Page 1 of 34. © Royal Aeronautical Society 2016
doi: 10.1017/aer.2016.1

Optimisation of aero-manufacturing characteristics of aircraft ribs

Taejoo Kim
teddy9835@gmail.com

Timoleon Kipouros

Department of Engineering, University of Cambridge, Cambridge, UK.

Alexandra Brintrup

Institute for Manufacturing, University of Cambridge, Cambridge, UK.

James Farnfield

GKN Aerospace Services Ltd, Filton, UK.

Davide Di Pasquale

School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, UK.

ABSTRACT

The main purpose of this study was to combine the currently separate objectives of aerodynamic performance and manufacturing efficiency, then find an optimal point of operation for both objectives. An additional goal of the study was to explore the effects of changes in design features, the position of the spars, and analyse how the changes influenced the optimal operating conditions. A machine-learning approach was taken to combine and model the gathered aero-manufacturing data, and a multi-objective optimisation approach utilising genetic algorithms was implemented to find the trade-off relationship between optimal target objectives, mission performance and manufacturability. The main achievements and findings of the study were: the study was a success in building a machine-learning model for the combined aero-manufacturing data utilising software library XGBoost; Multi-objective optimisation which did not include spar positions as a variable found the trade-off region between high manufacturability and high mission performance, with choices that can have reasonably high values of both; there was no clearly identified correlation between a small change in spar position and the target objectives; multi-objective optimisation with spar positions resulted in a trade-off relationship between target objectives which was different from the trade-off relationship found in optimisation without spar positions; multi-objective optimisation with spar positions also offered more flexibility in the choice of manufacturing processes available for a given design; and the range of bump amplitudes for solutions found by multi-objective-optimisation with spar positions was lower and more focused than those found by optimisation without spar positions.

Received DD MM YYYY; revised DD MM YYYY; accepted DD MM YYYY.

1.0 Introduction

The aerodynamic industry, and the field of aerodynamic design, is one of the most challenging and difficult industries today due to its complexity in the different factors involved as well as their interactions. As there are many different design factors and their interactions with the real world to consider, traditional methods of aerodynamic design involve complex modelling and simulations to test and validate solutions[1]. The traditional method also has the benefit of being able to test solutions and variations without massive capital investment involved with physically building an aircraft. However, it also has the downside of being very costly and slow as a software itself, often taking days or even weeks to accurately simulate a solution even when using a high-spec computer[2].

One of the newer methods of approaching this problem of designing and testing different solutions involve the principle of optimisation - more specifically, algorithm-based optimisation. Algorithm-based optimisation is a mathematics-based method of finding optimal operating conditions and trade-off relationships between different factors of interest. With the rapid development in computing capacity and software design, optimisation methods, such as multi-objective optimisation, have grown as an effective way of analysing the complex relationships between different factors, and are being used more widely across different fields of engineering[3]. They can also offer insights or possibilities that have not previously been considered before, as the data-driven approaches can consider a much wider variety of design factor combinations than conventional design approaches.

However, not all parts of the aerodynamic industry employ such methodology in their operations. Manufacturing in the aero-industry is one such field, where the manufacturers of various parts of the aircraft are mainly dependent on the designers of the parts to produce the specifications, without much regard to their ease of manufacturing. This stems mainly from the high-risk nature of the industry itself, where the monetary importance of being able to produce a part slightly more cheaply is much less significant compared to the safety-critical importance of a part being exactly compliant with the design specifications. Recently, the trend in aerospace industry is becoming more and more focused towards faster production and rapid development cycles, with division of demand based on customers' specific needs[4,5], which also puts more pressure on the manufacturers in meeting the various needs within time. While such trends has lead to more interest in and implementation of lean manufacturing[6], there are still significant room for improvement in the design and manufacturing of aerospace parts[7], as there has been little opportunity to optimise the design of a part with its manufacturing characteristics in mind.

Due to this disparity between the design and manufacturing of parts, an optimisation model which takes not only aerodynamic but also manufacturing characteristics in the design of a part was deemed necessary. For this purpose, a Python-based machine-learning model was created using the combined data between aerodynamic and manufacturing data of an aircraft rib. This model was then presented to a multi-objective optimisation algorithm to explore different options within the provided limits obtained from the machine learning models, to explore the impact on the design of parts induced by the inclusion of their corresponding manufacturing characteristics.

This study is composed of 2 main studies exploring two different aspects of optimisation with both design and manufacturing characteristics:

The first study involves building of a machine learning model which can accurately predict the behaviour of the rib based on the combined data between aerodynamic performance and manufacturing efficiency, then performing multi-objective optimisation to understand the trade-off relationship between manufacturing and design characteristics.

The second study explores the impact in aerodynamic performance due to the inclusion of manufacturing characteristics by introducing a new variable, the front and rear spar positions. The extended dataset including values from the new variables was used to build a machine learning model and perform multi-objective optimisation, same as the first study. The results of the optimisations from the first and second study were compared to analyse the change in the optimal trade-off points due to the inclusion of spar positions.

2.0 Problem Identification

The main identified problem, as briefly mentioned in the Introduction, was the lack of sufficient back-and-forth communication between design and manufacturing. The aircraft rib design is generated according to the desired airfoil shape by the designer of the aircraft, then the shape of the rib is communicated to the manufacturer of the rib for production. During this process, there is little input from the manufacturer on what the rib shape should look like, as the requirements of the shape depend on the aerodynamic performance of the rib, which is decided by the designer of the wing.

This means the manufacturer has little opportunity to express their opinion on what type of design is most desirable for the efficiency of their manufacturing process. The manufacturer tries to meet the specifications given to them as closely as possible, but depending on what kind of manufacturing processes or tools are available to the manufacturer, the final manufactured rib may deviate from the initial intended design (see Figure 1). Also, the manufacturer's choice in which process and tools to use depends largely on the given specification, rather than what is efficient for the manufacturer, so their efficiency in processes and manufacturing is less than optimal.

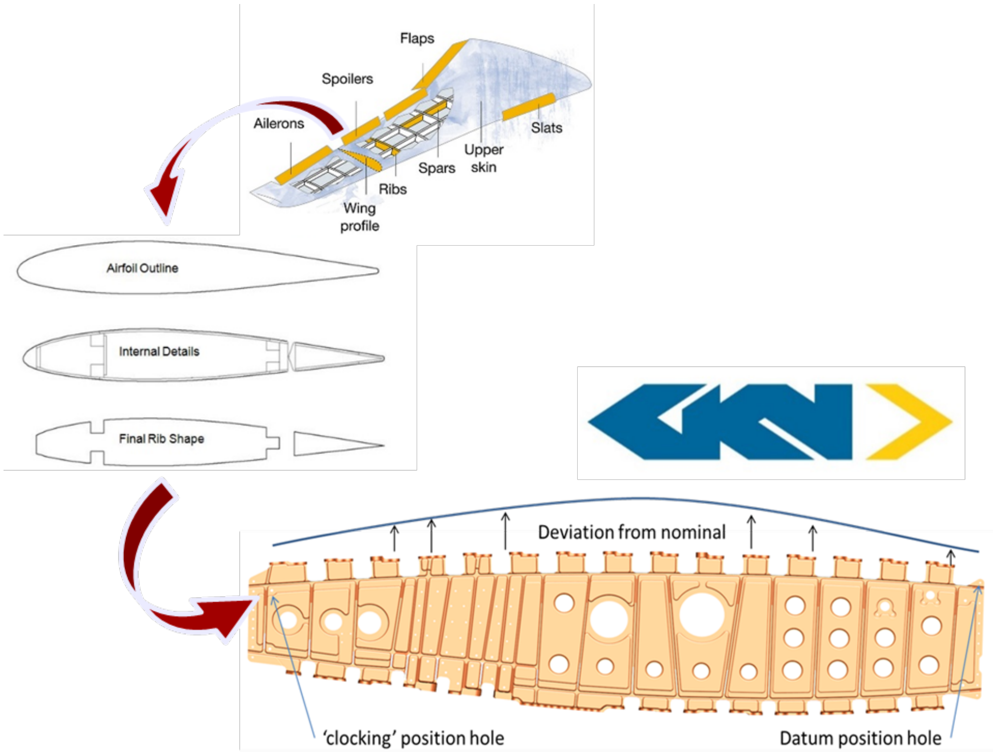

Figure 1. The design and manufacturing process of an aircraft rib

This means there is potential for improvement in both the performance of the rib and its manufacturing efficiency. Currently, the two main objectives in manufacturing a rib – aerodynamic performance and manufacturing efficiency – are considered separately. However, there is a great potential benefit in being able to consider both sides of the issue at once, and finding conditions which satisfy both sides the most at the same time. By considering the relevant manufacturing characteristics in the design stage of the rib, the relationships and trade-offs between better-performing design and their manufacturing challenges can be understood better, and optimal operating conditions can be found.

Furthermore, exploring multiple objectives at the same time can open more opportunities for improvement in design. Changes in design can also have a large potential influences in optimal performance and manufacturing, and considering design changes as a variable in manufacturing can provide further insight about the design space and potentially reveal better operating conditions that have never been considered before.

3.0 Software used in this study

The main focus of this paper is to explore the potential impact of using algorithm-based methods to consider both design and manufacturing characteristics. Thus, it is important to understand the specific software used for the study and consider how they can be utilised to achieve the desired methodology. A previous study[(8)] on optimisation of S-duct design was reviewed to aid the methodology design. This section discusses the software used in this study - for mathematical modelling, multi-objective optimisation (MOO) and visualisation and interpretation of the results.

3.1 Mathematical modelling through supervised learning – Extreme Gradient Boosted Learning (XGBoost)

In this section, the chosen method of modelling, supervised learning, and the specific software chosen for this study, XGBoost, is discussed.

3.1.1 Supervised Learning

There are various ways of conducting supervised machine learning (ML) to build mathematical models of systems and their behaviours. Some of the more well-known ML methods include[9]:

- Supprt Vector Machines, which involves finding an optimal hyperplane to classify data
- Linear modelling such as regression modelling, which involves a linear combination of different base functions to model a system
- Artificial Neural Networks, which involves modelling through adjustment of coefficients of multiple layers of output functions
- Tree learning, which involves training a model based on decision trees for a given dataset

A supervised ML model starts from defining an appropriate objective function, which measures how well a model fits the necessary data. Objective functions mainly consist of two factors: the training loss, which measure how predictive a model is against data used for training, and the regularisation term, which is a factor included to prevent the model from overfitting the data. Some common examples of training loss functions include the Mean Squared Error (MSE), which is the sum of squares of the difference between each prediction value and their true values, or Logistic Loss, which is a measure of error for fitting logistic regression functions. The goal of a supervised learning model is to define a proper objective function according to target objectives and to optimise the objective function for best performance.

