Budapest University of Technology and Economics
Department of Control Engineering and Information Technology

École Nationale Supérieure d'Ingénieurs de Poitiers
Laboratoire d'Automatique et d'Informatique Industrielle

Nonlinear dynamic model identification for aircraft applying time and frequency domain identification techniques

Student Science Conference 2011

Written by:
Dániel Vízer

Consultants:
Assoc. prof. Dr. Guillaume Mercère, ENSIP LAII
Prof. emer. Dr. Béla Lantos, BUTE CEIT

Abstract

Aircraft control is one of the most progressive fields of the modern control techniques. Control methods implemented on embedded flight computers enhance not only robot pilot functions but also support the pilot. These are known as stability and control augmentation systems. There are also several civil applications of these methods; for instance, in case of vehicles of commercial and personal air transport. However, the military is the most motivated to improve these applications and methods. It is indispensable to have an accurate nonlinear dynamic model of the aircraft to achieve an adequate control that satisfies all of the requirements. A mathematical model of the aircraft can be derived using kinematic, dynamic and navigational equations of the rigid body. Contrarily, it is a more difficult process to determine its parameters.

The determination of the nonlinear model and its parameters are parts of the field of system identification. In this present work I reviewed several identification methods such as the Maximum Likelihood (ML) method both in the time domain and in the frequency domain. Particular focus is placed on the output-error method. As a principal part of this work I demonstrate the methods mentioned above applying the SIDPAC Toolbox developed by NASA, and I compare the given results with those obtained in [4].

Kivonat

A repülőgépek irányításának problémája egy kiterjedt és rohamosan fejlődő ága a szabályozástechnikának. A repülőgépek fedélzeti számítógépeire implementált szabályozások nemcsak a robotpilóta funkciót valósítanak meg, hanem komoly szerepet játszanak a pilóta munkájának támogatásában is. Ezeket az eljárásokat ún. UAV (Unmanned Aerial Vehicle) pilótanélküli kisrepülőgépeken fejlesztik és tesztelik. A precíz szabályozások tervezéséhez elengedhetetlenül fontos, hogy ismerjük a repülőgép pontos nemlineáris dinamikus modelljét. A modell matematikai alakja egzakt módon előáll, azonban a modell paramétereinek meghatározása komoly méréstechnikai és jelfeldolgozási feladat.

A nemlineáris modell paramétereinek kiszámítása a rendszerek identifikációjának részét képező tudományághoz tartozik. Jelen munkában feldolgozásra és bemutatásra kerülnek a frekvencia és időtartománybeli Maximum Likelihood (ML) módszerek, különös tekintettel a kimeneti hiba (OE) módszerekre. A módszerek demonstrálása során felhasználásra kerülnek a SIDPAC identifikációs Toolbox [1] függvényei.

Table of contents

1 Introduction

Aircraft control is one of the most progressive fields of the modern control techniques. Embedded on board control systems enhance not only robot pilot functions but also support the pilot. In the past five decades the huge growth in aircraft development has focused on the improvement of modern control methods. This began in the late 60s when digital computers become available. Nowadays, most modern warplanes, also known as fourth generation jet fighters, require not only accurate dynamical models to obtain very precise control but also high speed computers on board. Furthermore, there are not any direct connections between control surface actuators and the pilot's instruments of control (pilot stick, pedals…). These aircraft are also known as *"fly-by wire"* controlled aircraft. These airplanes cannot be controlled without an on board computer because they are not stable aerodynamically. There are several control surfaces such as canards, flaperons…etc that cannot be controlled by the pilot. These additional control surfaces are controlled by embedded control systems in order to obtain the required stability. Thus, control and stability augmentation systems are indispensable. Consequently, having an accurate nonlinear dynamic model of the aircraft is necessary in order to develop an adequate control that satisfies all of the requirements. For many applications, an aircraft can be assumed as a rigid body, whose motion is governed by Newtonian physics. The navigational equations play also an important role in building the mathematical model of the aircraft. These equations enable transformations between the different coordinate-systems. Measured data can be gathered from ground measurements (wind-tunnel tests) or directly from the aircraft instrumentation. These measurements include input and output data sequences.

This current work is a continuation of [4]. Hence, detailed descriptions of the applied aircraft models can be found in [1],[4] and [5]. In this part, I reviewed several identification methods such as the time domain Maximum Likelihood (ML) and frequency domain methods based on [1]. Particular focus is placed on the output-error method based on [2] and [1]. The methods mentioned above are used in case of the same data sequences as in [4] in Matlab environment using the SIDPAC Toolbox. Eventually, the given results are also compared with those that obtained in [4].

2 Review of time-domain and frequency domain Maximum Likelihood identification methods

In this chapter the time domain Maximum Likelihood (ML) and the frequency domain ML identification methods are introduced. Also the fundamentals of influential directions are summarized. Particular focus is placed on the output-error (OE) method. The theory written in this chapter is based on the descriptions and definitions can be found in [1] and [2].

2.1 Why the Maximum Likelihood methods

During aircraft system identification, the applied measured time series are generated by nonlinear dynamic systems and corrupted by noises. Because of this, a very simple idea is used: assuming a model that depends on the unknown parameters, then minimizing a cost function in order to find the values of the parameters that make the outputs of the identified model most likely to correlate to the measured data.

2.2 Outline of estimation theory

The parameter estimation consists in finding adequate values of unknown model parameters θ in an assumed model, based on noisy measurements z. Since the estimator computes approximate values of the parameter vector $\hat{\theta}$ based on noisy measured data z, hence $\hat{\theta}$ is a random variable. In order to develop an estimator, four specifications are required:

(i) A model structure with unknown parameters;

(ii) Measurements z;

(iii) A mathematical model of the process;

(iv) Assumptions about the uncertainties in the measurement noise v, and in the unknown model parameters.

Observations can be linear or nonlinear functions of the parameters. A model is called linear in the parameters if its output y is given by

$$y = H\theta, \tag{2.1}$$

where the matrix H is assumed to be known. The measurement equation then can be expressed as

$$z = H\theta + v\,. \tag{2.2}$$

A model is nonlinear in the parameters, if its measurement equation is given by

$$z = h(\theta) + v\,, \tag{2.3}$$

where $h(\theta)$ is assumed to be a known function. Generally, there are n_0 measured outputs, a measured data vector consisting of N data points. Three models of uncertainties in the parameters and the observations are considered. They are known as the Bayesian model, the Fisher model and the least-squares model. The parameters and measurements are formed as follows:

Bayesian model

θ is a vector of random variables with probability density $p(\theta)$;

v is a random noise sequence with a probability density $p(v)$.

Fisher model

θ is a vector of unknown constant parameters;

v is a random noise sequence with probability density $p(v)$.

Least-squares model

θ is a vector of unknown constant parameters;

v is a random vector of measurement noise.

The time domain ML and frequency domain ML estimator were developed for the Fisher model. θ is the vector of unknown constant parameters and v has an expected value or mean that equal to 0, and a variance R. This can be written in the form $\mathbb{N}(0,R)$. The Fisher model is based on the Fisher estimation theory. Detailed description can be found in [1]. The concept of the likelihood function is used, where $L(z;\theta) = p(z|\theta)$. The parameter vector θ is assumed to be a vector of unknown constants. Hence, the most common estimator of the Fisher model is to maximize the likelihood function, which is equal to the value of the parameter vector that maximizes $L(z;\theta)$ for given z. If $p(z)$ is Gaussian, the likelihood function has the form

$$L(z;\theta) = \left[(2\pi)^N |R|\right]^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(z-H\theta)^T R^{-1}(z-H\theta)\right]. \tag{2.4}$$

where $|\bullet|$ denotes the determinant. Then the estimate

$$\hat{\theta} = \max_{\theta} L(z;\theta)\,, \tag{2.5}$$

minimizes

$$J(\theta)=\frac{1}{2}(z-H\theta)^{T}R^{-1}(z-H\theta). \tag{2.6}$$

Or in case of nonlinear measurement equation

$$J(\theta)=\frac{1}{2}(z-h(\theta))^{T}R^{-1}(z-h(\theta)). \tag{2.7}$$

The Fisher information matrix M is defined as

$$M(\theta)=E\left[\left(\frac{\partial \ln L}{\partial \theta}\right)\left(\frac{\partial \ln L}{\partial \theta}\right)^{T}\right]=-E\left(\frac{\partial^{2}\ln L}{\partial \theta \partial \theta^{T}}\right), \tag{2.8}$$

This equality is discussed in detail in [1].

2.3 Time-domain ML methods

It can be seen below that in case of a linear measurement equation $z=H\theta+v$ the ML estimator becomes a simple least-square estimator with weighting equal to the inverse of the noise covariance matrix. Furthermore, in case of a stochastic dynamic system with process noise, the measurements are nonlinear functions of the parameters. Thus, the general form of the measurement equation is $z=h(\theta)+v$.

The solution combines a state estimator and a nonlinear parameter estimator. The models to be estimated are in state-space form. The application of a state estimator that is a Kalman filter is necessary because of the presence of the process noise in the dynamic equations. This method is also known as the filter-error method. The nonlinear parameter estimator is required because of the nonlinear measurement equation, which means that there is nonlinear connection between model parameters and measured outputs. However, the combined estimation of parameters and states induce a difficult nonlinear estimation problem. For further practical implementation, three simplifications are made:

(i) The Gauss-Newton method, also known as the modified Newton-Raphson method is used for minimizing the cost function instead of the Newton-Raphson method;

(ii) In case of linear dynamic models, the general Kalman filter is replaced by its steady-states form;

(iii) The unknowns are divided into three subsets: states, model parameters and the elements of the noise covariance matrix.

In the following section, the filter-error method is introduced briefly. It is followed by the properties of ML methods. Properties of ML estimators are also introduced. For further practical applications, the assumption of no process noise is made. In this case, the states can be derived by direct numerical integration. The resulting ML estimator is known as the output-error method. This method is a generalization of the nonlinear least squares estimator presented in [1]. Flight testing is done preferably on a day with calm air in order to eliminate the process noise in the model. Finally, the equation-error method is presented. If the state variables are measured without error, the equation-error version of the ML estimator can be used that is identical to linear regression.

2.3.1 Filter-error method

In most cases of practical system identification problems, such as the aircraft dynamic modeling problem, discrete-time measurements made on a continuous-time dynamic system are given. A stochastic forcing term is added to the system to represent inadequacies of the linear model structure. The model equations then can be written as:

$$\begin{aligned} \dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t) + B_w(\theta)w(t), \\ y(t) &= C(\theta)x(t) + D(\theta)u(t), \\ z(i) &= y(i) + v(i), \\ i &= 1, 2, ..., N, \end{aligned} \tag{2.9}$$

$$\begin{aligned} &E\left[x(0)\right] = \bar{x}_0(\theta), \\ &E\left\{\left[x(0) - \bar{x}_0(\theta)\right]\left[x(0) - \bar{x}_0(\theta)\right]^T\right\} = P_0(\theta), \end{aligned} \tag{2.10}$$

where $w(t)$ is the process noise and $v(i)$ is the measurement noise, and they are white noise sequences.

$$\begin{aligned} &E\left[w(t)\right] = 0, \\ &E\left[v(i)\right] = 0, \\ &E\left[w(t_i)w^T(t_j)\right] = Q(t_i, \theta)\delta(t_i - t_j), \\ &E\left[v(i)v^T(j)\right] = R(i, \theta)\delta_{ij}, \end{aligned} \tag{2.11}$$

where θ is the unknown parameter vector.

The random vectors $w(t)$ and $v(i)$ are assumed to be white and uncorrelated. The Q and R matrices are usually diagonal. The likelihood function for a sequence of measurements will be denoted $L(Z_N;\theta)$ that means $Z_N = \left[z(1), z(2), ..., z(N)\right]^T$. The expression of the likelihood function can be derived by successive applications of the Bayes's rule.

$$\begin{aligned} L(Z_N;\theta) &= \left[z(1), z(2), \dots z(N), \theta\right] = L\left[z(N)\middle|Z_{N-1};\theta\right] L\left[Z_{N-1};\theta\right] \\ &= L\left[z(N)\middle|Z_{N-1};\theta\right] L\left[z(N-1)\middle|Z_{N-2};\theta\right] L\left[Z_{N-2};\theta\right] = \dots \\ &= \prod_{i=1}^{N} L\left[z(i)\middle|Z_{i-1};\theta\right]. \end{aligned} \tag{2.12}$$

As was mentioned below, in order to find an optimal parameter vector, the likelihood function has to be maximized. However, for computational purposes, it is advantageous to minimize the negative logarithm of the likelihood function instead. Thus, the ML estimator can be written in the form of

$$\hat{\theta} = \max_{\theta} L\left[Z_N;\theta\right] = \max_{\theta} \prod_{i=1}^{N} L\left[z(i)\middle|Z_{i-1};\theta\right] = \min_{\theta} \sum_{i=1}^{N} -ln\left\{L\left[z(i)\middle|Z_{i-1};\theta\right]\right\}. \tag{2.13}$$

If $w(t)$ and $v(i)$ are normally distributed and independent, then $z(i)$ will also have the same properties.

