Delft University of Technology

System Identification of Aerospace Systems

AE4320

Multivariate Simplex Splines

Authors:

Luc Kloosterman (4697006)

September 7, 2020

Contents

1 Report introduction

System identification can be used to reverse engineer a model which approximates the output of the original system. For this, only an input-output dataset of the original system is needed. System identification provides several tools to reconstruct models which is particularly useful for aircraft simulation & control. Some examples are:

- Creating realistic flight simulations
- Understanding human information processing and integration use in haptic feedback
- Control of high performance aircraft
- Fault tolerant control of damaged aircraft
- More efficient and safer drones

The model of a system is a mathematical abstraction of the system that aims to capture its input-output behaviour while at the same time simplifying and conceptualising the inner working of the original model. Generally, the equations of motion of a simple linear model can be derived directly and the transfer function becomes the model of the system. This model is called a white-box model and has a well understood input-output behavior, has a high prediction time horizon and is valid for the entire system domain. In order to obtain such a model a complete understanding of the physical principles of a system is required. Also, it can be impossible to solve differential equations exactly, even if they are understood.
A model where the equations of motion are completely unknown is called a black-box model. It is purely based on input-output relationships of a system. Also, the prediction power is difficult or impossible to verify and is only valid inside the domain of the given input data. On the other hand, no knowledge of the actual physics of a system is required and it can be used to model any system given a set of input-output data.
A model which lies between a white-box and a black-box model is called a grey-box model. It is based on physical principles where they are known and where they are practical and is based on input-output behavior where they are not. Because of this, less knowledge of the dynamics is needed compared to a white-box model. Also, it is easier to verify performance compared to a black-box model and harder compared to a white-box model. The same holds for the predictive power, which is the highest for a white-box model. At last a grey box model is often harder to create than a black box model.

1.1 State estimation techniques and parameter estimation methods

State estimation is defined as searching for the best estimate of the state vector while the parameter vector is known. It is needed because process and measurement noise leads to biased estimates, because certain states can not be measured directly or because the accuracy can be improved by combining different sensors. The main tool for state estimation is the Kalman Filter which working principle is the calculation of a weighted average between the measured and predicted state. The weight, also known as the Kalman gain, depends on the uncertainty in the measurement. The linear Kalman Filter is an optimal linear filter which only works for linear systems. However, many extensions like the Extendend Kalman Filter and the Iterated Extended kalman Filter exist. These extensions can deal with non-linear systems but make the filter less optimal. Furthermore, all noise statistical information should be known.

Parameter estimation is defined as the act of determining an optimal set of model parameters given a set of measurements. A very large number of parameter estimation techniques exist. Some advanced methods are neural networks and multivariate splines, which will be discussed in the next subsection, and some more basic methods like linear regression and simple polynomials. An important property to select a model structure is the linear-in-the-parameter property, which is valid when the first derivative is a function of only the state. When this is true, linear regression can be used which in general is simple to implement and has a low computational complexity. The most used optimiser in such problems is ordinary least squares where the squares of the errors are minimised instead of the errors itself. However, this estimator can only deal with zero-mean constant-variance noise which is not necessarily true for most model residuals. When this is not the case a variation to the ordinary least squares estimation routine has to be used. In order to analyse the noise sensitivity and variability a parameter covariance matrix is constructed from the estimator itself.

1.2 Advanced system identification methods

A neural network is an algorithm that tries to recognise underlying relationships in a set of data. This is done with a number of layers, each consisting of a set of neurons which contain a basis function. Popular basis functions are the Identity, the Sigmoid, the hyperbolic Tangent and the Gaussian function. A neural network with at least one hidden layer can approximate any continuous non-linear function well on a compact set, if a sufficient number of hidden neurons are present. The performance of a network is most often measured with the mean squared error cost function. The optimisation of a neural network is done by placing a weight on every neuron which then will be optimised. This is done by iterative updating these weights using partial derivatives, which are obtained through back-propagation, in the negative gradient direction. The main advantage of neural networks is that it can enables non-linear optimisation with a high approximation power at the cost of computational complexity.

Multivariate splines is a piece wise continuous polynomial function of multiple simplices with a predefined continuity. A simplex is a geometric structure that minimally spans a set of dimensions and has a local Barycentric coordinate system. The Barycentric coordinates form a stable local basis which consists of a summation of basis functions. In order to optimise the system an ordinary least square cost function is defined using a global regression matrix. Smoothness constraints are used to enforce smoothness between simplices. These constraints are equality constraints which can be solved using the Lagrange Multiplier method. The main advantages is the high approximation power while it has a lower computational complexity compared to neural networks. A disadvantage of multivariate splines is that it does not support non-linear optimisation.

2 State & Parameter estimation with F-16 flight data

Given is a highly non-linear and noisy wind tunnel dataset of the F-16 where the angle of attack (α), the side slip (β), velocity (V) and the pitching coefficient (C_m) is measured. However, it is known that there is an unknown bias ($C_{\alpha_{up}}$) in the measured angle of attack (α_m). In order to obtain the true angle of attack (α_{true}) an Extended Kalman Filter and an Iterative Extended kalman filter is implemented. After this, the pitching moment coefficient will be estimated form the true angle of attack and the side slip using an ordinary least squares estimator for a simple polynomial structure.

2.1 Kalman Filter

Estimating a bias in the angle of attack is done with the use of unbiased an noise-free linear accelerometers meter measurements sampled with $\Delta t = 0.01$ seconds. These measurements together form the input vector used in the Kalman filter. With this information the state vector, the measurement vector and the input vector can be defined by equation 1.

$$x_k = \begin{bmatrix} u \\ v \\ w \\ C_{\alpha_{up}} \end{bmatrix} z_k = \begin{bmatrix} \alpha_m \\ \beta_m \\ V_m \end{bmatrix} u_k = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \tag{1}$$

Furthermore, it is given that the measured angle of attack, slide slip and velocity are related to their true values as given in equation 2 with v_α, v_β and v_V being a white noise sequence.

$$\begin{aligned} \alpha_m = \alpha_{true}(1 + C_{\alpha_{up}}) + v_\alpha = \arctan{(\frac{w}{u})}(1 + C_{\alpha_{up}}) + v_\alpha \\ \beta_m = \beta_{true} + v_\beta = \arctan{(\frac{v}{\sqrt{u^2+v^2}})} + v_\beta \\ V_m = V_{true} + v_v = \sqrt{v^2+u^2+w^2} + v_V \end{aligned} \tag{2}$$

With all the information present the trivial system equation $\dot{x}_k$ and the non-trivial output equation z_k can be computed and are given by equations 3 and 4 respectively. It can be seen that the system dynamics are fully described by the system input and that the output equation is non-linear.

$$\dot{x}_k = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{C}_{\alpha_{up}} \end{bmatrix} = f(x_k, u_k, t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ C_{\alpha_{up}} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_u \\ w_v \\ w_w \end{bmatrix} \tag{3}$$

$$z_k = \begin{bmatrix} \alpha_m \\ \beta_m \\ V_m \end{bmatrix} = h(x_k, u_k, t) = \begin{bmatrix} \arctan{(\frac{w}{u})}(1 + C_{\alpha_{up}}) \\ \arctan{(\frac{v}{\sqrt{u^2+v^2}})} \\ \sqrt{v^2+u^2+w^2} \end{bmatrix} + \begin{bmatrix} v_\alpha \\ v_\beta \\ v_V \end{bmatrix} \tag{4}$$

