33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

SUCCESSIVE KNOWLEDGE BUILDUP AND TRANSFER DURING THE ANALYSIS OF SCALED UAV WINGS BY MEANS OF MACHINE LEARNING

Tim Klaproth[1] & Mirko Hornung[1]

[1]Chair of Aircraft Design, Technical University of Munich, Germany

Abstract

An omnipresent conflict in aircraft design optimization is the need for fast yet accurate analysis tools. For a broad search in a design space, a trade-off is required. One approach is to perform the broad search using low-fidelity methods and to perform higher-fidelity calculations only at a few design points. The knowledge gained at these higher-fidelity data points can be formulated as correction factors which are to be applied to the low-fidelity methods to approximate the higher-fidelity results. This paper explores the potential to transfer this knowledge across the entire design space considered, focusing on aerodynamic calculations and mass estimates of a scaled UAV wing. The paper presents curve fits which reveal relationships between the correction factors and specific aircraft parameters. On top of that, adaption-based multi-fidelity modeling with a successively increasing number of higher-fidelity samples is applied, in order to explore the potential for automatic, successive knowledge build-up and transfer. The results of the study indicate the feasibility of gradual knowledge build-up and transfer for the considered correction factors. Predictions seem to be possible with high accuracy already on the basis of a few higher-fidelity data points. Furthermore, the study addresses two issues in the context of the application of multi-fidelity methods to the problem at hand. The first issue is that, to date, there is a lack of designated semi-empiric wing mass estimation methods for the UAV class in question. The study uses semi-empiric formulas from other aircraft classes and demonstrates that the associated correction factors can be related to specific aircraft parameters. The second issue refers to the accuracy of the higher-fidelity model. Typically, multi-fidelity models try to approximate the higher-fidelity solution as accurate as possible. However, the "higher-fidelity" model itself is again only an estimation of the real behavior, and may yield physically inexplicable results. This paper presents an approach, which does not approximate the higher-fidelity aerodynamic data itself. Instead, it approximates a fit of the higher-fidelity aerodynamic polar, which incorporates a-priori knowledge about the expected shape of the polar.

Keywords: UAV, aerodynamics, wing mass estimation, multi-fidelity, machine learning

1. Introduction

Aircraft design optimization is a very time-consuming task as it involves a high number of system model evaluations. This requires fast system models. Typically, low-fidelity methods are used which are often semi-empiric estimations that capture the basic impact of the main design drivers. However, the simplified models can represent physics only to a limited extent. On the other hand, higher-fidelity methods, which represent physics more accurately, are computationally more expensive and therefore, not appropriate for an extensive search of a design space. One approach to combine the advantages of both is multi-fidelity modeling. Here, many samples are evaluated using the computationally inexpensive low-fidelity models. The results provide information about the principal trends of the underlying system behavior. Additionally, a few samples are evaluated by means of higher-fidelity models. These provide information about how the low-fidelity model needs to be adapted in order to approximate the higher-fidelity results. Eventually, the results obtained from both fidelity sources are combined into one final prediction model.

The first part of this paper presents the outcomes of the initial efforts to understand how and where in the design space the low-fidelity results tend to deviate from the higher-fidelity results. For this purpose, relationships between individual aircraft parameters and the obtained correction factors were studied in a first step. Relatively simple relationships were found which are presented in the form of curve fits in section 6. Since the computational effort of the higher-fidelity methods applied in this academic example is still manageable, a Latin Hypercube Sample of the design space was generated and both low-fidelity and higher-fidelity methods were applied to all sample points. This allowed to generate sufficient sample points to perform well-informed manual curve fitting.

Even though manual curve fitting was successfully applied in this study, is has several drawbacks. Manual curve fitting is laborious and only a limited number of parameters can be visualized in a 3D representation at the same time. Indeed, more than two parameters may be included in the fit, but the engineer needs to postulate the final form of the fit equation. That is, a-priori knowledge is required about the interlinked effects of multiple parameters. However, this a-priori knowledge may not always be available. For instance, in this study, it is not known a-priori on which aircraft parameters the correction factors depend and whether there are interlinked dependencies between them. When no a-priori knowledge is available more sophisticated methods must be sought. Furthermore, manual curve fitting is not suited to be used inside an automated design optimization process as it requires a human in-the-loop. In summary, whereas manual curve fitting allows to get insight into simple relationships, it struggles when the relationships get more complex, more dimensions are involved, and no a-priori knowledge is available.

Non-parametric machine learning (ML) provides a means to overcome the disadvantages of curve fitting outlined above. It allows to perform curve fitting in a more generic, high-dimensional space without the need for a-priori knowledge about the impact of specific parameters on the output parameter. Adaption-based multi-fidelity modeling can be combined with various machine learning techniques. A machine learning model can be trained to predict the adaptive corrections which need to be applied on available low-fidelity results to approximate the associated higher-fidelity solutions.

Usually, a design is subject to many changes during the entire design process. This applies not only to the initial optimization phase but also later when, for instance, requirements are changed and the design may need to be re-evaluated and re-sized. This means, higher-fidelity analyses are typically not performed once at the beginning of an aircraft design project, but rather gradually as the design evolves and gets more and more refined. A multi-fidelity optimization approach should therefore incorporate a successive inclusion of higher-fidelity data points, as performed by [1] and [2].

The second part of this paper will apply adaption-based multi-fidelity modeling to the aerodynamic calculations and mass estimates of a scaled UAV wing. The approach includes a supervised machine learning model, which gradually learns the correction factors between the low-fidelity and the higher-fidelity models, based on a successively increasing set of higher-fidelity samples.

In summary, the goals of this study are:

- Explore relationships between correction factors and aircraft parameters
- Investigate the usability of non-dedicated wing mass estimation methods for the UAV class considered
- Prevent unquestioned imitation of higher-fidelity aerodynamic results; provide a method with a postulated extent to which the lower-fidelity results are to be corrected
- Investigate potential for automated knowledge transfer during successive knowledge build-up

In the multi-fidelity context, levels of fidelity are always defined relative to each other. That is, one

Table 1 – Parameter space considered in this study.

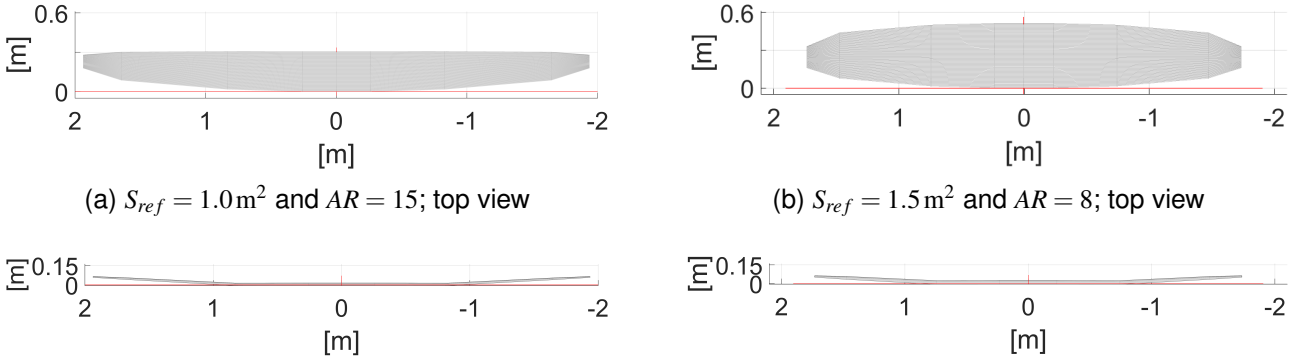
Parameter	Lower boundary	Upper boundary	Unit
Wing aspect ratio AR	5	15	$-$
Wing reference area S_{ref}	0.2	3.0	m^2
Design airspeed V_{design}	15	30	$\mathrm{m\,s}^{-1}$
Design altitude above MSL h_{design}	0	5000	m

level is denoted as "low-fidelity", and the the other level as "higher-fidelity", regardless of the actual physical level of fidelity of the associated tools. In this study, the physical level of fidelity of the "higher-fidelity" methods applied is actually medium. From now on, the two levels are referred to as "low-fidelity" and "medium-fidelity".

The next section introduces the scalable wing geometry and the parameter space considered in this study. Section 3. briefly presents the tools used for wing aerodynamic calculations. Section 4. outlines the wing mass estimation methods used. Following completion, section 5. presents the determination of the correction factors. The obtained curve fits are presented in section 6. Section 7. briefly describes the machine learning techniques used. The setup for the gradual training of the ML models is presented in section 8. and the results are shown in section 9. Finally, a conclusion and outlook are provided in sections 10. and 11., respectively.