The initial choice of ML method was Extreme Gradient Boosted Learning (XGBoost) and its software, as XGBoost is one of the most popular choices for ML modelling due to its efficiency and performance in learning non-linear decision boundaries[10]. After using the Python library for XGBoost, it was found (to be explained in later parts of the report) that XGBoost performed well in predicting the values for the given dataset, and offered various in-house functions and modelling methods. Thus, finding a different ML method to compare performances was deemed unnecessary. A brief theoretical background on XGBoost is explained below (original document can be found at[11]).

3.1.2 XGBoost - Finding Prediction Values

XGBoost builds models based on decision tree ensembles, which is a series of classification and regression trees (CARTs). A CART is a decision tree method which assigns real scores to each leaf found through classification.

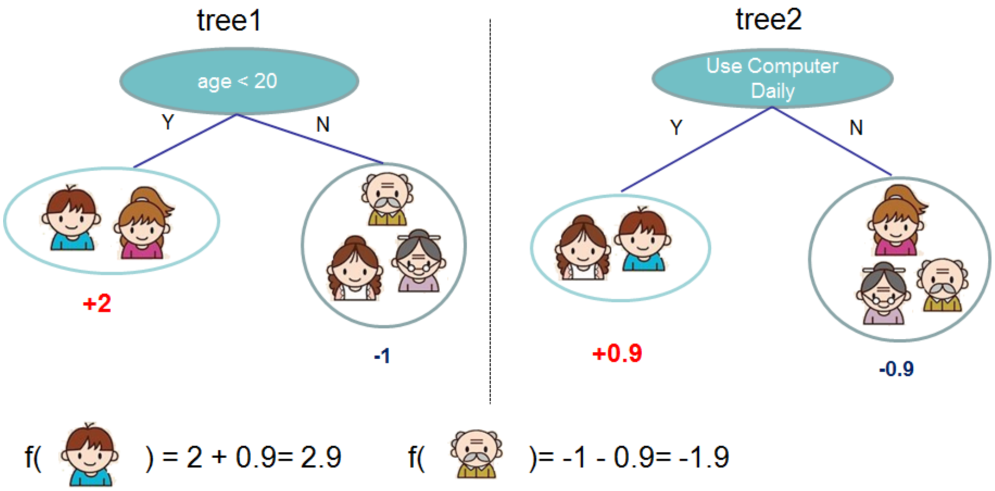

Figure 2. Diagram representing ensemble of CARTs

As visible from the above Figure 2, each tree represents a CART, and the prediction score of each output variable in an ensemble is defined as the sum of scores across the leaves of all trees. The prediction value for an input xi¬ then can be expressed in the form of the following equation:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F$$

With K being number of trees, f_k being a function in the functional space F and F being the set of all possible CARTs.

3.1.3 Objective Function of a Tree

With this prediction, the objective function can be written in the following format:

$$obj = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Where l(a,b) is the training loss function between the prediction and the real value, and Ω represents the regularisation term.

In XGBoost, the prediction value at the nth step is defined as the sum of the prediction values from the steps before. If prediction value at for input x_I at the nth step is defined as y_i^n, the formula for y_i^n is derived as follows:

$$y_i^0 = 0$$

$$y_i^1 = f_1(x_i) = y_i^0 + f_1(x_i)$$

$$y_i^2 = f_2(x_i) + f_1(x_i) = y_i^1 + f_1(x_i)$$

and so forth until the nth term:

$$y_i^n = f_n(x_i) = y_i^{n-1} + f_n(x_i)$$

The objective function at this nth step then can be written as:

$$obj = \sum_i^n l(y_i, y_i^{n-1} + f_n(x_i)) + \Omega(f_n) + constant$$

Then to optimise this objective function, a Taylor expansion of the loss function up to second order is taken and can be expressed as:

$$obj = \sum_i^n l(y_i, y_i^{n-1} + g_i f_n(x_i) + h_i f_n^2(x_i)) + \Omega(f_n) + constant$$

Where g_i and h_i are defined as the first-order and second-order partial derivatives of the loss function:

$$g_i = \partial_{y_i^{n-1}} l(y_i, y_i^{n-1})$$

$$h_i = \partial^2_{y_i^{n-1}} l(y_i, y_i^{n-1})$$

The specific objective to maximise at each step, without any constants, is then expressed as the following:

$$\sum_i^n (g_i f_n(x_i) + h_i f_n^2(x_i)) + \Omega(f_n)$$

3.1.4 Model Complexity and Overall Structure Score

If the definition of the tree f(x) is refined in vector form:

$$f_n(x) = \omega_{q_x}, \omega \in R^T, q : R^d \rightarrow 1, 2, 3, ..., T$$

With ω being vector of scores on leaves, q a function which assigns each data point to corresponding leaf and T being the number of leaves. Then the complexity Ω is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$$

The objective function for the nth tree then can be re-written as the following:

$$obj^n \approx \sum_i^n (g_i f_n(x_i) + \frac{1}{2} h_i f_n^2(x_i)) + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$$

$$= \sum_{j=1}^{T} \{(\sum_{i \in I_j} g_i)\omega_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda)\omega_j^2\} + \lambda T$$

with $I_j = \{i|q(x_i = j)\}$ being the set of indices of data points assigned to the j-th leaf. Further compressing the equation by setting $G_j = \sum_{i\in I_j} g_i$ and $H_j = \sum_{i\in I_j} h_i$:

$$obj^n = \sum_{j=1}^{T}\{G_j\omega_j + \frac{1}{2}(H_j+\lambda)\omega_j^2\} + \gamma T$$

In this case, ω_j are independent with respect to each other, so the form of $G_j\omega_j+\frac{1}{2}(H_j+\lambda)\omega_j^2$ is quadratic and the best ω_j given for a structure $q(x)$ and the best objective reduction becomes:

$$\omega_j^* = -\frac{G_j}{H_j+\lambda}$$

$$obj^* = -\frac{1}{2}\sum_{j=1}^{T}(\frac{G_j^2}{H_j+\lambda}) + \lambda T$$

which is a good measure of how good a tree structure f(x) is.
A diagram providing simpler explanation of the structure is shown in Figure 3 below:

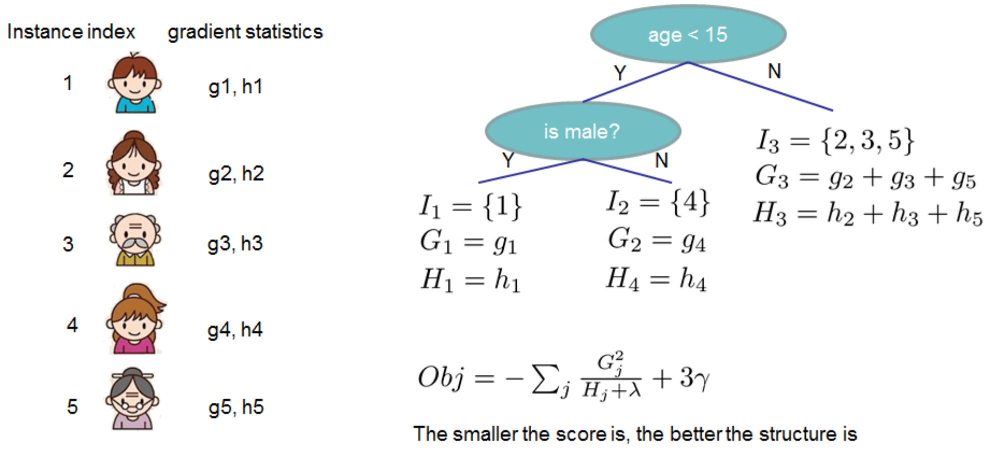

Figure 3. Diagram calculating structure score for given tree structure

For a given tree structure, statistics g_j and h_j are pushed to the leaves they belong to, the statistics are summed together, and the formula is used to calculate how good the tree is.
The final step is to understand where to split, which is done by splitting a leaf to two leaves and calculating the change in gain by doing so:

$$Gain = \frac{1}{2}\{\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} + \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\} - \lambda$$

Which is the difference between sum of scores on new left and right leaves and the score on the original leaf, subtracted by the regularisation on the additional leaf. This means that if the gain is smaller than λ it would be better not to add that leaf.

3.1.5 Limitations

Since it is intractable to enumerate all possible tree structures, one split is added at a time. This approach works well most of the time, but some edge cases fail due to this approach. For those edge cases, training results in a degenerate model because only one feature dimension is considered at a time.

3.1.6 DART Booster

An additional modelling method provided in XGBoost is called the Dart booster. The Dart booster was made with the idea that XGBoost combines a large number of trees with a small learning rate, which makes the algorithm consider trees added early significantly while trees added later are treated unimportant(12). The Dart booster, proposed by Vinayak and Gilad-Bachrach, adds techniques for dropping existing trees to change that balance, and reports better results in some situations. This method of boosting was also considered when fitting the model and finding the optimal modelling conditions.

3.2 Multi-Objective Optimisation – Genetic Algorithms (NSGA-II)

The purpose of this study was to find operating conditions which could optimise both the aerodynamic performance and manufacturing characteristics given a set of design specifications, then to do the same analysis when the spar positions of the rib changed. Based on these goals, the optimisation method selected was multi-objective optimisation (MOO) using genetic algorithms, which tries to find best-performing conditions for multiple objective functions simultaneously. The method involves finding the Pareto frontier through evaluating the solution from each simulation against another and taking the values that only reach the optimal. A brief explanation on the principles of genetic algorithms is explained below.

3.2.1 Genetic Algorithms in General

Genetic algorithms are algorithms which differ quite significantly from other ML algorithms in that:

- No input data goes into the algorithm
- The model itself is the prediction of the algorithm
- The prediction can be another algorithm

Genetic algorithms are mainly used when there are many possibilities to try and no mathematical algorithm to calculate the absolute optimum, or when the mathematical algorithm is too computationally expensive. Genetic algorithms require the information to be separable into building blocks(genes), and the information to be testable into better or worse information.
After defining the necessary input information and target objectives, genetic algorithms operate in the following steps(13):

1. It creates several random examples from the required information
2. It checks which of the samples perform the best according to a target objective
3. It generates new examples by using parts of the best examples

4. It mutates some of the examples by switching some of their building blocks with random ones

Then the algorithm repeats steps 2-4 until it reaches the target quality (in the case of MOO, the pareto front) or it runs out of time.