The following notation will be used: $\hat{x}(i|i-1)$ is the best estimate of the state $\hat{x}(i)$ using measurements $\left[z(1), z(2), \dots, z(i-1)\right]$. Then $L\left[z(i)\middle|Z_{i-1};\theta\right] = L\left[z(i);\theta\right]$ will be determined by the mean and its covariance matrix. These quantities can be expressed as

$$\begin{aligned} E\left[z(i);\theta\right] &= \hat{y}(i|i-1) \\ Cov\left[z(i);\theta\right] &= E\left\{\left[z(i) - \hat{y}(i|i-1)\right]\left[z(i) - \hat{y}(i|i-1)\right]^T\right\} \\ &= \left[\mathbf{v}(i)\mathbf{v}^T(i)\right] = \mathcal{B}(i), \end{aligned} \tag{2.14}$$

where

$$\mathbf{v}(i) = z(i) - \hat{y}(i|i-1) \ , \tag{2.15}$$

is the vector of innovations and $\mathcal{B}(i)$ is the innovation covariance matrix. Hence, the probability density of the innovation approaches a Gaussian distribution, as the sampling rate increases. In case of a sufficiently high sample rate, the likelihood function has the form of

$$L\left[z(i);\theta\right] = (2\pi)^N \left|\mathcal{B}(i)\right|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{v}(i)^T \mathcal{B}(i)^{-1}\mathbf{v}(i)\right]. \tag{2.16}$$

Then, the negative log-likelihood function can be expressed as

$$-ln\left[L\left(Z_N;\theta\right)\right] = \frac{1}{2}\sum_{i=1}^{N}\left[\mathbf{v}(i)^T \mathcal{B}(i)^{-1}\mathbf{v}(i) + ln\left|\mathcal{B}(i)\right|\right] + \frac{Nn_0}{2} ln(2\pi), \tag{2.17}$$

where n_0 is the number of output variables. It can be seen that the last term on the right side is a constant, and it has no effect on the optimization. Thus, it can be neglected, which results in

$$-ln\left[L\left(Z_N;\theta\right)\right]=\frac{1}{2}\sum_{i=1}^{N}\left[\mathbf{v}(i)^T\mathcal{B}(i)^{-1}\mathbf{v}(i)+ln\left|\mathcal{B}(i)\right|\right]. \tag{2.18}$$

In order to determine a negative log-likelihood function the mean value and the covariance matrix of the innovation have to be calculated. These quantities can be derived using a Kalman filter. The filtering has two steps, consisting of prediction and measurement update. The following equations are derived for linear time-invariant system.

Initial condition:

$$\begin{aligned} x(0)&=E\left[x(0)\right]=\overline{x}_0(\theta),\\ P(0)&=E\left\{\left[x(0)-\overline{x}_0\right]\left[x(0)-\overline{x}_0\right]^T\right\}=P_0(\theta), \end{aligned} \tag{2.19}$$

Prediction:

$$\begin{aligned} &\frac{d}{dt}\left[\hat{x}(t\left|i-1\right.)\right]=A(\theta)\hat{x}(t\left|i-1\right.)+B(\theta)u(t),\\ &\frac{d}{dt}\left[P(t\left|i-1\right.)\right]=A(\theta)P(t\left|i-1\right.)+P(t\left|i-1\right.)A^T(\theta)+B_w(\theta)QB_w^T(\theta),\\ &\quad(i-1)\Delta t\le t\le i\Delta t, \end{aligned} \tag{2.20}$$

Measurement Update:

$$\begin{aligned} \hat{x}(i\left|i\right.)&=\hat{x}(i\left|i-1\right.)+K(i)\mathbf{v}(i),\\ K(i)&=P(i\left|i-1\right.)C(\theta)^T\mathcal{B}^{-1}(i\left|i-1\right.),\\ P(i\left|i\right.)&=\left[I-K(i)C(\theta)\right]P(i\left|i-1\right.), \end{aligned} \tag{2.21}$$

where

$$\begin{aligned} \mathcal{B}(i\left|i-1\right.)&=C(\theta)P(i\left|i-1\right.)C^T(\theta)+R(\theta),\\ \mathbf{v}(i)&=z(i)-C(\theta)\hat{x}(i\left|i-1\right.)-D(\theta)(i), \end{aligned} \tag{2.22}$$

Optimization algorithm

The ML parameter estimates are obtained by minimizing the negative log-likelihood function

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{N}\mathbf{v}^T(i)\mathcal{B}^{-1}(i)\mathbf{v}(i)+\frac{1}{2}\sum_{i=1}^{N}\left|\mathcal{B}(i)\right|. \tag{2.23}$$

In order to solve this optimization problem, generally, the Newton-Raphson method is used. To determine a solution to the optimization-problem given in (2.23), the cost function θ is expanded in a Taylor-series around a nominal parameter estimation θ_0. The choice of this nominal parameter estimation plays an important role in both the convergence and computational time of the optimization. The Taylor-series expansion is given by

$$J(\theta) = J(\theta_0) + \Delta\theta^T J'(\theta_0) + \frac{1}{2}\Delta\theta^T J''(\theta_0)\Delta\theta + \ldots, \tag{2.24}$$

We approximate $J(\theta)$ with the second-order expansion of its Taylor-series,

$$J(\theta) \approx J(\theta_0) + \Delta\theta^T J'(\theta_0) + \frac{1}{2}\Delta\theta^T J''(\theta_0)\Delta\theta, \tag{2.25}$$

where $J'(\theta)$ is the Jacobian, $J''(\theta)$ is the Hessian of the cost function at the nominal parameter estimation and $\Delta\theta$ is the vector of changes in θ.

$$J'(\theta) = \frac{\partial J(\theta)}{\partial\theta} = \begin{bmatrix} \dfrac{\partial J(\theta)}{\partial\theta(1)} \\ \dfrac{\partial J(\theta)}{\partial\theta(2)} \\ \vdots \\ \dfrac{\partial J(\theta)}{\partial\theta(n_p)} \end{bmatrix}, \tag{2.25}$$

$$J''(\theta) = \frac{\partial^2 J(\theta)}{\partial\theta\partial\theta^T} = \begin{bmatrix} \dfrac{\partial^2 J(\theta)}{\partial\theta(1)\partial\theta(1)} & \dfrac{\partial^2 J(\theta)}{\partial\theta(1)\partial\theta(2)} & \cdots & \dfrac{\partial^2 J(\theta)}{\partial\theta(1)\partial\theta(n_p)} \\ \dfrac{\partial^2 J(\theta)}{\partial\theta(2)\partial\theta(1)} & \dfrac{\partial^2 J(\theta)}{\partial\theta(2)\partial\theta(2)} & \cdots & \dfrac{\partial^2 J(\theta)}{\partial\theta(2)\partial\theta(n_p)} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 J(\theta)}{\partial\theta(n_p)\partial\theta(1)} & \dfrac{\partial^2 J(\theta)}{\partial\theta(n_p)\partial\theta(2)} & \cdots & \dfrac{\partial^2 J(\theta)}{\partial\theta(n_p)\partial\theta(n_p)} \end{bmatrix}. \tag{2.26}$$

The necessary condition to minimize $J(\theta)$ becomes

$$J'(\theta_0) + J''(\theta_0)\Delta\theta = 0. \tag{2.27}$$

The solution of this equation gives an estimate for the vector of parameter changes,

$$\Delta\hat{\theta} = -\left[J''(\theta)\right]^{-1} J'(\theta). \tag{2.28}$$

We assumed that the Hessian is nonsingular. Since the first and second order derivative of the cost function are computed around a nominal parameter estimation θ_0, the updated parameter vector estimation $\hat{\theta}$ is computed by,

$$\hat{\theta} = \theta_0 + \Delta\theta. \tag{2.29}$$

Because of the approximation in the Taylor series, it is necessary to repeat the estimation using the estimated vector $\hat{\theta}$ as the new nominal value θ_0. That is important because the outputs are nonlinear functions of the parameters, thus the cost function is more complicated than quadratic. In case of linear least-squares problems, which are discussed in detail in [1], the output depends linearly on the parameters. Thus, the cost function is a quadratic function of the parameters and the term given in (2.25) is an exact equality. Accordingly, the parameter estimates could be derived in one iteration. In this case, the repeated applications of the quadratic cost to the nonlinear function on the parameters are used to iteratively arrive to the solution. This iterative process will be arrived at when the convergence criteria are satisfied. Convergence criteria that have been used in practice are introduced later in this section.

Minimization of the negative-log likelihood function for the filter-error problem is a very difficult optimization problem. Computing the second order derivatives in the Hessian matrix is an extremely time demanding task and is not robust enough. Hence, for practical applications the modified Newton-Raphson method is used. Details of this approach are introduced later. A block diagram of the filter-error method is shown below in Figure.2.1.

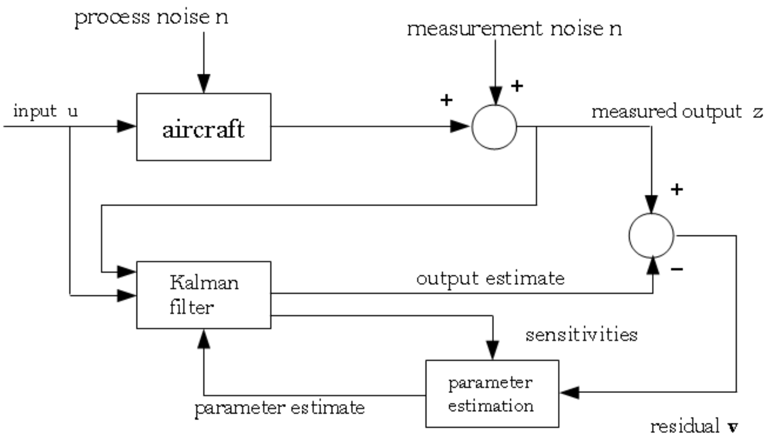

Figure 2.1 Block diagram of filter-error method

Simplification possibilities

In order to obtain simplifications of this optimization problem, several solutions are introduced in [1] and [2]. Comparisons and summaries of these simplifications are also outlined in [1]. The ML estimation algorithm for a time-invariant stochastic linear dynamic system can be simplified through replacing the Kalman filter by its steady-states form.

Initial condition:

$$\begin{aligned} x(0) &= E\left[x(0)\right] = \bar{x}_0(\theta), \\ P(0) &= E\left\{\left[x(0)-\bar{x}_0\right]\left[x(0)-\bar{x}_0\right]^T\right\} = P_0(\theta), \end{aligned} \tag{2.30}$$

Prediction:

$$\begin{gathered} \frac{d}{dt}\left[\hat{x}(t|i-1)\right] = A(\theta)\hat{x}(t|i-1) + B(\theta)u(t), \\ (i-1)\Delta t \le t \le i\Delta t, \end{gathered} \tag{2.31}$$

Measurement update:

$$\begin{gathered} \hat{x}(i|i) = \hat{x}(i|i-1) + K\boldsymbol{v}(i), \\ K = PC^T(\theta)\mathcal{B}^{-1}, \end{gathered} \tag{2.32}$$

where

$$\begin{gathered} \mathcal{B} = C^T(\theta)PC(\theta) + R(\theta), \\ \boldsymbol{v}(i) = z(i) - C(\theta)\hat{x}(i|i-1) - D(\theta)u(i). \end{gathered} \tag{2.33}$$

In this case, the steady-state form of the Riccati equation can be solved for P.

$$P = \Phi\left[P - PC^T(CPC^T + R)^{-1}CP\right]\Phi^T + \Gamma_w Q\Gamma_w^T, \tag{2.34}$$

Φ and Γ_w can be written in the form:

$$\begin{aligned} \Phi &= e^{A\Delta t}, \\ \Gamma_w &= \left[A^{-1}(e^{A\Delta t} - I)\right]B_w. \end{aligned} \tag{2.35}$$

This means that using a steady-state filter brings simplification to the equation for the first order derivatives of the cost function, because the partial derivatives of the cost function with respect the unknown parameter vector are calculated from a linear matrix equation. Henceforward, the estimation algorithm has some difficulties. Further simplification to algorithm can be obtained by recasting the vector of unknown parameters. The elements of K and $\mathcal{B}$ are included into the parameter vector, instead of the elements of Q and R. This leads to a new cost function in which an approximation of the covariance matrix is used according to (2.36) that is derived in [1].

$$\hat{\mathcal{B}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{v}(i)\mathbf{v}^{T}(i). \tag{2.36}$$

The resulting cost can be minimized by using a nonlinear optimization technique, such as the Newton-Raphson method as it was mentioned earlier. This is also an iterative optimization procedure that continues until selected convergence criteria are satisfied. In this case, the unknown parameters are estimated in subsets. First, the parameters contained in A,B,C,D and K held constant while the estimation of the parameters in $\mathcal{B}$ is in progress, and then the parameters contained in $\mathcal{B}$ are held constant while the parameters in A,B,C,D and K are estimated. The estimate of the measurement covariance matrix can be derived from (2.32-33):

$$\hat{R} = (I - \hat{C}\hat{K})\hat{\mathcal{B}}. \tag{2.37}$$

This method carries also many problems in the computation of B_w and Q in (2.32-33). Because of these difficulties in the simplified algorithms a third formulation was proposed. It is advantageous to recast the unknown parameters as the elements of A, B, B_w, C, D, Q and $\mathcal{B}$.

The resulting optimization algorithm has six steps:

(i) Assembling the system and covariance matrices A, B, B_w, C, D, Q and R using the nominal parameter values;

(ii) Computing the covariance matrix of the innovations (2.36);

(iii) Estimating of the unknown parameters in the state and covariance matrices, using a nonlinear optimization technique;

(iv) For $\mathcal{B} = \hat{\mathcal{B}}$, compute $\hat{K}$ and $\hat{R}$ from (2.32) and (2.37);

(v) Update the nominal values of the unknown parameters;

(iv) Repeat steps 1-5 until convergence criteria are satisfied.

