System Identification of Aerospace Vehicles - AE4320

MULTIVARIATE SIMPLEX SPLINES

MOHAMMED AADIL AHMED-5200695

Contents

Chapter 1

Introduction

1.1 System identification

1.1.1 Principle

System identification is the branch of control systems engineering that uses tools to identify mathematical models of a system given experimental input-output data. Some tools employ better identification of the model if we have theoretical understanding of the system.

The mathematical models that can be identified can either be models that fit the input-output data, time-domain relationships or even frequency-domain relationships.

1.1.2 Relevance to aircraft control and simulation

System identification can be employed to model complex mathematical models of aircraft systems which can be difficult to determine by available physical laws or empirical relationships. Once the mathematical model has been identified, it can be used for designing the flight control laws.

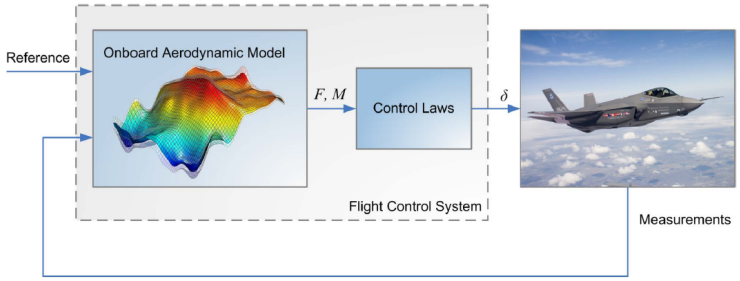


Figure 1.1: system identification of aerospace model (from lecture slides)

As it is seen in the figure 1.1 how an identified aerodynamic model can help in the development of flight control laws for an aircraft.

Another application is to model the behavior of a pilot and make an equation that mimics a real pilot's response while flying the aerospace vehicle. This is done for simulators use typically.

1.2 State and parameter estimation

1.2.1 State estimation

There are noise present inherently with any measurement involving physical sensors. The noise is unavoidable in most scenarios. Incorrect values of states can greatly effect flight control laws such as state-feedback controllers. State estimation techniques such as Kalman-filter are able to give a better estimate of the states by having statistical knowledge of the noise such as mean and variance.

State estimation also solves the issue when states cannot be measured directly with physical sensors and some state estimates can be improved by using multiple sensors.

Based on the linearity of the system equations either a simple linear Kalman filter can be used or if its a highly non-linear system then an extended Kalman filter or Iterated Extended Kalman filter can be used.

1.2.2 Parameter estimation

Parameter estimation is a technique widely used in all fields of science and engineering which makes it possible to determine an optimal set of model parameters given the set of measurements. Based on the limited measurements from tests, it is possible to extrapolate to a the model structure that best fits the data. This model can then be used to employ in other tasks such as development of flight control laws as mentioned previously.

The choice of model structure takes place before parameter estimation can take place. The possible model structures are: polynomial models, multivariate spline models, neural networks etc. Once the model structure is chosen, then the optimal parameters are estimated using an estimator such as Ordinary Least squares, Weighted Least squares etc in order to minimize the error.

1.3 Advanced System Identification methods

These are typically non-linear identification methods employed for creating complex models when simpler identification methods don't work well. **Neural networks** and **Multivariate simplex B-splines** are such techniques.

Neural Networks are network structures where neurons are connected to other neurons that map the inputs to outputs. The network would recognise patterns between the input values in order to map to correct outputs. These are non-linear structures contain weighted sum of activation functions. Changing the weights essentially shapes the neural network and to fit the data, an optimal set of weights need to be calculated.

Multivariate simplex B-splines have some advantages and disadvantages over neural networks and standard polynomials. It has a stable local basis compared to a global basis function in standard polynomials. Another advantage is that its continuity order can be chosen. Firstly, Cartesian coordinates are converted to barycentric coordinates to get a B-form polynomial. A linear method such as ordinary least squares can be used to identify the B-coefficients of the B-form polynomial.

Chapter 2

State & Parameter estimation with F-16 flight data

2.1 State estimation with Kalman Filter

2.1.1 Formulating the system and output equations

The system equation $\dot{x} = f(x_k, u_k, t)$ is written in a linear form as the state derivatives $(\dot{u}, \dot{v}, \dot{w})$ are perfectly measurable. The up-wash coefficient being a constant will have dynamics described by $\dot{C}_{\alpha_{up}} = 0$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{C}_{\alpha_{up}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ C_{\alpha_{up}} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} \tag{2.1}$$

The system equation 2.1 turns out to be linear with a null state matrix F. The output equations $z = h(x_k, u_k, t)$ are non-linear and are given by:

$$\begin{bmatrix} \alpha_m \\ \beta_m \\ V_m \end{bmatrix} = \begin{bmatrix} tan^{-1}\left(\frac{w}{u}\right)(1 + C_{\alpha_{up}}) \\ \tan^{-1}\left(\frac{v}{\sqrt{u^2 + w^2}}\right) \\ \sqrt{u^2 + v^2 + w^2} \end{bmatrix} \tag{2.2}$$

Linearising the nonlinear output equation 2.2 we obtain:

$$\begin{bmatrix} \alpha_m \\ \beta_m \\ V_m \end{bmatrix} = \begin{bmatrix} \frac{-w^*(1 + C^*_{\alpha_{up}})}{\sqrt{u^{*2} + w^{*2}}} & 0 & \frac{(1 + C^*_{\alpha_{up}})}{\sqrt{u^{*2} + w^{*2}}} & tan^{-1}\left(\frac{w^*}{u^*}\right) \\ \frac{-3u^*v^*}{(u^{*2} + v^{*2} + w^{*2})\sqrt{u^{*2} + w^{*2}}} & \frac{\sqrt{u^{*2} + w^{*2}}}{(u^{*2} + v^{*2} + w^{*2})} & \frac{-3w^*v^*}{(u^{*2} + v^{*2} + w^{*2})\sqrt{u^{*2} + w^{*2}}} & 0 \\ \frac{u^*}{\sqrt{u^{*2} + v^{*2} + w^{*2}}} & \frac{v^*}{\sqrt{u^{*2} + v^{*2} + w^{*2}}} & \frac{w^*}{\sqrt{u^{*2} + v^{*2} + w^{*2}}} & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ C_{\alpha_{up}} \end{bmatrix}$$

2.1.2 Motivation of type of Kalman filter

Since the output equation is nonlinear, the Extended Kalman filter or Iterated Extended Kalman filter can be used. A good guess of the initial states can be given based on looking at the data provided. The first element of α_m is equal to 0 $\implies$ $\alpha_{m_0} = 0$ $\implies$ $tan^{-1}\left(\frac{w_0}{u_0}\right) = 0$ $\implies$ $w_0 = 0$. Similarly, $\beta_{m_0} = 0$ $\implies$ $v_0 = 0$ and $V_{m_0} = 150$ $\implies$ $\sqrt{u_0^2 + v_0^2 + w_0^2} = 150$ $\implies$ $u_0 = 150$

By looking at the formula $\alpha_m = tan^{-1}\left(\frac{w}{u}\right)(1 + C_{\alpha_{up}})$ the up-wash coefficient causes a percentage increase in the angle of attack values. This percentage increase comes approximately as 30% based on information in [1]. So, initial value of $C_{\alpha_{up}}$ equal to 0.3 would be a good guess.