3.2.2 NSGA-II Algorithm

The NSGA-II algorithm is one of the most popular MOO genetic algorithms, due to its advantages over conventional multi-objective evolutionary algorithms in computational complexity, their non-elitism approach, and the need to specify a sharing parameter(14). NSGA-II has three special characteristics:

- Fast non-dominated sorting approach
- Fast crowded distance estimation procedure
- Simple crowded comparison operator

The NSGA-II roughly follows these steps:

1. Population initialisation based on problem range and constraint
2. Sorting process based on non-domination criteria of the initialised population
3. Assigning of crowding distance value (an estimate of density of solutions surrounding a particular solution)
4. Selection of individuals using binary tournament selection and crowded distance operator
5. Real coded genetic algorithm (GA) using simulated binary crossover and polynomial mutation
6. Recombination of offspring and current populations and repeat of selection

The specific code for implementing NSGA-II optimisation was done through the usage of Kanpur Genetic Algorithms Library (KanGAL) software library.

3.3 Parallel Coordinates System (||-coordinates)

The parallel coordinates system is a system developed by Professor Alfred Inselberg to aid visualisation and analysis of multi-dimensional space(15,16). For a given data or design space with N dimensions, he suggested plotting all N dimensions as a series of parallel axes, then plotting the values of each points for each dimension on the axes to visualise a point on the N-dimensional system(17). This method of geographical representation of multi-dimensional space has the ability to(18):

- Apply for any positive integer N variables
- Be the same for any N
- Have complexity of the representation increase linearly with N
- Give rise to a number of efficient geometrical algorithms

For the purposes of this study, parallel coordinates visualised through the Cambridge Advanced Modeller 2 software(19) were only used to help visualise and understand the relationships from the optimisation results, where different pareto points were observed.

4.0 Methodology

A diagram showing the approach taken for this study is shown below:

The same study approach was used twice for this study - once for doing optimisation without considering changes in rib design (i.e. without including spar positions as a variable) and once for optimisation with changes in design. This was because both study objectives had the same end goal of doing MOO to analyse the relationships between different involved variables, and the same process approach was required to meet that goal.

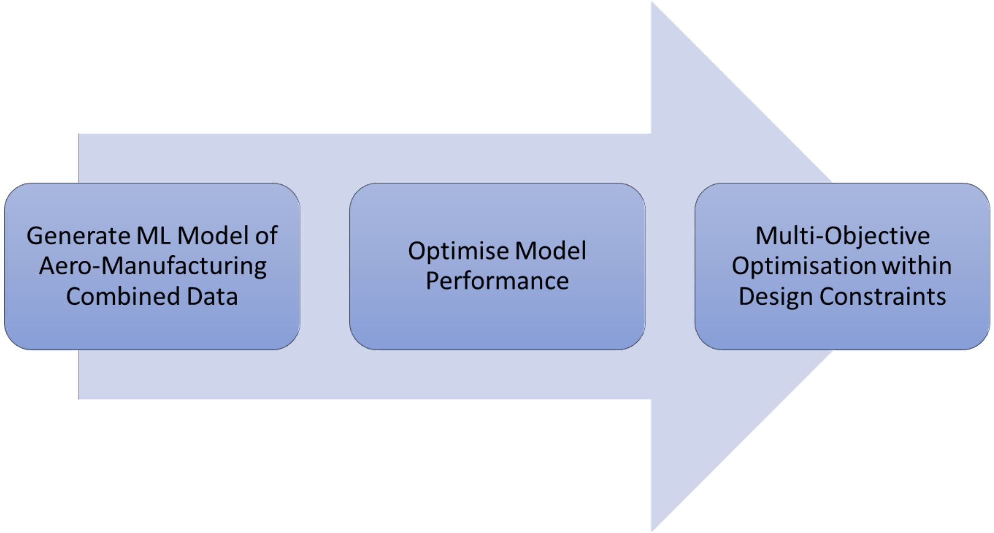
Generate ML Model of Aero-Manufacturing Combined Data
Optimise Model Performance
Multi-Objective Optimisation within Design Constraints

Figure 4. Diagram showing study approach

The first step of the approach was to generate a ML model of the combined aero-manufacturing dataset. This step needed to be completed first not only because it was related directly to one of the study objectives, but also because the successful prediction of targets within the given dataset was necessary to do MOO. MOO using the KanGAL software required the target variables, or fitness functions, to be expressible in terms of the input variables, which could be achieved by building an ML model which predicted the output target based on the chosen input variables. Only once this was accomplished to a satisfactory degree could the optimisation process begin.

4.1 Generation of ML Models

Before ML models could be generated, the provided data needed to be suitably filtered and processed. The data for modelling and optimisation without including spar positions was provided as a set of 3060 points on simulated aerodynamic and manufacturing data for a single pair of spar positions (Front Spar position 12%, Rear Spar position 65%). The extra data for additional spar positions, however, was only available for simulated aerodynamic data. The bump amplitude values were the same for all of the simulations, and there were manufacturing data for each given bump amplitudes available from the previous data, so the same manufacturing data was used for different spar positions when including the spar positions as the variables. The total number of datapoints for combined spar positions was 3060*9 = 27540 points.

In the initial data on aerodynamic performances and corresponding manufacturing characteristics, there were 21 different values and 2 objective functions. We decided that only several of the 21 values should be prioritised in the modelling process, as some of the values were dependent on other variables and time limitations on the study also needed to be considered. Thus, the data was filtered to include only the following variables:

- Alpha, meaning the angle of attack on the rib
- Mach number (Mach), representing the speed of airflow on the rib
- Drag Count, a measure for drag on the rib
- L/D, the ratio between lift generated by the rib and the drag it encounters
- Bump Amplitude (BumpAmp), a measure for the deviation in manufactured airfoil from nominal design
- Cpk, an index measuring the manufacturing capability
- Price of manufacturing a certain rib
- Manufacturing Process (MP) used (between 3 choices)
- Positions of front and rear spars (only for modelling and optimisation including spar positions)
- Mission Perfomrnace = L/D + 1/DragCount as one target objective
- Manufacturability = 1000/Price as the other target objective

It is important to note here that for the first study objective (modelling and optimisation without spar positions) the data for only a single value of spar positions was used. This was because the datset including spar positions created extra sets of data points for the same input variables which were unexplainable by variables other than changes in the spar position, and this duplicity of datapoints resulted in errors during the training procedure. The combined dataset combinations with spar positions included was only used for the second study objective.

The next step was to select the combination of input and output variables for the modelling process. In an ideal situation with sufficient time and resources, the modelling could be done for all combinations of input and output variables, and the performances of all models could be analysed. In the scope of this study, however, Cpk and L/D were not considered as an input variable because they acted as constraints in the optimisation problem, and thus needed to be modelled by other inputs in order to be able to construct the optimisation problems correctly. Mission Performance and Manufacturability were the target objectives, and by the same logic, Drag Count (which was used to calculate Mission Performance) and Price (which was used to calculate Manufacturability) were excluded from being inputs as well. This left alpha, Mach, BumpAmp, MP, FS (front spar position) and RS (rear spar position), which were used as the input variables for modelling.

Then the ML models could be generated with the chosen set of input and output variables. To do this, the data needed to be parsed into a format used internally by XGBoost in order to be processed. For training and testing the model, the provided data was split randomly (the weight of the split being 7:3 with training 7) into training and testing datsets every time a model training and prediction process was called, to prevent the problem of models overfitting to a specific training dataset. The resulting models were saved, and the values of predictions and the corresponding labels (true values) were saved separately as well.

4.2 Optimisation of Model Performance

The next step after modelling was to analyse the performance of the model and optimise the parameter inputs for the modelling process. XGBoost offers various parameters which influence the performances or even the type of model, to enable users to test a wider variety of conditions and tune the software for their specific purposes. For this study, there were four main parameters which were altered and compared for their performance:

- Booster, which decides the what kind of modelling XGBoost will perform – the three choices were gbtree (regression tree), gblinear (generalised linear regression), and dart. For dart boosting, the parameter drop_rate was set to 1, which makes it equivalent to random forests
- Num_rounds, which decides how many boosting rounds to do – this parameter was varied from 10 to 100 in spaces of 10
- Eta, which is the weight of the sum of new features against the existing tree features – this parameter was varied from 0.2 to 1 in spaces of 0.2
- Max_depth, which is the maximum depth of each tree (i.e. number of splits) – this was varied from 2 to 4

In order to evaluate the performance of these models, several methods and metrics were defined and used for evaluation of the models. The list of methods are shown below (the implementation of the evaluation metrics were imported from the sklearn.metrics library in Python):

- Visual evaluation – creating the scatterplot of predicted vs real values of a modelling result to see if the scatterplot was linear (see Figure 5 for more details)
- Mean Squared Error – calculating the mean squared error between prediction and real values to evaluate the absolute scale of error, is best when minimised
- % Accuracy – calculating the error between prediction and real values as a percentage of the corresponding real value, is best when minimised
- Adjusted R2 Score – calculating the adjusted r2 score for prediction vs real values to evaluate the degree of deviation between the two, is best when maximised (maximum is 1)

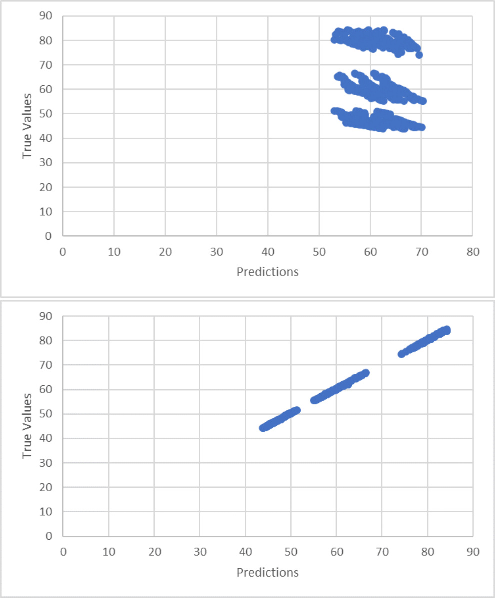

Figure 5. Comparison of bad scatterplot(above) and good scatterplot(below)

The three calculatable evaluation metrics were calculated for every modelling iteration, and the values of these metrics were recorded in a separate sheet for comparison. Based on the values of the scores, if they were good, the scatterplot of prediction against real values were evaluated to ensure that there were no wrong behaviours or unexpected errors which contributed to a good metric but bad model performance in real life.