2. Wing Geometry Used and Parameter Space Considered for the Analyses

The basic wing planform used for this study is depicted in Fig. 1. Two differently scaled versions are illustrated. The scaling of the wing is performed based on specified values of the wing reference area S_{ref} and the aspect ratio AR. During the scaling of the wing, other wing parameters such as taper ratio, sweep of the leading edge, dihedral, twist, etc. are held constant. The wing area and aspect ratio are



(a) $S_{ref} = 1.0\,\mathrm{m}^2$ and $AR = 15$; top view

(b) $S_{ref} = 1.5\,\mathrm{m}^2$ and $AR = 8$; top view

(c) $S_{ref} = 1.0\,\mathrm{m}^2$ and $AR = 15$; front view

(d) $S_{ref} = 1.5\,\mathrm{m}^2$ and $AR = 8$; front view

Figure 1 – Wing planform scaled according to different values for S_{ref} and AR.

two of the four parameters which define the parameter space considered in this study. The remaining parameters are the design airspeed V_{design} and the design altitude h_{design}. Whereas S_{ref} and AR have a direct influence on the wing geometry, V_{design} and h_{design} have an impact on the operating conditions, in particular, on the Reynolds number. The boundaries of the entire considered parameter space are shown in Table 1. In order to obtain samples which are well distributed across the entire parameter space, a Latin Hypercube Sampling (LHS) with 500 samples was performed. Samples resulting in operating Reynolds numbers which lie outside the region of available airfoil data were discarded. After this, 450 samples remained. For each sample, the basic wing planform from Figure 1 was scaled according to the values of S_{ref} and AR of the associated sample.

3. Wing Aerodynamic Calculation Methods Applied

This section introduces the methods used for the calculation of the wing aerodynamics. First, the medium-fidelity tool is briefly presented. Following completion, the low-fidelity formulas are provided.

3.1 Medium-Fidelity Wing Aerodynamics Calculations

The medium-fidelity tool used for aerodynamic calculations is PAWAT (Propeller and Wing Aerodynamics Tool) [3, 4], which is based on a modified three-dimensional lifting line theory. Classical lifting line theory uses the two-dimensional Kutta-Joukowski law. In contrast, PAWAT applies a three-dimensional vortex lifting law, which is an adaption described by [5]. PAWAT requires airfoil data which must be calculated beforehand, e.g., by means of XFOIL [6]. The aerodynamic airfoil data needs to be provided in the form of multidimensional data tables, which depend on the angle of attack, the Reynolds number, the Mach number, and the flap deflection angle. PAWAT can handle nonlinear and viscous airfoil data.

For further information regarding theory and implementation of PAWAT, the reader is referred to [3] and [4].
The aerodynamic panel data obtained by means of PAWAT are required by the medium-fidelity tool used for the wing mass calculation.

3.2 Low-Fidelity Wing Aerodynamics Calculations

The low-fidelity wing aerodynamic calculations are based on simple assumptions for the aerodynamic polars. The c_L over α curve of the wing is assumed to have a linear slope. The lift curve slope $c_{L\alpha}$ is obtained by means of a semi-empiric formula provided by [7]:

$$c_{L\alpha} = \frac{a_0 \cdot AR}{\frac{a_0}{\pi} + \sqrt{\left(\frac{AR}{\cos\Lambda_{50}}\right)^2 + \left(\frac{a_0}{\pi}\right)^2 - (AR \cdot Ma)^2}} \cdot \left(\frac{1}{57.3}\right) \tag{1}$$

where Λ_{50} is the sweep angle at the half chord line in degrees and AR the aspect ratio. Since no compressibility effects are considered, Ma in Eq. 1 is set to zero. Furthermore, a_0 is obtained from thin airfoil theory, which yields $a_0 = 2\pi$. The lift polar is approximated by a linear expression as a function of the angle of attack α:

$$c_L(\alpha) = c_{L\alpha} \cdot (\alpha - \alpha_{c_{L,0}}) \tag{2}$$

where $\alpha_{c_{L,0}}$ denotes the angle of attack at zero lift. According to [8], $\alpha_{c_{L,0}}$ is equal to the airfoil's zero lift angle of attack at the underlying Reynolds number. The airfoil data for various Reynolds numbers is obtained by means of XFOIL [6].

The drag polar of the wing is approximated as a quadratic parabola (k_2 is set to 1 for the low-fidelity polar estimation), as a function of c_L:

$$c_D = c_{D0} + k_1 \cdot c_L^2 + (k_2 - 1) \cdot c_L \tag{3}$$

where c_{D0} is the wing's zero lift drag coefficient. According to [8], c_{D0} can be approximated as the airfoil's zero lift drag coefficient c_{d0}, which is obtained from the airfoil data. The factor k_1 in Eq. 3 is obtained by:

$$k_1 = \frac{1}{\pi \cdot AR \cdot e} \tag{4}$$

where e is the oswald factor which is assumed to be 0.98 in this study. The use of the term $(k_2 - 1)$ instead of just k_2 is due to numerical reasons and serves to avoid division by zero in the implemented code. For the low-fidelity polar estimation, k_2 is set to 1. Hence, Eq. 3 describes a symmetric polar.

4. Wing Mass Estimation Methods Applied

This section presents the wing mass estimation methods used in this study. First, the medium-fidelity tool is introduced. Following completion, two semi-empiric formulas are presented.

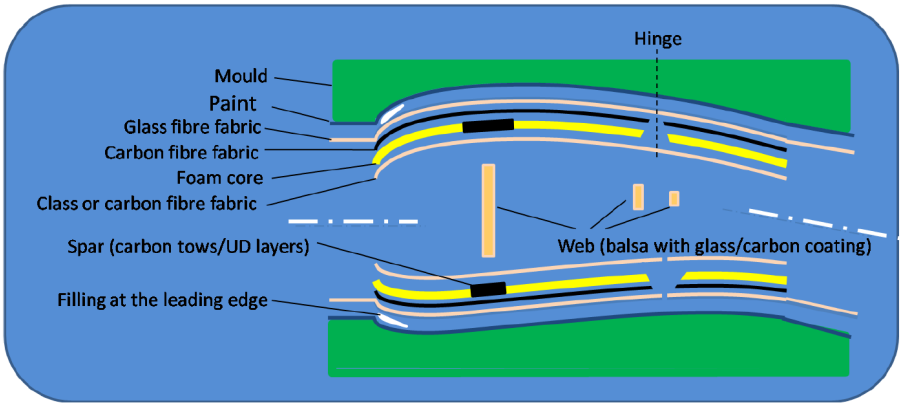


Figure 2 – Schematic build up of a hollow moulded wing [9].

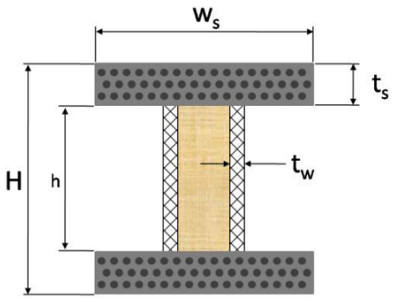


Figure 3 – Wing spar gemetry [9].

4.1 Medium-Fidelity Wing Mass Estimates

The medium-fidelity wing mass estimation method was developed by [9]. It can be used to calculate the mass of a hollow moulded wing, as depicted in Fig. 2. The main components of the wing are a sandwich shell, a main spar and additional webs. Whereas the shell is designed to carry the torsion load, the main spar is assumed to carry the entire bending load. The carbon layer of the shell is sized based on a method developed by [10]. The equation for the thickness dimensioning of the carbon layer at the root chord is:

$$t_{carbonlayer} = \frac{M_t \cdot b \cdot U}{8 \cdot \phi \cdot G \cdot A} \tag{5}$$

where M_t denotes the torsion moment at the root chord, b the span, U the circumference of the root airfoil chord, A the surface area of the root airfoil chord, G the shear modulus, and ϕ the twist angle. The carbon layer thickness obtained is rounded up to the next available fabric thickness.

The method [9] used for the sizing of the spar is based on [11]. The calculation bases on the wing spar geometry depicted in Fig. 3. The thickness t_s of the spar caps is set equal to the sandwich core thickness. The sandwich core material thickness, the width of the wing closing web, the area mass of the cover and the inner sandwich layer are not sized by the method. These parameters are assumed to be fixed. The spar cap width is calculated by:

$$w_s = \frac{6 \cdot M_b \cdot H}{\sigma_b \cdot (H^3 - h^3)} \tag{6}$$

where M_b is the bending moment and σ_b the maximum allowable bending stress. The spar cap width is determined for every wing section.