2.1.3 Kalman Filter

Following are the results from the Kalman filter (code available in my_EKF1.m):

Figure 2.1 contains the reconstructed states which result from the implementation of the filter. Figure 2.2 contains the standard deviation of P_correlation that describes the the state estimation error. This converges

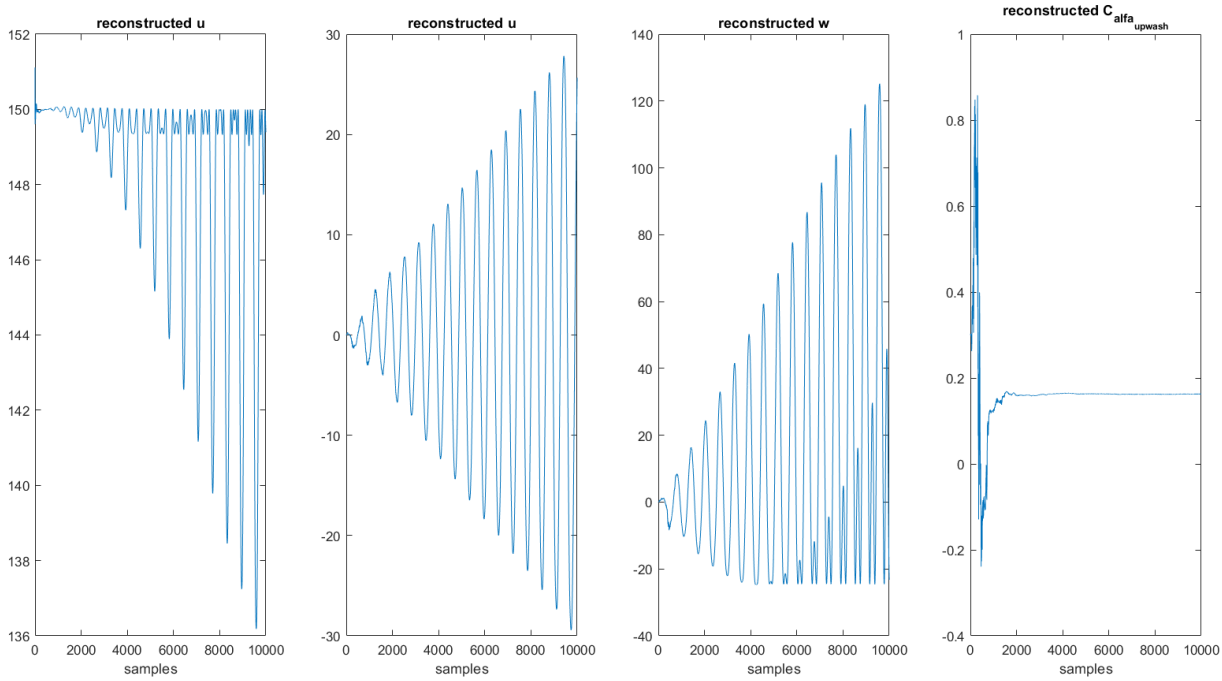


Figure 2.1: Reconstruction of states

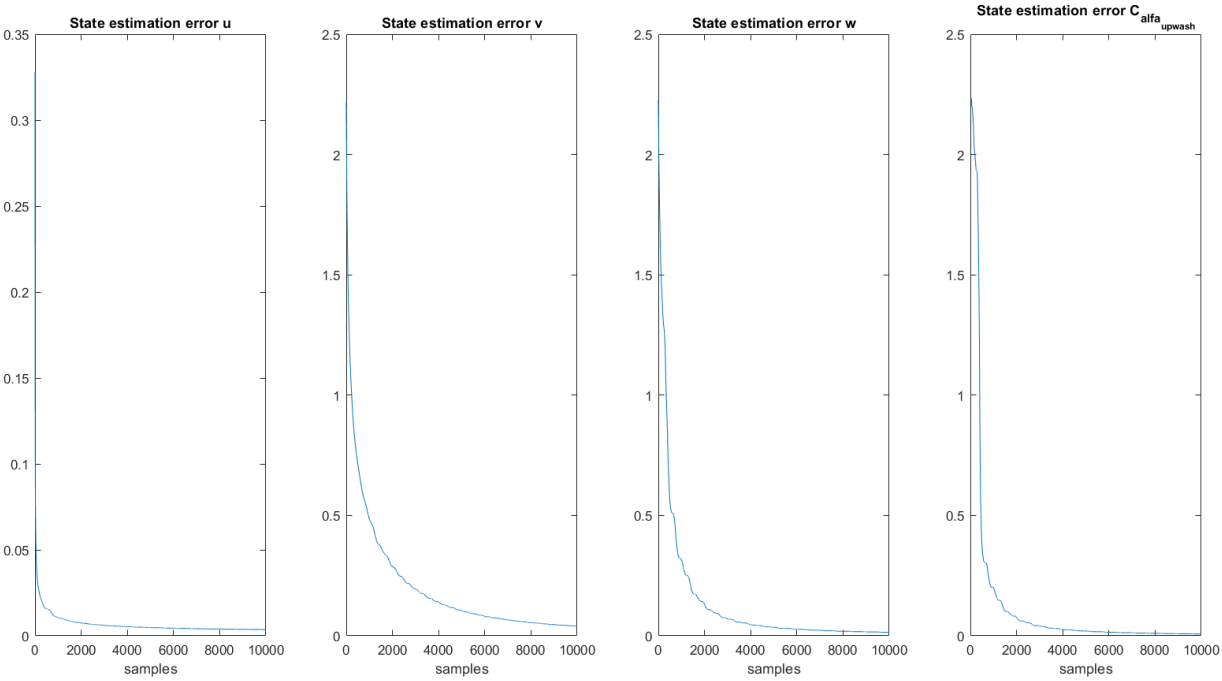


Figure 2.2: States error

fast towards the end of sample window as seen from the plot. The estimate of $C_{\alpha_{up}}$ turns out to be 0.1624 . This is taken by averaging the second half of the $C_{\alpha_{up}}$ vector as the filter has converged by then.

2.1.4 Reconstruction of α_{true}

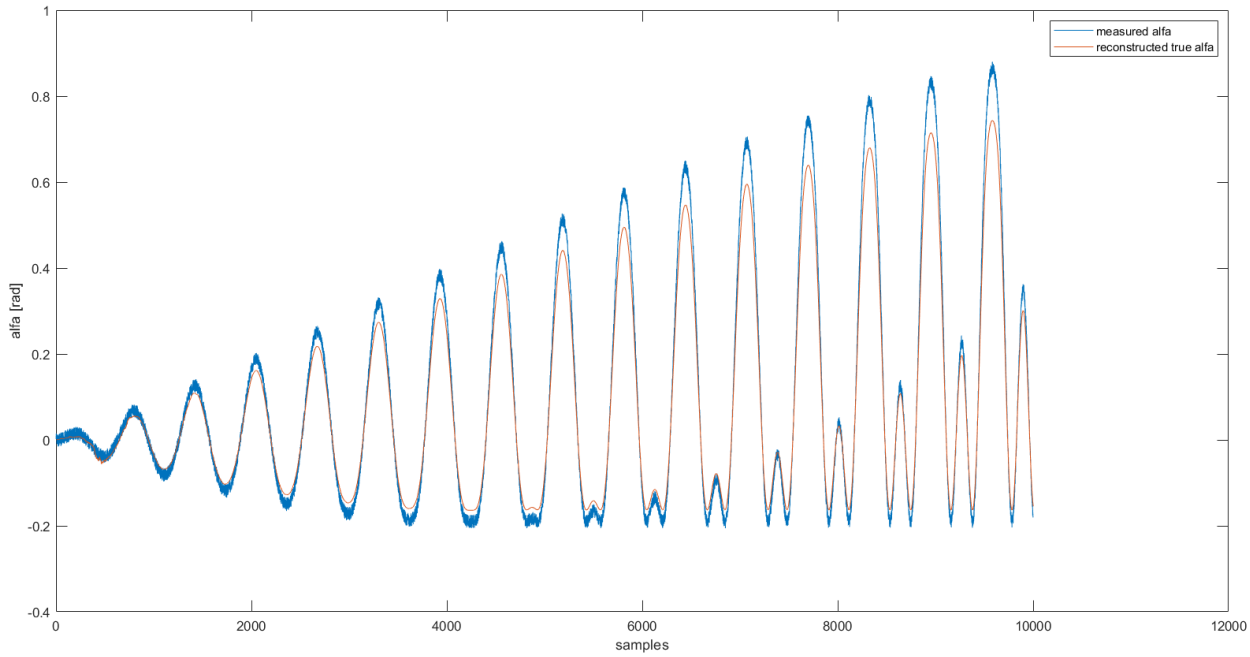


Figure 2.3: Comparison of α_{true} and α_m

As seen in figure 2.3 the reconstructed α_{true} is very similar to α_m in the initial samples, and as the samples progress, the estimates of u and v are improved, thereby improving the estimate of α_{true}.

2.2 Parameter estimation with Ordinary least squares

2.2.1 Formulation of the Least squares

In this section an ordinary Least squares method will be used in order to estimate the parameter C_m using α and β. The model structure chosen is a polynomial model such as:

$$C_m(\alpha, \beta) = c_0 + c_1\alpha + c_2\beta + c_3\alpha\beta + c_4\alpha^2 + c_5\beta^2 + \cdots + c_t\beta^n \tag{2.3}$$

2.2.2 Data sorting

The α_{true} values will be taken from this section onwards instead of α_m. The α_{true},β_m and C_m are vectors containing $N \times 1$ elements. The vectors are randomly permutated and 70% elements (N_{tr}) will be used for the purpose of training while the remaining 30% (N_{val}) for validation.