To initialise the Kalman Filter the initial states x_{k0}, the initial covariance matrix P_0, the process noise matrix Q and the sensor noise matrix R need to be defined which is done as depicted by equation 5. It is assumed that the initial flight condition is symmetric and therefore u is equal to the velocity at $t = 0$ while all other initial states are zero.

$$x_{k0} = \begin{bmatrix} 151 \\ 0 \\ 0 \\ 0 \end{bmatrix} P_0 = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} Q = \begin{bmatrix} 10^{-6} & 0 & 0 & 0 \\ 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 10^{-6} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} R = \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 1.69e \cdot 10^{-6} & 0 \\ 0 & 0 & 0.0121 \end{bmatrix} \tag{5}$$

Due to the fact that the system is non-linear, a linear Kalman Filter will not be able to estimate the bias in the measured angle of attack. Therefore, the Extended Kalman Filter or the Iterative Extended Kalman Filter has to be used. Both filters are able to deal with non-linear systems however the Extended Kalman Filter does not provide the guarantee of global convergence. Hence, both Kalman Filters will be evaluated. In the case of not obtaining global convergence with the Extended Kalman Filter, the results of the Iterative Extended kalman Filter will be used throughout this report.

As can be seen in figure 1 the Extended Kalman Filter and the Iterative extended Kalman Filter converge to the same value with the same performance. Therefore, an Extended Kalman filter will be suitable for this problem as it is less computational complex. Figure 2 shows the estimated output states and the error between the measured and estimated states. As can be seen all errors have the properties of white noise which proves that the output states are correctly estimated.

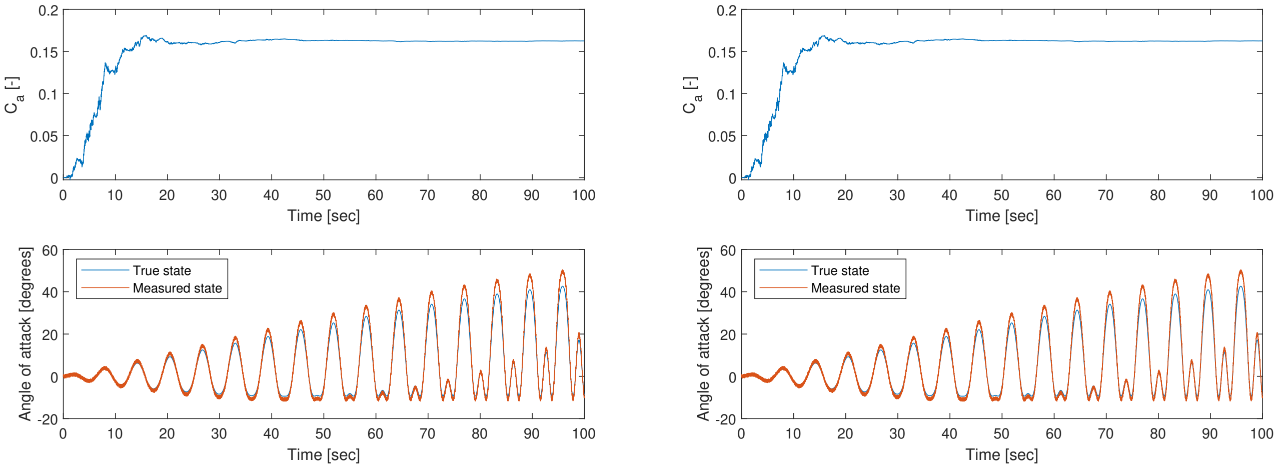

Figure 1: Estimation of angle of attack bias constant with the Extended Kalman Filter (left) and the Iterative Extended Kalman Filter (right). The bottom graphs shows the measured angle of attack compared to the true angle of attack.

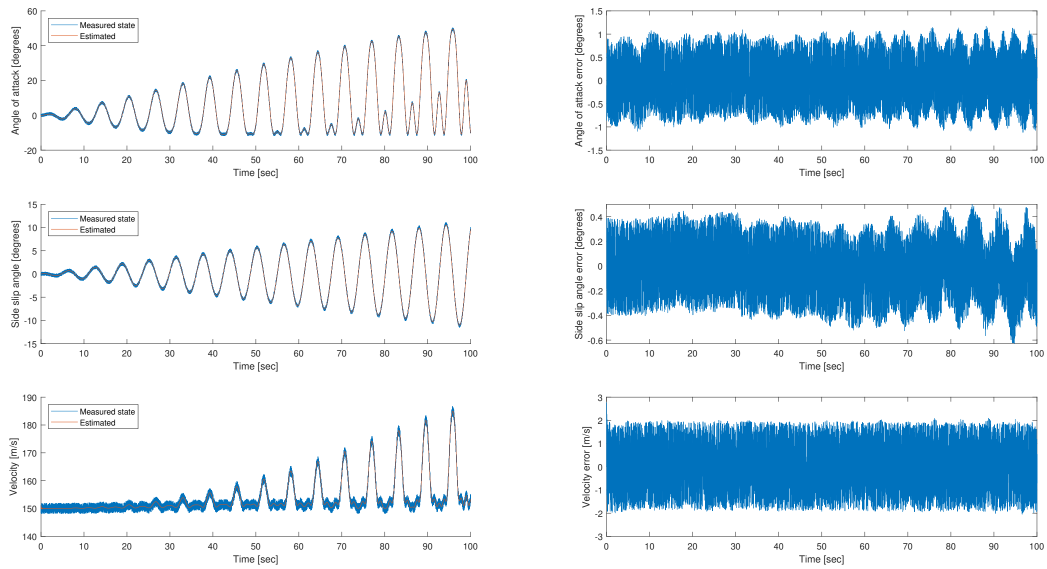

Figure 2: The measured and estimated outputs of the Extended Kalman Filter (left) and the error between measured and estimated outputs (right).

2.2 Polynomial Least squares Estimator

With the true angle of attack and the slide slip estimated an ordinary least squares estimator for a simple polynomial structure can be implemented. With this model, the noisy pitching coefficient measurements which are shown in figure 3 will be estimated. The input data is randomly split in half for identification and validation using matlabs 'cvpartition' function.

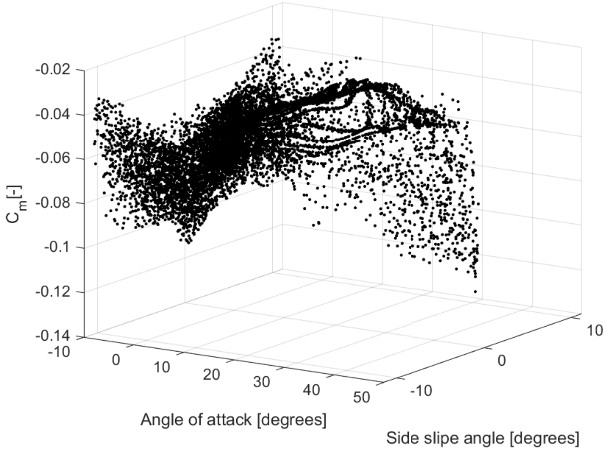

Figure 3: Pitching coefficient input data.