The filter-error method is the most general estimation method for practical aircraft system identification problems because it allows the presence of both the measurement and process noise in the model. However, the main practical problem with this method is that there are a lot of parameters to be estimated which often leads to difficulties.

2.3.2 Properties of ML methods

The accuracy of the estimation concerns itself with the properties of the estimates. Using the definitions introduced in [1].

(i) ML estimates of dynamic system parameters are asymptotically unbiased;

$$E(\hat{\theta}) \to \theta, \\ N \to \infty.$$

(ii) ML estimates are consistent;

$$\hat{\theta} \to \theta, \\ N \to \infty.$$

(iii) ML estimates are asymptotically efficient;

$$Cov(\hat{\theta}) \to M^{-1}, \\ N \to \infty,$$

where M the Fisher information matrix defined earlier by (2.8).

This means that the main diagonal elements of the inverse information matrix provide the lower bounds on the parameter variances, named the Cramer-Rao bounds. These diagonal elements represent the achievable accuracy for the parameter estimates.

(iv) ML estimates are asymptotically normal. The distributions of the estimates are asymptotical approaches a normal distribution with mean θ and variance M^{-1}, namely $\hat{\theta}$ is $\mathbb{N}(\theta, M^{-1})$.

2.3.3 Output-error method

Further simplification can be achieved when ML estimation method is applied to a deterministic linear dynamic system. In this case, there is no process noise, and the system can be written in the form of

$$\begin{aligned} \dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t), \\ y(t) &= C(\theta)x(t) + D(\theta)u(t), \\ z(i) &= y(i) + v(i), \quad i = 1, 2, ..., N, \end{aligned} \tag{2.38}$$

where v is $\mathbb{N}(0, R)$ and $Cov\left[v(i)\right] = E\left[v(i)v^T(i)\right] = R\delta_{ij}$.

Furthermore, because of the lack of process noise, Q is zero and the state equations are deterministic. The Kalman gain K is also zero, so the Kalman filter can be replaced by a simple integration of the state equations. The innovations in the method become:

$$\boldsymbol{v} = z(i) - y(i) = z(i) - C(\theta)x(i) - D(\theta)u(i), \quad i = 1, 2, ..., N. \tag{2.39}$$

Practically, they are equal to output-errors. The negative log-Likelihood function then can be written as:

$$-ln\left[L\left(Z_N;\theta\right)\right] = \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{v}(i)^T R^{-1}\boldsymbol{v}(i) + \frac{N}{2}ln\left|R\right| + \frac{Nn_0}{2}ln(2\pi), \tag{2.40}$$

The two last terms of (2.40) can be dropped because they do not depend on the unknown parameter vector θ. The unknown parameters are elements of state matrices as can be seen above and the initial condition vector. Solving (2.40) with respect to R is done by differentiating with respect to R. The result then is set equal to zero and solved for R. That gives (2.41) which is similar to (2.36).

$$\hat{R} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{v}(i)\boldsymbol{v}^T(i). \tag{2.41}$$

Generally, only the diagonal elements of the R matrix are estimated from (2.40) assuming that the measurement noises are uncorrelated with one another. For a given $\hat{R}$ the cost function can be written as:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{v}(i)R^{-1}\boldsymbol{v}^T(i) = \frac{1}{2}\sum_{i=1}^{N}\left[z(i) - y(i)\right]R^{-1}\left[z(i) - y(i)\right]^T. \tag{2.42}$$

This cost is minimized by using a relaxation technique in which first θ held fixed while computing $\hat{R}$ from (2.41), then minimizing the cost with respect to θ for a given fixed R, where $R = \hat{R}$. These two steps are repeated until convergence criteria are satisfied. Optimization of the cost in (2.42) can be done using the Newton-Raphson method as discussed earlier. The first and second order derivative can be calculated by (2.25-26). The first order derivative has the following form:

$$\frac{\partial J(\theta)}{\partial\theta} = \sum_{i=1}^{N}\frac{\partial\boldsymbol{v}^T(i)}{\partial\theta}\hat{R}^{-1}\boldsymbol{v}(i) = -\sum_{i=1}^{N}\frac{\partial y^T(i)}{\partial\theta}\hat{R}^{-1}\boldsymbol{v}(i), \tag{2.43}$$

where the elements of the $\frac{\partial J(\theta)}{\partial\theta}$ vector can be written as

$$\frac{\partial J(\theta)}{\partial\theta_j} = -\sum_{i=1}^{N}\frac{\partial y^T(i)}{\partial\theta_j}\hat{R}^{-1}\boldsymbol{v}(i), \quad j = 1, 2, ...n_p. \tag{2.44}$$

The elements of the second order derivative then have the form of

$$\frac{\partial^2 J(\theta)}{\partial\theta_j\partial\theta_k}=\sum_{i=1}^{N}\frac{\partial y^T(i)}{\partial\theta_j}\hat{R}^{-1}\frac{\partial y(i)}{\partial\theta_k}-\sum_{i=1}^{N}\frac{\partial y^2(i)}{\partial\theta_j\partial\theta_k}\hat{R}^{-1}\mathbf{v}(i),\quad j,k=1,2,...n_p \tag{2.45}$$

The term on the far right side can be neglected. Thus, the resulting optimization method is called Gauss-Newton method or modified Newton-Raphson method. This simplification is made for practical reasons. Obtaining the second-order derivative is computationally expensive. Furthermore, it is susceptible to numerical errors. The parameter vector change then can be written using the approximate second-order derivative:

$$\Delta\hat{\theta}=\left[\sum_{i=1}^{N}\frac{\partial y^T(i)}{\partial\theta_0}\hat{R}^{-1}\frac{\partial y(i)}{\partial\theta_0}\right]^{-1}\left[\sum_{i=1}^{N}\frac{\partial y(i)}{\partial\theta_0}\hat{R}^{-1}\mathbf{v}(i)\right], \tag{2.46}$$

where $\frac{\partial y}{\partial\theta}$ is an $n_0 \times n_p$ matrix called the output sensitivity matrix and its elements are called the output sensitivities. The elements of this matrix quantify the changes in the outputs due to changes in the parameters. $\hat{R}^{-1}$ is usually diagonal. According to (2.46), it can be seen that output sensitivities must be linearly independent and nonzero for a good matrix inversion. When this condition is held, each model parameter has a unique and significant impact on the model outputs. Henceforward, in case when the approximation for the second-order derivative of the cost function is used, the Fisher information matrix can be written as

$$M(\theta)=E\left(\frac{\partial^2 \ln L(Z_N,\theta)}{\partial\theta\partial\theta^T}\right)=\sum_{i=1}^{N}\frac{\partial y^T(i)}{\partial\theta}\hat{R}^{-1}\frac{\partial y(i)}{\partial\theta}. \tag{2.47}$$

Then the ML estimator has the form of:

$$\hat{\theta}=\theta_0-M^{-1}(\theta_0)\left[\frac{\partial J(\theta_0)}{\partial\theta_0}\right]. \tag{2.48}$$

The parameter covariance matrix satisfies the Cramer-Rao inequality

$$Cov(\hat{\theta})\geq M(\theta)^{-1}, \tag{2.49}$$

which means that the inverse of the Fisher information matrix gives the lower bound for the parameter covariance matrix. A block diagram of the output-error method can be seen in Fig 2.2.

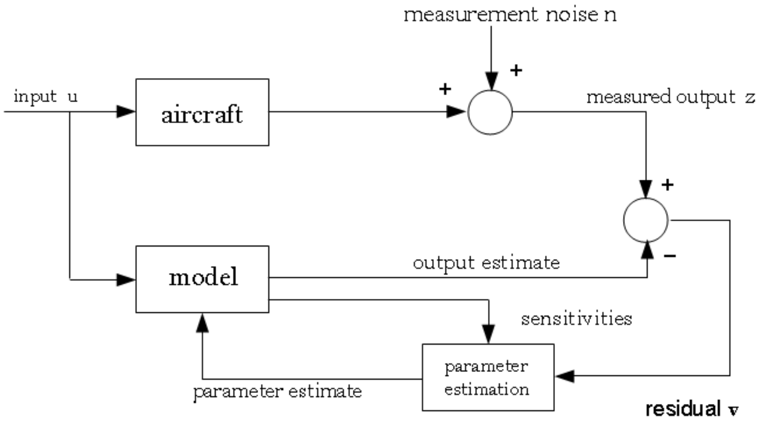

Figure 2.2 Block diagram of output-error method

Convergence criteria, computing sensitivities and nearly singular information matrix

It was shown in [1] that the practical application of ML estimator to flight data sometimes leads difficulties. For instance, the dependence in the likelihood function can be far from quadratic. Furthermore, the information matrix can be singular or nearly singular. The optimization method for the output-error method discussed earlier can be summarized as:

$$\begin{aligned} \hat{\theta} &= \theta_0 + \Delta\theta, \\ \Delta\theta &= -M^{-1}(\theta_0) g(\theta_0), \\ Cov(\hat{\theta}) &\geq M^{-1}(\theta_0), \\ M(\theta_0) &= \sum_{i=1}^{N} \left[S^T(i,\theta_0) \hat{R}^{-1} S(i,\theta_0) \right], \\ g(\theta_0) &= \sum_{i=1}^{N} \left[S^T(i,\theta_0) \hat{R}^{-1} \mathbf{v}(i) \right], \\ S(i) &= \left[\frac{\partial y_j(i,\theta)}{\partial \theta_k} \right], \\ \mathbf{v}(i) &= z(i) - \hat{y}(i,\theta). \end{aligned} \tag{2.50}$$

where the information matrix M has dimensions $n_p \times n_p$ and the sensitivity matrix $S(i)$ has dimensions $n_0 \times n_p$. In order to reach at the global maximum of the likelihood function the modified Newton-Raphson method is used, but the nominal parameter value of θ must be near to the global minimum. The modified Newton-Raphson method is based on this assumption. If the nominal parameter vector is far from the optimal solution, then the result will be either convergence to a local minimum, or divergence. There are several methods to obtain excellent starting values; for instance, regression methods or in particular cases the subspace identification method. The applicability of the subspace identification methods computing a nominal parameter vector is under investigation. Furthermore, there is a method that can be used to eliminate the need of a perfect starting value. This method will be discussed later in this chapter. When there are problems with model structure, model structure determination methods can be used not only to identify an adequate model structure, but also to determine a good starting value. Also, it is necessary to have an initial value for R that can be obtained using θ_0 and (2.41). This optimization is finished when convergence criteria are satisfied. There are criteria for both θ and $\hat{R}$.

The criteria can be:

(i) Absolute value of the elements of $\Delta\theta$ are sufficiently small;

(ii) Changes in $J(\theta)$ are sufficiently small;

(iii) The absolute values of the elements of the cost derivative are close to zero;

(iv) Changes in the elements of $\hat{R}$ are sufficiently small.

Some convergence criteria that worked well for aircraft parameter estimation problems in [1] are introduced here.

$$\left|(\theta_j)_k - (\theta_j)_{k-1}\right| < 1\times10^{-5}, \quad j = 1,2,...n_p, \tag{2.51a}$$

or

$$\frac{\left\|\hat{\theta}_k - \hat{\theta}_{k-1}\right\|}{\left\|\hat{\theta}_{k-1}\right\|} < 0.001, \tag{2.51b}$$

where k denotes the kth iteration and $\| \;\|$ denotes the Euclidien norm. This can also be written:

$$\begin{gathered}\left|\frac{J(\theta_k) - J(\theta_{k-1})}{J(\theta_{k-1})}\right| < 0.001, \\ \left|\frac{(\hat{r}_{jj})_k - (\hat{r}_{jj})_{k-1}}{(\hat{r}_{jj})_{k-1}}\right| < 0.05, \quad j = 1,2,...n_{0,}\end{gathered} \tag{2.52}$$

where $\hat{r}_{jj}$ denotes the estimate of the jth diagonal element of $\hat{R}$. Furthermore, the magnitude of the changes can be varied by directly applying a multiplicative term in $\Delta\theta$:

$$\Delta\hat{\theta} = -cM(\theta_0)g(\theta_0). \tag{2.52}$$

Two methods can be used to compute the elements of the sensitivity matrix. These two methods are the analytical approach and the numerical approach.

