-Body-Tail Configuration, J. Aircr. (2015) 1–29. doi:10.2514/1.C033328.

[30] J. Gundlach, Designing Unmanned Aircraft Systems: A Comprehensive Approach, American Institute of Aeronautics and Astronautics, Reston, VA, United States of America, 2012.

[31] S. Rajagopal, R. Ganguli, Multidisciplinary Design Optimization of Long Endurance Unmanned Aerial Vehicle Wing, 1680 (2012) 1–34.

[32] K. Sastry, D. Goldberg, G. Kendall, Search Methodologies, 2005. doi:10.1007/978-1-60761-842-3_19.

[33] R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Struct. Multidiscip. Optim. 26 (2004) 369–395. doi:10.1007/s00158-003-0368-6.

[34] T. Jones, Mirrazavi, Multi-objective meta-heuristics : An overview of the current, Eur. Iournal Oper. Reaearch. 137 (2002) 1–9. S0377-2217(01)00123-0.

[35] N. Chase, M. Rademacher, E. Goodman, R. Averill, R. Sidhu, A Benchmark Study of Multi-Objective Optimization Methods Multi-objective Optimization Problem, Red Cedar Technol. East Lansing, MI, USA. (2009) 1–24.

[36] D.R. Tobergte, S. Curtis, A Multi-objective Tabu Search Algorithm for Constrained Optmisation Problems, J. Chem. Inf. Model. 53 (2013) 1689–1699. doi:10.1017/CBO9781107415324.004.

[37] A.K. Kundu, Aircraft Design, Cambridge University Press, 2010.

doi:10.1038/145368a0.

[38] J.D. Anderson, Fundamentals of AERODYNAMICS, Fifth Edit, McGraw-Hill, Americas, New York, 2011.

[39] T.C. Shafer, C.E. Lynch, N. Williams, Comparison of Computational Approaches for Rapid Aerodynamic Assessment of Small UAVs, (n.d.) 1–18.

[40] R. Austin, Unmanned Aircraft Systems: UAVS Design, Development and Deployment 2010, first, 2010 John Wiley & Sons Ltd, United Kingdom, 2010.

[41] J. Gundlach, Designing unmanned aircraft systems: A comprehensive approach, American Institute of Aeronautics and Astronautics, Inc, Manassas, Virginia, 2012.

[42] M. Sadraey, Aircraft Performance Analysis, VDM Verlag Dr. Müller, German, 2009.

[43] S. Gudmundsson, General aviation aircraft design: Applied Methods and Procedures, Oxford ; Boston : Butterworth-Heinemann, 2014.

[44] U. De Lisboa, Optimizing UAV Aerodynamics with Computational Fluid Dynamics, (2016).

[45] M.H. Sadraey, Aircraft Design: A Systems Engineering Approach, John Wiley & Sons, Chichester, West Sussex, United Kingdom, 2012.

[46] C.A. Mader, J.R.R.A. Martins, Computing Stability Derivatives and Their Gradients for Aerodynamic Shape Optimization, AIAA J. 52 (2014) 2533–2546. doi:10.2514/1.J052922.

[47] J.S. Arora, Computational design optimization: A review and future directions, Struct. Saf. 7 (1990) 131–148. doi:10.1016/0167-4730(90)90063-U.

[48] H. Ashley, On Making Things the I Best- Aeronautical Uses of Optimization, in: AIAA Aircr. Syst. Technol. Conf., American Institute of Aeronautics and

Astronautics, Dayton, Ohio, 1981. doi:10.2514/6.1981-1738.

[49] K.Y. Maalawi, M.A. Badr, Design Optimization of Mechanical Elements and Structures : a Review with Application, 5 (2009) 221–231.

[50] T.R. Kelley, Optimization , an Important Stage of Engineering Design Optimization, Res. Digit. 69 (2010) 18–23.

[51] J.N. and S.J. Wright, Numerical Optimization, Springer-Verlag New York, Inc, Verlag New York, 1999. 0-387-98793-2.

[52] G.S. Kirgat, A.N. Surde, Review of Hooke and Jeeves Direct Search Solution Method Analysis Applicable To Mechanical Design Engineering, Int. J. Innov. Eng. Res. Technol. 1 (2014) 1–14.

[53] P.S. Els, P.E. Uys, J.A. Snyman, M.J. Thoresson, Gradient-based approximation methods applied to the optimal design of vehicle suspension systems using computational models with severe inherent noise, Math. Comput. Model. 43 (2006) 787–801. doi:10.1016/j.mcm.2005.08.018.

[54] R. Salomon, Evolutionary Algorithms and Gradient Search : Similarities and Differences, 2 (1998) 45–55.

[55] R.M. Hicks, E.M. Murman, G.N. Vanderplaats, An assessment of airfoil design by numerical optimization, Washington, D.C. : National Aeronautics and Space Administration, NASA Ames Research Center Moffett Field, Calif. 94035, 1974. http://hdl.handle.net/2060/19740020369.

[56] D.S. Lee, L.F. Gonzalez, K. Srinivas, J. Periaux, Robust evolutionary algorithms for UAV/UCAV aerodynamic and RCS design optimisation, Comput. Fluids. 37 (2008) 547–564. doi:10.1016/j.compfluid.2007.07.008.

[57] B.J. Hromkovi, R.A. Salomaa, Theoretical Aspects of Local Search, Springer is a part of Springer Science+BusinessMedia springer.com, 2007.

[58] Z. Dingni, L. Yi, A. Mechanics, A. Mechanics, RLV Reentry Trajectory Optimization through Hybridization of an Improved GA and a SQP Algorithm, (2011) 1–11. doi:10.2514/6.2011-6658.

[59] L.F. González, K. Srinivas, J. Périaux, E.J. Whitney, Multidisciplinary Design Optimisation of Unmanned Aerial Vehicles (UAV) using Multi-Criteria Evolutionary Algorithms, (2005).

[60] A. Jahangirian, A. Shahrokhi, Aerodynamic shape optimization using efficient evolutionary algorithms and unstructured CFD solver, Comput. Fluids. 46 (2011) 270–276. doi:10.1016/j.compfluid.2011.02.010.

[61] J.-K. Shiau, D.-M. Ma, C.-W. Chiu, J.-R. Shie, Optimal Sizing and Cruise Speed Determination for a Solar-Powered Airplane, J. Aircr. 47 (2010) 622–629. doi:10.2514/1.45908.

[62] S. Rajagopal, R. Ganguli, A. Pillai, A. Lurdharaj, Conceptual Design of Medium Altitude Long Endurance UAV using Multi Objective Genetic Algorithm, 48th AIAA/ASME/ASCE/AHS/ASC. (2007) 1–13.

[63] A. Della Cioppa, Domenico Quagliarella, Genetic algorithms applied to the aerodynamic design of transonic airfoils, J. Aircr. Vol. 32 (1995) 889–891.

[64] H. Pirim, E. Bayraktar, B. Eksioglu, Tabu Search : A Comparative Study, (n.d.). https://www.semanticscholar.org/paper/1-Tabu-Search-%3A-A-Comparative-Study-Pirim-Bayraktar/9ef2e960ab56772747156fd79b841d8d2707f215 (accessed March 10, 2016).

[65] C.A. Coello Coello, G.B. Lamont, D. a Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer Science+Business Media, LLC All, New York, USA, 2007. doi:10.1007/978-0-387-36797-2.

[66] G.A.F. Alfarisy, A.N. Sihananto, T.N. Fatyanosa, M.S. Burhan, W.F. Mahmudy, Hybrid Genetic Algorithm and Simulated Annealing for Function Optimization, J. Inf. Technol. Comput. Sci. 1 (2016) 82–97. doi:10.25126/jitecs.20161215.

[67] B. Al-Khateeb, W.Z. Tareq, Solving 8-queens problem by using genetic algorithms, simulated annealing, and randomization method, Proc. - 2013

6th Int. Conf. Dev. ESystems Eng. DeSE 2013. (2013) 187–191. doi:10.1109/DeSE.2013.41.

[68] D.M. Jaeggi, G.T. Parks, T. Kipouros, P.J. Clarkson, The development of a multi-objective Tabu Search algorithm for continuous optimisation problems, Eur. J. Oper. Res. 185 (2008) 1192–1212. doi:10.1016/j.ejor.2006.06.048.

[69] A. Connor, J.P. Clarkson, S. Shaphar, P. Leonard, Engineering design optimisation using Tabu search, in: Proc. Des. Excell. Eng. Des. Conf. 2000, Brunel University, UK, 2000: pp. 371–378.

[70] T. Ghisu, G.T. Parks, D.M. Jaeggi, J.P. Jarrett, P.J. Clarkson, The benefits of adaptive parametrization in multi-objective Tabu Search optimization, Eng. Optim. 42 (2010) 959–981. doi:10.1080/03052150903564882.

[71] J. Eberhart, R., Kennedy, Particle swarm optimization, in: Proc. IEEE Int. Conf. Neural Networks, Perth, WA, Australia, Australia, 1995: pp. 1942–1948. http://link.springer.com/10.1007/s11721-007-0002-0.

[72] A. Ouarda, A Comparison of Evolutionary Algorithms : PSO , DE and GA for Fuzzy C-Partition, 91 (2014) 32–39. doi:10.5120/15919-5028.

[73] M. Iwan, R. Akmeliawati, T. Faisal, H.M.A.A. Al-Assadi, Performance comparison of differential evolution and particle swarm optimization in constrained optimization, Procedia Eng. 41 (2012) 1323–1328. doi:10.1016/j.proeng.2012.07.317.

[74] C.A. Coello Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proc. 2002 Congr. Evol. Comput. CEC 2002, Honolulu, HI, USA, 12–17 May 2002, n.d.: pp. 1051–1056. doi:10.1109/CEC.2002.1004388.

[75] C.A. Coello Coello, M. Reyes-Sierra, Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art, Int. J. Comput. Intell. Res. 2 (2006) 1–48. doi:10.5019/j.ijcir.2006.68.

[76] M.S.L. Carlos A. Coello Coello, Member, IEEE, Gregorio Toscano Pulido, Handling multiple objectives with particle swarm optimization, Evol. Comput. IEEE Trans. 8 (2004) 256–279. doi:10.1109/TEVC.2004.826067.

[77] M.J.W. Riley, T. Peachey, D. Abramson, K.W. Jenkins, Multi-objective engineering shape optimization using differential evolution interfaced to the Nimrod/O tool, in: IOP Conf. Ser. Mater. Sci. Eng., IOP Publishing Ltd, Sydney, Australia, 19–23 July, 2010, n.d. doi:10.1088/1757-899X/10/1/012189.

[78] M. Ehrgott, Vilfredo Pareto and Multi-objective Optimization, Doc. Math. I (2012) 447–453. http://math.uiuc.edu/documenta/vol-ismp/63_ehrgott-matthias.pdf.

[79] M. Yoshimi, T. Kuhara, K. Nishimoto, M. Miki, T. Hiroyasu, Visualization of Pareto Solutions by Spherical Self-Organizing Map and It's acceleration on a GPU, J. Softw. Eng. Appl. 05 (2012) 129–137. doi:10.4236/jsea.2012.53020.