It is worth noting that the goal of this parameter tuning was to find a set of working parameters which was able to model all necessary input and output variable combinations with high accuracy, not necessarily find the very best model performance for a single combination. This was due to the constraints on time and resources available for the study. An ideal set of parameters would be able to model for all combinations of inputs and outputs perfectly, but in exploring the parameters to find optimal values for the dataset itself, a set of model parameters which could predict without compromising accuracy significantly for any of the combinations could be considered a good set of parameters.

4.3 Multi-Objective Optimisation within Design Constraints

Once the model building process was complete, and the parameters for accurately modelling all necessary input and output combinations were found, the last step of the whole study was to run the MOO and analyse the results.

The constraints and the list of variables and objectives needed to be clearly defined before the optimisation could be run. The main target objectives, as explained previously, were mission performance and manufacturability. The problem space also had two main constraints:

- Limits on value of Cpk due to manufacturing constraints: 1.33<Cpk<2
- Limits to not harm aerodynamic performance: L/D > 64.325

Verified ML models for targets and constraints were translated into a format which could be incorporated into the nsga-ii software. This process was necessary because the software for XGBoost was in python, while the KanGAL software for nsga-ii was coded in c. This was achieved through parsing of the dump files generated from the selected models, which contained all of the decision criteria and weights in each tree, using an open-source library called m2cgen. This parsing was tested to prove it produced the same results as loading the model before it was incorporated into nsga-ii.

After loading all the constraints and objectives into the nsga-ii software, the limits on variables themselves were defined according to the maximum and minimum values present within the given dataset:

- 2.1<alpha< 2.4
- 0.71<Mach<0.74
- 0.32<BumpAmp<2
- MP = 1,2,3
- 0.1<FS<0.14
- and 0.63<RS<0.67 (FS and RS are in decimals as the values are position of spars along the rib in % of total length)

The MOO also required setting of other variables, such as population size and number of generations. The population size and number of generations was set after a first-round MOO showed that running the MOO for unnecessarily large number of generations resulted in the objective function being stuck to a local minima, which did not always prove to be on the pareto front. As such, the values of population size and number of generations were set to 50 and 35 respectively after optimisation history showed the values of MOO converged at that point. Effects of other indexes in the nsga-ii software were not explored due to time constraints, and the indexes were set to fixed values. The list of indexes and their values are as follows:

- cross-over probability (0.5)
- mutation probability for real-coded vectors (0.1)
- distribution index for real-coded crossover (50)
- distribution index for real-coded mutation (250)

The limits on real-coded values were set to be rigid in the optimisation.

The MOO was conducted multiple rounds under same conditions to generate more population of results, and the results were compiled together to create the complete history and the pareto fronts of the optimisation. These pareto fronts were further analysed using Cambridge Advanced Modeller (CAM) software, and was mainly used in this instance to generate the parallel coordinates plots of the pareto points.

5.0 Results and Discussion

5.1 First Study: Modelling and Multi-Objective Optimisation without Spar Positions

5.1.1 ML Modelling of Objective Variables

The first ML modelling was predicting the objective functions (Mission Performance and Manufacturability) from the input variables alpha, Mach, BumpAmp and MP. The results of the modelling are shown below. The 5 models with best metrics are shown in Table 1.

Table 1
Evaluation Metrics for ML Models of Mission Performance

max_depth	Eta	booster	num_round	mean_accuracy	mean_squared_error	adjusted_r2
4	0.8	gbtree	100	99.84773	0.01987	0.999896
4	1	gbtree	100	99.84465	0.020348	0.999893
4	1	gbtree	90	99.83748	0.021372	0.999888
4	0.8	gbtree	90	99.83984	0.021375	0.999888
4	0.8	gbtree	80	99.83286	0.023161	0.999879

The best performing set of parameters are shown to be max_depth = 4, eta = 0.8, booster = gbtree, num_round = 100, and the metric values are: mean_accuracy = 99.847%, mean_squared_error = 0.01987, and adjusted r2 score nearly 1. The resultant scatterplot of predictions against real values for this set of parameters is shown in Figure 6 below.

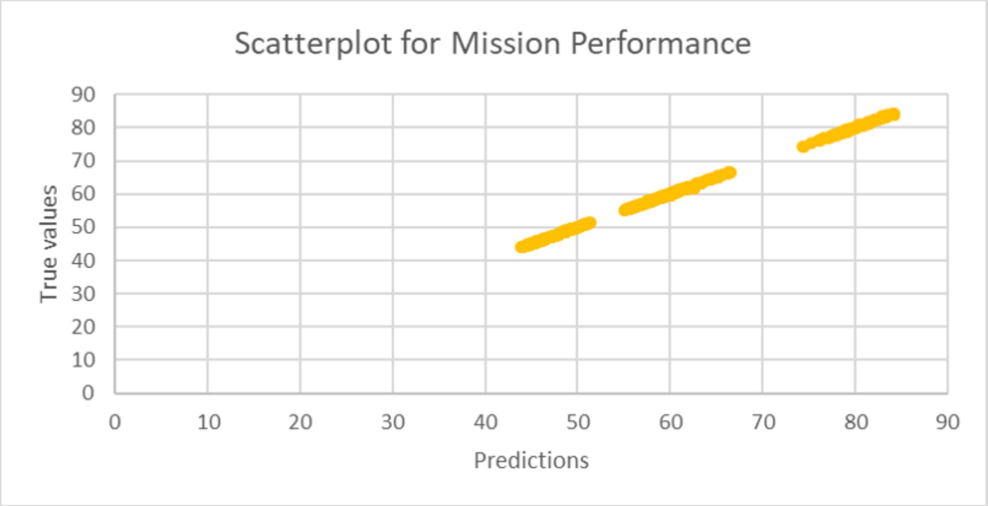

Figure 6. Scatterplot of Predictions against True values for Mission Performance without spar

As it can be seen from the, the prediction values match the true values very well. It can be said that this set of modelling parameters performed very well in predicting the Mission Performance correctly.

Similar results are shown for predicting Manufacturability with the given inputs:

Table 2
Evaluation Metrics for ML Models of Manufacturability

max_depth	Eta	booster	num_round	mean_accuracy	mean_squared_error	adjusted_r2
4	0.2	gbtree	100	99.8257	3.37E-07	0.999972
4	0.2	gbtree	70	99.82569	3.37E-07	0.999972
4	0.2	gbtree	60	99.82563	3.37E-07	0.999972
4	0.2	gbtree	50	99.82506	3.38E-07	0.999972
4	0.2	gbtree	40	99.80524	3.78E-07	0.999969
4	0.8	gbtree	100	99.68652	9.13E-07	0.999926

In Table 2, the values evaluation metrics for the best performing conditions in mission performance was included to see the difference from the best results, and it can be seen that the performance of the model varies very little (approximately 0.15% in mean accuracy and almost no difference in adjusted r2 score). To verify whether Manufacturability could also be predicted with the best performing conditions for Mission Performance (from this point onwards, these conditions will be referred to as the “base set of parameters”), the scatterplot for Manufacturability for the same modelling conditions is shown in Figure 7below.

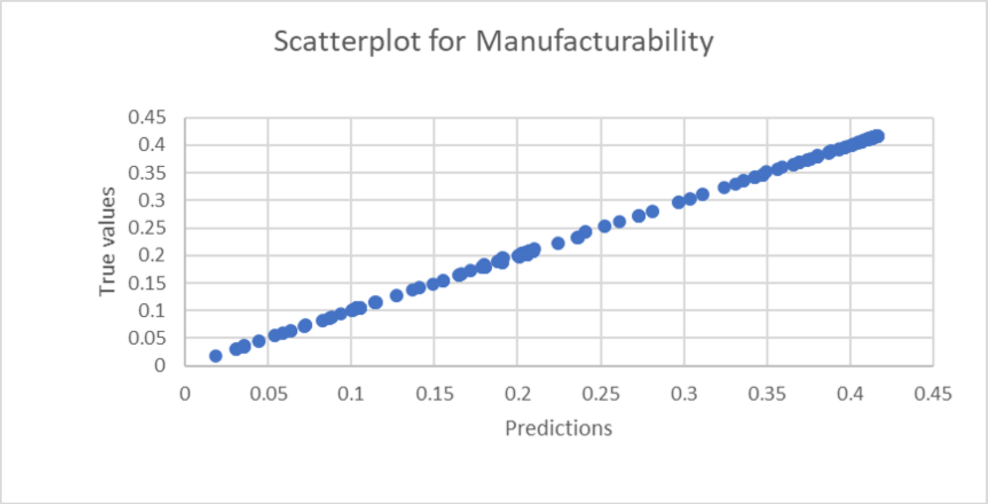

Figure 7. Scatterplot of Predictions against True values for Manufacturability without spar

It can be seen that the base set of parameters, which gave the best model performance for Mission Performance, also gave very good results for predicting Manufacturability.

5.1.2 ML Modelling of Constraint Variables

The next stage in ML modelling was modelling the constraints (Cpk and L/D) using the chosen inputs (alpha, Mach, BumpAmp and MP). The table of metrics for best-performing parameters for predicting L/D is shown in Table 3 below.