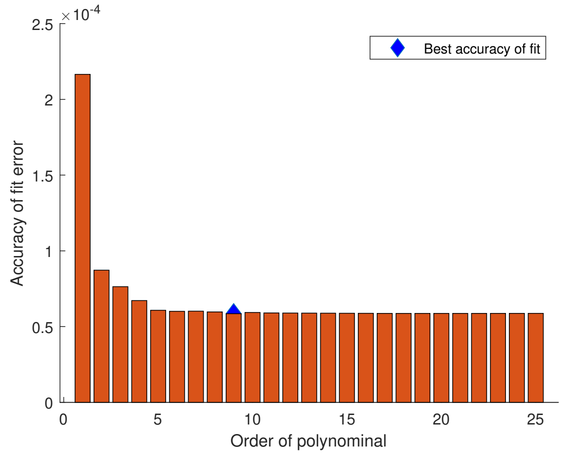

Figure 4: Mean squared error for different order polynomials.

The order of the polynomial depends the accuracy of fit. If the order is too low the polynomial is not able to fit the data accurately. On the other hand, a too high order will cause the polynomial to overfit. The accuracy of fit is evaluated using the mean squared error and in figure 4 it can been seen that a 9th order polynomial has the lowest mean squared error of $5.7754 \cdot 10^{-5}$. With this polynomial order the model can estimate the dataset as shown in figure 5.

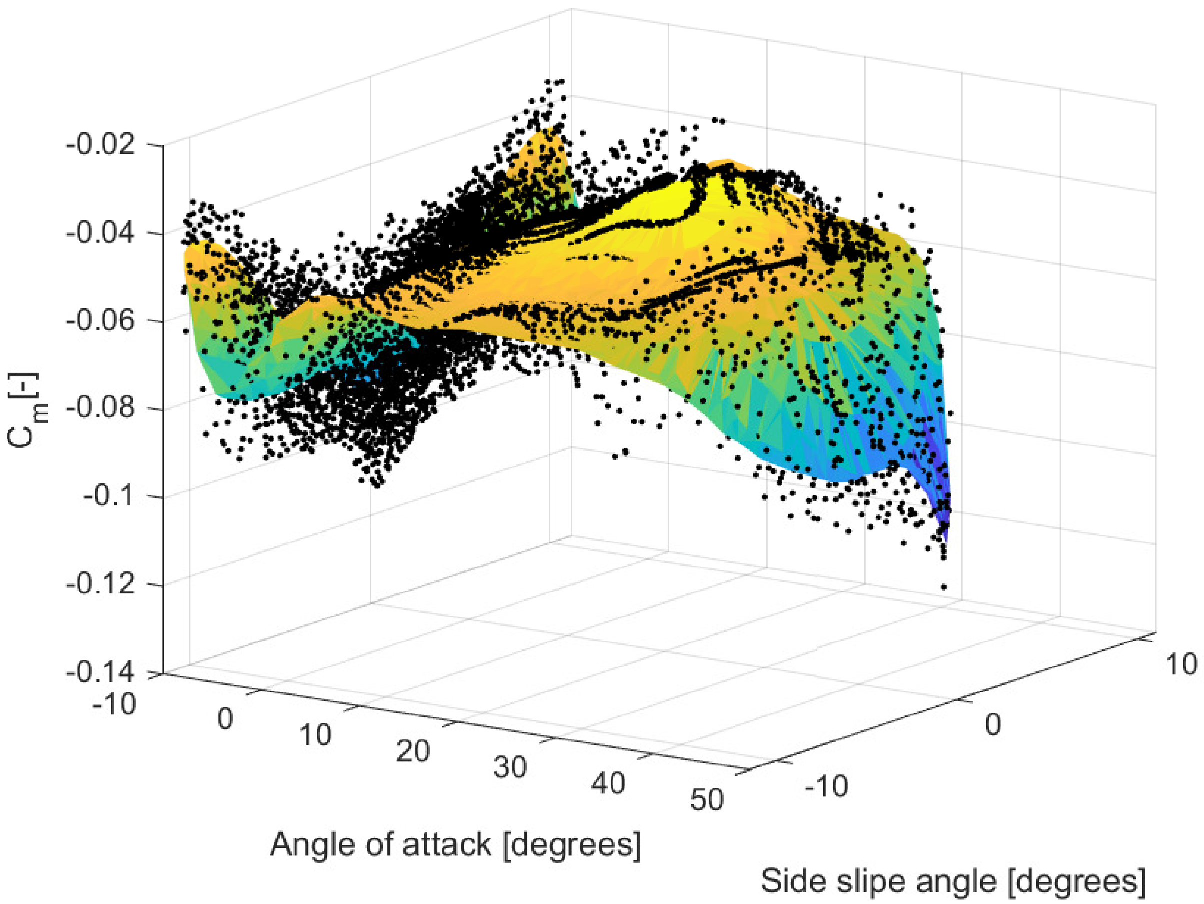

Figure 5: Estimation of the dataset with a 9th order polynomial.

Figure 6 shows the results of the model-error based validation. As can be seen in the figure the mean residual is with $-2.3243 \cdot 10^{4}$ and the auto-correlation function lies for 98.07% within the 95% confidence bounds. Figure 7 show the resutls from the statistical model quality analyses.

