Some results on closed-loop identification of quadcopters

Du Ho
This is a Swedish Licentiate's Thesis.

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree. A Doctor's Degree comprises 240 ECTS credits (4 years of full-time studies). A Licentiate's degree comprises 120 ECTS credits, of which at least 60 ECTS credits constitute a Licentiate's thesis.
To my family!
Abstract

In recent years, the quadcopter has become a popular platform both in research activities and in industrial development. Its success is due to its increased performance and capabilities, where modeling and control synthesis play essential roles. These techniques have been used for stabilizing the quadcopter in different flight conditions such as hovering and climbing. The performance of the control system depends on parameters of the quadcopter which are often unknown and need to be estimated. The common approach to determine such parameters is to rely on accurate measurements from external sources, i.e., a motion capture system. In this work, only measurements from low-cost onboard sensors are used. This approach and the fact that the measurements are collected in closed-loop present additional challenges.

First, a general overview of the quadcopter is given and a detailed dynamic model is presented, taking into account intricate aerodynamic phenomena. By projecting this model onto the vertical axis, a nonlinear vertical submodel of the quadcopter is obtained. The Instrumental Variable (IV) method is used to estimate the parameters of the submodel using real data. The result shows that adding an extra term in the thrust equation is essential.

In a second contribution, a sensor-to-sensor estimation problem is studied, where only measurements from an onboard Inertial Measurement Unit (IMU) are used. The roll submodel is derived by linearizing the general model of the quadcopter along its main frame. A comparison is carried out based on simulated and experimental data. It shows that the IV method provides accurate estimates of the parameters of the roll submodel whereas some other common approaches are not able to do this.

In a sensor-to-sensor modeling approach, it is sometimes not obvious which signals to select as input and output. In this case, several common methods give different results when estimating the forward and inverse models. However, it is shown that the IV method will give identical results when estimating the forward and inverse models of a single-input single-output (SISO) system using finite data. Furthermore, this result is illustrated experimentally when the goal is to determine the center of gravity of a quadcopter.
Acknowledgments

I would like to express my sincere thanks to my supervisor, Assoc. Prof. Martin Enqvist, for your valuable support and supervision. Your advice, guidance, encouragement, and inspiration have been invaluable over the years. My grateful thanks are also extended to my co-supervisor, Assoc. Prof. Gustaf Hendeby for your open mind, advice and valuable helps whenever I asked for assistance, both in research and LaTeX. Many thanks go to my coauthor, Dr. Jonas Linder for our successful collaboration. Thanks to the former head of the division Prof. Svante Gunnarsson for accepting me to join the group and Ninna Stensgård for organizing all administrative stuff that makes the group run smoothly.

Working at the Automatic Control group, ISY, Linköping University is really a pleasure. The warmest thanks go to my former and present colleagues for a stimulating and fun environment. I really enjoy fika breaks over the past three years in which we discuss, learn and grow. Sometimes I thought our fika breaks are more up-to-date than any newspaper. Special thanks to my roommate, Lic. Martin Lindfors for sharing the office, joining sauna and visiting Vietnam. Thanks to my former roommate Dr. Tohid Ardeshiri for helping me out when I just arrived in Linköping. Many thanks to the high probability group: Erik Hedberg, Måns Klingspor, Lic. Gustav Lindmark, Oskar Ljungqvist, Kristoffer Bergman, Per Boström-Rost, Christian Andersson Naesseth, Lic. Clas Veibäck, Daniel Arnestrom, Robin Forsling, and the heavy-tailed distributions group: Angela Fontan, Alberto Zenere, Lic. Yuxin Zhao, Lic. Parinaz Kasebzadeh, Lic. Kamiar Radnorsrati. Thanks to Shervin Parvini Ahmadi and Hamed Haghshenas for nice sports activities out of the office.

I wish to thank the European Union for financial support to the MarineUAS project in the Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 642153. I am also thankful to all professors, researchers, administrative staffs and PhD fellows in the MarineUAS project for the great times in all workshops and events.

Last but not least, I would like to thank my parents, my brothers, my wife Elena (Luu) Chu and my son Hai (Mit) Phong, for your constant love, encouragement and limitless support throughout my studies.

Linköping, November 2018
Du Ho
## Contents

Notation xi

1 Introduction 1
   1.1 Research motivation 1
   1.2 Goals 4
   1.3 Contributions 5
   1.4 Thesis outline 5

2 Models in system identification 7
   2.1 Linear time-invariant systems 7
      2.1.1 Rational transfer function model 7
      2.1.2 Linear regression model 9
      2.1.3 State space model 10
   2.2 Nonlinear systems 10
      2.2.1 Hammerstein models 11
      2.2.2 Wiener models 12
   2.3 Parameter transformations 13

3 Estimation methods 15
   3.1 Prediction error method 15
   3.2 Least-squares method 16
   3.3 Extended Kalman filter 17
   3.4 Instrumental Variable method 18
      3.4.1 Open-loop Refined Instrumental Variable method 18
      3.4.2 Closed-loop Refined Instrumental Variable method 21
   3.5 A comparison of closed-loop identification approaches 25

4 Estimation of forward and inverse models 31
   4.1 Introduction 31
   4.2 Problem formulation 32
      4.2.1 Forward model 32
      4.2.2 Inverse model 33
   4.3 On equivalence of the estimation of the forward and inverse models 34
### Contents

4.3.1 The basic IV method ............................................ 34  
4.3.2 The extended IV method ........................................ 37  
4.4 Simulation study .................................................. 40  

5 Quadcopter modeling ................................................. 45  
5.1 Quadcopter dynamics .............................................. 45  
5.1.1 Representation of rigid body motions ......................... 46  
5.1.2 Kinematic relations ............................................ 47  
5.1.3 Kinetic relations .............................................. 47  
5.2 Quadcopter submodels ............................................ 50  
5.2.1 Longitudinal model ........................................... 50  
5.2.2 Vertical model ............................................... 52  
5.3 Inertial measurement sensor ..................................... 53  
5.4 Pulse-width modulation ......................................... 54  