Table 3
Evaluation Metrics for ML Models of L/D

max_depth	Eta	booster	num_round	mean_accuracy	mean_squared_error	adjusted_r2
4	1	gbtree	100	99.82969	0.022305	0.999883
4	0.8	gbtree	100	99.83319	0.023907	0.999875
4	1	gbtree	90	99.82092	0.024822	0.99987
4	0.8	gbtree	90	99.82573	0.025506	0.999866
4	0.6	gbtree	100	99.81725	0.027086	0.999858

It can be seen that the base set of parameters are present as well. Plotting the scatterplot of predictions against real values for L/D of the model generated with base set of parameters is provided below in Figure 8:

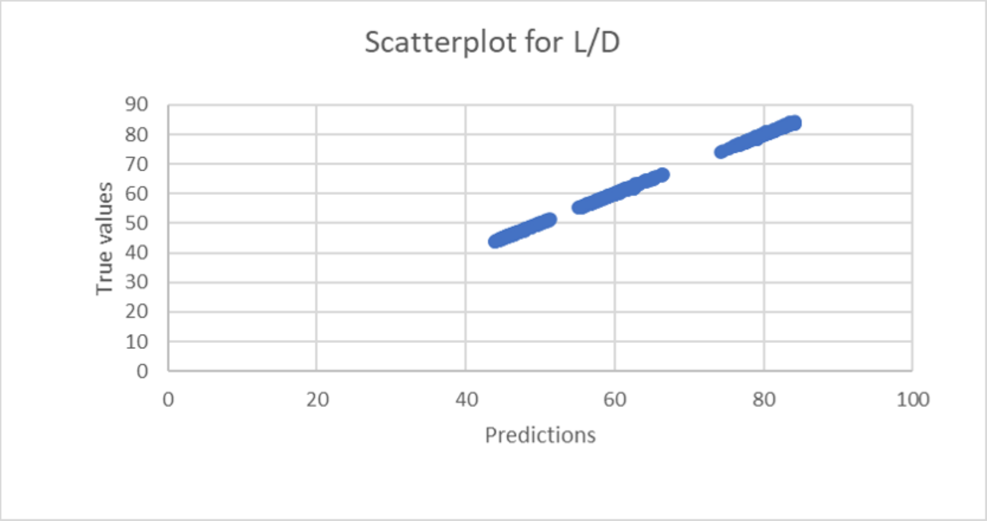

Figure 8. Scatterplot of Predictions against True values for L/D without spar

It can be seen from the scatterplot that the base set of parameters perform very well in predicting values of L/D as well.
Lastsly, the table for models with top evaluation metric values for predicting Cpk is shown below in 4:

Table 4
Evaluation Metrics for ML Models of Cpk

max_depth	Eta	booster	num_round	mean_accuracy	mean_squared_error	adjusted_r2
4	1	gbtree	100	98.48402	8.67E-06	0.999999
4	1	gbtree	90	98.31495	1.41E-05	0.999998
4	1	gbtree	80	98.18123	2.03E-05	0.999998
4	0.6	gbtree	100	98.20436	2.69E-05	0.999997
4	0.8	gbtree	100	98.20258	3.43E-05	0.999996

The base set of parameters can be seen here as well, which indicates that models using the base set of parameters may be able to predict all of the constraints and objectives well. To test this, the scatterplot of predicted vs real values for models predicting Cpk using base set of parameters is shown in Figure 9 below.

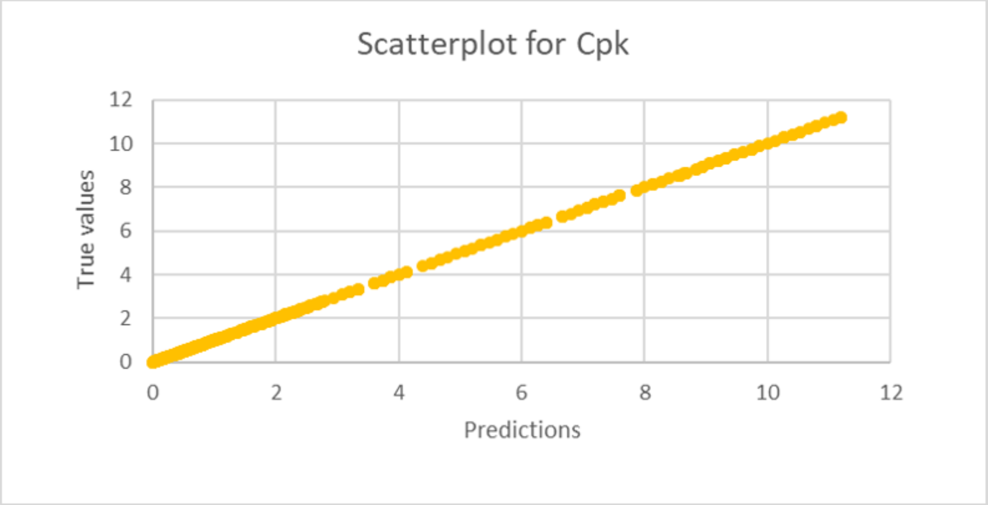

Figure 9. Scatterplot of Predictions against True values for Cpk without spar

It can be seen from the scatterplot that the model with base set of parameters performed very well in predicting Cpk. This proved that the base set of parameters may be used in modelling all of the constraints and objective functions when modelling without considering spar positions as an input variable.

5.1.3 Multi-Objective Optimisation Results

The next step was to run the Multi-objective optimisation using the trained ML models to find the trade-off relationships. The MOO was conducted for 10 repetitive iterations under same optimisation conditions to collect more data points before the whole collected history was filtered to find the pareto front. The whole history and resulting pareto front of the MOO without spar positions is shown in Figure 10 below:

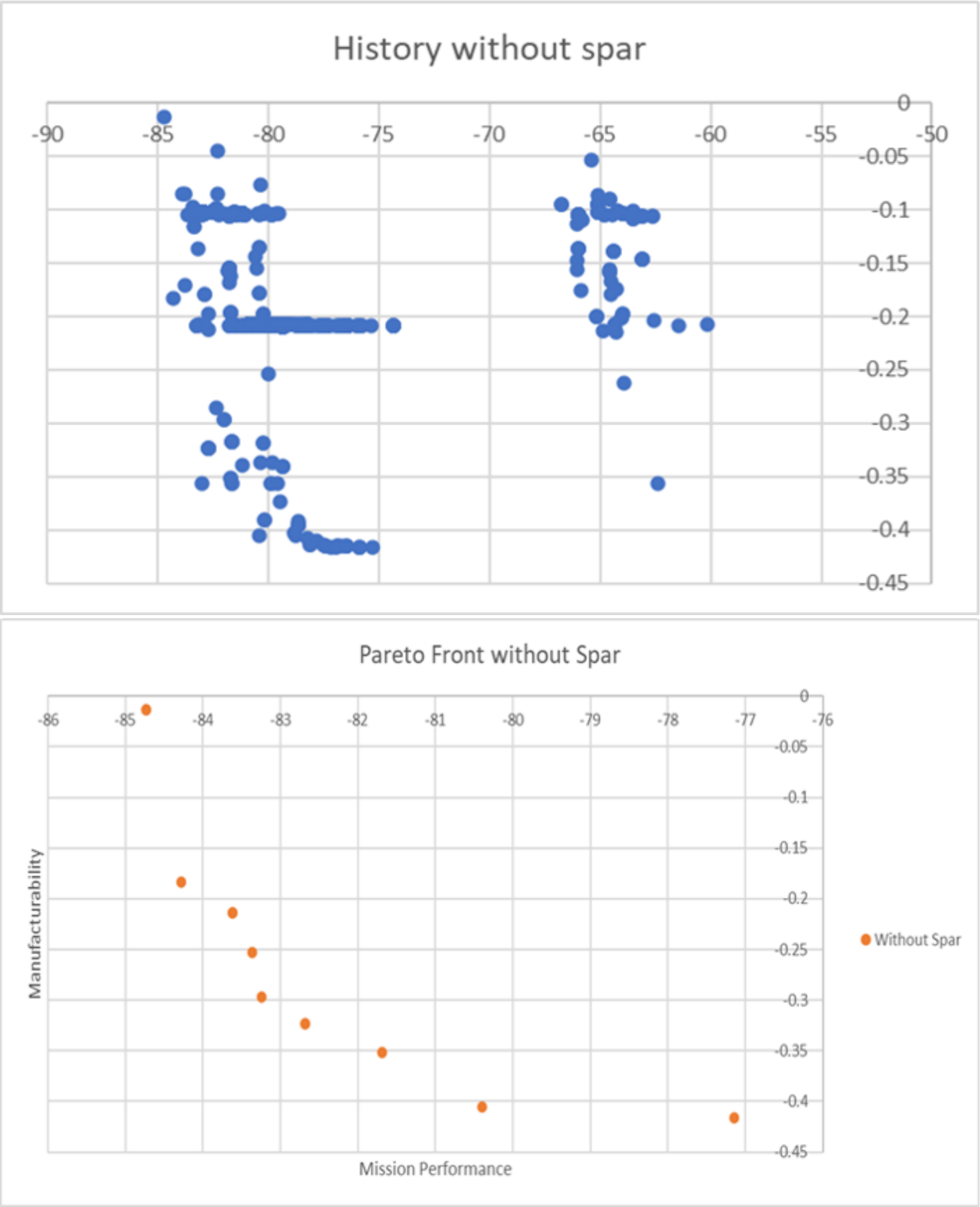

Figure 10. Complete history (above) and resulting pareto front(below) for MOO without spar positions

The values of Mission Performance and Manufacturability are shown as negative due to the nature of the NSGA-II software, which required the definition of objective functions as minimisation problems. Thus, the ideal front of this graph is towards the bottom-left, where the absolute values of Mission Performance and Manufacturability are at maximum.
From the pareto front it can be seen that there is a clear trade-off relationship between the two objectives, which was something to be expected. It can also be seen that the intermediate

values between the two extremes could give reasonably high values for both target objectives. To analyse the pareto front more effectively, the points on the pareto front were plotted on the parallel coordinates system (see Figure 11).

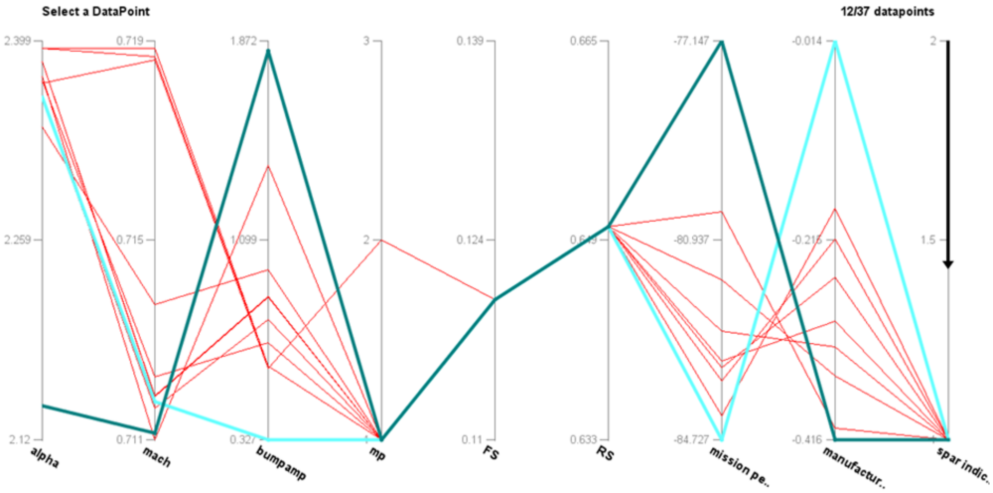

Figure 11. Parallel Coordinates Plot of Pareto Front without Spar

On the graph, the spar index is an arbitrary variable which indicates whether the values shown in the graph were optimised with spar positions or not. Also, the axes for the target objectives are negative - the higher the value of the two target objectives are, the lower on the graph they are located.