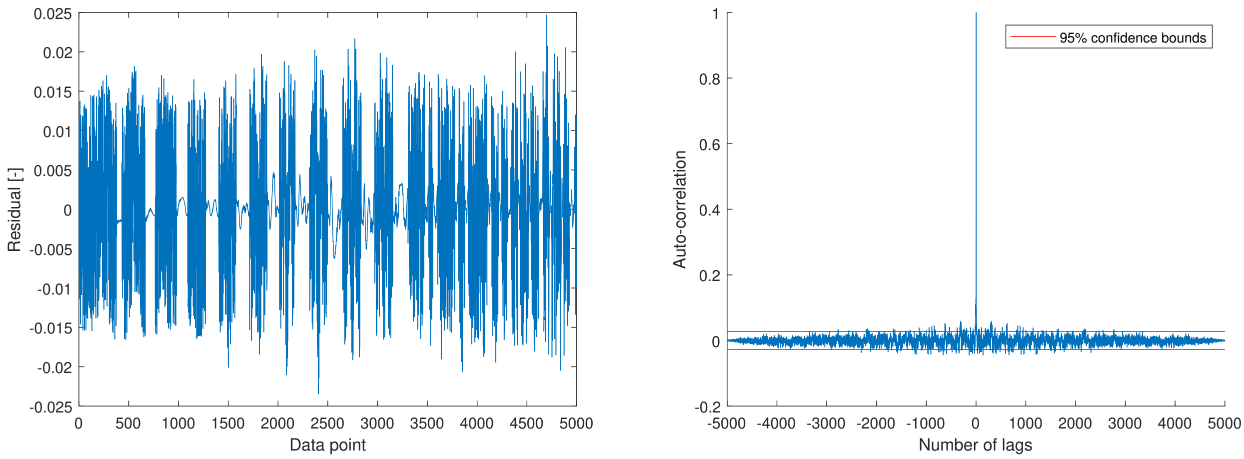

Figure 6: Model residual analysis

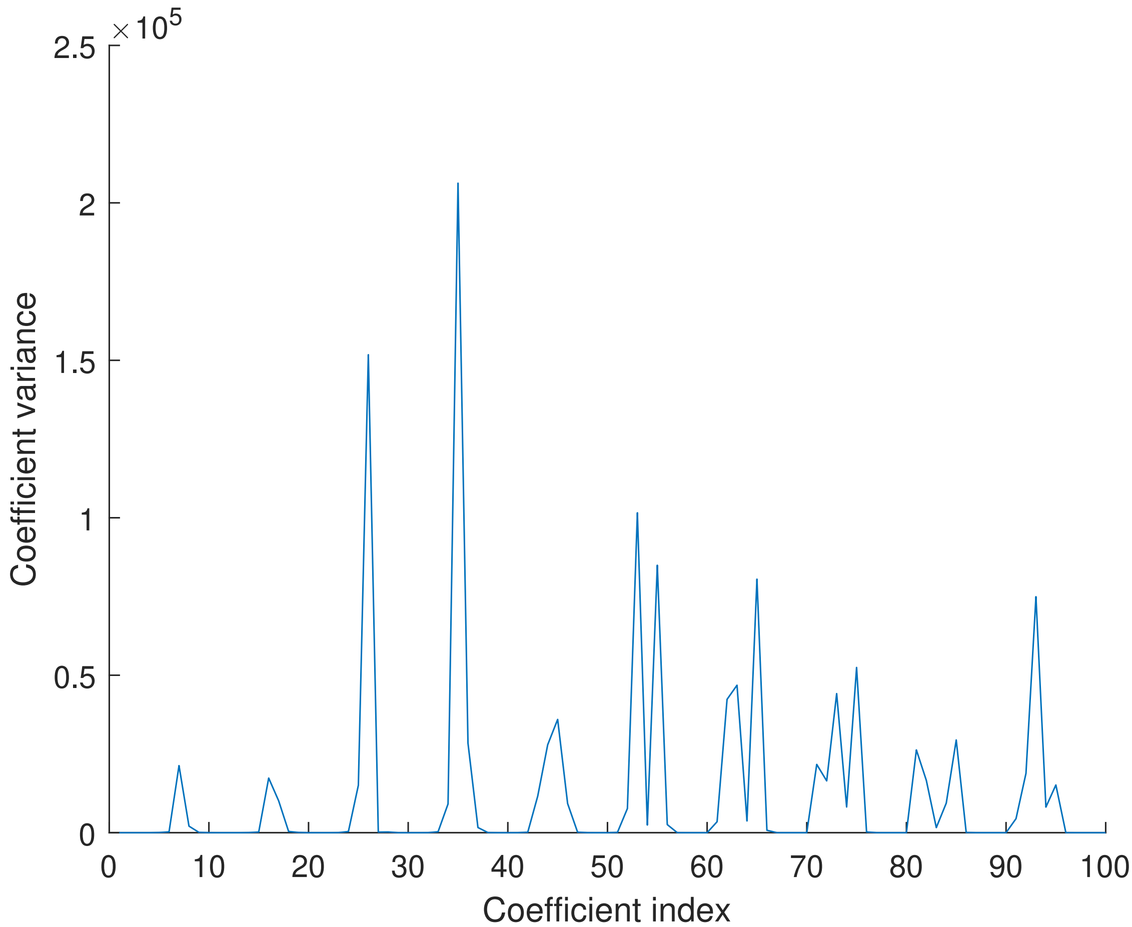

Figure 7: Statistical model quality analyses.

3 Deriving a Simplex Polynomial

The reconstructed F-16 input data shown in figure 3 can also be estimated with a single B-coefficient simplex spline. The vertices of this single 2-D simplex, which is figure 8, are placed in such a way the vertices encloses the entire dataset. According to the Boor's theorem, any polynomial $p(x)$ of degree d can be written in the B-form as shown in equation 6. With $c_K^{t_j}$ the polynomial coefficients, or B-coefficients, and with $B = (b_0, b_1, ..., b_n)$ the barycentric coordinates of x with respect to n-simplex t_j.

$$p(x) = \sum_{|K|=d} c_K^{t_j} B_k^d(b_{t_j}(x)) \tag{6}$$

Similarly to the previous section, the dataset is randomly split in half and the accuracy of fit is evaluated using the mean squared error. From the dataset, converted to barycentric coordinates, a sorted B-form regression matrix is constructed. With this matrix the coefficients of the final polynomial are calculated. The accuracy of fit for different simplex orders are shown in Figure 9 and it can be seen that a 13th order simplex estimates the dataset most accurately with a mean squared error of $5.4862 \cdot 10^{-5}$. The corresponding B-net is shown in figure 10.

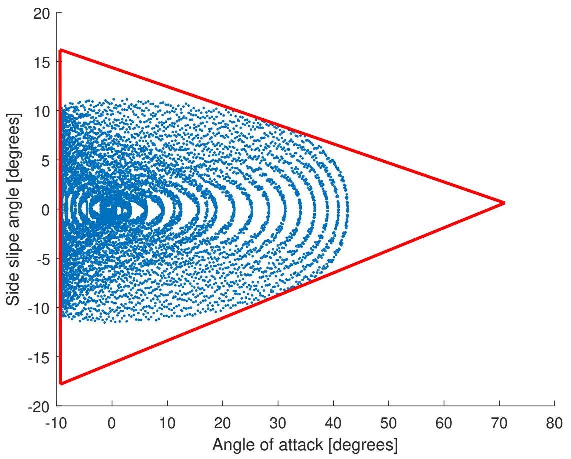

Figure 8: Single simplex enclosing entire data set.

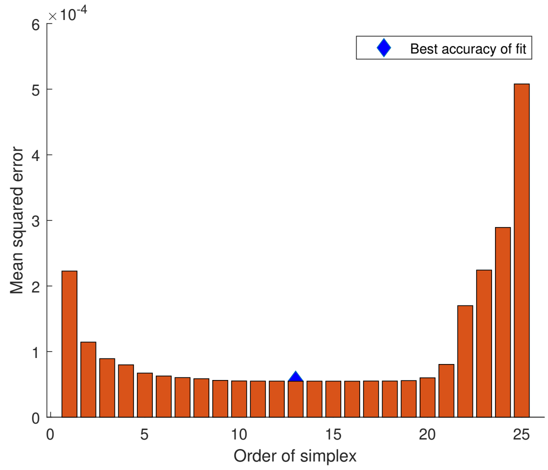

Figure 9: Mean squared error for different order polynomials.

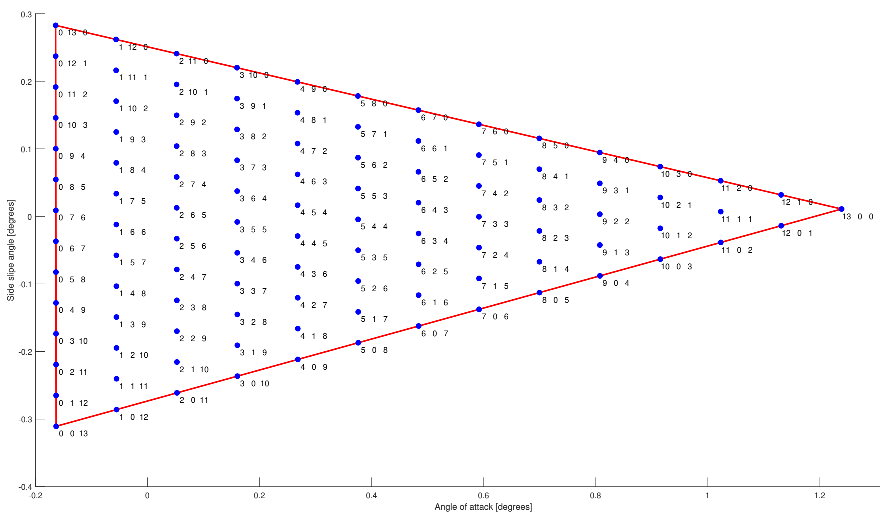

Figure 10: 13th order B-net of the single B-coefficient simplex spline that encloses the entire dataset.