The spar's shear web is composed of balsa wood, which is coated with glass fiber in $\pm 45°$ orientation. The shear load is carried by the glass fiber coating. The area weight of the glass fiber fabric for each side of the web is determined by:

$$m_s = \frac{Q}{g \cdot k_\tau \cdot h} \tag{7}$$

where Q is the shear force and g the acceleration due to gravity. The factor k_τ is a generalized dimensioning parameter for E-glass fabric. Based on [12] and in accordance with [9], k_τ is set to $10\,\mathrm{km}$.

The final wing mass $m_{wing,Roessler}$ also includes estimates of the required glue and paint mass. A manufacturing skill factor is included, which has an influence on the amount of glue needed and the fiber volume coefficient obtained.

For more information regarding the medium-fidelity wing mass estimation method, the reader is referred to [9].

4.2 Low-Fidelity Wing Mass Estimates

No semi-empiric wing mass estimation formula was found, which has been developed specifically for the UAV class considered in this study ($< 50\,\mathrm{kg}$). Hence, formulas, which have originally been developed for other aircraft classes, are used and compared in this study. The first low-fidelity wing mass estimation method used in this paper is given by [13]:

$$m_{wing,Gundlach} = 0.0038 \cdot (N_Z \cdot m_{TO})^{1.06} \cdot AR^{0.38} \cdot S_{ref}^{0.25} \cdot (1+\lambda)^{0.21} \cdot (t/c)_{root}^{-0.14} \quad [\mathrm{kg}] \tag{8}$$

where N_Z denotes the ultimate load factor, m_{TO} the maximum takeoff mass in kg, AR the aspect ratio, S_{ref} the wing reference area in m^2, λ the taper ratio, and $(t/c)_{root}$ the thickness to chord ratio at the wing root. The equation was derived based on sailplanes ranging from 250 to $889\,\mathrm{kg}$.

The second wing mass estimation formula considered in this paper is provided by [14]:

$$m_{wing,Raymer} = 0.036 \cdot S_{ref}^{0.758} \cdot m_{fuel}^{0.0035} \cdot \left(\frac{AR}{\cos^2 \Lambda_{25\,\%}}\right)^{0.6} \cdot {q_c}^{0.006} \cdot \lambda^{0.04} \cdot \left(\frac{100 \cdot t/c}{\cos \Lambda_{25\,\%}}\right)^{-0.3} \cdot (N_Z \cdot m_{TO})^{0.49} \quad [\mathrm{lb}] \tag{9}$$

where S_{ref} denotes the wing reference area in ft^2, AR the aspect ratio, q_c the dynamic pressure at cruise in $\mathrm{lb\,ft^{-2}}$, λ the wing taper ratio, t/c the thickness-to-chord ratio, N_Z the ultimate load factor, and m_{TO} the flight design gross weight in lb. According to [14], the equation is applicable to general aviation aircraft. As the wing in this study does not carry any fuel, m_{fuel} in Eq. 9 is set to 1 so its effect is ignored. Since the wing planform of this study has multiple wing segments with individual sweeps, a surrogate sweep angle needs to be determined for the entire wing so it can be inserted in Eq. 9. For this purpose, a weighted average value is determined according to:

$$\bar{\Lambda}_{25\,\%} = \frac{b_{segment} \cdot \Lambda_{25\,\%,segment}}{b} \tag{10}$$

where b denotes the span of a segment or of the entire wing, respectively. There may be several ways to determine a surrogate sweep for a wing with multiple individually swept segments. Indeed, the formula used to calculate this sweep may affect the accuracy of the estimation obtained by Eq. 9. However, in the context of this study, it is irrelevant how the surrogate sweep is determined. A different method to determine $\bar{\Lambda}_{25\,\%}$ will just result in different correction factors. The correction factors basically *correct* the error Eq. 9 makes with regard to the medium-fidelity tool. Errors which are artificially introduced in Eq. 9 (e.g., inaccurate $\bar{\Lambda}_{25\,\%}$, use of $\bar{\Lambda}_{50\,\%}$ instead of $\bar{\Lambda}_{25\,\%}$, inserting m_{TO} in kg instead of lb,...) are just additional errors to be corrected by the correction factors. However, the calculation of the surrogate sweep must be consistent throughout the determination and usage of the correction factors.

5. Determination of the Correction Factors

A very popular adaption-based multi-fidelity method is the multiplicative scaling approach as presented in [15]. In this approach, correction factors are obtained by the fraction of medium-fidelity (index PAWAT) and low-fidelity solution (index LF) as a function of $\mathbf{x} = \{x_1, ...x_p\}$, which contains the constellation of the p input variables:

$$\beta(\mathbf{x}) = \frac{y_{\mathrm{PAWAT}}(\mathbf{x})}{y_{\mathrm{LF}}(\mathbf{x})} \tag{11}$$

For any given $\mathbf{x}$, the scaling function $\beta(\mathbf{x})$ specifies how the low-fidelity solution needs to be corrected to obtain the corresponding medium-fidelity solution.

To estimate $\beta(\mathbf{x})$ at points where it has not been calculated, various options are possible. [15] propose a sensitivity-based approach, where the correction factors are assumed to vary linearly in space. [16] and [17] demonstrated the inclusion of second-order information to approximate the scaling function. This paper, however, will apply more generic methods from the field of supervised machine learning to predict the correction factors at arbitrary points in the design space. A comparison of three machine learning methods is carried out: decision tree ensembles, support vector machines and Gaussian process regression. The latter has already been used in the context of variable fidelity optimization, e.g., in [18–20].

For the wing aerodynamic calculations, the multiplicative scaling approach presented above is modified. It is important to emphasize that the medium-fidelity solutions are not the all-encompassing truth but rather just a more accurate estimation of the real behavior. This estimation may still be inaccurate, e.g., due to numerical instabilities which occurred during the calculations, which is a common issue in aerodynamic analyses. Therefore, an unquestioned approximation of the medium-fidelity solutions is not always desirable. In fact, the engineer must be well aware of the extent to which a correction of the low-fidelity results is actually appropriate. In line with this motivation, a different approach is chosen for the multi-fidelity modeling of the wing aerodynamics. For the approach to be suitable within automated analyses, it is important that no human in-the-loop is needed to decide to what extent the low-fidelity solutions shall be adjusted. The approach proposed in this paper does not approximate every single data point of the higher-fidelity aerodynamic polars. Instead, it first postulates the mathematical form of the expected lift polar to be the same as in the low-fidelity estimation:

$$c_{L_{fit,PAWAT}} = \beta_1 \cdot c_{L\alpha} \cdot (\alpha - \beta_2 \cdot \alpha_{c_{L,0}}) \tag{12}$$

Then, this postulated equation is fitted on the data obtained by PAWAT to determine the unknown parameters β_1 and β_2. These parameters basically specify how the initial estimations for $c_{L\alpha}$ and $\alpha_{c_{L,0}}$ need to be *corrected*.

For the drag polar, the procedure is similar. A general non-symmetric quadratic polar is postulated:

$$c_{D_{fit,PAWAT}} = \beta_3 \cdot c_{D0} + \beta_4 \cdot k_1 \cdot C_L^2 + (\beta_5 \cdot k_2 - 1) \cdot c_L \tag{13}$$

and the unknown parameters β_3 to β_5 are determined by fitting Eq. 13 to the data obtained by PAWAT. The parameters β_3 to β_5 define how the initial estimations for c_{D0}, k_1 and k_2 need to be *corrected*.

Figure 4 illustrates the concept of fitting the postulated aerodynamic polar equations to the data obtained from PAWAT.

The wing mass correction factors are obtained by means of the unmodified multiplicative scaling approach. The correction factors are given by:

$$\beta_{6,Gundlach} = \frac{mass_{wing,Roessler}}{mass_{wing,Gundlach}} \tag{14}$$

$$\beta_{6,Raymer} = \frac{mass_{wing,Roessler}}{mass_{wing,Raymer}} \tag{15}$$

The next section presents the correction factors for all wings, that have been designed according to the samples introduced in section 2.