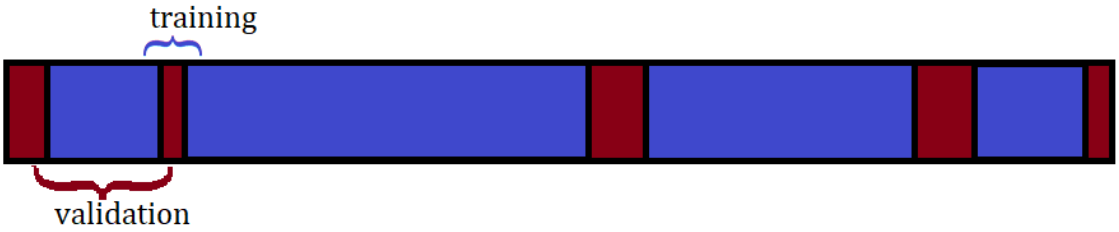


Figure 2.4: Division of training and validation in a vector

2.2.3 Creating the regression matrix

First a column vector of ones with length N_{tr} is created, the a matrix $\begin{bmatrix} \alpha(1) & \beta(1) \\ \vdots & \vdots \\ \alpha(N_{tr}) & \beta(N_{tr}) \end{bmatrix}$ is created, subsequently the matrix $\begin{bmatrix} \alpha^2(1) & \alpha(1)\beta(1) & \beta^2(1) \\ \vdots & \vdots & \vdots \\ \alpha^2(N_{tr}) & \alpha(N_{tr})\beta(N_{tr}) & \beta^2(N_{tr}) \end{bmatrix}$ and so on up to $\begin{bmatrix} \alpha^n(1) & \alpha^{n-1}(1)\beta^1(1) & \alpha^{n-2}(1)\beta^2(1) & \dots & \beta^n(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha^n(N_{tr}) & \alpha^{n-1}(N_{tr})\beta^1(N_{tr}) & \alpha^{n-2}(N_{tr})\beta^2(N_{tr}) & \dots & \beta^n(N_{tr}) \end{bmatrix}$ is created. Then all the matrices are concatenated in order to get the **Regression Matrix**:

$$A = \begin{bmatrix} 1 & \alpha(1) & \beta(1) & \alpha^2(1) & \alpha(1)\beta(1) & \beta^2(1) & \dots & \alpha^n(1) & \dots & \beta^n(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha(N_{tr}) & \beta(N_{tr}) & \alpha^2(N_{tr}) & \alpha(N_{tr})\beta(N_{tr}) & \beta^2(N_{tr}) & \dots & \alpha^n(N_{tr}) & \dots & \beta^n(N_{tr}) \end{bmatrix}$$

The parameters vector is $\theta = \begin{bmatrix} p_1 & p_2 & p_3 & \dots & p_k \end{bmatrix}^T$ where $k = \dfrac{(n+1)(n+2)}{2}$ and n is the order of the polynomial.

2.2.4 Estimating the coefficients

The set of optimum coefficients will be the one that minimizes the cost-function $J = (C_m - A.\theta)^2$. This is done by applying the formula:

$$\hat{\theta} = (A^T A)^{-1} A^T C_m \tag{2.4}$$

The parameters are estimated with orders ranging from 1 to 8. Higher orders 7 onwards make it inefficient to estimate the parameters since the matrix $\Phi = A^T A$ becomes rank deficient so the inverse of Φ doesn't exist.

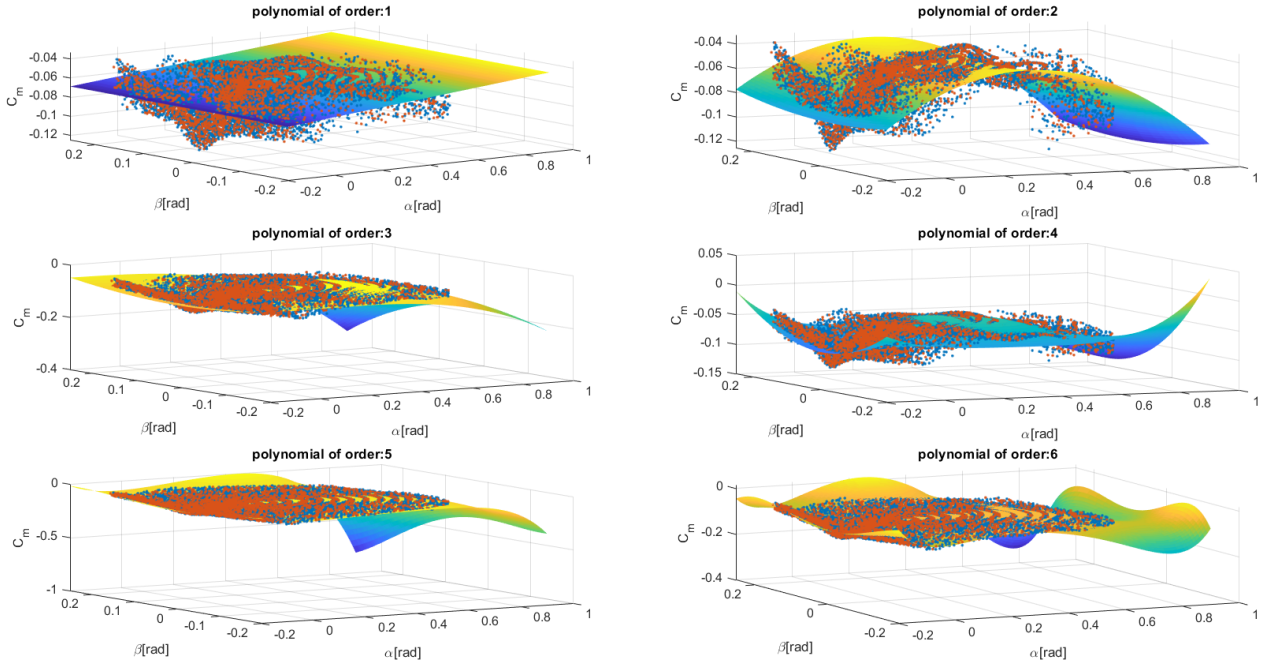


Figure 2.5: polynomial from order 1 to 6

Figure 2.5 contains the different polynomial orders whose coefficients have been estimated by using the equation 2.4. In this figure the training data and validation data are represented by red and blue dots respectively.

2.2.5 Model Validation using Residual analysis

Residual is given by the difference between the polynomial output evaluated at validation points $(C_m(\alpha_{val}, \beta_{val}))$ and validation data $(C_{m_{val}})$

$$e = C_{m_{val}} - C_m(\alpha_{val}, \beta_{val})$$

The relative RMS residual is given by $RMS_{res_{rel}} = \frac{RMS(e)}{max(C_{m_{val}}) - min(C_{m_{val}})}$), where $RMS(e)$ is the root mean square of e.

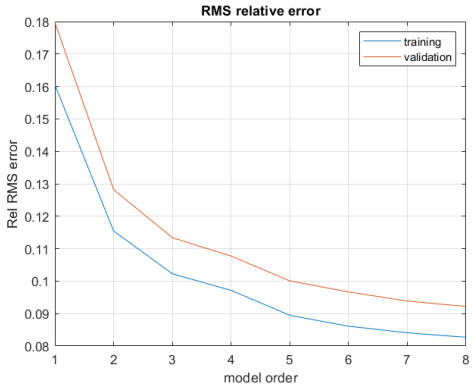


Figure 2.6: Relative RMS of residual order 1 to 8

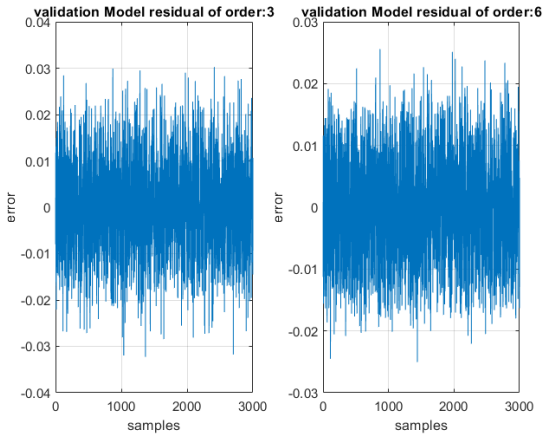


Figure 2.7: Residual comparison of order 1 and 6

Figure 2.6 shows the variation of relative RMS residual as the model order increases. The Relative RMS significantly drops after model order = 3 onwards. Figure 2.7 shows the comparison of residuals using a 3^{rd} and 6^{th} order polynomial.