With a 13th order polynomial the dataset can be estimated as shown in figure 11. The results of the model-error based validation are shown in figure 12. The mean residual of the of the estimation is $2.0859 \cdot 10^{-4}$ and the auto-correlation lies for 98.25% within the 95% confidence bounds. The results from the statistical model quality analyses can be found in figure 13.

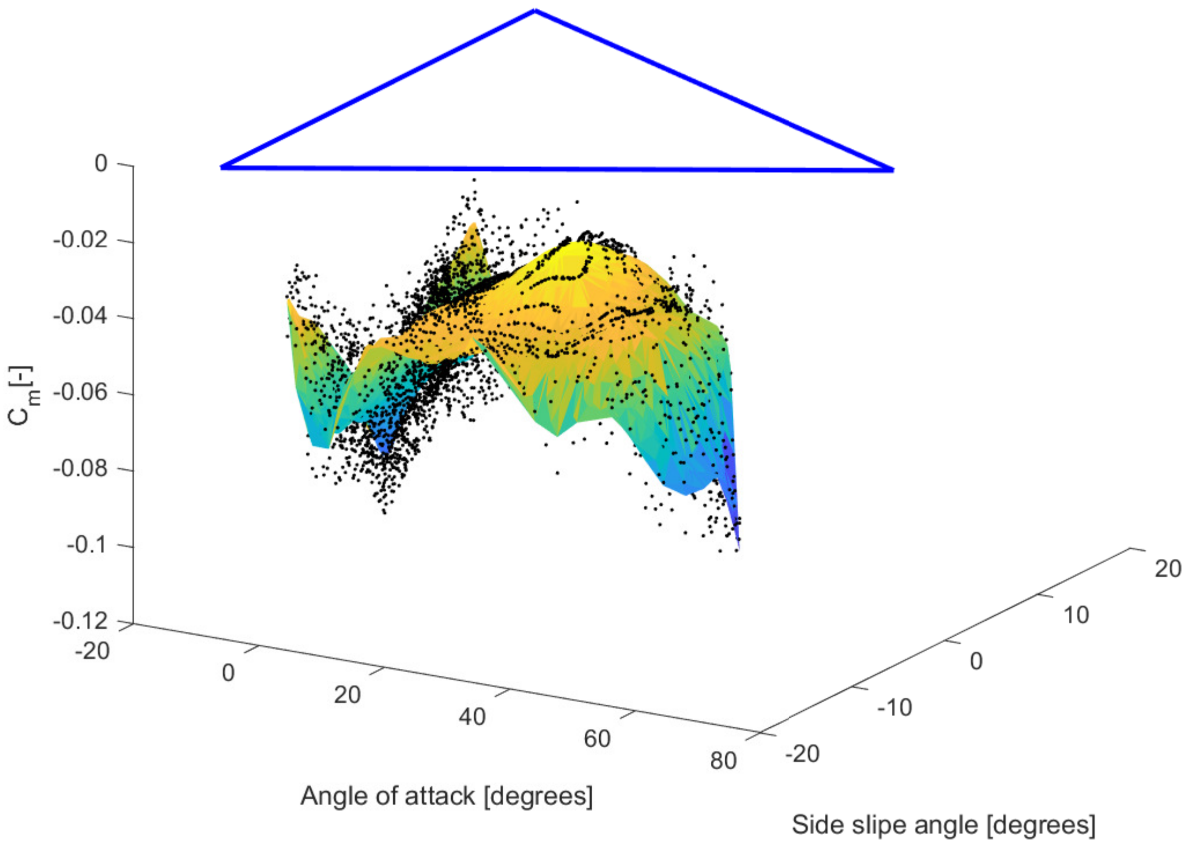

Figure 11: 13th order B-net of the single B-coefficient simplex spline that encloses the entire dataset.