6 Estimation of the roll submodel .................................... 57  
6.1 Introduction .................................................... 57  
6.2 Approximate roll model of a quadcopter ......................... 58  
6.3 Numerical verification .......................................... 60  
6.3.1 Simulation study ........................................... 60  
6.3.2 Experimental results ....................................... 63  
6.4 Estimating the center of gravity of a quadcopter ............... 66  

7 Estimation of the vertical submodel ................................ 71  
7.1 Introduction .................................................... 71  
7.2 Approximate model of the vertical dynamics of a quadcopter ........................................... 72  
7.3 Numerical studies .............................................. 73  
7.3.1 Simulation study ........................................... 73  
7.3.2 Experimental study ....................................... 74  
7.4 Comparison with the horizontal dynamic model ............... 76  
7.5 Diagnosis and discussion ....................................... 77  

8 Conclusion ....................................................... 81  

A Aerodynamics of a quadcopter ..................................... 83  
A.1 Propeller aerodynamic .......................................... 83  
A.2 Other aerodynamic effects ..................................... 85  
A.2.1 Ground effect ............................................... 85  
A.2.2 Vertical descent ........................................... 86  

B Quaternion rotations .................................................. 87  
B.1 Quaternion rotations ........................................... 87  
B.2 Quadcopter kinematic and kinetic equations .................... 88  
B.3 Conversion between Euler angles and quaternions ............ 88  

Bibliography ........................................................ 91
Notation

Math symbols

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>Set of integer numbers</td>
</tr>
<tr>
<td>R</td>
<td>Set of real numbers</td>
</tr>
<tr>
<td>E</td>
<td>Expectation</td>
</tr>
</tbody>
</table>

Dynamical systems and System Identification

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{y}_{t</td>
<td>t-1} )</td>
</tr>
<tr>
<td>( \tilde{y}_t )</td>
<td>Filtered ( y_t )</td>
</tr>
<tr>
<td>( \theta )</td>
<td>Parameter vector</td>
</tr>
<tr>
<td>( \theta_c )</td>
<td>Parameter vector in continuous-time domain</td>
</tr>
<tr>
<td>( \theta_d )</td>
<td>Parameter vector in discrete-time domain</td>
</tr>
<tr>
<td>Z</td>
<td>Dataset</td>
</tr>
</tbody>
</table>
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D</td>
<td>Three dimensions</td>
</tr>
<tr>
<td>ARMAX</td>
<td>Autoregressive moving average with exogenous input</td>
</tr>
<tr>
<td>ARX</td>
<td>Autoregressive with exogenous input</td>
</tr>
<tr>
<td>AsCov</td>
<td>Asymptotic covariance</td>
</tr>
<tr>
<td>BJ</td>
<td>Box-Jenkins</td>
</tr>
<tr>
<td>CoG</td>
<td>Center of gravity</td>
</tr>
<tr>
<td>CoR</td>
<td>Center of rotation</td>
</tr>
<tr>
<td>CL</td>
<td>Closed-loop</td>
</tr>
<tr>
<td>DC</td>
<td>Direct current</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic control unit</td>
</tr>
<tr>
<td>EIV</td>
<td>Errors-in-variables</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman filter</td>
</tr>
<tr>
<td>EMF</td>
<td>Electromotive force</td>
</tr>
<tr>
<td>ESC</td>
<td>Electronic speed controller</td>
</tr>
<tr>
<td>FDI</td>
<td>Fault detection and isolation</td>
</tr>
<tr>
<td>FTC</td>
<td>Fault tolerant control</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global navigation satellite system</td>
</tr>
<tr>
<td>IV</td>
<td>Instrumental variable</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial measurement unit</td>
</tr>
<tr>
<td>LPV</td>
<td>Linear parameter-varying</td>
</tr>
<tr>
<td>LS</td>
<td>Least-squares</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear time-invariant</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-electro-mechanical systems</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple-input multiple-output</td>
</tr>
<tr>
<td>MISO</td>
<td>Multiple-input single-output</td>
</tr>
<tr>
<td>OE</td>
<td>Output error</td>
</tr>
<tr>
<td>OL</td>
<td>Open-loop</td>
</tr>
<tr>
<td>PEM</td>
<td>Prediction error method</td>
</tr>
<tr>
<td>PI</td>
<td>Proportional, integral (controller)</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse-width modulation</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root mean square error</td>
</tr>
<tr>
<td>SISO</td>
<td>Single-input single-output</td>
</tr>
<tr>
<td>SLAM</td>
<td>Simultaneous localization and mapping</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned aerial vehicle</td>
</tr>
</tbody>
</table>
Introduction

In this work, the problem of estimating characteristics of a quadcopter based on measurements available from the onboard sensors is studied. Models of particular subsystems are derived since the entire model of the quadcopter is too complex. Moreover, the unstable and underactuated quadcopter performs its maneuvers under closed-loop control, which complicates the estimation algorithms.

1.1 Research motivation

System identification aims to use noise-corrupted input-output observations of a system to obtain mathematical models of the system dynamics. The mathematical models help to understand the system more deeply and to simulate or predict behaviors of the system with respect to different inputs without performing experiments on the real system. Another benefit of using mathematical models is for analysis and design. For instance, properties of the system such as unstable poles or nonminimum phase zeros can be extracted and a controller can be designed based on these features to ensure the performance of the system. We could also use the model for fault detection and isolation.