There were several things to note from this parallel coordinates plot. First, the trade-off between mission performance and manufacturability is visible on the plot as well, shown by the series of intersecting lines between the two axes, and it could be seen that there are choices which have reasonably good values for both target objectives. Also, the highest observed value of manufacturability in the pareto front was 0.416 (highlighted in dark green), which corresponded to a price of £2404 per rib. The extreme values for each target objective are hightlighted in dark green and sky blue, and it can be seen that the bump amplitudes for those values vary massively, which could be an explanation for the deviation as bump amplitude can be seen a measure of the manufacturing precision. It was also noticeable that almost all of the optimal operating conditions were focused on high values of alpha while operating conditions for Mach number were clustered around two values – 0.7187 and 0.7118.

It can also be seen that almost all optimal manufacturing conditions indicated Manufacturing Process 1 – there were no points which considered manufacturing process 3 as a choice. Lastly, comparison of two points with the nearly the same bump amplitude but different manufacturing process showed that manufacturing process 2 had higher mission performance but lower manufacturability, and had a lower alpha as its operating condition, as shown in Figure 12.

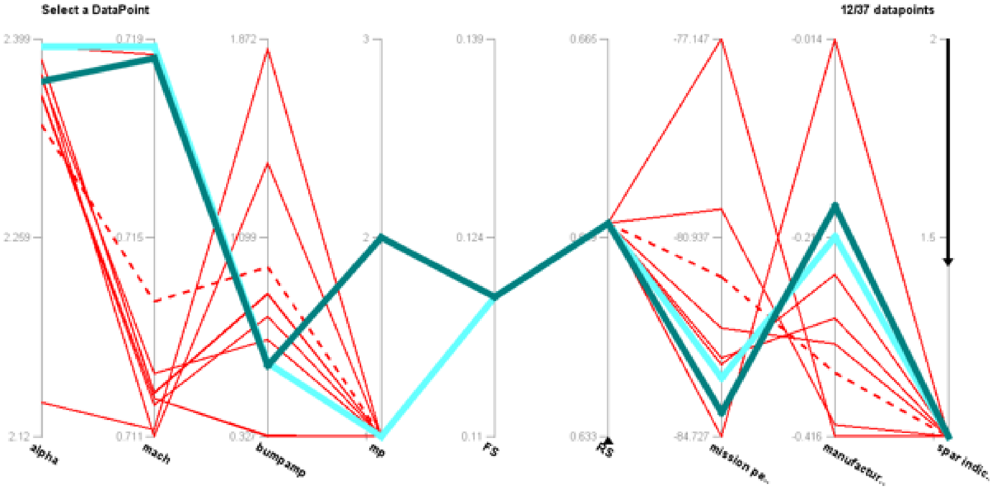

Figure 12. Comparison between two manufacturing processes for MOO without spar

5.2 Second Study: Modelling and Multi-Objective Optimisation including Spar Positions

The ML modelling results of objectives and constraints with spar positions were analysed in the same way as the modelling without spar positions were.

5.2.1 ML Modelling of Objective Variables

The table of evaluation metrics for modelling Mission Performance with the chosen inputs including spar positions is shown below:

Table 5
Evaluation Metrics for ML Models of Mission Performance with Spar

max_depth	Eta	booster	num_round	mean_accuracy	mean_squared_error	adjusted_r2
4	0.8	gbtree	100	99.57458	0.095325	0.999108
4	0.8	gbtree	90	99.56337	0.101286	0.999053
4	0.8	gbtree	80	99.54838	0.108401	0.998986
4	0.6	gbtree	100	99.55882	0.108787	0.998982
4	0.6	gbtree	90	99.55114	0.113409	0.998939

It is worth noting that the best performing parameters for predicting Mission Performance with spars is the same as those for predicting without – namely, the base set. The scatterplot of prediction against real values for this base set model is shown below (Figure 13).

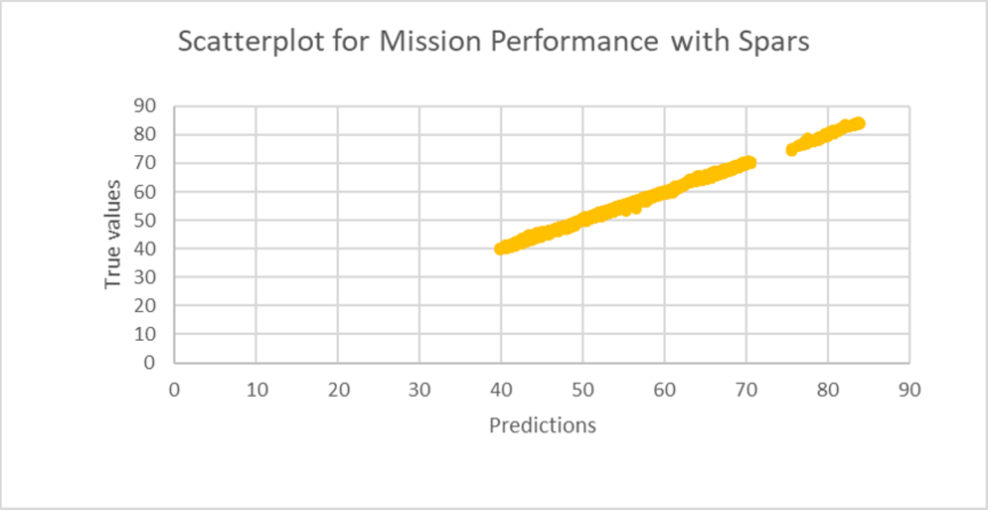

Figure 13. Scatterplot of Predictions against True values for Mission Performance with Spar

It can be seen that the scatterplot also indicates a good performance of the model, and that this base set is an acceptable set of parameters for predicting Mission Performance with spar positions as well.

Next was modelling Manufacturability with spar positions. The table of evaluation metrics is shown below:

Table 6
Evaluation Metrics for ML Models of Manufacturability with Spar

max_depth	Eta	booster	num_round	mean_accuracy	mean_squared_error	adjusted_r2
4	0.2	gbtree	100	99.94183	5.49E-08	0.999996
4	0.2	gbtree	90	99.94183	5.49E-08	0.999996
4	0.2	gbtree	80	99.94182	5.49E-08	0.999996
4	0.2	gbtree	70	99.94181	5.49E-08	0.999996
4	0.2	gbtree	60	99.94173	5.49E-08	0.999996
4	0.8	gbtree	100	99.85437	1.89E-07	0.999986

The evaluation metrics for base set is shown here as well, and it can be seen that the difference in accuracy is even smaller than it was for the case of predicting Manufacturability without spars using base set of parameters. The scatterplot for this model is shown below (Figure 14):

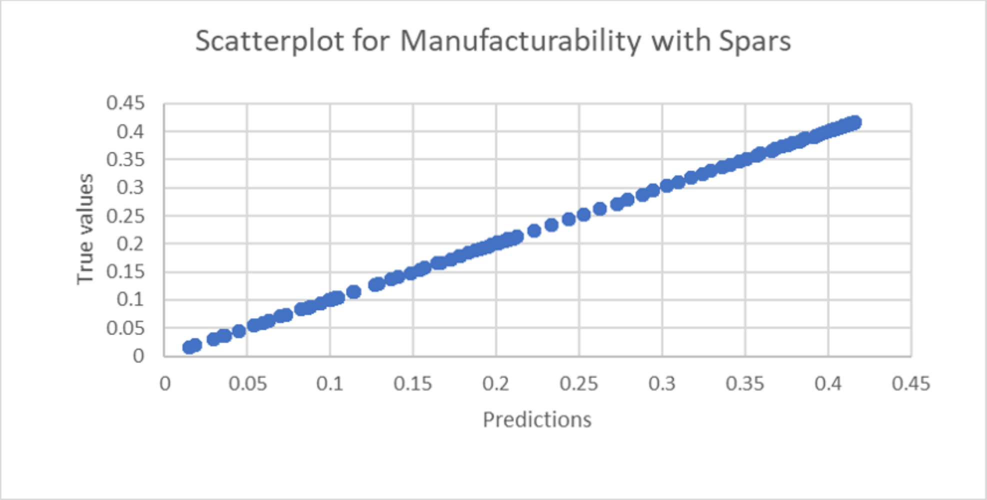

Figure 14. Scatterplot of Predictions against True values for Manufacturability with Spars.png

It can be seen that the base set of parameters also performed well in the predictions.

5.2.2 ML Modelling of Constraint Variables

The table of evaluation metrics for modelling L/D with spars included in input variables is shown below:

Table 7
Evaluation Metrics for ML Models of L/D with Spar

max_depth	Eta	booster	num_round	mean_accuracy	mean_squared_error	adjusted_r2
4	0.8	gbtree	100	99.57458	0.095296	0.999108
4	0.8	gbtree	90	99.56337	0.101255	0.999053
4	0.8	gbtree	80	99.54839	0.108368	0.998986
4	0.6	gbtree	100	99.55881	0.108756	0.998982
4	0.6	gbtree	90	99.55114	0.113377	0.998939

Table 7 also shows the base set of parameters as the best-performing of all tested combinations of model parameters, with over 99.5% mean accuracy of prediction values. The corresponding scatterplot of the model is shown below (Figure 15):

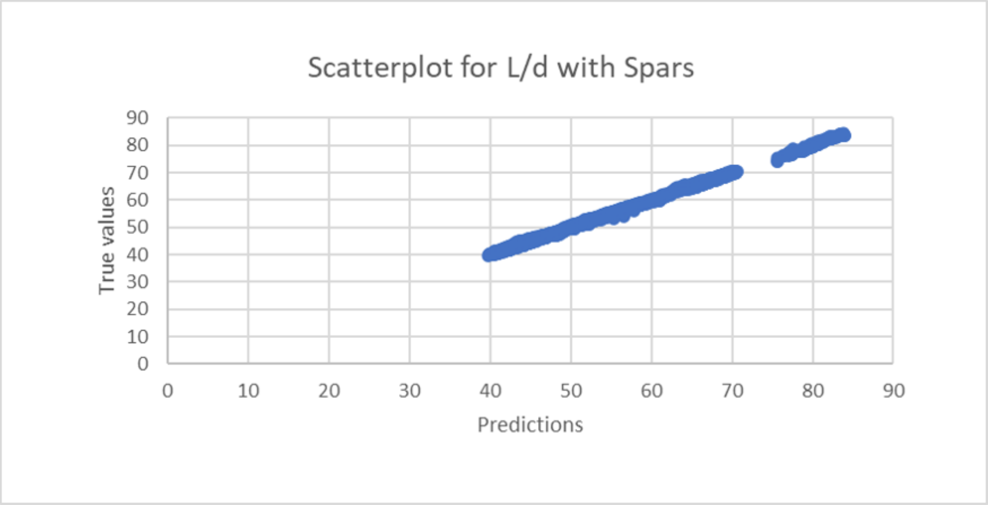

Figure 15. Scatterplot of Predictions against True values for L/D with Spars

The scatterplot also indicates a good performance of the model in predicting L/D with spar positions as inputs.