6. Curve-Fits of the Correction Factors

This section presents the results for the correction factors, which were obtained for various wing designs. Figure 5 shows the fits obtained for the aerodynamic correction factors β_1 to β_5. Very good

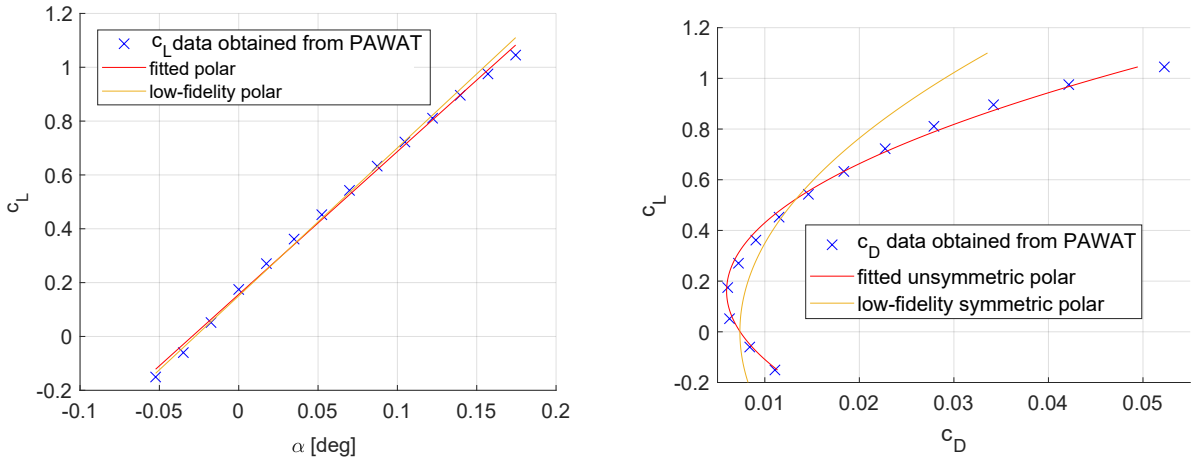


Figure 4 – Concept of fitting polars with postulated shapes to the data obtained from PAWAT.

fits were obtained for β_1, β_2, β_4 and β_5. For β_1 and β_2, some data points needed to be excluded in order to obtain clean fits. Consequently, the fits are not valid in the regions where the data points were excluded ($Re < 1.7 \times 10^5$ for β_1 and $Re < 3.2 \times 10^5$ for β_2). For β_3, no fit was found. However, in the region $2.8 \times 10^5 < Re < 7.2 \times 10^5$, β_3 can be approximated as 1. Apparently, for Reynolds numbers below and above this interval, assuming $c_{D0} = c_{d0}$ as in Eq. 3 results in a considerable error with regard to the results obtained by PAWAT. Obviously, at these Reynolds numbers, some effects are not captured by solely looking at the 2D airfoil data, which is retrieved using the Reynolds number at the mean aerodynamic chord of the wing.

The equations for the fits of the aerodynamic correction factors are:

$$\beta_1 = 0.1612 \cdot Re^{0.1666} \cdot AR^{-0.1225} - 3.342 \times 10^{-7} \cdot Re + 0.00756 \cdot AR \tag{16}$$

$$\beta_2 = 0.8061 + 1.894 \times 10^{8} \cdot Re^{-1.594} \tag{17}$$

$$\beta_4 = 9.791 \cdot Re^{-0.2238} \cdot AR^{0.8243} + 5.929 \times 10^{-7} \cdot Re - 0.1948 \cdot AR \tag{18}$$

$$\beta_5 = 0.8993 \cdot Re^{0.007461} \cdot AR^{-0.00243} \tag{19}$$

Figure 6 shows the fits obtained for the wing mass correction factors $\beta_{6,Gundlach}$ and $\beta_{6,Raymer}$. Very good fits were obtained for both $\beta_{6,Gundlach}$ and $\beta_{6,Raymer}$. This shows that either of the methods can be used for the considered wing planform; a correction of the low-fidelity predictions appears to be simply predictable based on S_{ref} and V_{design}.

The equations for the fits of the wing mass correction factors are:

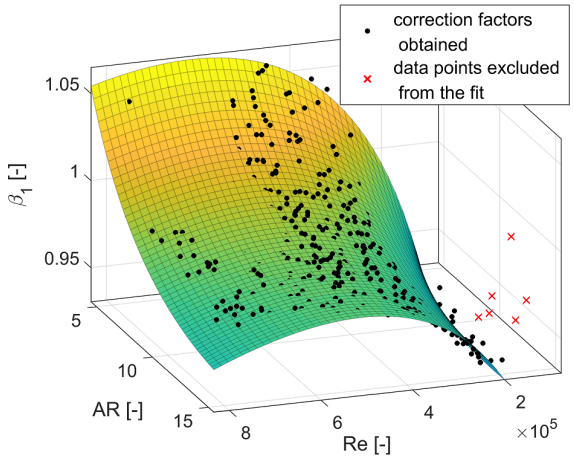
$$\beta_{6,Gundlach} = 0.6111 + 0.002931 \cdot V_{design}^{1.965} \cdot S_{ref}^{1.403} \tag{20}$$

$$\beta_{6,Raymer} = 0.5505 + 0.0002084 \cdot V_{design}^{2.479} \cdot S_{ref}^{0.9715} \tag{21}$$

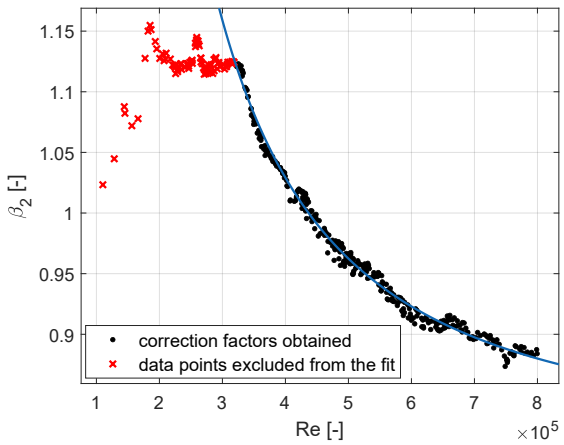
7. Machine Learning Models Applied

This section provides a brief description of the machine learning methods applied and is mostly adapted from [21–24].

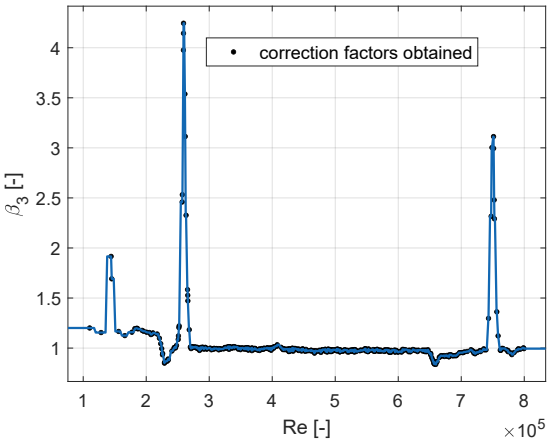
The goal of supervised machine learning for regression applications is to build a model which accurately makes predictions about the value of a continuous variable (*response variable*) based on provided input variables, denoted as *predictors* or *features*. The space defined by the predictors is called *predictor space*. A point in this space is called *sample*. Samples with associated responses used for the training of a model define the *training data set*.



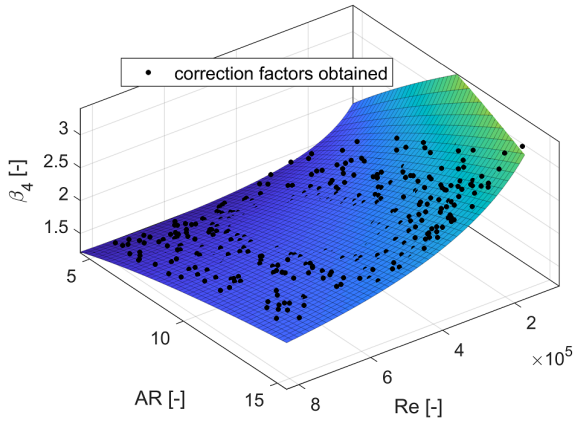
(a) Fit for β_1, with excluded data points; Adjusted $R^2 = 0.9702$, MSE $= 1.87 \times 10^{-5}$.



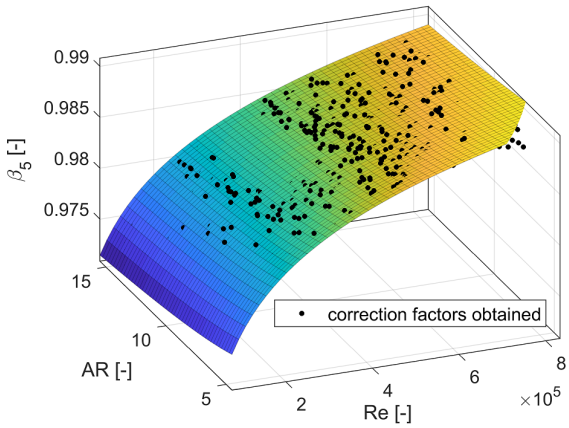
(b) Fit for β_2, with excluded data points; Adjusted $R^2 = 0.9913$, MSE $= 3.31 \times 10^{-5}$.