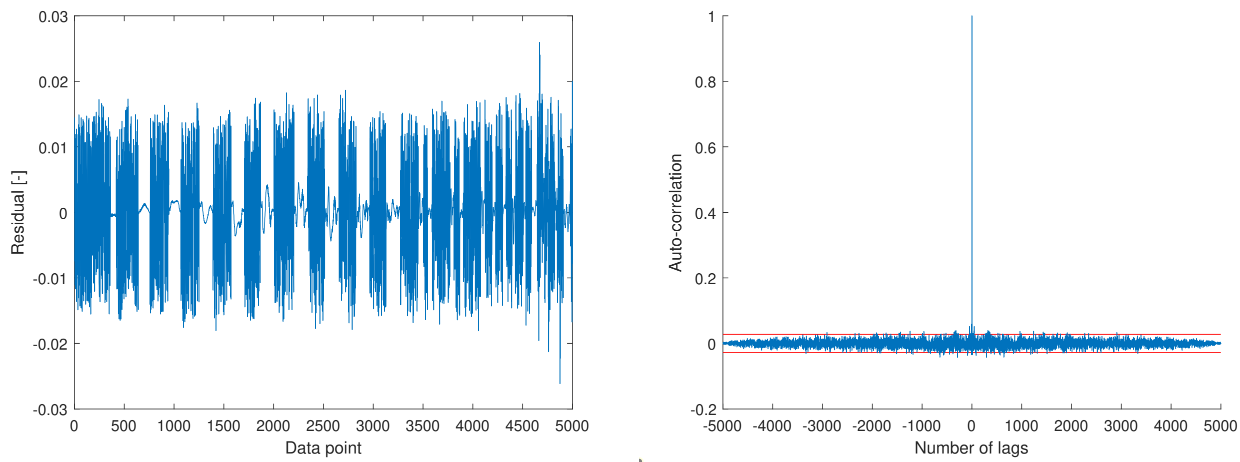

Figure 12: Model residual analysis

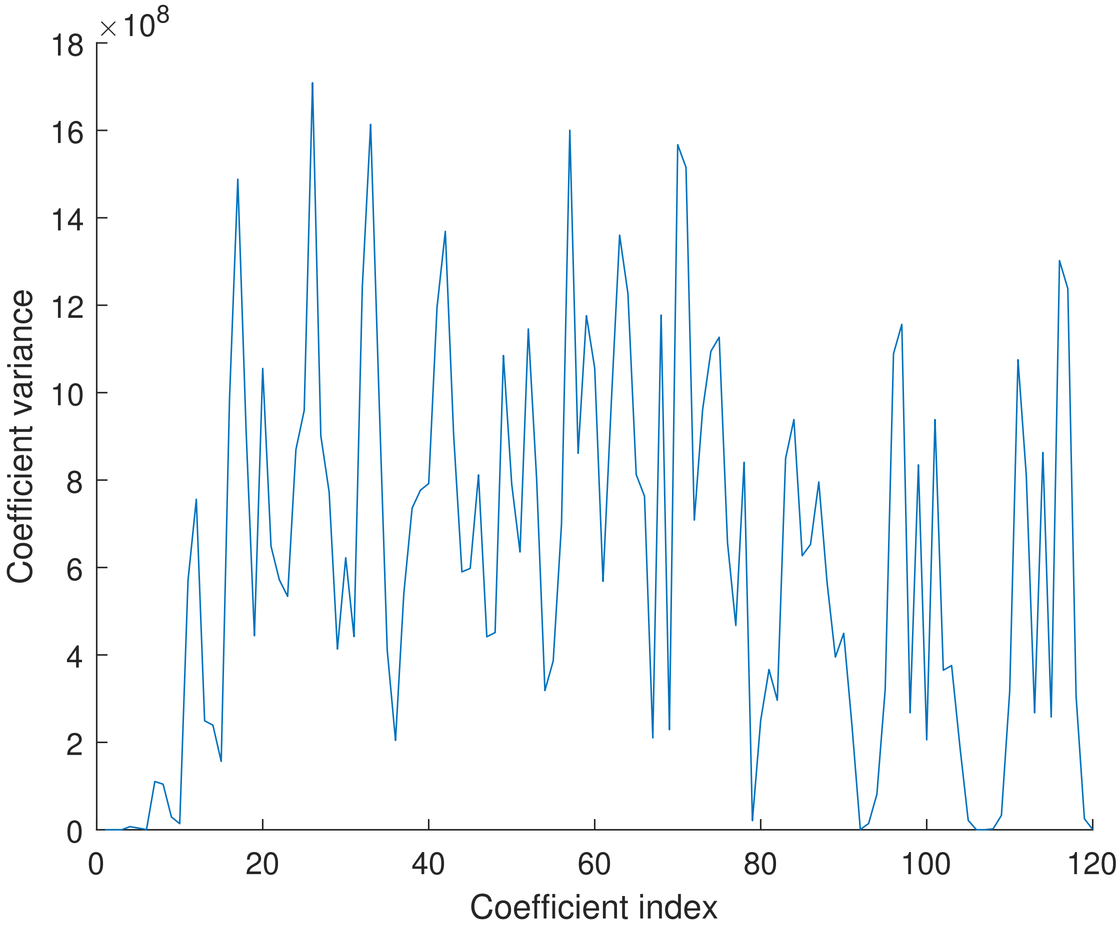

Figure 13: Statistical model quality analyses.

4 System Identification with Simplex Splines

Proceeding on the previous section, this section will estimate the dataset from figure 3 using multiple simplex splines. Hence, the vertices, from which a triangulation of the dataset can be obtained, has to be defined. In the ideal situation the triangulation is defined such that every simplex has an equal number of data points. However, this assignment does not focus on getting a most optimal triangulation and does not require a solution with a lot of simplices. Therefore, a simpler algorithm is introduced which uses K-means clustering. A rectangle is drawn in such a way it encloses the entire dataset and the four corners are taken as vertices. After this the locations of the centers of the K-means clustering algorithm are added to the vertices. with this method the triangulation is not ideal but is dependent on data complexity. Figure 14 gives an example triangulation with 10 simplices.

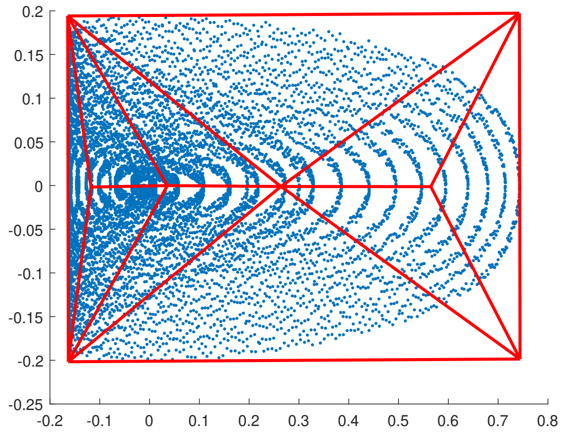

Figure 14: Triangulation with 10 simplices.

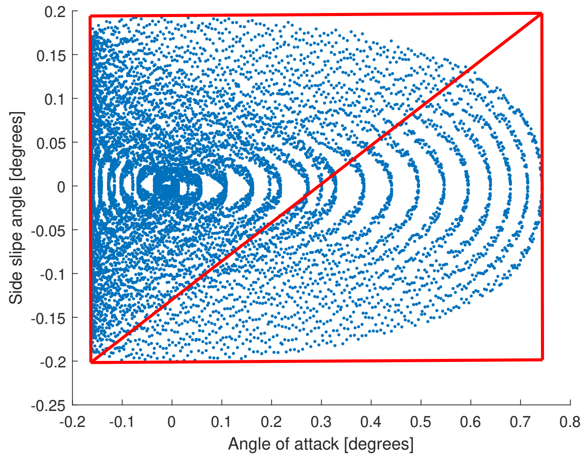

Figure 15: Most optimal triangulation.

Similarly to the previous sections the data is randomly spilt in half for identification and validation using the 'cvpartition' function from Matlab. This is done to prevent overfitting and enables the possibility to validate the model. From all the local sorted B-form regression matrices a global B-form regression matrix is constructed. After this, the 0th order continuity matrix H is constructed according to de Boor's Continuity Equations theorem which is given by equation 7.

$$c^{t_2}_{K_0,m,K_1} = \sum_{|\gamma|=m} c^{t_2}_{(K_0,0,K_1)+\gamma} B^m_\gamma(v_*) \tag{7}$$

The dataset is estimated using the Karush-Kuhn-Tucker matrix with an ordinary least squares estimator and a weighted least squares estimator. Figure 16 shows the accuracy of fit, measured with the mean squared error, for different number of simplices and different number if simplex orders for the ordinary least squares solver on the left and the weighted least squares solver on the right. As can be seen from the figure, the simplex spline is able the dataset accurately for all mid-range number of simplices and simplex orders. However, the lowest mean squared error obtained with the ordinary least squares estimator is $5.4665 \cdot 10^{-5}$ with 2 simplices and a simplex order of 10. For the weighted least squares estimator, the lowest mean squared error is $5.5025 \cdot 10{-}5$ with 2 simplices and a simplex order of 9. From these numbers it can be concluded that the ordinary least squares estimator can estimate the dataset the most accurately. The triangulation with 2 simplices is shown in figure 15 and from this the 10th order B-net is defined as shown in figure 17.

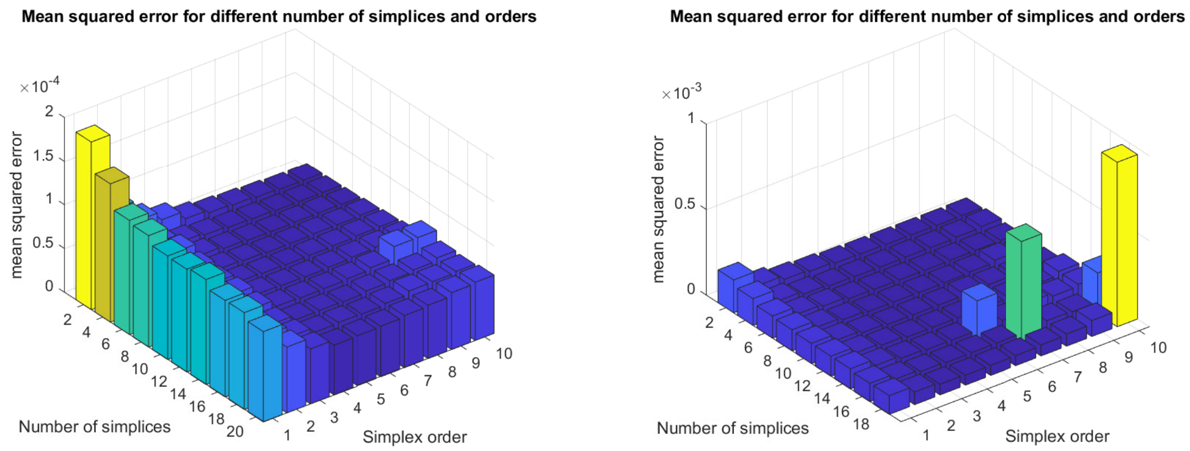

Figure 16: Accuracy of fit for different number of simplices and simplex order obtained with the ordinary least squares estimator (left) and the weighted least squares estimator (right).