A mathematical model of a system can be obtained if the physical laws governing the system dynamics are known. The resulting model is called a white-box model and all parameters and variables are basically known physical entities and quantities. However, it is common that these physical entities and quantities are unknown due to a lack of system knowledge. In this case, a black-box model can be identified based on the system input-output measurements without any insights about the physical system. A grey-box model, in many practical cases, is used when partial prior knowledge of the system is available, i.e., the system structure and order. The unknown parameters are then estimated from measured input-output data using system identification algorithms.
System identification is often carried out iteratively. Firstly, the data is collected using an optimal experiment design where the excitation signals are selected. Other choices in this stage include the choice, for instance, of sampling time and which inputs/outputs to be measured. The second step is the model structure selection, i.e., the model structure with the suitable model order. Based on these model structures and measured data, system identification methods can then be applied to estimate the model parameter in the sense of minimizing a loss function or optimizing a criterion. After obtaining the estimated model, model validation methods using the second dataset can be performed to compare the estimated and measured outputs. If the difference between these outputs is small enough according to some criterion, the model is valid and can be used for special purposes. Otherwise, the identification procedure should be repeated until satisfactory results are obtained.

In order to succeed, the dynamic systems have to be excited to generate adequately informative data. This can be performed in an open- or closed-loop setup. The input in the open-loop experiment is varied freely while the output will be fed back to the input by means of some feedback mechanism in the closed-loop situation, as in Figure 1.1. Some reasons to perform a closed-loop experiment are that the dynamic system is unstable or must be controlled for economic or safety reasons [Forssell and Ljung, 1999]. Another reason is that the feedback is inherent in the system and cannot be affected by the user. A challenge in the closed-loop setting is that the input signal typically correlates to the process and measurement noises. Due to the presence of correlations, many identification methods which work well in the open-loop setting fail in providing consistent estimates.

The considered dynamic system in this thesis is the quadcopter, which is an inherently unstable and underactuated system [Hua et al., 2013, Mahony et al., 2012]. A quadcopter is a small unmanned aerial vehicle (UAV) that uses four symmetrically placed rotors. The propellers have fixed pitch and are arranged in counter-rotating pairs, which gives the quadcopter a simpler mechanical structure and easier maintenance than a conventional helicopter. Each rotor produces a thrust and a torque, which combined create the main thrust, and the roll, pitch and yaw torques. Therefore, due to the high degree of freedom, quadcopters have the ability to perform quick and complex maneuvers, e.g., aggressive flight maneuvers, such as spins and flips [Lupashin et al., 2010] and dancing in the air [Dinh et al., 2017].
Because of these two main advantages, the quadcopter has become a standard platform in the robotics research society and for general public uses. This leads to a rapid growth of the commercial drone market and there are now a number of available products such as the Parrot AR Drone [Parrot, SA], Yuneec Typhoon [Yuneec], 3D Robotics Solo [3D Robotics, Inc], DJI Mavic Pro [DJI], and DJI Phantom 4 [DJI], see Figure 1.2. At the same time, many research groups and university labs around the world are currently developing methods to explore the capabilities of the quadcopter and potentially commercialize new industrial applications, for instance, for surveillance, search and rescue [Cai et al., 2010] and exploring and mapping 3-D environments [Bills et al., 2011, Fraundorfer et al., 2012]. This seems to be caused by the recent advancements in numerous technologies, i.e., new materials (carbon fiber), designing and manufacturing (3D sketching, laser cutting or 3D printing), hardware (lithium-polymer batteries, brushless DC electric motor), micro-electro-mechanical systems (MEMS) sensors, orientation and positioning systems (global navigation satellite systems (GNSS) or camera-based localization), and algorithms (adaptive system identifi-
As already mentioned, system identification methods enable advanced control design, based on accurately estimated models, to enhance the performance of the whole quadcopter system. However, when confronted with a quadcopter whose dynamics need to be identified, there are a number of questions that should be answered. One question is which signals are to be considered as outputs and which are to be considered as inputs [Ljung, 1999]. The standard framework for system identification includes the notion of an input signal that enters a well-defined system and produces an output signal. The system and thus also the output signal are usually affected by noise but the input signal is typically assumed to be known exactly. Under these assumptions, it is most common to estimate a model from the input to the output of the system and most system identification methods are designed like this.

However, the standard system identification assumptions are not always satisfied. For example, in applications, the inputs might be unknown and should be measured and these measurements are affected by measurement noises. Some reasons are the infectious disease transmission modeling or fault detection and diagnosis that leads to an errors-in-variables (EIV) problem [Söderström, 2007]. Furthermore, the system might be more complex than in the standard framework such that there is still a dynamical relation between the available signals but no clear distinction between input and output. An example is the identification of a control-oriented multivariable model of the structural response of a helicopter where several sensors are located in different places to measure vibrations [Lovera et al., 2015].

Other industrial applications where the input/output selections are not obvious can be found in electrical power grids, communication systems, process industry applications, and biochemical reactions. These complex systems are modeled as dynamic networks which are often too complex to be estimated as a whole. Instead, estimating a particular part of the network is computationally cheaper. Approaches to obtain consistent estimators of a network subsystem have been studied intensively [Chiuso and Pillonetto, 2012, Dankers et al., 2015, Linder and Enqvist, 2017a,b, Van den Hof et al., 2013, Weerts et al., 2015].

1.2 Goals

The work has two primary goals. The first one is to explore the possibilities of using system identification methods to estimate the parameters of interesting submodels of a quadcopter. The main idea is to estimate linear time-invariant (LTI) models to approximate the nonlinear system. However, the LTI model can in some cases not describe the system behavior in a large enough operating region and it is then reasonable to use nonlinear models. This results in a need to estimate the parameters of a vertical Hammerstein model of a quadcopter. Secondly, the theoretical aspects of the instrumental variable method regarding forward and inverse model estimation is also considered.
1.3 Contributions

This work comprise three main contributions. Firstly, a general dynamic model of a quadcopter is derived in Chapter 5, considering the aerodynamic characteristics. If this model is linearized with respect to a single axis a submodel is obtained. The first contribution is to investigate possibilities to perform battery and weight diagnosis by detecting changes in the parameters of a nonlinear submodel. Simulations and experimental results are described in Chapter 7, which shows that the IV method provides more accurate estimates of the parameters than some common methods. The second contribution, in Chapter 6, is to consider a sensor-to-sensor problem where only measured input-output data is used. The third contribution is to investigate the use of the IV method in complex settings where it is not obvious whether the measured signal $u_t$ or $y_t$ should be viewed as the input. Theoretical analysis and simulation studies are presented in Chapter 4.