The analytical approach uses the output equations of the system with respect to the unknown parameters $\theta_j, j=1,2,...n_p$. In case of the linear system given by (2.38) this can be written as:

$$\frac{\partial y}{\partial \theta_j} = C\frac{\partial x}{\partial \theta_j} + \frac{\partial C}{\partial \theta_j}x + \frac{\partial D}{\partial \theta_j}u, \quad j=1,2,...n_p. \tag{2.53}$$

The states can be derived from the state equations of the dynamic system:

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ x_0 &= x(0). \end{aligned} \tag{2.54}$$

The state sensitivities are computed by solving the state sensitivity equations, which are obtained by differentiating the state equations with respect to the unknown parameters:

$$\frac{d}{dt}\left(\frac{\partial x}{\partial \theta_j}\right) = A\frac{\partial x}{\partial \theta_j} + \frac{\partial A}{\partial \theta_j}x + \frac{\partial B}{\partial \theta_j}u, \quad j=1,2,...n_p. \tag{2.55}$$

In the numerical approach, an approximation is used to compute the output sensitivities:

$$\frac{\partial y}{\partial \theta_j} = \frac{y(\theta_0 + \delta\theta) - y(\theta_0)}{\left|\delta\theta_j\right|}, \quad j=1,2,...n_p. \tag{2.56}$$

or

$$\frac{\partial y}{\partial \theta_j} = \frac{y(\theta_0 + \delta\theta) - y(\theta_0 - \delta\theta)}{2\left|\delta\theta_j\right|}, \quad j=1,2,...n_p. \tag{2.57}$$

where θ_j denotes a perturbation vector for parameter θ_j and $\left|\delta\theta_j\right|$ is scalar magnitude of θ_j. The first equation (2.56) implements forward finite differences and the second (2.57) implements central finite differences. Forward finite differences requires (n_p+1) steps, whereas the central finite differences requires $(2n_p)$ steps. The sensitivities must be calculated for every iteration of the modified Newton-Raphson method.

As was mentioned earlier, the information matrix can be singular. In other words, it can be ill-conditioned. There are three frequent reasons for an ill-conditioned information matrix:

(i) There are too many unknown parameters in the information matrix.

(ii) There is misspecification in the model. Changes in one or more model parameters produce nearly equivalent change in outputs, or one or more model parameter have little or no effect on outputs.

(iii) There is insufficient information in the data.

Hence, there are two main consequences about a nearly singular information matrix.

(iv) The information matrix M can be negative definite that causes a cost increase θ_j for the modified Newton-Raphson step.

(v) The step size $\Delta\theta$ can be large in one or more directions.

The latter problem can be seen from the decomposition of the inverse M matrix as follows:

$$M^{-1} = \sum_{i=1}^{n_p} \frac{1}{\lambda_j} t_j t_j^T, \tag{2.58a}$$

$$\Delta\hat{\theta}_{\lambda_j} = -\frac{1}{\lambda}(t_j^T g)t_j, \tag{2.58b}$$

where λ_j are the eigenvalues of M corresponding to eigenvectors t_j. Then the next step in the modified Newton-Raphson method can be derived by combining (2.58b) with the second equation in (2.50).

Thus, the step will be large for small λ_j in a direction of t_j. The parameter step $\Delta\theta$ is therefore dominated by inappropriate information. There are several techniques that can be used to address this problem. Three of them are introduced here.

Rank deficient method

This method is based on the reduced inverse of M applying the singular value decomposition (SVD).

$$M = UDT^T \tag{2.59}$$

The reduced-order inverse then can be computed as

$$M^{-1} = T\begin{bmatrix} \frac{1}{\mu_1} & & & & & \\ & \frac{1}{\mu_2} & & 0 & & \\ & & \ddots & & & \\ & & & \frac{1}{\mu_{n_p-m}} & & \\ & & & & 0 & \\ & 0 & & & & \ddots \\ & & & & & 0 \end{bmatrix} U^T = \sum_{j=1}^{n_p-m} \frac{1}{\mu_j} t_j u_j^T. \tag{2.60}$$

where t_j and u_j are the jth column of T and U matrices, respectively. The singular values are arranged from the largest to smallest in magnitude.

$$\mu_1 \geq \mu_2 \geq \ldots \geq \mu_{n_p-m} \geq \mu_{n_{\,p-m+1}} \geq \ldots \geq \mu_{n_p} \tag{2.61}$$

The smallest m singular values are then dropped. The selection criteria to drop the singular values is

$$\frac{\mu_j}{\mu_{\max}} < N\varepsilon, \tag{2.62}$$

where N is the number of data points, and ε is the computing machine precision. The rank of the reduced-order inverse of M matrix is $n_p - m$.

Levenberg-Marquardt method

This method augments the information matrix to improve its conditioning and produce a reasonable inverse. The augmentation term can be written as

$$M^{-1} = (M_0 + kA)^{-1}, \tag{2.63}$$

here M_0 is the original information matrix, k is a positive nonzero scalar parameter and A is positive definite matrix. Generally, A is chosen as the identity matrix,

$$M^{-1} = (M_0 + kI)^{-1}. \tag{2.64}$$

The scalar parameter k can be obtained by an iterative procedure. An initial value for k is recommended, as $k = 0.01$.

Bayes-like method

This method improves the conditioning of the information matrix by combining the measured data with prior estimates of some or all of the parameters in the model. The resulting cost function is similar to that for the Bayesian estimator [1].

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{v}^T(i)R^{-1}\boldsymbol{v}(\mathrm{i}) + \frac{1}{2}(\theta - \theta_p)^T \textstyle\sum_p^{-1}(\theta - \theta_p), \tag{2.65}$$

where θ_p is the vector containing the prior parameter values, and $\sum_p$ is a positive semidefinite covariance matrix that represents the confidence in the parameters of θ_p. $\Delta\theta$ then can be computed by using the modified Newton-Raphson method:

$$\Delta\hat{\theta} = -(M(\theta_0) + \textstyle\sum_p^{-1})^{-1}\left[g + \sum_p^{-1}(\theta_0 - \theta_p)\right], \tag{2.66}$$

in cases when the information matrix is nearly singular. In other words, it is singular only at certain points in the progression towards the global optimum. In these cases, the methods discussed above can be used temporarily. The rank deficiency method is the easiest to use in this way. Computing the inverse of the information matrix is always done by calculating the SVD. The singular value ratios are checked against the criteria (2.62). If any singular values are too small, their corresponding terms will be dropped from the inverse according to (2.60-61).

2.4 Frequency domain ML methods

Switching to frequency domain has many advantages. Several methods for data analysis and system modeling can be formulated in frequency domain. Furthermore, frequency domain analysis requires a smaller number of data points and the results can be applied directly to control system design. The basis for these methods is the finite Fourier transform that is the procedure for transforming time-domain data to frequency domain. Frequency domain ML methods are sensitive for transforming errors. Any errors in the transformation affect the accuracy of the frequency domain data. For parametric modeling in the frequency domain, the Fisher model is considered similarly to time-domain analysis. The Fisher model has the following form:

(i) $\tilde{z} = \tilde{h}(\theta) + \tilde{\mathbf{v}}$;

(ii) θ is a vector of unknown constant parameters;

(iii) $\tilde{\mathbf{v}}$ is a random noise sequence with probability density $p(\tilde{\mathbf{v}})$.

where $\tilde{h}$ is a known complex vector function, $\tilde{x}$ and $\tilde{z}$ are functions of the transformed states and controls, respectively. The measurement vector $\tilde{z}$ is the Fourier transform of the time-domain data vector z. In the following section, the transformation from time-domain to frequency domain is presented; then, the ML estimator for linear dynamic model with process noise is introduced followed by the frequency domain output-error method where there is no process noise. Finally, equation-error method in the frequency domain is presented.

2.4.1Transforming between time-domain and frequency domain

The finite Fourier transform was considered as

$$\tilde{x}(f) = \int_0^T x(t) e^{-j2\pi ft} dt, \tag{2.67}$$

and it can be approximated by

$$\tilde{x}(f) \approx \Delta t \sum_{i=0}^{N-1} x(i) e^{-j2\pi fi\Delta t}, \tag{2.68}$$

Another approximation for the finite Fourier transform can be given by applying discrete frequencies:

$$\tilde{x}(f) \approx \Delta t \sum_{i=0}^{N-1} x(i) e^{-j(2\pi k/N)i}, \quad k = 0,1,2,\ldots,N-1 \tag{2.69}$$

The summation in (2.69) is the discrete Fourier transform $X(k)$, also the finite Fourier transform can be approximated by multiplying the discrete Fourier transform by the sampling interval Δt.

$$\tilde{x}(k) = \Delta t X(k), \tag{2.70}$$

where

$$X(k) = \sum_{i=0}^{N-1} x(i) e^{-j(2\pi k/N)i}, \quad k = 0,1,2,\ldots,N-1 \tag{2.71}$$

In case of flight data analysis, there are two important disadvantages to using the approximation mentioned above:

(i) The spacing of the discrete frequencies for the discrete Fourier transform is equal to the reciprocal of the length of data sequence. Hence, the frequencies are fixed at values defined by N and Δt. To achieve an efficient and fast Fourier transformation, this particular selection of frequencies has to be applied. Which means, that the frequencies cannot be chosen arbitrarily.

(ii) Hence, the approximation of the finite Fourier transform is a zeroth-order Euler approximation of the integrand $x(t)e^{-j\omega t} = x(t)\left[\cos(\omega t) - j\sin(\omega t)\right]$. This integrand oscillates as the time variable t changes causing inaccuracies in the approximation. The oscillation increases as the frequency increases. There is the same problem when Δt gets larger.

It can be seen that using the fast Fourier transformation causes inaccuracies as the data sequence decreases, leading to a loss of detail in the frequency domain. Furthermore, modal frequencies for a rigid-body dynamics of a full-scale aircraft are usually below 2Hz. For such a small frequency band, most of the processing in the transformation rests outside the range of interest.

To address these problems in [1], the defining integrand of the Fourier transformation given by (2.67) is approximated by using an approach based on the chirp z-transform, where the frequencies can be chosen arbitrarily regardless of the length of the data sequence. Applying this method for the frequencies of the frequency band of interest, a high accuracy approximation of the Fourier transform can be achieved with excellent computational efficiency.

The discrete Fourier transformation in (2.70-71) can be written as

$$X(k)=\sum_{i=0}^{N-1} x(i)AW^{-ki}, \tag{2.72}$$

where

$$A=1,\;\; W=e^{j\left(\frac{2\pi}{N}\right)}. \tag{2.73}$$

AW^k for $k=0,1,2,...,N-1$ represents equally spaced points on the unit circle in the complex plane. Hence, the AW^k is a complex transform variable z, the selection given in (2.73) is a z-transform of the data in the time-domain $x(i), i=0,1,...,N$. A and W can be changed to achieve different contours in the complex plane. This method is called the chirp z-transform. In fact, angular steps are determined by W, and the value of A can be selected as a starting point in the complex plane. So the chirp z-transform is a discrete Fourier transform with arbitrarily selectable frequency range and resolution.

When the contour in the complex plane is along the unit circle, W and A have the forms of

$$\begin{aligned} A&=e^{j\theta_0}, \\ W&=e^{j\Delta\theta}, \end{aligned} \tag{2.74}$$

where $\omega_0=\theta_0/\Delta t$ is the lower limit of the frequency band in the chirp z-transform in rad/s and $\Delta\omega=\Delta\theta/\Delta t$ is the frequency resolution. Values of θ_0 and $\Delta\theta$ can be chosen arbitrarily. The only limitations for the resolution and the lower limit is both must be in the range $[0,\pi/\Delta t]$, where $\pi/\Delta t$ is the Nyquist angular frequency in rad/s. This chirp z-transform can be used to calculate the finite Fourier transform when the selected frequencies are regularly spaced in the frequency range of interest. In the case of aircraft system identification, evenly spreading the frequencies is the preferred approach to capture all important features in the frequency domain.

The chip z-transform evaluated at M arbitrarily frequencies f_k in Hz, where the selected frequency range is $0\le f_k \le f_N$ yields

$$\begin{gathered} X(k)=\sum_{i=0}^{N-1} x(i)e^{-j2\pi f_k i}, \\ 0\le f_k\le f_N,\;\; k=0,1,2,...,M-1 \end{gathered} \tag{2.75}$$

The number of frequencies can be chosen with the limitation that M cannot be larger then 2N to avoid numerical problems:

$$\theta=\omega_k\Delta t=2\pi f_k\Delta t. \tag{2.76}$$

The expression for the high-accuracy calculation can be written as

$$\begin{aligned}\tilde{x}(\theta) \approx \Delta t \{ & W(\theta) X(k) \\ & + \gamma_0(\theta) x(0) + \gamma_1(\theta) x(1) + \gamma_2(\theta) x(2) + \gamma_3(\theta) x(3) \\ & + e^{j\theta T/\Delta t} \left[\gamma_0^*(\theta)\, x(N-1) + \gamma_1^*(\theta) x(N-2) \right. \\ & \left. + \gamma_2^*(\theta) x(N-3) + \gamma_3^*(\theta)\, x(N-4) \right] \},\end{aligned} \tag{2.77}$$

where $W(\theta), \gamma_0(\theta), \gamma_1(\theta), \gamma_2(\theta)$ and $\gamma_3(\theta)$ are the weights calculated by analytically evaluating the finite Fourier transform integral in (2.77) using cubic Lagrange interpolation applied to the time-domain data. According to [1], the resulting expressions for the weights have the form:

$$\begin{aligned} W(\theta) &\approx 1 - \frac{11}{720}\theta^4 + \frac{23}{15120}\theta^6, \\ \gamma_0(\theta) &\approx -\frac{2}{3} + \frac{1}{45}\theta^2 + \frac{103}{15120}\theta^4 - \frac{169}{226800}\theta^6 \\ &\quad - j\theta\left(\frac{2}{45} + \frac{2}{105}\theta^2 + \frac{8}{2835}\theta^4 + \frac{86}{467775}\theta^6\right), \end{aligned} \tag{2.78a}$$

$$\begin{aligned} \gamma_1(\theta) &\approx \frac{7}{24} - \frac{7}{180}\theta^2 + \frac{5}{3456}\theta^4 - \frac{7}{259200}\theta^6 \\ &\quad - j\theta\left(\frac{7}{72} - \frac{1}{168}\theta^2 + \frac{11}{72576}\theta^4 - \frac{13}{5987520}\theta^6\right), \\ \gamma_2(\theta) &\approx -\frac{1}{6} + \frac{1}{45}\theta^2 - \frac{5}{6048}\theta^4 + \frac{1}{64800}\theta^6 \\ &\quad - j\theta\left(-\frac{7}{90} + \frac{1}{210}\theta^2 - \frac{11}{90720}\theta^4 + \frac{13}{7484400}\theta^6\right), \\ \gamma_3(\theta) &\approx \frac{1}{24} - \frac{1}{180}\theta^2 + \frac{5}{24192}\theta^4 - \frac{1}{259200}\theta^6 \\ &\quad - j\theta\left(\frac{7}{360} - \frac{1}{840}\theta^2 + \frac{11}{362880}\theta^4 - \frac{13}{29937600}\theta^6\right). \end{aligned} \tag{2.78b}$$

Furthermore, it is advantageous to select small distance between frequencies in the frequency range of interest, to ensure that all details of the time-domain data are captured.