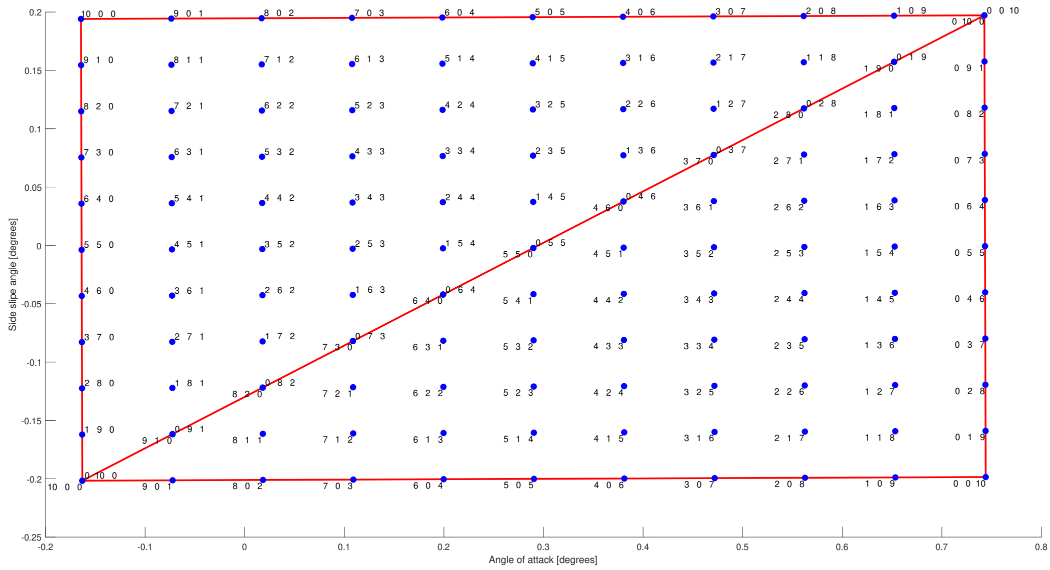

Figure 17: Most optimal B-net

With the system identification framework for multivariate splines implemented, the model is able to identify the multivariate spline based aerodynamic model for the F-16 given the flight data. This is done with a B-coefficient simplex spline consisting of 2 simplices from the 10 order with a 0th order continuity. The output of the model is shown in figure 18. As can be seen in the figure, but better visualised in Matlab, the polynomial has 0th order continuity. This 0th order continuity can be proven by the spy plot shown in fig 19. Looking at the equality constraints of the smoothness matrix, it can be seen that the vertices placed on the mutual edge have an equality relationship with only one vertex from the other simplex.

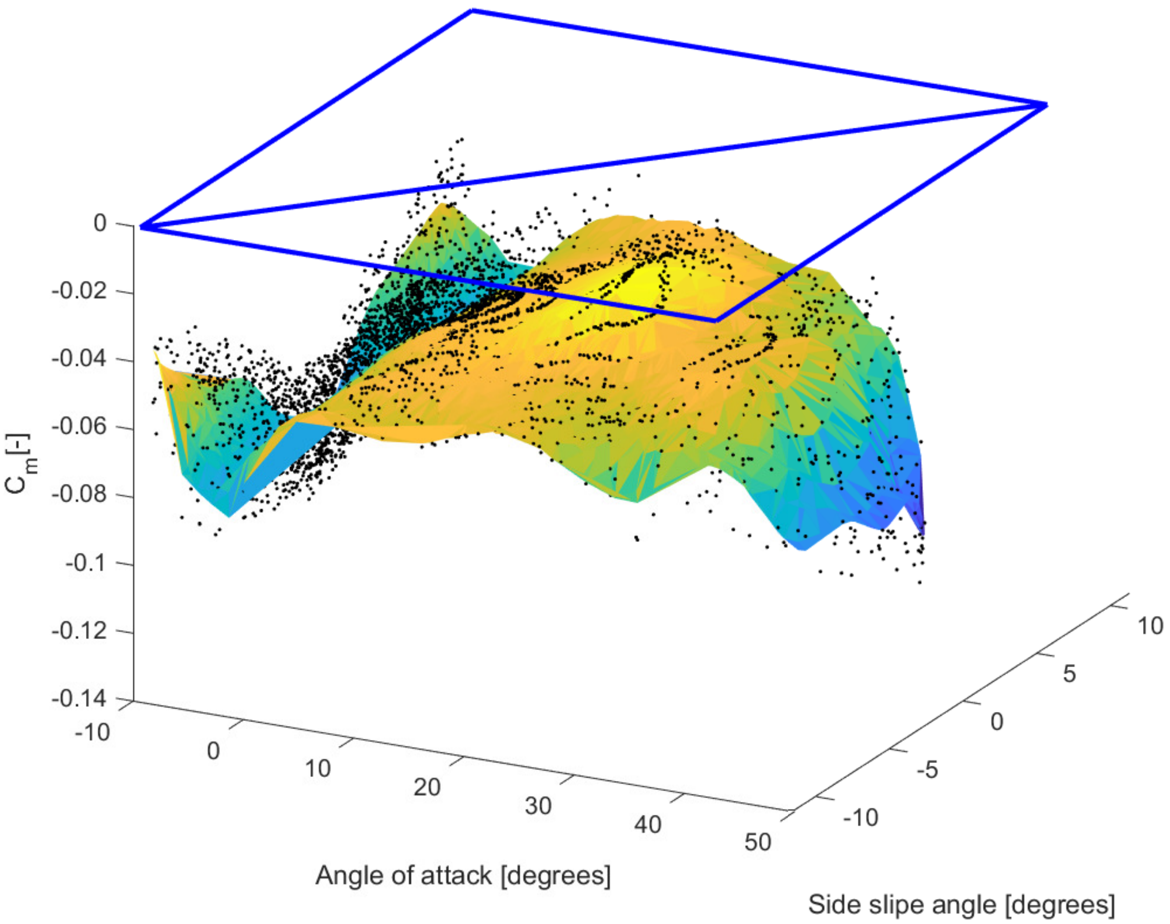

Figure 18: Simplex spline with 2 10th order simplices and 0th order continuity.

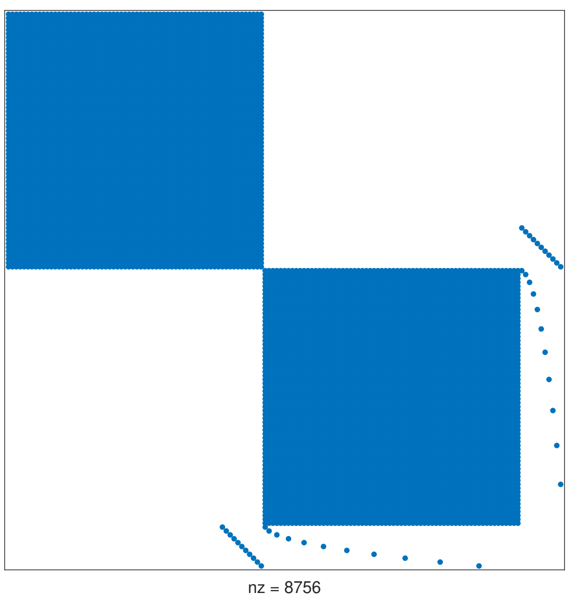

Figure 19: Simplex spline with 2 10th order simplices and 0th order continuity.