Finally, ML modelling was done for predicting Cpk from the given inputs including spar positions. The resulting table of evaluation metrics is shown below:

Table 8
Evaluation Metrics for ML Models of Cpk with Spar

max_depth	Eta	booster	num_round	mean_accuracy	mean_squared_error	adjusted_r2
4	1	gbtree	100	98.32678	6.02E-07	1
4	1	gbtree	90	98.29662	9.55E-07	1
4	0.8	gbtree	100	98.23349	2.89E-06	1
4	1	gbtree	80	98.23399	3.90E-06	1
4	0.8	gbtree	90	98.19224	5.05E-06	0.999999

It is necessary to mention that the adjusted r2 values in the table, shown as 1, is not actually 1.0 – rather, it is shown that way due to the number of significant digits. But the model with base parameters still seems to perform very well according to the evaluation metrics.

The scatterplot of the model with base parameters is shown below (Figure 16):

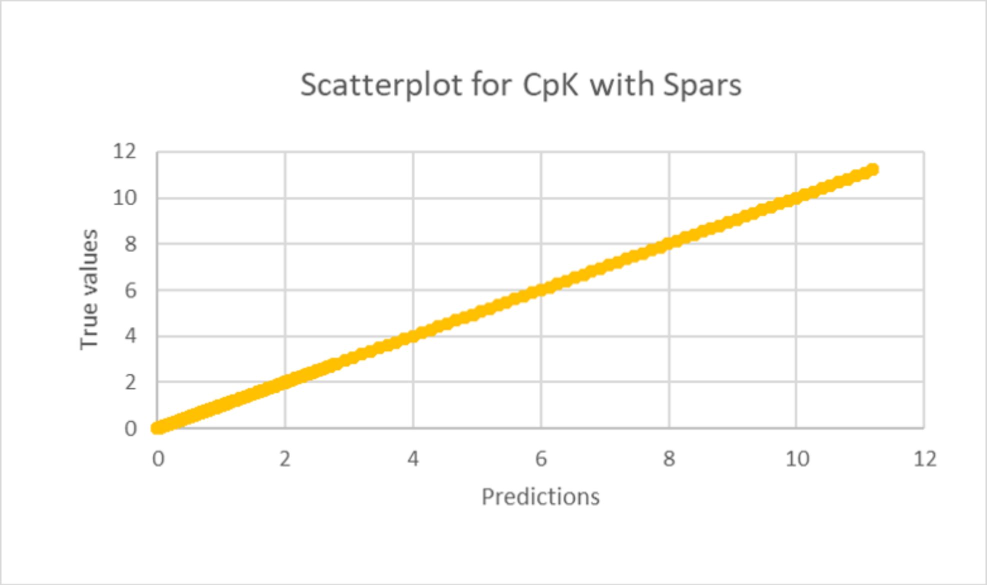

Figure 16. Scatterplot of Predictions against True values for Cpk with Spars

The scatterplot also indicates a good model performance using the base set of parameters. Based on the results of all of the ML model evaluations shown above, the base set of parameters – max_depth = 4, eta = 0.8, num_rounds = 100, booster = gbtree – was set as the final modelling set of parameters for all modelling processes to be used in the optimisation process.
Other noteworthy trends in the evaluation metrics observed during running different modelling scenarios included:

- out of the three boosting methods, gbtree performed the best
- increasing number of boosting rounds increased the accuracy of prediction, although it was seemingly more effective for increasing the accuracy in Mission Performance than Manufacturability
- Increasing max depth improved the accuracy of the models

It may be noticeable that there are potential improvements in evaluation metrics from the base set of parameters for the targets or constraints. However, the difference in performance varied very little, and as the goal of the parameter tuning was to find a set of parameters that could perform well in all combinations, the difference in performance was not analysed closely in this study.
It is worth mentioning again at this point that the full dataset was split randomly into training and testing dataset before each model was trained to prevent the potential problem of overfitting. The same training was also tested several times, and each time the models performed similarly well in predicting the data (with different randomised training datasets).

5.2.3 Multi-Objective Optimisation Results

The same method of analysis was used for MOO with spar positions as well, and the two results were compared against each other to see the effects of including spar positions in the optimisation. As before, the ML models with base parameters were selected as the objective and constraints for the MOO algorithm, and the MOO was re-iterated 10 times to collect more data points for the pareto front. The resulting complete history and pareto front are shown below (Figure 17).

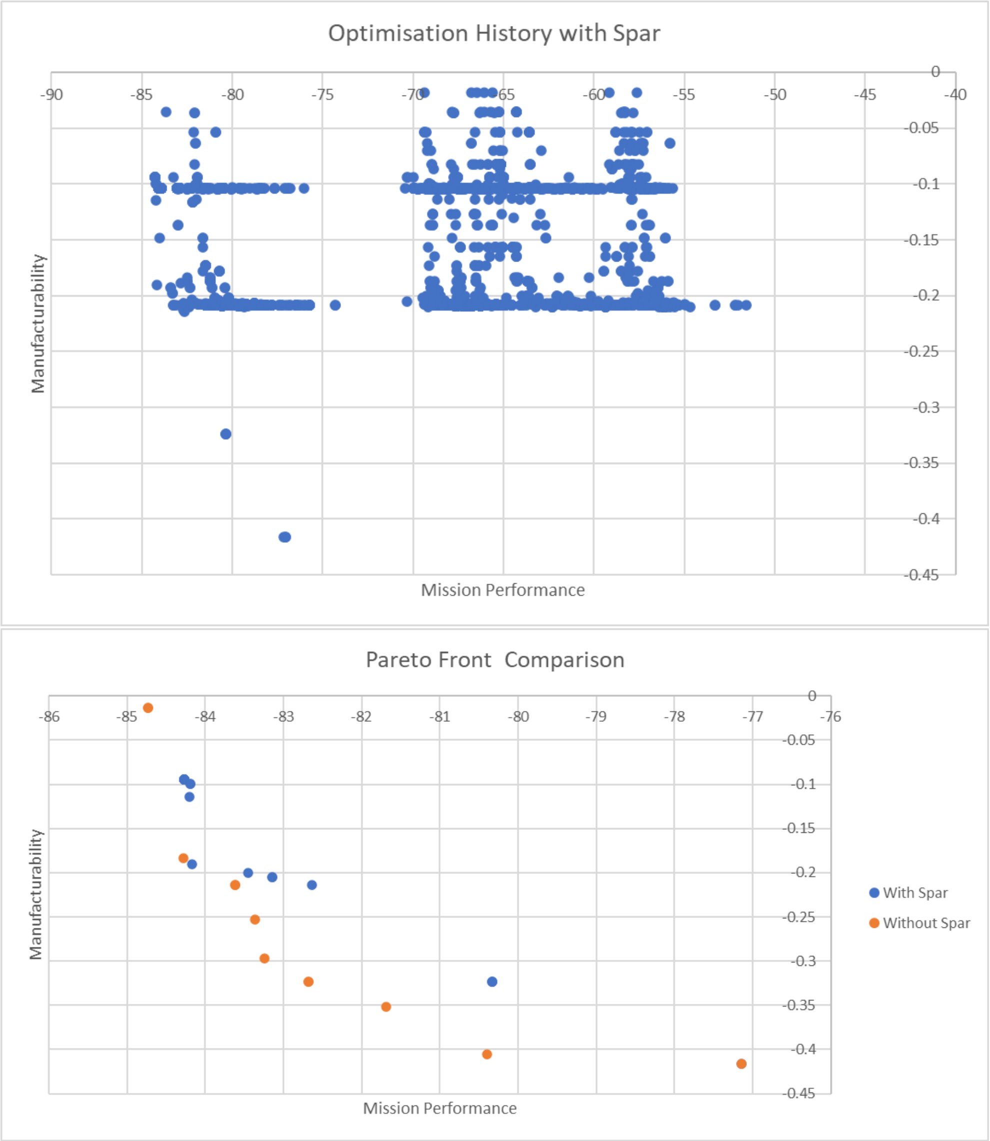

Figure 17. The complete history (above) for MOO with spar positions and the Pareto plot (below) for MOO with and without spar

One thing to note in the complete history plot is the difference in scale of the axes – in the complete history without spar, the smallest values for mission performance started around 60, whereas for history with spar the starting values are lower, and goes down to nearly 50.
Looking at the pareto fronts plotted together, one thing is clearly visible: the pareto fronts are located at different positions. The pareto front for MOO with spar is located slightly more to the upper-right hand side. This change in the position of the pareto front is a positive outcome, as it indicates that including the spar positions as a variable created a difference in what kind of optimal trade-off relationships are available between the two target objectives.
To analyse the pareto front generated by MOO with spar positions further, the pareto points were plotted on the parallel coordinates (see Figure 18).