Figure 2.8 shows the autocorrelation of the model residuals for degree 1 and degree 6, increasing the order reduces the autocorrelation away from the bounds of 96% confidence interval.

2.2.6 Model Validation using statistical analysis

The parameter variances can be obtained by the co-variance matrix which is given by $Cov(\hat{\theta}) = (A^T.A)^{-1}$.

The matrix diagonal elements represent the parameter variances while the off-diagonal elements represent the parameter co-variances. Since the variances increase to high order of magnitudes as the model order increases, a logarithmic of variance is taken in order to observe the trend of variances as shown in figure.

$$\log_{10}(\overline{Var(\theta)} = \log_{10}(\frac{1}{N}\sum_{i} Var(\theta)))$$

2.2.7 Comparison with MATLAB polynomial fit

The coefficients of the polynomial estimated by the least squares created in this section is compared with MATLAB's **fit** command. The difference between the coefficients is checked at different orders and are found to be very insignificant. One such plot of the difference between coefficients for a polynomial of order 3 is shown in figure 2.10.

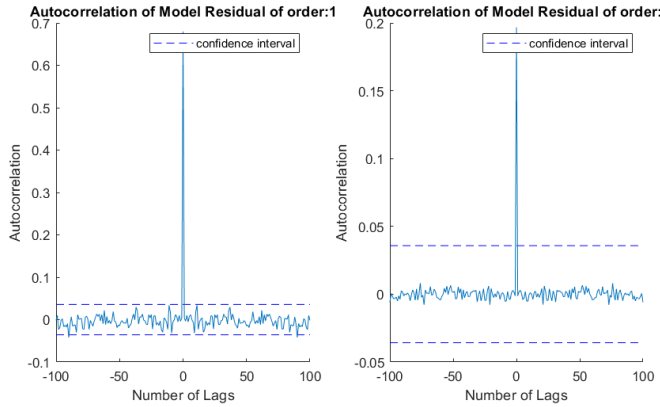


Figure 2.8: Resdiual Autocorrelation : Least Squares

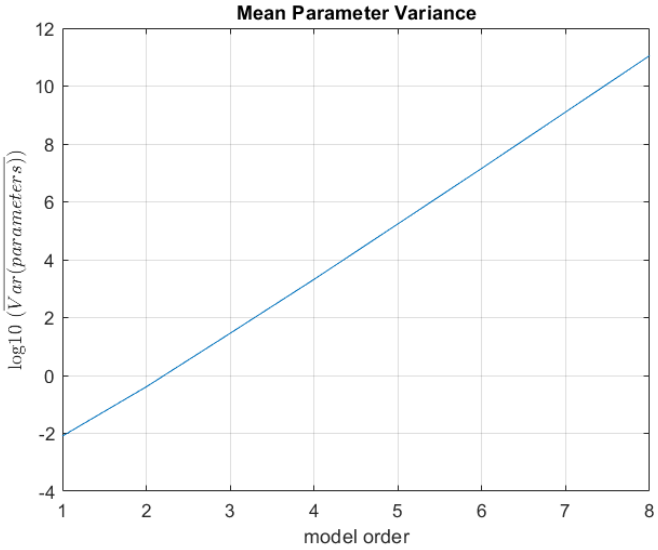


Figure 2.9: Parameter variances order 1 to 8

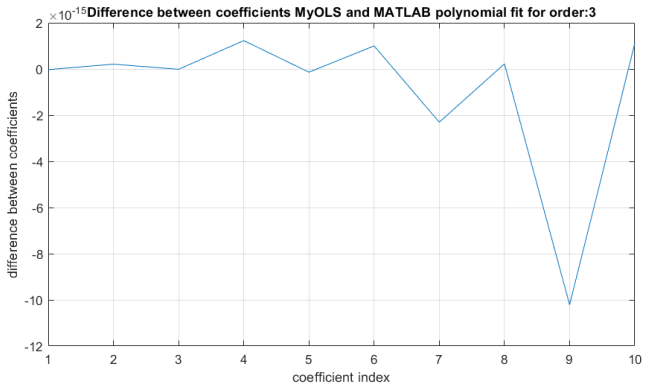


Figure 2.10: Difference between MATLAB coefficients and my coefficients

Chapter 3

Deriving a Simplex polynomial

A simplex spline polynomial is defined on a geometrical structure (simplex). A simplex is a geometrical structure that minimally spans a set of dimensions. Multiple simplices can be used together locally to identify the model. But in this section only a single simplex will be used. An n-dimensional simplex has n+1 vertices. A 2 dimensional simplex will be used (a triangle occupying the 2 dimensional space).

3.1 Simplex Polynomial

The **Barycentric coordinates** is a set of weights assigned to the vertices of the simplex in order to make the center of gravity same as the Cartesian coordinates. A simplex is defined by the Barycentric coordinates.The transformation from Cartesian to Barycentric coordinates is a *linear transformation.* For an n-dimensional simplex:

$$x = \sum_{i=0}^{n} b_i v_i \quad and \quad \sum_{i=0}^{n} b_i = 1 \tag{3.1}$$

where x: Cartesian coordinate and b: Barycentric coordinate $(b_0, b_1, \ldots, b_n)$

The simplex polynomials are expressed in terms of Barycentric coordinates:

$$p(b_0, b_1, \ldots, b_n) = (b_0 + b_1 + \cdots + b_n)^d \tag{3.2}$$

where d is the degree of polynomial

3.2 Basis Function for general polynomial degrees

For convenience the equation 3.2 can be re-written with Basis function by using the Multinomial Theorem :

$$p(b_0, b_1, \ldots, b_n) = \sum_{\kappa_0+\kappa_1+\cdots+\kappa_n=d} \frac{d!}{\kappa_0!\kappa_1!\ldots\kappa_n!} b_0^{\kappa_0} b_1^{\kappa_1} \ldots b_n^{\kappa_n} \tag{3.3}$$

$$|\kappa| = \kappa_0 + \kappa_1 + \cdots + \kappa_n = d \quad and \quad \kappa! = \kappa_0!\kappa_1!\ldots\kappa_n!d \quad and \quad b^k = b_0^{\kappa_0} b_1^{\kappa_1} \ldots b_n^{\kappa_n} \tag{3.4}$$

Equation 3.3 can be re-written as

$$p(b_0, b_1, \ldots, b_n) = \sum_{|\kappa|=d} \frac{d!}{\kappa!} b^\kappa = \sum_{|\kappa|=d} B_k^d(b) = 1 \tag{3.5}$$

where $B_k^d(b)$ is the local stable basis function. The number of permutations for the multi-index $\kappa_0, \kappa_1, \ldots, \kappa_n$ is given by: $\binom{n+d}{d} = \frac{(n+d)!}{n!d!}$.

3.2.1 B form Notation

From the De Boor's theoram any polynomial $p(x)$ with degree d can be expressed in the B form , a vectorial representation can be written as:

$$p(x) = B^d(b_{t_j}(x))c^{t_j} \qquad \forall x \in t_j \tag{3.6}$$

$$B^d(b_{t_j}(x)) = [B^d_{d,0,0}(b_{t_j}(x))\ B^d_{d-1,0,0}(b_{t_j}(x))\ \ldots\ B^d_{0,1,d-1}(b_{t_j}(x))\ B^d_{0,0,d}(b_{t_j}(x))] \tag{3.7}$$

$$c^{t_j} = [c^{t_j}_{d,0,0}\ c^{t_j}_{d-1,1,0}\ \cdots\ c^{t_j}_{0,1,d-1} c^{t_j}_{0,0,d}\ ]^T \tag{3.8}$$

The B coefficients c^{t_j} are what give shape to the simplex polynomial. **As the degree increases, the number of permutations $\binom{n+d}{d}$ of multi-index also increase, subsequently increasing the number of B-coefficients**.