The results of the model-error based validation are shown in figure 20. The mean residual of the estimation is $-1.8365 \cdot 10^{-4}$ and the auto-correlation function lies for 98.47% within the 95% confidence bounds. In figure 21 the results from the statistical model quality analyses are shown.

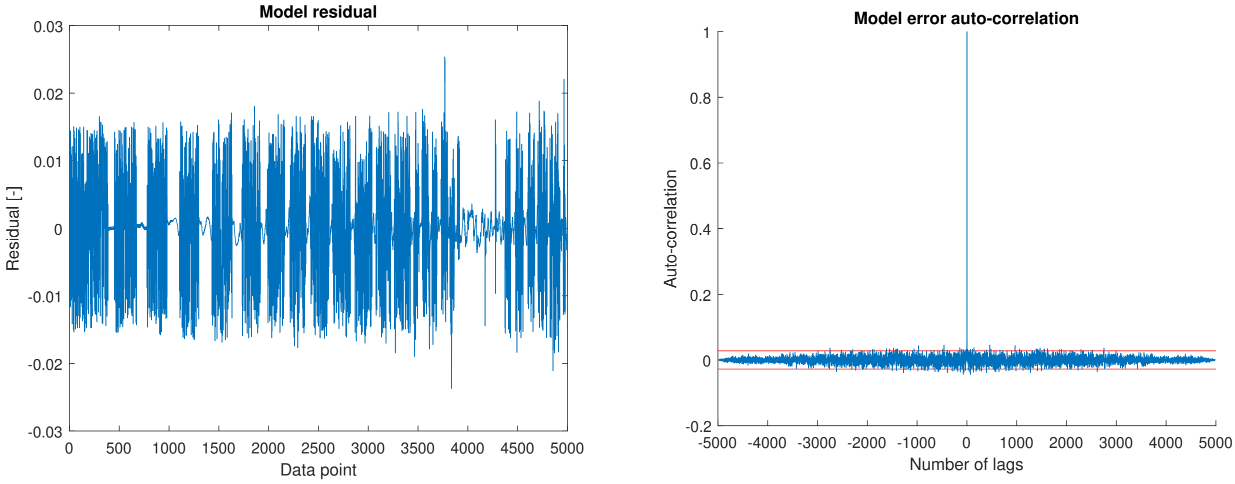

Figure 20: Model residual analysis

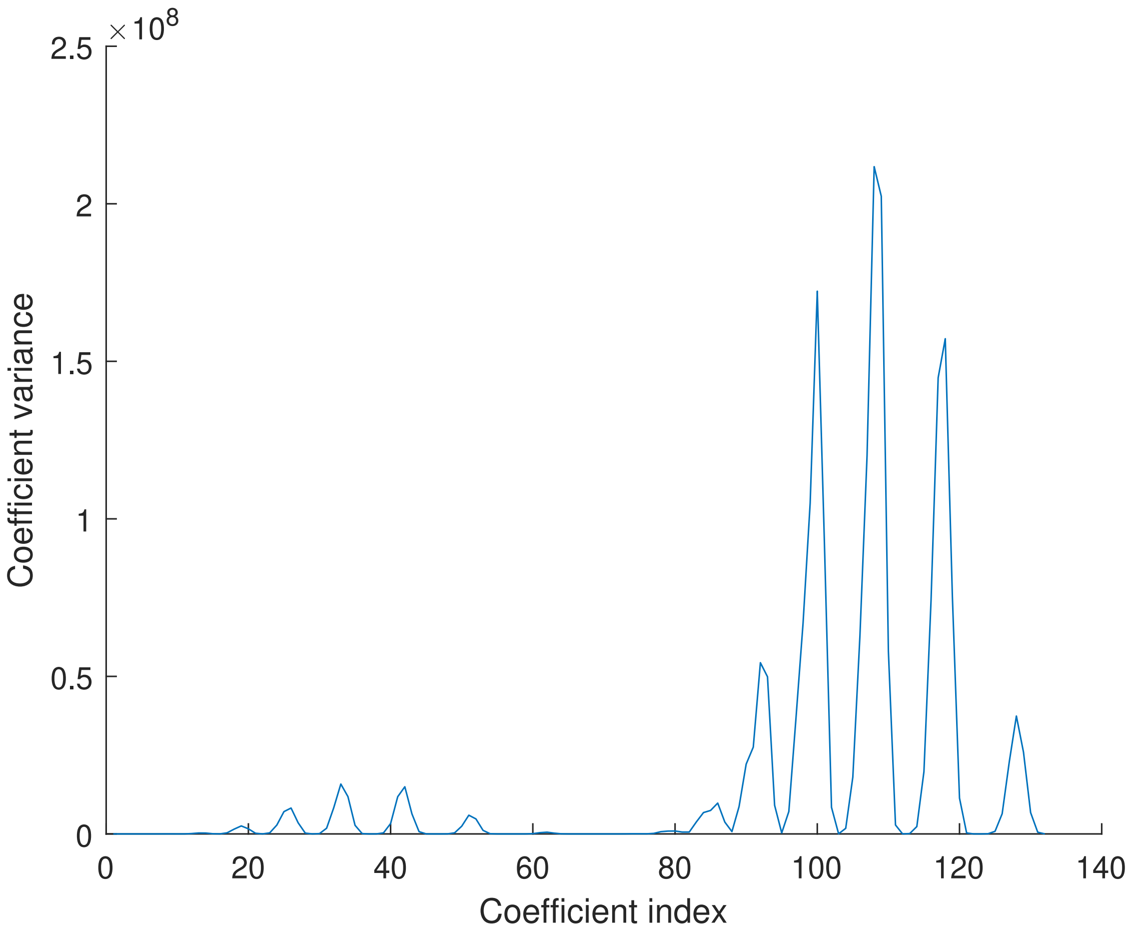

Figure 21: Statistical model quality analyses.

Conclusion

In this report, multiple techniques to estimate a non-linear and noisy wind tunnel dataset of the F-16. First, a bias in the angle of attack and the true states were identified using an Extended Kalman Filter. With this reconstructed flight data a simple polynomial structure was created for estimation. It is shown that a 9th order polynomial estimates the dataset the most accurately. After this, the same flight data was estimated using a single simplex B-coefficient spline. In this case, a 13th order simplex estimated the dataset most accurately. At last, a simplex B-coefficient spline with multiple simplices and a 0th order cotinuity was used for estimation. It was shown that using only 2 simplices both with a 10th order gave the best performance. furthermore, it was shown that the ordinary least squares estimator had a higher performance compared to the weighted least squares estimator.

For all the three estimation techniques the accuracy of fit was tested using the mean squared error. Furthermore a residual validation an a statistical model quality analyses is conducted. Table 1 shows the final results for all estimation techniques. As can be seen, the simplex spline has the best accuracy of fit, a mean residual the closest to zero and the highest 95% confidence bounds. Therefore, the simplex spline is the most accurate method to estimate the reconstructed flightdata. This is an expected conclusion as it is also the most complex method. While the other two methods are considerable less complex, they also estimate the dataset very accurately.

Model structure	Mean squared error	Mean residual	95% confidence bounds
Simple polynomial	$5.7754 \cdot 10^{-5}$	$-2.3243 \cdot 10^{-4}$	98.07%
Single simplex	$5.4862 \cdot 10^{-5}$	$-2.0859 \cdot 10^{-4}$	98.25%
Simplex spline	$5.4665 \cdot 10^{-5}$	$-1.8365 \cdot 10^{-4}$	98.47%

Table 1: Final results for the simple polynomial, the single simplex and the simplex spline.