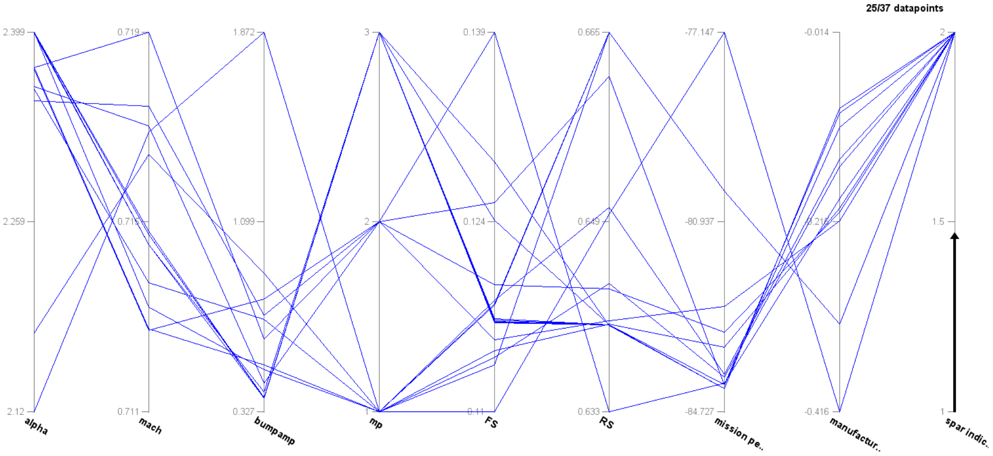

Figure 18. parallel coordinates plot of MOO with spar

One of the things immediately noticeable from the graph is that MOO with spar positions offers manufacturing process 3 as a viable choice. In the MOO without spar positions it was seen that manufacturing process 3 was not considered at all. This is another potential benefit of including spar positions as a variable, as it gives the rib design more flexibility in which manufacturing process to choose from.
The trade-off between Mission Performance and Manufacturability is also observable here, but it is interesting to note that the values of Mission Performance and Manufacturability seem to be more clustered around specific values than they were in the case of MOO without spars. There are a lot of values which have similar Mission Perfomances but varying Manufacturabilities, which was something that was not observed before in the MOO without spar, where the Mission Performance was more widely varied. This also may indicate a positive of including spar positions in MOO, as it seems to succeed in increasing the manufacturability for a same given value of mission performance through combination of other variables available, which was something that was not observed in MOO without spars.
To compare both MOOs more equally, both parallel coordinates plots were filtered then plotted on the same graph. The filters applied were:

- Limits on minimum Mission Performance at 80
- Limits on minimum Manufacturability at 0.17

- Limits on maximum BumpAmplitude at 1.5

The resultant graph is shown below in Figure 19:

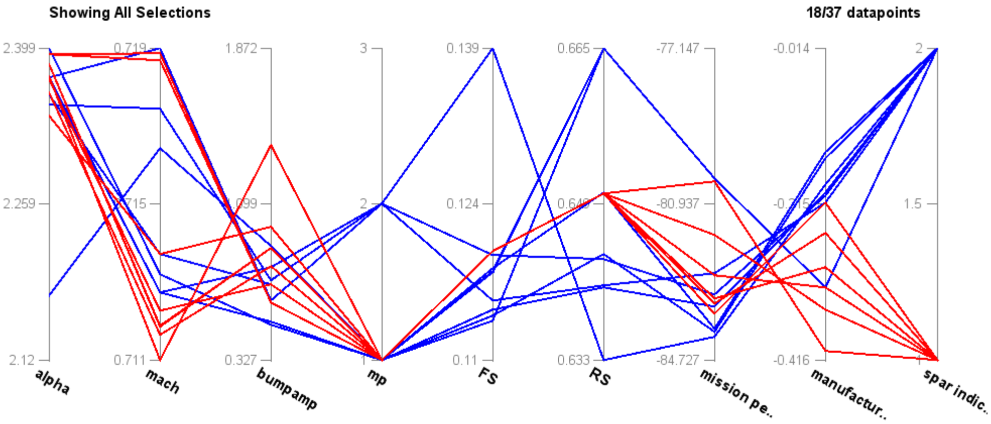

Figure 19. comparison of moo with and without spar positions

The range of values for the operating conditions in within these limits are summarised in the table below.

Table 9
Comparison of ranges in operating conditions for pareto points

MOO without spar		**MOO with spar**	
Variable	**Range**	**Variable**	**Range**
Alpha	2.338 2.394	Alpha	2.177 2.371
Mach	0.7117 0.7189	Mach	0.7128 0.7176
Bump Amp	0.611 1.39	Bump Amp	0.7 0.89
MP	1	MP	1,2
FS	0.12	FS	0.1135 0.1196
RS	0.65	RS	0.6403 0.6648

The comparison shows that MOO with spar position still offers more flexibility in manufacturing processes than MOO without spar position does. MOO with spars is more centered around a small range of bump amplitudes that are also lower than the values for MOO without spar positions, which could be interpreted as a positive result in terms of aerodynamic performance as it means the error between the nominal and manufactured airfoil will be smaller and more controlled. The difference in Manufacturability suggests that optimisation without spar positions yielded in more low-cost solutions.

It was also observed, however, that small changes in just the spar positions affected the mission performance and manufacturability very little. Figure 20 highlights two different points with slightly different spar positions and nearly identical values for all other factors. This may potentially be due to the presence of less-dominating points in the pareto front with spar positions, or insufficient data points to analyse the relationship to a satisfactory level of confidence.

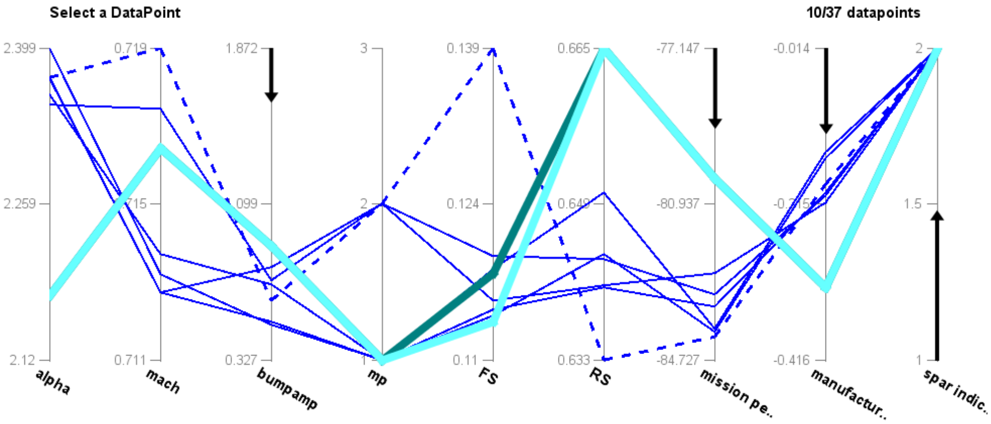

Figure 20. two points (dark green and sky blue) with different front spar positions

6.0 Conclusion and Further Considerations

6.1 Study Conclusions

The main objectives of the study were:

- Building a ML model which could predict aero-manufacturing combined data accurately
- Performing Multi-Objective Optimisation with and without the influence of spar positions to find the pareto fronts and analysing the effects of spar positions on the optimisation

The study was a success in building a ML model for the combined aero-manufacturing data utilising XGBoost. The obtained model could predict the combined aero-manufacturing data with high accuracy and reliability, and verified the potential of combining the distant areas of mission performance and manufacturability to consider both factors at the same time. The MOO without spar positions as a variable found a visible trade-off region between high manufacturability and high mission performance, and offered choices of different combinations of manufacturability and mission performance which can be adjusted according to different needs.

Comparison of MOO with and without spar positions resulted in a trade-off relationship between mission performance and manufacturability which was different from the trade-off relationship found in MOO without spar positions. The difference in trade-off relationships showed that the inclusion of different variables in the modelling had the potential to find relationships that reflected the real world more accurately. It was also worth noting that the MOO with spar positions offered more flexibility in the choice of manufacturing processes available for a given design.

There was no clearly identified correlation between a small change in spar position and the optimal values for the target objectives; however, the range of bump amplitudes for solutions found by MOO with spar positions was lower and more focused than those found by MOO without spar positions.

6.2 Further Considerations

There are also a number of things to be noted as potential future directions of this study.

First, while the study results suggested a meaningful results in modelling of the combined data, a further study on the optimisation with more repetitions and sufficient data could help add confidence to the findings of this study.

Also, there is the inherent uncertainty in the true "continuity" of variables analysed in this study to consider. For example, the locations of spar positions, considered to be truly continuous for this study context, can be subject to physical limitations in manufacturing or assembly. Exploration of this uncertainty can help the results of the study reflect the real world conditions better.

Last but not least, there are numerous other variables not considered in this study that could help build a more accurate model. For example, the necessary shape of the rib and its aerodynamic performance changes according to the position of a rib down the length wing – considering the changes in rib shape and performance according to its position along the wing as additional variables may provide richer interpretation of the design space and new opportunities for improvement.

7.0 Acknowledgements

The research leading to these results has received support by members of the consortium of the Advanced Product Concept Analysis Environment (APROCONE) project (Ref no. 113092) which was funded by the Aerospace Technology Institute (ATI) in the UK.

List of Figures

REFERENCES

1. A. Jameson. Successes and challenges in computational aerodynamics. *8th Computational Fluid Dynamics Conference*, 1987.
2. A. Jameson. A perspective on computational algorithms for aerodynamic analysis and design. *Progress in Aerospace Sciences*, 37(2):197–243, 2001.
3. R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.
4. M. Maginness, E. Shehab, C. Beadle, and M. Carswell. Principles for aerospace manufacturing engineering in integrated new product introduction. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 228(7):801–810, 2013.
5. E. Rebentisch, W. Myles Murman, and M. Earll. Challenges in the better, faster, cheaper era of aeronautical design, engineering and manufacturing. *Massachusetts Institute of Technology. Engineering Systems Division*, March 2000.
6. V. V. S. Nikhil Bharadwaj, P. Shiva Shashank, M. Harish, and P. Garre. A review on lean manufacturing to aerospace industry. *International Journal of Engineering Research and General Science*, 3(4), July-August 2015.
7. V. Crute, Y. Ward, S. Brown, and A. Graves. Implementing lean in aerospace—challenging the assumptions and understanding the challenges. *Technovation*, 23(12):917–928, 2003.
8. M.M. Longato. Support an s-duct optimization design study using state-of-the-art machine learning techniques. Master's thesis, Universita degli Studi di Padova, Apr 2020.
9. R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd international conference on Machine learning - ICML 06*, 2006.
10. T. Chen and C. Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
11. Introduction to boosted trees. `https://xgboost.readthedocs.io/en/latest/tutorials/model.html`. Accessed 27 April 2020.
12. Xgboost booster: gbtree v.s. dart v.s. gblinear. `https://medium.com/@xzz201920/xgbosst-booster-gbtree-v-s-dart-v-s-gblinear-82d8fcbb07d2`, Feb 2020. Accessed 7 June 2020.
13. W. Darrell. A genetic algorithm tutorial. *Statistics and Computing*, 4(2), 1994.
14. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
15. A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.
16. A. Inselberg. Don't panic . . . just do it in parallel! *Computational Statistics*, 14(1):53–77, 1999.
17. J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *IEEE Transactions on*

Visualization and Computer Graphics, 15(6):1531–1538, 2009.

18. A. Inselberg. *Parallel Coordinates Visual Multidimensional Geometry and Its Applications*. Springer Verlag, 2009.

19. W. Piotrowski, T. Kipouros, and P.J. Clarkson. Enhanced interactive parallel coordinates using machine learning and uncertainty propagation for engineering design. *IEEE eScience*, 2019.