[80] E. Dilettoso, S. Rizzo, N. Salerno, A Weakly Pareto Compliant Quality Indicator, Math. Comput. Appl. 22 (2017) 25. doi:10.3390/mca22010025.

[81] D.S. and X.Y. Ke Li, Renzhi Chen, Interactive Decomposition Multi-Objective Optimization via Progressively Learned Value Functions, Neural Evol. Comput. ArXiv1801.00609. (2018) 1–25.

[82] M. Gendreau, An Introduction to Tabu Search, Handb. Metaheuristics. 57 (2003) 37–54. doi:10.1007/0-306-48056-5.

[83] Y. Azabi, A. Savvaris, T. Kipouros, Initial Investigation of Aerodynamic Shape Design Optimisation for the Aegis UAV, Transp. Res. Procedia. 29 (2018) 12–22. doi:10.1016/j.trpro.2018.02.002.

[84] T. Kipouros, A. Inselberg, G. Parks, A.M. Savill, Parallel Coordinates in Computational Engineering Design, in: AIAA Multidiscip. Des. Optim. Spec., American Institute of Aeronautics and Astronautics, Inc., Boston,

Massachusetts;April 8-11, 2013, n.d.: pp. 1–11. doi:10.2514/6.2013-1750.

[85] J. Blaas, C.P. Botha, F.H. Post, Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets, IEEE Trans. Vis. Comput. Graph. 14 (2008) 1436–1443. doi:10.1109/TVCG.2008.131.

[86] S. Shahpar, A Review of Automatic Optimisation Applications in Aerodynamic Design of Turbomachinery Components, J. Turbomach. (2004) 1–12.

[87] T. Kipouros, D.M. Jaeggi, W.N. Dawes, G.T. Parks, Biobjective Design Optimization for Axial Compressors Using Tabu Search, 46 (2008). doi:10.2514/1.32794.

[88] R.C. Purshouse, K. Deb, M.M. Mansor, S. Mostaghim, R. Wang, A Review of Hybrid Evolutionary Multiple Criteria Decision Making Methods, IEEE Congr. Evol. Comput. (CEC), Beijing, China. (2014) 1147–1154. doi:10.1109/CEC.2014.6900368.

[89] A. Inselberg, Parallel coordinates: Visilization multidimensional geometry and its applications, Spring science, New York, New York, 2009.

[90] J. Heinrich and D. Weiskopf, Parallel Coordinates for Multidimensional Data Visualization:, (n.d.). http://joules.de/files/heinrich_parallel_2015.pdf (accessed March 25, 2016).

[91] Z. He, G.G. Yen, An improved visualization approach in many-objective optimization, 2016 IEEE Congr. Evol. Comput. CEC 2016. on 2016 Ju (2016) 1618–1625. doi:10.1109/CEC.2016.7743982.

[92] M. Novotn, Outlier-preserving Focus + Context Visualization in Parallel Coordinates, IEEE Trans. Vis. Comput. Graph. 12 (2006) 893–900.

[93] M. Optimization, Z. He, G.G. Yen, Visualization and Performance Metric in, 20 (2016) 386–402.

[94] P.C. Wong, R.D. Bergeron, 30 Years of Multidimensional Multivariate Visualization, Sci. Vis. Overviews Methodol. Tech. (1997) 3–33.

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.4639.

[95] K.J. Siirtola, H. and Räihä, Interacting with parallel coordinates, Interact. Comput. 18 (2006) 1278–1309.

[96] A. Inselberg, The plane with parallel coordinates, Vis. Comput. 1 (1985) 69–91.

[97] J. Heinrich, D. Weiskopf, State of the Art of Parallel Coordinates, Eurographics Conf. Vis. (2013) 95–116. doi:10.2312/conf/EG2013/stars/095-116.

[98] T. Kipouros, M. Mleczko, M. Savill, Use of parallel coordinates for post-analyses of multi-objective aerodynamic design optimisation in turbomachinery, AIAA Pap. 2138 (2008) 2138. doi:doi:10.2514/6.2008-2138.

[99] T. Kipouros, A. Inselberg, G. Parks, A.M. Savill, Parallel Coordinates in Computational Engineering Design, 54th AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. (2013). doi:10.2514/6.2013-1750.

[100] J. Hettenhausen, A. Lewis, T. Kipouros, A web-based system for visualisation-driven interactive multi-objective optimisation, Procedia Comput. Sci. 29 (2014) 1915–1925. doi:10.1016/j.procs.2014.05.176.

[101] A. Lewis, T. Kipouros, Methods of Interactive Optimisation in Engineering Design, 13th Int. Conf. Comput. Appl. (2015). doi:10.13140/2.1.1706.8320.

[102] K. Miettinen, Introduction to multiobjective optimization: Noninteractive approaches, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 5252 LNCS (2008) 1–26. doi:10.1007/978-3-540-88908-3-1.

[103] C. Gaiddon, A., Knight, D.D. and Poloni, Multicriteria Design Optimization of a Supersonic Inlet Based upon Global Missile Performance, J. Propuls. Power. 20 (2004) 542–558.

[104] L. Li, I. Yevseyeva, V. Basto-Fernandes, H. Trautmann, N. Jing, M.

Emmerich, An Ontology of Preference-Based Multiobjective Metaheuristics, (2016). http://arxiv.org/abs/1609.08082.

[105] R. Baxter, N. Hastings, A. Law, E.J.. Glass, Multi-objective Evolutionary Optimisation for Product Design and Manufacturing, Springer London Dordrecht Heidelberg New York, 2008.

[106] D. Abramson, R. Sosic, J. Giddy, B. Hall, Nimrod: A Tool for Performing Parameterised Simulations Using Distributed Workstations, Hpdc. (1995) 112–121.

[107] A. Clarich, R. Russo, M. Carriglio, Multi-Objective Optimization With Modefrontier Interfaces for Ansa and Metapost, 4 Th ANSA MicroETA Int. Conf. (2008).

[108] S. Padula, J. Korte, H. Dunn, A. Salas, Multidisciplinary Optimization Branch Experience Using iSIGHT Software, 1999 Int. ISIGHT Users' Conf. Chapel Hill, North Carolina Oct. 4-6,1999. (1999).

[109] D. Abramson, A. Lewis, T.O.M. Peachey, Nimrod / O : A Tool for Automatic Design Optimization Using Parallel and Distributed Systems, (2000). doi:10.1142/9789812792037.

[110] G. Lusignani, https://www.cranfield.ac.uk/press/news-2017/0816-supercomputer powers up at cranfield university. (accessed on 27 Sep 2018), (n.d.).

[111] D. Abramson, T. Peachey, A. Lewis, Model Optimization and Parameter Estimation with Nimrod / O, in: Proc. 6th Int. Conf. Comput. Sci., Reading, UK, 28–31 May 2006, n.d.: pp. 720–727. doi:10.1007/11758501_96.

[112] N. Srinivas, K. Deb, Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms Evolutionary Computations, Clim. Chang. 2013 - Phys. Sci. Basis. 2 (1994) 1–30. doi:10.1017/CBO9781107415324.004.

[113] D. Abramson, A. Lewis, T. Peachey, C. Fletcher, An Automatic Design

Optimization Tool and its Application to Computational Fluid Dynamics Searching for Optimal Designs, Proc. 2001 ACM/IEEE Conf. Supercomput. (2001).

[114] T. Peachey, D. Abramson, A. Lewis, D. Kurniawan, R. Jones, Optimization using Nimrod / O and its Application to Robust Mechanical Design, Stress Int. J. Biol. Stress. (2004) 1–8.

[115] D. Abramson, B. Bethwaite, C. Enticott, S. Garic, T. Peachey, Parameter space exploration using scientific workflows, in: Proc. Int. Conf. Comput. Sci., Springer, Baton Rouge, LA, USA, 2009: pp. 104–113.

[116] M.P.E. Pratiksha Saxena, Dipti Singh, problem solving and uncertainty modeling through optimization and soft computing application, Information Science Reference, Idea Group,U.S., Harrisburg, PA, United States, 2016.

[117] G.N. Vanderplaats, C. Springs, Design Optimisation a Powerful Tool For the Competitive Edge, 1 St AIAA Aircraft, Technol. Integr. Oper. Los Angeles, CA, USA, 16–18 Oct. (2001) 8. doi:10.2514/6.2001-5214.

[118] M. Zhang, A. Jungo, A. Gastaldi, T. Melin, Aircraft Geometry and Meshing with Common Language Schema CPACS for Variable-Fidelity MDO Applications, Aerospace. 5 (2018) 47. doi:10.3390/aerospace5020047.

[119] T. Chau, D.W. Zingg, Aerodynamic shape optimization of a box-wing regional aircraft based on the reynolds-averaged Navier-Stokes equations, in: 35th AIAA Appl. Aerodyn. Conf., American Institute of Aeronautics and Astronautics, Denver, CO, USA, 5–9 June, 2017: pp. 1–29. doi:10.2514/6.2017-3258.

[120] U. Iemma, M. Diez, Optimal Conceptual Design of Aircraft Including Community Noise Prediction, in: 12th AIAA/CEAS Aeroacoustics Conf., Cambridge, Massachusetts, 2006: pp. 8–10. doi:10.2514/6.2006-2621.

[121] R.A. Reuter, S. Iden, R.D. Snyder, D.L. Allison, An Overview of the Optimized Integrated Multidisciplinary Systems Program, in: 57th AIAA/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf., American Institute

of Aeronautics and Astronautics, San Diego, California, USA;4-8 January 2016, n.d.: pp. 1–11. doi:10.2514/6.2016-0674.

[122] Z. Lyu, G.K.W. Kenway, J.R.R.A. Martins, Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark, AIAA J. 53 (2015) 968–985. doi:10.2514/1.J053318.

[123] D. Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J, and Saunders, Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1, J. Aircr. 36 (1999) 51–60. doi:10.2514/2.2413.

[124] J.E. Hicken, D.W. Zingg, Induced-Drag Minimization of Nonplanar Geometries Based on the Euler Equations, AIAA J. 48 (2010) 2564–2575. doi:10.2514/1.J050379.

[125] Z. Lyu, J.R.R.A. Martins, Aerodynamic Design Optimization Studies of a Blended-Wing-Body Aircraft, J. Aircr. 51 (2014) 1604–1617. doi:10.2514/1.C032491.

[126] Q. Deng, S. Shao, L. Fu, H. Luan, Z. Feng, An Integrated Design and Optimization Approach for Radial Inflow Turbines—Part II: Multidisciplinary Optimization Design, Appl. Sci. 8 (2018) 2030. doi:10.3390/app8112030.

[127] M.K. Zahir, Z. Gao, Variable-fidelity optimization with design space reduction, Chinese J. Aeronaut. 26 (2013) 841–849. doi:10.1016/j.cja.2013.06.002.

[128] T. Kipouros, D.M. Jaeggi, W.N. Dawes, G.T. Parks, A.M. Savill, P.J. Clarkson, Insight into high-quality aerodynamic design spaces through multi-objective optimization, C. - Comput. Model. Eng. Sci. 37 (2008) 1–44.