The results have also been published in the following articles:


Du Ho and Martin Enqvist. On the equivalence of forward and inverse IV estimators with application to quadcopter modeling. In the 18th IFAC Symposium on System Identification, Stockholm, Sweden, July 9-11 2018.

1.4 Thesis outline

The thesis can be divided into two parts. The first part provides the theoretical foundation of the system identification that is relevant to the applications later on. First, Chapter 2 lists some models that will be used in this work. The linear time-invariant model and its one-step-ahead prediction are given. Moreover, the nonlinear block-oriented models which are combinations of static nonlinear and linear blocks are also discussed and a literature review on identification of nonlinear block-oriented models is given. Chapter 3 presents some estimation methods that can be used with the aforementioned models. The consistency and asymptotic covariance are also discussed. Chapter 4 shows the equivalence of the IV method when the forward and inverse linear time-invariant models are estimated. This identity property of the forward/inverse estimates is useful in many application since the question of the input/output selection is critical.
The application part starts with Chapter 5 where the modeling of a quadcopter is presented and useful submodels are derived. Chapter 6 deals with the details of estimating the coefficients of a quadcopter based on a linear roll model derived in Chapter 5 while a Hammerstein model is estimated in Chapter 7. Finally, conclusions and future work concerning the previously presented topics are given in Chapter 8.
In this chapter, the objective is to provide a brief description of linear and nonlinear models. These models can be written in the continuous-time or discrete-time domain. However, since the input-output data is collected in the discrete-time domain, the discrete-time models are mainly considered. Note that the parameters of the continuous-time model can be derived from those of the corresponding discrete-time model.

2.1 Linear time-invariant systems

2.1.1 Rational transfer function model

A causal, linear time-invariant (LTI) system can be completely described by its impulse response $g_t$, $t = 1, 2, \ldots$. For a given input $u_t$, the output is given by

$$y_t = \sum_{k=0}^{\infty} g_k u_{t-k}, \quad t = 1, 2, \ldots \tag{2.1}$$

Introducing the time shift operator $q$, i.e., $q y_t = y_{t+1}$, the linear system output can be written as

$$y_t = G(q)u_t \tag{2.2}$$

where $G(q) = \sum_{k=1}^{\infty} g_k q^{-k}$ is the transfer function. However, in reality, the output cannot be measured exactly. There are always unobservable disturbances affecting the system. The disturbance is often assumed to be zero mean and generated by passing white noise $e_t$ through an inversely stable noise filter $H(q)$. The complete linear system is given by

$$y_t = G(q)u_t + H(q)e_t \tag{2.3}$$
Models in system identification

Figure 2.1: The general linear time-invariant model.

Although the system is uniquely determined by its impulse response, it is not always a practical way to work with it. An alternative is to characterize the models $G(q)$ and $H(q)$ by a finite number of parameters. The parameters can be collected into parameter vectors $\vartheta$ and $\eta$ and the candidate set of models is given by

$$y_t = G(q, \vartheta)u_t + H(q, \eta)e_t$$

(2.4)

where the parameter vectors $\vartheta$ and $\eta$ belong to subsets of $\mathbb{R}^{n_G}$ and $\mathbb{R}^{n_H}$, respectively. Note that the transfer function $G(q, \vartheta)$ could be unstable and non-minimum phase. The structure of the model (2.4) is shown in Figure 2.1. For a given $\vartheta$ and $\eta$, the model can be used to predict the output of the system from present and past samples of input and output. The one-step-ahead predictor of $y_t$, denoted by $\hat{y}_{t|t-1}$ is

$$\hat{y}_{t|t-1} = H^{-1}(q, \eta)G(q, \vartheta)u_t + [1 - H^{-1}(q, \eta)]y_t$$

(2.5)

In principle, a common choice in (2.4) is to select the transfer function as a rational polynomial function where the numerator and denominator coefficients are parameters. Note that the orders of the numerator and denominator are not necessarily the same. The output of the linear system is

$$y_t = \frac{1}{A(q, \vartheta)}B(q, \vartheta)u_t + \frac{1}{A(q, \vartheta)}C(q, \eta)e_t$$

(2.6)

where $B(q, \vartheta)$, $A(q, \vartheta)$ and $F(q, \vartheta)$ are parameterized as

$$B(q, \vartheta) = b_0 q^{-n_k} + b_1 q^{-n_k-1} + \cdots + b_{n_b-1} q^{-n_k-n_k+1}$$

$$A(q, \vartheta) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_a} q^{-n_a}$$

$$F(q, \vartheta) = 1 + f_1 q^{-1} + f_2 q^{-2} + \cdots + f_{n_f} q^{-n_f}$$

with $\vartheta = [b_0 \ldots b_{n_b-1} \hspace{1em} a_1 \ldots a_{n_a} \hspace{1em} f_1 \ldots f_{n_f}]$, and the noise rational polynomials as

$$C(q, \eta) = 1 + c_1 q^{-1} + c_2 q^{-2} + \cdots + c_{n_c} q^{-n_c}$$

$$D(q, \eta) = 1 + d_1 q^{-1} + d_2 q^{-2} + \cdots + d_{n_d} q^{-n_d}$$
where $\eta = [c_1 \ldots c_n, d_1 \ldots d_{n_d}]$. However, in many applications, the model (2.6) is too flexible. Instead, special cases of it can be used by setting some of the polynomials to unitary. For instance, with $F(q, \vartheta) = D(q, \eta) = 1$, the autoregressive moving average with exogenous input (ARMAX) model

$$A(q, \vartheta)y_t = B(q, \vartheta)u_t + C(q, \eta)e_t \quad (2.7)$$

is obtained [Ljung, 1999]. By inserting (2.7) into (2.5), the one-step-ahead predictor is

$$\hat{y}_{t|t-1} = \frac{B(q, \vartheta)}{C(q, \eta)}u_t + \left[1 - \frac{A(q, \vartheta)}{C(q, \eta)}\right]y_t \quad (2.8)$$