(c) No fit found for β_3, however, for $2.8 \times 10^5 < Re < 7.2 \times 10^5$, β_3 can be approximated as 1.

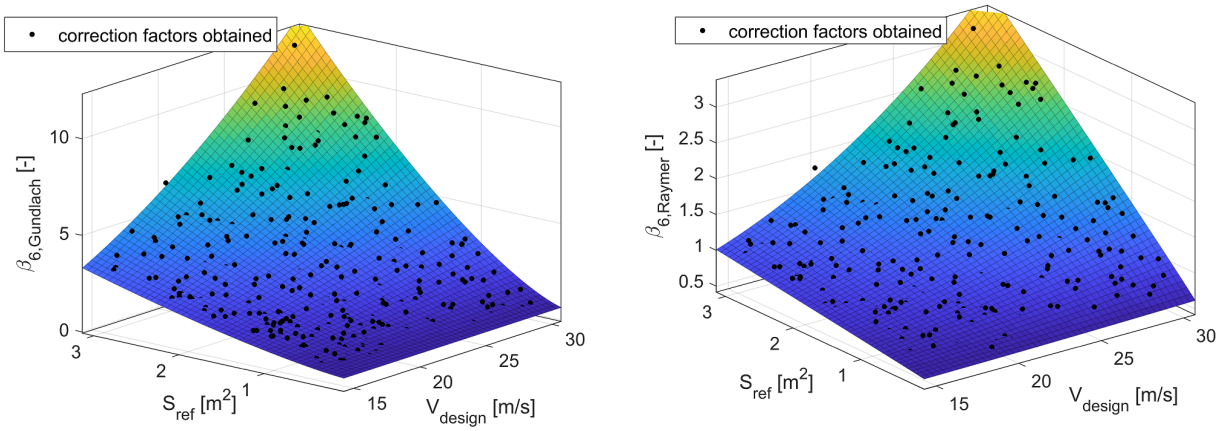


(d) Fit for β_4; Adjusted $R^2 = 0.9829$, MSE $= 3.0 \times 10^{-3}$.



(e) Fit for β_5; Adjusted $R^2 = 0.9690$, MSE $= 2.9 \times 10^{-7}$.

Figure 5 – Fits obtained for the aerodynamic correction factors β_1 to β_5. In the fit for β_1, data points with $Re < 1.7 \times 10^5$ were excluded. In the fit for β_2, data points with $Re < 3.2 \times 10^5$ were excluded. For β_3, no fit was obtained.



(a) Fit for $\beta_{6,Gundlach}$; Adjusted $R^2 = 0.9126$, MSE $= 3.6 \times 10^{-1}$.

(b) Fit for $\beta_{6,Raymer}$; Adjusted $R^2 = 0.9056$, MSE $= 2.4 \times 10^{-2}$.

Figure 6 – Fits obtained for the wing mass correction factors $\beta_{6,Gundlach}$ and $\beta_{6,Raymer}$.

To mitigate overfitting, i.e., the lack of generalizability of a model to previously unseen data, 5-fold cross-validation is used in this study. In k-fold cross-validation, the training data is randomly split into k data sets (folds) of approximately equal size. The model is trained k times; each time a different combination of $k-1$ data sets is used for the model training. The respective remaining data set is used as a validation data set. In this manner, 5 different values of the model error are obtained. The model error is assessed by the mean squared error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}(x_i))^2 \tag{22}$$

where $\hat{f}(x_i)$ denotes the predicted response for the ith sample, y_i the corresponding true response and N the total number of used samples. By averaging the 5 MSE values obtained through cross-validation, an estimate of the model error on new, unseen data is obtained.

The hyperparameters of the used machine learning methods are tuned by means of Bayesian optimization. As an acquisition function, we use the expected improvement at a certain point given the posterior distribution of the MSE. The Bayesian optimization stops after 30 iterations.

7.1 Decision Tree Ensembles

In decision trees, the model is built by splitting the predictor space into multiple regions. Each region is assigned a designated response value, which equals the mean of all training samples belonging to that region. The location of a split is denoted as node. The terminal nodes are called leaves. The decision where to split the predictor space is based on recursive binary splitting. This top-down approach successively divides the predictor space into pairs of regions [21]:

$$R_1(j,s) = \{X|X_j < s\} \text{ and } R_2(j,s) = \{X|X_j \geq s\} \tag{23}$$

where X_j denotes the jth predictor, and s the cutpoint along that predictor. A split is performed at that j and s which lead to the greatest reduction in MSE, i.e., minimize the equation

$$\frac{N_{R_1}}{N}\sum_{i:\, x_i \in R_1(j,s)}(y_i - \hat{y}_{R_1})^2 + \frac{N_{R_2}}{N}\sum_{i:\, x_i \in R_2(j,s)}(y_i - \hat{y}_{R_2})^2 \tag{24}$$

Here, a sample is denoted by x_i and its true response by y_i. N_{R_j} is the number of samples belonging to the jth region, and N the total number of samples. $\hat{y}_{R_1}$ is the mean response for the training samples in $R_1(j,s)$, and $\hat{y}_{R_2}$ the mean response for the training samples in $R_2(j,s)$.

To counteract overfitting we apply three different methods to combine multiple decision trees into one final regression model [22, 25, 26].

7.1.1 Bagging

One method to combine multiple trees into one model is bootstrap aggregation (bagging). Here, many subsets (bootstrapped training sets) of the training data are drawn via sampling with replacement. An individual tree is fit on each bootstrapped subset. Final predictions are obtained by averaging the predictions of all individual trees. [21]

7.1.2 Random Forests

Random forests are an extension of bagged trees. In bagged trees, all p predictors are available to the algorithm at each split. In random forests, on the other hand, the algorithm may only use a random subset of m predictors, where $m \in 1, ..., p$. [21]

7.1.3 Boosting

In boosted trees, individual trees are built sequentially. Each tree learns from the errors made by the respective previous tree(s). The procedure stops when a defined number of trees has been built. The final model is a combination of all single trees. The boosting algorithm used in this study is LSBoost. [21]

7.1.4 Tree Ensembles - Considered Hyperparameters

The tree ensemble hyperparamters considered in this paper are:

- Ensemble aggregation method — bagged/random forest, or LSBoost
- Number of trees — $\in [10, 500]$
- Learning rate (only for LSBoost) — $\in [1 \times 10^{-3}, 1]$
- Minimum leaf size — $\in [1, \max(2, \lfloor(\text{number of samples}/2)\rfloor)]$
- Maximum number of splits — $\in [1, \max(2, \text{number of samples} - 1)]$
- Number of predictors to sample — $\in [1, \max(2, \text{number of predictors})]$

7.2 Support Vector Machines

In order to use Support Vector Machines (SVMs) for regression based on a set of samples x, a linear hyperplane is constructed [21]:

$$f(x) = x^T w + w_0 \tag{25}$$

which is as "flat" as possible, i.e. the norm of the fitting coefficients $||w||^2$ should be minimized. Furthermore, the absolute error of the model should be less or equal than a threshold ε [22]:

$$|(y_n - f(x_n))| \leq \varepsilon \tag{26}$$

where x_n is a given sample with response y_n. The value of ε defines a range around the prediction function $f(x)$ in which errors are tolerated. This range is called ε-insensitive tube. Only absolute errors that lie outside this tube cause a penalty in the model. This penalty is represented by the slack variable ξ which denotes an error's deviation from the tube. Even though the model allows such deviations, they are to be minimized. Consequently, the objective function for SVM regression can be written as [27]:

$$J(w) = \frac{1}{2}||w||^2 + C \sum_{n=1}^{N} |\xi_n| \tag{27}$$

subject to [27]:

$$\forall n : |y_n - (x_n^T w + w_0)| \leq \varepsilon + |\xi_n| \tag{28}$$

The constant C is a positive numeric value that determines the trade-off between the flatness of $f(x)$ and the amount up to which errors larger than ε are tolerated. An increase in C leads to a higher penalty for the errors ξ [23].

