Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2016

Real-Time Estimation of Aerodynamic Parameters

Sofia Larsson Cahlin

LINKÖPING
UNIVERSITY

Master of Science Thesis in Electrical Engineering

Real-Time Estimation of Aerodynamic Parameters

Sofia Larsson Cahlin

LiTH-ISY-EX--16/4929--SE

Supervisor: **Jonas Linder**
ISY, Linköping University
Roger Larsson
SAAB Aeronautics

Examiner: **Martin Enqvist**
ISY, Linköping University

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

Extensive testing is performed when a new aircraft is developed. Flight testing is costly and time consuming but there are aspects of the process that can be made more efficient. A program that estimates aerodynamic parameters during flight could be used as a tool when deciding to continue or abort a flight from a safety or data collecting perspective. The algorithm of such a program must function in real time, which for this application would mean a maximum delay of a couple of seconds, and it must handle telemetric data, which might have missing samples in the data stream.

Here, a conceptual program for real-time estimation of aerodynamic parameters is developed. Two estimation methods and four methods for handling of missing data are compared. The comparisons are performed using both simulated data and real flight test data.

The first estimation method uses the least squares algorithm in the frequency domain and is based on the chirp z-transform. The second estimation method is created by adding boundary terms in the frequency domain differentiation and instrumental variables to the first method. The added boundary terms result in better estimates at the beginning of the excitation and the instrumental variables result in a smaller bias when the noise levels are high. The second method is therefore chosen in the algorithm of the conceptual program as it is judged to have a better performance than the first. The sequential property of the transform ensures functionality in real-time and the program has a maximum delay of just above one second.

The four compared methods for handling missing data are to discard the missing data, hold the previous value, use linear interpolation or regard the missing samples as variations in the sample time. The linear interpolation method performs best on analytical data and is compared to the variable sample time method using simulated data. The results of the comparison using simulated data varies depending on the other implementation choices but neither method is found to give unbiased results. In the conceptual program, the variable sample time method is chosen as it gives a lower variance and is preferable from an implementational point of view.

Sammanfattning

Omfattande provning genomförs när ett nytt flygplan utvecklas. Flygprov är dyra och tidskrävande men det finns aspekter i processen som kan effektiviseras. Ett program som skattar aerodynamiska derivator under flygning skulle kunna användas som ett verktyg för att avgöra om man ska avbryta eller fortsätta en flygning ur ett säkerhets- eller datainsamlingsperspektiv. Algoritmen för ett sådant program måste fungera i realtid, vilket för detta användningsområde innebär en maximal fördröjning på ett par sekunder, och den måste hantera telemetrerad data, vilken kan ha sampel som saknas i dataströmmen.

Här har ett konceptuellt program för realtidsskattning av aerodynamiska parametrar utvecklats. Två skattningsmetoder och fyra metoder för att hantera saknad data jämförs. Jämförelserna görs både med hjälp av simulerad data och riktig flygprovdata.

Den första skattningsmetoden använder minstakvadratmetoden i frekvensdomänen och baseras på chirp z-transformen. Den andra skattningsmetoden skapas genom att lägga till extra termer i deriveringen i frekvensdomänen och instrumentvariabler till den första metoden. De extra termerna resulterar i bättre skattningar i början av excitationen och instrumentvariablerna resulterar i en mindre bias vid höga brusnivåer. Den andra metoden väljs därför som algoritmen i det konceptuella programmet då den bedöms prestera bättre än den första. Den sekventiella egenskapen hos transformen säkerställer programmets funktionalitet i realtid och det har en maximal fördröjning på drygt en sekund.

De fyra metoder för att hantera saknad data som jämförs är att bortse från den saknade datan, hålla föregående värde konstant, använda linjär interpolation eller att se de saknade samplen som variationer i sampeltiden. Linjärinterpolationsmetoden presterar bäst på analytisk data och jämförs med variabel sampeltidsmetoden på simulerad data. Resultaten från jämförelsen varierar beroende på övriga implementationsval men ingen av metoderna ger väntevärdesriktiga resultat. I det konceptuella programmet väljs variabel sampeltids-metoden då denna ger en lägre varians och är att föredra ur ett implementationshänseende.

Acknowledgments

First, I would like to thank those at the departments of Flight Test & Verification and Aeronautical Engineering & Weapons that have been involved in this work. Thank you for taking the time to answer all of my questions. I would especially like to thank my supervisor Roger for giving me the chance to do this interesting work and for all his support along the way. I would also like to thank my supervisor Jonas and my examiner Martin for all their help, time and commitment.

Linköping, April 2016
Sofia Larsson Cahlin

Contents

Notation

Abbreviations

Abbreviation	Meaning
AB	Aktiebolag/ Private Corporation
ARES	Aircraft Rigid body Engineering Simulation
CIV	Complex Instrumental Variable
CZT	Chirp z-Transform
CLS	Complex Least Squares
DFT	Discrete Fourier Transform
DOF	Degrees of Freedom
EM	Expectation Maximisation
FCS	Flight Control System
FIR	Finite Impulse Response
GUI	Graphical User Interface
ISA	International Standard Atmosphere
ISO	International Organization for Standardization
IV	Instrumental Variable
JAS	Jakt, attack, spaning/Multi Role
LS	Least Squares
MC	Monte Carlo
NRMSE	Normalised Root Mean Square Error
ROMAC	Real-time On-line Model and Aerodata Control
SAAB	Svenska aeroplan aktiebolaget
VST	Variable Sample Time

Aeronautics

Notation	Meaning
X, Y, Z	Aerodynamic forces on the aircraft
T, C, N	Aerodynamic forces, tangential (=-X), side (=-Y) and normal force (=-Z)
C_T, C_C, C_N	Aerodynamic force coefficients
$\mathcal{L}, \mathcal{M}, \mathcal{N}$	Aerodynamic moments in roll, pitch and yaw
C_l, C_m, C_n	Aerodynamic moment coefficients
C_D	Aerodynamic drag coefficient
C_L	Aerodynamic lift coefficient
$\boldsymbol{F}$	Total force vector, $[F_x, F_y, F_z]^T$
$\boldsymbol{F}_a$	Aerodynamic force vector
$\boldsymbol{F}_g$	Gravitational force vector
$\boldsymbol{F}_e$	Thrust force vector
$\boldsymbol{M}$	Total moment vector, $[M_x, M_y, M_z]^T$
$\boldsymbol{M}_a$	Aerodynamic moment vector
$\boldsymbol{M}_e$	Thrust moment vector
$\boldsymbol{H}$	Total angular momentum vector, $[H_x, H_y, H_z]^T$
$\boldsymbol{H}_e$	Engine angular momentum vector
$\boldsymbol{n}$	Load factor vector
ϕ, θ, ψ	Euler angles, roll, pitch and yaw
u, v, w	Body fixed velocities
p, q, r	Body fixed angular velocities in roll, pitch and yaw
α	Angle of attack
β	Angle of sideslip
M	Mach number, $M = V/a$
$\boldsymbol{I}$	Moment of inertia matrix, in centre of gravity
I_{ij}	Moment of inertia about the i, j-axis, in centre of gravity
q_a	Dynamic pressure
h	Altitude of flight
T	Temperature at altitude of flight
ρ	Air density at altitude of flight
$\mathrm{d}T/\mathrm{d}h, \lambda$	Temperature gradient (Lapse rate)
a	Speed of sound at temperature of flight
V	Airspeed
m	Mass

Constants

Notation	Meaning
h_0	Reference altitude
T_0	Temperature at reference altitude
ρ_0	Air density at reference altitude
R	Gas constant of air
γ	Ratio of specific heats
g	Gravity constant
S	Wing area
c	Mean aerodynamic chord
b	Wing span

Control surfaces

Notation	Meaning
δ_c	Canard deflection angle
δ_{lc}	Left canard deflection angle
δ_{rc}	Right canard deflection angle
δ_{loe}	Left outer elevon deflection angle
δ_{lie}	Left inner elevon deflection angle
δ_{roe}	Right outer elevon deflection angle
δ_{rie}	Right inner elevon deflection angle
δ_e	Elevator deflection angle
δ_a	Aileron deflection angle
δ_r	Rudder deflection angle
δ_{le}	Leading edge deflection angle
δ_{lole}	Left outer leading edge deflection angle
δ_{lile}	Left inner leading edge deflection angle
δ_{role}	Right outer leading edge deflection angle
δ_{rile}	Right inner leading edge deflection angle
δ_{br}	Air brake deflection angle
δ_{lbr}	Left air brake deflection angle
δ_{rbr}	Right air brake deflection angle

System Identification

Notation	Meaning
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$	Linear system matrices
$\boldsymbol{x}$	State vector
$\boldsymbol{u}$	Input vector
$\boldsymbol{y}$	Output vector
$\boldsymbol{w}$	Process noise vector
$\boldsymbol{v}$	Measurement noise vector
$\boldsymbol{\phi}$	Regressor matrix
$\boldsymbol{\theta}$	Parameter matrix
ξ	Instrumental variable matrix
$\tilde{\boldsymbol{Y}}$	Transformed output vector
$\tilde{\boldsymbol{\Phi}}$	Transformed regressor matrix
$\tilde{\boldsymbol{\Xi}}$	Transformed instrumental variable matrix
$\hat{\boldsymbol{\theta}}$	Estimated parameter matrix
ϵ	Prediction error
$\hat{\sigma}$	Estimated residual variance
$\hat{s}$	Estimated standard deviation
n_p	Number of parameters
n_i	Number of inputs
n_o	Number of outputs
n_f	Number of frequencies
T_s	Sampling time
N	Number of samples

1 Introduction

This is the report of a master's thesis work in control conducted at SAAB and examined at Linköping University. In this thesis, two methods for parameter estimation in real time are compared. The intended application is for flight testing at SAAB. Both of the methods use the *Least Squares* (LS) algorithm in the frequency domain and a frequency transformation called the *Chirp z-Transform* (CZT). One of the methods also include *Instrumental Variables* (IV), a method for handling noisy data. Additionally four methods for handling of missing data in real time are compared. For this application, real-time would mean a maximum delay of a couple of seconds, regardless of the length of the experiment. The final result is an implemented conceptual real-time parameter estimation program.

In this first, introductory chapter, some background to the thesis is presented along with its purpose, goal, problem formulation and limitations. The chapter also includes the approach used to reach the specified goal and the outline of the thesis. Throughout the thesis, *the application* will be used to refer to the application of flight testing at SAAB whereas *this work* refers to the implementation and results presented in this thesis.

1.1 Background

Svenska aeroplan aktiebolaget (SAAB) is a Swedish company that provides a range of defence and security products. This thesis work is conducted at the business area within SAAB that concerns the aviation products, SAAB Aeronautics, in Linköping. The main product of SAAB Aeronautics is the JAS 39 Gripen aircraft. In the process of developing the next generation of Gripen, called Gripen E, extensive testing will be performed. Flight testing is costly and time consuming but there are aspects of the process that can be made more efficient. One possible way to improve the flight testing is investigated in this work.

An aircraft such as the JAS 39 Gripen constitutes a system that is highly nonlinear and unstable (Ahlstöm and Torin, 2002). Such an aircraft needs a *Flight Control System* (FCS) in order to execute the pilot's commands and stabilise the aircraft to ensure safe flying. The FCS uses feed-forward with mathematical models of the aircraft, together with feedback of the measured output, to calculate input signals to the aircraft's control surfaces (Ahlstöm and Torin, 2002). Models are also used in simulators that are used for training and to augment the efficiency in the process of designing the FCS. Both of these applications therefore need highly accurate models of the aircraft. The models are initially derived from calculations and wind tunnel data. Flight tests are performed in order to evaluate and validate the derived models, and to increase the accuracy of the estimated parameters (Stavöstrand, 2011).

Two aspects are considered when a flight test is performed. The first is the safety issue and the second is to maximise the information content in the data to enable post-flight analysis with good results. During a flight, several signals are monitored in real time to ensure that the aircraft does not leave the flight envelope and that the actual output does not diverge too much from the expected, simulated output. If the model used in the design of the FCS differs too much from the reality, the FCS might need to be updated and proper safety measures should be taken.

In the current flight test process at SAAB, the information content of the data from the flight test is not analysed until after the flight. If the data is found to be inadequate at this point, the flight must be redone. If there was a tool available, able to determine if the data is sufficient in near real time during the flight test, specific manoeuvres could be redone directly instead of having to schedule a new flight. This would reduce both cost and time spent on flight testing.

By performing parameter estimation in real time, information regarding both the information content of the data and the consistency with existing models can be obtained. One method for this, a recursive LS method in the time domain, is analysed in Andersson (2013). However, this method does not work in real time as intended due to the cost of performing an estimation in each iteration and in Andersson (2013), a frequency domain method is suggested instead. In Larsson (2013), improvements to a sequential frequency domain method presented in Klein and Morelli (2006) are suggested. This method should be better adapted to the real-time application due to the sequential property of the digital Fourier transform and the two methods are therefore compared in this work.

1.2 Purpose

The purpose of this thesis work is to verify whether the specific sequential frequency domain method presented in Klein and Morelli (2006) together with the improvements suggested by Larsson (2013) can be used for real-time parameter estimation. Furthermore, the extent to which the suggested modifications change the results is evaluated and analysed. The purpose of this verification in turn is to make the flight test process more efficient and thereby to reduce the cost and time spent on flight testing.

1.3 Goal

The goal of this thesis work is to produce a program that uses the given method in order to perform parameter estimation in real time. The program should also handle telemetric data to ensure functionality in real time. To reach this goal, a number of partial goals are set. These are listed below.

- The two estimation methods of Klein and Morelli (2006) and Larsson (2013) should be implemented.
- The two estimation methods should be evaluated and compared on simulated data.
- The two estimation methods should be evaluated and compared on real flight test data.
- Methods for handling of telemetric data with missing samples should be chosen and implemented.
- The methods for handling of telemetric data should be evaluated and compared on simulated data.
- A conceptual real-time parameter estimation program should be implemented with methods chosen based on the results of the performed evaluations.
- The final program should be evaluated on telemetric data.

1.4 Limitations

This work only concerns the estimation of parameters in a linearised model of the JAS 39 Gripen aircraft's aerodynamic properties. The algorithm is implemented in Matlab in a SAAB module called StellaRT. The final algorithm is not evaluated live during flight test but on recorded telemetric data.

1.5 Problem Definition

Several questions are addressed in this thesis. One question is how the proposed method should be used on a system using aerodynamic coefficients instead of an ordinary state-space model. Larsson (2013) shows that the suggested improvements result in a more accurate estimator for the state-space model. Are similar results obtained when using the aerodynamic coefficients model?

Another question is how the telemetric data should be handled when aiming to minimise the information loss in the data. The telemetric data received in real time imposes problems that need to be overcome. The data does not arrive sample by sample but in batches of irregular size, at irregular times. There might also be sequences of data that are missing in the data stream. How does this affect the implementation and performance of the estimation algorithm?

1.6 Approach

The tools used throughout this project are Matlab, Simulink and Matlab toolboxes available at SAAB.

In order to verify the functionality of the methods, a simulation model with known parameters is implemented. After implementation of the algorithm, it is tested on simulated data from the known model. The method is then tested on real flight test data.

An analysis of the telemetric data is performed to be able to simulate a telemetric data flow. Using the simulated telemetric data, solutions to the related problems are developed, implemented and tested.

1.7 Thesis Outline

In Chapter 2, a brief introduction to system identification is given, followed by a more thorough presentation of each of the methods used in this work. In Chapter 3, the model that contains the unknown parameters is derived. For the interested reader, the flight mechanical models are derived and explained in Section 3.2. However, the results of this work can be understood given only the expressions in (3.5) and (3.9) together with the presentation of the aerodynamic model i Section 3.3. In sections 4 and 5, the practical details of this work are presented. These include a description of the simulation model used for comparing the different methods, the requirements on the final estimation program, a description of the utilised software and some properties of the flight test data.

The results of this work are presented in chapters 6 to 8. The comparison between the two estimation algorithms is presented in Chapter 6. In Chapter 7, the comparisons of the different methods for handling missing data are presented. The implementation of the final conceptual program is presented in Chapter 8 together with the results and performance when using simulated, real-time flight test data. The results are discussed in Chapter 9 before the conclusions och future work are presented in Chapter 10.

2
System Identification

In this chapter, the basic concept of system identification is presented along with the mathematical tools needed in this particular application. The two specific algorithms studied in this work use the *Complex Least Squares* (CLS) parameter estimation method. The *Least Squares* (LS) regression is transformed into the frequency domain using a transform method called the *Chirp z-Transform* (CZT). The CLS and the CZT are therefore presented initially, before describing the two estimation algorithms. Then details regarding telemetric data and data quality at SAAB are presented along with the different methods for handling of missing data investigated in this thesis. Lastly, the goodness of fit, which is the validation metric used, is presented. Initially, however, a brief introduction to system identification is provided.

For further details and deeper understanding, a number of references are available. The specific method investigated is taken from Klein and Morelli (2006) and the suggested modifications come from Larsson (2013). Ljung (1999) gives an introduction to the area of system identification in general while Pintelon and Schoukens (2001) focus on the frequency domain methods.

2.1 Introduction to System Identification

System identification is the problem of finding a description of the system S by observing its input $\boldsymbol{u}$ and output $\boldsymbol{y}$. In accordance with Klein and Morelli (2006), this is one of the three main problems of aircraft dynamics and control, the other two being control and simulation. Control is the problem of finding the proper input $\boldsymbol{u}$ to create a desired output $\boldsymbol{y}$ from a known system S. Simulation is the problem of finding the expected output $\boldsymbol{y}$ that a given input $\boldsymbol{u}$ to a known system

S will create. In both problems above, the known system S is required, making system identification an essential part of both control and simulation.

The system of interest in applications of aircraft dynamics and control is the aircraft itself. The aircraft is influenced by measurable inputs such as pilot commands, and non-measurable inputs, called disturbances, such as wind or turbulence. All these inputs together create outputs from the system.

The relationship between observed inputs and outputs of a system is called a model. There are three types of models and using the framework of Ljung (1999) these are mental, graphical and mathematical models. Mental models are intuitive models learnt from experience and with no mathematical formulation. An example of a mental model is the knowledge required to ride a bike. A graphical model is a model where the system is described by using tables or plots of numerical data from the system. A graphical model does not include any mathematical relation between an input and an output, only an association between the two. The mathematical model however, describes the relationship between inputs and outputs analytically, typically, for dynamic systems, using differential equations. The process of deriving a model using physical laws and other previous knowledge of the system is called modelling. If this is not possible, data from the system can be used instead. The process of finding a mathematical or graphical model of a system by observing and analysing data from the system is called system identification. In Gripen, all three model types are used. The pilot uses a mental model to steer the plane whereas the FCS uses both graphical and mathematical models. More details about these are presented in Chapter 3.

If no a priori knowledge about the system is available, it is possible to use several techniques known as *black box* techniques that use standard models with parameters without physical meaning and fit them to the data. In this application, however, the model structure is known but contains physical parameters for which the true values are unknown. A model with known structure such as this is called a *grey box* model. The LS method is used to estimate the unknown parameters. The estimation method is described in the coming sections and the physical model structure is described in Chapter 3.

2.2 Parameter Estimation Using Complex Least-Squares

The estimation method used in the studied algorithms is the *Complex Least Squares* (CLS). The CLS is the equivalent of the ordinary LS but used in the frequency domain. The ordinary LS is found for example in Ljung (1999) and Söderström and Stoica (1989). It is a method for estimating linear parameters in a relation such as

$$y = \phi\theta \tag{2.1}$$

where $\boldsymbol{y}$ is the measured output, $\boldsymbol{\phi}$ contains the regressors and $\boldsymbol{\theta}$ contains the parameters to be estimated. For example, the regressors can be the measurable inputs and states that influence the system. With one output, n_p unknown parameters and N observed data points of the output and regressors, $\boldsymbol{y}$ is an N-dimensional column vector and $\boldsymbol{\phi}$ is an $N \times n_p$ matrix. The method finds the estimate that gives the best fit between the estimated and the real output by minimising the cost function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} \epsilon_i^2 = \sum_{i=1}^{N} (\boldsymbol{y}_i - \boldsymbol{\phi}_i \boldsymbol{\theta})^2 = (\boldsymbol{y} - \boldsymbol{\phi}\boldsymbol{\theta})^T (\boldsymbol{y} - \boldsymbol{\phi}\boldsymbol{\theta}) \tag{2.2}$$

where ϵ_i is the prediction error for the i^{th} sample and $\boldsymbol{y}_i$ and $\boldsymbol{\phi}_i$ indicate the i^{th} row of $\boldsymbol{y}$ and $\boldsymbol{\phi}$ respectively. Since (2.2) is quadratic in $\boldsymbol{\theta}$, it can be minimised analytically by putting its derivative to zero. The parameter estimate can then in principle be calculated as

$$\hat{\boldsymbol{\theta}} = \left(\sum_{i=1}^{N} \boldsymbol{\phi}_i^T \boldsymbol{\phi}_i \right)^{-1} \sum_{i=1}^{N} \boldsymbol{\phi}_i^T \boldsymbol{y}_i = \left(\boldsymbol{\phi}^T \boldsymbol{\phi} \right)^{-1} \boldsymbol{\phi}^T \boldsymbol{y}. \tag{2.3}$$

If the LS estimate is calculated using this batch algorithm, the matrices will grow as N grows, making the computations in (2.3) more time consuming. This is not ideal for a real-time application where the estimation shall be done at regular intervals. However, there is a recursive LS algorithm, described, for example, in Ljung (1999) and Klein and Morelli (2006). This algorithm uses the previous estimate together with the newest data points to make a new estimate, meaning that an estimation is performed every time new data is presented. This algorithm was used in Andersson (2013) but was still found too time consuming for the real-time application used at SAAB and a frequency domain approach was suggested as a remedy.

The use of a frequency domain method has two advantages with regard to a time efficient calculation of the LS estimate. Firstly, as explained later in Section 2.3.1, the number of frequencies n_f can be fixed meaning that the matrices $\boldsymbol{y}$ and $\boldsymbol{\phi}$ will have fixed sizes of $n_f \times 1$ and $n_f \times n_p$, respectively. As is the case with the recursive LS algorithm, this fixates the complexity, and the computation time, also when using the batch LS algorithm. Secondly, as explained later in Section 2.3.2, the transform can be calculated using a sequential method. Using this method, the value of the transform is updated every time new data arrives and no previous data points need to be saved. This makes the algorithm less time consuming since a new estimate does not have to be calculated each time new data is presented, unlike when using the recursive LS algorithm. Instead, the transform is calculated each time new data is presented and the estimation is only performed at specified times using the batch LS algorithm. In the frequency domain, the CLS method is used. The CLS estimator is the same as the LS estimator for real numbers but with complex conjugate transposes instead of only transposes (Klein and Morelli, 2006).

2.3 Frequency Domain Estimation Methods

The two estimation methods that are compared in this work are presented in Section 2.3.2 and Section 2.3.3. They are both sequential frequency domain estimation methods and the first is taken from Klein and Morelli (2006). The second method is presented in Larsson (2013). It is based on the first method but includes two modifications. The first method, from Klein and Morelli (2006), is referred to as Morelli's method and the other, from Larsson (2013), is referred to as Larsson's method. In both identification methods, a special transform method called the *Chirp z-Transform* (CZT) is used. This method has a fixed number of frequencies which eliminates the resolution's dependence on the number of samples in the time domain. Additionally, it limits the frequency interval to the actual interval of interest. A description of the CZT that relates it to more conventional transform methods is therefore given first in Section 2.3.1 before presenting the two estimation methods.

2.3.1 The Fourier Transform and the Chirp z-Transform

The CZT is obtained by slightly modifying the finite Fourier transform. The resulting transform has two properties that make it suitable for the identification of aircraft aerodynamics. The first property is that it is finite, coming from the finite Fourier transform, and the second is that it allows for arbitrary choices of frequencies. Sundararajan (2001) explains the basic concepts of the Fourier transform, while Cooley et al. (1969) describe the finite Fourier transform and the *discrete Fourier transform* (DFT). The CZT is defined in Rabiner et al. (1969a) and discussed in further detail in Rabiner et al. (1969b).

The finite Fourier transform of a continuous-time signal that is sampled at time intervals T_s can be approximated as

$$\tilde{f}(\omega) = \int_0^T f(t)e^{-j\omega t}dt \approx T_s \sum_{k=0}^{N-2} f_k e^{-j\omega k T_s} \tag{2.4}$$

using a zeroth-order Euler approximation in accordance with Morelli (1997). The constant N is the total number of samples, meaning that $T = (N-1)T_s$. When integrating from 0 to T, the final sample does not contribute to the integral and the final index of the summation becomes $N-2$, as illustrated in Figure 2.1.