The terms to compute the finite Fourier transform applying the chirp z-transform have the form:

$$f_k = f_0 + k\Delta f, \quad k = 0, 1, 2, \ldots, M-1 \tag{2.79}$$

$$\Delta f = \frac{(f_1 - f_0)}{(M-1)}. \tag{2.80}$$

The steps of the procedure now can be written

(i) Choose three of the four values in (2.80) and then compute the remaining value. This determines the frequencies for the analysis.

(ii) Calculate the M values of θ from (2.76).

(iii) Use the chirp z-transform to compute the discrete Fourier transform for the selected frequencies applying (2.72-73).

(iv) Compute the high accuracy finite Fourier transform by using (2.77-78)

2.4.2 Frequency domain ML estimator for linear dynamic model

In order to derive frequency domain ML methods, it is first necessary to transform the linear dynamic model to the frequency domain. Model equations for linear continuous-time dynamic system with process noise and discrete-time measurements have the form:

$$\begin{aligned} \dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t) + B_w(\theta)w(t), \\ z(i) &= C(\theta)x(i) + D(\theta)u(i), + v(i), \\ E\left[x(0)\right] &= 0, \\ E\left[x(0)x^T(0)\right] &= P_0, \quad i = 1, 2, \ldots N. \end{aligned} \tag{2.86}$$

where noises $w(t)$ and $v(i)$ are assumed to be independent and white. They have the following properties:

$$\begin{aligned} E\left[w(t)\right] &= 0, \\ E\left[v(i)\right] &= 0, \\ E\left[w(t)w^T(t)\right] &= Q\delta(t_j - t_i), \\ E\left[v(i)v^T(i)\right] &= R\delta_{ij}. \end{aligned} \tag{2.87}$$

The unknown parameter vector θ occurs in the system and covariance matrices. As was shown in the previous section, estimation of all of the parameters is extremely difficult. This problem can be simplified by replacing the Kalman filter with its steady-state version. Using terms from (2.14) and (2.30-33), the modified filter equations can be written in the form

$$\begin{aligned} \frac{d}{dt}\left[\hat{x}(t|i-1)\right] &= A(\theta)\hat{x}(t|i-1) + B(\theta)u(t), \quad (i-1)\Delta t \le t \le i\Delta t, \\ \hat{x}(i|i) &= \hat{x}(i|i-1) + K(\theta)\boldsymbol{\nu}(i), \end{aligned} \tag{2.88}$$

where the innovations

$$\boldsymbol{\nu}(i) = z(i) - C\hat{x}(i|i-1) - Du(i), \tag{2.89}$$

with

$$\begin{aligned} E\left[\boldsymbol{\nu}(i)\right] &= 0, \\ E\left[\boldsymbol{\nu}(i)\boldsymbol{\nu}^T(i)\right] &= \mathcal{B}\delta_{ij}. \end{aligned} \tag{2.90}$$

The elements of the unknown parameter vector are the elements of A,B,C,D,K and B. Then, the equations are written in the form of their Fourier series expansion. The components of $x(i)$ can be written by (2.78) as

$$\tilde{x}(k) = \Delta t \sum_{i=0}^{N-1} x(i) e^{-j(2\pi k/N)i}, \quad k = 0,1,2,...N-1 \qquad (2.91)$$

As was mentioned earlier, the finite Fourier transform can be approximated by (2.91) and for practical applications it can be replaced by a high-accuracy calculation presented in the next chapter. Dynamic equations now can be written as

$$j\omega_k \tilde{x}(k) = A(\theta)\tilde{x}(k) + B(\theta)\tilde{u}(k) + K\tilde{v}(k), \qquad (2.92)$$

or

$$\begin{aligned} \tilde{x}(k) &= (j\omega_k I - A(\theta))^{-1} B(\theta)\tilde{u}(k) + (j\omega_k I - A(\theta))^{-1} K(\theta)\tilde{\mathbf{v}}(k), \\ \tilde{z}(k) &= C(\theta)\tilde{x}(k) + D(\theta)\tilde{u}(k) + \tilde{\mathbf{v}}(k), \quad k = 0,1,2,...N-1 \end{aligned} \qquad (2.93)$$

where ω_k is the kth frequency. The transformed innovations $\tilde{\mathbf{v}}(k)$ are uncorrelated Gaussian random variables.

$$\begin{aligned} E\left[\tilde{\mathbf{v}}(k)\right] &= 0, \\ E\left[\tilde{\mathbf{v}}(k)\tilde{\mathbf{v}}^T(k)\right] &= \frac{S_{vv}}{N}, \end{aligned} \qquad (2.94)$$

where S_{vv} is a real matrix with diagonal elements equal to the power spectral densities of the elements of $\mathbf{v}(i)$, and $\tilde{\mathbf{v}}^{\dagger}(i)$ is the complex conjugate transpose of $\mathbf{v}(i)$. The left side of (2.93) requires that the trend and bias term be removed from the sequence $x(t)$, since

$$\begin{aligned} \mathcal{F}\left[\dot{x}(t)\right] &= \int_0^T \dot{x}(t) e^{-j\omega t} dt = j\omega \int_0^T \dot{x}(t) e^{-j\omega t} dt + x(T)e^{-j\omega T} - x(0) \\ &= j\omega \tilde{x}(\omega) + x(T)e^{-j\omega T} - x(0), \end{aligned} \qquad (2.95)$$

It can be written from (2.93)

$$\begin{aligned} \tilde{z}(k) &= \left[C(\theta)(j\omega_k I - A(\theta))^{-1} B(\theta) + D(\theta)\right]\tilde{u}(k) \\ &\quad + \left[C(\theta)(j\omega_k I - A(\theta))^{-1} K(\theta) + I\right]\tilde{\mathbf{v}}(k) \\ &= G_1(k,\theta)\tilde{u}(k) + G_2(k,\theta)\tilde{\mathbf{v}}(k), \end{aligned} \qquad (2.96)$$

where G1 and G2 are the system transfer matrices.

$$\begin{aligned} G_1(k,\theta) &= \left[C(\theta)(j\omega_k I - A(\theta))^{-1} B(\theta) + D(\theta)\right], \\ G_2(k,\theta) &= \left[C(\theta)(j\omega_k I - A(\theta))^{-1} K(\theta) + I\right]. \end{aligned} \qquad (2.97)$$

According to the Kalman filter representation in (2.88) is invertible; thus G2 is nonsingular, and the innovations can be written as:

$$\tilde{\mathbf{v}}(k) = G_2(k,\theta)^{-1}\tilde{z}(k) - G_2(k,\theta)^{-1}G_1(k,\theta)\tilde{u}(k), \qquad (2.98)$$

In order to develop the likelihood function, a V_N vector that consists of all innovations up to and including the frequency for which $k = N-1$ is introduced.

$$V_N = \left[\tilde{\mathbf{v}}(0) \quad \tilde{\mathbf{v}}(1) \quad \ldots \quad \tilde{\mathbf{v}}(N-1)\right], \qquad (2.99)$$

$\tilde{\mathbf{v}}(k)$ has a probability density $\tilde{p}(\tilde{\mathbf{v}})$, then it can be written applying the definition of probability densities:

$$p(V_N) = p\left[\tilde{\mathbf{v}}(N-1)\middle|V_{N-1}\right]p(V_{N-1}). \qquad (2.100)$$

Repeated use of (2.100) leads to the expression of the likelihood function:

$$\begin{aligned} L(V_N;\theta) &= p(V_N\,|\theta) \\ &= p\left[\tilde{\mathbf{v}}(N-1)\middle|V_{N-1}\right]p\left[\tilde{\mathbf{v}}(N-2)\middle|V_{N-2}\right]\ldots p\left[\tilde{\mathbf{v}}(1)\middle|V_1\right]p\left[\tilde{\mathbf{v}}(0)\right]. \end{aligned} \qquad (2.101)$$

The distribution of $\tilde{\mathbf{v}}(k)$ is Gaussian, thus the distribution of V_k is also Gaussian. Hence, the probability density for this complex Gaussian vector is

$$p\left[\tilde{\mathbf{v}}(k)\middle|V_k\right] = \frac{N^{n_0}}{\pi^{n_0}\left|S_{vv}\right|}\exp\left[-N\tilde{\mathbf{v}}^{\dagger}(k)S_{vv}^{-1}\tilde{\mathbf{v}}(k)\right]. \qquad (2.102)$$

Combining (2.101) and (2.102), the negative log-likelihood function $L(V_N;\theta)$ is

$$-ln\left[L(V_N;\theta)\right] = J(\theta) = N\sum_{k=0}^{N-1}\tilde{\mathbf{v}}^{\dagger}(k)S_{vv}^{-1}\tilde{\mathbf{v}}(k) + Nln\left|S_{vv}\right| + Nn_0 ln\left(\frac{\pi}{N}\right). \qquad (2.103)$$

The term on the far right side can be neglected, because it is unaffected by θ, the cost function to be minimized then can be written as

$$J(\theta) = N\sum_{k=0}^{N-1}\tilde{\mathbf{v}}^{\dagger}(k)S_{vv}^{-1}\tilde{\mathbf{v}}(k) + Nln\left|S_{vv}\right|. \qquad (2.104)$$

The ML parameter vector estimate is the value of θ that minimizes the (2.104). The estimate of $\hat{S}_{vv}$ has the form of

$$\hat{S}_{vv} = \sum_{k=0}^{N-1}\tilde{\mathbf{v}}(k)\tilde{\mathbf{v}}^{\dagger}(k) = \sum_{k=0}^{N-1}\left[\tilde{z}(k) - \tilde{y}(k,\hat{\theta})\right]\left[\tilde{z}(k) - \tilde{y}(k,\hat{\theta})\right]^{\dagger}, \qquad (2.105)$$

where $\hat{\theta}$ is an estimate of the parameter vector θ. The initial state of $\hat{S}_{vv}$ can be computed by using θ_0 that is the nominal starting value of the unknown parameter vector θ. The estimates of the remaining parts can be done by using the modified Newton-Raphson method. For a given $\hat{S}_{vv}$ the cost function is

$$J(\theta) = \sum_{k=0}^{N-1}\tilde{\mathbf{v}}^{\dagger}(k)S_{vv}^{-1}\tilde{\mathbf{v}}(k), \qquad (2.106)$$

that is minimized by iteratively computing

$$\hat{\theta} = \theta_0 + \Delta\theta, \tag{2.107}$$

where the parameter vector update is

$$\Delta\hat{\theta} = -M(\theta_0)^{-1} J'(\theta), \tag{2.108}$$

and M is the Fisher information matrix:

$$M = -E\left[\frac{\partial^2 lnL(V_N;\theta)}{\partial\theta\partial\theta^T}\right] = E\left[\frac{\partial^2 J(\theta)}{\partial\theta\partial\theta^T}\right]. \tag{2.109}$$

Since, $\Delta\hat{\theta}$ is a vector with only real elements, and the negative log-likelihood function is also real, the first and second-order derivatives of the cost function $J(\theta)$ are also real:

$$\begin{aligned} J(\theta) &= 2N\,\mathrm{Re}\left[\sum_{k=0}^{N-1} \tilde{\mathbf{v}}^\dagger(k)\hat{S}_{vv}^{-1} \frac{\partial\tilde{\mathbf{v}}(k)}{\partial\theta}\right] \\ &= -2N\,\mathrm{Re}\left[\sum_{k=0}^{N-1} \tilde{\mathbf{v}}^\dagger(k)\hat{S}_{vv}^{-1} \frac{\partial\tilde{y}(k)}{\partial\theta}\right], \end{aligned} \tag{2.110}$$

$$\begin{aligned} \frac{\partial^2 J(\theta)}{\partial\theta\partial\theta^T} &= 2N\,\mathrm{Re}\left[\sum_{k=0}^{N-1} \frac{\partial\tilde{\mathbf{v}}^\dagger(k)}{\partial\theta}\hat{S}_{vv}^{-1} \frac{\partial\tilde{\mathbf{v}}(k)}{\partial\theta}\right] \\ &= 2N\,\mathrm{Re}\left[\sum_{k=0}^{N-1} \frac{\partial\tilde{y}^\dagger(k)}{\partial\theta}\hat{S}_{vv}^{-1} \frac{\partial\tilde{y}(k)}{\partial\theta}\right]. \end{aligned} \tag{2.111}$$

The elements of both cost derivatives can be expressed as

$$\frac{\partial J(\theta)}{\partial\theta_j} = -2\,\mathrm{Re}\left\{\sum_{k=0}^{N-1} Tr\left[\frac{\partial G_1^\dagger}{\partial\theta_j}(G_2^\dagger)^{-1}\hat{S}_{vv}^{-1}G_2^{-1}(\hat{S}_{zu} - G_1\hat{S}_{uu})\right] - Tr\left[\frac{\partial G_2^\dagger}{\partial\theta_j}(G_2^\dagger)^{-1}\right]\right\}, \tag{2.112}$$

$$M_{ij} = \frac{\partial^2 J(\theta)}{\partial\theta_i\partial\theta_j} = 2\,\mathrm{Re}\left\{\sum_{k=0}^{N-1} Tr\left[\frac{\partial G_1^\dagger}{\partial\theta_j}(G_2^\dagger)^{-1}\hat{S}_{vv}^{-1}G_2^{-1}\frac{\partial G_1}{\partial\theta_i}\hat{S}_{uu}\right] + Tr\left[\frac{\partial G_2^\dagger}{\partial\theta_j}(G_2^\dagger)^{-1}G_2^{-1}\frac{\partial G_2}{\partial\theta_i}\right]\right\} \tag{2.113}$$

where the estimates of the input spectral density matrix and the cross spectral density matrix, are respectively

$$\begin{aligned} \hat{S}_{uu}(k) &= \tilde{u}(k)\tilde{u}^\dagger(k), \\ \hat{S}_{zu}(k) &= \tilde{z}(k)\tilde{u}^\dagger(k). \end{aligned} \tag{2.114}$$

The final estimates have the same properties given in section 2.3.2.