3.3 Positioning of Vertices

The B-Coefficients determine the shape of the polynomial and each coefficient has a spatial location as given in the figure.

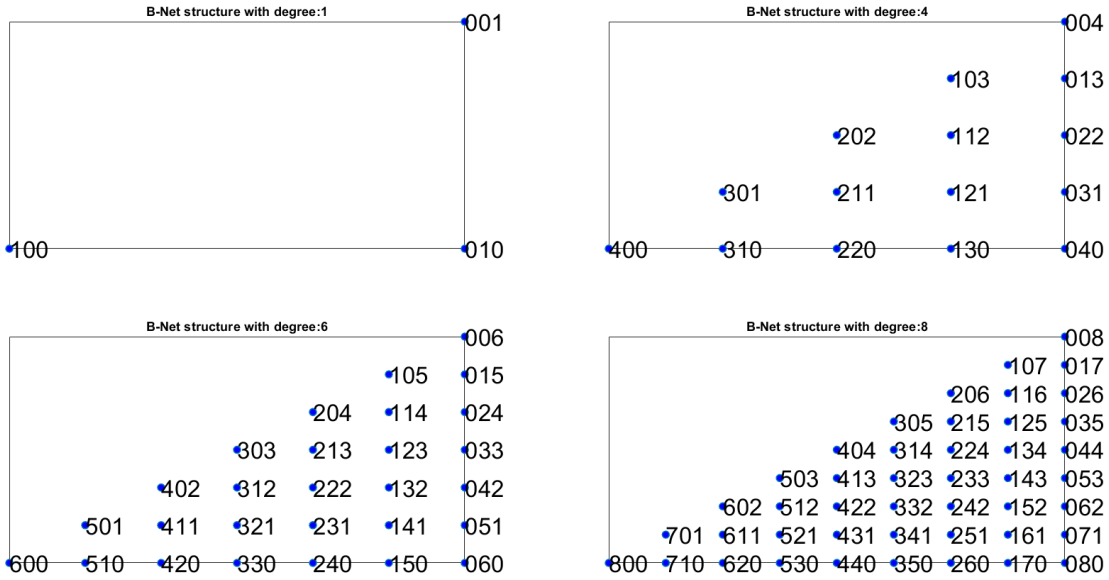


Figure 3.1: B-net structure with increasing degrees

The positioning of the vertices must be done in order to encompass the entire data. However, the data is D shaped rather than an exact triangle. The vertices are chosen that exceed the boundaries of the data by some units in order to encompass the entire area as shown in the figure below. The vertices are decided by using the

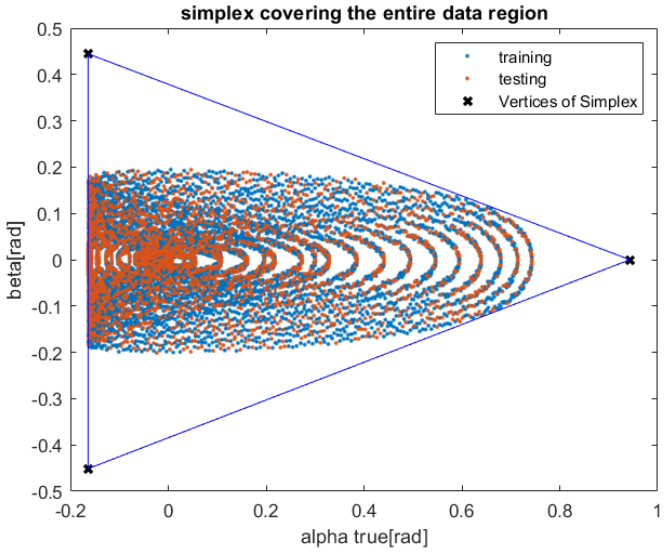


Figure 3.2: Simplex encompassing the data

following formula:

$$\begin{bmatrix} min(\alpha_{true}) & min(\beta) + V1_{offset} \\ min(\alpha_{true}) & max(\beta) + V2_{offset} \\ max(\alpha_{true}) + V3_{offset} & 0 \end{bmatrix} \tag{3.9}$$

where $V1_{offset} = -0.25, V2_{offset} = +0.25$ and $V3_{offset} = +0.2$ are chosen in a manner to cover the entire data.

3.4 Estimating the B coefficients

The optimal set of B coefficients that can describe the simplex polynomial fitting the data can be determined by the same formula used in Least squares:

$$\hat{c} = (B^T B)^{-1} B^T Y \tag{3.10}$$

where Y is the vector containing the samples of C_m: $Y = \begin{bmatrix} C_m(1) & C_m(2) & \dots & C_m(N_{tr}) \end{bmatrix}^T$ and B is the B-form regression matrix given by:

$$B = \begin{bmatrix} B^d_{d,0,0}(b(1)) & B^d_{d-1,0,0}(b(1)) & \dots & B^d_{0,1,d-1}(b(1)) & B^d_{0,0,d}(b(1)) \\ B^d_{d,0,0}(b(2)) & B^d_{d-1,0,0}(b(2)) & \dots & B^d_{0,1,d-1}(b(2)) & B^d_{0,0,d}(b(2)) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B^d_{d,0,0}(b(N_{tr})) & B^d_{d-1,0,0}(b(N_{tr})) & \dots & B^d_{0,1,d-1}(b(N_{tr})) & B^d_{0,0,d}(b(N_{tr})) \end{bmatrix} \tag{3.11}$$

where $b(1), b(2), \dots, b(N_{tr})$ contains the barycentric transformation of $(\alpha(1), \beta(1)), (\alpha(2), \beta(2)), \dots, (\alpha(N_{tr}), \beta(N_{tr}))$ respectively.

Single simplex B-form polynomial are made with degrees ranging from 1 to 12. The rank of the matrix $B^T B$ loses full rank from degree 12 onwards, so the inverse won't exist. The plots of degrees 1 to 6 are shown in figure 3.3.

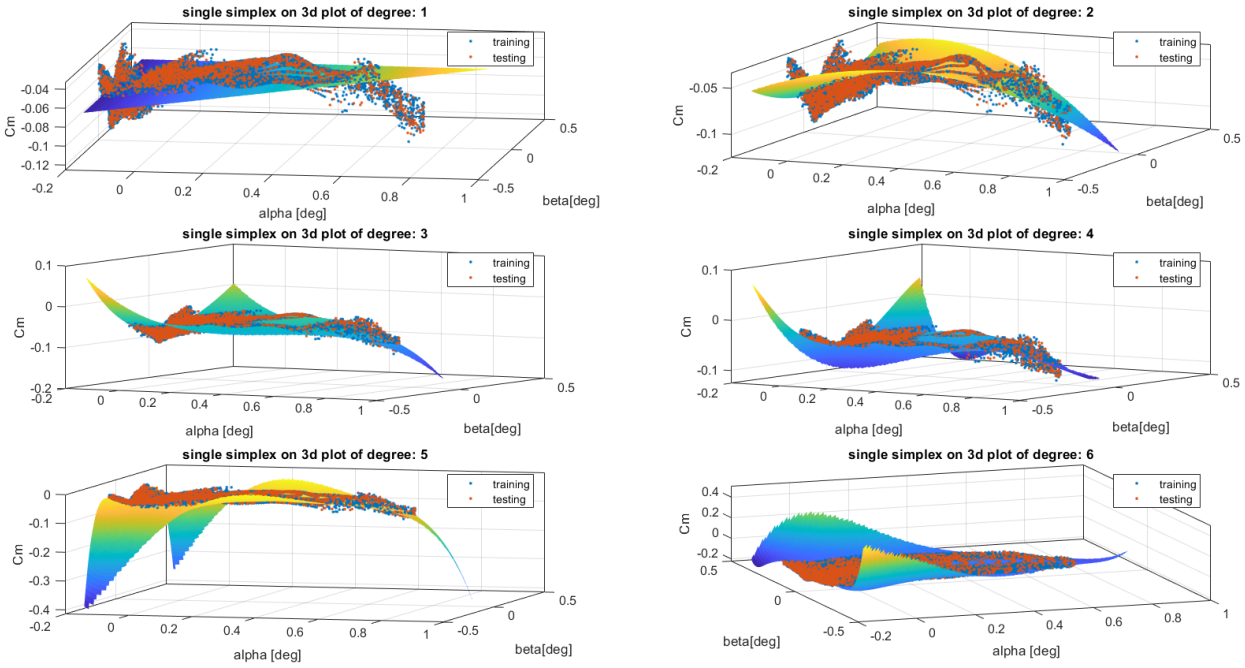


Figure 3.3: 3d Plots of estimated single simplex with different degrees

3.5 Model Validation using Residual analysis

In the Chapter 2 , the formula used for Residual analysis is repeated and used in this section. A simplex with dimension equal to 2 and very small degree still has a better relative RMS residual error as compared to the Ordinary least square estimator.