The summation in (2.4) can be compared to the z-transform

$$\tilde{x}(z) = \sum_{k=-\infty}^{\infty} x[k]z^{-k} \tag{2.5}$$

where $x[k]$ are samples of a discrete signal x with index k and $z \in \mathbb{C}$. The summation in (2.4) corresponds to the summation in (2.5) mulitipied with T_s, only with the summation limited to go from 0 to $N-2$ and with z limited to values $e^{j\omega T_s}$, meaning the unit circle, since f_k and $x[k]$ are the sample values of the two signals respectively.

If the frequencies ω_i are chosen as

$$\omega_i = 2\pi f_i = 2\pi \frac{i}{(N-1)T_s}, \qquad i = 0, 1, \ldots, N-2 \tag{2.6}$$

which is conventional, the approximation in (2.4) becomes

$$\tilde{f}(\tfrac{2\pi i}{(N-1)T_s}) \approx T_s \sum_{k=0}^{N-2} f_k e^{-j(2\pi i/(N-1))k}, \qquad i = 0, 1, \ldots, N-2 \tag{2.7}$$

which is the *discrete Fourier transform* (DFT) for $N-1$ samples multiplied with the sampling interval T_s. The DFT is consequently a special case of the z-transform with $N-1$ values of z evenly spaced around the unit circle with an interval of $2\pi/(N-1)$. As can be seen in (2.6), however, the frequency resolution and the number of frequencies are given by T_s and N. This means that for a low N, the frequency resolution is low, and for a high N many frequencies are outside the interval of interest.

Although the above choice of contour that gives the DFT is the most commonly used, other contours in the z-plane could be chosen instead. Rabiner et al. (1969a) present a method that allows for the calculation of the z-transform on any spiral, or circle, with arbitrary starting point. This is called the *Chirp z-Transform* (CZT). In this application, an arc of the unit circle is chosen and the number of frequencies is fixed which eliminates the resolution's dependance on the number of samples in the time domain. A sufficient number of samples is however still needed for the estimation quality. Additionally, the frequency interval is limited to the actual interval of interest. According to Morelli (1999), a frequency band

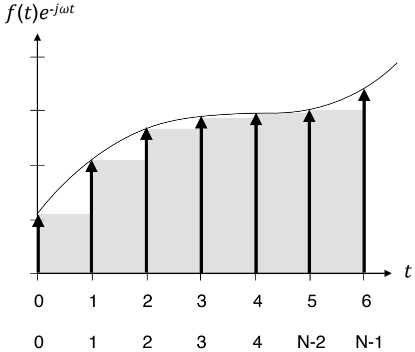


Figure 2.1: *The approximation of the continuous-time Fourier transform. The area under the graph is approximated by summing the area of each rectangle with height $f(t)e^{-j\omega t}$ and width T_s. In this example, $T = 6$ seconds and $N = 7$ samples. In order for the area to be correct, the summation ends at the second last sample, having the index $N-2$ meaning that only $N-1$ sample values are used.*

of [0.1 1.5] Hz with a resolution of 0.04 Hz is adequate for this application. However, according to Klein and Morelli (2006), the frequency interval of interest for rigid body aircraft dynamics is [0.01 2.0] Hz. Larsson (2013) uses the frequency band [0.1 2] Hz and the same frequency band is chosen also in this work to make the results comparable. This gives a frequency vector $\boldsymbol{\omega}$ of fixed length $n_f = 48$, which also has other advantages as previously discussed in Section 2.2.

Using (2.4) normalised with $1/T_s$ and the 48 chosen frequencies within [0.1 2.0] Hz, the final CZT transform used in this work becomes

$$\mathscr{F}_{\text{CZT}}(f(t)) = \tilde{F}(\omega_i) = \sum_{k=0}^{N-2} f_k e^{-j\omega_i k T_s}, \qquad \omega_i \in \boldsymbol{\omega} = \{0.10, 0.14, \ldots, 1.98\}. \tag{2.8}$$

2.3.2 Morelli's Sequential Frequency Domain Method

In this section, the original sequential frequency domain method taken from Klein and Morelli (2006) is presented.

The transform of (2.8) can be calculated sequentially as

$$\begin{aligned}\tilde{F}(\omega_i) = \sum_{k=0}^{N-2} f_k e^{-j\omega_i k T_s} = \sum_{k=0}^{N-3} f_k e^{-j\omega_i k T_s} + f_{N-2} e^{-j\omega_i (N-2) T_s} = \\ = \tilde{F}_{-1}(\omega_i) + f_{N-2} e^{-j\omega_i (N-2) T_s}, \qquad \omega_i \in \boldsymbol{\omega}\end{aligned} \tag{2.9}$$

where the −1 subscript denotes the previously calculated transform. This sequential property is what makes the frequency domain approach suitable for a real-time application. The transform is simply updated every time new data arrives. By using the CZT to transform the linear relation of (2.1), the complex regression becomes

$$\tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{\Phi}}\boldsymbol{\theta} \tag{2.10}$$

where $\tilde{\boldsymbol{Y}}$ and $\tilde{\boldsymbol{\Phi}}$ are the transformed output and regressor matrices, respectively. The parameter vector $\boldsymbol{\theta}$ remains the same due to the linearity of the transform. Due to the transform, the dimensions of the matrices change. The output vector that in the time domain is an N-dimensional column vector becomes an n_f-dimensional column vector in the frequency domain. The regressor matrix $\boldsymbol{\phi}$ is an $N \times n_p$ matrix in the time domain and an $n_f \times n_p$ matrix in the frequency domain.

The CLS estimate is calculated as

$$\hat{\boldsymbol{\theta}} = \left(\tilde{\boldsymbol{\Phi}}^* \tilde{\boldsymbol{\Phi}}\right)^{-1} \tilde{\boldsymbol{\Phi}}^* \tilde{\boldsymbol{Y}} \tag{2.11}$$

which, according to Klein and Morelli (2006), can be simplified to

$$\hat{\boldsymbol{\theta}} = \left(\text{Re}\left(\tilde{\boldsymbol{\Phi}}^* \tilde{\boldsymbol{\Phi}}\right)\right)^{-1} \text{Re}\left(\tilde{\boldsymbol{\Phi}}^* \tilde{\boldsymbol{Y}}\right) \tag{2.12}$$

since the parameter vector is known to be real in the aircraft application. Similarly, the estimated covariance matrix is also simplified and becomes

$$\hat{\mathrm{C}}\mathrm{ov}(\hat{\boldsymbol{\theta}}) = \hat{\sigma}^2 \left(\mathrm{Re}\left(\tilde{\boldsymbol{\Phi}}^* \tilde{\boldsymbol{\Phi}}\right)\right)^{-1} \tag{2.13}$$

where the estimate of the residual variance is

$$\hat{\sigma}^2 = \frac{1}{n_f - n_p}((\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{\Phi}}\hat{\boldsymbol{\theta}})^*(\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{\Phi}}\hat{\boldsymbol{\theta}})). \tag{2.14}$$

By taking the square root of the diagonal elements of the estimated covariance matrix, the standard deviations of the parameters can be calculated as

$$\hat{s}_i = \sqrt{\hat{\mathrm{C}}\mathrm{ov}_{ii}(\hat{\boldsymbol{\theta}})}, \qquad i = 1, \ldots, n_p \tag{2.15}$$

and these are used to quantify the quality of the estimates.

When calculating the CZT of a derivative, the approximation

$$\tilde{\dot{F}}(\omega_i) = \sum_{k=0}^{N-2} \dot{f}_k e^{-j\omega_i k T_s} \approx j\omega_i \tilde{F}(\omega_i) \tag{2.16}$$

is used in Klein and Morelli (2006), which is the relation that holds for the ordinary Fourier transform of a derivative.

2.3.3 Larsson's Improved Sequential Frequency Domain Method

Larsson (2013) suggests two improvements to the previously presented method from Klein and Morelli (2006). These consist of a correction of the frequency domain differentiation of (2.16) and the addition of *Instrumental Variables* (IV).

Corrected Differentiation in the Frequency Domain

As described in Larsson (2013), the transform of the derivative in (2.16) is true only for the ordinary Fourier transform but not for the finite Fourier transform. By using integration by parts

$$\tilde{\dot{f}}(\omega) = \int_{t_0}^{t_1} \dot{f}(t) e^{-j\omega t} dt = j\omega \int_{t_0}^{t_1} f(t) e^{-j\omega t} dt + [f(t) e^{-j\omega t}]_{t_0}^{t_1} \tag{2.17}$$

the ordinary Fourier transform of a derivative, $\tilde{\dot{f}}(\omega) = j\omega \tilde{f}(\omega)$ is obtained if $t_0 = -\infty$ and $t_1 = +\infty$. Since this is not the case for the finite Fourier transform, the second term in (2.17) must be kept. This gives

$$\tilde{\dot{f}}(\omega) = j\omega \tilde{f}(\omega) + f(t_1) e^{-j\omega t_1} - f(0) \tag{2.18}$$

since $t_0 = 0$ here. Using the normalised transform in (2.8), the final transform with added extra terms becomes

$$\tilde{\dot{F}}(\omega_i) = j\omega_i \tilde{F}(\omega) + \frac{1}{T_s}\left(f(t_1) e^{-j\omega_i t_1} - f(0)\right), \qquad \omega_i \in \boldsymbol{\omega}. \tag{2.19}$$

Larsson (2013) found that the approximation in (2.16) results in a bias during the excitation. For a state space model, this bias was shown to be reduced by adding the extra terms. It is investigated in this work whether the same improvement can be found for a model using aerodynamic coefficients.

Instrumental Variables

For a LS estimate to be consistent, the noise has to be white and uncorrelated with the regressors (Ljung, 1999). This is not true in the Gripen case for three reasons as described in Larsson (2013). Firstly, the atmospheric turbulence, that accounts for most of the process noise, is not white. Secondly, the FCS will try to compensate for the turbulence and thereby create a correlation between the disturbances and the control surface displacements, which are regressors. Thirdly, a correlation between measurement noise and regressors is created since the states are approximated from the measurements. These problems are shown in Larsson (2013) to result in biased estimates for noise levels corresponding to medium and severe atmospheric turbulence. The consistency is however improved by using *Complex Instrumental Variables* (CIVs), or *Instrumental Variables* (IVs) for short.

Instrumental variables ξ are any variables that are correlated with the regressors but uncorrelated with the noise. Using IVs will then give the IV estimator

$$\hat{\boldsymbol{\theta}} = \left(\xi^T \phi\right)^{-1} \xi^T y \tag{2.20}$$

that can be compared to the estimator in (2.3) (Ljung, 1999). When transformed to the frequency domain using the CZT, the IV estimator becomes

$$\hat{\boldsymbol{\theta}} = \left(\mathrm{Re}\left(\tilde{\boldsymbol{\Xi}}^* \tilde{\boldsymbol{\Phi}}\right)\right)^{-1} \mathrm{Re}\left(\tilde{\boldsymbol{\Xi}}^* \tilde{\boldsymbol{Y}}\right). \tag{2.21}$$

In this application, a parallel simulation is used to create the instrumental variables. The existing simulation model is run in parallel with the flight test using the true pilot commands as inputs. This creates regressors that approximate the true regressors but that are uncorrelated with the process and measurement noise. Other examples of instrumental variables and further theory can be found, for instance, in Söderström and Stoica (1983).

In Larsson (2013), the estimated standard deviation of the estimates is incorrectly calculated. According to Pintelon and Schoukens (2001) the estimated covariance matrix when using IVs is calculated as

$$\hat{\mathrm{C}}\mathrm{ov}(\hat{\boldsymbol{\theta}}) = \hat{\sigma}^2 \left(\mathrm{Re}\left(\tilde{\boldsymbol{\Xi}}^* \tilde{\boldsymbol{\Phi}}\right)\right)^{-1} \mathrm{Re}\left(\tilde{\boldsymbol{\Xi}}^* \tilde{\boldsymbol{\Xi}}\right) \left(\mathrm{Re}\left(\tilde{\boldsymbol{\Xi}}^* \tilde{\boldsymbol{\Phi}}\right)\right)^{-T} \tag{2.22}$$

and this calculation is used in this work. If the regression vector is chosen as the IVs, the results of Morelli's algorithm presented in Section 2.3.2 are obtained once again. In that case, the only difference between Larsson's and Morelli's algorithms consists of the corrected differentiation in Larsson's algorithm.

2.4 Telemetric Data and Missing Data

There are several possible reasons for data to be missing in collected data sets. Wallin (2004) gives the examples of temporary faults in data transmission and temporary sensor failure, which are both common in applications. The missing data causes difficulties when extracting information from the data set. Several solutions to the related problems are suggested, for example in Wallin (2004) and Pintelon and Schoukens (1999). This section will describe the data quality at SAAB and thereby the conditions that the chosen algorithm must handle along with some examples of different algorithms.

2.4.1 Telemetric Data at SAAB

The following details regarding the telemetric data at SAAB have been provided by Mats Svensson who works at the department *Instrumentation and software systems* at SAAB Aeronautics (Svensson, 2015).

During flight tests at SAAB, all sensor data is saved locally in the aircraft and is accessible after the flight is finished. The locally saved data is complete and contains all data except for data missing due to sensor failure. However, sensor failure is rare and not specifically handled in post flight analysis. Some of the local data is transmitted from the aircraft to the control room during the flight to allow for real-time monitoring. This data is called telemetric data. The telemetric data can have additional missing data points due to errors in the radio transmission. The transmitted data is packed, encrypted and sent at a frequency of 16 Hz in frames of 16 kbyte. Each frame has a check sum in order to verify correct transmission. If the check sum is wrong, the entire frame is disregarded. Since all signals are sent in the same frame, all data, both in- and output, is missing at a disruption. For a 60 Hz sensor signal, this means that about four samples are missing for each disregarded frame. This gives the lower limit for the amount of missing data when there is a disruption.

The upper limit for the amount of missing data, however, cannot be determined exactly. The quality of the transmission depends on many factors including the alignment between the transmitter on the aircraft and the antenna. For this reason, more than one antenna is used for reception of data. Only one antenna, the one giving the best results, is used at a time and all antennas are monitored and switched manually. However, an antenna switch generates a loss of at least one frame. The transmission gets worse at certain heights, if you fly further away from the antenna or in direct alignment with the antenna. It also gets worse if several aircraft are monitored at a time since this disables the use of multiple antennas. Even though there is no exact measure of what is a standard amount of data loss, the algorithm is expected to have to tolerate that there are between 4 and 16 frames of lost data in one manoeuvre, meaning as much as 64 missing samples at worst.

2.4.2 Parameter Estimation with Missing Data

There are three alternatives when handling missing data. You can throw away the data set and redo the experiment, divide the data into several data sets that are each complete and then combine them, or you can estimate the missing data. Since experiments in this application, flight tests, are costly and have high safety requirements, the alternative of throwing away the data set is not an alternative. Because of the additional constraints of more sophisticated methods and the fact that this is an initial study, four simple methods are chosen for study in this work. Ljung (1999) presents a method for combining several data sets. The method weighs the estimates of the separate sets with their respective inverse covariance matrices. The method might be useful in this application but for simplicity, the study is limited to four methods that are even more straight forward.

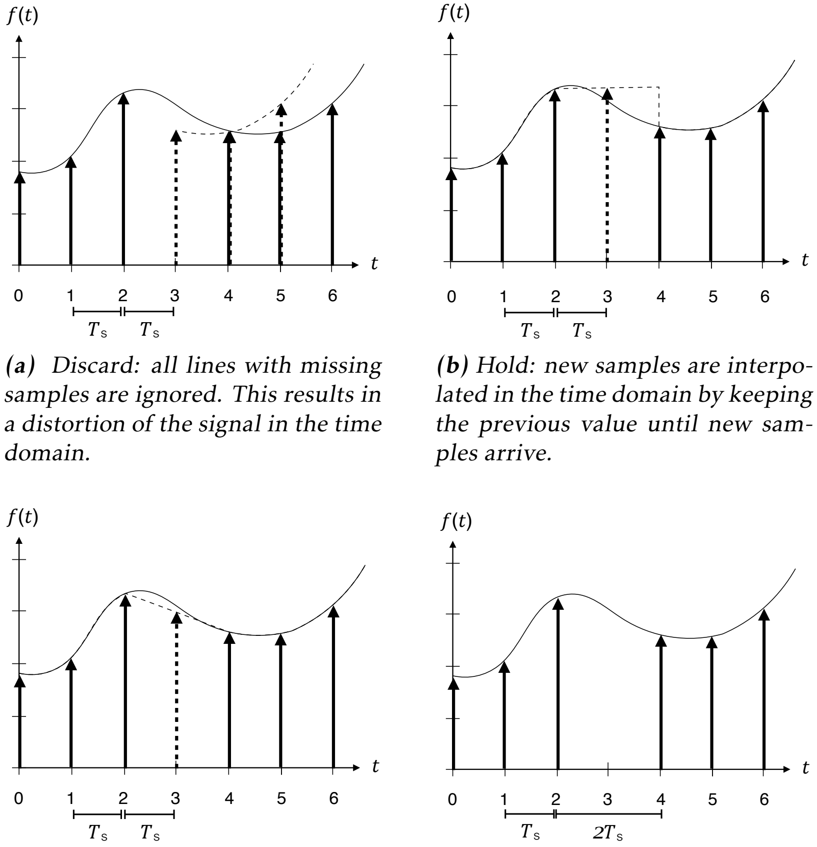
An overview of the four chosen methods is presented in Figure 2.2. According to Wallin (2004), one simple method is to ignore the rows in the vectors of the linear regression of (2.1) where data is missing. This method is presented in Figure 2.2a. As the samples after a disruption are shifted to fill the positions of the missing samples, this results in a distortion of the signal. This is not a problem in the time domain, where the time vector has no part of the estimation, and the method performs well for low numbers of missing data (Wallin, 2004). The performance in the frequency domain, however, needs to be investigated, and this is further discussed in Section 7.2. Another simple method is some sort of interpolation of the missing data. Although interpolation methods always result in biased estimates according to Wallin (2004), the advantages of simplicity might be greater than the disadvantages. Two interpolation methods are chosen for study in this work. The simplest is to hold the previous value constant until new data arrives. This method is illustrated in Figure 2.2b. The other method is to linearly interpolate the missing data points between the last sample before interruption and the first new sample. The linear interpolation is illustrated in Figure 2.2c.

The fourth method takes place in the conversion from the time domain to the frequency domain where the missing samples are considered as variations in the sample time. An illustration of this method is presented in Figure 2.2d. The new equation for calculation of the transform becomes

$$\mathscr{F}_{\mathrm{CZT}}(f(t)) = \tilde{F}(\omega_i) = \sum_{k=0}^{N-2} \Delta_k f_k e^{-j\omega_i t_k}, \qquad \omega_i \in \boldsymbol{\omega} = \{0.10, 0.14, \ldots, 1.98\}. \quad (2.23)$$

where Δ_k denotes the distance to the next sample in number of T_s and kT_s has been replaced by t_k compared to (2.8). This means that when no samples are missing $\Delta_k = 1$ and when samples are missing, $\Delta_k = n_m + 1$, where n_m is the number of missing samples. This method is referred to as the *Variable Sample Time* (VST) method.

There are several sophisticated techniques that, at first glance, could seem suitable for this application but that have not been chosen for further study. These have other constraints when it comes to the amount of missing data or compu-



*(**a**) Discard: all lines with missing samples are ignored. This results in a distortion of the signal in the time domain.*

*(**b**) Hold: new samples are interpolated in the time domain by keeping the previous value until new samples arrive.*

*(**c**) Linear interpolation: new samples are interpolated linearly in the time domain between the last sample before disruption and the first new sample to arrive.*

*(**d**) Variable sample time: missing samples are regarded as variations in the sample time in the conversion to the frequency domain. See Figure 2.1 for comparison.*

Figure 2.2: *The four methods for handling of missing data that are investigated are presented in (a) to (d). The fourth sample is missing in all figures. The solid line is the true signal without missing samples and the dashed line represents the interpolated signal.*

tational effort. For example, Pintelon and Schoukens (1999) present a frequency domain method that treats the missing data as unknown parameters and estimates the model parameters and the missing data simultaneously using a maximum likelihood estimator. In the case where both input and output data are missing, however, Pintelon and Schoukens (1999) show that parameter estimation can become impossible. When the number of missing samples is equal to or greater than the order of the system, the cost function becomes ill-conditioned.

The condition number grows fast with an increasing number of missing samples, and within a few samples, the model becomes non-identifiable. The system of interest in this case is of order six, meaning that there cannot be more than just above six samples missing. This method is therefore not useful as the algorithm must tolerate as much as 64 missing samples.

Another example is found in Wallin (2004), where several methods for parameter estimation with missing data are compared. The *Expectation Maximisation* (EM) algorithm is designed to handle missing data and gives unbiased results. Wallin (2004) compares the original EM algorithm with two modified versions from Isaksson (1993) and Wallin et al. (2000). However, all of these methods are computationally heavy, which makes them time consuming and not suitable for a real-time application.

2.5 Validation

For validation of estimations on simulated data, the goodness of fit is used. The goodness of fit can be calculated using the *Normalised Root Mean Square Error* (NRMSE) as

$$\text{NRMSE}(\hat{\boldsymbol{\theta}}) = \left(1 - \frac{|y - \hat{y}|}{|y - \bar{y}|}\right) \tag{2.24}$$

where $\hat{\boldsymbol{\theta}}$ is the parameter estimate, $\boldsymbol{y}$ is the measured signal, $\hat{\boldsymbol{y}}$ is the simulated signal using $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{u}$, and $\bar{y}$ is the mean of the measured signal. The measured signal $\boldsymbol{y}$ has to be different from the one used to estimate the parameters. The NRMSE is a value between minus infinity and one, where one means perfect fit.

3

Flight Mechanical and Aerodynamic Models

When performing system identification with a grey box model, some information about the system is already known, as described in Chapter 2. In this application, a well known mathematical model for the flight mechanics is available. The aerodynamic properties, however, are more difficult to model and a linear grey box model with unknown parameters is used in this work. The two models, the flight mechanical and the aerodynamic, are presented in this chapter. The assumptions and approximations made in the models are also presented.

3.1 Assumptions and Approximations

In order to simplify the models, some assumptions and approximations are made. For example, according to Larsson (2013), the assumption that the earth is flat and non-rotating is made in most practical applications. The specific approximations made at SAAB are found in Kastman (2004) and presented below.

- Position equations:
  - Flat and non-rotating earth
- Force equations:
  - Flat and non-accelerating earth
  - No time derivative of the mass
- Moment equations:
  - Non-rotating earth
  - No time derivative of the moments of inertia

Note that even though the time derivatives are neglected for the mass and moments of inertia, they are not regarded as constant. The mass and moments of inertia used in the equations change, for example due to the consumption of fuel, but these changes are slow.

3.2 Flight Mechanical Model Equations

The flight mechanical model equations describe the complete motion of the aircraft and consist of twelve equations. The first six are needed to relate the motion experienced on board the aircraft to the absolute motion in relation to a fixed reference frame. The remaining six equations describe the forces and moments that act upon the aircraft and their resulting motions. In order to write the force and moments equations, the relation between the reference frames is needed and this is therefore presented initially.

The equations of motion of an aircraft can be found for example in Roskam (1995), Etkin and Reid (1996) and Nelson (1998) and have been verified against internal documentation and code at SAAB. The equations are derived from Newton's equations of motion and these only hold for an inertial system. The measurements of the aircraft's states, speed and angular velocities, etc., however, are made on the aircraft. For this reason, two coordinate systems are needed, one fixed to the earth (x_f, y_f, z_f), that is considered an intertial system, and one fixed to the aircraft (x, y, z). The earth-fixed system (x_f, y_f, z_f) has the directions north, east and down, respectively. The body-fixed system (x, y, z) is centred at the aircraft's centre of gravity and pointing towards the front, over the right wing and down through the aircraft body respectively, as can be seen in Figure 3.1. The two coordinate systems are related as described in Figure 3.2 by the angles known as the Euler angles, roll ϕ, pitch θ and yaw ψ.