2.4.3 Output-error method

As was done in the time-domain analysis, the ML in the frequency domain can be simplified when there is no process noise. Hence, the resulting system is deterministic.

$$\begin{aligned} j\omega_k \tilde{x}(k) &= A(\theta)\tilde{x}(k) + B(\theta)\tilde{u}(k), \\ \tilde{y}(k) &= C(\theta)\tilde{x}(k) + D(\theta)\tilde{u}(k), \\ \tilde{z}(k) &= G(k,\theta)\tilde{u}(k) + \tilde{\mathbf{v}}(k), \\ E\left[\tilde{\mathbf{v}}(k)\right] &= 0, \\ E\left[\tilde{\mathbf{v}}(k)\tilde{\mathbf{v}}^{\dagger}(k)\right] &= \frac{S_{vv}}{N}, \end{aligned} \tag{2.115}$$

where

$$\begin{aligned} G(k,\theta) &= G_1(k,\theta) = C(\theta)(j\omega I - A(\theta))^{-1}B(\theta) + D(\theta), \\ \omega_k &= 2\pi k / T, \quad k = 0,1,2,\ldots,N-1 \end{aligned} \tag{2.116}$$

and S_{vv} the spectral density matrix of $\mathbf{v}(i)$. The innovations are reduced to the output-errors

$$\tilde{\mathbf{v}}(k,\theta) = \tilde{z}(k) - \tilde{y}(k,\theta) = \tilde{z}(k) - G(k,\theta)\tilde{u}(k), \tag{2.117}$$

The negative log-likelihood function and $\hat{S}_{vv}$ then can be written

$$J(\theta) = N\sum_{k=0}^{N-1}\tilde{\mathbf{v}}^{\dagger}(k,\theta)S_{vv}^{-1}\tilde{\mathbf{v}}(k,\theta) + Nln\left|S_{vv}\right|. \tag{2.118}$$

$$\hat{S}_{vv} = \sum_{k=0}^{N-1}\tilde{\mathbf{v}}(k,\theta)\tilde{\mathbf{v}}^{\dagger}(k,\theta) = \sum_{k=0}^{N-1}\left[\tilde{z}(k) - \tilde{y}(k,\hat{\theta})\right]\left[\tilde{z}(k) - \tilde{y}(k,\hat{\theta})\right]^{\dagger}. \tag{2.119}$$

In order to estimate the unknown parameters, the modified Newton-Raphson method is used. The elements of the first and second-order derivatives are computed by simplifying (2.112) and (2.113), where G2 = I and G1 = G:

$$\frac{\partial J(\theta)}{\partial \theta_j} = -2\operatorname{Re}\left\{\sum_{k=0}^{N-1} Tr\left[\frac{\partial G^{\dagger}}{\partial \theta_j}\hat{S}_{vv}^{-1}(\hat{S}_{zu} - G\hat{S}_{uu})\right]\right\}, \tag{2.120}$$

$$M_{ij} = \frac{\partial^2 J(\theta)}{\partial \theta_i \partial \theta_j} = 2\operatorname{Re}\left\{\sum_{k=0}^{N-1} Tr\left[\frac{\partial G^{\dagger}}{\partial \theta_j}\hat{S}_{vv}^{-1}\frac{\partial G_1}{\partial \theta_i}\hat{S}_{uu}\right]\right\}, \tag{2.121}$$

Estimates of the input spectral density matrix and the cross spectral density matrix have the same form given in (2.114). The negative log-likelihood function, its derivative and the information matrix can be expressed as

$$J(\theta) = N\sum_{k=0}^{N-1}\tilde{\mathbf{v}}^{\dagger}(k,\theta)\hat{S}_{vv}^{-1}\tilde{\mathbf{v}}(k,\theta), \tag{2.122}$$

$$\frac{\partial J(\theta)}{\partial\theta} = -2N\,\mathrm{Re}\left[\sum_{k=0}^{N-1}S^{\dagger}(k)\hat{S}_{vv}^{-1}\tilde{\mathbf{v}}(k,\theta)\right], \tag{2.123}$$

$$M(\theta) = \frac{\partial^2 J(\theta)}{\partial\theta\partial\theta^T} = 2N\,\mathrm{Re}\left[\sum_{k=0}^{N-1}S^{\dagger}(k)\hat{S}_{vv}^{-1}S(k)\right], \tag{2.124}$$

where $S(k)$ is the output sensitivity matrix with dimensions $n_0 \times n_p$ given by

$$S(k) = \frac{\partial G(k,\theta)\tilde{u}(k)}{\partial\theta}. \tag{2.125}$$

The modified Newton-Raphson step is given by (2.107) and (2.108).

When the measured data is in the form of frequency response functions, the output-error parameter estimation algorithm can be easily modified. In this particular case, the cost function can be written

$$J(\theta) = N\sum_{k=0}^{N-1}\tilde{u}^{\dagger}(k)\left[G_E(k) - G(k,\theta)\right]^{\dagger}S_{vv}^{-1}\left[G_E(k) - G(k,\theta)\right]\tilde{u}(k), \tag{2.126}$$

where $G(k,\theta)$ is matrix with transfer functions as elements, and $G_E(k)$ is a matrix of experimentally determined frequency responses. Cost functions in (2.122) or (2.126) then can be minimized with respect to unknown parameter vector θ. When the dynamic system has only one input, the output-error cost function can be written as

$$J(\theta) = N\sum_{k=0}^{N-1}\tilde{u}^{*}(k)\left[G_E(k) - G(k,\theta)\right]^{\dagger}S_{vv}^{-1}\left[G_E(k) - G(k,\theta)\right]\tilde{u}(k), \tag{2.127}$$

In order to optimize the cost functions, it is not necessary to include all of the frequencies. In the next chapter, a method that allows using arbitrary frequencies is introduced. This method is called the high-accuracy finite Fourier transform. Using arbitrary frequencies, automatically discards some of the power in the time-domain data. Generally, this is equivalent to a zero-lag, low-pass filtering procedure, which improves the convergence and accuracy of the parameter estimates. The weighting matrix in the cost function is the spectral density matrix, which is ideally estimated as a function of frequency index k. Nevertheless, in most practical cases, the frequencies that correspond to the bandwidth of the dynamic system modes are used; which is also the bandwidth where most of the residual power resides in case of colored residuals. Hence, using a constant S_{vv} over frequencies used in the analysis is a good approximation of the spectral density of the residual power. Accordingly, the assumed noise model matches well to reality, and it is not necessary to compute the estimated error bounds for colored residuals when parameter estimates are done in the frequency domain.

3 Identification methods applied to simulated flight data

In this chapter, identification methods presented in the earlier parts are used to estimate non-dimensional aerodynamic parameters of an aircraft. In order to verify our experiments, the methods are used in the case of measured data gathered virtually in MATLAB/Simulink environment.

3.1 Outline of aircraft modeling

In this subsection, necessary information about aircraft modeling is given based mainly on [7]. Similarly to [4], the FDC-1.2 Simulink Toolbox [7] was used for the simulations. In this toolbox, a well parameterized nonlinear dynamic model of the De Havilland DHC-2 Beaver can be found. The aircraft is depicted on Figure 3.1.

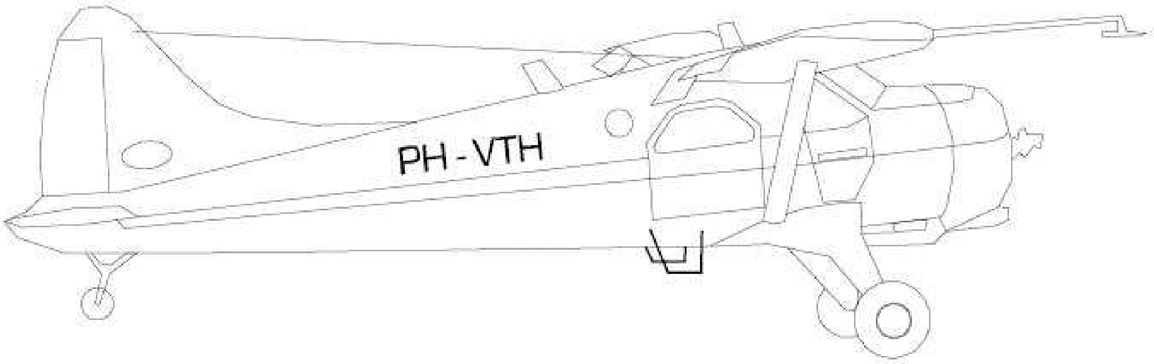

Figure 3.1 De Havilland DHC-2 Beaver

This work, does not deal how to build a nonlinear dynamic model for an aircraft; therefore the results and notations as in [4] and [5] are used to describe aerodynamic parameters, angular rates/velocities/accelerations and the orientation of the aircraft. The state and the input vector of the Beaver are

$$x=\begin{bmatrix} V & \alpha & \beta & p & q & r & \psi & \theta & \varphi & x_e & y_e & H \end{bmatrix}^T \qquad (3.1)$$

$$u = \begin{bmatrix} \delta_e & \delta_a & \delta_r & \delta_f \end{bmatrix}^T \tag{3.2}$$

An explication of the elements of the state and input vector can be found in the Table 3.1.

Table 3.1 Description of the elements of the state and input vector

State	Description	Dimension
V	absolute velocity	m/s
α	angle of attack	rad
β	sideslip angle	rad
p	roll rate	rad/s
q	pitch rate	rad/s
r	yaw rate	rad/s
ψ	Euler heading angle	rad
θ	Euler pitch angle	rad
φ	Euler roll angle	rad
x_e	x coord. in NED frame	m
y_e	y coord. in NED frame	m
H	altitude	m
δ_e	elevator deflection	rad
δ_a	aileron deflection	rad
δ_r	rudder deflection	rad
δ_f	flaperon deflection	rad

In aircraft control and simulation theory, several coordinate frames are used. Generally, measured data are gathered in the Aircraft Body Coordinate system (ABC) or in the Wind-axis coordinate system. In case of both, the origin is the center of gravity of the aircraft. The ABC and the Wind frame can be seen in Figure 3.2 and 3.3, respectively. It can be seen on Figure 3.3 that the angle of attack (α), the sideslip angle (β) and the absolute velocity are considered in the Wind-axis coordinate system. Furthermore, the angular rates (p, q, r), the Euler angles (φ, θ, ψ) are considered in the ABC frame. X,Y,Z are the body-axis components of aerodynamic force acting on the aircraft and L,M,N are the body axis components of aerodynamic moment acting on the aircraft (Figure 3.2). The input vector contains the deflections of the actuators (Figure 3.3). The position of the aircraft is considered in the NED (North-East-Down) coordinate system; in which the origin is on the surface of the Earth, the x axis points to north, the y axis points to east and the z axis points to down. To build the nonlinear dynamic model of an aircraft, kinematic, aerodynamic, moment and navigational equations are used [5]. Furthermore, the given model can be divided into two decoupled subsets [5]. One of them describes the longitudinal while the other one describes the lateral motion. The longitudinal state variables are the angle of attack (α), absolute velocity (V), pitch rate (q) and the Euler pitch angle (θ).The variables describing the lateral motion are the sideslip angle (β), roll rate (p), yaw rate (r), Euler roll angle (φ) and Euler heading angle (ψ).