[129] J. Demange, A.M. Savill, T. Kipouros, Multifidelity Optimization for High-Lift Airfoils, 54th AIAA Aerosp. Sci. Meet. (2016) 1–13.

[130] D. Abramson, A.L. T., Peachey, C. Flecher, An Automatic Design Optimization Tool and its Application to Computational Fluid Dynamics,

Conf. High Perform. Netw. Comput. Proc. 2001 ACM/IEEE Conf. Supercomput. 10 (2001).

[131] T. Kipouros, Human-in-the-loop Computational Engineering Design, Evolve. (2014) 2–4. doi:10.13140/2.1.4197.2004.

[132] Marjan van Herwijnen., Available online: http://www.ivm.vu.nl/en/Images/MCA1_tcm234-161527.pdf (accessed on 23 Febr 2018), (n.d.).

[133] S. (Pamuk) Phelps, M. Koksalan, An Interactive Evolutionary Metaheuristic for Multiobjective Combinatorial Optimization, Manage. Sci. 49 (2003) 1726–1738. doi:10.1287/mnsc.49.12.1726.25117.

[134] J.W. Fowler, E.S. Gel, M.M. Köksalan, P. Korhonen, J.L. Marquis, J. Wallenius, Interactive evolutionary multi-objective optimization for quasi-concave preference functions, Eur. J. Oper. Res. 206 (2010) 417–425. doi:10.1016/j.ejor.2010.02.027.

[135] S. Greco, B. Matarazzo, R. Slowinski, Rough sets theory for multicriteria decision analysis, Eur. J. Oper. Res. 129 (2001) 1–47. doi:10.1016/S0377-2217(00)00167-3.

[136] T. Kipouros, T. Peachey, D. Abramson, A.M. Savill, Enhancing and Developing the Practical Optimization Capabilities and Intelligence of Automatic Design Software, in: 53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf., Honolulu, HI, USA, 23–26 April 2012, n.d.: pp. 1–7. doi:10.2514/6.2012-1677.

[137] H. Drela, M. and Youngren, AVL 3.26 user primer, (n.d.). http://web.mit.edu/drela/Public/web/avl/ (accessed November 25, 2015).

[138] S.K. Jyoti, K.D. Manjusha, K.D. Pavan, Review on Computing Machinery and Intelligence, Int. J. Curr. Microbiol. Appl. Sci. (2018) 442–451.

[139] S.B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, Informatica. 31 (2007) 249–268. doi:10.1115/1.1559160.

[140] C. Mair, G. Kadoda, M. Le, K. Phalp, C. Scho, M. Shepperd, S. Webster, An investigation of machine learning based prediction systems, J. Syst. Softw. 53 (2000) 23–29. doi:10.1016/S0164-1212(00)00005-4.

[141] M. Jordan, J. Kleinberg, B. Scho, Pattern Recognition and Machine Learning, Microsoft, 2006 Springer Science+Business Media, LLC, New York, NY 10013, USA, 2006.

[142] F. Chen, H. Li, Z. Xu, S. Hou, D. Yang, User-friendly optimization approach of fed-batch fermentation conditions for the production of iturin A using artificial neural networks and support vector machine, Electron. J. Biotechnol. 18 (2015) 273–280. doi:10.1016/j.ejbt.2015.05.001.

[143] A. Mekky, T.E. Alberts, Design of a Stochastic Basis Function Artificial Neural Network Controller for Quadrotors Flight in the Presence of Model and Aerodynamic Uncertainties, NAECON 2018 - IEEE Natl. Aerosp. Electron. Conf. (2018) 395–402. doi:10.1109/NAECON.2018.8556745.

[144] K.C. Giannakoglou, D.I. Papadimitriou, I.C. Kampolis, Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels, Comput. Methods Appl. Mech. Eng. 195 (2006) 6312–6329. doi:10.1016/j.cma.2005.12.008.

[145] H. Peng, X. Ling, Optimal design approach for the plate-fin heat exchangers using neural networks cooperated with genetic algorithms, Appl. Therm. Eng. 28 (2008) 642–650. doi:10.1016/j.applthermaleng.2007.03.032.

[146] E. Amirian, Z. John Chen, Cognitive Data-Driven Proxy Modeling for Performance Forecasting of Waterflooding Process, Glob. J. Technol. Optim. 08 (2017) 1–9. doi:10.4172/2229-8711.1000207.

[147] R.J. Kuo, S.C. Chi, S.S. Kao, A decision support system for selecting convenience store location through integration of fuzzy AHP and artificial neural network, Comput. Ind. 47 (2002) 199–214. doi:10.1016/S0166-3615(01)00147-6.

[148] M. Cilla, E. Borgiani, J. Martínez, G.N. Duda, S. Checa, Machine learning techniques for the optimization of joint replacements: Application to a short-stem hip implant, PLoS One. 12 (2017) 1–16. doi:10.1371/journal.pone.0183755.

[149] A. Jovic, K. Brkic, N. Bogunovic, An overview of free software tools for general data mining, Inf. Commun. Technol. Electron. Microelectron. (MIPRO), 2014 37th Int. Conv. (2014) 1112–1117. doi:10.1109/MIPRO.2014.6859735.

[150] S.C.A. Dr. Singh Yashpal, Neural Networks In Data Mining, J. Theor. Appl. Inf. Technol. 5 (2009) 1–154.

[151] M. Soundarya, R. Balakrishnan, Survey on Classification Techniques in Data mining, Int. J. Adv. Res. Comput. Commun. Eng. 3 (2014) 7550–7552.

[152] C.E. Volume, A. Electrical, A. Engineering, Redesign of Morphing UAV for Simultaneous Improvement of Directional Stability and Maximum Lift / Drag Ratio, 18 (2018) 57–62.

[153] S. Kalogirou, Applications of artificial neural networks in energy systems, Energy Convers. Manag. 40 (1999) 1073–1087. doi:10.1016/S0196-8904(99)00012-6.

[154] S. Mohaghegi, Y. Del Valle, G.K. Venayagamoorthy, R.G. Harley, A comparison of PSO and backpropagation for training RBF neural networks for identification of a power system with statcom, Proc. - 2005 IEEE Swarm Intell. Symp. SIS 2005. 2005 (2005) 391–394. doi:10.1109/SIS.2005.1501646.

[155] G. Bin Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: Theory and applications, Neurocomputing. 70 (2006) 489–501. doi:10.1016/j.neucom.2005.12.126.

[156] D.F. Specht, A general regression neural network, Neural Networks, IEEE Trans. 2 (1991) 568–576. doi:10.1109/72.97934.

[157] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, B.W. Suter, The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function, IEEE Trans. Neural Networks. 1 (1990) 296–298. doi:10.1109/72.80266.

[158] M. W. GARDNER and S. R. DORLING, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmos. Environ. 32 (1998) 2627–2636. doi:10.1016/S1352-2310(97)00447-0.

[159] Z. Liu, H. Li, X. Tang, X. Zhang, F. Lin, K. Cheng, Extreme learning machine: a new alternative for measuring heat collection rate and heat loss coefficient of water-in-glass evacuated tube solar water heaters, Springerplus. 5 (2016) 1–8. doi:10.1186/s40064-016-2242-1.

[160] D.J. Sarma, S.C. Sarma, Neural Networks and their Applications in Industry, DESlDOC Bull. Inf. Technol. 20 (2000) 29–36.

[161] H. Li, Z. Zhang, Z. Liu, Application of Artificial Neural Networks for Catalysis: A Review, Catalysts. 7 (2017) 306. doi:10.3390/catal7100306.

[162] G. La Rocca, M.J. L. Van Tooren, Knowledge-Based Engineering Approach to Support Aircraft Multidisciplinary Design and Optimization, J. Aircr. 46 (2009) 1875–1885. doi:10.2514/1.39028.

[163] M.M. Rai, M. Field, Robust Optimal Aerodynamic Design Using Evolutionary Methods and Neural Networks, 42nd AIAA Aerosp. Sci. Meet. Exhib. (2004) 1–27.

[164] A. Askarzadeh, A. Rezazadeh, Artificial neural network training using a new efficient optimization algorithm, Appl. Soft Comput. J. 13 (2013) 1206–1213. doi:10.1016/j.asoc.2012.10.023.

[165] A. Boutemedjet, M. Samardžić, L. Rebhi, Z. Rajić, T. Mouada, UAV aerodynamic design involving genetic algorithm and artificial neural network for wing preliminary computation, Aerosp. Sci. Technol. 84 (2019) 464–483. doi:10.1016/j.ast.2018.09.043.

[166] M.M. Rai, Three-Dimensional Aerodynamic Design Using Artificial Neural Networks, Sci. York. (2002).

[167] N.K.M. Man Mohan Rai, Aerodynamic design using Neural Networks, AIAA J. 38 (2000) 173–182.

[168] R. Mukesh, K. Lingadurai, U. Selvakumar, Airfoil Shape Optimization based on Surrogate Model, J. Inst. Eng. Ser. C. 99 (2017) 1–8. doi:10.1007/s40032-017-0382-x.

[169] F. Duchaine, L.Y.M. Gicquel, D. Bissières, C. Bérat, T. Poinsot, Automatic Design Optimization Applied to Lean Premixed Combustor Cooling., (n.d.). https://scholar.google.co.uk/scholar?hl=en&as_sdt=0%2C5&q=Automatic+Design+Optimization+Applied+to+Lean+Premixed+Combustor+Cooling&btnG= (accessed June 15, 2018).

[170] I. Nunes da Silva, D. Hernane Spatti, R. Andrade Flauzino, L.H. Bartocci Liboni, S.F. dos Reis Alves, Artificial Neural Networks, 2017. doi:10.1007/978-3-319-43162-8.

[171] M. Ebrahimi, A. Jahangirian, Accelerating global optimization of aerodynamic shapes using a new surrogate-assisted parallel genetic algorithm, Eng. Optim. 49 (2017) 2079–2094. doi:10.1080/0305215X.2017.1289741.

[172] W. Annicchiarico, Metamodel-assisted distributed genetic algorithms applied to structural shape optimization problems, Eng. Optim. 39 (2007) 757–772. doi:10.1080/03052150701514980.

[173] Y. Lian, A. Oyama, M.S. Liou, Progress in design optimization using evolutionary algorithms for aerodynamic problems, Prog. Aerosp. Sci. 46 (2010) 199–223. doi:10.1016/j.paerosci.2009.08.003.

[174] H.O. Verma, N.K. Peyada, S. Singh, Aerodynamic Modelling of Quasi Steady Stall Using Neural- Network Based Gauss Newton Method, (2017).

[175] E. Magrini, A and Benini, Aerodynamic Optimization of a Morphing Leading

Edge Airfoil with a Constant Arc Length Parameterization, J. Aerosp. Eng. 31 (2018). 04017093.

[176] S.N.S.M.H.S. Qasem, RADIAL BASIS FUNCTION NETWORK BASED ON MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION, Proceeding Ofthe 6th Int. Symp. Mechatronics Its Appl. (ISMA09), Sharjah, UAE, March. (2009) 1–6.

[177] M. Yaghini, M.M. Khoshraftar, M. Fallahi, A hybrid algorithm for artificial neural network training, Eng. Appl. Artif. Intell. 26 (2013) 293–301. doi:10.1016/j.engappai.2012.01.023.