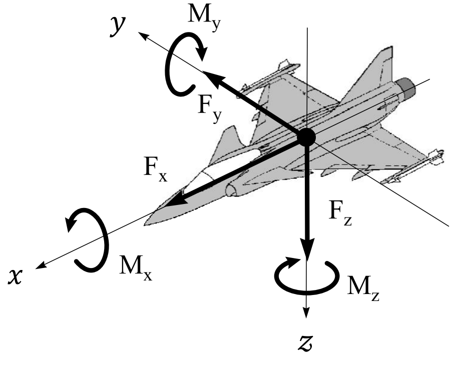
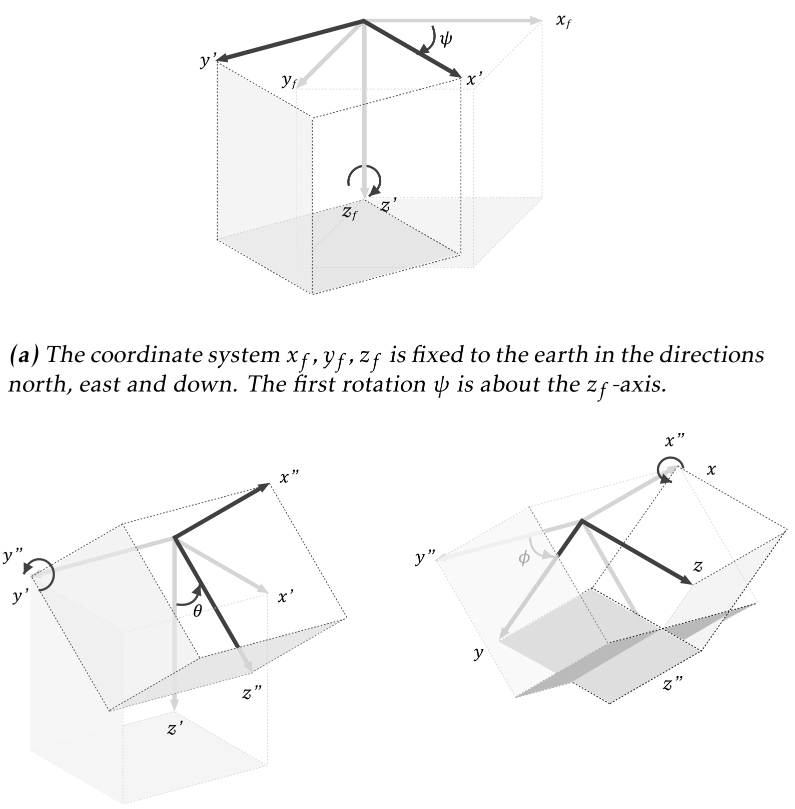


Figure 3.1: The coordinate system fixed to the aircraft together with the forces and moments working on it. The picture is taken from Stavöstrand (2011) with permission from SAAB.



*(**a**) The coordinate system x_f, y_f, z_f is fixed to the earth in the directions north, east and down. The first rotation ψ is about the z_f-axis.*

*(**b**) The second rotation θ is about the y'-axis of the intermediate system x', y', z'.*

*(**c**) The third rotation ϕ is about the x''-axis of the intermediate system x'', y'', z'' and gives the final system x, y, z fixed to the aircraft.*

Figure 3.2: *The three rotations that relate the aircraft's orientation to the earth-fixed coordinate system are given in (a) to (c). They are called the Euler angles. Note that the order of the rotations is important.*

Using the Euler angles, the velocities of the aircraft in the body-fixed system, u, v and w, can be transformed to velocities in the inertial system as

$$\begin{bmatrix} \dot{x}_f \\ \dot{y}_f \\ \dot{z}_f \end{bmatrix} = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi - \mathrm{c}\phi\mathrm{s}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\psi \\ \mathrm{c}\theta\mathrm{s}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi + \mathrm{c}\phi\mathrm{c}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\psi \\ -\mathrm{s}\theta & \mathrm{s}\phi\mathrm{c}\theta & \mathrm{c}\phi\mathrm{c}\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{3.1}$$

where $\mathrm{c}x = \cos(x)$ and $\mathrm{s}x = \sin(x)$.

Similarly, the body-fixed angular velocities p, q and r are transformed into angular velocities in the inertial system as

$$\begin{aligned}\dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= (q\sin\phi + r\cos\phi)\sec\theta\end{aligned} \tag{3.2}$$

where $\sec\theta = 1/\cos\theta$.

The equations in (3.1) and (3.2) are the first six of the twelve equations that fully describe the flight mechanical motion. Newton's second law of forces and moments is used to complete the equations of motion. Since the forces are written for a non-inertial system centred at the body centre of gravity, the force equation can be rewritten using the angular velocity. The total of all forces can then be written as

$$\sum \boldsymbol{F}_f = \tfrac{\mathrm{d}}{\mathrm{d}t}(m\boldsymbol{v}_f) \Leftrightarrow \sum \boldsymbol{F}_B = \tfrac{\mathrm{d}}{\mathrm{d}t}(m\boldsymbol{v}_B) + \boldsymbol{\omega} \times (m\boldsymbol{v}_B) \approx m\dot{\boldsymbol{v}}_B + \boldsymbol{\omega} \times (m\boldsymbol{v}_B) \tag{3.3}$$

where the time derivative of the mass is neglected,

$$\boldsymbol{v}_f = \begin{bmatrix}\dot{x}_f \\ \dot{y}_f \\ \dot{z}_f\end{bmatrix}, \qquad \boldsymbol{\omega} = \begin{bmatrix}p \\ q \\ r\end{bmatrix} \qquad \text{and} \qquad \boldsymbol{v}_B = \begin{bmatrix}u \\ v \\ w\end{bmatrix}. \tag{3.4}$$

The velocity vector $\boldsymbol{v}_f$ is the aircraft's velocity expressed in the inertial, earth-fixed, system while $\boldsymbol{v}_B$ denotes the velocity expressed in the body-fixed system. The angular velocity is always expressed in the body-fixed system and has therefore no subscript. Expressed as the forces in the different directions, (3.3) becomes

$$F_x = m(\dot{u} + qw - rv) \tag{3.5a}$$

$$F_y = m(\dot{v} + ru - pw) \tag{3.5b}$$

$$F_z = m(\dot{w} + pv - qu) \tag{3.5c}$$

in body-fixed coordinates.

Newton's second law of moments is rewritten for the body fixed system in the same way as the force equation and becomes

$$\sum \boldsymbol{M}_f = \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{H}_f \Leftrightarrow \sum \boldsymbol{M}_B = \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{H}_B + \boldsymbol{\omega} \times \boldsymbol{H}_B \approx \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega}) \tag{3.6}$$

where the time derivative of the inertia matrix is neglected,

$$\boldsymbol{\omega} = \begin{bmatrix}p \\ q \\ r\end{bmatrix} \qquad \text{and} \qquad \boldsymbol{H}_B = \boldsymbol{I}\boldsymbol{\omega} = \begin{bmatrix}I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz}\end{bmatrix}\boldsymbol{\omega}. \tag{3.7}$$

Note that when flying in a trimmed condition, the angular rates and total moments acting on the aircraft are zero. The moments of inertia are calculated as

$$\begin{aligned} I_{xx} &= \int_m (y^2 + z^2)\mathrm{d}m, \quad I_{yy} = \int_m (x^2 + z^2)\mathrm{d}m, \quad I_{zz} = \int_m (x^2 + y^2)\mathrm{d}m, \\ I_{xy} &= \int_m (xy)\mathrm{d}m, \qquad I_{yz} = \int_m (yz)\mathrm{d}m, \qquad I_{xz} = \int_m (xz)\mathrm{d}m \end{aligned} \tag{3.8}$$

which are given in body-fixed coordinates. The total moment equation then becomes

$$M_x = \dot{p}I_{xx} - \dot{q}I_{xy} - \dot{r}I_{xz} + qr(I_{zz} - I_{yy}) + (r^2 - q^2)I_{yz} - pqI_{xz} + rpI_{xy} \tag{3.9a}$$

$$M_y = -\dot{p}I_{xy} + \dot{q}I_{yy} - \dot{r}I_{yz} + rp(I_{xx} - I_{zz}) + (p^2 - r^2)I_{xz} - qrI_{xy} + pqI_{yz} \tag{3.9b}$$

$$M_z = -\dot{p}I_{xz} - \dot{q}I_{yz} + \dot{r}I_{zz} + pq(I_{yy} - I_{xx}) + (q^2 - p^2)I_{xy} - rpI_{yz} + prI_{xz}. \tag{3.9c}$$

The equations (3.1), (3.2), (3.5) and (3.9) together give the complete flight mechanical model.

3.3 Aerodynamical Model Equations

The complete flight mechanical model presented in the previous section contains all forces and moments that influence the aircraft. These consist of different contributions from the gravity, the aerodynamics and the engine thrust. The gravitational components are easily calculated using standard theory. The contribution from the engine is calculated using an existing model at SAAB. This model is not of interest for investigation in this work and is therefore considered to be accurate. The remaining contributions come from the aerodynamic model. The aerodynamic model has an established structure with aerodynamic coefficients and unknown parameters. It is these parameters that this work aims to find. The standard aerodynamic model can be found for example in Nelson (1998). The model structure and the parameters of interest will be presented in this section.

3.3.1 Aerodynamic Model Structure

The aerodynamical model used in this work is a linear model of the form

$$y = \phi\theta \tag{3.10}$$

where the outputs

$$y = \begin{bmatrix} C_T & C_C & C_N & C_l & C_m & C_n \end{bmatrix}^T \tag{3.11}$$

are non-dimensional versions of the aerodynamic forces and moments, called aerodynamic coefficients.

In reality, the aerodynamics is non-linear and it is difficult to formulate a complete analytical model of the aerodynamics (Stavöstrand, 2011). Therefore, at SAAB, a combination of a mathematical and graphical model (see Section 2.1) of the aerodynamics is used instead. This model consists of a multidimensional look-up table where the coefficients can be found given the flight condition (height, mach number, etc.), inputs and states of the aircraft. The different points in the table correspond to different conditions in the flight envelope where the behaviour is considered linear between adjacent points and the values between points are found using interpolation. The table therefore has different point density in different regions depending on the region's linear or non-linear properties.

Even though a linear analytical model cannot be formulated for the entire flight envelope, a linearised model for small disturbances around a specific trimmed condition can be found using a Taylor series expansion. All but the first derivatives of the coefficient with respect to the dependent variables are neglected. The estimation of these derivatives is the purpose of the investigated methods. Which dependent variables to include in the model depend on the level of complexity of the model. For example, the cross coupling derivatives are needed only with an asymmetric aircraft configuration. With a symmetric aircraft, a pure longitudinal motion would result in only a longitudinal response. However, if the centre of gravity is displaced toward one wing, there would be a lateral response as well. As an example, an elevator deflection intended to give a pitch moment would result in a roll moment as well if the centre of gravity is displaced towards the wing.

3.3.2 Aerodynamic Model Inputs

The choice of which dependent variables to use as inputs to the model is presented in this section. It is also presented how these are calculated, if they are not existing signals from the aircraft or the surrounding software.

The aerodynamics influencing an aircraft depend on the one hand on the geometry and aerodynamical properties of the aircraft's structure. On the other hand, it depends on the orientation of the aircraft in relation to the incoming wind and the deflection of the aircraft's different control surfaces. These last two are considered as inputs to the aerodynamic model. The notation and sign convention for the aircraft's orientation and control surfaces are presented in Figure 3.3.

Most inputs are measured and/or calculated on board the aircraft, or calculated directly on arrival to the control room by surrounding software and are available as real-time signals. Due to the delta wing configuration of the Gripen fighter it does not have specific ailerons δ_a and elevators δ_e. Instead, there are four elevons (δ_{loe}, δ_{lie}, δ_{rie} and δ_{roe}) which serve as both ailerons and elevators. The elevator deflection is calculated as

$$\delta_e = \frac{1}{4}(\delta_{loe} + \delta_{lie} + \delta_{rie} + \delta_{roe}) \tag{3.12}$$

and the aileron deflection is calculated as

$$\delta_a = \frac{1}{4}(\delta_{loe} + \delta_{lie} - \delta_{rie} - \delta_{roe}). \tag{3.13}$$

The canard deflection is not given as a signal either but calculated by adding the contributions from the two sides as

$$\delta_c = \frac{1}{2}(\delta_{lc} + \delta_{rc}). \tag{3.14}$$

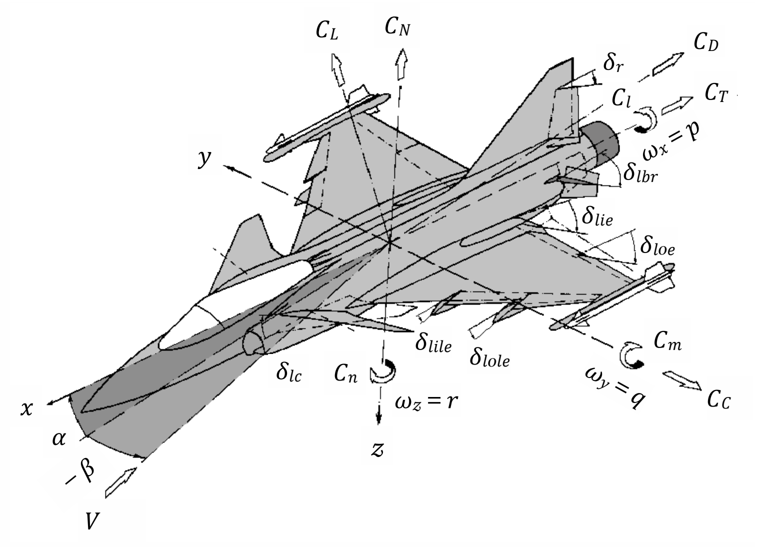


Figure 3.3: *The picture shows the notation and sign convention used in this work and at SAAB. Full names of the parameters can be found in the notation list on page xii. The picture is taken from Stavöstrand (2011) with permission from SAAB.*

In the aerodynamic model, the angular roll, pitch and yaw rates (p, q and r) are not used directly as inputs. Instead, normalised, non-dimensional versions of the angular rates are used. These are called the normalised roll, pitch and yaw rates and are calculated as

$$\hat{p} = \frac{b}{2V}p, \qquad \hat{q} = \frac{c}{2V}q \qquad \text{and} \qquad \hat{r} = \frac{b}{2V}r \tag{3.15}$$

where V is the wind speed, c is the mean aerodynamic chord of the wing and b is the wing span.

The dependent variables have been chosen after discussions with Helena Johansson and Fredrik Ljungberg at the *Aerodynamics section* at SAAB Aeronautics (Johansson and Ljungberg, 2015). The ones that are chosen are the ones that are included in the aerodata model today at SAAB and that are possible to measure. These are the angle of attack α, the side slip angle β, the normalised angular rates $\hat{p}$, $\hat{q}$ and $\hat{r}$, and the canard δ_c, elevator δ_e, aileron δ_a, rudder δ_r and leading edge flap δ_{le} deflections. This gives the total regressor vector

$$\boldsymbol{\phi} = \begin{bmatrix} \alpha & \beta & \hat{p} & \hat{q} & \hat{r} & \delta_c & \delta_e & \delta_a & \delta_r & \delta_{le} \end{bmatrix}. \tag{3.16}$$

The air brake deflection δ_{br} and the landing gear are examples of inputs that also influence the aerodynamics. However, these cannot be measured with today's measurement configuration and are only used in certain manoeuvres. They are therefore not included in the model.

3.3.3 Calculation of the Aerodynamic Coefficients

The aerodynamic coefficients that are given as outputs in (3.11) are not measured but calculated from other signals. To formulate the aerodynamic model, the aerodynamic contributions to the force and moment equations in the previous section must therefore be isolated. By removing the contributions from the gravity and the thrust in the force equation, the aerodynamic force is given as

$$\boldsymbol{F}_a = m\dot{\boldsymbol{v}}_B + \boldsymbol{\omega} \times (m\boldsymbol{v}_B) - \boldsymbol{F}_g - \boldsymbol{F}_e \tag{3.17}$$

where $\boldsymbol{F}_g$ is the gravity force vector and $\boldsymbol{F}_e$ is the thrust force vector. Using standard notation, the aerodynamic forces for each direction become

$$\begin{aligned} F_{a_x} &:= X = m(\dot{u} + qw - rv) + mg\sin\theta - F_{e_x} \\ F_{a_y} &:= Y = m(\dot{v} + ru - pw) - mg\cos\theta\sin\phi - F_{e_y} \\ F_{a_z} &:= Z = m(\dot{w} + pv - qu) - mg\cos\theta\cos\phi - F_{e_z} \end{aligned} \tag{3.18}$$

where $F_{e_y} = 0$ in the engine model at SAAB due to the placement of the engine in the y–symmetry plane. The gravity contribution in (3.18) has been converted to the body-fixed system using Euler angles. The notation at SAAB is slightly different from that in the literature and to be consistent with Stavöstrand (2011)

$$T = -X, \qquad C = -Y, \qquad \text{and} \qquad N = -Z \tag{3.19}$$

are used instead to denote the aerodynamic forces.

When calculating these forces at SAAB, a measurement of the load factors is used. The load factor is all forces except the gravity, divided by the aircraft's weight. It is a unit-less quantity but often said to have the unit g. The load factor has the opposite sign of the force and can be thought of as the weight perceived by the pilot. For example, if the lift is in the negative z–direction, pressing the aircraft upwards, the pilot will feel heavier, as if he is being pressed down into the seat. The load factor will therefore be pointing down and having a positive sign. This gives the relation

$$\boldsymbol{n} = -\frac{\boldsymbol{F}_a + \boldsymbol{F}_e}{mg} \Leftrightarrow \boldsymbol{F}_a = -mg\boldsymbol{n} - \boldsymbol{F}_e \tag{3.20}$$

where $\boldsymbol{n}$ is the load factor vector and g is the gravity constant.

The same reasoning as for the force equation is applied to the moment equation and all contributions except for the aerodynamics are removed. The gravity however, does not contribute to the moment as the body-fixed coordinate system is centred at the aircraft's centre of gravity. Instead, the aerodynamic forces $\boldsymbol{F}_a$ result in a moment as these have their point of attack in the aerodynamic centre, which does not generally coincide with the centre of gravity. By removing this component and the contributions from the thrust, the aerodynamic moment equation becomes

$$\boldsymbol{M}_a = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \dot{\boldsymbol{I}}\boldsymbol{\omega} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{H}_e) - \boldsymbol{M}_e - (\boldsymbol{r}_{ac,cg} \times \boldsymbol{F}_a) \tag{3.21}$$

where $\boldsymbol{H}_e$ is the angular momentum caused by the spinning rotors of the engine,

$\boldsymbol{M}_e$ is the moment contribution from the thrust due to the engine not being in the centre of gravity and $\boldsymbol{r}_{a.c,c.g}$ is the distance from the centre of gravity to the aerodynamic centre. In applications at SAAB, the derivatives of the moments of inertia are neglected as stated in Section 3.1, eliminating the term $\dot{\boldsymbol{I}}\omega$ from (3.21). Note however that the moments of inertia matrix is not constant. Both the inertia matrix and the position of the centre of gravity change depending on the fuel consumption and distribution. The gyroscopic effect from the engine only gives a contribution in the x-direction and is therefore given as $\boldsymbol{H}_e = \left[\, H_e \;\, 0 \;\, 0 \,\right]^T$ and the thrust moment is given as $\boldsymbol{M}_e = \left[\, M_{e_x} \; M_{e_y} \; M_{e_z} \,\right]^T$. Inserting these in (3.21) and introducing the standard notation $\mathcal{L}, \mathcal{M}, \mathcal{N}$ give the moment equations

$$\begin{aligned} M_{a_x} := \mathcal{L} = \dot{p}I_{xx} - \dot{q}I_{xy} - \dot{r}I_{xz} + qr(I_{zz} - I_{yy}) + (r^2 - q^2)I_{yz} - pqI_{xz} + rpI_{xy} - \\ M_{e_x} - \Big((cg_y - ac_y)F_{a_z} - (ac_z - cg_z)F_{a_y}\Big), \quad \text{(3.22a)} \end{aligned}$$

$$\begin{aligned} M_{a_y} := \mathcal{M} = -\dot{p}I_{xy} + \dot{q}I_{yy} - \dot{r}I_{yz} + rp(I_{xx} - I_{zz}) + (p^2 - r^2)I_{xz} - qrI_{xy} + pqI_{yz} + \\ H_e r - M_{e_y} - \Big((ac_z - cg_z)F_{a_x} - (cg_x - ac_x)F_{a_z}\Big) \quad \text{(3.22b)} \end{aligned}$$

and

$$\begin{aligned} M_{a_z} := \mathcal{N} = -\dot{p}I_{xz} - \dot{q}I_{yz} + \dot{r}I_{zz} + pq(I_{yy} - I_{xx}) + (q^2 - p^2)I_{xy} - rpI_{yz} + prI_{xz} - \\ H_e q - M_{e_z} - \Big((cg_x - ac_x)F_{a_y} - (cg_y - ac_y)F_{a_x}\Big). \quad \text{(3.22c)} \end{aligned}$$

At SAAB, the positions of the centre of gravity and the aerodynamic centre are not given in the body-fixed coordinate system but in the main construction coordinate system (the S85 system) (Stavöstrand, 2011). This coordinate system has its centre at the nose of the aircraft and its x_{85}– and y_{85}–axes are pointing in the negative direction of the body-fixed x– and y–axes, respectively. This is the reason for the changed order of $(cg_x - ac_x)$ and $(cg_y - ac_y)$ in (3.22). Note that $M_x \neq \mathcal{L}$ since it is the total of all moments whereas $\mathcal{L}$ is only the aerodynamic moment.

Now that the aerodynamic forces and moments are isolated, the non-dimensional aerodynamic coefficients can be calculated as

$$\begin{aligned} C_T &= \frac{T}{q_a S}, & C_C &= \frac{C}{q_a S}, & C_N &= \frac{N}{q_a S}, \\ C_l &= \frac{\mathcal{L}}{q_a S b}, & C_m &= \frac{\mathcal{M}}{q_a S c}, & C_n &= \frac{\mathcal{N}}{q_a S b} \end{aligned} \tag{3.23}$$

where

$$q_a = \frac{1}{2}\rho V^2 \tag{3.24}$$

is the dynamic pressure, ρ is the air density, V is the wind speed, S is the wing area, c is the mean aerodynamic chord of the wing and b is the wing span. Note the difference in notation between N, which is the normal force, and $\mathcal{N}$, which is the aerodynamic yaw moment.

Given the results from the previous sections, the complete linear aerodynamic model can be formulated as

$$y = \theta^T \phi^T \Leftrightarrow \begin{bmatrix} C_T \\ C_C \\ C_N \\ C_l \\ C_m \\ C_n \end{bmatrix} = \frac{1}{q_a S} \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{b} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{c} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{b} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ \mathcal{L} \\ \mathcal{M} \\ \mathcal{N} \end{bmatrix} =$$

$$\begin{bmatrix} C_{T_0} & C_{T_\alpha} & C_{T_\beta} & 0 & 0 & 0 & C_{T_{\delta_c}} & C_{T_{\delta_e}} & C_{T_{\delta_a}} & C_{T_{\delta_r}} & C_{T_{\delta_{le}}} \\ C_{C_0} & C_{C_\alpha} & C_{C_\beta} & 0 & 0 & 0 & 0 & 0 & C_{C_{\delta_a}} & C_{C_{\delta_r}} & 0 \\ C_{N_0} & C_{N_\alpha} & C_{N_\beta} & 0 & C_{N_{\hat{q}}} & 0 & C_{N_{\delta_c}} & C_{N_{\delta_e}} & 0 & 0 & C_{T_{\delta_{le}}} \\ C_{l_0} & C_{l_\alpha} & C_{l_\beta} & C_{l_{\hat{p}}} & 0 & C_{l_{\hat{r}}} & 0 & 0 & C_{l_{\delta_a}} & C_{l_{\delta_r}} & 0 \\ C_{m_0} & C_{m_\alpha} & C_{m_\beta} & 0 & C_{m_{\hat{q}}} & 0 & C_{m_{\delta_c}} & C_{m_{\delta_e}} & C_{m_{\delta_a}} & 0 & C_{m_{\delta_{le}}} \\ C_{n_0} & C_{n_\alpha} & C_{n_\alpha} & C_{n_{\hat{p}}} & 0 & C_{n_{\hat{r}}} & 0 & 0 & C_{n_{\delta_a}} & C_{n_{\delta_r}} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \\ \beta \\ \hat{p} \\ \hat{q} \\ \hat{r} \\ \delta_c \\ \delta_e \\ \delta_a \\ \delta_r \\ \delta_{le} \end{bmatrix} \tag{3.25}$$

where the matrices on the right-hand side have been transposed only to make the expressions fit on the page. The model includes all cross coupling derivatives to make it as complete as possible. This allows for it to also handle asymmetric aircraft configurations. The model is analogous to the one presented in Stavöstrand (2011) except that the derivatives $C_{l_{\hat{q}}}, C_{m_{\hat{p}}}$ and $C_{n_{\hat{q}}}$ have been removed. These are not part of the aerodata model according to Helena Johansson and Fredrik Ljungberg (Johansson and Ljungberg, 2015).

As explained previously, the model is only valid for small disturbances from a trimmed condition. This means that the output and regressor vectors in reality become

$$y = \begin{bmatrix} \Delta C_T & \Delta C_C & \Delta C_N & \Delta C_l & \Delta C_m & \Delta C_n \end{bmatrix}^T \tag{3.26}$$

and

$$\phi = \begin{bmatrix} \Delta\alpha & \Delta\beta & \Delta\hat{p} & \Delta\hat{q} & \Delta\hat{r} & \Delta\delta_c & \Delta\delta_e & \Delta\delta_a & \Delta\delta_r & \Delta\delta_{le} \end{bmatrix} \tag{3.27}$$

respectively, where Δ denotes a deviation from a trimmed condition. Since the first element of the regressor vector in (3.25) is constant, this and the first column of the parameter matrix in (3.25) are not part of the estimation model.