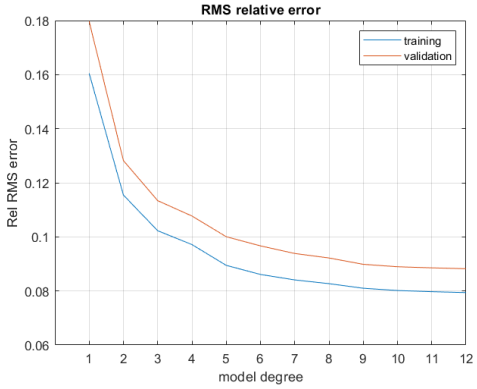


Figure 3.4: RMS relative error with single simplex

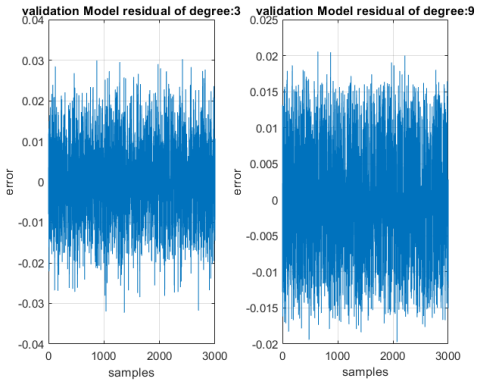


Figure 3.5: comparing residuals with different degrees

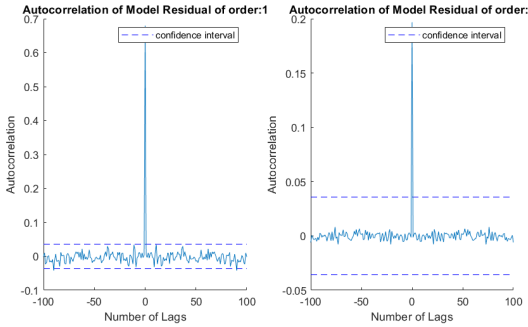


Figure 3.6: Residual Autocorrelation: Single Simplex

As you can see in Figure 3.5 the magnitude or residuals decreases as the degree of simplex polynomial increases, this is due to the increase in the density of B-coefficients, thereby being more finer in estimating the polynomial.

Figure 3.6 shows the autocorrelation of the model residuals for degree 1 and degree 6, increasing the order reduces the autocorrelation away from the bounds of 96% confidence interval as similar observed in Ordinary Least squares.

3.6 Model Validation using statistical analysis

Applying the same formula used for statistical analysis in chapter 2, the trend of the variance of the estimated parameters is very similar to the Ordinary least squares as model degree increases.

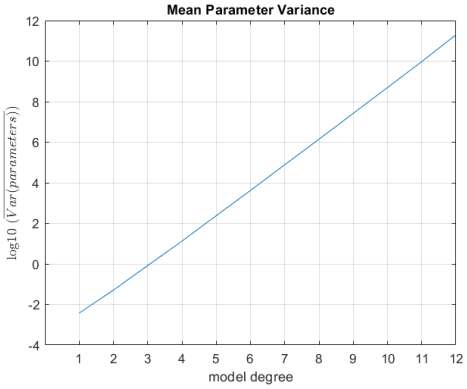


Figure 3.7: Mean Parameter Variance

Chapter 4

System Identification with Simplex splines

4.1 Complete system identification algorithm

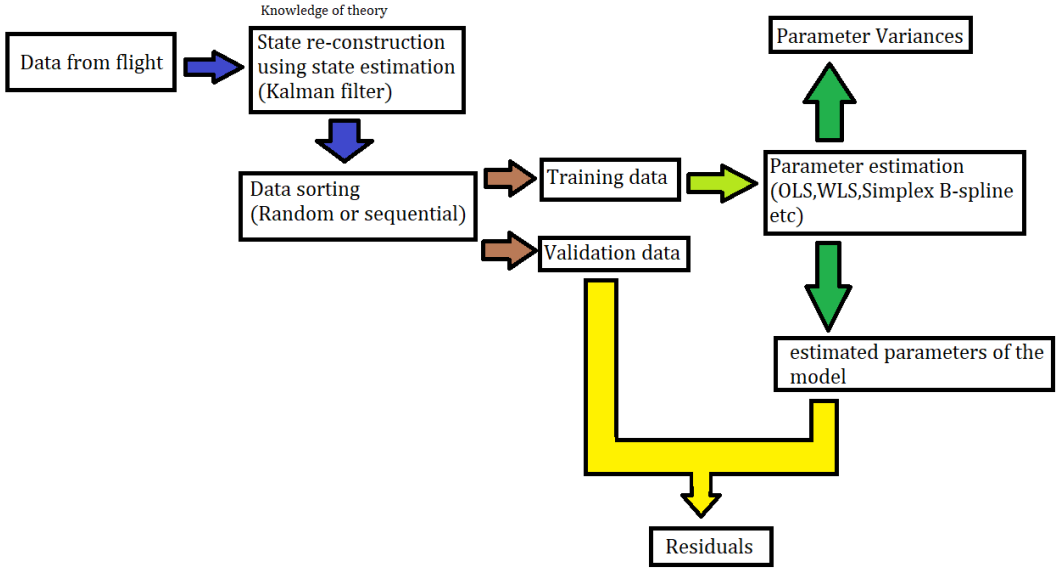


Figure 4.1: Complete system Identification algorithm

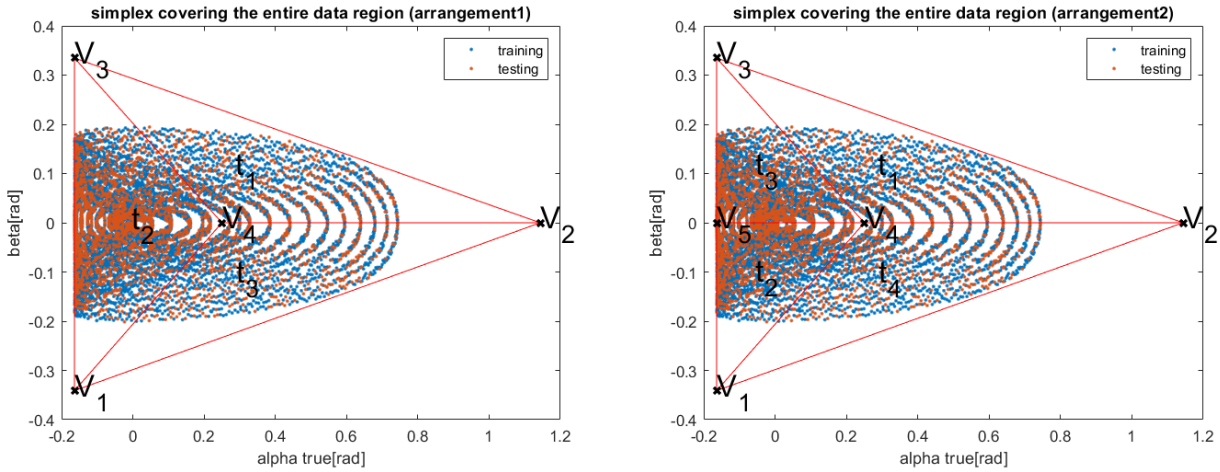
Figure 4.1 shows that first the data from flight experiments are required. Then based on theoretical knowledge of the system, a state estimator can be made in order to compensate any measurement errors due to sensors. The filtered data can be sorted into two datasets: training and validation. Based on the spatial orientation of the data, a simplex surface is chosen in such a way that can encapsulate most or the entire data. The training dataset can be used to create the parameter estimator which can be a simplex B spline whose coefficients can be determined using Least squares.

4.2 Treating the data

The α obtained from Kalman filtering is used in this section as well. The Vertices of the simplices are placed in such a manner that encapsulates the entire region. As seen from the figure, the density of data points is much more in the lower α region as compared to the higher α region.