[178] M.S. Khurana, H. Winarto, A.K. Sinha, Airfoil Optimisation by Swarm Algorithm With Mutation and Artificial Neural Networks, 47th AIAA Aerosp. Sci. Meet. Incl. New Horizons Forum Aerosp. Expo. (2009) 1–19. doi:10.2514/6.2009-1278.

[179] J. Hadjiev, H. Panayotov, Comparative Investigation of VLM Codes For Joined-Wing Analysis, Int. J. Res. Eng. Technol. 02 (2013) 478–482.

[180] C. Beaverstock, B. Woods, J. Fincham, M. Friswell, Performance Comparison between Optimised Camber and Span for a Morphing Wing, Aerospace. 2 (2015) 524–554. doi:10.3390/aerospace2030524.

[181] D. schoaefer J. Hoefling, J. Schirra, A. Spohr, Induced Drag Computation With Wake Vortex model Schemes For Highly Non-Planar Wing systems, Detscher Luft-Und Raumfahrtkongress. 0 (2013) 1–10.

[182] T. Melin, A.T. Isikveren, M.I. Friswell, Induced-Drag Compressibility Correction for Three-Dimensional Vortex-Lattice Methods, J. Aircr. 47 (2010) 1458–1460. doi:10.2514/1.C000197.

[183] T. Kipouros, Nimrod / O Users ’ Guide, 2012.

[184] A.M. Connor, K.A. Seffen, G.T. Parks, P.J. Clarkson, Efficient Optimisation of Structures Using Tabu Search, Inis.Iaea.Org. (1993).

[185] G. Tilocca, Interactive Optimisation for Aircraft Application, Cranfield

University, Msc Thesis, 2016.

[186] J. Demšar, T. Curk, A. Erjavec, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Stajdohar, L. Umek, L. Zagar, J. Zbontar, M. Zitnik, B. Zupan, Orange: Data Mining Toolbox in Python, J. Mach. Learn. Res. 14 (2013) 23492353.

[187] J. Demsar, B. Zupan, Orange: data mining fruitful and fun, Informatica. 37 (2013) 55.

[188] M. Stajdohar, J. Demsar, Interactive Network Exploration with **Orange**, J. Stat. Softw. 53 (2013). doi:10.18637/jss.v053.i06.

[189] J. Ngiam, A. Coates, On Optimization Methods for Deep Learning, … Mach. Learn. …. (2011) 1–3. doi:10.1.1.220.8705.

[190] S. Keast, Modeling, Simulation, and Sil Testing of the Aegis UAV, MSc, Cranfield University, 2015. doi:10.1017/CBO9781107415324.004.

[191] C. Turquand, Aerodynamic Analysis and Optimisation of the Aegis TUAV, MSc, Cranfield University, 2011.

[192] A. Dunne, Aerodynamic Optimisation of the Aegis UAV, Cranfield Univesity, 2012.

[193] J. Lee, General Aviation Aircraft Design, AIAA J. 54(2) (2015) pp.793-794.

[194] G.K.W. Kenway, J.R.R.A. Martins, Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration, J. Aircr. 51 (2014) 144–160. doi:10.2514/1.C032150.

[195] S. KEAST, MODELLING, SIMULATION, AND SIL TESTING OF THE AEGIS UAV, Cranfield Univesity, 2015.

[196] T.J. Mueller, J.D. DeLaurier, Aerodynamic of Small Vehicles, Annu. Rev. Fluid Mech. 35 (2003) 89–111. doi:10.1146/annurev.fluid.35.101101.161102.

[197] K. Reynolds, N. Nguyen, Design optimization of light weight tail geometry

using gradient-based and GA search methods, AIAA Infotech Aerosp. Conf. Exhib. 2012. (2012) 1–17.

[198] J. PAUL, E.P. CAMPBK, Experimental Verification of a Simplified Vee-Tail Theory and Analysis of Avaliable Data on Complete Models with Vee Tails, n.d.

[199] M.H. Sadraey, Aircraft Design: Asystem Engineering Approach, John Wiley & Sons, 2012.

[200] K. Deb, Multi-objective optimization using evolutionary algorithms., 16 (2001) 1–24.

[201] C. Holden, A. Keane, Visualization Methodologies in Aircraft Design, in: 10th AIAA/ISSMO Multidiscip. Anal. Optim. Conf., American Institute of Aeronautics and Astronautics, Inc., Albany,NY, USA; 30 August - 1 September 2004, n.d.: pp. 1–13. doi:10.2514/6.2004-4449.

[202] C.D. Mhalungekar, EXPERIMENTAL and ANALYTICAL ANALYSIS of FLOW PAST D-SHAPED CYLINDER, 1 (2014) 218–223.

[203] D. Raymer, Aircraft Design : A Conceptual Approach, Second Edi, American Institute of Aeronautics & Astronautics, Inc, Sylmar, California, 1999.

[204] M.J. Hoffmann, R. Reuss Ramsay, G.M. Gregorek, Effects of grit roughness and pitch oscillations on the NACA 4415 airfoil, National Renewable Energy Lab., Golden, CO (United States); Ohio State Univ., Columbus, 1996. doi:10.2172/266691.

[205] R.W.B. C. Ferris, Robert J. McGhee, Low-Speed Wind-Tunnel Results for Symmetrical NASA LS(1)-003 Airfoil, Virginia, 1987.

[206] J.L. Petrilli, R.C. Paul, A. Gopalarathnam, N.T. Frink, A CFD Database for Airfoils and Wings at Post-Stall Angles of Attack, 31st AIAA Appl. Aerodyn. Conf. (2013). doi:10.2514/6.2013-2916.

[207] K. Deb, S. Pratab, S. Agarwal, T. Meyarivan, A Fast and Elitist

Multiobjective Genetic Algorithm: NGSA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197. doi:10.1109/4235.996017.

Appendix A – Drag Model

A.1 Parasite drag

Since the AVL code is capable of evaluating only inviscid drag, empirical formulas, commonly called “build-up technique” are used to evaluate zero lift drag. The majority of the equations used in this technique are based on data gathered from flight tests and wind tunnel experiments [42]. Zero lift drag is also referred to as parasite drag, and includes drag that does not depend on the production of lift. Parasite drag for low-speed aircraft (low subsonic, incompressible regime) may be divided into two types; skin friction drag and form drag [26,202]. For the optimisation process, it is essential to provide a quick method for calculating the UAV drag since optimisation will require repetitive aerodynamic calculations of the newly optimised configurations.

To calculate UAV zero-lift drag coefficients, every contributing component must be included. The total zero-lift drag (CD_0) is defined as:

$$\mathrm{CD_0} = \mathrm{CD_{0,w}} + \mathrm{CD_{0,f}} + \mathrm{CD_{0,t}} + \mathrm{CD_{0,b}} \quad \text{(A-1)}$$

Where $\mathrm{CD_{0,w}}$, $\mathrm{CD_{0,f}}$, $\mathrm{CD_{0,t}}$, $\mathrm{CD_{0,b}}$ respectively represent the zero-lift drag coefficients for wing, fuselage, tail, and boom.

In the case of the U-tail arrangement, $\mathrm{CD_{0,t}}$ is defined as:

$$\mathrm{CD_{0,t}} = \mathrm{CD_{0,ht}} + \mathrm{CD_{0,vt}} \quad \text{(A-2)}$$

Where $\mathrm{CD_{0,ht}}$ and $\mathrm{CD_{0,vt}}$ are the zero lift drag coefficients for the horizontal and vertical tail, respectively.

In the case of the Inverted V-tail, the zero lift drag ($\mathrm{CD_{0,t}}$) is calculated as:

$$\mathrm{CD_{0,t}} = \mathrm{CD_{0,vee}} \quad \text{(A-3)}$$

Where the $\mathrm{CD_{0,vee}}$ is the zero lift drag for the V-tail of the Aegis UAV.

A.1.1 Fuselage and boom

The zero-lift drag coefficients for fuselage and boom are defined as:

$$\mathrm{CD_{0,f}} = \mathrm{C_{f,f}} * \mathrm{f_{LD,f}} * \mathrm{f_M} * \frac{\mathrm{S_{wet,f}}}{\mathrm{S_{ref}}} \qquad \text{(A-4)}$$

$$\mathrm{CD_{0,b}} = \mathrm{C_{f,b}} * \mathrm{f_{LD,b}} * \mathrm{f_M} * \frac{\mathrm{S_{wet,b}}}{\mathrm{S_{ref}}} \qquad \text{(A-5)}$$

Where, $\mathrm{C_{f,f}}$ and $\mathrm{C_{f,b}}$ are skin friction coefficients that characterize the flow by Reynolds number, either laminar or turbulent. They are calculated as:

$$\mathrm{C_{f,f}} = \mathrm{C_{f,b}} = \frac{0.455}{[\log_{10} * \mathrm{Re}]^{2.58}}\ , \qquad \text{for turbulent} \qquad \text{(A-6)}$$

$$\mathrm{C_{f,f}} = \mathrm{C_{f,b}} = \frac{1.327}{\sqrt{\mathrm{Re}}}\ , \qquad \text{for laminar} \qquad \text{(A-7)}$$

The Reynolds number Re is defined as:

$$\mathrm{Re} = \frac{\rho * \mathrm{v} * \mathrm{L}}{\mu} \qquad \text{(A-8)}$$

Where ρ is air density, v is flight speed, and μ is dynamic viscosity of air for the flight regime. For the boom and fuselage, L represents the length.

Normally, the air flow over the surfaces of airborne aircraft starts as laminar and then changes to turbulent, though it is usually the case that an aircraft will pass through a region of combined laminar and turbulent flow. However, at the conceptual and preliminary design phases it is permissible to assume for simplicity that the flow is either completely laminar or turbulent [203].

The terms $\mathrm{f_{LD,f}}$ and $\mathrm{f_{LD,b}}$ in Equations (A4) and (A-5) are a function of the length to diameter ratio of the fuselage and boom

$$\mathrm{f_{LD}} = \mathrm{f_{LD,f}} = \mathrm{f_{LD,b}} = 1 + \frac{60}{(\mathrm{L/D})^3} + 0.0025 * (\mathrm{L/D}) \qquad \text{(A-9)}$$

Where L and D represent the maximum length and diameter of the fuselage or boom respectively. The $\mathrm{f_M}$ term in Equations (A-4) and (A-5) is a function of the UAV Mach number and defined as:

$$\mathrm{f_M} = 1 - 0.08\mathrm{M}^{1.45} \qquad \text{(A-10)}$$

The last two terms in Equations (A-4) and (A-5) are $\mathrm{S_{wet}}$ and $\mathrm{S_{ref}}$. The $\mathrm{S_{wet}}$ is the surface area of the fuselage or boom that is in real contact with the flow during flight, while S_{ref} is the reference area. The reference area used in these calculations is wing surface area.