4 Two Degrees of Freedom Simulation Model

The implementation of a simulation model is described in this chapter. The aircraft model is implemented in Simulink as a tool to analyse the two estimation methods described in Section 2.3 and the four methods for handling of missing data described in Section 2.4.2. The simulation model gives an environment where all coefficients are known exactly and process and measurement noise levels can be adjusted. The simulated flight data can also be modified and missing data can be introduced. This makes it possible to compare the different algorithms and methods for handling of missing data since the true model is known. Since the data quality can be modified, the comparisons can be made for a wider range of cases than would have been the case if only real flight test data was studied. This allows for a more extensive analysis even though a limited amount of real datasets are available for study.

The implemented model is a simplification based on an F-16 fighter since the numerical values for this aircraft are public. The conclusions drawn from the simulation results are, however, assumed to be applicable for the case of Gripen as well. The simplified model together with the made approximations are presented in Section 4.1 and the final simulation model with numerical values from the F-16 is presented in Section 4.2.

4.1 The Short-Period Approximation

The implemented simulation model is a simplified version of a complete flight mechanical model. The method used for this simplification is called the short-period approximation and is described in this section.

In order to generate simulated data in Simulink, the required model structure is a state-space representation such as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{w} \tag{4.1a}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v} \tag{4.1b}$$

where $\boldsymbol{x}$ denotes the states, $\boldsymbol{u}$ denotes the inputs, $\boldsymbol{y}$ denotes the measured outputs, $\boldsymbol{w}$ denotes the process disturbance vector and $\boldsymbol{v}$ denotes the measurement noise vector. To obtain such a model, the flight mechanical equations are used. The complete rigid-body aircraft model has six *degrees of freedom* (DOF) and nine states. In order to get a lower order model that is more easily handled, only longitudinal motion is considered, (x-, z- and pitch-motion). Then only the longitudinal equations, namely (3.5a), (3.5c) and (3.9b), are used.

Having only a longitudinal motion, these equations reduce to

$$F_x = m(\dot{u} + qw), \tag{4.2a}$$

$$F_z = m(\dot{w} - qu) \tag{4.2b}$$

and

$$M_y = \dot{q}I_{yy} \tag{4.2c}$$

where all parameters describing the lateral motion have been set to zero. This representation can be simplified further by limiting the study to the short-period mode, in accordance with Nelson (1998). For a more detailed description of the short-period approximation, Etkin and Reid (1996) or Nelson (1998) are recommended.

The short-period is a highly damped oscillatory motion in angle of attack with short period time. The short-period approximation is obtained by assuming that the motion takes place at constant altitude and with constant speed V. The motion is further assumed to consist of small deviations in angle of attack from a trimmed flight condition parallel to the earth's surface, that is $\theta_0 = 0$. The equations are therefore linearised around this trimmed state. The changes in angle of attack are considered so small that the force in the x-direction and variation of the speed u are neglected entirely. The small angle assumption also allows for the approximation $\Delta w = u_0 \sin \Delta\alpha \approx u_0 \Delta\alpha$. Applying all these assumptions to (4.2) gives

$$\Delta F_z = m u_0(\Delta\dot{\alpha} - \Delta q) \tag{4.3a}$$

and

$$\Delta M_y = \Delta\dot{q} I_{yy} \tag{4.3b}$$

where u_0 is the trimmed speed in the x-direction and Δ denotes the deviation from the trimmed condition. Note that $u_0 = V$ as a result of the approximations.

The force and moment in (4.3) include all forces and moments acting on the aircraft. The thrust is assumed to be constant and the variation in pitch angle $\Delta\theta$ is assumed small enough to consider the gravitational contribution as constant. The result is that the only forces and moments to be considered are the aerodynamic contributions. The conversions from force and moment into the dimensionless aerodynamic coefficients are

$$\Delta F_z = \Delta F_{a_z} = -\Delta N = -\Delta C_N q_a S \tag{4.4a}$$

and

$$\Delta M_y = \Delta M_{a_y} = \Delta \mathcal{M} = \Delta C_m q_a S c \tag{4.4b}$$

where $q_a = \frac{1}{2}\rho V^2$ is the dynamic pressure.

The aerodynamic coefficients are then expressed as linear combinations of the angles, accelerations and control surface deflection angles that are part of the simplified model. By removing the lateral elements in (3.25) and linearising, the simplified aerodynamic coefficients become

$$\Delta C_N = C_{N_\alpha}\Delta\alpha + C_{N_{\hat{q}}}\Delta\hat{q} + C_{N_{\delta_e}}\Delta\delta_e \tag{4.5a}$$

and

$$\Delta C_m = C_{m_\alpha}\Delta\alpha + C_{m_{\hat{q}}}\Delta\hat{q} + C_{m_{\delta_e}}\Delta\delta_e \tag{4.5b}$$

where $\hat{q} = \frac{qc}{2V}$ and the aerodynamic derivatives depend on the trimmed flight condition.

By combining (4.3), (4.4) and (4.5), the state-space formulation becomes

$$\Delta\dot{\alpha} = -\frac{Sq_a}{mu_0}\left(C_{N_\alpha}\Delta\alpha + \left(C_{N_{\hat{q}}}\frac{c}{2V} - \frac{mu_0}{Sq_a}\right)\Delta q + C_{N_{\delta_e}}\Delta\delta_e\right) \tag{4.6a}$$

and

$$\Delta\dot{q} = \frac{Scq_a}{I_{yy}}\left(C_{m_\alpha}\Delta\alpha + C_{m_{\hat{q}}}\frac{c}{2V}\Delta q + C_{m_{\delta_e}}\Delta\delta_e\right) \tag{4.6b}$$

which has the desired structure of (4.1) but with only two DOF. To summarise, (4.6) is the state-space representation used in Simulink to drive the simulation while (4.5) is the linear model whose parameters are estimated.

The desired input to the system is a pulse in order to maximise the frequency content. However, a perfect pulse cannot be obtained in a fighter aircraft. This is partly due to the dynamics between the pilot command δ_{se} and the control surface δ_e and partly due to the presence of an FCS, constantly trying to stabilise the aircraft. In order to make the simulated data more realistic, these aspects are added to the Simulink model in addition to the state-space model, as can be seen in Figure 4.1. The transfer function of the servo is assumed to be

$$G_{\text{servo}}(s) = \frac{0.0495}{0.0495s + 1} \tag{4.7}$$

which can be found in Stevens and Lewis (2003). For simplicity, the closed loop is implemented only as a feedback in α with gain 1.

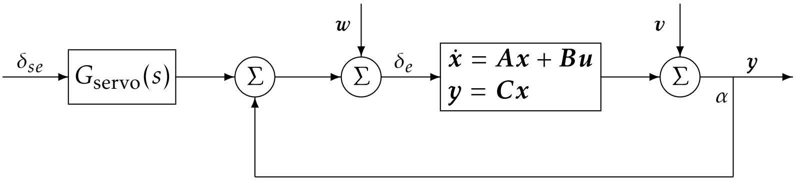


Figure 4.1: *The implementation of the two DOF-aircraft model in Simulink follows the structure above with a servo, process and measurement disturbances and a feedback in α.*

4.2 Numerical Simulation Model

For the generation of simulated data, a model of the F-16 fighter is used. The state-space model and numerical values for this aircraft are taken from Morelli (1999). The model has the desired structure of (4.1) with the states, input and outputs

$$\boldsymbol{x} = \begin{bmatrix} \alpha \\ q \end{bmatrix}, \qquad \boldsymbol{u} = \begin{bmatrix} \delta_e \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{y} = \begin{bmatrix} \alpha \\ q \end{bmatrix} \tag{4.8}$$

and the numerical values

$$\boldsymbol{A} = \begin{bmatrix} -0.6 & 0.95 \\ -4.3 & -1.2 \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} -0.115 \\ -5.157 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.9}$$

The numerical values of the aerodynamic derivatives can be calculated as

$$\begin{aligned} C_{N_\alpha} &= -\frac{A_{11} m u_0}{q_a S}, & C_{N_{\hat{q}}} &= \frac{2 V m u_0}{S c q_a}(1 - A_{12}), & C_{N_{\delta_e}} &= \frac{-B_1 m u_0}{S q_a}, \\ C_{m_\alpha} &= \frac{A_{21} I_{yy}}{S c q_a}, & C_{m_{\hat{q}}} &= \frac{A_{22} 2 V I_{yy}}{S c^2 q_a} \quad \text{and} & C_{m_{\delta_e}} &= \frac{B_2 I_{yy}}{S c q_a} \end{aligned} \tag{4.10}$$

where A_{ij} and B_i are the elements of the A and B matrices. These calculated values are presented in Table 4.1.

According to Roskam (1995), the short-period approximation is only valid for inherently stable airplanes. The F-16 is an unstable aircraft in its conventional configuration (Frawley and Thorn, 1996), and the approximation would therefore not be valid. However, the particular F-16 from Morelli (1999) used in this application has a forward centre of gravity position ($0.2c$) which makes it inherently stable and the short-period approximation can therefore be used.

The numerical values of all remaining parameters used in the simulation model are presented in Table 4.2. The parameter values associated with the specific trimmed condition (height, mach number and angle of attack) are taken from

Table 4.1: *The calculated numerical values for the aerodynamic derivatives.*

Parameter	Calculated value	Unit
C_{N_α}	3.6268	[-]
$C_{N_{\hat{q}}}$	21.2876	[-]
$C_{N_{\delta_e}}$	0.6951	[-]
C_{m_α}	-0.5046	[-]
$C_{m_{\hat{q}}}$	-9.9176	[-]
$C_{m_{\delta_e}}$	-0.6051	[-]

Morelli (1999) and the parameters associated with the aircraft (mass, wing area, etc.) are taken from Stevens and Lewis (2003). The atmospheric parameters (temperature, density and lapse rate) follow the *International Standard Atmosphere* (ISA) and can be found in ISO (2533:1975). The remaining parameters are conventional constants or derived using the expressions

$$T = T_0 + \lambda h - h_0, \tag{4.11}$$

$$\rho = \rho_0 \left(\frac{T}{T_0} \right)^{-(1+g/(R\lambda))}, \tag{4.12}$$

$$a = \sqrt{\gamma R T} \tag{4.13}$$

and

$$M = \frac{V}{a} \tag{4.14}$$

from Nelson (1998).

The numerical model presented above is used to generate simulated flight test data. However, in order to fully investigate Larsson's algorithm, data to use as IVs is needed as well. Therefore, a second model is run in parallel with the first to simulate a parallel simulation. The numerical values for the parameters of the second model are modified with 5% in comparison to those in the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices of (4.9) and it lacks process and measurement noise. This way, the true case where the simulation model is similar to, but not the same as, the aircraft, is simulated.

Table 4.2: *The numerical values used in the two DOF-longitudinal aircraft model of the F-16.*

Parameter	Value	Unit	Description
T_s	1/60	[s]	Sampling time
h	3048	[m]	Altitude of flight
h_0	0	[m]	Reference altitude
T	268.348	[K]	Temperature at altitude of flight
T_0	288.16	[K]	Temperature at reference altitude
ρ	0.9045	[kg/m^3]	Air density at altitude of flight
ρ_0	1.225	[kg/m^3]	Air density at reference altitude
α_0	7	[deg]	Trimmed angle of attack
R	287	[m^2/Ks^2]	Gas constant of air
γ	1.4	[-]	Ratio of specific heats
$\mathrm{d}T/\mathrm{d}h$, λ	-0.0065	[K/m]	Temperature gradient (Lapse rate)
g	9.81	[m/s^2]	Gravity constant
M	0.37	[-]	Mach number
a	328.36	[m/s]	Speed of sound at temperature of flight
V	121.5	[m/s]	Airspeed
S	28.0	[m^2]	Wing area
c	3.45	[m]	Mean aerodynamic chord
m	9 300	[kg]	Mass
I_{yy}	75 674	[kgm^2]	Moment of inertia about the y axis

5 Requirements, Constraints and Evaluation Conditions

In this chapter, the details regarding the implementation constraints posed by the surrounding software and the requirements of the final algorithm are presented. The different alternatives and conditions for the evaluation are also presented. Firstly, the ideal functionality of the algorithm is presented in order to formulate the requirements for the algorithm. This presentation also includes a description of the visualisation of the results used throughout this work. Secondly, the surrounding software at SAAB is described to give an understanding of the framework in which the algorithm is working. This framework has limitations and puts constraints on the implementation and performance of the algorithm. Lastly, the data used in this work is described. This gives an understanding of the evaluation conditions when analysing the different methods and the limitations given the available data.

5.1 Ideal Real-Time Parameter Estimation Algorithm

For the real-time parameter estimation algorithm to be fully functional and usable in flight tests at SAAB, some basic elements are needed. In this section, the functionality requirements of the ideal algorithm are presented.

The purpose of the real-time parameter estimation algorithm is to estimate aerodynamic derivatives in real time and compare these to the aerodynamic derivatives of the existing aerodynamic model. A visual comparison of the estimated parameter and the model parameter is needed and it must be good enough to draw conclusions upon. The visualisation must therefore include the quality of the data used for the estimation and the quality of the estimated parameter.

Figure 5.1 shows an example of what the ideal algorithm might output. For sim-

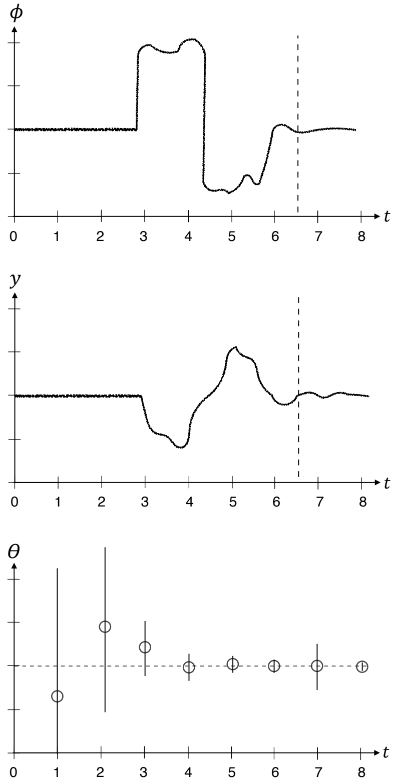


***Figure 5.1:** A theoretical example of what the output from the real-time estimation program might look like. The three graphs show the regressor, the output and the estimates, respectively. The value of an estimate is indicated by a circle and its quality is indicated by two times its estimated standard deviation, visualised by a vertical line. The current model parameter is indicated by a horizontal dashed line for comparison. The excitation and telemetry disruptions, which indicate the quality of the data, can be seen in the plots of the output and regressor. Disruptions are visualised by a vertical dashed line.*

plicity, the example has one regressor ϕ, one output y and one parameter θ. The regressor is a state or input, the output is a coefficient and the parameter is the aerodynamic derivative with respect to the regressor. The time progress of the signals is shown to be able to study the manoeuvre and see when different modes are excited. In this example, the excitation starts just before three seconds. A disruption in the data stream is indicated in the graph, for example by a dashed vertical line. In this example, only the presence of a disruption is indicated and not its length. This is the presentation used in the real-time presentation system today to indicate data loss. The estimation is performed at regular intervals, in this example every second. The estimate is indicated by a circle. The accuracy of the estimate is indicated by two times its estimated standard deviation, calculated using (2.15) and visualised by a vertical line. This accuracy measure is chosen in order to be consistent with Larsson (2013) and Morelli (1999). The dashed horizontal line indicates the current value of the parameter in the aerodynamic model. Since more and more data is used in the estimator it is expected to converge and give more accurate values. Before excitation, there is little information in the data. Ideally, this is visible in the variation of the estimates and in their standard deviations. The true parameter should, however, be included in the interval. After excitation, the standard deviation should become smaller and the estimate should converge towards the true value. In this example, the true value appears to be the same as the model value as the estimates converge towards the dashed line. Missing data could also be visible in the estimation by giving a higher standard deviation.

5.2 Surrounding Software at SAAB

The final parameter identification algorithm is not stand alone but integrated with existing software for real-time monitoring at SAAB. The algorithm is implemented in a Matlab shell called StellaRT that allows for communication with the real-time data server. This server also communicates with the program for parallel simulation that in turn uses the complete aircraft model. All this surrounding software puts constraints on the parameter identification algorithm and affects the implementation choices. To understand these constraints, the software is presented below.

5.2.1 VuSoftNT

The real-time presentation system at SAAB is called VuSoftNT and is described in Andersson et al. (2002). It decodes the telemetry data streams and communicates with other programs such as ROMAC and StellaRT, which are presented below. All telemetry data and simulated data are transferred to VuSoftNT for presentation on screen. VuSoftNT is also capable of real-time calculations. Some signals are calculated within VuSoftNT and available just as any telemetry signal.

5.2.2 ARES

The complete SAAB aircraft model is called *Aircraft Rigid body Engineering Simulation* (ARES). It is a six DOF, non-linear state-space model that contains all sub-models such as the aerodynamic model (Andersson et al., 2002).

5.2.3 ROMAC

As described in Andersson et al. (2002), the internal SAAB tool for real-time parallel simulation is called *Real-time On-line Model and Aerodata Control* (ROMAC). The true pilot commands are sent to ROMAC via VuSoftNT, and ROMAC uses ARES to make a parallel simulation with the true pilot commands. The outputs from this simulation are sent back to VuSoftNT where they are presented as any real-time data. The outputs have a delay of about 0.5 seconds but the resulting signals still have the correct timestamp.

The model contains filters that cannot handle telemetric interruptions. This is handled by ROMAC in two ways, either by restarting the filter or by linear interpolation. Except for in the filter algorithms, however, there are no procedures for handling missing data.

ROMAC also allows for analysis of aerodynamic coefficients. This is done by the two scripts *arcoft* and *WT*. Arcoft calculates the aerodynamic coefficients by doing the calculations described in Chapter 3 while WT performs a search in the look-up tables of the aerodynamic database. When calling Arcoft or WT, all parameters are low pass filtered and a differentiation filter is used in the calculation of the coefficients. These filtrations are found to result in a 35 sample delay of the coefficients with respect to the other signals. Each signal has a corresponding time vector where each sample is associated with a time. The aerodynamic coefficients are delayed also with respect to their own time vector, meaning that the samples also have the wrong timestamp.

In the current release of ROMAC, not all internal parameters are returned to VuSoftNT and thereby available in StellaRT. Additional parameters cannot easily be added to VuSoftNT since ROMAC has reached the limit for how many parameters that can be returned.

5.2.4 StellaRT

Stella Real-Time (StellaRT) is a Matlab toolbox developed at SAAB for performing customised real-time evaluation, and is documented in SAAB AB (2012). StellaRT is a shell that contains a *Graphical User Interface* (GUI). StellaRT connects to the VuSoftNT server and loads the telemetric data and ROMAC data into the Matlab workspace as it arrives. By adding personal scripts to this framework, users can create customised real-time evaluation programs. StellaRT also has a simulation mode where recorded telemetry data is replayed instead of getting data from VuSoftNT.

5.3 Description of the Data

In this section, the different types of data used in this work are presented along with their purpose and constraints. The different types of used data are analytical data, simulated data, post-processed flight test data and recorded real-time flight test data.

5.3.1 Created Data

For some analysis, an analytical signal is used since such a signal is clean from noise, can be designed to have specific properties and can be differentiated analytically. An analytical signal is used as a first stage analysis of parts of the algorithm.

Simulated data is used to determine general properties of the algorithm by performing *Monte Carlo* (MC) simulations. When doing a MC simulation, the means of the estimates and the means of their estimated standard deviations are presented along with the standard deviations over all simulations. This allows to evaluate the bias and variance properties of the estimator.

The simulated data is generated by the two DOF simulation model described in Chapter 4. The simulated data is used as a complete set or with randomly introduced missing data points. The model also allows for varying noise levels, which is required in order to study the influence of the instrumental variables.

5.3.2 Flight Test Data

Two types of flight test data are used in this work. These are post-processed flight test data and recorded real-time flight test data. All flight test data and numerical values in this work have been distorted for security reasons.

The post-processed flight test data is used to evaluate the algorithm under ideal conditions. This data is the recorded data taken from the aircraft after flight and processed using a software called Bas39. This means that the data is complete, with no disruptions. All signals are synchronised to a common frequency of 60 Hz and the data is processed to remove outliers. Some signals are also low pass filtered. In the post-processed data, all delays due to filters are compensated for. Since the data only includes flight data and no simulated data, no analysis using IVs is performed using post-processed flight test data in this work, even though simulated data could be created afterwards.

The recorded real-time data is used to give an example of what results to expect when running the algorithm in real time. The data is recorded during flight test using a software called Nyreg and contains all telemetry disruptions and simulated signals. In the recording, all signals are synchronised to a common frequency of 60 Hz. Other than that, the data is not processed and no delays are compensated for. This means that there is a delay difference between the coefficients of the post-processed data and the recorded real-time data. This delay can be seen in Figure 5.2.

For studies using flight test data, only one data set was available. This data set comes from a roll manoeuvre, and the roll moment coefficient C_l is therefore mostly studied for flight test data. The states and outputs for this data set are presented in Figure 5.3. As can be seen, the manoeuvre starts from a trimmed condition and the roll command arrives after about 13 seconds.

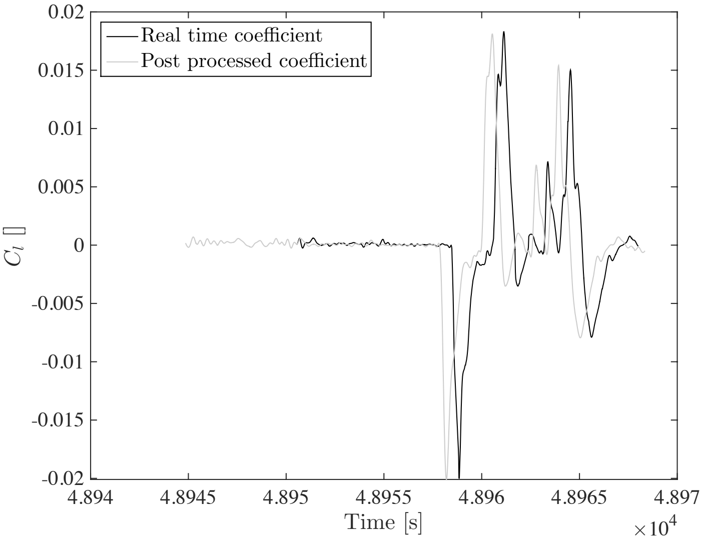


Figure 5.2: *Illustration of the delay of the real-time coefficient C_l compared to the coefficient from the post-processed data.*

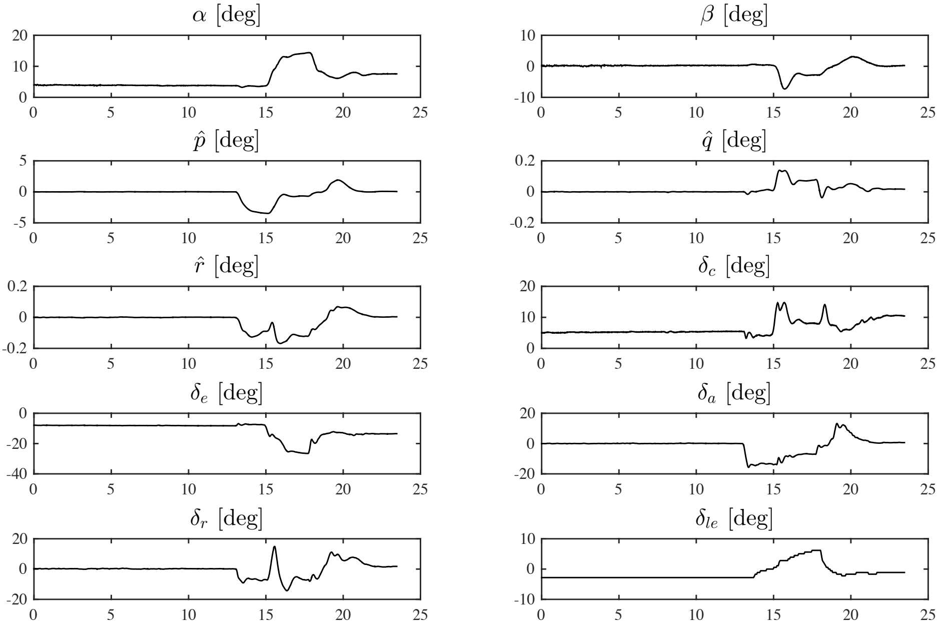


Figure 5.3: *Flight test data for a roll manoeuvre from trimmed condition. The states and inputs shown are those included in the aerodynamic model. The time scale is given in seconds.*

6

Evaluation of Larsson's and Morelli's Algorithms

In chapters 2 and 3, the tools and models used in this work are presented in a general way and the adaptions made for this particular application are described. In this chapter, the results from these two chapters are tied together and the challenges and alternatives in the implementation of the real-time algorithm are presented. The final choice among these implementation alternatives depends on their performance and on their suitability from an implementation point of view. The implementation alternatives and their properties are presented initially. Then, the results using the different implementation alternatives for Larsson's and Morelli's algorithms are presented. The presentation of the results follow the description of the example in Section 5.1 where the ideal estimation algorithm is presented.