This optimization problem is solved in its Lagrangian formulation. Non-negative multipliers α_n and α_n^* are introduced for each sample x_n. The problem is solved by minimizing [22]:

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)G(x_i,x_j) + \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \tag{29}$$

subject to [23]:

$$\begin{aligned} \sum_{n=1}^{N}(\alpha_n - \alpha_n^*) &= 0 \\ \forall n : 0 \leq \alpha_n &\leq C \\ \forall n : 0 \leq \alpha_n^* &\leq C \end{aligned} \tag{30}$$

The Gram matrix $G(x_i,x_j)$ is obtained based on the chosen kernel function. The kernel enlarges the predictor space by including certain functions or combinations built from the predictors.

The resulting solution function to make predictions is [22]:

$$f(x) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)G(x_n,x) + w_0 \tag{31}$$

where the parameter w_0 is determined by exploiting the Karush-Kuhn-Tucker conditions [28, 29].

7.2.1 SVMs - Considered Hyperparameters

The SVM hyperparamters considered in this paper are:

- C — $\in [1\times10^{-3}, 1\times10^{3}]$
- ε — $\in [1\times10^{-3}, 1\times10^{2}]\cdot \text{iqr}(Y)/1.349$; $\text{iqr}(Y)$: interquartile range of the response Y
- Kernel function — gaussian, linear, or polynomial
- Polynomial kernel function order (only for polynomial kernel) — $\in [2,4]$

7.3 Gaussian Processes

We define a Gaussian process (GP) [24]:

$$f(x) \sim GP(m(x), k(x,x')) \tag{32}$$

with mean function $m(x)$ [24]:

$$m(x) = \mathbb{E}[f(x)] \tag{33}$$

and covariance function (kernel) [24]:

$$k(x,x') = Cov(f(x), f(x')) = \mathbb{E}[(f(x) - m(x)) - (f(x') - m(x'))] \tag{34}$$

The kernel functions considered in this study are specified in the next subsection.

We choose a zero mean function and add a set of basis functions that transform the original feature vector x into a new feature vector $h(x)$. We obtain the regression model [24]:

$$g(x) = h(x)^T w + f(x) \tag{35}$$

with w being a vector of basis function coefficients and $f(x)$ a Gaussian process:

$$f(x) \sim GP(0, k(x,x')) \tag{36}$$

The considered basis functions are provided in the next subsection.

We augment the model to also consider Gaussian noise in the data and obtain the Gaussian process regression model [24]:

$$P(y_i|f(x_i),x_i) \sim N(y_i|h(x_i)^T w + f(x_i), \sigma_n^2) \tag{37}$$

with the noise variance σ_n^2.

7.3.1 GPs - Considered Hyperparameters

The GP hyperparameters used in this paper are:

- Basis function — none, constant, linear, or pure quadratic
- Kernel function with kernel hyperparameters— exponential, matern, or rational quadratic
- Initial value for the noise standard deviation σ_n — $\in [1 \times 10^{-4}, \max(1 \times 10^{-3}, 10 \cdot \sigma(y))]$

8. Setup for the Gradual Training

The goal of this study is to investigate the feasibility of automatic gradual knowledge build-up and transfer for various correction factors. For this purpose, the size of the training data set is increased gradually. To exclude cold-starts, the analysis considers a minimum training set size of 5. A portion of $20\,\%$ of the available 450 samples were excluded from the training set for testing. Consequently, the gradual training started with 5 samples and stopped with $(1-0.2) \cdot 450 = 360$ samples. The performance of the machine learning models is sensitive to the samples included in the training, especially at small numbers of included training samples. A disadvantageous distribution of training samples may not allow to capture the behavior of the response in all dimensions of the predictor space. In order to mitigate the dependency on the actually included training samples, the random splitting into training set and complementary test set was repeated 10 times. Independent gradual training and hyperparameter tuning was performed on each of the 10 training sets. In this manner, 10 different model performances were obtained after each gradual training step. To obtain an overall performance at each gradual training step, the median MSE of all 10 trainings was evaluated. The used predictors are the parameters from Table 1. Additionally, the wing's Reynolds number is included, as it represents the operating condition of a wing.

9. Results of the Gradual Training

This section presents the results of the gradual training, which was carried out as described above. Figure 7 displays the gradual training of different machine learning models for the prediction of the correction factors $\beta_1 - \beta_4$. The lower border of the greyed-out areas in the left plots of Fig. 7 marks the loss (MSE) which is obtained by the associated curve fits from Fig. 5. In contrast to the ML models, the curve fits were generated using all available 450 samples. Furthermore, the MSE of the ML models is evaluated on previously unseen data, whereas the MSE of the curve fits is evaluated on the very same data that was used for training. Moreover, in some curve fits, some data points were excluded from the fits, because they were not compatible with the simple fit equations obtained. Consequently, the comparison of the MSEs between curve fit and ML model is disadvantageous for the ML models. However, the comparison still allows a conservative estimation of whether the ML model yields smaller losses than the curve fit.

The left plot in Fig. 7 a) shows the loss obtained in the prediction of β_1. At the starting conditions with 5 training samples, all ML models present a median MSE of less than 1.0×10^{-3}. With increasing training set size, the GP model presents the fastest decrease in loss. Furthermore, the GP model achieves a smaller loss than the corresponding curve fit after only approximately 50 included training samples. The ensemble and SVM models, on the other hand, do not yield smaller losses than the associated curve fit throughout the entire gradual training. The time needed for training and optimization of the ML models for each training set size is presented on the right side of Fig. 7 a). The shortest times are achieved by the GP model. The time of the GP model demonstrates a non-linear increase with the number of included samples. At a large training set size, its training time is close to the one of the ensemble model. The time of the ensemble model is rather insensitive to the training set size. The SVM time shows an approximate linear increase with an increasing number of included training samples. Up to a training set size of about 100, the SVM time is between the ensemble and the GP model. Above this number, the SVM model requires the longest time for training and optimization.

Figure 7 b) illustrates the gradual training for the prediction of β_2. All models start with a loss of less than 1.0×10^{-2} at a training set size of 5. As before, the GP model demonstrates the steepest

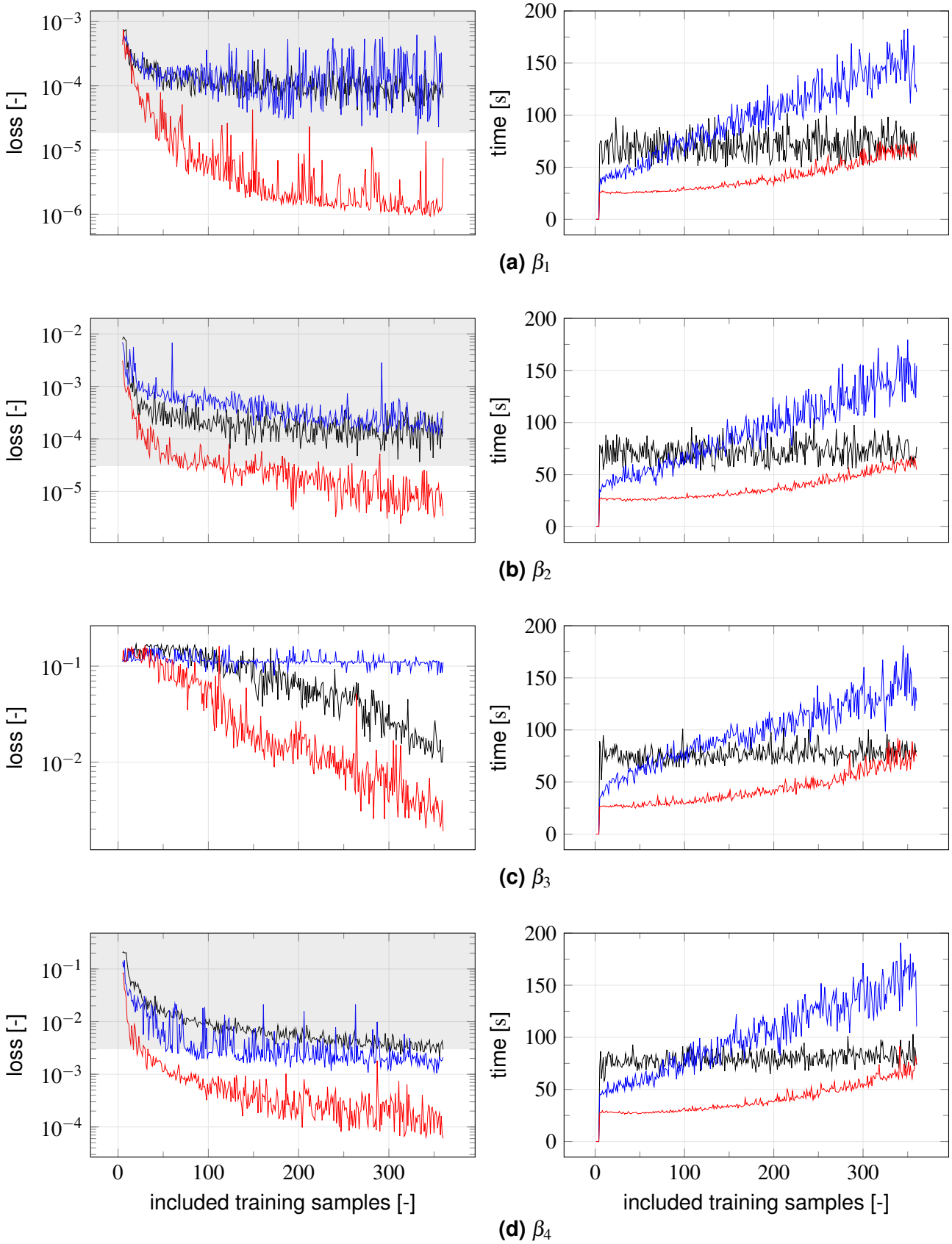


Figure 7 – Loss (MSE) and required training time for $\beta_1 - \beta_4$ using different machine learning models and a gradually increasing training data set size. The plots are the median of 10 runs with individual, random training sets.