A.1.2 Wing and tail

The wing and tail are considered as lifting surfaces. Each tail surface (horizontal and vertical) should be treated separately as shown in Equations (A-2) and (A-3). The zero lift drag coefficient of the wing, $CD_{0,w}$, and of the tail, $CD_{0,t}$, are given by the equations below:

$$CD_{0,w} = C_{f,w} * f_{tc,w} * f_M * \left(S_{wet,w}/S_{ref}\right) * \left(Cd_{min,w}/0.004\right)^{0.4} \qquad \text{(A-11)}$$

$$CD_{0,t} = C_{f,t} * f_{tc,t} * f_M * \left(S_{wet,t}/S_{ref}\right) * \left(Cd_{min,t}/0.004\right)^{0.4} \qquad \text{(A-12)}$$

The terms $C_{f,w}$ and $C_{f,t}$ are the same as in Equation (A-7). The second terms in the above two equations is defined as:

$$f_{tc,w} = f_{tc,t} = 1 + 2.7 * (t/c)_{max} + 100(t/c)^4_{max} \qquad \text{(A-13)}$$

Where $(t/c)_{max}$ is the maximum thickness to chord ratio of the lifting surface. For the Aegis UAV, it is 15% for the wing and 13% for the horizontal, vertical, and Inverted V-tail.

The terms $S_{wet,w}$ and $S_{wet,t}$ represent the wetted area for the wing and tail respectively, and can be calculated in various ways. However, for thin surfaces ($(t/c)_{max}$ around 15%) the following equation can be used with high accuracy:

$$S_{wet} = 2 * [1 + 0.5 * (t/c)_{max}] * b_w * C_w \qquad \text{(A-14)}$$

Where b_w and C_w are the span, and chord, respectively, of the lifting surface. The terms $Cd_{min,w}$ and $Cd_{min,t}$ represent the minimum drag coefficients for the wing and tail airfoil cross-sections, respectively. At Re values of about 1.5 x 10^6, the minimum drag coefficient for the NACA4415 airfoil is around 0.008 [204] (see Figure A-1), and the minimum drag coefficient for the NACA0013 airfoil is around 0.006 at Re ≈ 0.7 x 10^6 [205] (see Figure A-2).

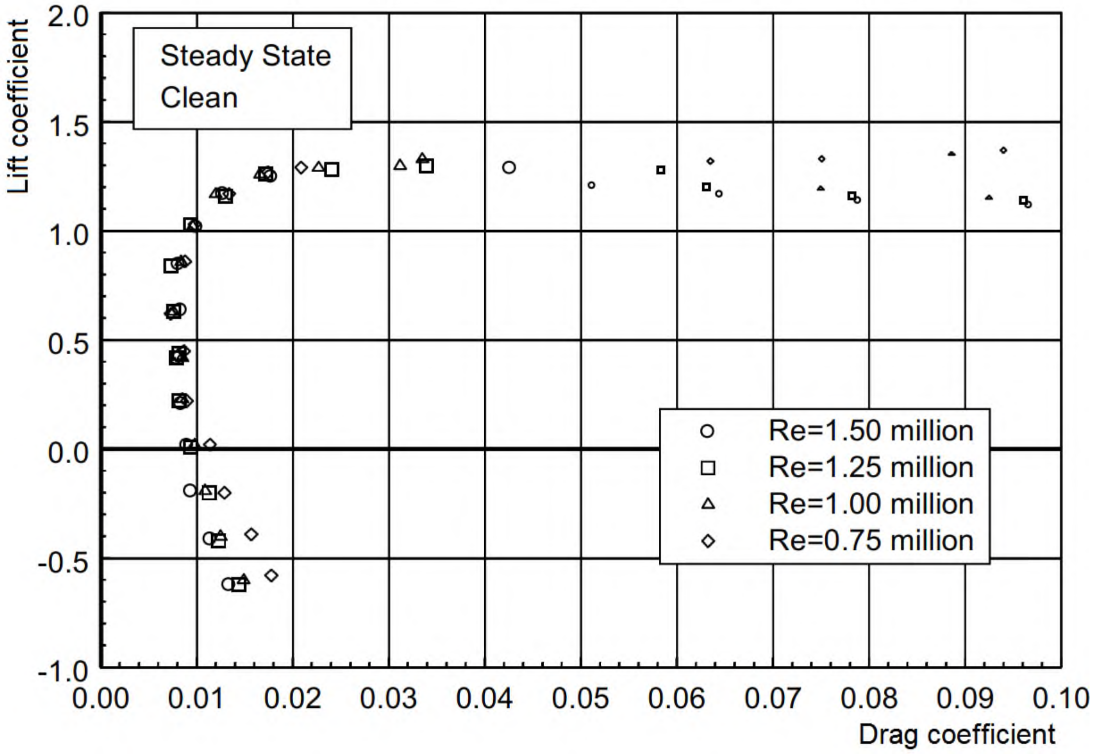


Figure A-1: Lift coefficient versus drag coefficient for NACA 4415 at various values of Reynold number [204]

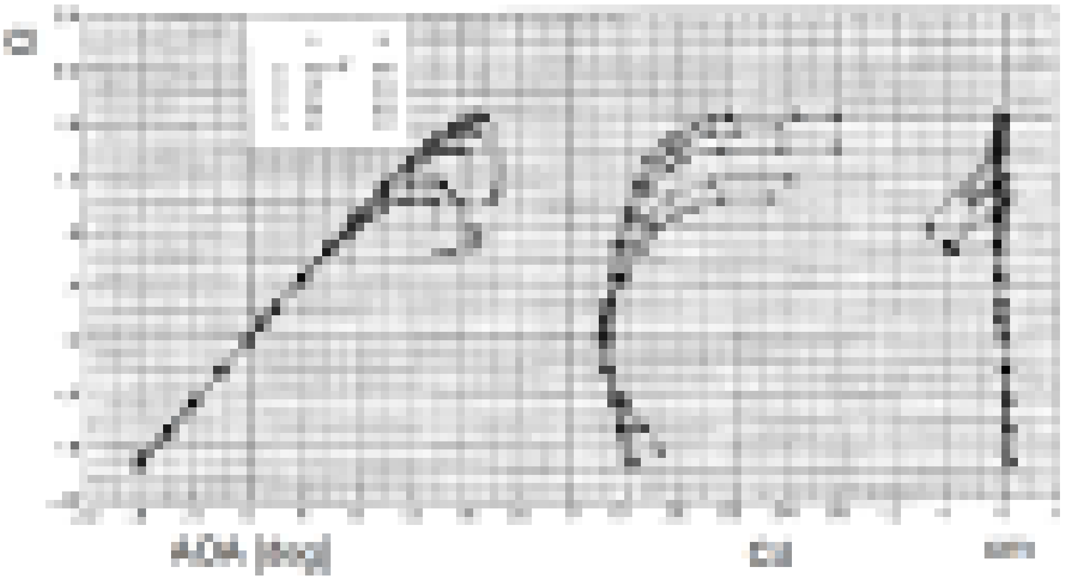


Figure A-2: Variations of lift coefficient versus angle of attack, drag coefficient, and pitching moment coefficient at Mach number less than or equal to 0.15 for NAC0013 at various values of Reynold number [205]

Appendix B – Validation Procedure

B.1 Validation and verification of the flow solver

Before applying the methodology developed here to the Aegis UAV configuration, three different test cases taken from the literature were used to validate and verify of the results obtained from the flow solver and the Interface-AVL.

B.1.1 3-D wing (NACA4415)

The first objective was to determine the lift coefficient versus angle of attack for 3-D wing of AR= 12 with a rectangular planform using the AVL code and to compare the results obtained using Computational Fluid Dynamics (CFD) and available in [206]. Figure B-1 shows the CFD results obtained at Re = 3 x 10^6 based on the chord and a Mach number of 0.2.

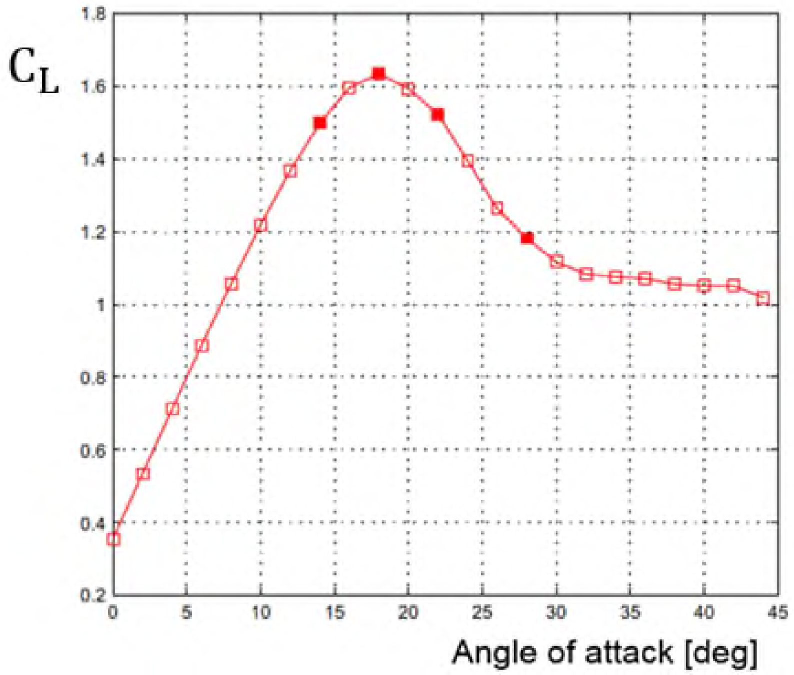


Figure B-1: Lift coefficient versus angle of attack for rectangular wing AR= 12, Re= 3x10^6, and M= 0.2 using computational fluid dynamic [206]

The comparison of the results showed good agreement for the overall lift coefficient versus angle of attack, see Table B-1. The AVL code shows high accuracy at a low angle of attack but started to differ slightly as the angle of attack increased [137].

Table B-1: Comparison of the lift coefficient that obtained using AVL and CFD [206] **at various angle of attack**

AOA [deg]	0	2.5	5	7.5	10
C_L using AVL	0.38	0.60	0.82	1.04	1.25
C_L using CFD	0.38	0.60	0.80	1.00	1.23

B.1.2 ATHENA project (Aegis UAV)

The Aegis UAV is under development by a team at Cranfield University, and is available with two different tail configurations. The first version of the UAV (copy the Israeli MALE UAV, IAI Scout) was studied at an operational velocity of 50 m/s, where the Reynolds number - based on the chord - was around 2.5 x 10^6. When CFD was used for this simulation it was assumed the flow was fully turbulent. The objective of the study was to investigate the performance of the UAV for various tail angles at an angle of attack equal to zero [191].

The coefficients of lift and drag as computed by CFD and AVL is presented in Table B-2.

Table B-2: Lift and drag coefficients computed by AVL and CFD for Aegis UAV at zero AOA

Parameter	Inverted V-tail		% difference	H-tail		% difference
	AVL	CFD		AVL	CFD	
C_L	0.782	0.660	15.0	0.780	0.670	14.0
C_D	0.053	0.059	11.0	0.054	0.060	11.0

It is evident that CFD and AVL are in good agreement. Generally, the AVL software predicted higher values of lift coefficient than the CFD. On the other hand, CFD shows higher values of drag coefficient than obtained by AVL for both configurations. The reason may be because the CFD calculations assumed the flow on the UAV is entirely turbulent which, of course, will increase the drag slightly and reduce the lift coefficient. However, the difference over-all is less than 15%.