6.1 Formulating the Linear Regression

In this section, the different alternatives for formulating and implementing the linear regression are presented along with their challenges and constraints. There are several alternatives when using the aerodynamic model of (3.25) to formulate the complex regression of (2.10). Additionally, the aerodynamic coefficients that constitute the output of the model can be calculated in different ways and one of these must be chosen.

The linear regression is formulated by using the CZT to transform the expression in (3.25). Since not all coefficients have the same inputs and the CLS is implemented to handle parameter vectors, not matrices, one regression is formulated for each coefficient.

6.1.1 Choosing Outputs

The force coefficients can be calculated using (3.20) and (3.23) either in the Arcoft script or in the estimation algorithm. When calculated in the Arcoft script, the coefficients get a delay of 35 samples. Besides the delay, the calculated coefficients are equivalent and calculation in the algorithm is preferred since no delay compensation is needed in that case.

The moment coefficients can similarly be calculated either in the Arcoft script or in the estimation algorithm by combining (3.21) and (3.23). The final equation becomes

$$\boldsymbol{C}_M = \frac{\boldsymbol{I}\dot{\boldsymbol{\omega}} + \dot{\boldsymbol{I}}\boldsymbol{\omega} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{H}_e) - \boldsymbol{M}_e - (\boldsymbol{r}_{ac,cg} \times \boldsymbol{F}_a)}{q_a S l_{\text{ref}}} \tag{6.1}$$

where $\boldsymbol{C}_M$ is a vector containing the moment coefficients (C_l, C_m, C_n) and l_{ref} denotes the reference length, either b or c depending on the direction of the moment. As described in Chapter 3, the term $\dot{\boldsymbol{I}}\boldsymbol{\omega}$ is normally neglected at SAAB and the transform becomes

$$\tilde{\boldsymbol{C}}_M = \mathscr{F}_{\text{CZT}}\left(\frac{\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{H}_e) - \boldsymbol{M}_e - (\boldsymbol{r}_{ac,cg} \times \boldsymbol{F}_a)}{q_a S l_{\text{ref}}}\right) \tag{6.2}$$

where $\mathscr{F}_{\text{CZT}}$ denotes the CZT. Due to the presence of a differentiation in the calculation of the moment coefficients, there are several alternatives for how to calculate them using different approximations.

Calculation of the Moment Coefficients within the Algorithm

In order to use either of Morelli's or Larsson's methods, described in Section 2.3.2 and Section 2.3.3, to transform the derivative, the differentiated term $\dot{\omega}$ must be isolated. For simplicity, the notation $\zeta_{\text{num}} = \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{H}_e) - \boldsymbol{M}_e - (\boldsymbol{r}_{ac,cg} \times \boldsymbol{F}_a)$, for the rest of the numerator, and $\zeta_{\text{denom}} = q_a S l_{ref}$, for the denominator, is introduced. Since neither ζ_{num}, ζ_{denom} or $\boldsymbol{I}$ are constant, neither can be isolated directly. The isolation can be done in several ways and these are presented below.

Approximating the inertia and dynamic pressure as constant: The simplest solution would be to neglect the variations of the inertia and dynamic pressure and put these as constant. The coefficients would then become

$$\tilde{\boldsymbol{C}}_M = \mathscr{F}_{\text{CZT}}\left(\frac{\boldsymbol{I}_0\dot{\boldsymbol{\omega}}}{q_{a0} S l_{\text{ref}}}\right) + \mathscr{F}_{\text{CZT}}\left(\frac{\boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{H}_e) - \boldsymbol{M}_e - (\boldsymbol{r}_{ac,cg} \times \boldsymbol{F}_a)}{q_{a0} S l_{\text{ref}}}\right) \tag{6.3}$$

where $\boldsymbol{I}_0$ and q_{a0} denote the initial values of the inertia and dynamic pressure, respectively. Whether this approximation is valid or not depends on how much the variations of the inertia and dynamic pressure influence the coefficients.

Convolution: Another alternative is to transform all including parts separately and convolute them together. This gives the transformed coefficients

$$\tilde{\boldsymbol{C}}_M = (\mathscr{F}_{\text{CZT}}(\boldsymbol{I}) * \mathscr{F}_{\text{CZT}}(\dot{\boldsymbol{\omega}}) + \mathscr{F}_{\text{CZT}}(\zeta_{\text{num}})) * \mathscr{F}_{\text{CZT}}(\frac{1}{\zeta_{\text{denom}}}) \tag{6.4}$$

where

$$\mathscr{F}_{\mathrm{CZT}}(\boldsymbol{I}) * \mathscr{F}_{\mathrm{CZT}}(\dot{\boldsymbol{\omega}}) = \mathscr{F}_{\mathrm{CZT}}(\boldsymbol{I}) * \left(j\boldsymbol{\omega}_f \mathscr{F}_{\mathrm{CZT}}(\boldsymbol{\omega}) + \frac{1}{T_s} \left(\boldsymbol{\omega}_{t_1} e^{-j\boldsymbol{\omega}_f t_1} - \boldsymbol{\omega}_0 \right) \right). \tag{6.5}$$

Confusion between the angular rates vector and the frequency vector has been avoided by adding the subscript f to the frequency vector. The vector $\boldsymbol{\omega}_0$ denotes the angular rates at time 0 and $\boldsymbol{\omega}_{t_1}$ those at the evaluation time. Since the transform is finite, circular convolution should be used. However, this method is not preferable from an implementation point of view due to its complexity and a simpler solution is preferred.

Modified regressors: One simpler solution is to move the denominator to the right hand side of the regression and create new regressors as

$$\boldsymbol{\phi}' = \begin{bmatrix} \alpha & \beta & \hat{p} & \hat{q} & \hat{r} & \delta_c & \delta_e & \delta_a & \delta_r & \delta_{le} \end{bmatrix} \cdot (q_a S l_{\mathrm{ref}}). \tag{6.6}$$

However, this still leaves the convolution $\mathscr{F}_{\mathrm{CZT}}(\boldsymbol{I}) * \mathscr{F}_{\mathrm{CZT}}(\dot{\omega})$. By reintroducing the term $\dot{\boldsymbol{I}}\boldsymbol{\omega}$, which is usually neglected, this convolution can be avoided as well since

$$\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{I}\boldsymbol{\omega}) = \boldsymbol{I}\dot{\boldsymbol{\omega}} + \dot{\boldsymbol{I}}\boldsymbol{\omega} \tag{6.7}$$

and the differentiation becomes

$$\mathscr{F}_{\mathrm{CZT}}(\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{I}\boldsymbol{\omega})) = j\boldsymbol{\omega}_f \mathscr{F}_{\mathrm{CZT}}(\boldsymbol{I}\boldsymbol{\omega}) + \frac{1}{T_s} \left((\boldsymbol{I}\boldsymbol{\omega})_{t_1} e^{-j\boldsymbol{\omega}_f t_1} - (\boldsymbol{I}\boldsymbol{\omega})_0 \right), \tag{6.8}$$

still with the modified regressors. In this case, $\boldsymbol{I}\boldsymbol{\omega}$ is considered as one signal.

Calculation of the Moment Coefficients within Arcoft

The last alternative is to use the already available coefficients calculated in Arcoft and transform these without performing any differentiation within the algorithm. In that case, a compensation for the delay is needed in the implementation. As mentioned in Section 5.2.3, the coefficients are calculated using a differentiation filter. The angular accelerations $\dot{p}$, $\dot{q}$ and $\dot{r}$ are calculated using a 10th order asymmetric *Finite Impulse Response* (FIR) filter. Whether this is a good alternative or not depends on the quality of this filter compared to the quality of the transform differentiation.

Another aspect to take into account in the implementation is the limitations of the surrounding software. As explained in Section 5.2.3, not all internal parameters are returned to VuSoftNT and thereby available in StellaRT. The parameters that are not available in today's release include $\boldsymbol{I}$ and $\boldsymbol{F}_a$. Without these, none of the alternatives where the moment coefficients are calculated within the algorithm can be used with today's version of ROMAC.

6.2 Sensitivity Analysis for Changes in the Dynamic Pressure and Inertia Matrix

As described in Section 6.1, the simplest solution for transforming the moment coefficients would be to neglect the variations in the dynamic pressure and moments of inertia. To be able to make this approximation, the resulting error in the coefficients must be significantly smaller than the existing margin of error in the coefficients at SAAB. This margin is in the order of 10%. The sensitivity analysis is performed on post-processed flight test data.

The approximation is done by assuming that the dynamic pressure and inertia matrix are constant with the same values as for the trimmed condition. Since the model is valid for small disturbances from a trimmed condition, the variations of the dynamic pressure and moments of inertia might be small and the approximation might be valid.

For this analysis, the real data set from the roll manoeuvre is studied. The variations

$$\delta = 100 \frac{|I_{\max} - I_{\min}|}{|I|_{\min}} \tag{6.9}$$

of the different signals are calculated in percent. The moments of inertia are fairly constant during this particular manoeuvre and I_{xx}, I_{yy} and I_{zz} vary with 2.8%, 0.6% and 0.9% respectively. The products of inertia vary more, 160 %, 2.2 % and 266 % for I_{xy}, I_{xz} and I_{yz} respectively. These do not, however, contribute to a great extent to the total moments. As defined in (3.9), the moments are calculated as

$$\begin{aligned} M_x &= \dot{p}I_{xx} - \dot{q}I_{xy} - \dot{r}I_{xz} + qr(I_{zz} - I_{yy}) + (r^2 - q^2)I_{yz} - pqI_{xz} + rpI_{xy} \\ M_y &= -\dot{p}I_{xy} + \dot{q}I_{yy} - \dot{r}I_{yz} + rp(I_{xx} - I_{zz}) + (p^2 - r^2)I_{xz} - qrI_{xy} + pqI_{yz} \\ M_z &= -\dot{p}I_{xz} - \dot{q}I_{yz} + \dot{r}I_{zz} + pq(I_{yy} - I_{xx}) + (q^2 - p^2)I_{xy} - rpI_{yz} + prI_{xz}. \end{aligned}$$

The contribution from the first three terms to the total roll moment is presented in Figure 6.1. The main contribution comes from the roll acceleration and I_{xx}. The contribution to the pitch moment looks similar to Figure 6.1, where the pitch acceleration and I_{yy} are the main contributors. The yaw moment, however, looks different. It can be seen in Figure 6.2 that in addition to the yaw acceleration and I_{zz}, the roll acceleration and I_{xz} give a large contribution.

In order to study the worst-case scenario, the maximum and minimum values of the moments are calculated in the same way as

$$\begin{aligned} M_{x_{\max}} = \bar{\dot{p}}I_{xx_{\max}} - \bar{\dot{q}}I_{xy_{\min}} - \bar{\dot{r}}I_{xz_{\min}} + \bar{q}\bar{r}(I_{zz_{\min}} - I_{yy_{\max}}) + (\bar{r}^2 - \bar{q}^2)I_{yz_{\min}} \\ - \bar{p}\bar{q}I_{xz_{\max}} + \bar{r}\bar{p}I_{xy_{\max}} \end{aligned} \tag{6.10}$$

where the bar denotes the mean value and max or min have been chosen depending on whether they give a positive or negative contribution.

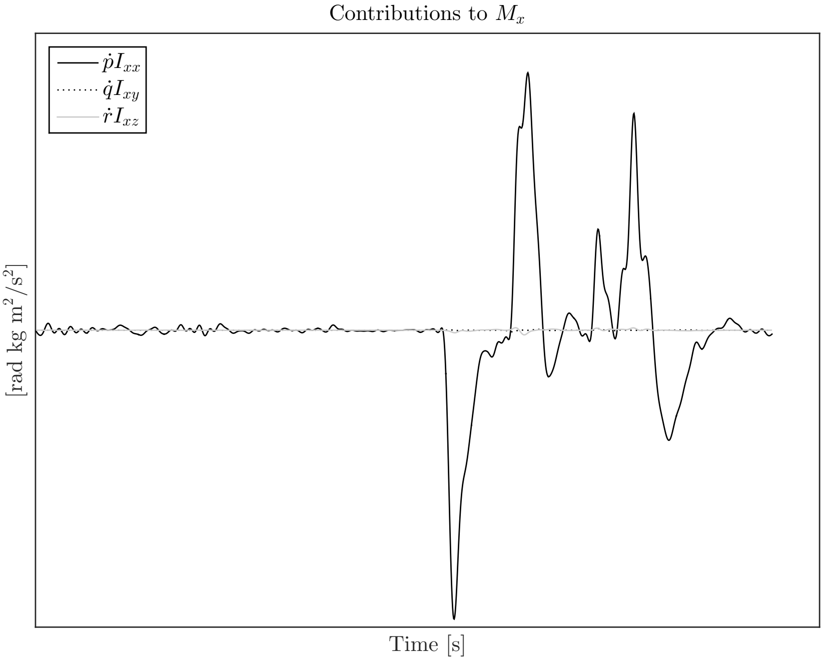


Figure 6.1: *The contribution from the different angular accelerations and moments of inertia to the roll moment.*

The variations are calculated using (6.9) and become 10.7 %, 6.5 % and 4.8 % for roll, pitch and yaw respectively.

The dynamic pressure q_a defined in (3.24) vary with 7.2 %. In order to investigate how the changes in dynamic pressure and inertia affect the coefficients, these are approximated. It is assumed that the aerodynamic moments are the only moments acting on the aircraft and the coefficients are calculated by dividing the roll and yaw moments by $q_a S b$ and the pitch moment by $q_a S c$. The variations of the coefficients C_l, C_m and C_n then become 18.6 %, 14.6 % and 12.5% respectively. This is considered too much for the approximation of constant dynamic pressure and moments of inertia to be likely to give good results.

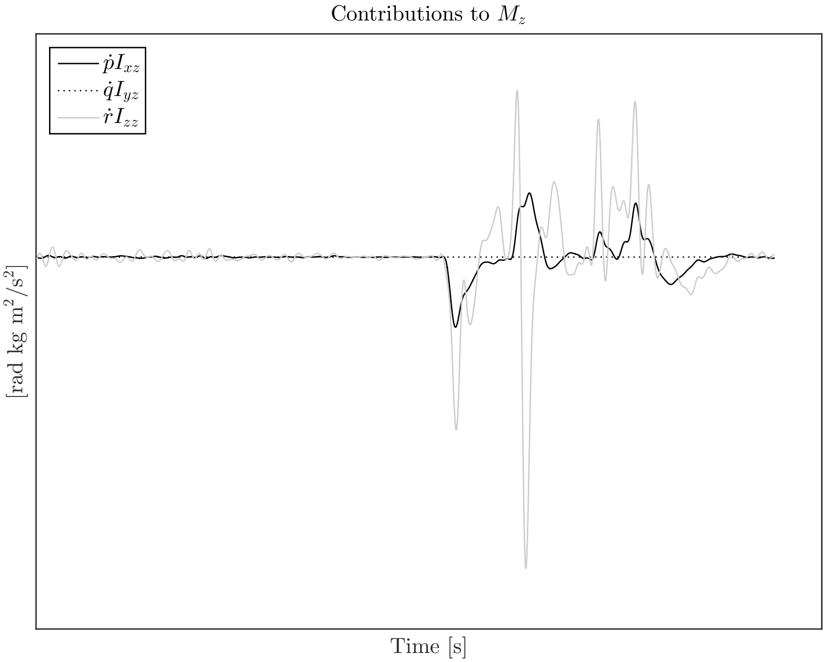


Figure 6.2: *The contribution from the different angular accelerations and moments of inertia to the yaw moment.*

6.3 Analysis of Differentiation Methods

In this section, the accuracy of the differentiation filter and the accuracy of the differentiation in the frequency domain are analysed. An analytical signal which is analytically differentiated and transformed using the CZT is used in the analysis. This gives a true reference of the derivative, which is not available for simulated or real flight test data. The same analytical signal is differentiated with the FIR differentiation filter and transformed, and differentiated within the transform using Larsson's algorithm. Also Morelli's algorithm contains a differentiation but since this is approximated and neglects more terms than Larsson's algorithm, it is assumed to give worse results. Only Larsson's algorithm is therefore considered in this analysis.

Both Larsson's algorithm and the FIR filter are limited to a frequency interval of 0.1 to 2.0 Hz. The analytical signal is therefore a multisine constructed within this frequency interval. The signal is chosen as

$$y = A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t) \tag{6.11}$$

with $\omega_1 = 0.3 \cdot 2\pi$, $\omega_2 = 1.5 \cdot 2\pi$ and the analytical derivative

$$\dot{y} = A_1 \omega_1 \cos(\omega_1 t) + A_2 \omega_2 \cos(\omega_2 t). \tag{6.12}$$

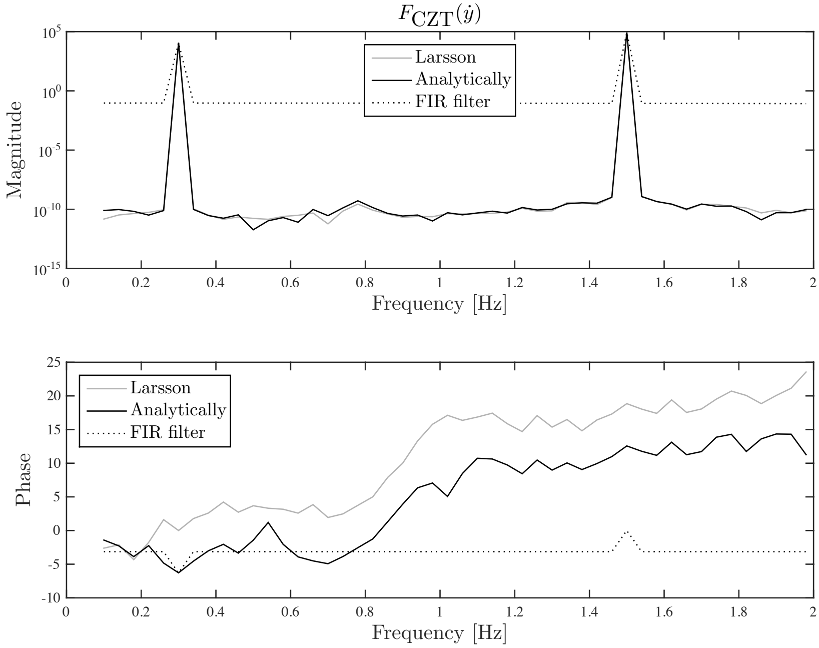


Figure 6.3: *The magnitude and phase for the transformed analytical signal y differentiated using Larsson's algorithm, analytically and using the SAAB FIR filter for differentiation.*

The differentiated and transformed analytical signals are presented in Figure 6.3. It is clear that Larsson's method of differentiation in the frequency domain is better than the FIR filter since it has a closer resemblance to the analytically differentiated signal. The weakness of the FIR filter is mainly in holding down the frequencies not present in the signal. For a real signal with a more continuous frequency content, the difference should be smaller. Figure 6.4 shows the transformed real data of the pitch angular acceleration $\dot{q}$ for which the difference is most clear. There is a visible difference between the two differentiation methods also for real data, but it is smaller than for analytical data as expected. However, since the true derivative $\dot{q}$ is not known, no conclusion as to which is better can be drawn from this.

How the difference in differentiation method affects the estimates can be seen in Figure 6.5. Estimation results using the differentiation in the frequency domain and the differentiation using the FIR filter are compared using post-processed flight test data. The excitation of the different signals is given together with the description of the data in Figure 5.3. The estimates using the two differentiation methods are generally very similar, especially at the end of the excitation. In the beginning of the excitation, the difference is bigger. The FIR filter estimates are better than the ones using Larsson's differentiation for example at the 14th time-step in the estimation of $C_{l_{\delta_a}}$ and $C_{l_{\delta_r}}$, and the 15th and 16th time-steps

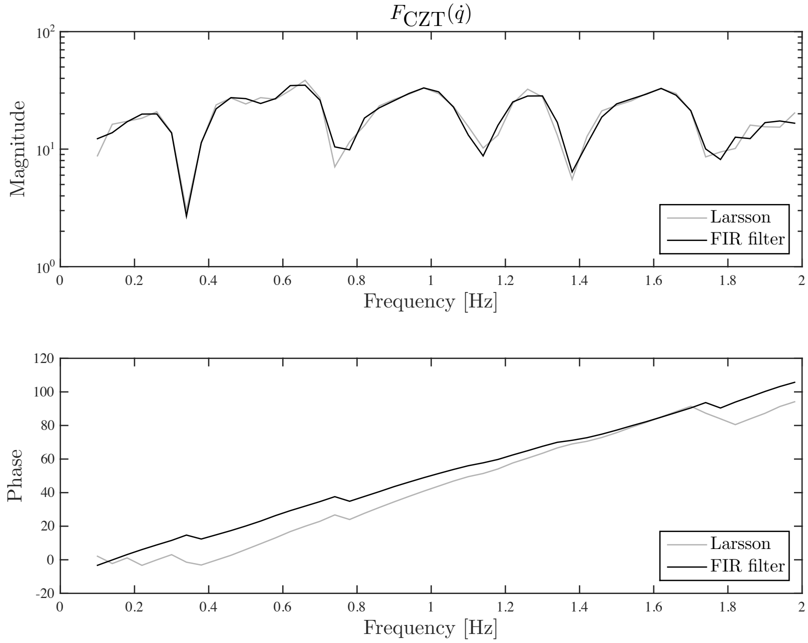


Figure 6.4: *The magnitude and phase of the transformed pitch angular acceleration $\dot{q}$ differentiated using Larsson's algorithm and using the SAAB FIR filter for differentiation.*

in the estimation of C_{l_α}. For both $C_{l_{\hat{p}}}$ and $C_{l_{\hat{r}}}$, Larsson's differentiation gives better estimates at the 14^{th} time-step but the FIR filter differentiation gives better estimates at the 15^{th} time-step. The results are not conclusive and no method can be said to have a better general performance than the other based on this single data set. However, it can be concluded that the significant weakness of the FIR differentiation seen on analytical data does not appear to have a significant negative affect on the estimates using flight test data.

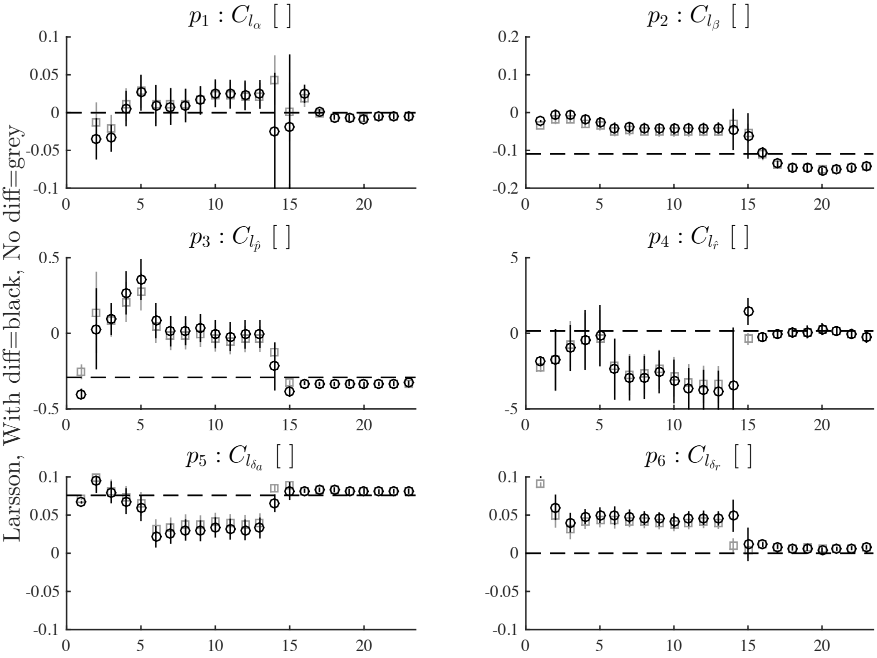


Figure 6.5: *Estimation using Larsson's algorithm and with the inertia and dynamic pressure approximated as constant. The black circles are estimates where the differentiation is performed within the algorithm. The grey squares are estimates that use the coefficient calculated using the FIR filter. The vertical lines indicate two standard deviations of the estimate. No IVs are used in the algorithms.*

6.4 Comparison of Algorithms Using Simulated Data

In this section, the results from the evaluation of Larsson's and Morelli's algorithms on simulated data are presented. The simulated data is generated using the simulation model described in Chapter 4 and 200 MC simulations with varying noise are used for all evaluations. Since the simulation model assumes the moments of inertia and dynamic pressure to be constant, the moment coefficient expression can be differentiated within the algorithm without taking the issue described in Section 6.1.1 into account.