In the following, non-dimensional aerodynamic parameters will be identified. So a particular focus is placed on the aerodynamic force equation. Aerodynamic forces and moments depend always on the flight condition. In the case of Beaver aircraft, it has the following form

$$F_{aero} = dp_1(x, \dot{x}, u_{aero}) \tag{3.3}$$

where F_{aero} is a vector of aerodynamic forces, p_1 is a polynomial vector-functions containing the non-dimensional aerodynamic coefficients and d converts the non-dimensional coefficients to dimensional forces and moments. Note the polynomial function depends on $\dot{x}$ too. It is only taken into account by $\dot{\beta}$ in the case of aerodynamic side force coefficient (Y_a). p_1 describes the aerodynamic forces and moment coefficients in ABC. It has the following form:

$$\begin{aligned}
C_{X_a} &= C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\alpha^2}}\alpha^2 + C_{X_{\alpha^3}}\alpha^3 + C_{X_q}\frac{q\bar{c}}{V} + C_{X_{\delta r}}\delta_r + C_{X_{\delta_f}}\delta_f + C_{X_{\alpha\delta_f}}\alpha\delta_f \\
C_{Y_a} &= C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{pb}{2V} + C_{Y_r}\frac{rb}{2V} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r + C_{Y_{\delta_r\alpha}}\delta_r\alpha + C_{Y_\beta}\frac{\dot{\beta}b}{2V} \\
C_{Z_a} &= C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_{\alpha^3}}\alpha^3 + C_{Z_q}\frac{q\bar{c}}{V} + C_{Z_{\delta_e}}\delta_e + C_{Z_{\delta_e\beta^2}}\delta_E\beta^2 + C_{Z_{\delta_f}}\delta_f + C_{Z_{\alpha\delta_f}}\alpha\delta_f \\
C_{l_a} &= C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V} + C_{l_r}\frac{rb}{2V} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_a\alpha}}\delta_l\alpha \\
C_{m_a} &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\alpha^2}}\alpha^2 + C_{m_q}\frac{q\bar{c}}{V} + C_{m_{\delta_e}}\delta_e + C_{m_{\beta^2}}\beta^2 + C_{m_r}\frac{rb}{2V} + C_{X_{\delta_f}}\delta_f \\
C_{n_a} &= C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V} + C_{n_r}\frac{rb}{2V} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta r}}\delta_r + C_{n_q}\frac{q\bar{c}}{V} + C_{Y_{\beta^3}}\beta^3
\end{aligned} \tag{3.4}$$

Coefficients in the equations above are constant in the entire modeled region. Real values are given in the next subsection.

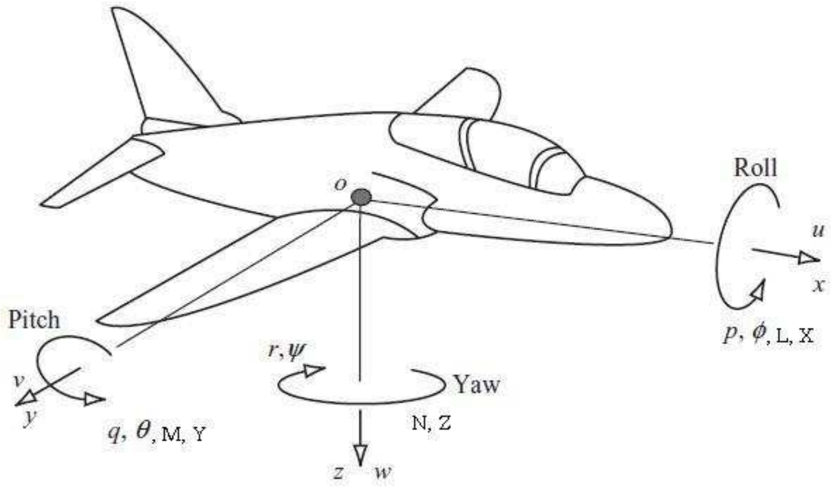

Figure 3.2 The ABC coordinate frame

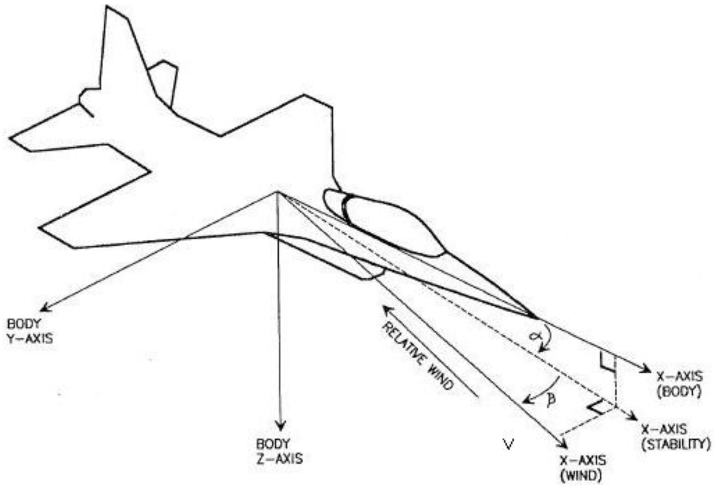

Figure 3.3 The Wind-axis coordinate frame

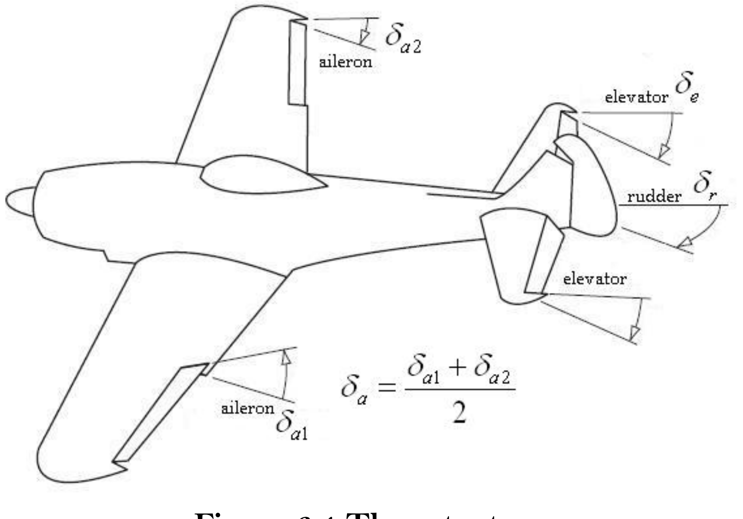

Figure 3.4 The actuators

3.2 Virtual test flight

In order to estimate the values of the coefficients in p_1, necessary data sequences are obtained by virtual test flight using FDC 1.2 Toolbox [7] in MATLAB/Simulink environment. In this work, only the lateral coefficients are estimated. Hence, only the second, fourth and the last row of (3.4) are used. In the previous work [4], data sequences were gathered by simulation. However for this work I executed a new virtual test flight which was a lateral maneuver. For the simulation, the nonlinear dynamic model of the Beaver aircraft was used. Table 3.2 contains the applied trim values. Gathered simulated data is depicted on Figure 3.5.

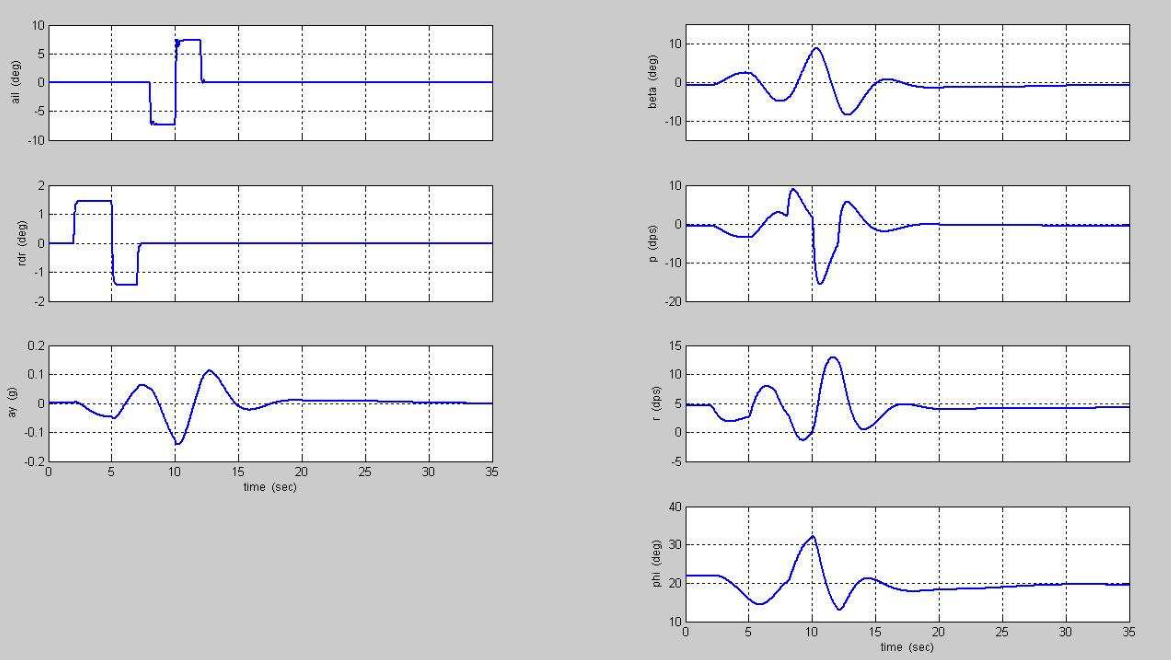

Figure 3.5 Measured input and output variables

Table 3.2 Applied trim values in the virtual test flight

State	Trim value
V	45 m/s
α	0.14724 rad
β	-0.011814 rad
p	-0.009.7782 rad/s
q	0.032429 rad/s
r	0.080425 rad/s
φ	0 rad
θ	0.11229 rad
ψ	0.38328 rad
x_e	0 m
y_e	0 m
H	1000 m

As was mentioned earlier, this work is a continuation of a previous and similar work [4] that was prepared for the Student Science Conference in 2010. In that work, regression techniques and equation-error method were used to estimate the aerodynamic coefficients. So the equations in (3.4) were used directly to perform a one-step least squares estimation. Here, least squares estimates are used to obtain decent initial values for recursive maximum likelihood output-error methods, both in frequency and time domain. Note that in the case of maximum likelihood output-error methods linear dynamic equations (3.5) are used instead of the entire nonlinear model given in (3.4). This reduced model came from [1] and [5].

$$\begin{aligned}
\dot{\beta} &= \frac{\overline{q}_0 S}{m V_0}\left(C_{Y_\beta}\beta + C_{Y_r}\frac{rb}{2V_0} + C_{Y_{\delta_r}}\delta_r \right) \\
&\quad + p\sin\alpha_0 - r\cos\alpha_0 + \frac{g\cos\theta_0}{V_0}\phi + b_{\dot{\beta}}, \\
I_x\dot{p} + I_{xz}\dot{r} &= \overline{q}_0 Sb\left(C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V_0} + C_{l_r}\frac{rb}{2V_0} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \right) + b_{\dot{p}}, \qquad (3.5a) \\
I_x\dot{r} + I_{xz}\dot{p} &= \overline{q}_0 Sb\left(C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V_0} + C_{n_r}\frac{rb}{2V_0} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r \right) + b_{\dot{r}}, \\
\dot{\phi} &= p + \tan\theta_0 r + b_{\phi},
\end{aligned}$$

$$a_y = \frac{\bar{q}_0 S}{mg}\left(C_{Y_\beta}\beta + C_{Y_r}\frac{rb}{2V_0} + C_{Y_{\delta_r}}\delta_r \right) + b_{a_y}, \tag{3.5b}$$

where values with subscript 0 are constants denoting the trim values from Table 3.2. Furthermore, I_{xz}, I_x, I_z are inertia quantities, S, b, m are the wing reference area, the wingspan and the mass of the aircraft, respectively. a_y is the acceleration along the y axis. All of the dynamic equations (3.5) include a bias parameter. These are to estimate the constant terms in (3.4). Note that all of the variables on the left side of the equations are measured quantities. Equations (3.5) and (3.6) can then be written in state space form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -I_{xx} & -I_{xz} & 0 \\ 0 & -I_{xz} & -I_{zz} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} k_1 C_{Y_\beta} & \sin\alpha_0 & (k_1 C_{Y_r} k_2 - \cos\alpha_0) & \dfrac{g\cos\theta_0}{V_0} \\ k_3 C_{l_\beta} & k_3 C_{l_p} k_2 & k_3 C_{l_r} k_2 & 0 \\ k_3 C_{n_\beta} & k_3 C_{n_p} k_2 & k_3 C_{n_r} k_2 & 0 \\ 0 & 1 & \tan\theta_0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & k_1 C_{Y_{\delta_r}} & b_{\dot{\beta}} \\ k_3 C_{l_{\delta_a}} & k_3 C_{l_{\delta_r}} & b_{\dot{p}} \\ k_3 C_{n_{\delta_a}} & k_3 C_{n_{\delta_r}} & b_{\dot{r}} \\ 0 & 0 & b_{\dot{\phi}} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \\ 1 \end{bmatrix}, \tag{3.6a}$$

$$\begin{bmatrix} \beta \\ p \\ r \\ \phi \\ a_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ k_4 C_{Y_\beta} & 0 & k_4 C_{Y_r} k_2 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ q \\ \phi \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & k_4 C_{Y_{\delta_r}} & b_{a_y} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \\ 1 \end{bmatrix}. \tag{3.6b}$$

$$k_1 = \frac{\bar{q}_0 S}{mV_0}, k_2 = \frac{b}{2V_0}, k_3 = \bar{q}_0 Sb, k_4 = \frac{\bar{q}_0 S}{mg}$$

The parameter vector to be estimated has the following form:

$$\begin{aligned} \theta = \Big[& C_{Y_\beta}, C_{Y_r}, C_{Y_{\delta_r}}, b_{\dot{\beta}}, \\ & C_{l_\beta}, C_{l_p}, C_{l_r}, C_{l_{\delta_a}}, C_{l_{\delta_r}}, b_{\dot{p}}, \\ & C_{n_\beta}, C_{n_p}, C_{n_r}, C_{n_{\delta_a}}, C_{n_{\delta_r}}, b_{\dot{r}}, b_{\dot{\phi}}, b_{a_y} \Big]^T . \end{aligned} \tag{3.7}$$

3.3 Simulation results

Firstly, time domain output-error method is used to estimate the lateral aerodynamic coefficients. In the case of output-error method, all of the coefficients are estimated at the same time. Note that these algorithms depend highly on the initial parameter values. Several experiments were made using different starting values; most of them diverged. Hence, results of the equation-error method were used as a starting value. Modified Newton-Raphson method was used to minimize the cost function in (2.42). The estimated parameter covariance matrix was computed from (2.50) as the Cramer-Rao lower bound $Cov(\hat{\theta}) \geq M^{-1}(\theta_0)$. Note that the residuals are assumed to be white. The solution took 53 Newton-Raphson iterations. Parameter estimation results are written in Table 3.3-3.5. Time domain output-error model fit to measured data is shown in Figure 3.6. It can be seen that a quite good prediction is obtained by using the estimated parameters.