B.1.3 Aegis UAV (ESDU)

In this case, the AVL results for the Aegis UAV with U-tail shape were compared with the available results computed using the Engineering Sciences Data Unit (ESDU) in [190]. Figure B-3 shows the comparison of the lift coefficients versus angle of attack.

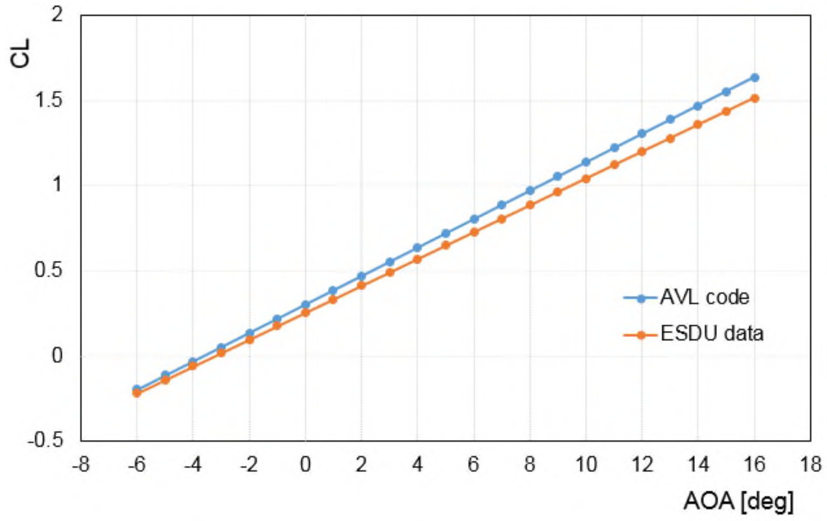


Figure B-3: Comparison of lift coefficient obtained using AVL code and EDSU data as a function of angle of attack for Aegis UAV with U-tail configuration

The compared results show good agreement in the overall lift coefficient versus angle of attack between the AVL results and the ESDU data, though a slight but increasing difference is observed as the angle of attack increases. It is known that AVL is more appropriate for a small angle of attack [137]. In addition, the alfa zero lift is equal to -3.64 deg using AVL and -3.24 deg using ESDU.

Appendix C – Performance Analysis of the Aegis UAV

C.1 Performance analysis of the base design

A full study was performed to explore the Aegis UAV design space and its characteristics in level flight. Simulation of each configuration as a function of the angle of attack was performed. The lift coefficient, drag coefficient, drag polar, CL/CD ratio, and pitching moment coefficient curves were plotted to visualise the performance at a various angles of attack, see Figure C-1 to Figure C-3.

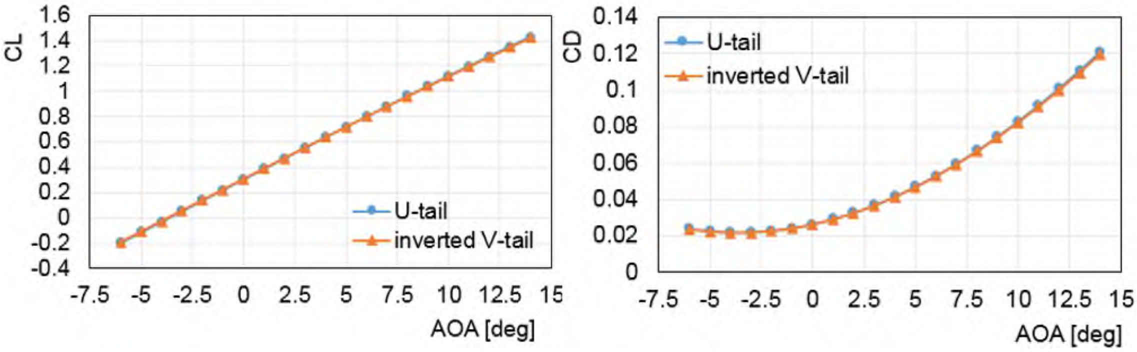


Figure 9: left panel; lift coefficient versus angle of attack: right panel; drag coefficient versus angle of attack

The lift and drag coefficients versus angle of attack for the Aegis UAV with both tail arrangements are shown in Figure C-1. It is observed that both configurations show the same trend. However, the Aegis with U-tail showed the higher lift coefficient, whereas the Aegis with inverted V-tail showed the lower drag coefficient at various angles of attack. Furthermore, both configurations have a zero lift angle of attack around -3.6 degrees; and lift at zero angle of attack of around 0.304. The left panel of Figure C-1 shows an idealized lift curve, in reality, the slope of the curve would decrease at higher angles of attack approaching the maximum lift coefficient.

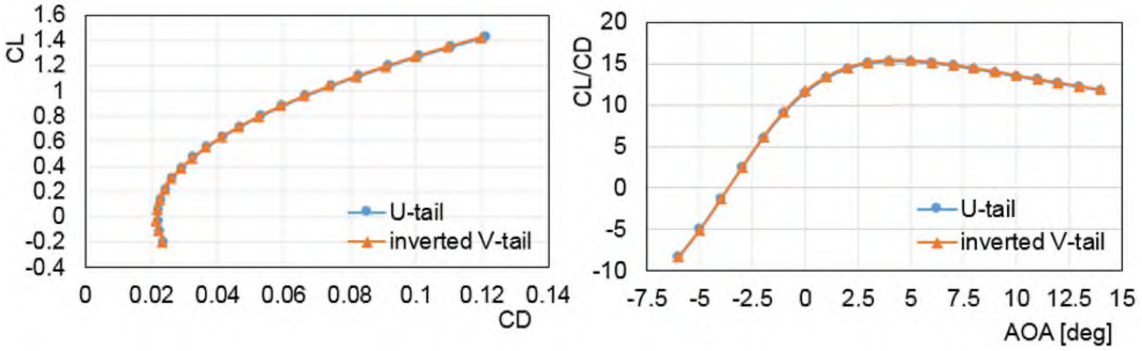


Figure 9: left panel; lift coefficient versus drag coefficient: right panel; lift to drag ratio versus angle of attack

The left panel of Figure C-2 shows the relation between lift and drag ratios for Aegis UAV configurations. It is seen that the drag coefficient at zero lift for Aegis with U-tail and inverted V-tail is 0.021620 and 0.021480, respectively, which is considered to be the minimum drag acting on the configurations. The Aegis UAV with inverted V-tail suffers less drag because of less tail surface area. In addition, lift to drag coefficient ratio versus angle of attack presented in the right panel of Figure C-2 shows a slightly higher value of CL/CD for the Aegis UAV with inverted V-tail. The maximum value of CL/CD for the Aegis UAV with U-tail and inverted V-tail is 15.31 and 15.42, respectively at AOA equal to 4 deg.

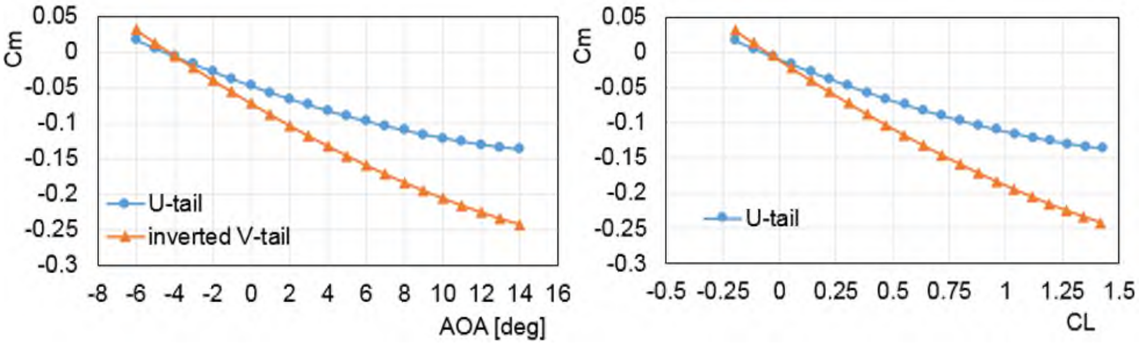


Figure C-3: left panel; pitching moment versus angle of attack: right panel; pitching moment versus lift coefficient

The measurement of pitching moment was taken at the centre of gravity of each configuration (CG location) using the AVL code. Figure C-3 indicates a longitudinally stable UAV. The pitching moment at zero AOA is -0.0470 for the Aegis with U-tail and -0.0710 for inverted V-tail. It is obvious that the slopes of these curves are negative and are in the region of negative Cm for a typical range of angles of attack. The positive moment at a negative angle of attack for all configurations shows a tendency to “nose up” during the pitching down, whereas the pitching moment for all configurations becomes negative at a positive angle of attack which gives a tendency to “nose down” at these angles. Thus, both configurations are statistically stable.

On the other hand, neither configurations is balanced. The balance criteria is satisfied if the pitching moment at UAV centre of gravity (Cm_{xcg}) is zero for the lift coefficient value equal to that for level flight. It is obvious that the pitching moment coefficients (for both configurations) at zero angle of attack are not equal to zero. Commonly, one of the following is required to achieve a trimmed UAV during level flight: (1) change of location of Xcg of the UAV by mass redistribution, (2) adjustment of the horizontal tail incident angle, (3) adjustment of the horizontal tail volume, and (4) change elevator deflection.

Following is a more detailed discussion of each of the above four cases:

- **Relocate the centre of gravity by changing mass distribution.**

The possibility of moving the centre of gravity of the Aegis UAV, so that the pitch moment will be equal to zero was investigated. Figure C-4 shows locations of Xcg for the UAV, and neutral point (np) relative to the wing leading edge. It is obvious that both configurations are stable, since the neutral point is located behind the centre of gravity.

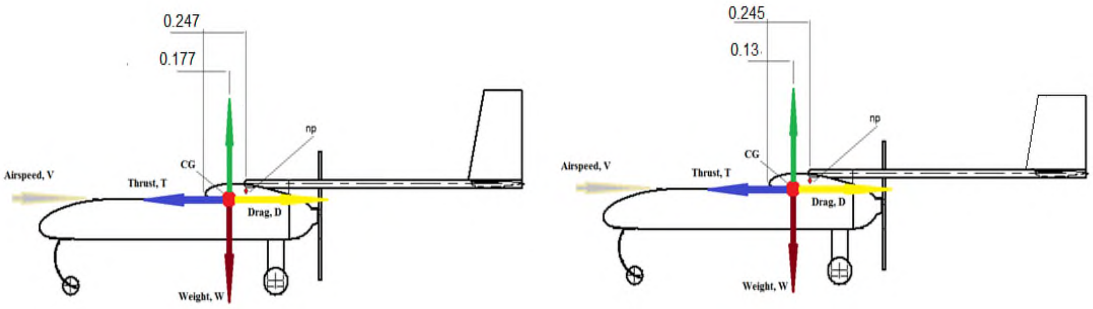


Figure C-4: Centre of gravity and natural point location for Aegis UAV; left panel Aegis UAV with U-tail and right panel Aegis UAV with Inverted V-tail

The study showed that as the pitching moment becomes close to zero (which is our aim) by relocating the centre of gravity, the aircraft becomes less stable, see Table C-1. It is obvious that the Static Margin (SM) becomes close to zero. Thus, relocating the centre of gravity of the UAV using internal mass redistribution not suitable.