Figure 6.6 shows the simulated regressors and the estimated parameters using Larsson's and Morelli's algorithms. There are no estimates using Larsson's algorithm before excitation since the instrumental variables are noise free and therefore zero before excitation. Since the force coefficient contains no differentiation, the only difference between the two algorithms for C_N is the presence of the IVs in Larsson's algorithm. For low levels of measurement noise and process disturbances, such as in Figure 6.6, the IVs give a negligible contribution and the

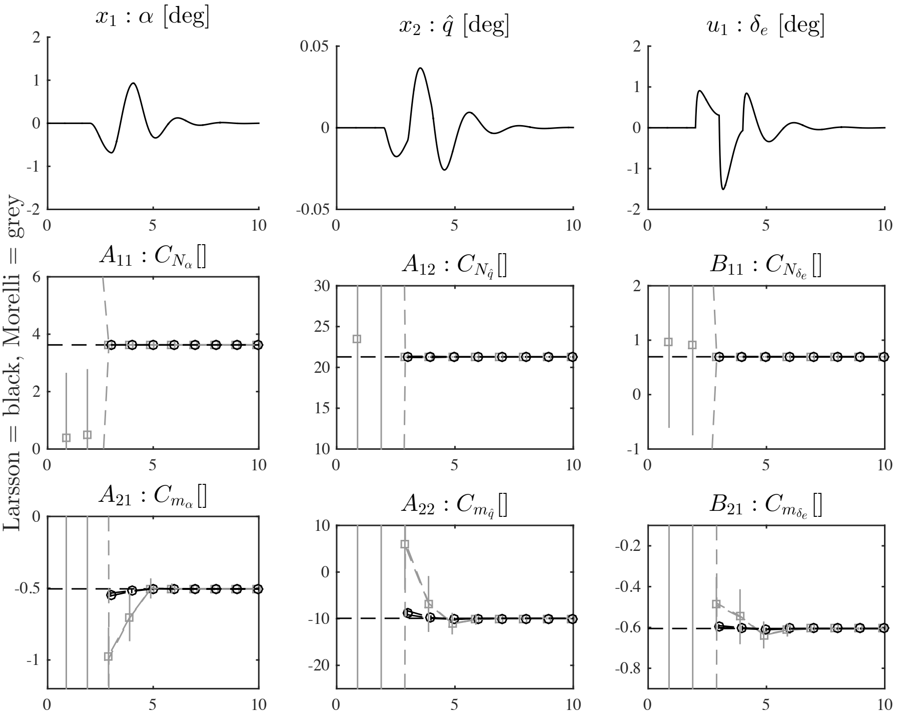


Figure 6.6: *Estimation using Larsson's (black) and Morelli's (grey) algorithms on simulated data with low noise. The moment coefficient C_m is differentiated within the algorithms. The circles/squares are the mean estimates over 200 simulations and the vertical lines are the mean of their estimated two standard deviations. The dashed black and grey lines are the two standard deviations based on the MC simulations. These can be hard to see in the figure because the standard deviations are so small for low noise. The straight horizontal dashed lines indicate the true parameters used in the simulation model.*

estimates of C_{N_α}, $C_{N_{\hat{q}}}$ and $C_{N_{\delta_e}}$ are the same using the two methods. This can be seen in Table 6.1 as well, where the mean value of the last estimate is presented. For the moment coefficient, however, the difference in accuracy due to the correction of the differentiation is clearly visible. For all three derivatives C_{m_α}, $C_{m_{\hat{q}}}$ and $C_{m_{\delta_e}}$, Larsson's algorithm gives better estimates. This is especially clear at the beginning of the excitation before the two methods converge towards the true model value. By the end of the excitation, the difference becomes significantly smaller. In Table 6.1, the mean of the last estimates are presented. There is a small difference in the estimates of $C_{m_{\hat{q}}}$ and $C_{m_{\delta_e}}$, where Larsson's gives the smaller bias. Morelli's algorithm also results in a slightly larger estimated standard deviation for the last estimate of the moment derivatives. The true parameter is, however, included in the interval of two estimated standard deviations

Table 6.1: *The mean value of the last estimates using Larsson's and Morelli's algorithms compared to the true value used in the simulation model.*

Parameter	Model	Larsson	Morelli	Unit
C_{N_α}	3.6268	3.6271 ± 0.0003	3.6271 ± 0.0003	[-]
$C_{N_{\hat{q}}}$	21.2876	21.2879 ± 0.0105	21.2879 ± 0.0105	[-]
$C_{N_{\delta_e}}$	0.6951	0.6952 ± 0.0003	0.6952 ± 0.0003	[-]
C_{m_α}	-0.5046	-0.5052 ± 0.0016	-0.5052 ± 0.0018	[-]
$C_{m_{\hat{q}}}$	-9.9176	-9.9255 ± 0.0513	-9.9276 ± 0.0593	[-]
$C_{m_{\delta_e}}$	-0.6051	-0.6054 ± 0.0015	-0.6055 ± 0.0017	[-]

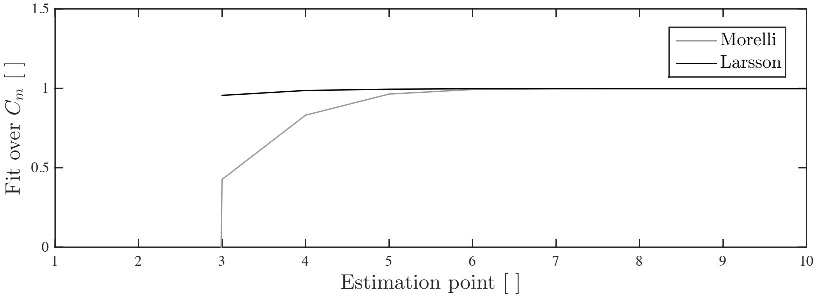


Figure 6.7: *The mean goodness of fit for the pitch moment coefficient. The mean is calculated over all estimates for the* MC *simulation presented in Figure 6.6.*

for all parameters using both algorithms. The difference between the algorithms can also be seen in the goodness of fit, calculated using the NRMSE and presented in Figure 6.7. The goodness of fit is visibly better using Larsson's method for the first three estimates after the start of the excitation before both methods converge towards perfect fit.

To summarise, both algorithms give a small bias for low noise. For the force coefficient, the results are the same using both algorithms indicating that the IVs do not have a negative effect on the estimates in this case. For the moment coefficient, Larsson's algorithm gives better estimates, especially at the beginning of the excitation. This result is comparable to what was found in Larsson (2013) for the state space model. It is the corrected differentiation that results in better estimates during the excitation. This is explained in Larsson (2013) to be caused by the fact that the added last sample becomes a correction term for the CZT.

Figure 6.8 shows the result of 200 MC simulations with high disturbance levels and using already calculated moment coefficients instead of differentiation within the algorithms. When calculated coefficients are used, the only difference

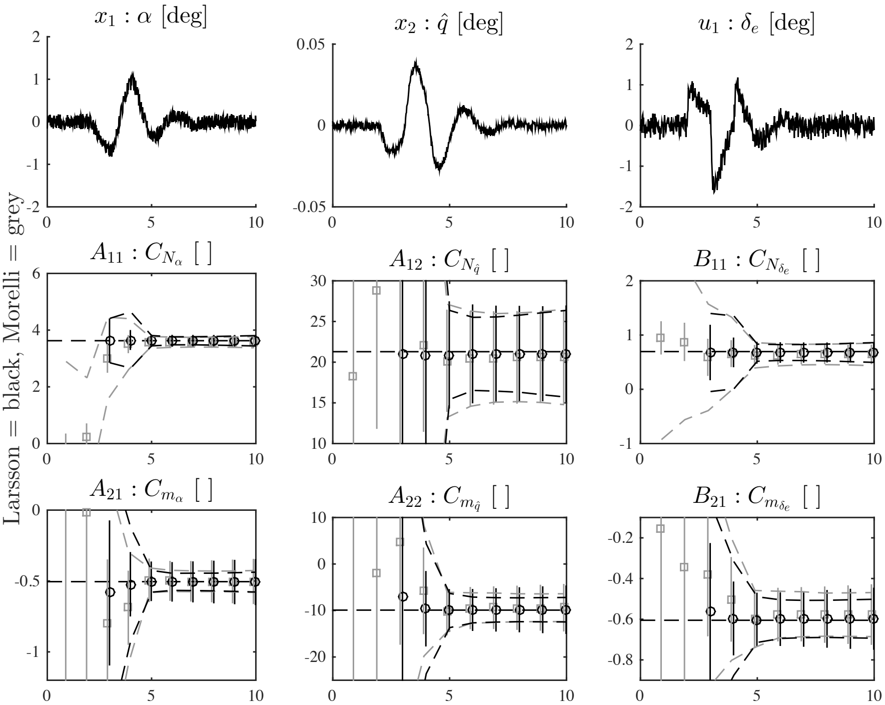


__Figure 6.8:__ Estimation using Larsson's (black) and Morelli's (grey) algorithms on simulated data with high noise. The moment coefficient C_m is calculated before transformation and no differentiation is therefore part of the algorithms. The circles/squares are the mean estimates over 200 simulations and the vertical lines are the mean of their estimated two standard deviations. The dashed black and grey lines are the two standard deviations based on the MC simulations. The straight horizontal dashed lines indicate the true parameters used in the simulation model.

between the two algorithms consist of the IVs and this allows to isolate the effect of the IVs. At low disturbance levels, Larsson's and Morelli's algorithms give the same results and only the result with high disturbance levels is presented.

When the disturbance levels are increased, the standard deviation of the estimates becomes significantly larger as can be seen when comparing Figure 6.8 to Figure 6.6. The estimated standard deviation follows the standard deviation over the MC simulations. This suggests that the estimated standard deviation can in fact be used as a measure of the reliability of the estimates.

A zoom of Figure 6.8 is presented in Figure 6.9 to illustrate the differences between the algorithms. For high disturbance levels, Larsson's algorithm performs better than Morelli's due to the IVs. Both methods result in estimates with a larger bias for high disturbance levels than for low disturbance levels. However,

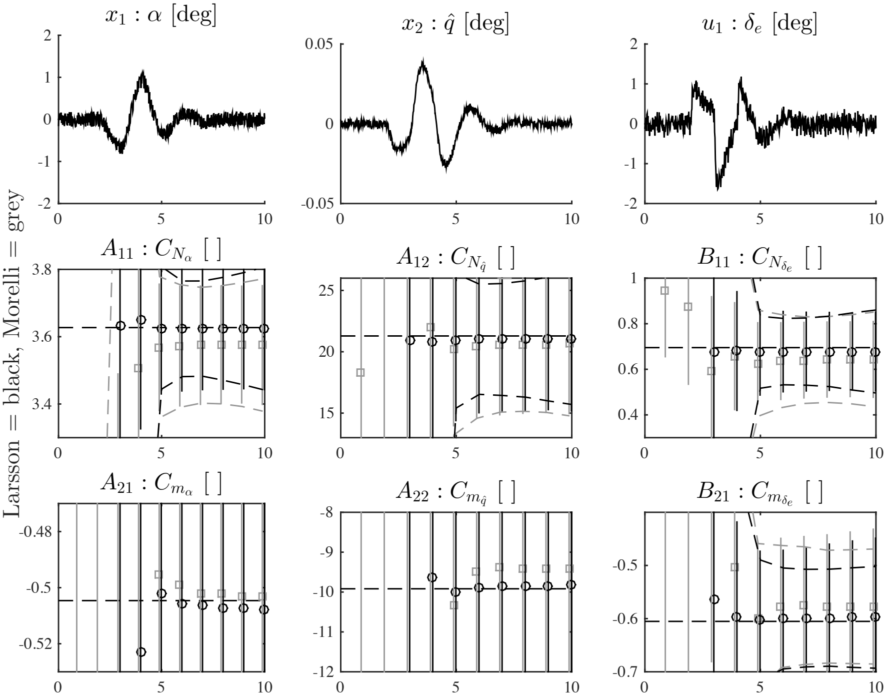


Figure 6.9: *Estimation using Larsson's (black) and Morelli's (grey) algorithms on simulated data with high noise and calculated coefficients. A zoom of the graph presented in Figure 6.8.*

the bias of Larsson's method is smaller than that of Morelli's for all derivatives. Also this result is comparable to what was found in Larsson (2013).

6.5 Determining the Reference Derivatives

When Larsson's and Morelli's algorithms are compared using simulated data, the true parameters are known since they are used in the simulation model creating the data. In the final application, however, one goal is to evaluate the quality of the simulation model and the true aerodynamic derivatives are unknown. The model derivatives used for comparison are called the reference derivatives. These should be taken from the aerodynamic model for the trimmed condition right before the start of the manoeuvre. However, the aerodynamic model does not generally include aerodynamic derivatives explicitly. Instead, they must be calculated from the look-up table of the model. The simplest solution, a first order Euler approximation, is used for this purpose in this work. In order to access the aerodynamic model in real time, ROMAC must be used. However, as explained in Section 6.1.1, ROMAC has reached the limit for how many parameters that can be returned and this cannot be realised with today's version of ROMAC.

6.6 Comparison of Algorithms Using Post-Processed Flight Test Data

A comparison between Larsson's and Morelli's algorithms is performed using post-processed flight test data and the results are presented in this section. The states and inputs are given in Figure 5.3 together with the presentation of the data. Since post-processed flight test data is used for the comparison, no IVs are available, as explained in Section 5.3.2. The reference derivatives used in the comparison are obtained as described in Section 6.5.

Larsson's and Morelli's algorithms are compared with the moments of inertia and dynamic pressure approximated as constant. The result is presented in Figure 6.10. Note that some of the estimates using Morelli's algorithm are outside the borders of the graph. Since the true parameters are unknown, it is difficult to evaluate the result but it is clear that Larsson's algorithm gives better estimates at the beginning of the excitation. Later during the excitation, the estimates using the two algorithms converge and give similar values. For the estimation at the 20th time-step, however, the estimates using Morelli's algorithm deviate slightly

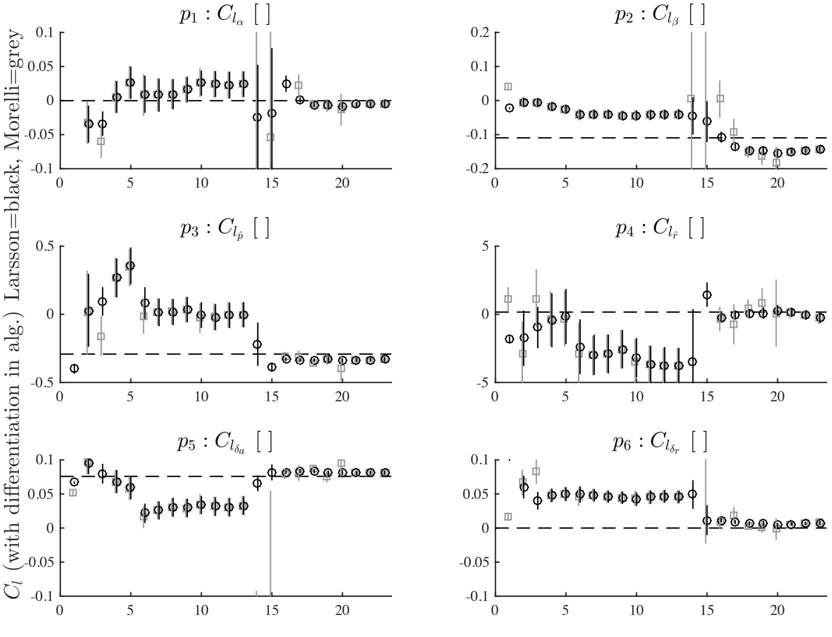


Figure 6.10: *A comparison between Larsson's (black) and Morelli's (grey) algorithms on post-processed flight test data. The moments of inertia and dynamic pressure are approximated as constant and no IVs are used. The circles/squares are the estimates and the vertical lines are their estimated two standard deviations. The straight horizontal dashed lines indicate the reference derivatives taken from the model.*

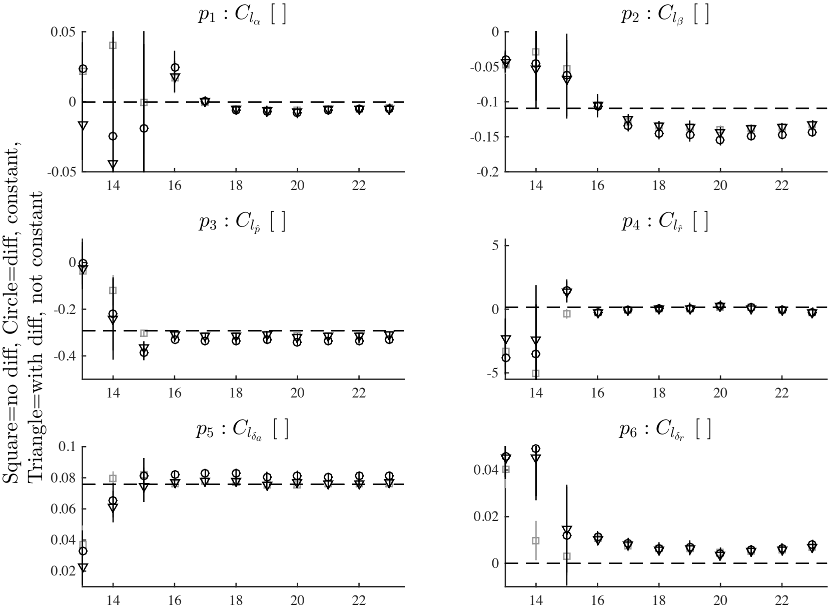


Figure 6.11: *A comparison between the estimates using regression A (grey square), B (black circle) and C (black triangle). For all estimations, Larsson's algorithm is used. The vertical lines are the estimated two standard deviations and the straight horizontal dashed lines indicate the reference derivatives taken from the model. A zoom of this figure is given in Figure 6.12.*

and inexplicably.

The estimated standard deviation has a generally small value and it is difficult to determine from the figure when the excitation starts. This property is non-desirable when trying to draw conclusions from a data collecting perspective. Unlike for simulated data, the final estimate is not included in the interval of the estimated standard deviations of previous estimates. Since the estimated standard deviation is so small, this could result in the wrong conclusions from a safety perspective as well. The small estimated standard deviation could be caused by the fact that the data is processed.

In Figure 6.11, three of the different ways to calculate the outputs and formulate the regressors discussed in Section 6.1 are compared. These will be called regression A, B and C. For all variants of the regression, the estimates are calculated using Larsson's algorithm. Regression A uses previously calculated coefficients and the ordinary regressors of (3.27). Regression B uses differentiation within the algorithm, the ordinary regressors and the moments of inertia and dynamic pressure approximated as constant. Regression C uses differentiation within the algorithm and the true moments of inertia and dynamic pressure. As described in Section 6.1.1, this requires a modification of the regressors and the reinstating

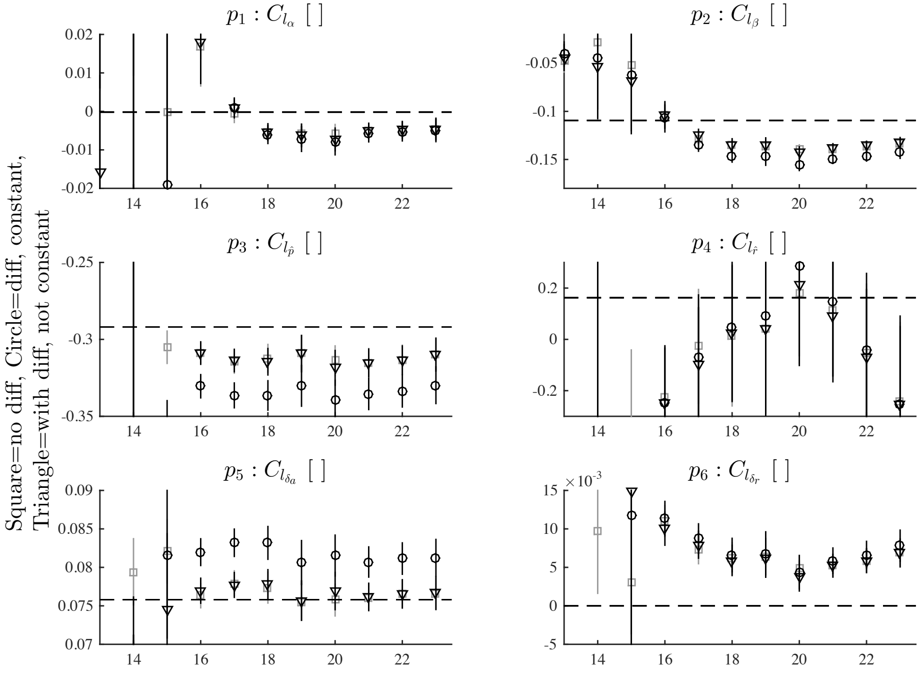


Figure 6.12: *A zoom of the comparison in Figure 6.11. Estimation using regression A (grey square), B (black circle) and C (black triangle).*

of the term $\dot{\boldsymbol{I}}\boldsymbol{\omega}$. The modified regressors of (6.6) become

$$\begin{aligned}\boldsymbol{\phi}' = [&\Delta(\alpha q_a S l_{ref}) \quad \Delta(\beta q_a S l_{ref}) \quad \Delta(\hat{p} q_a S l_{ref}) \quad \Delta(\hat{q} q_a S l_{ref})... \\ &...\Delta(\hat{r} q_a S l_{ref}) \quad \Delta(\delta_c q_a S l_{ref}) \quad \Delta(\delta_e q_a S l_{ref})... \\ &...\Delta(\delta_a q_a S l_{ref}) \quad \Delta(\delta_r q_a S l_{ref}) \quad \Delta(\delta_{le} q_a S l_{ref})]\end{aligned} \tag{6.13}$$

when linearised. As described previously, Δ denotes the disturbance from the trimmed condition and the new, calculated regressors are considered as single signals and not multiplications of several signals.

A zoom of Figure 6.11, where the results are more clearly visible, is presented in Figure 6.12. Both graphs are limited to the interval of excitation. For $C_{l_{\hat{r}}}$ and $C_{l_{\delta_r}}$, the variations between the estimates is greater than the variations using the different methods and no conclusions can be drawn. However, for the remaining derivatives the approximation of constant moments of inertia and dynamic pressure results in a greater bias from the reference derivatives. Note that the reference derivatives are taken from the model and not necessarily true but judging by the sensitivity analysis and the size of the difference, it is likely that the estimates of regression A and C are closer to the truth. Regression A and C result in similar estimates and no conclusion can be drawn as to which is better.

7 Evaluation of Methods for Handling of Missing Data

In this chapter, the different methods for handling of missing data presented in Section 2.4 are compared. The final choice of method depends on its performance and on its suitability from an implementation point of view. The different implementations and their advantages and disadvantages are presented initially. Secondly, an evaluation of the methods using analytical data is presented. Lastly, the results when two of the methods are compared using simulated data are presented.

7.1 Implementation

The four investigated methods for handling of missing data have different suitability from an implementation and computation perspective. These affect the simplicity and the run-time of the estimation algorithm. Their complexity depend on how the CZT is implemented. The CZT can be implemented to use multiples of the sampling interval and assume all samples to be spaced by T_s, previously defined in (2.8) as

$$\mathscr{F}_{\text{CZT}}(f(t)) = \tilde{F}(\omega_i) = \sum_{k=0}^{N-2} f_k e^{-j\omega_i k T_s}, \qquad \omega_i \in \boldsymbol{\omega} = \{0.10, 0.14, \ldots, 1.98\}.$$

Alternatively, it can be implemented to use the true times and spacing between the samples, previously formulated in (2.23) as

$$\mathscr{F}_{\text{CZT}}(f(t)) = \tilde{F}(\omega_i) = \sum_{k=0}^{N-2} \Delta_k f_k e^{-j\omega_i t_k}, \qquad \omega_i \in \boldsymbol{\omega} = \{0.10, 0.14, \ldots, 1.98\}.$$

The simplest method is the discard method as it requires no implementation provided that (2.8) is used to implement the CZT. The hold method requires a calculation of the number of missing samples. It also requires the last value of all signals to be saved but this is not a problem given the functionality of StellaRT.

The linear interpolation method requires the most implementation and the most computations. All used signals must be linearly interpolated. This could mean as many as 60 signals, depending on how the regression is formulated and the outputs are calculated. The variable sample time method requires no extra implementation provided that (2.23) is used. This implementation of the CZT does, however, require an extra calculation compared to that of (2.8) since the spacing between samples Δ_k is needed. In reality, the spacing between samples is calculated using all metods except the discard method as it is used to decide whether samples are missing or not.