It can be seen that the prediction is obtained by filtering the input $u_t$ and output $y_t$ through a filter with the denominator $C(q, \eta)$. However, the denominator $C(q, \eta)$ in (2.8) is typically unknown and needs also to be identified. Hence, the predictor is nonlinear-in-parameter, which complicates the estimation problem.

Finally, by setting $A(q, \vartheta) = 1$ the Box-Jenkins model is obtained

$$y_t = B(q, \vartheta)F(q, \vartheta)u_t + C(q, \eta)D(q, \eta)e_t \quad (2.9)$$

The one-step-ahead predictor of the Box-Jenkins (BJ) model (2.9) is

$$\hat{y}_{t|t-1} = \frac{D(q, \eta)B(q, \vartheta)}{C(q, \eta)F(q, \vartheta)}u_t + \left[1 - \frac{D(q, \eta)}{C(q, \eta)}\right]y_t \quad (2.10)$$

### 2.1.2 Linear regression model

Multiplying both sides of (2.8) with $C(q, \eta)$ yields

$$C(q, \eta)\hat{y}_{t|t-1} = C(q, \eta)u_t + \left[C(q, \eta) - A(q, \vartheta)\right]y_t \quad (2.11)$$

Adding $[1 - C(q, \eta)]\hat{y}_{t|t-1}$ to both sides of (2.11), the predictor of the ARMAX model can be rewritten as follows

$$\hat{y}_{t|t-1} = B(q, \vartheta)u_t + \left[1 - A(q, \vartheta)\right]y_t + \left[C(q, \eta) - 1\right][y_t - \hat{y}_{t|t-1}] \quad (2.12)$$

Introducing the prediction error

$$\epsilon_t = y_t - \hat{y}_{t|t-1} \quad (2.13)$$

and vector

$$\varphi_t = [-y_{t-1} \ldots -y_{t-n_a} \ u_{t-n_k} \ldots u_{t-n_k-n_b+1} \ \epsilon_{t-1} \ldots \epsilon_{t-n_c}]^T \quad (2.14)$$

Then (2.12) can be rewritten as

$$\hat{y}_{t|t-1} = \varphi_t^T \vartheta \quad (2.15)$$

where the present value of the one-step-ahead predictor of the ARMAX model depends on the known values of $u_t, y_t$ and its past values. Note that the past
values of the prediction error depend on the true parameter vectors $\vartheta$ and $\eta$ of
the system and noise models, respectively. A special case of the ARMAX model is
the ARX model, where the numerator of the noise filter in (2.7) is simply unitary
$C(q, \eta) = 1$. Therefore, the past values of the prediction error $\varepsilon_t$ in the regression
vector $\varphi_t$ of the ARX model are removed and the remaining elements in $\varphi_t$ are
known. In this case, the least-squares method can be applied directly to input-
output data to obtain a parameter estimate.

### 2.1.3 State space model

Another way to describe a linear time-invariant (LTI) system is with a state space
model structure. This model describes the relations between input, output and
state variables. In detail, the state variable $x_t$ is a vector whose elements are
called states. The output $y_t$ is a combination of the states and $\vartheta$ represents typical
unknown physical coefficients.

A general discrete-time linear state space model can be written

$$
x_{t+1} = A(\vartheta)x_t + B(\vartheta)u_t + w_t \tag{2.16a}
$$

$$
y_t = C(\vartheta)x_t + D(\vartheta)u_t + v_t \tag{2.16b}
$$

where $A(\vartheta), B(\vartheta)$ and $C(\vartheta)$ are matrices with appropriate dimensions, $y_t \in \mathbb{R}^{n_y}$ is
the sampled output, $x_t \in \mathbb{R}^{n_x}$ is the state vector at time instant $t$, $u_t$ is the sampled
input, $w_t$ and $v_t$ are the process and measurement noises, respectively.

The discrete-time state space model can be transformed into a transfer func-
tion model structure

$$
y_t = G(q, \vartheta)u_t + H(q, \eta)v_t \tag{2.17}
$$

where $G(q, \vartheta) = C(\vartheta)[qI - A(\vartheta)]^{-1} B(\vartheta) + D(\vartheta)$ and $H(q, \eta)v_t = C(\vartheta)[qI - A(\vartheta)]^{-1} w_t + v_t$. Note that if the measurement and process noises are white, the parameter vec-
tor $\eta$ in $H(q, \eta)$ will be parametrized partly in common with those in $G(q, \vartheta)$.

### 2.2 Nonlinear systems

Linear models of dynamic systems are derived based on a strong assumption: the
underlying physical process behaves similarly throughout the operating area of
interest. However, most physical systems, for example, mechanical systems, UAV
systems and others are inherently nonlinear in nature. Therefore, a linear model
might not be able to produce simulated output that fits the measured output.
Furthermore, the performance of the model-based control cannot be improved as
the order of the linear model increases. Therefore, a nonlinear model is needed
to capture the dynamic behaviors of the system.