Table C-1: Redistribution of the Ages UAV mass results in a pitching moment equal to zero, but an unstable aircraft

No	U-tail			Inverted Vee-tail		
	Xcg	Cm	S.M	Xcg	Cm	S.M
1 (BD)	0.17	-0.0470	11.500	0.13	-0.0719	19.220
2	0.18	-0.0457	11.126	0.17	-0.0457	11.503
3	0.20	-0.0353	8.020	0.20	-0.0352	8.415
4	0.23	-0.0196	3.370	0.23	-0.0195	3.789
5	0.25	-0.0091	0.287	0.25	-0.0090	0.706

- **Adjustment of the horizontal tail incident angle**

This method required the possibility that the horizontal tail could be rotated so that the pitching moment would be zero during level flight. In this work, the optimiser was used to find this angle in the case of the Aegis UAV with U-tail (case 2 and case 4).

- **Adjustment of the horizontal tail volume**

Adjusting the horizontal tail surface area and its distance from the UAV's centre of gravity may result in obtaining a pitching moment coefficient equal to zero. However, such solutions usually are limited by geometrical and flight constraints. Unfortunately, the optimiser could not satisfy Cm=0 using the proposed design space (see Table 5-1).

- **Elevator deflection**

It is common to use elevator deflection to get zero pitching moment during level flight. It can be pre-set before the flight.

Appendix D – Optimisation Results

D.1 Results for different design optimisation cases using MOTS algorithm

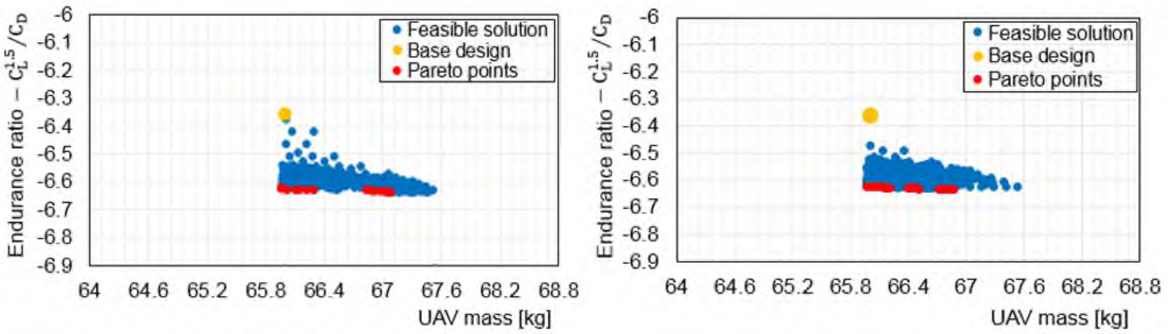


Figure D-1: Optimisation results obtained using 5 regions and 2400 evaluations under pitching moment constraint for the UAV with U-tail using wing design variables

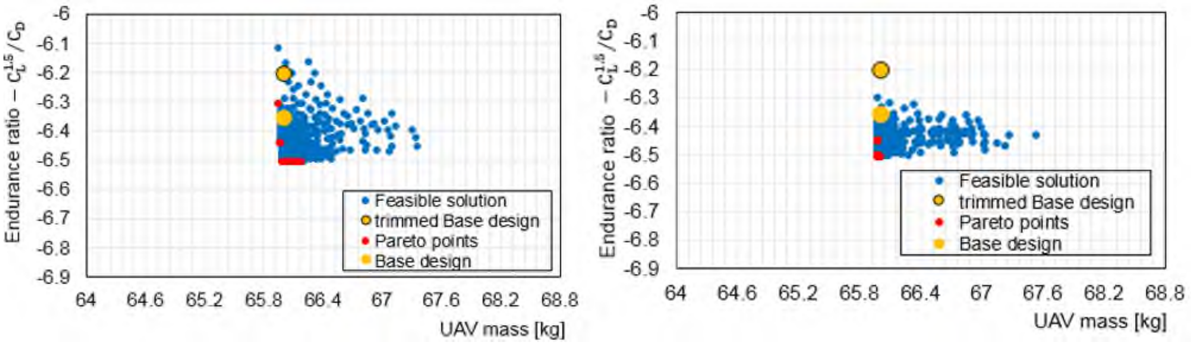


Figure D-2: Trimming optimisation results obtained using 5 regions and 3000 evaluations for the UAV with U-tail shape using wing, tail, and horizontal tail rotation angle design variables

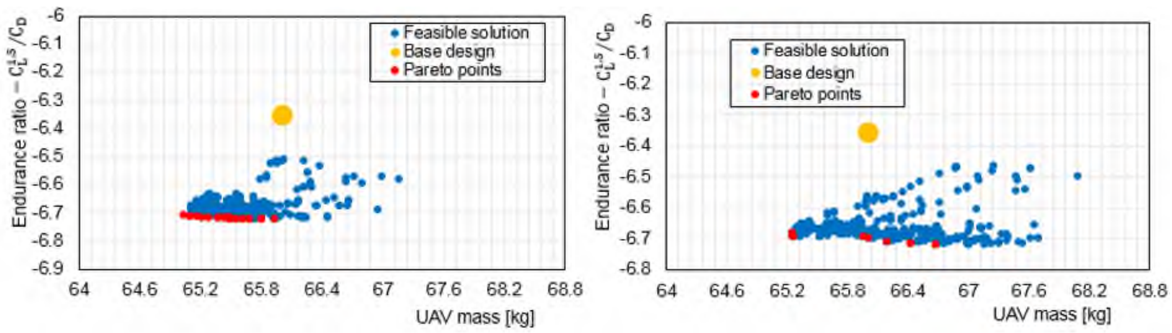


Figure D-3: Feasible solutions obtained using 5 regions and 5500 evaluations for the UAV with U-tail shape using wing and tail design variables simultaneously

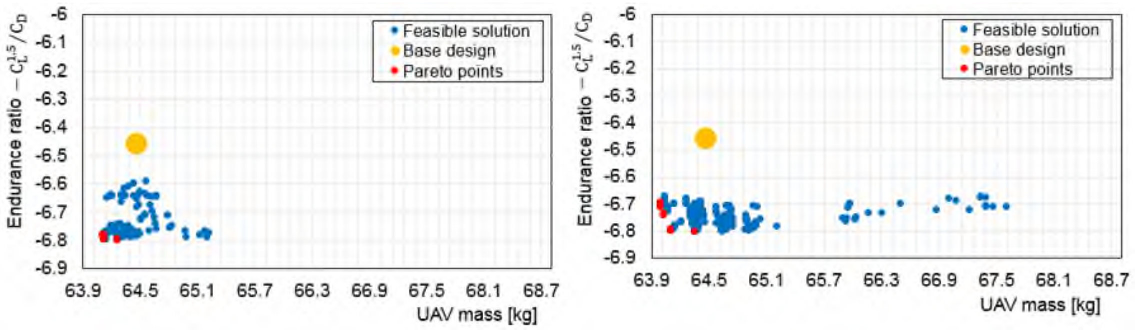
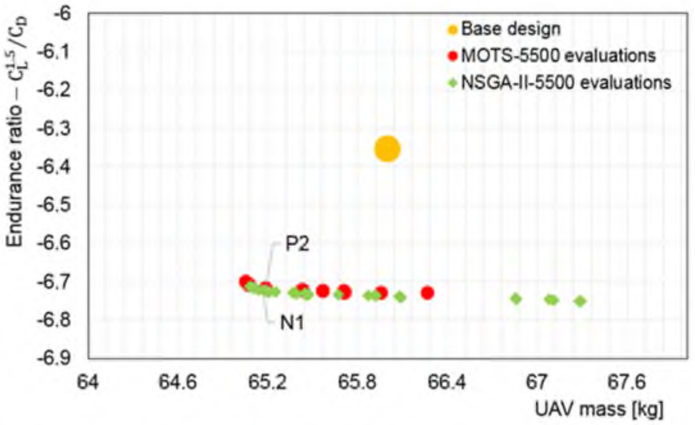
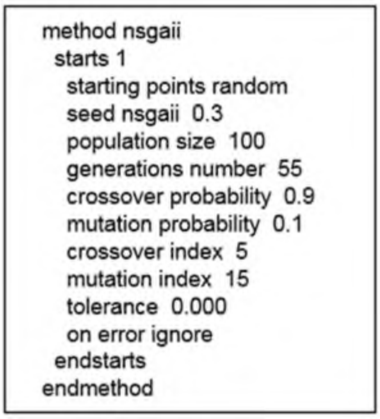


Figure D-4: Feasible solutions obtained by optimising the UAV with inverted V-tail using wing and tail design variables simultaneously

D.2 Performance analysis

To assess the relative effectiveness of using MOTS for this study, a comparison with the optimisation results obtained by NSGA-II [207] was performed for one design scenario. Case 3 was selected as it had the best improvement and the highest number of design variables. Since the NSGA-II algorithm has a non-deterministic behaviour, the results of multiple runs using the same design variables in case 3 were collected and assessed. In every case the experiments were run for 5500 function evaluations. The NSGA-II algorithm used 55 generations for a population size of 100. The left panel of Figure E-5 presents the optimisation results obtained using MOTS for 5500 evaluations and NSGA-II also for 5500 evaluations. The right panel of Figure E-5 shows the NSGA-II parameter settings.





```
method nsgaii
  starts 1
    starting points random
    seed nsgaii  0.3
    population size  100
    generations number  55
    crossover probability  0.9
    mutation probability  0.1
    crossover index  5
    mutation index  15
    tolerance  0.000
    on error ignore
  endstarts
endmethod
```

Figure D-5: Left panel; comparison of the optimisation results obtained using MOTS and NSGA-II for 5500 evaluations: right panel; shows the parameter setting for NSGA-II

It is obvious that both algorithms achieved almost the same quality solutions, but the non-dominated solutions obtained by NSGA-II have greater variance. On the other hand, regardless of the optimisation and search techniques used by each algorithm, the run-times for MOTS and NSGA were 135 and 142 minutes respectively. This indicates that the MOTS algorithm is slightly more efficient than NSGA-II for such design problems. Overall, the MOTS algorithm exhibited better performance when compared to the leading multi-objective GA, NSGA-II. In addition, given the similarity in the obtained solutions, it is not worth investigating the relative performances of these solutions.