7.2 Initial Evaluation Using Analytical Data

In this section, the different methods for handling of missing data are compared by removing samples from an analytical signal. The difference between the transforms of the reconstructed signals and the original signal are presented for a varying number of missing samples. The analytical signal used is the same as in the differentiation analysis, namely

$$z = A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t)$$

with $\omega_1 = 0.3 \cdot 2\pi$ and $\omega_2 = 1.5 \cdot 2\pi$ that are both within the frequency spectra of the CZT. The sample frequency is 60 Hz which means that one data frame consists of three to four samples since the data is sent in 16 Hz, as explained in Section 2.4.

Figure 7.1 shows the signals $Z_{1_x} - Z_{\text{ref}}$ where $Z = \mathscr{F}_{\text{CZT}}(z)$ and the subscript x indicates the method for handling the removed samples. The subscript 1 indicates that one frame, that is four samples, has been removed. The figure thus shows the magnitude and phase for the difference between the transformed original signal and the transformed signal with missing data, using all four methods. The 2-norm of this difference is presented in Table 7.1.

Both the hold method and the linear interpolation method diverge little from the reference signal. Surprisingly, the hold method gives a smaller difference than the linear interpolation method when only one frame is missing. The difference from the reference signal is also uniform for the entire frequency interval using the hold method and the linear interpolation method. This is not the case for the other two methods. The discard method gives the greatest difference and has peaks that show that the difference is larger at the frequencies present in the signal.

The same comparison but with four consecutive missing frames is presented in Figure 7.2. The 2-norm of the difference is presented in Table 7.2. The discard

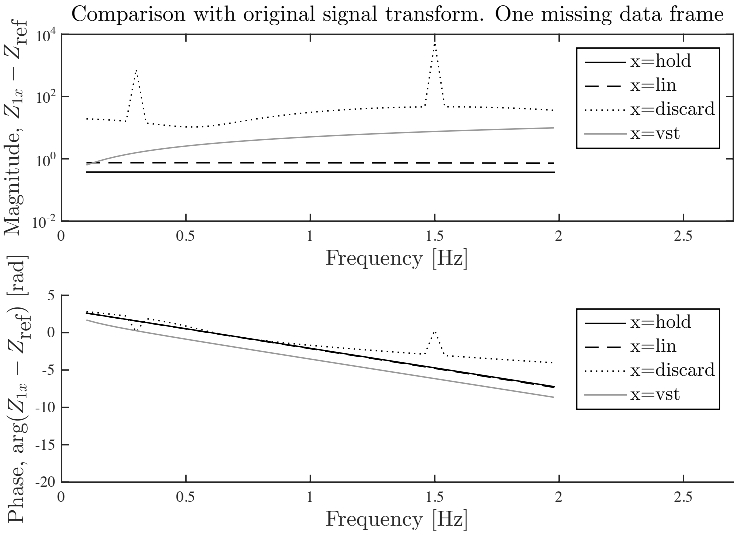


Figure 7.1: *Magnitude and phase for the CZT of the signals with one missing data frame compared to the CZT of the original signal. The different compared methods for handling the missing data are hold, linear interpolation, discard and VST.*

Table 7.1: *The 2-norm of the difference between the CZT of the complete analytical signal and the CZT of the analytical signal with one missing data frame. The specified methods are the ones used to handle the missing data.*

Method	**2-Norm of difference**
Discard	$5.6 \cdot 10^3$
Hold	2.6
Linear interpolation	5.1
VST	41

method has even larger peaks compared to when only one frame is missing and is more uneven in general. The remaining three methods are quite similar in shape but the size of the difference vary. The linear interpolation method gives the smallest difference followed by the hold method and the VST method.

Figure 7.3 shows the comparison when eight consecutive frames have been removed. The 2-norm of the difference is presented in Table 7.3. The changes are similar to those between one and four frames where the discard method performs even worse, and the remaining three methods have become more similar. The linear interpolation method still gives the smallest difference, followed by the hold method and the VST method.

When sixteen frames are removed, the hold method gives a significant dip as

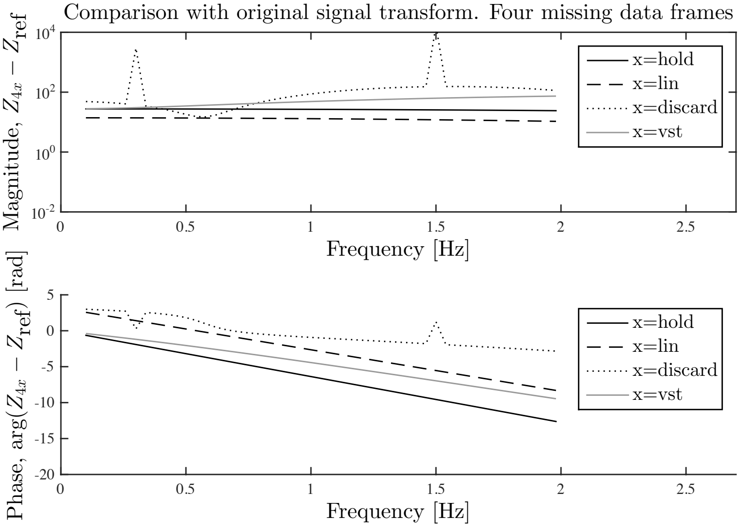


Figure 7.2: *Magnitude and phase for the CZT of the signals with four missing data frames compared to the CZT of the original signal. The different compared methods for handling the missing data are hold, linear interpolation, discard and VST.*

Table 7.2: *The 2-norm of the difference between the CZT of the complete analytical signal and the CZT of the analytical signal with four missing data frames. The specified methods are the ones used to handle the missing data.*

Method	**2-Norm of difference**
Discard	$1.7 \cdot 10^4$
Hold	$1.8 \cdot 10^2$
Linear interpolation	88
VST	$3.6 \cdot 10^2$

shown in Figure 7.4. The 2-norm of the difference compared to the complete signal is presented in Table 7.4. The discard method still performs the worst. The linear interpolation and the VST methods give more uniform differences compared to the others. However, the linear interpolation still has the smallest difference, followed by the hold method and the VST method.

To summarise, the linear interpolation method gives the smallest difference for all cases except when only one frame is missing. In this case, the hold method gives the smallest difference. However, as described i Section 2.4, the method needs to be able to handle at least four to sixteen frames of missing data, so this first case where only one frame is missing is of less interest. The discard method gives the largest difference for all cases and is also the method for which the difference has the greatest variation throughout the frequency band. The VST method

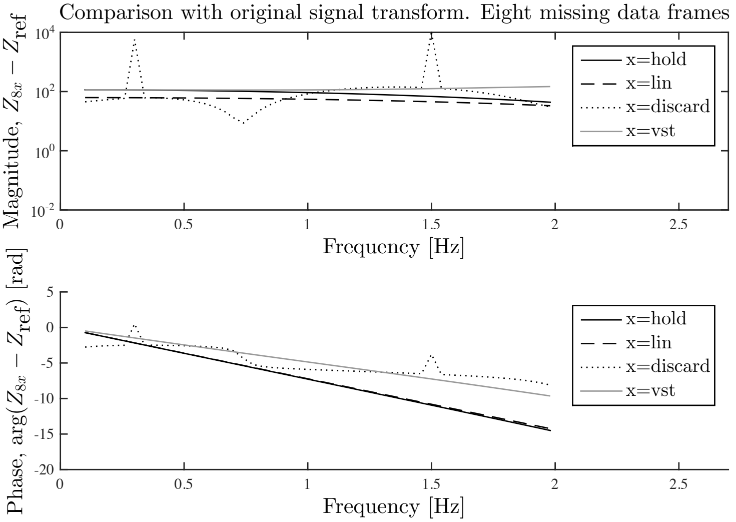


Figure 7.3: *Magnitude and phase for the CZT of the signals with eight missing data frames compared to the CZT of the original signal. The different compared methods for handling the missing data are hold, linear interpolation, discard and VST.*

Table 7.3: *The 2-norm of the difference between the CZT of the complete analytical signal and the CZT of the analytical signal with eight missing data frames. The specified methods are the ones used to handle the missing data.*

Method	**2-Norm of difference**
Discard	$1.2 \cdot 10^4$
Hold	$6.2 \cdot 10^2$
Linear interpolation	$3.7 \cdot 10^2$
VST	$8.4 \cdot 10^2$

gives a greater difference than both the linear interpolation and hold methods for all cases. For all cases except the first, where only one frame is missing, the difference using VST is fairly uniform. This makes VST preferable over the hold method for a higher number of missing data. Additionally, both hold and linear interpolation are interpolation methods and are therefore expected to give similar results in estimations. Since the linear interpolation method suggests a better performance, the hold method is not chosen for further study. The discard method is not chosen either due to its poor performance. The VST is chosen for further study as it gives fairly good results and is structurally different to the linear interpolation which makes it interesting for a comparison.

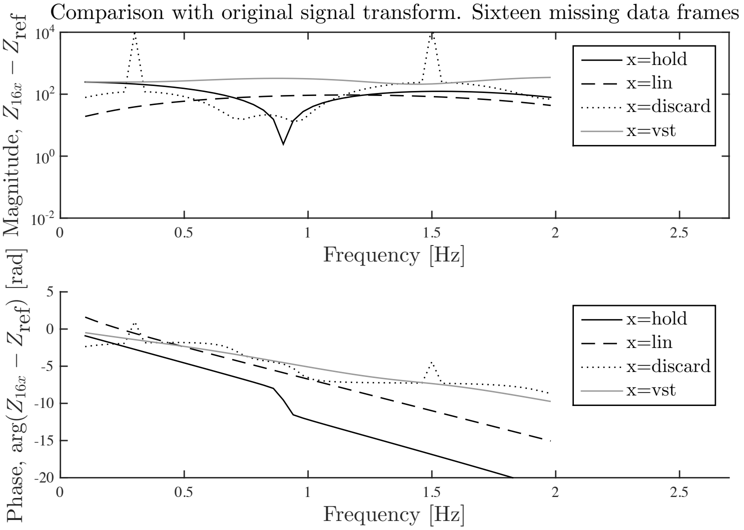


***Figure 7.4:** Magnitude and phase for the CZT of the signals with sixteen missing data frames compared to the CZT of the original signal. The different compared methods for handling the missing data are hold, linear interpolation, discard and VST.*

***Table 7.4:** The 2-norm of the difference between the CZT of the complete analytical signal and the CZT of the analytical signal with 16 missing data frames. The specified methods are the ones used to handle the missing data.*

Method	2-Norm of difference
Discard	$2.0 \cdot 10^4$
Hold	$9.0 \cdot 10^2$
Linear interpolation	$5.0 \cdot 10^2$
VST	$1.9 \cdot 10^3$

7.3 Evaluation Using Simulated Data

The result when evaluating the general properties of the estimators using the linear interpolation and VST methods for handling of missing data are presented in this section. The simulation model described in Chapter 4 is used in the evaluation by performing MC simulations. To make the study more limited, only one of the algorithms is studied. Larsson's algorithm is chosen for the comparison since this includes the effect of the IVs.

Figure 7.5, with a zoom in Figure 7.6, and Figure 7.7, with a zoom in Figure 7.8, show the results from 200 simulations where six consecutive frames of data have been removed randomly within the interval of excitation. The vertical lines on the estimates indicate the mean of the estimated two standard deviations. The

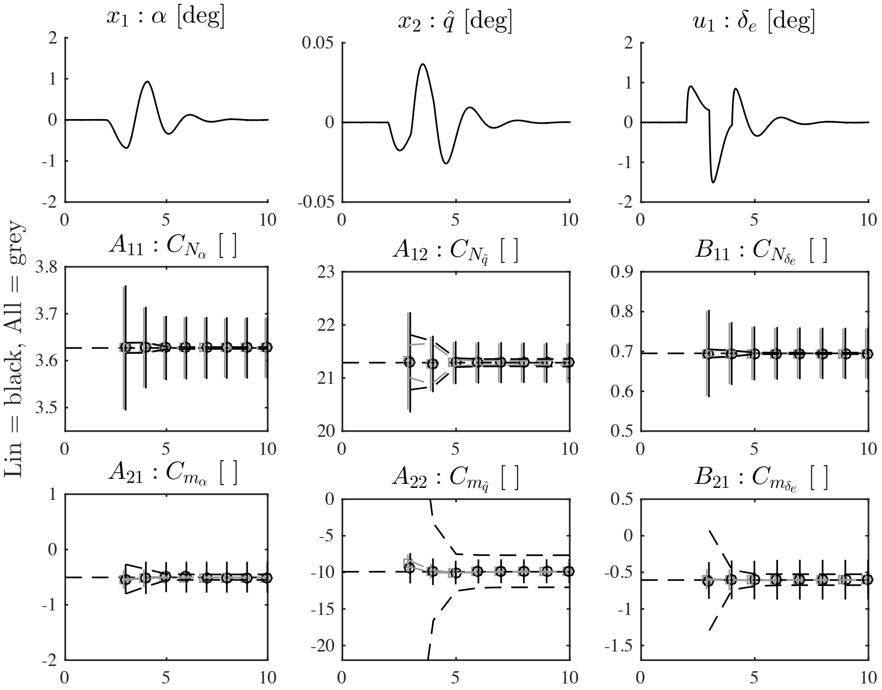


__Figure 7.5:__ Comparison between estimates with all data present (grey) and using the linear interpolation method (black) to handle missing data. The result of 200 MC simulations with six consecutive data frames randomly missing during excitation. The algorithm used is Larsson's algorithm with differentiation within the algorithm. The circles/squares are the mean estimates over 200 simulations and the vertical lines are the mean of their estimated two standard deviations. The dashed black and grey lines are the two standard deviations based on the MC simulations. The straight horizontal dashed lines indicate the true parameters used in the simulation model.

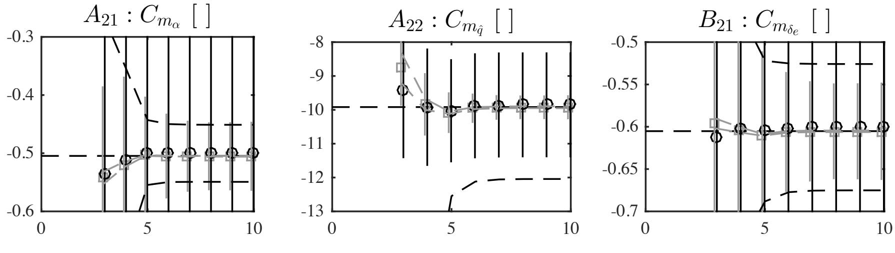


__Figure 7.6:__ A zoom of the moment coefficients in Figure 7.5. Estimation with all data present (grey) and using the linear interpolation method (black) to handle missing data.

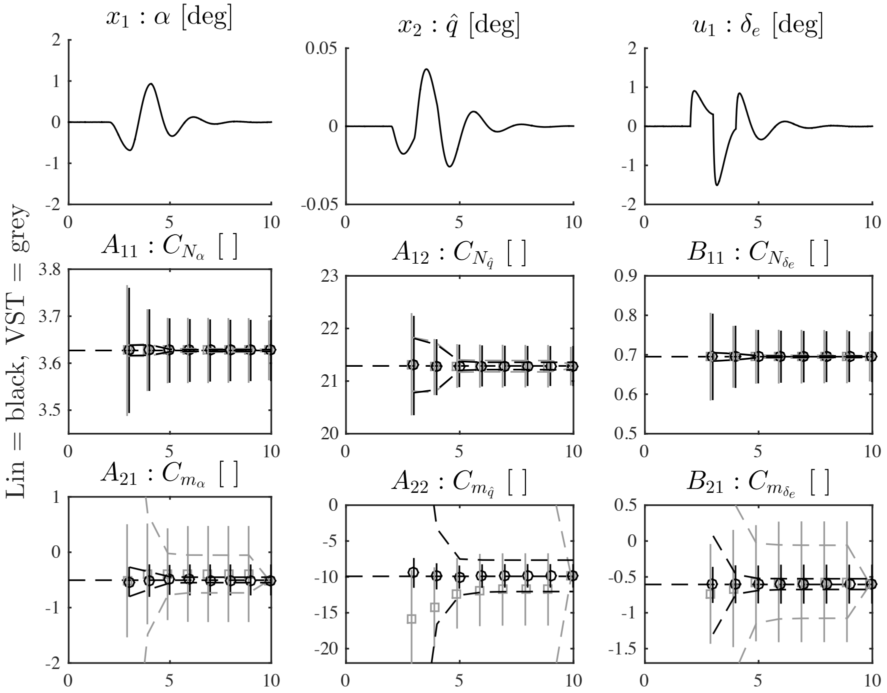


Figure 7.7: *Comparison between estimates using the linear interpolation method (black) and the* VST *method (grey) to handle missing data. The result of 200* MC *simulations with six consecutive data frames randomly missing during excitation. The algorithm used is Larsson's algorithm with differentiation within the algorithm. The circles/squares are the mean estimates over 200 simulations and the vertical lines are the mean of their estimated two standard deviations. The dashed black and grey lines are the two standard deviations based on the* MC *simulations. The straight horizontal dashed lines indicate the true parameters used in the simulation model.*

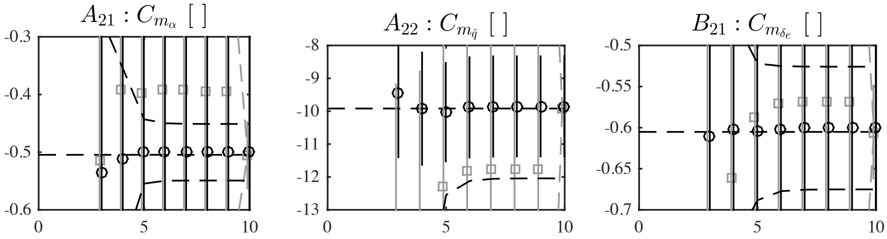


Figure 7.8: *A zoom of the moment coefficients in Figure 7.7. Estimation using the linear interpolation method (black) and the* VST *method (grey) to handle missing data.*

black and grey dashed lines indicate the two standard deviations over all MC simulations. Figure 7.5 shows the comparison between estimates where all data is present and estimates where the linear interpolation method is used. Figure 7.7 shows the comparison between the linear interpolation method and the VST method.

When data is missing in the data set, the estimates have a larger bias. For the force coefficient derivatives, this bias is similar using both methods and so small that it is not visible in the figures. The bias of the moment coefficient derivatives is larger. A zoom of the moment coefficient derivatives of Figure 7.7 is presented in Figure 7.8. In this figure, the bias is clearly visible. The size of the bias is of the same magnitude as that of the bias for high noise levels presented in Figure 6.9. The moment derivatives, especially $C_{m_{\hat{q}}}$, appear to be more difficult to estimate accurately and show a greater difference between the methods, where the VST method gives a greater bias. These estimates also have a larger variance for both methods compared to the estimates where all data is present. The estimates using the VST method have a larger variance than the ones where the linear interpolation method is used. The fact that the estimates for the force are more accurate than those for the moment suggests that the differentiation affects the estimates in a negative way when data is missing. The effect of the differentiation seems to be larger when using the VST method since there are small differences between the methods when studying the force coefficient.

Figure 7.9, with a zoom in Figure 7.10 and Figure 7.11, with a zoom in Figure 7.12 show the corresponding estimates using previously calculated moment coefficients. This means that no differentiation is performed within the algorithms. The results are from 200 simulations where six consecutive frames of data have been removed randomly within the interval of excitation. These simulations have a slightly lower variance for the moment derivative estimates using the linear interpolation method, compared to those in Figure 7.7. Using the VST method, the variances are significantly smaller compared to those in Figure 7.7. When using already calculated coefficients, the VST method gives smaller variances than the linear interpolation method, as opposed to in Figure 7.7, where the linear interpolation method performs better.

On real flight test data, however, the improvement using calculated coefficients is expected to be smaller. In this simulation, the removed samples do not affect the quality of the remaining samples. In reality, the filters used to calculate the coefficients are reset or use linear interpolation in order to function where data is missing. This might have a negative affect also on values surrounding the missing data. An analysis of this behaviour cannot currently be performed since ROMAC cannot return additional parameters.

To summarise the observations of these results, the method of choice for handling of missing data appears to depend on the presence of a differentiation within the algorithm. If differentiation within the algorithm is used, the results indicate that the linear interpolation method has a better performance. With previously calculated coefficients, the results indicate the VST metod instead.

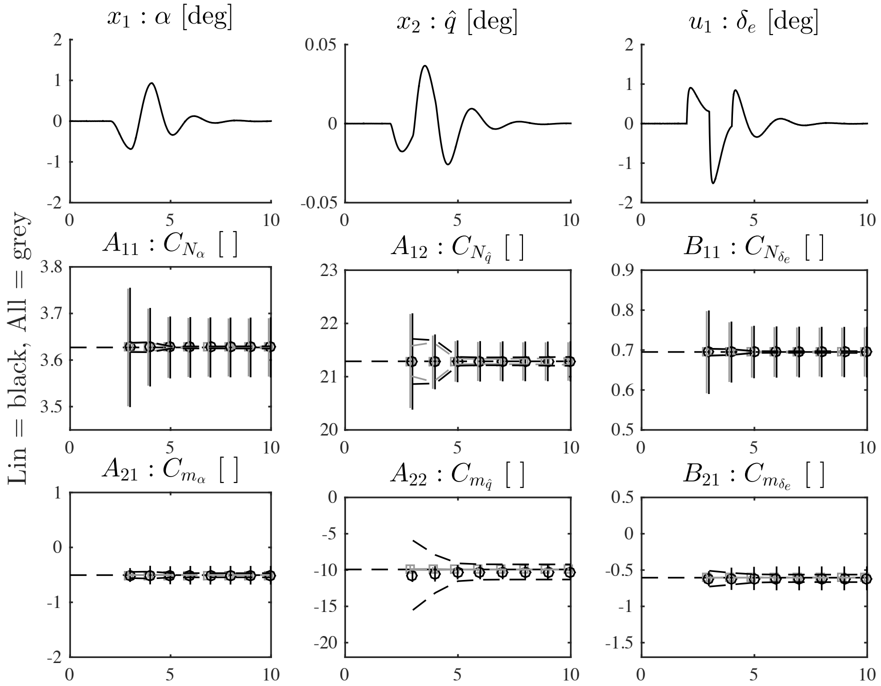


Figure 7.9: *Comparison between estimates with all data present (grey) and using the linear interpolation method (black) to handle missing data. The result of 200 MC simulations with six consecutive data frames randomly missing during excitation. The algorithm used is Larsson's algorithm with previously calculated coefficients. The circles/squares are the mean estimates over 200 simulations and the vertical lines are the mean of their estimated two standard deviations. The dashed black and grey lines are the two standard deviations based on the MC simulations. The straight horizontal dashed lines indicate the true parameters used in the simulation model.*

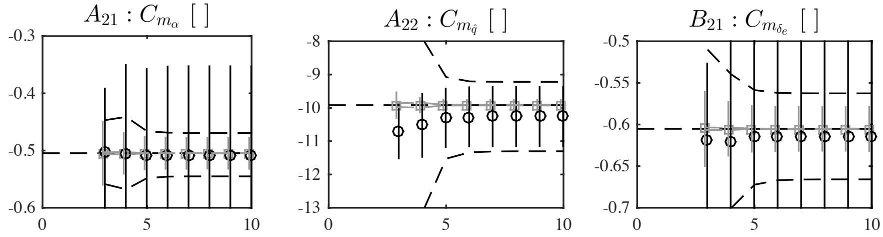


Figure 7.10: *A zoom of the moment coefficients presented in Figure 7.9. Estimation with all data present (grey) and using the linear interpolation method (black) to handle missing data.*

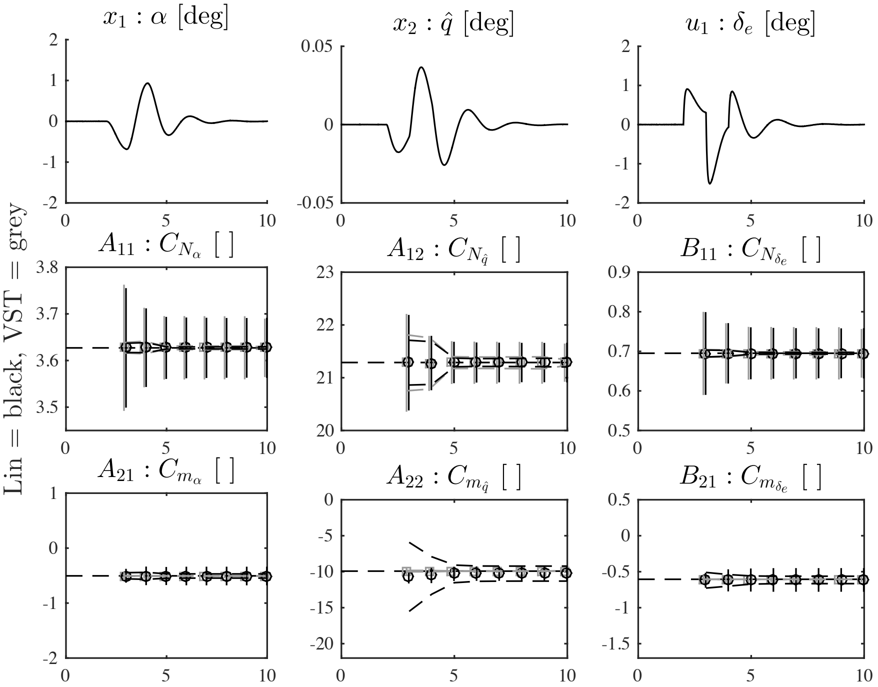


Figure 7.11: *Comparison between estimates using the linear interpolation method (black) and the VST method (grey) to handle missing data. The result of 182 MC simulations with six consecutive data frames randomly missing during excitation. The algorithm used is Larsson's algorithm with previously calculated coefficients. The circles/squares are the mean estimates over 200 simulations and the vertical lines are the mean of their estimated two standard deviations. The dashed black and grey lines are the two standard deviations based on the MC simulations. The straight horizontal dashed lines indicate the true parameters used in the simulation model.*

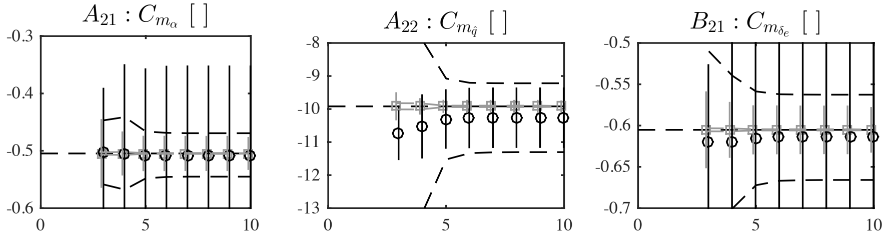


Figure 7.12: *A zoom of the moment coefficients presented in Figure 7.11. Estimation using the linear interpolation method (black) and the VST method (grey) to handle missing data.*

8 The Final Implementation

The final implementation choices among the alternatives presented in Chapter 6 and Chapter 7 are presented in this chapter. The choices are made based on the results presented in the same chapters. Along with the motivations for these choices, a summary of the final implementation is presented. The chapter also contains the output from the program and its performance with respect to the time constraints.