decrease in loss with increasing training set size. Also, it is the only model that achieves a smaller loss than the corresponding curve fit. The times required for training and optimization are similar to the case of β_1.

Figure 7 c) displays the gradual training for the prediction of β_3. No greyed-out area is included in Fig. 7. This is because no fit was found in Fig. 5 for β_3. The only statement that could be made was that β_3 can be estimated to 1 in the range of $2.8 \times 10^5 < Re < 7.2 \times 10^5$. In Fig. 7 c), all models demonstrate a median MSE of slightly greater than 1.0×10^{-1} at starting conditions with 5 included training samples. The loss of the SVM model is not reduced significantly with increasing size of the training set. The ensemble method yields a reduced loss only above approximately 100 included samples. The GP model presents a reduced loss after about 50 included samples. In general, it achieves the smallest loss throughout the entire gradual training. The training and optimization times are similar to the previous cases.

Figure 7 d) presents the gradual training for the prediction of β_4. All models begin with a median MSE of approximately 1.0×10^{-1} at starting conditions with 5 included training samples. The GP model shows the strongest decrease in loss with increasing number of training samples. The loss of the SVM model is mostly between that of the GP and the ensemble models. After only abut 25 included samples, the GP model yields a smaller loss than the associated curve fit. The SVM model follows after about 50 samples but shows considerable oscillations in the loss when more samples are included. Because of these oscillations, the SVM model occasionally yields a higher loss than the associated curve fit for training set sizes greater than 50. The training and optimization times are similar to the previous cases.

Figure 8 a) shows the gradual training for the prediction of β_5. All models begin with a median MSE of approximately 1.0×10^{-5} at starting conditions with 5 included training samples. The loss decreases fastest in the GP model, followed by the SVM and then the ensemble model. The models demonstrate a very strong initial decrease in loss but nearly stagnate, with some fluctuations, for training set sizes greater than about 100. The only model that occasionally yields smaller losses than the curve fit is the GP model. The training and optimization times are similar to the previous cases.

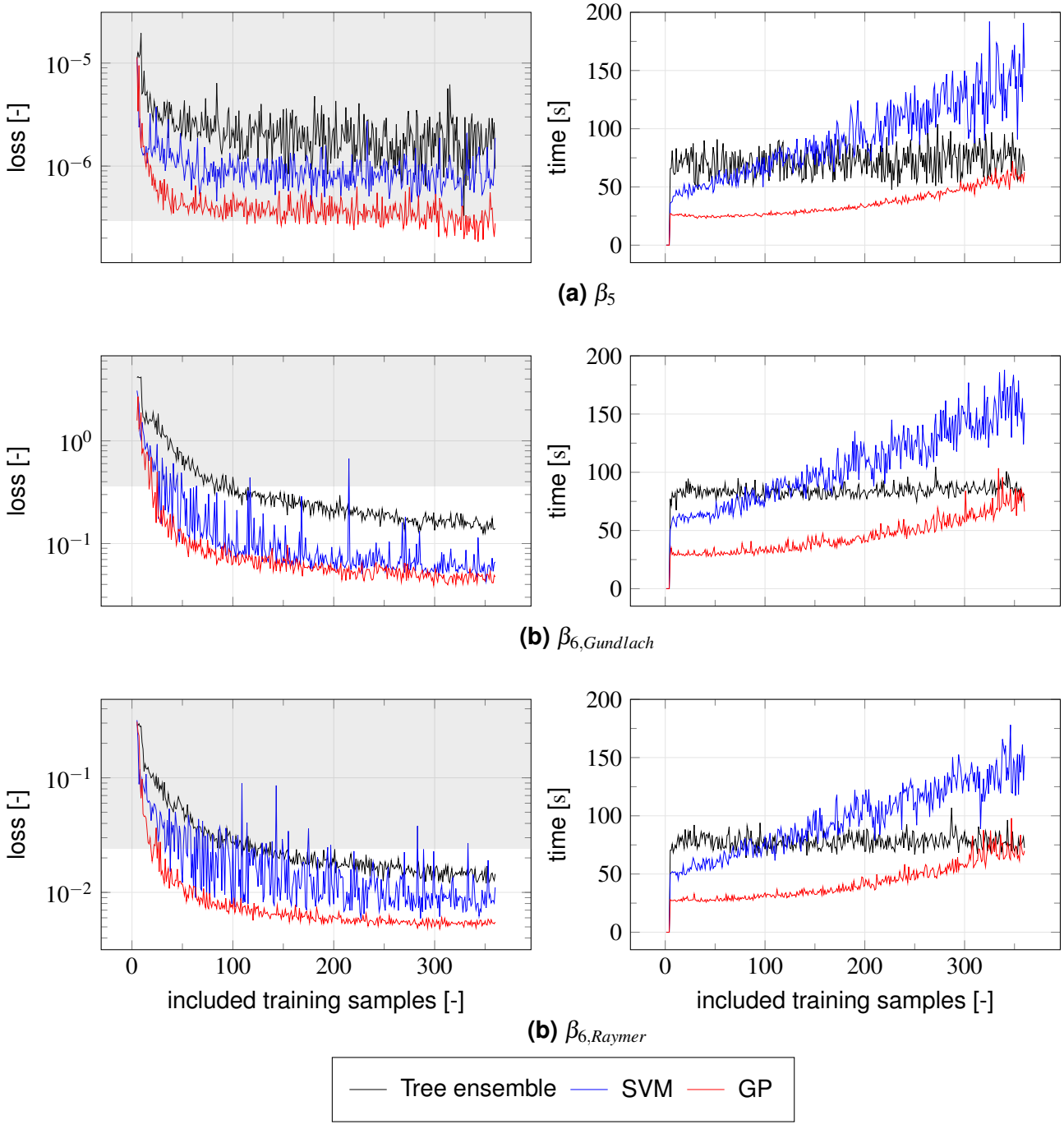
Figure 8 b) depicts the gradual training for the prediction of $\beta_{6,Gundlach}$. All models begin with a median MSE of about 5.0×10^{-1} at starting conditions with 5 included training samples. Again, the loss decreases fastest in the GP model, followed by the SVM and then the ensemble model. The GP model achieves a smaller loss than the corresponding curve fit after only approximately 25 included training samples. The SVM model follows with slightly more included samples and the ensemble model requires approximately 100 samples. In general, the smallest loss throughout the entire gradual training is obtained by the GP model. The training and optimization times are similar to the previous cases.

Figure 8 c) shows the gradual training for the prediction of $\beta_{6,Raymer}$. The results regarding the loss and time are similar to the results obtained for $\beta_{6,Gundlach}$. The left plot shows an offset towards smaller losses compared to $\beta_{6,Gundlach}$. However, as can be seen in Fig. 6, the range of the response is also much smaller for $\beta_{6,Raymer}$ compared to $\beta_{6,Gundlach}$.

10. Conclusion

This study addressed several issues regarding automated stepwise knowledge build-up and transfer during the analysis of scaled UAV wings. The first part of the study presents the modeling methods for the wing aerodynamic calculations and the wing mass estimation, which form the basis of this work. For each discipline, methods of two different levels of fidelity are introduced.

Correction factors are determined which specify how the low-fidelity results need to be adjusted to approximate the higher-fidelity results. Curve fits are developed to identify relationships between the correction factors obtained and certain aircraft parameters.