In order to adequately describe the nonlinear behavior of a system over a
larger range of the operating area, a nonlinear block-oriented model is often used.
The model is, in principle, a combination of linear dynamic and nonlinear static
2.2 Nonlinear systems

Figure 2.2: The Hammerstein system.

subsystems [Giri and Bai, 2010]. The linear dynamic subsystem can be characterized by the transfer function, finite impulse response, or state space representations. The nonlinear subsystems are usually assumed to be memoryless. These components are connected in different ways such as in series or in parallel in order to approximate the true system well [Schoukens and Tiels, 2017]. Two simple nonlinear structures are Wiener and Hammerstein models where two blocks are connected in series.

2.2.1 Hammerstein models

A Hammerstein system involves a dynamic linear time-invariant block followed by a static nonlinear one, see Figure 2.2. The nonlinearities of the system are accounted for in the first block while the second block describes the dynamics of the system. Hence, the Hammerstein structure is used to model systems where the static nonlinearity is at the input (actuator) of the systems according to

\[ x_t = f(u_t) \]
\[ y_t = G(q)x_t + H(q)e_t \]  

Although it is a simple structure, many practical systems can be described accurately by the Hammerstein model, such as chemical processes [Eskinat et al., 1991], power amplifiers [Kim and Konstantinou, 2001], solid oxide fuel cells [Jurado, 2006], and quadcopter subsystems [Ho et al., 2017b]. It is then natural that the identification of Hammerstein systems has attracted a considerable attention from the research society.

Over the years, a number of identification methods have been proposed in the literature. In [Risuleo et al., 2017], the impulse response of the linear subsystem is modeled as a realization of a zero-mean Gaussian process. The stable-spline kernel is used to represent the covariance matrix (or kernel) of this process. The linear dynamic system can also be described by a state-space representation and nonparametric kernel-based algorithms exist [Greblicki and Mzyk, 2009]. In [Wang et al., 2009], the Hammerstein model is estimated using a two-stage method. It is shown that the optimal solution to the weighted nonlinear least-squares problem is obtained. Other interesting methods are the separable least-squares method [Bai, 2002, Westwick and Kearney, 2001] and the subspace
method [Wang et al., 2014]. Note that these methods are based on the measured input-output data. If the input is missing the system's unknown information can also be retrieved from its output only [Bai and Fu, 2002, Vanbeylen et al., 2008]. Moreover, many works have been done to estimate MIMO (multiple-input multiple-output) Hammerstein systems offline [Goethals et al., 2005] and online [Chen and Chen, 2011, Mu and Chen, 2015].

### 2.2.2 Wiener models

The Wiener model is another type of block-oriented model. It describes a system where a linear time-invariant block precedes a static nonlinear one. The static nonlinear block basically represents sensor nonlinearities as well as other nonlinear effects at the output of the system. The mathematical description of a Wiener system is

\[
x_t = G(q)u_t
\]

\[
y_t = g(x_t) + H(q)e_t
\]  

(2.19)

Numerous identification methods have been proposed to estimate Wiener systems [Giri and Bai, 2010]. The linear part of the Wiener systems can be modeled as an impulse response function [Pawlak et al., 2007] or a state-space model [Lindsten et al., 2013]. In [Pawlak et al., 2007], the system nonlinearity is estimated by a pilot nonparametric kernel regression estimator. Moreover, a proposed approach is to select the order of the Wiener model using the output error cost function and a frequency domain algorithm is then applied to determine the model parameters [Zhu and Tai-Ji]. If the input is chosen as a stationary process the best linear approximation can be used to find the linear model of the nonlinear system in a sense of minimizing the mean-square-error cost function [Enqvist and Ljung, 2005]. This input can also be used to recursively estimate the Wiener system [Mu and Chen, 2013]. The missing input is considered in the blind identification algorithm [Vanbeylen et al., 2009]. Moreover, the nonlinear block is sometimes assumed to be invertible. In this case, if the nonlinear block is parametrized as a sum of known basis functions, an overparametrization and linear-in-parameter model will be obtained [Kalafatis et al., 1997, Ni et al., 2012]. The MIMO Wiener system case is considered in [Janczak, 2007, Westwick and Verhaegen, 1996].
Most of the methodologies in the literature consider measurement noises only at the output of the system. However, the process noise between the linear and nonlinear blocks can also be handled [Giordano and Sjöberg, 2016, Hagenblad et al., 2008, Lindsten et al., 2013, Wahlberg et al., 2014, 2015].

2.3 Parameter transformations

Physical modeling often leads to differential equations describing the rates of change of physical quantities. If the model is time-invariant, it is common to characterize the input-output behavior in terms of a transfer function. Specially, for an input $u_t$ and output $y_t$, the continuous-time transfer function is given as a function of a differential operator $p$ and parameter vector $\varphi_c$ as

$$G_c(p, \varphi_c) = \frac{b_1 p^{n_b-1} + b_2 p^{n_b-2} + \cdots + b_{n_b}}{p^{n_a} + a_1 p^{n_a-1} + \cdots + a_{n_a}}$$ (2.20)

where each coefficient $a_i, i = 1 \ldots n_a$ and $b_j, j = 1 \ldots n_b$ can be derived as a function of the parameter vector $\varphi_c$. The parameter vector $\varphi_c$ contains physical quantities such as mass, moment of inertia, center of gravity and so on. However, since all the measurements are taken in the discrete-time domain a discrete-time model is a natural way to relate these measurements to the dynamic system. To transform the original model $G_c(p, \varphi_c)$ into a corresponding discrete-time model $G_d(q, \varphi_d)$, several discretization methods can be used. The Euler transform $p = \frac{q^{-1}}{T}$ maps the left half plane in the continuous-time domain to an unbounded subset of the discrete-time domain while the bilinear transform $p = \frac{2}{T} \frac{q^{-1}}{q+1}$ transforms the entire left half plane of the continuous-time domain in the interior of unit circle in the discrete-time domain. The advantage of using the bilinear transform is that the discrete-time model always retains the stability properties of the system.