8.1 Motivations

A conceptual program to investigate and illustrate the feasibility of the ideal program described in Section 5.1 is implemented. The implementation choices for this program are presented below.

- Larsson's algorithm is chosen as the estimation algorithm for the program since the simulation results indicate that it performs better than Morelli's.
- Already calculated coefficients using Arcoft are chosen as outputs in the regressions since not all the parameters that are required to calculate the coefficients within the algorithm are available. The study of the differentiation FIR filter also shows that the expected results when using already calculated coefficients are not worse than when differentiation within the algorithm is used. When data is missing in the estimation, the FIR filter differentiation might even give better results.
- The VST method is chosen for the handling of the missing data. Based on the results using simulated data, the VST method is the method of choice when previously calculated coefficients are used. Additionally, the VST method

is preferable from a time performance and implementation point of view. Contrary to the linear interpolation method, the VST method requires no extra calculations in addition to the calculation of the distance between samples. This makes the algorithm more easily implemented, the code more readable and the execution-time shorter.

- Only the regression for the roll moment coefficient C_l is implemented. Since the implementation of the different regressions is analogous, only with different dependent variables, time has not been spent on implementing all of them. However, all the required signals for the remaining coefficients are loaded and initialised as a first preparatory step towards a final program. The roll moment coefficient C_l is chosen since the studied data set comes from a roll manoeuvre.
- The implementation has been done for the simulation mode of StellaRT only, given the limitations of this work. In the recording of real-time data, all parameters are synchronised to a common sample frequency of 60 Hz. The program does, therefore, not include any synchronisation, filtering or resampling of any parameters.
- No reading of the reference derivatives from the aero data model is implemented since ROMAC cannot return additional parameters. The reference derivatives for the specific case studied have been taken manually from the aerodynamic model using a first order Euler approximation. This approximation is used since it is the simplest alternative. The reference derivatives are then hard-coded into the program in order to get the desired appearance of the output plot.

8.2 Summary of Algorithm

The final algorithm is implemented in Matlab within the program StellaRT. New data from VuSoftNT is loaded into the workspace through StellaRT in batches each time new data arrives. The outputs and regressors are transformed sequentially using the CZT each time new data arrives and the entire batch is added at once.

The outputs used are the coefficients calculated by the Arcoft script. Each second, or as close to as possible, depending on when the data arrives, a parameter estimate is computed. The parameter estimation method used is Larsson's algorithm. This method uses the batch CLS algorithm and IV. The improved differentiation of Larsson's method is not used since the calculated coefficients of Arcoft are used instead. Since the calculated coefficients are used, a compensation for their delay is done within the program. The parameters from the ROMAC parallel simulation are used as instrumental variables and interruptions in the data stream are handled by using the VST method. Since ROMAC does not handle disruptions, this is applied to the simulated signals as well as the flight test signals.

The transform and estimator are implemented for the roll moment coefficient C_l

only. However, all initialisations of parameters needed for the remaining coefficients are implemented as well. The implementation also include one real-time plot of all states and inputs present in the aerodynamic model and one of the six estimated derivatives of C_l. This plot includes the estimates' estimated standard deviations and the reference derivatives taken from the model. The reference derivatives of the specific studied data set are hard-coded into the algorithm.

8.3 Results Using Recorded Real-Time Flight Test Data

The output from the implemented real-time program is presented in this section. The estimates are presented in Figure 8.1 and the states and inputs are given in Figure 8.2. In this data set, telemetric disruptions occur four times during excitation and a total of 47 samples are missing.

The algorithm succeeds in performing parameter estimation on real-time data. As expected, the estimates are not as smooth as on complete, post-processed flight test data. The variations of the estimates and the standard deviations are larger, especially before excitation. It is clearly visible when the excitation starts and

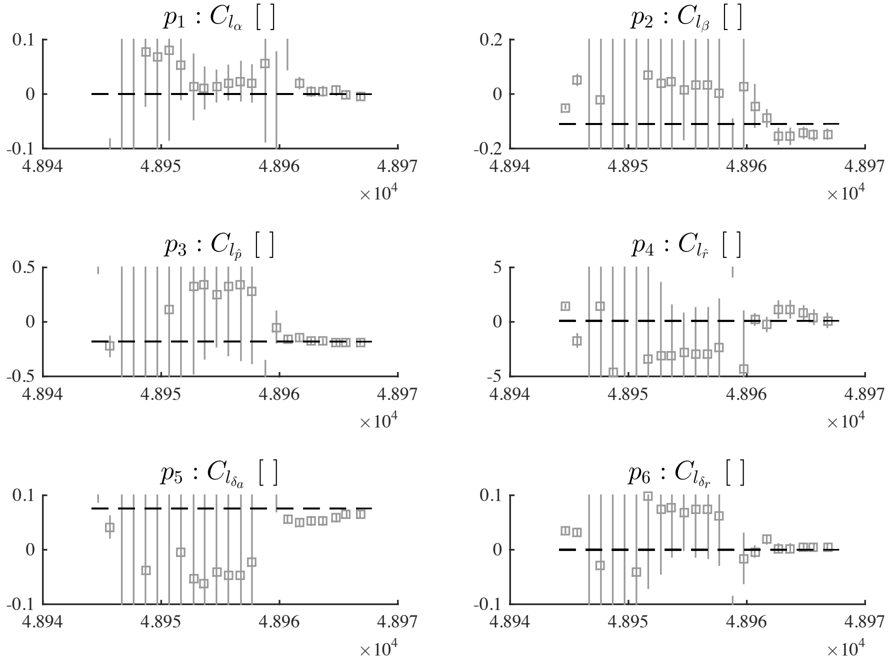


Figure 8.1: *The resulting estimate by the real-time program. The squares indicate each new estimate and the vertical lines are the estimated standard deviations corresponding to each estimate.*

when enough data to make a good estimate is present. Conclusions as to whether the model parameters appear to be conservative or not from a stability perspective might also be drawn from the results. Thanks to the clear distinction between the quality of the estimates before and after excitation, the real-time program appears more practically useful than what is indicated using post-processed data.

8.4 Time Constraints and Performance

Real-time for this application is defined to mean a maximum delay of a couple of seconds, regardless of the length of the experiment. This means that the initial delay of the real time program cannot exceed a couple of seconds and that the execution-time on average has to be shorter than the time between arrivals of new data. This means that there is no additional delay build-up dependent on the length of the data set. Since the data is transmitted in 16 Hz, the average execution time needs to be shorter than 1/16 = 0.0625 seconds. In reality, several frames might arrive simultaneously, making this time constraint longer.

The initial delay of the program is caused by two factors. The first is the 0.5 seconds delay of the signals caused by the calculations in ROMAC. The second is the 35 sample delay of the calculated coefficients. This results in a delay of about 0.5 seconds as the program has to wait before the coefficient values arrive. In total, the initial delay of the program is just above one second.

A typical execution time of the program run over the entire data set is presented in Table 8.1. As can be seen, the execution time of the calculations is well within the limit and it is the plotting that take the longest time. Even with the two plots of the final implementation, however, the mean execution time is within the limit even though the maximum time is too long. When the time of one execution is longer than the time between arrivals of new data, the new data will be buffered making the next batch to process bigger. Since the implementation of the transform adds the entire batch in one command, regardless of size, this has little negative effect on the execution time. Therefore, the execution time might even be longer than 0.0625 seconds and still not result in a build-up delay of the program.

Table 8.1: *The execution time of a typical run.*

Execution time	**With plots**	**Without plots**
Mean	0.054s	0.0024s
Max	0.15s	0.011s
Min	0.021s	0.0010s

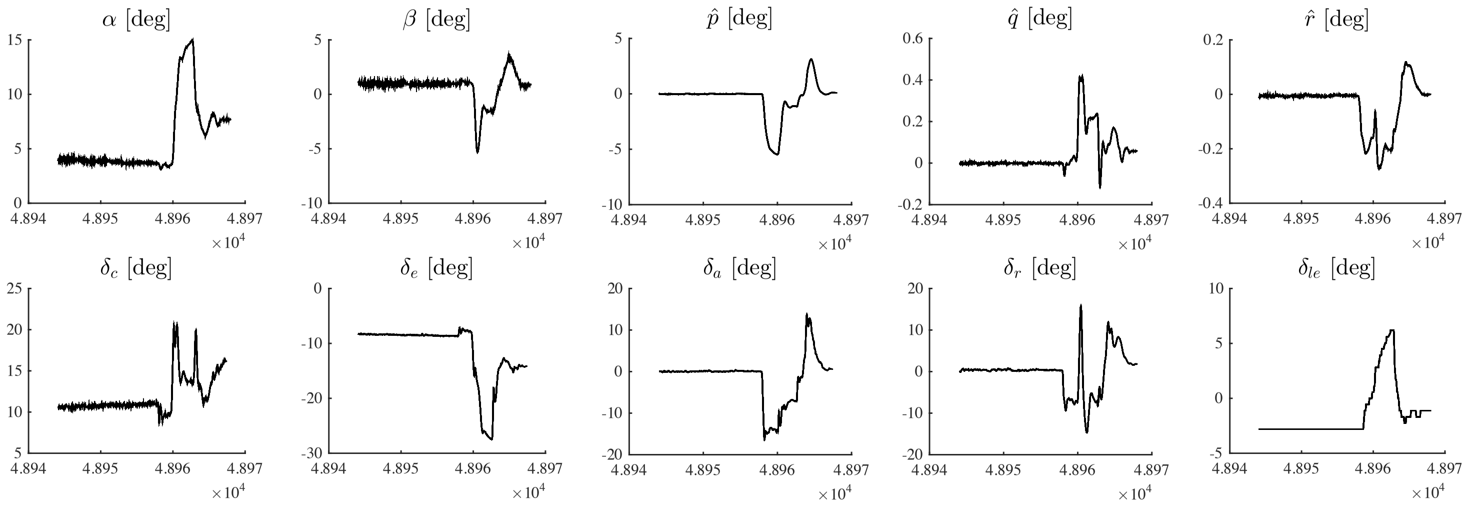


Figure 8.2: *The states and inputs as plotted by the real-time program.*

9 Discussion

The results obtained and presented throughout this work are discussed in this chapter.

9.1 Aerodynamic Model

The aerodynamic model used in this work contains all the relevant and available dependent variables as discussed in Section 3.3.2. The air brake and the landing gear, for example, are not possible to include with the current measurement equipment on Gripen. If, in the future, these are included, the model can easily be modified to take them into account.

Some model inputs have a very high correlation. This means that they can be hard to separate in the estimation. To separate them, specific manoeuvres might be needed. This also means that knowledge of the inputs, derivatives and the specific studied manoeuvre is needed. Some caution when interpreting the results of derivatives that cannot be properly separated is also required.

9.2 Real-Time Parameter Estimation Algorithm

The results when comparing Larsson's and Morelli's algorithms are comparable to those found in Larsson (2013) for the state space model. The corrected differentiation appears to result in better estimates at the beginning of the excitation. This is explained in Larsson (2013) to be caused by the fact that the added last sample becomes a correction term for the CZT. The presence of the IV in the algorithm improves the estimates by reducing the bias for high noise levels. Also this result is analogous to what was found in Larsson (2013). The result is, however, only shown using simulated data in this work since only one data set of real data was available. Since this real data has low noise levels, the effect cannot be studied.

There is a significant difference in the size of the estimated standard deviation before excitation using real-time data and post-processed data. The only obvious differences between the post-processed data and the recorded real-time data are the presence of telemetric disruptions in the real-time data and the low-pass filtering of certain signals in the post-processed data. Since the telemetric disruptions all occur during excitation, the filtering is assumed to be the reason for the difference. However, one single data set is not enough to draw any real conclusions.

When coefficients calculated using Arcoft are used as outputs, the VST method gives the best results when data is missing. However, these results are only shown on simulated data. On real data, the missing samples affect the remaining samples due to the filters. An additional study where the VST and the linear interpolation method are compared on real data is therefore needed to see if the results of the real-time program can be improved.

9.3 Real-Time Parameter Estimation Program

All aspects of the ideal program are not reached in this work for various reasons, but the main concept is still shown. The main reasons why some goals are not reached are constraints posed by, and limitations in, the surrounding software. With the information presented in this work of what these constraints and limitations are, these can be solved in the future. These changes, however, require modifications of software that affect many other operations and cannot be performed within the scope of a master's thesis work. This work can be continued when a new version of, for example, ROMAC is available.

Even though a final program will be able to function as decision support when deciding to continue or abort the flight from a safety or data collecting perspective, the importance of experience and savoir-faire of the operators shall not be underestimated. Based on the results of this thesis and the number of aspects to take into account when assessing the quality of the estimates, the judgement of the operator is needed to draw the right conclusions and it is unlikely that such conclusions can be automated.

As previously explained, the linear model is a Taylor expansion where all higher order terms are neglected. This means that it is only valid for small deviations from the trimmed condition. This, in turn, means that the real-time estimation program will only function on certain manoeuvres. In today's flight tests, far from all manoeuvres are pulses where the flight condition is aimed to be kept constant. The different manoeuvres serve different purposes. For example, a long manoeuvre spanning over several heights and speeds helps the pilot feel the behaviour and manoeuvrability of the aircraft. His or her assessment is also important in the overall result of the flight test. Also during these manoeuvres, monitoring of the signals is important. A program such as this could therefore not change the flight test manoeuvres or replace existing software for monitoring, but serve as a compliment. For specific studies using this software, however, special manoeuvres that are well suited might be performed. An example of such a manoeuvre is when an automatic pulse, or other manoeuvre, is used instead of a pilot command. This creates a more ideal manoeuvre than what the pilot can create and would be useful for this type of study.

10 Conclusions

The goal of this thesis was to produce a program that uses either Larsson's or Morelli's algorithm to perform real-time parameter estimation. The two methods should be implemented and compared on both simulated and real flight test data. Additionally, the final program should handle telemetric data. Several alternative methods for handling of missing data should be chosen and analysed on simulated data. Based on these results, the final program should be implemented and evaluated on telemetric data.

All of these goals have been reached within this work. The conclusion is that Larsson's algorithm performs better than Morelli's and that it might be used for real-time parameter estimation together with the VST method for handling of missing data. For this application, real-time means a maximum delay of a couple of seconds. Estimating aerodynamic derivatives in real-time during flight testing could reduce the number of flights. Reducing the number of flights is positive both from a financial and an environmental perspective as they are both costly, time consuming and fuel consuming.

The final program uses the VST method to handle missing data, ensuring functionality on recorded telemetric data. The VST method was compared to three other methods, namely discard, hold and linear interpolation. Given the other implementation choices made, the VST performed best among the compared methods. None of the methods investigated on simulated data, the VST and linear interpolation, gave unbiased estimates. The final estimator used on telemetric data is therefore biased.

10.1 Future Work

In this work, a conceptual real-time estimation program has been implemented and it has been shown that the algorithms have the potential of working in a decision support program for flight testing at SAAB. This is one step further, building on the works of Andersson (2013) and Larsson (2013), towards a final program. However, to continue this work and to reach the final goal, additional work is needed.

An update of ROMAC is needed to continue this work. This is a prerequisite both to implement the wanted functionality and to conduct further analysis. Since no additional parameters can be returned with today's version, the reference derivatives cannot be obtained in real time. The possibility to return these parameters as well as internal parameters needed for the calculation of the moment coefficients is needed. Additionally, a module within ROMAC needs to be implemented that can obtain, and return, the reference derivatives. In this work, a first order Euler approximation is used to calculate the reference derivatives. There might be other methods that are more sophisticated and better suited for this purpose.

If the internal parameters needed to calculate the moment coefficients were available, further investigations regarding the differentiation methods and the comparison between the VST and linear interpolation methods could be performed on real data. Given the results in this work, the VST performs better when calculated coefficients are used. When the differentiation can be performed in the algorithm using telemetric data, the linear interpolation method might perform better. Given this comparison, the effect of the missing data on the filters in ROMAC can also be studied.

A compensation for the delay caused by the filters is needed in ROMAC. Since the filters can be changed and modified, the delay is not necessarily constant and should not be hard-coded into the estimation algorithm. This hard-coded compensation also greatly complicates the implementation and deteriorates the readability of the estimation program. These are aspects that should be prioritised in a final program as they are important in the integration and maintenance of the software.

Further studies of methods for handling of missing data could be performed in order to find a final estimator that is unbiased. However, there might be a tradeoff between the quality of the estimates and the real-time suitability of the estimator. Even though a method for handling of missing data is available that gives unbiased estimates, it is not certain that it would work in this application.

For a final program that can be used as decision support, the estimation program must be extended to include all six aerodynamic coefficients. This would give a total of 36 estimated derivatives. It serves no purpose to study all of these derivatives at once for all manoeuvres. Additionally, the plotting has a negative affect on the execution time. Therefore, a good solution for the user to choose which plots are of interest, and to easily change the view of the plots, is needed.

The final appearance of the plots regarding size, visualisation and highlighting of different effects must also be decided in cooperation with the final users. The functionality in real time might become a challenge as more plots are added and investigations regarding different plotting alternatives might be needed.

For a future program to function not only on recorded real-time data, but on real telemetric data, signals of different sample frequencies must be handled. The synchronisation of the real-time data to a common frequency results in delays in the signals. These delays must be compensated for in such a way that the real-time functionality is ensured.

In order for the parallel simulation to better follow the flight path of the real aircraft, a pilot model or manual reset of the error to zero, is used (Andersson et al., 2002). These solutions, and especially the manual reset of the model states, distort the signals used as IV. It is not known how this affects the estimates. The manual error handling is mostly used between manoeuvres and one alternative could be to simply not allow any reset of the error during estimations. However, an analysis of the effect on the estimates might still be interesting.

Bibliography

K. Ahlstöm and J. Torin. Future architecture of flight control systems. *IEEE Aerospace & Electronic Systems*, 17:21–7, 2002. Cited on page 2.

I. Andersson. *Real-time Parameter Identification - Parameter Identification of Aerodynamic derivatives using Recursive Least Square Method*. Master's thesis, Department of Management and Engineering, Linköping University, Linköping, Sweden, 2013. Cited on pages 2, 7, and 78.

K. Andersson, M. Karlsson, and M. Staaf. Aerodynamic and flight dynamic real-time analysis during spin and carefree maneuvering tests of the SAAB JAS 39 Gripen. In *23rd Congress of ICAS, International Council of Aeronautical Sciences*, Toronto, Canada, September 2002. Cited on pages 35 and 79.

J. W. Cooley, P. A. W. Lewis, and P. D. Welch. The finite Fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 17:77–85, 1969. Cited on page 8.

B. Etkin and L. D. Reid. *Dynamics of flight - Stability and Control*. John Wiley & sons, Inc., New York, USA, third edition, 1996. Cited on pages 18 and 28.

G. Frawley and J. Thorn. *The International Directory of Military Aircraft*. Aerospace Publications Pty Ltd, Weston Creek, Australia, 1996. Cited on page 30.

A. Isaksson. Identification of ARX models subject to missing data. *IEEE Transactions on Automatic Control*, 38:813–819, 1993. Cited on page 16.

ISO. Standard atmosphere. ISO, 2533:1975. Cited on page 31.

H. Johansson and F. Ljungberg. Personal communication, 2015. Cited on pages 23 and 26.

M. Kastman. *ARES: Beskrivning av delmodell RIGID_BODY*. Internal document, SAAB AB, Linköping, Sweden, 2004. Cited on page 17.

V. Klein and E. A. Morelli. *Aircraft System Identification - Theory and Practice*. American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia, USA, 2006. Cited on pages 2, 3, 5, 7, 8, 10, and 11.

R. Larsson. *System Identification of Flight Mechanical Characteristics*. Licentiate thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2013. Cited on pages 2, 3, 5, 8, 10, 11, 12, 17, 35, 49, 51, 73, and 78.

L. Ljung. *System Identification - Theory for the User*. P T R Prentice Hall, Eaglewood cliffs, New Jersey, USA, second edition, 1999. Cited on pages 5, 6, 7, 12, and 14.

E. A. Morelli. *High Accuracy Evaluation of the Finite Fourier Transform Using Sampled Data*. Technical Manual, NASA, Hampton, Virginia, USA, 1997. TM-110340. Cited on page 8.

E. A. Morelli. Real-time parameter estimation in the frequency domain. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, AIAA-99-4043, Portland, Oregon, USA, August 1999. Cited on pages 9, 30, 31, and 35.

R. C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill, second edition, 1998. Cited on pages 18, 21, 28, and 31.

R. Pintelon and J. Schoukens. Identification of continuous-time systems with missing data. In *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference*, volume 2, Venice, Italy, May 1999. Cited on pages 13, 15, and 16.

R. Pintelon and J. Schoukens. *System Identification - A Frequency Domain Approach*. IEEE Press, Piscataway, NJ, USA, 2001. Cited on pages 5 and 12.

L. R. Rabiner, R. W. Schafer, and C. M. Rader. The chirp z-transform algorithm. *IEEE Transactions on Audio and Electroacoustics*, 17:86–92, 1969a. Cited on pages 8 and 9.

L. R. Rabiner, R. W. Schafer, and C. M. Rader. The chirp z-transform algorithm and its application. *The Bell Sytem Technical Journal*, 48:1249–1292, 1969b. Cited on page 8.

J. Roskam. *Airplane Flight Dynamics and Automatic Flight Controls*, volume 1. Design, Analysis and Research Corporation, Lawrence, Kansas, USA, 1995. Cited on pages 18 and 30.

T. Stavöstrand. *Aerodynamic Flight Test*. Internal document, Flight Test and Verification, SAAB Aeronautics, Linköping, Sweden, 2011. Cited on pages 2, 18, 21, 23, 24, 25, and 26.

B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, second edition, 2003. Cited on pages 29 and 31.

D. Sundararajan. *Discrete Fourier Transform: Theory, Algorithms and Applications*. World Scientific, River Edge, NJ, USA, 2001. Cited on page 8.

M. Svensson. Personal communication, 2015. Cited on page 13.

T. Söderström and P. G. Stoica. *Instrumental Variable Methods for System Identification*. Springer-Verlag, Berlin Heidelberg, Germany, 1983. Cited on page 12.

T. Söderström and P. G. Stoica. *System Identification*. Prentice Hall International (UK) Ltd, Hemel Hempstead, United Kingdom, 1989. Cited on page 6.

Internal document SAAB AB. *Stella Function Reference*. Linköping, Sweden, 2012. Cited on page 36.

R. Wallin. *Optimization Algorithms for System Analysis and Identification*. PhD thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2004. Cited on pages 13, 14, and 16.

R. Wallin, A. J. Isaksson, and L. Ljung. An iterative method for identification of ARX models from incomplete data. In *Proceedings of the 39:th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000. Cited on page 16.