(a) β_5

(b) $\beta_{6,Gundlach}$

(b) $\beta_{6,Raymer}$

Figure 8 – Loss (MSE) and required training time for β_5, $\beta_{6,Gundlach}$ and $\beta_{6,Raymer}$ using different machine learning models and a gradually increasing training data set size. The plots are the median of 10 runs with individual, random training sets.

For the corrections of the aerodynamic results, an approach is presented that prevents unconditional imitation of the higher-fidelity results. This is desirable, as the "higher-fidelity" results themselves may be inaccurate due to, e.g., incorrect airfoil input data. Also, numerical issues may lead to physically inexplicable results obtained by the higher-fidelity tool. Instead of unconditionally approximating the higher-fidelity aerodynamic results, the approach incorporates a-priori knowledge about the expected shape of the aerodynamic polars. The approach does not require a human in-the-loop to decide to what extent the low-fidelity results are to be adjusted.

No low-fidelity methods were found that were specifically developed for the wing mass estimation of the considered UAV class. Instead, formulas provided by [13] and [14], originally developed for other classes of aircraft, are used to estimate wing mass. Whereas the methods fail to accurately predict wing mass, simple curve fits are obtained, which describe how the results need to be adjusted to obtain the higher-fidelity results. This suggests that, even though the formulas are not eligible to the UAV class in question in terms of absolute accuracy, they still seem to contain valid information about how wing mass tends to change with certain parameters.

The last part of the paper investigates the feasibility of automatically and gradually creating a knowledge base, which can be used to make predictions about the value of the considered correction factors based on given parameters. For this purpose, different machine learning techniques were applied: tree ensembles, support vector machines, and Gaussian processes. The models were trained in a stepwise manner. At the beginning, only 5 data points were included in the training. Step by step, additional training points were included. For each training set size, the different ML models were trained, the hyperparameters were optimized, and the model performance was evaluated on a test set, which was unseen by the models during training. The test set remained the same for all training set sizes. This allows to compare the results among each other and to track the actual improvement of a given model as the training set size increases. The model performance was evaluated on the basis of the mean squared error (MSE) on the test set. In general, the different ML models show a rapid decrease in MSE as the training set size is started to increase. Overall, the Gaussian process model shows not only the fastest decrease in MSE but also the lowest MSEs in general. For most of the correction factors, the Gaussian process model requires less than 50 training samples to yield smaller MSEs than the corresponding manually generated curve fits, which are based on 450 samples. Furthermore, the time needed for training and hyperparamter optimization is lowest for the Gaussian process model. At the maximum number of considered training samples (360), the Gaussian process model requires only about one minute for training and hyperparamter optimization.

11. Outlook

This paper indicates the general feasibility of successive automated knowledge build-up and transfer during the design of a UAV. The feasibility is manifested in the small errors obtained by the Gaussian process models (even with small training sets), paired with short times needed for training and hyperparameter optimization.

The next step is the implementation of the approach into a design optimization process, where the knowledge about the correction factors is successively gained and re-used as the optimization progresses.

12. Contact Author Email Address

mailto: tim.klaproth@tum.de

13. Copyright Statement

References

[1] Alexandrov, N, NIELSEN, E, LEWIS, R and Anderson, K. First-order model management with variable-fidelity physics applied to multi-element airfoil optimization. *8th Symposium on Multidisciplinary Analysis and Optimization*. 2000. https://doi.org/10.2514/6.2000-4886.

[2] Alexandrov, N, Lewis, R, Gumbert, C, Green, L and Newman, P. Approximation and Model Management in Aerodynamic Optimization With Variable-Fidelity Models. *Journal of Aircraft*. Vol. 38, pp. 1093–1101. 2001. https://doi.org/10.2514/2.2877.

[3] Steiner, HJ. Preliminary Design Tool for Propeller-Wing Aerodynamics: Part II: Theory. , 2010.

[4] Steiner, HJ. Preliminary Design Tool for Propeller-Wing Aerodynamics: Part I: Implementation and Reference Manual. , 2010.

[5] Phillips, WF and Snyder, DO. Modern Adaptation of Prandtl's Classic Lifting-Line Theory. *Journal of Aircraft*. Vol. 37, No. 4, pp. 662–670. 2000. https://doi.org/10.2514/2.2649.

[6] Drela, M and Youngren, H. XFOIL: Subsonic Airfoil Development System. , 2013. URL https://web.mit.edu/drela/Public/web/xfoil/.

[7] Lowry, JG and Polhamus, EC. A method for predicting lift increments due to flap deflection at low angles of attack in incompressible flow. 1957.

[8] Anderson, J. *Fundamentals of Aerodynamics*. 5th ed.. McGraw-Hill Education, 2010.

[9] Rößler, CO. Conceptual Design of Unmanned Aircraft with Fuel Cell Propulsion System. Dissertation. Technical University of Munich, 2012.

[10] Dolch, S. *Rippenflügel aus Faserverbundwerkstoffen. Leichtbau mit Rohrholmen in Theorie und Praxis*. 2nd ed.. Verlag für Technik und Handwerk, Baden-Baden, 1999.

[11] Quabeck, H. Design und Berechnung eines Tragflächenholms. , 2007. URL http://www.hq-modellflug.de/theory.htm. checked on 2022-01-25.

[12] VDI Society Materials Engineering. VDI 2013 Blatt 1 (1970-01-00). , 1970. Artical id: DE_1444025.

[13] Gundlach, J. *Designing Unmanned Aircraft Systems: A Comprehensive Approach*. 2nd ed.. American Institute of Aeronautics and Astronautics, 2014.

[14] Raymer, DP. *Aircraft Design: A Conceptual Approach*. 5th ed.. American Institute of Aeronautics and Astronautics, 2012.

[15] Chang, KJ, Haftka, RT, Giles, GL and Kao, PJ. Sensitivity-based scaling for approximating structural response. *Journal of Aircraft*. Vol. 30, No. 2, pp. 283–288. 1993. https://doi.org/10.2514/3.48278. URL https://doi.org/10.2514/3.48278.

[16] Gano, S, Pérez, V, Renaud, J, Batill, S and Sanders, B. Multilevel Variable Fidelity Optimization of a Morphing Unmanned Aerial Vehicle. 2004. https://doi.org/10.2514/6.2004-1763.

[17] Eldred, M, Giunta, A and Collis, S. Second-Order Corrections for Surrogate-Based Optimization with Model Hierarchies. 2004. https://doi.org/10.2514/6.2004-4457.

[18] Gano, S and Renaud, J. Variable Fidelity Optimization Using a Kriging Based Scaling Function. *Collection of Technical Papers - 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Vol. 3. 2004. https://doi.org/10.2514/6.2004-4460.

[19] Gano, SE. Simulation-Based Design Using Variable Fidelity Optimization. Dissertation. University of Notre Dame. Notre Dame, Indiana, 2005.

[20] Gano, SE, Renaud, JE, Martin, JD and Simpson, TW. Update strategies for kriging models used in variable fidelity optimization. *Structural and Multidisciplinary Optimization*. Vol. 32, No. 4, pp. 287–298. 2006. https://doi.org/10.1007/s00158-006-0025-y.

[21] James, G, Witten, D, Hastie, T and Tibshirani, R. *An introduction to statistical learning: With applications in R*. corrected at 8th printing 2017 ed.. Springer texts in statistics. Springer, New York, 2017. URL https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6312402.

[22] Hastie, T, Tibshirani, R and Friedman, JH. *The elements of statistical learning: Data mining, inference, and prediction*. second edition ed.. Springer Series in Statistics. Springer, New York, NY, 2017. URL https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6314834.

[23] Wang, L (ed.). *Support Vector Machines: Theory and Applications*. Springer eBook Collection Engineering, Vol. 177. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. https://doi.org/10.1007/b95439.

[24] Rasmussen, CE and Williams, CKI. *Gaussian processes for machine learning*. 3rd ed.. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 2008.

[25] Breiman, L. Bagging predictors. *Machine Learning*. Vol. 24, No. 2, pp. 123–140. 1996. https://doi.org/10.1007/BF00058655.

[26] Breiman, L. Random Forests. *Machine Learning*. Vol. 45, No. 1, pp. 5–32. 2001. https://doi.org/10.1023/A:1010933404324.

[27] Vapnik, VN. *The nature of statistical learning theory*. second edition ed.. Statistics for engineering and information science. Springer, New York, NY, 2010.

[28] Karush, W. Minima of functions of several variables with inequalities as side condition. Master's thesis. The University of Chicago. Chicaco, IL, 1939.

[29] Kuhn, H.W. and Tucker, A.W.. Nonlinear programming. *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, Berkely, CA*. 1951. pp. 481–492.