Figure 4.2 shows the two kinds of arrangements:

- Arrangement 1: 3 simplices, with 1 simplex in the lower α region
- Arrangement 2: 4 simplices, with 2 simplices in the lower α region



(a) Arrangement 1

(b) Arrangement 2

Figure 4.2: Simplices arrangement

4.3 Formulating the Smoothness matrix

In Addition to the equations used in the previous section for a single simplex, it is essential to have a set of conditions that can impose continuity when multiple simplices are being used. These set of equations are given by de Boor's Continuity Equations:

$$c^{t_2}_{k0,m,k1} = \sum_{|\gamma|=m} c^{t1}_{(k0,0,k1)+\gamma} * B^m_\gamma, 0 \leq m \leq r \tag{4.1}$$

where the position of zero and m are the location of single non-zero in the multi-index of out-of-edge vertices. Sending the coefficients to the right hand side, The coefficients can be grouped into a vector in order to generate the smoothness matrix.

An example of the expansion of equations is given for arrangement 1 with order zero and one. The continuity equations are written for simplices t2 and t3 for degree 4.

$$\begin{aligned}
c^{t2}_{0,0,4} &= c^{t3}_{0,0,4} \\
c^{t2}_{1,0,3} &= c^{t3}_{1,0,3} \\
c^{t2}_{2,0,2} &= c^{t3}_{2,0,2} \\
c^{t2}_{3,0,1} &= c^{t3}_{3,0,1} \\
c^{t2}_{4,0,0} &= c^{t3}_{4,0,0} \\
c^{t2}_{3,1,0} &= c^{t3}_{4,0,0}B^1_{1,0,0}(V_3) + c^{t3}_{3,1,0}B^1_{0,1,0}(V_3) + c^{t3}_{3,0,1}B^1_{1,0,0}(V_3) \\
c^{t2}_{2,1,1} &= c^{t3}_{3,0,1}B^1_{1,0,0}(V_3) + c^{t3}_{2,1,1}B^1_{0,1,0}(V_3) + c^{t3}_{2,0,2}B^1_{1,0,0}(V_3) \\
c^{t2}_{1,1,2} &= c^{t3}_{2,0,2}B^1_{1,0,0}(V_3) + c^{t3}_{1,1,2}B^1_{0,1,0}(V_3) + c^{t3}_{1,0,3}B^1_{1,0,0}(V_3) \\
c^{t2}_{0,1,3} &= c^{t3}_{1,0,3}B^1_{1,0,0}(V_3) + c^{t3}_{0,1,3}B^1_{0,1,0}(V_3) + c^{t3}_{0,0,4}B^1_{1,0,0}(V_3)
\end{aligned} \tag{4.2}$$

The H matrix is constructed using the first 5 equations for order 0 and all 9 equations for order 1. The coefficients are estimated using:

$$\begin{bmatrix} \hat{c} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} B^T_{tr}B_{tr} & H^T \\ H & 0 \end{bmatrix}^+ \cdot \begin{bmatrix} B^T_{tr} \\ Y \end{bmatrix} \tag{4.3}$$

which comes from the constrained OLS optimization problem: $\hat{c}$ =arg min $[\frac{1}{2}(Y - B_{tr}.c)^T(Y - B_{tr}.c)]$ subject to $H.c = 0$. In order to check the continuity order of zero it is necessary to check whether the equations for the smoothness matrix have been satisfied. This can be done by inspecting the coefficients after coefficients are estimated:

$$\begin{aligned}
c^{t2}_{0,0,4} &= c^{t3}_{0,0,4} = -0.0334 \\
c^{t2}_{1,0,3} &= c^{t3}_{1,0,3} = -0.0720 \\
c^{t2}_{2,0,2} &= c^{t3}_{2,0,2} = 0.0421 \\
c^{t2}_{3,0,1} &= c^{t3}_{3,0,1} = -0.2532 \\
c^{t2}_{4,0,0} &= c^{t3}_{4,0,0} = 0.0234
\end{aligned} \tag{4.4}$$

The 3 Dimensional plots of degree 4 and varying continuity order are given in figure 4.3. By visual inspection, the smoothness of the curves increases at the simplex edges as the order increases. As the continuity order

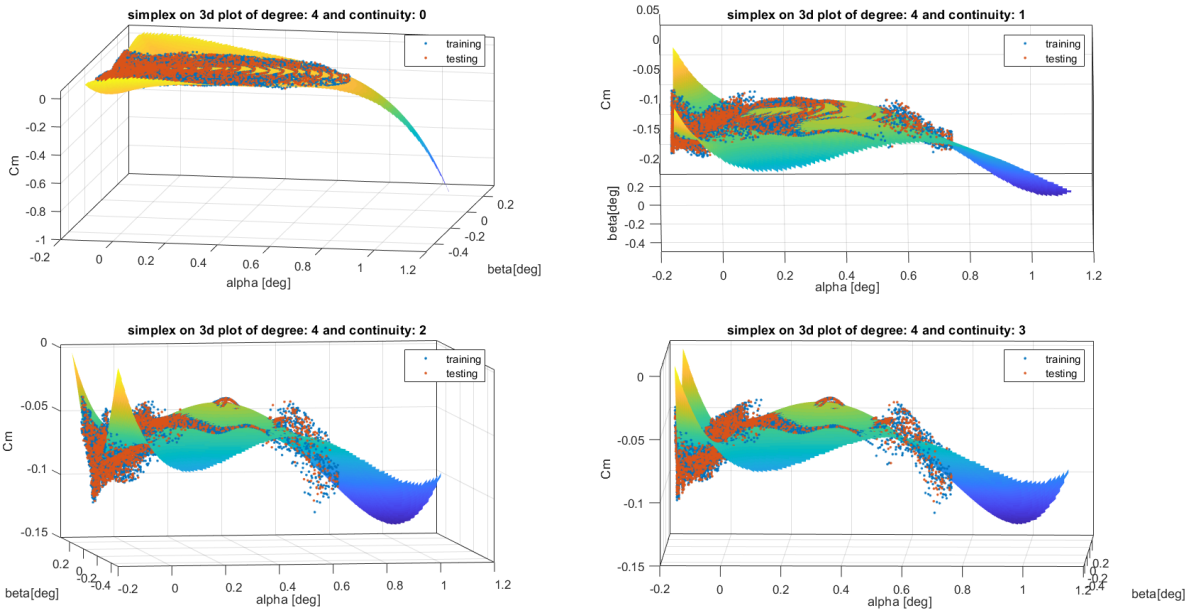


Figure 4.3: 3D plots of varying continuity

increases, although the simplices becomes smoother, but they become less accurate as evident from the 3D plots.

4.4 Model residual analysis

The Relative RMS residual error is compared between the two arrangement of multi-Simplex B splines. It is seen that by increasing the simplices, the relative RMS residual error drops faster as shown in figure 4.4. The relative RMS residual error approaches 0.09 faster in arrangement 2 than in arrangement 1.

The simplices residuals with fixed continuity order and 2 different degrees are compared. Increasing the degree reduces the magnitudes of residuals with arrangement1 (having lesser simplices). Whereas the reduction in magnitude is not so significant when degree increases in case of arrangement2 (having more simplices) as shown in figure 4.5

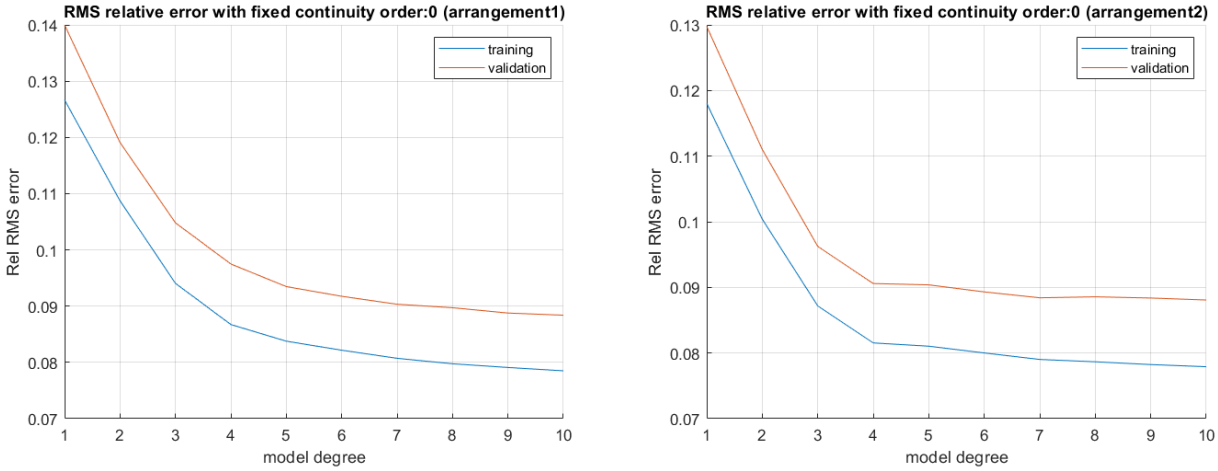
An opposite trend is observed as the degree is fixed and continuity increases. The gradient of increase in relative RMS of residual is lower when there are lower simplices (arrangement 1) as compared to more simplices (arrangement 2) as shown in figure 4.6.