Appendix E – MOPSO and ANN Input Files

E.1 Setting for the MOPSO input file

```
=======================INPUT FILE =================

------------------- -GENERAL - ---------------------------------------------------

-(1) PARAMETERS:   "9"

=>      number of parameters -> EX: "6"

-(2) OBJECTIVES:  "2"

=>      number of objectives -> EX: "2"

-(3) DIRECTORY:    "/server/airfoilservice/"

=>      directory of the model/solver -> EX: "/user/Caesar/Desktop"

-(4) NAME:   "'/usr/local/bin/matlab -nosplash -nodisplay -r  .

\'Endurance(%s,%s,%s,%s,%s,%s,%s,%s,%s)\',exit'%(params[0],      params[1],
params[2], params[3], params[4], params[5], params[6], params[7], params[8])"

=>      name of the model/solver executable -> EX: "model.exe"

-(5) MAX PAR:            "4.5   0.74   1.0   0.55   0.035    2.0   4.0    2.5   1.0"

=>      maximum values ("+1" by default - if not desired leave '0' as default) ->
EX: "4 0.5 -2 3 1 4"

-(6) MIN PAR:            "2.59   0.55   0.6   0.35   0.02   1.45   3.0   1.5   0.5"

=>      minimum values ("-1" - if not desired leave '0' as default) -> EX: "0 200 200
0 -200 -200"

-(7) SCALE OBJ:     "1 -1"

=>     objective scaling factor (optional - if not desired leave '0' as default) -> EX:
"-1 1"

--------------- - INITIALISATION - ----------------------------------------------------------
```

the following parameters affect the initialisation in the NORMALISED interval [-1,1]

-(8) INITIALISATION: "g"

=> choose the initialisation method: 'g' for Gaussian, 'u' for Uniform

-(9) UPPER BOUND uniform: "0.02"

=> upper bound for uniform initialisation - you can insert a different distribution for each parameter for fixed UPP_BOUND (all par.) type: "value" - EX: "-1" - in alternative specify each par. (like in MAX-PAR)

-(10) LOWER BOUND uniform: "-0.2"

=> lower bound for uniform initialisation - you can insert a different distribution for each parameter for fixed LOW_BOUND (all par.) type: "value" - EX: "+1" - in alternative specify each par. (like in MIN-PAR)

-(11) MEAN gauss: "0"

=> gauss mean for Gaussian initialisation - you can insert a different distribution for each parameter for fixed mean (all par.) type: "value" - EX: "0" - otherwise specify "value1 value2 ... valueN"

-(12) SIGMA gauss: "0.2"

=> gauss sigma for Gaussian initialissation - you can insert a different distribution for each parameter for fixed sigma (all par.) type: "value" - EX: "0.2" - otherwise specify "value1 value2 ... valueN"

--------------- - MOPSO - ---

-(13) C1: "2.0"

=> personal weight

-(14) C2: "2.0"

=> leader's weight

-(15) w: "0.4"

=> inertia weight -> EX: "0.4"

-(16) SIGMA-TURBULENCE: "0.05"

=> stardard variation for the Turbulence correction -> EX: "0.1"

-(17) Archive size: "100"

=> Size of the archive of non-dominated particles

E.2 Setting for the ANN-MOPSO input file

======================INPUT FILE =======================

(1) PARAMETERS: "9"

=> *number of parameters -> EX: "6"*

(2) OBJECTIVES: "2"

=> number of objectives -> EX: "2"

(3) DIRECTORY: "/server/airfoilservice/"

=> directory of the model/solver -> EX: "/user/Caesar/Desktop"

(4) NAME: "'/usr/local/bin/matlab -nosplash -nodisplay -r

\'Endurance(%s,%s,%s,%s,%s,%s,%s,%s,%s)\',exit' % (params[0],params[1], params[2], params[3], params[4], params[5], params[6],

params[7],params[8])"

=> name of the model/solver executable -> EX: "model.exe"

(5) MAX PAR: "4.5 0.74 1.0 0.55 0.035 2.0 4.0 2.5 1.0"

=> maximum values ("+1" by default - if not desired leave '0' as default) -> EX: "4 0.5 -2 3 1 4"

(6) MIN PAR: "2.59 0.55 0.6 0.35 0.02 1.45 3.0 1.5 0.5"

=> minimum values ("-1" - if not desired leave '0' as default) -> EX: "0 200 200 0 -200 -200"

(7) SCALE OBJ: "1 -1"

=> objective scaling factor (optional - if not desired leave '0' as default) -> EX: "-1 1"

(8) OUTPUT DIRECTORY path: "/home/airfoil"

=> directory of the model/solver -> EX: "/user/Caesar"

E.3 Single training versus continuous training

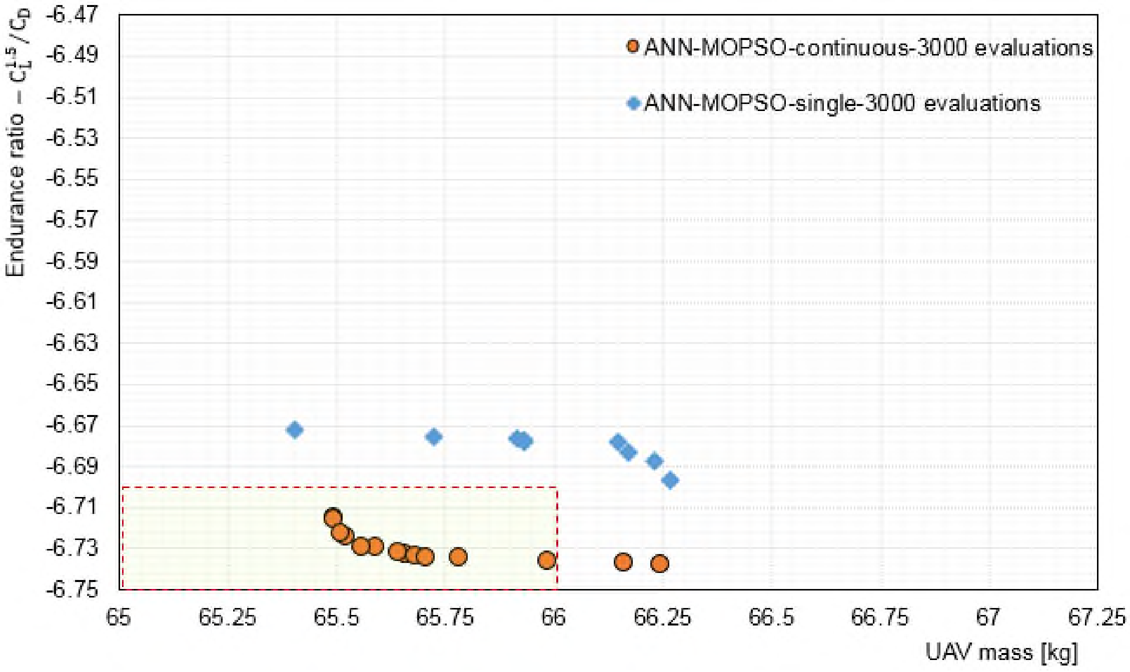


Figure E-1: Comparison of the solutions obtained using single training with continuous training for 3000 evaluations

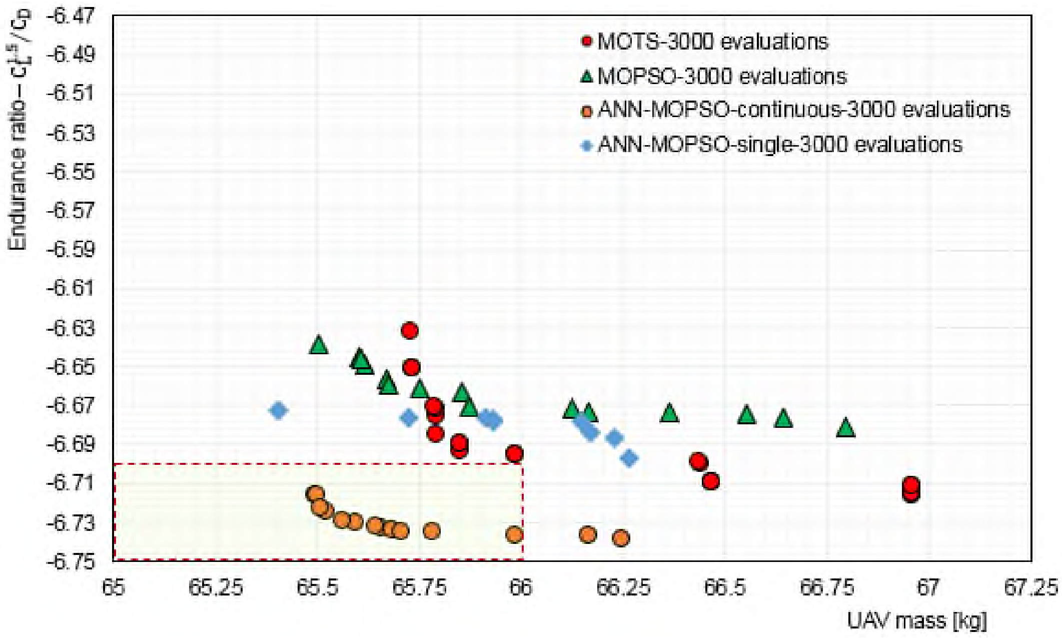
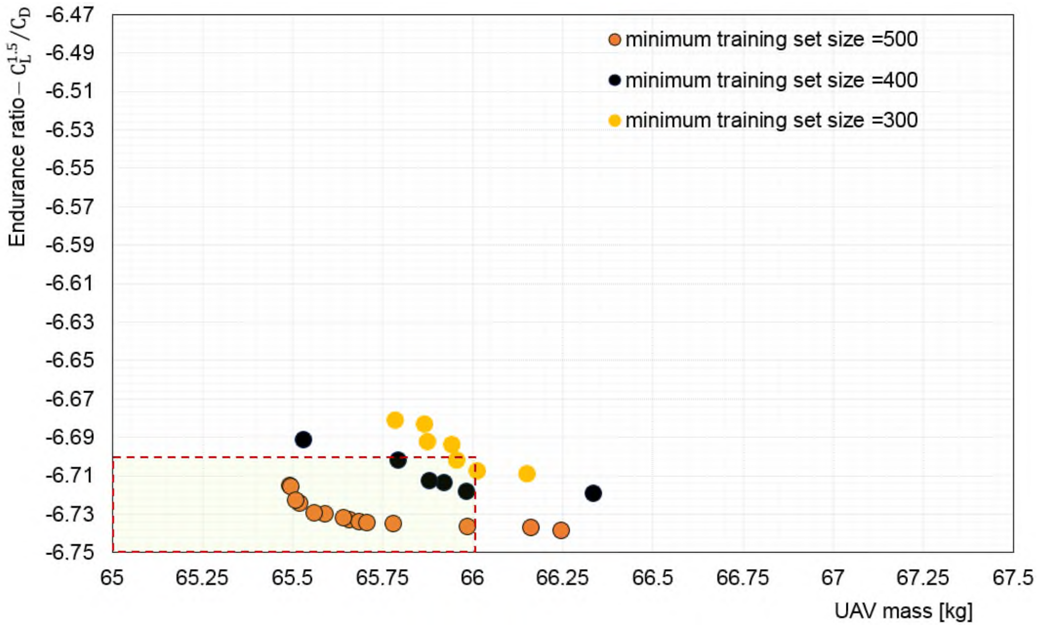


Figure E-2: Comparison of Pareto front using MOTS, MOPSO, I-MOPSO for 3000 evaluations with ANN-MOPSO using continuous and single training for 3000 evaluations



E-3: Comparison using different training set size in the case of continuous training using 3000 evaluations