In order to reconstruct the continuous-time model, an inverse transformation is needed. Note that the discrete-time model is characterized by a parameter vector $\varphi_d$. However, even though the Euler and bilinear transforms are simple algebraic relations which are fairly accurate with short sampling time, the relation between the discrete-time parameter vector $\varphi_d$ and the continuous-time parameter vector $\varphi_c$ is nonlinear. Furthermore, the dimension of $\varphi_c$ may be higher than that of $\varphi_d$, which leads to an undetermined equation system, i.e., the degrees of freedom are higher than the number of equations. When all continuous-time parameters are estimated, the model fitted to the data is too flexible and a high variance of the estimated parameters is obtained [Ljung, 1999]. One approach to handle this issue is to use linear constraints based on mathematical manipulations involving known coefficients [Linder, 2014]. In this case, a subset of all parameters is estimated whereas the rest are assumed to be known. Other experiments, such as a tilting test to estimate the center of gravity, can be used to estimate the parameters assumed known. The result is a smaller computational burden and a reduced variance of the estimated parameters. In this thesis, the
opposite problem is experienced where the system is an overdetermined equation system. In detail, there are fewer parameters than the number of equations, which indicates that the dimension of $\hat{\vartheta}_d$ is higher than that of $\vartheta_c$. The nonlinear parametrization relations can be exploited without using constraints in this case. Since the estimate of the discrete-time parameter vector $\hat{\vartheta}_d$ gives information about the continuous-time parameter vector $\vartheta_c$, the estimate of $\vartheta_c$ can be obtained from the estimated mean $\hat{\vartheta}_d$ and covariance $\hat{P}_{\vartheta_d}$ by solving a nonlinear weighted least-squares problem [Gustafsson, 2010]

$$\hat{\vartheta}_c = \arg \min_{\vartheta_c} \left[ \hat{\vartheta}_d - \vartheta_d(\vartheta_c) \right]^T \hat{P}_{\vartheta_d}^{-1} \left[ \hat{\vartheta}_d - \vartheta_d(\vartheta_c) \right]$$

(2.21)

and using Gauss approximation formula [Ljung, 1999] to estimate the covariance of $\hat{P}_{\vartheta_c}$ as

$$\hat{P}_{\vartheta_c} = \left[ \frac{\partial \vartheta_d^T}{\partial \vartheta_c} \hat{P}_{\vartheta_d}^{-1} \frac{\partial \vartheta_d}{\partial \vartheta_c} \right]^{-1} \bigg|_{\vartheta_c = \hat{\vartheta}_c}$$

(2.22)
Numerous useful estimation methods using either iterative or noniterative identification schemes have been proposed for system identification of linear dynamic systems with measured input-output data [see, for example, Ljung, 1999]. These methods are applicable in the time/frequency domains and open/closed-loop setups. In this chapter, several methods will be described and some of their properties are also given.

### 3.1 Prediction error method

For given input-output data $Z_t$, the Prediction Error Method (PEM) uses a parameterized model to predict the next output as

$$\hat{y}_{t|t-1} = f(Z_{t-1}, \vartheta) \quad (3.1)$$

where $\hat{y}_{t|t-1}$ is the one-step-ahead prediction output, $f$ denotes a function of the past input/output data and $\vartheta$ is a finite dimensional parameter vector of the model [Ljung, 1999].

A PEM estimates the parameter vector $\vartheta$ by finding the minimum distance between the predicted outputs and the measured outputs, i.e.,

$$\hat{\vartheta}_{PEM} = \arg \min_{\vartheta} V_N(\vartheta) \quad (3.2)$$

$$V_N(\vartheta) = \sum_{t=1}^{N} \ell(y_t - \hat{y}_{t|t-1}) = \sum_{t=1}^{N} \ell(\epsilon_t(\vartheta))$$

where $\ell$ is a suitable distance measure, such as $\ell(\epsilon) = \|\epsilon\|^2$, and $N$ is the number of data samples.
The PEM is applicable for both open- and closed-loop systems. However, the nonconvex optimization problem requires a search routine that depends significantly on the shape of the loss function. In some cases, the search routine of the parameters may get stuck in local minima which are not useful. If $V_N(\vartheta)$ is a positive quadratic function, there are no local minima and the search method can guarantee global convergence. However, this is rarely encountered in practice. A simple way to counteract this issue is to let the optimizer run based on a number of different starting points and hope that one of them results in the global optimum.

Moreover, if the model is assumed to be in the model set, i.e., there exists a $\vartheta$ such that $G(q, \vartheta) = G_0(q)$ and $H(q, \vartheta) = H_0(q)$ and the input is persistently exciting, the PEM will result in an asymptotically unbiased estimate which has minimal asymptotic variance under some mild assumptions [Ljung, 1999] as

$$\sqrt{N}(\hat{\vartheta}_{PEM} - \vartheta) \sim N(0, P) \quad \text{as} \quad N \to \infty \quad (3.3)$$

The asymptotic variance of $\hat{\vartheta}_{PEM}$ is given as

$$P = \lambda_0 [\mathbb{E}\psi_t(\vartheta)\psi_t^T(\vartheta)]^{-1} \quad (3.4)$$

where $\lambda_0$ is the variance of the driving stochastic noise, $\mathbb{E}$ is expectation and the vector

$$\psi_t(\vartheta) = -\frac{d\epsilon_t(\vartheta)}{d\vartheta} \quad (3.5)$$

is the gradient of the prediction error evaluated at the true parameters.