4.5 Statistical model quality analysis

Figure 4.7 shows the mean variance of B-coefficients. It can be seen again that the increasing degree causes the increase in the B-coeffcient variance but the arrangement 2 has a higher variance as the number of B-coefficents have increased. However, The parameter variance using the multi-simplex B-splines is lower than that of ordinary least squares or single simplex.

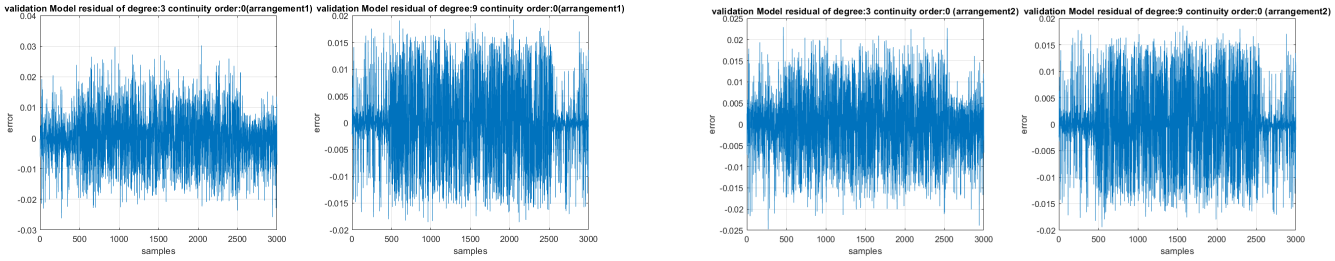
4.6 Conclusion

In this section, the model residuals from selected models, obtained from above parameter estimation techniques are summarised. Table 4.1 shows that the lowest model residual is for the Multi-Simplex having more number of simplices making it the most accurate. Even increasing the continuity doesn't deteriorate the accuracy of arrangement 2 by a large amount. The Least Squares approach has the lowest accuracy of them all. While the simplices do have very similar model residual , overall it can be further reduced at the expense of increasing the number of simplices.



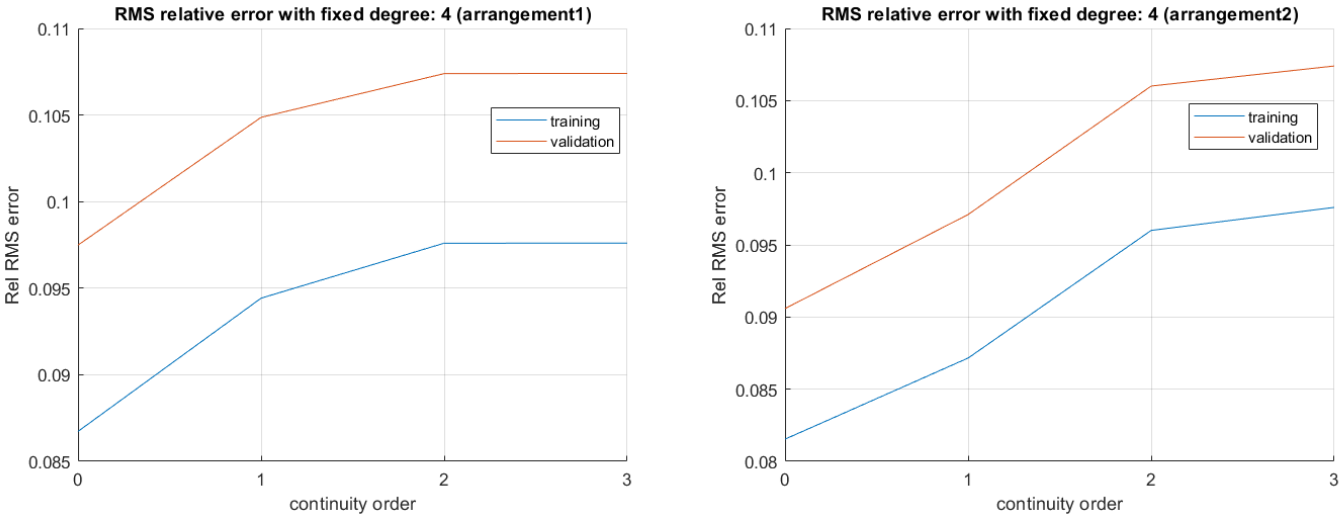
(a) Arrangement 1 (b) Arrangement 2

Figure 4.4: Relative RMS Residual as degree increases



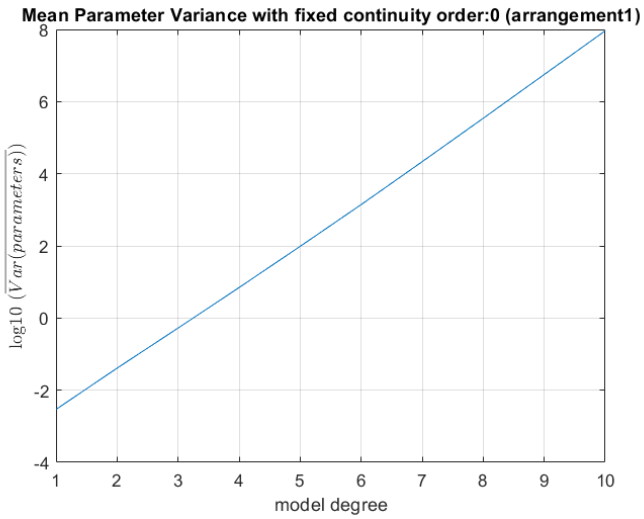
(a) Arrangement 1 (b) Arrangement 2

Figure 4.5: Residual compare

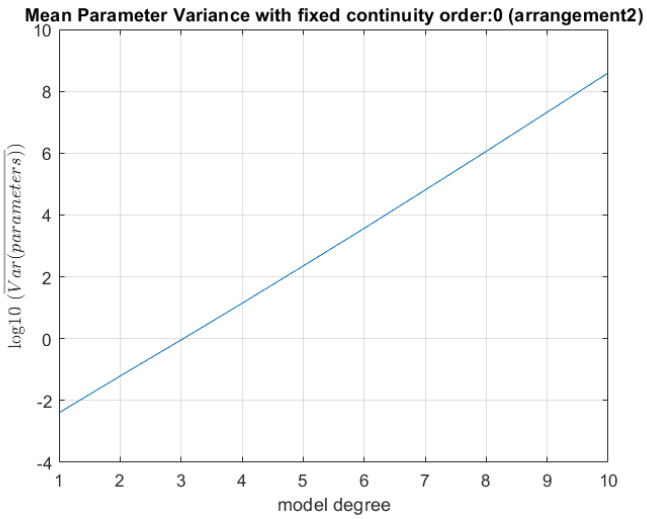


(a) Arrangement 1 (b) Arrangement 2

Figure 4.6: Relative RMS Residual as continuity increases



(a) Arrangement 1



(b) Arrangement 2

Figure 4.7: Mean Parameter variance

Model	Validation Model Residual (Relative RMS)
Polynomial (Least Squares)	0.10772
Single Simplex	0.0971
Multi Simplex (r=0, simplices=3)	0.0974
Multi Simplex (r=1, simplices=3)	0.1049
Multi Simplex (r=0, simplices=4)	0.0906
Multi Simplex (r=1, simplices=4)	0.0971

Table 4.1: Model residuals for different models and degree 4

Bibliography

[1] Jack Kevin Ly. Angle of attack determination using inertial navigation system data from flight tests.