Secondly, estimation of the same lateral non-dimensional aerodynamic parameters is done by applying frequency domain output-error method as well. Similarly, results from equation-error linear regression [1] done in frequency domain were used as initial parameters values. All relevant time series were transformed into frequency domain using the high-accuracy finite Fourier transformation described in subsection 2.4.1 with the frequency vector $f = [0.10 \quad 0.12 \quad 0.14 \quad \ldots \quad 1.5]^T \, Hz$. All significant spectral components caught by applying this frequency range [1].

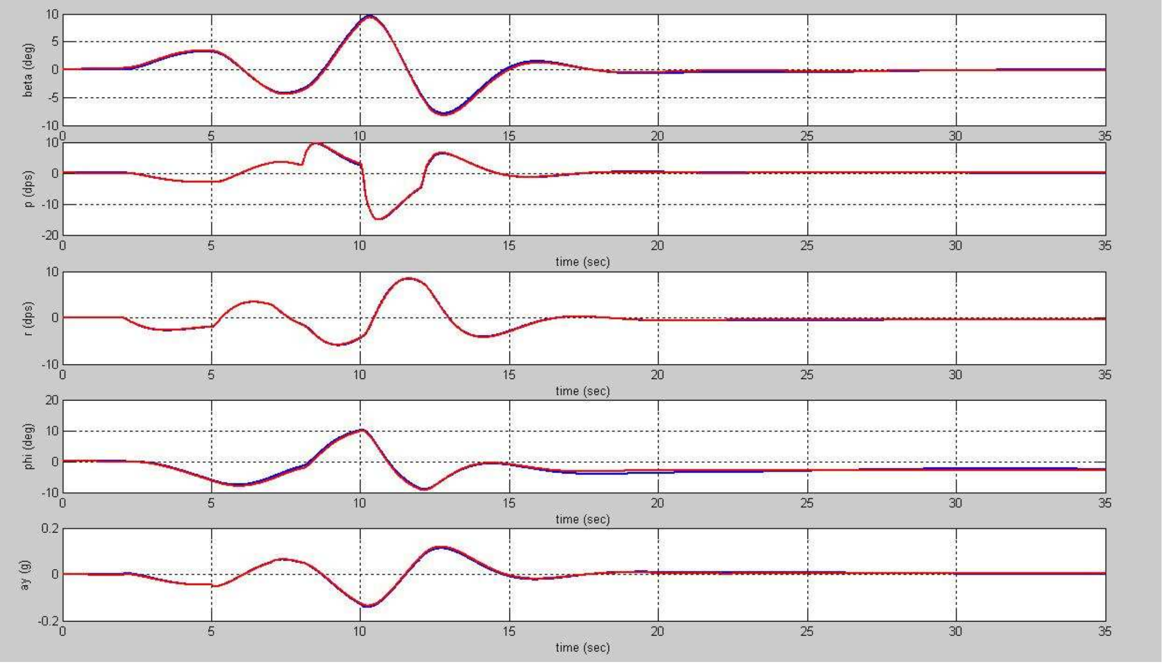

Figure 3.6 Time domain output-error model (red) fit measured data (blue)

The same linear dynamic model (3.6) was used to estimate the non-dimensional aerodynamic coefficients. The cost gradient and the information matrix were computed using (2.123) and (2.124), respectively. Parameter estimation results are also written in Table 3.3-3.5. It can be seen on Figure 3.7 that predicted outputs match quite well to the measured data.

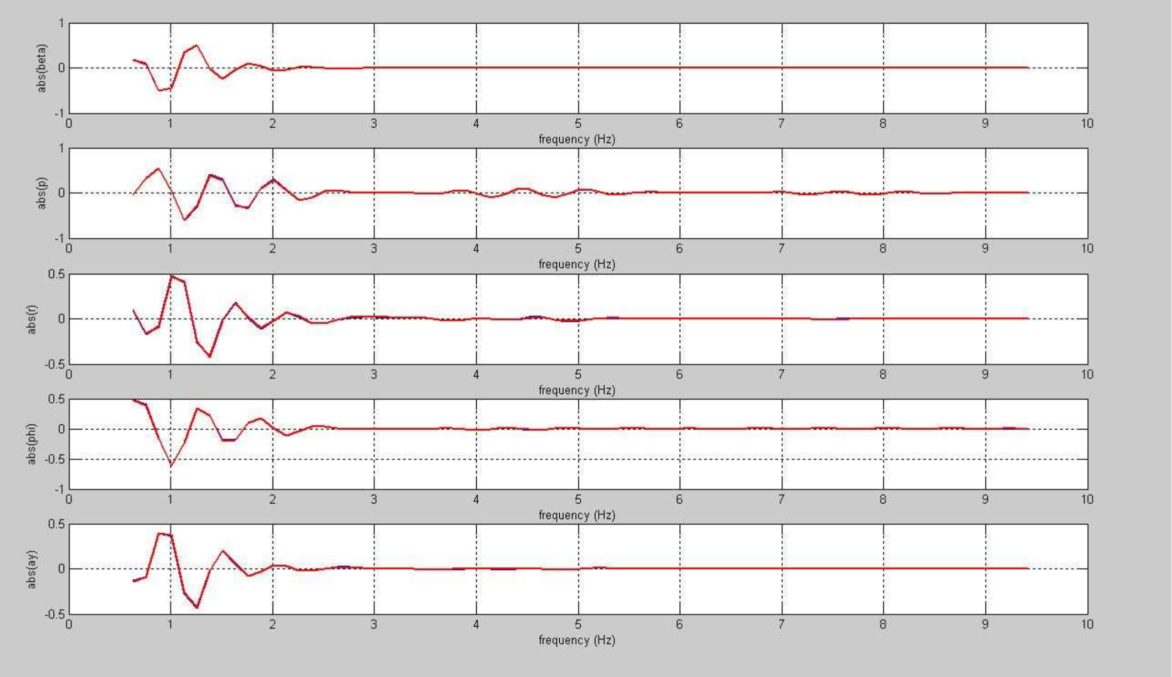

Figure 3.7 Frequency domain output-error model (red) fit measured data (blue)

In Table 3.3-3.5 parameter estimation results are listed. In the first column, equation-error least squares estimates, based on (3.4) directly, can be found. The second column contains parameter estimation results applying the same method as in the first column. However, a reduced linear dynamic model (3.5) was applied. These values were used as initial parameter values in both time and frequency domain output-error method. Third and fourth columns show the results obtained by applying time and frequency domain output-error method, respectively. Real values of the coefficients are written in the last column. Note that in the case of frequency domain methods, constant parameters cannot be estimated because biases and trends were removed before Fourier transformation. The constant parameters and other biases must be estimated in time domain. Furthermore, it can be seen that the estimation procedures have to be developed in order to obtain more precise parameter estimates. Note that the use of the identification of an aircraft with known parameters is a verification procedure of the applied methods. In the case of real flight data there will also be measurement noises and any other disturbances which make the estimation procedure more complex.

Table 3.3 Parameter estimation results for C_Y

	C_Y Parameter				
Parameter	Nonlinear model, LS	Linear model, LS	Output-error in time domain	Output-error in freq. domain	Real value
constant	-0.0142	-0.0136	-0.0016	-	-0,002226
β	-0.7653	-0.7497	-0.7185	-0.7166	-0,7678
$\frac{pb}{2V}$	-0.1092	-	-	-	-0,1240
$\frac{rb}{2V}$	0.3809	0.3662	0.4614	0.3761	0,3666
δ_a	-0.0280	-	-	-	-0,02956
δ_r	0.0601	0.2035	0.2094	0.1502	0,1158
$\delta_r\alpha$	0.8853	-	-	-	0,5238
$\frac{\dot{\beta}b}{2V}$	-0.0352	-	-	-	-0,1600

Table 3.4 Parameter estimation results for C_n

	C_n Parameter				
Parameter	Nonlinear model, LS	Linear model, LS	Output-error in time domain	Output-error in freq. domain	Real value
constant	0.0013	0.0013	-0.0028	-	-0,003117
β	0.0067	0.0106	0.0110	0.0131	0,006719
$\frac{pb}{2V}$	-0.1585	-0.1639	-0.1398	-0.1202	-0,1585
$\frac{rb}{2V}$	-0.1112	-0.0875	-0.1087	-0.1259	-0,1112
δ_a	-0.0039	-0.0075	-0.0018	0.0043	-0,003872
δ_r	-0.0826	-0.0799	-0.0809	-0.0864	-0,08265
δ_r	0.1595	-	-	-	0,1595
β^3	0.1373	-	-	-	0,1373

Table 3.5 Parameter estimation results for C_l

	C_l Parameter				
Parameter	Nonlinear model, LS	Linear model, LS	Output-error in time domain	Output-error in freq. domain	Real value
constant	-0.0036	-0.0036	0.0043	-	0,0005910
β	-0.0615	-0.0616	-0.0566	-0.0572	-0,06180
$\frac{pb}{2V}$	-0.5040	-0.5045	-0.5013	-0.4942	-0,5045
$\frac{rb}{2V}$	0.1719	0.1722	0.2000	0.1632	0,1695
δ_a	-0.0992	-0.1122	-0.1111	-0.1047	-0,09917
δ_r	0.0072	0.0073	0.0144	0.0017	0,006934
$\delta_\alpha \alpha$	-0.0828	-	-	-	-0,08269

4 Summary and concluding results

In this work, time and frequency domain maximum likelihood identification methods were presented. Particular focus was placed on the output-error methods. In order to calculate decent initial values for the recursive output-error algorithms, equation-error least squares methods were used in time and in frequency domain as well. These latter methods were in the focus of a previous work [4]. According the obtained results, it can be seen that the applied methods are accurate in the estimation of non-dimensional aerodynamic parameters. Although, estimated parameters are not exactly the same as the real values; however, they are precise enough to build a decent linear dynamic model of the aircraft, as documented in Figures 3.6 and 3.7 both in frequency and time domain. Concerning the nonlinear terms in the vector-polynomial function (3.4), they can be estimated by applying equation-error least squares method.

In reality, the measurements are always corrupted by noises or any other disturbances. Because of this, test flights are preferable on calm days without winds and gusts. Hence, other signal processing procedures will have to be used before identification methods. In addition, a part of the necessary variables sometimes are not measurable on the board. So, these variables have to be pre-estimated by applying; for instance: Kalman filtering.

In the following, longitudinal non-dimensional parameters will also be estimated by applying identification methods presented above. Then, additional measurement noises will be used in the virtual test flights. In other words, an adequate noise model of the sensors will be applied during simulations to test filtering and smoothing methods. Finally, the given and well tested identification algorithms will be used in the case of real measurements gathered on a UAV (Unmanned Aerial Vehicle).

Future research goal may be the identification of a real aircraft based on real flight data. In order to use the algorithms tested here, it is necessary to develop the fusion of GPS, IMU and magnetometer data and solve the state estimation problem. The methods presented in this work solved the first part of this large program and documented the applicability of the parameter estimation methods.

5 Appendix

A. List of figures and tables

B. References

[1] Vladislav Klein and Eugene A. Morelli, Aircraft System Identification: Theory and Practice, 1st ed.: AIAA, 2006

[2] Micheal Verhaegen and Vincent Verdult, Filtering and System Identification: A Least Squares Approach, Cambridge, 2007

[3] Micheal Verhaegen, Vincent Verdult and Niek Bergboer, Filtering and System Identification: An Introduction to using Matlab Software, TU Delft, 2007

[4] Gerlai Tibor, Vízer Dániel, Repülőgép nemlineáris dinamukus modelljének identifikációja szenzorfúzióval, BME-IIT, TDK dolgozat, 2010

[5] Brian L. Stevens, Frank L. Lewis, Aircraft Control and Simulation, John Wiley and Sons, 1992

[6] Hungarian Academy of Sciences, Computer and Automation Research Institute, Systems and Control Laboratory, Flight data sequences measured on UltraStick-25

[7] Marc Rauw, FDC 1.2, A Simulink Toolbox for flight data analysis, 2011

[8] Bauer Péter, Háromüzemmódú kibővített Kalman szűrők repülőgép orientációjának becslésére, 2010 Repüléstudoányi közlemények