### 3.2 Least-squares method

The PEM method (3.2) can be significantly simplified in some cases. One such example is the ARX model. Consider a linear time-invariant system

$$y_t = G(q, \vartheta)u_t + H(q, \eta)e_t = \frac{B(q, \vartheta)}{F(q, \vartheta)} u_t + \frac{C(q, \eta)}{D(q, \eta)} e_t \quad (3.6)$$

where $e_t$ is zero mean white noise. The ARX model is obtained when $C(q, \eta) = 1$ and $D(q, \eta) = F(q, \vartheta)$. The linear system (3.6) can then be rewritten in a regression form as

$$y_t = \varphi_t^T \vartheta + w_t \quad (3.7)$$

where the regression vector $\varphi_t$ depends on the past values of the output as well as the past and current values of the input signals, $\vartheta$ is the true parameter vector and $w_t$ is the disturbance. Since the ARX model is linear-in-parameter, the one-step-ahead prediction is given by $\hat{y}_{t|t-1} = \varphi_t^T \vartheta$ and a least-squares (LS) estimate is obtained by minimizing

$$\hat{\vartheta}_{LS} = \arg \min_{\vartheta} \frac{1}{N} \sum_{t=1}^{N} \|y_t - \varphi_t^T \vartheta\|_2^2 \quad (3.8)$$
which has the analytical solution

$$\hat{\vartheta}_{LS} = \left[ \frac{1}{N} \sum_{t=1}^{N} \varphi_t \varphi_t^T \right]^{-1} \left[ \frac{1}{N} \sum_{t=1}^{N} \varphi_t y_t \right] = R^{-1}_{qq} f_{qq}$$

(3.9)

Note that the input is assumed to be persistently excited, which guarantees that $R^{-1}_{qq}$ is invertible. The estimated covariance matrix of the estimated parameters is given by

$$\hat{P}_{LS} = \hat{\sigma}^2 R^{-1}_{qq}$$

(3.10)

where $\hat{\sigma}^2$ is the estimated variance of the residual. The least-squares method, in principle, provides an unbiased estimate for an ARX model but it is in general not consistent when the underlying system is not in the model class.

### 3.3 Extended Kalman filter

An extended Kalman filter (EKF) can also be used to estimate the unknown parameters [Gustafsson, 2010]. For the state space model (2.16), the state vector can be extended as

$$x^a_t = \left[ x^T_t \ \vartheta^T \right]^T$$

(3.11)

which results in a new nonlinear state space model

$$x^a_{t+1} = f(x^a_t, u_t, v_t)$$

(3.12a)

$$y_t = h(x^a_t, u_t) + e_t$$

(3.12b)

The process and measurement noises are typically assumed to be zero-mean Gaussian processes with constant covariance $v_t \sim \mathcal{N}(0, Q)$ and $e_t \sim \mathcal{N}(0, R)$, respectively.

The EKF estimates the state $x^a_t$ sequentially by performing a time update and a measurement update. It means that a new estimate of the parameters is obtained at each time when a new measurement is available. In detail, the time update uses the model (3.12a) to predict the state in the next step according to

$$\hat{x}^a_{t+1|t} = f(\hat{x}^a_{t|t}, u_t, 0)$$

(3.13a)

$$P_{t+1|t} = A_t P_{t|t} A_t^T + G_t Q G_t^T$$

(3.13b)

where

$$A_t = \frac{\partial f}{\partial x^a} \bigg|_{x^a_t = \hat{x}^a_{t|t}, \ v_t = 0}$$

$$G_t = \frac{\partial f}{\partial v} \bigg|_{x^a_t = \hat{x}^a_{t|t}, \ v_t = 0}$$
The measurement update step uses the measurement model (3.12b) and the current measurement $y_t$ to correct the estimate of the state vector $x_t^a$ as

$$K_t = P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R)^{-1} \quad (3.14a)$$

$$\hat{x}_{t|t}^a = \hat{x}_{t|t-1}^a + K_t (y_t - \hat{y}_{t|t-1}) \quad (3.14b)$$

$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1} \quad (3.14c)$$

where

$$C_t = \left. \frac{\partial h}{\partial x} \right|_{x_t^a = \hat{x}_{t|t-1}^a}$$

and $\hat{y}_{t|t-1} = h(\hat{x}_{t|t-1}^a)$ is the one-step-ahead predictor of the output while $K_t$ is the Kalman gain [Gustafsson, 2010].

For initialization, one common choice is to set the state to zero and the augmented parameter vector to the value obtained from the least-squares method. The covariance matrix is also set to a large positive value.

### 3.4 Instrumental Variable method

#### 3.4.1 Open-loop Refined Instrumental Variable method

If the noise $w_t$ in the regression model (3.7) is either not white or correlated with the control signal $u_t$, the conventional least-squares estimator cannot provide a consistent estimate [Ljung, 1999, Young, 2012]. An alternative method to overcome this limitation is to use Instrumental Variable (IV) estimation which is a correlation-based method. In the most basic version, an instrument $\zeta_t \in \mathbb{R}^{n_a+n_b}$ is used to obtain vector-matrix equations

$$\left[ \sum_{t=1}^{N} \zeta_t^T \right] \vartheta - \left[ \sum_{t=1}^{N} \zeta_t y_t \right] = 0 \quad (3.15)$$

which has the closed-form solution

$$\hat{\vartheta}_{IV} = \left[ \frac{1}{N} \sum_{t=1}^{N} \zeta_t^T \right]^{-1} \left[ \frac{1}{N} \sum_{t=1}^{N} \zeta_t y_t \right] = R_{\zeta \vartheta}^{-1} R_{\zeta y} \quad (3.16)$$

The basic IV estimator provides an consistent parameter estimator ($\lim_{N \to \infty} \hat{\vartheta}_{IV} \to \vartheta$) under the following two conditions

C.1 $\mathbb{E}[\zeta_t^T q_t^T]$ is non-singular,

C.2 $\mathbb{E}[\zeta_t w_t] = 0$. 
However, the estimate (3.16) is not asymptotically statistical efficient, i.e., the estimated variance is not minimal [Young, 2012]. In order to achieve an optimal estimate, i.e., a unbiased and minimum variance estimate, a prefilter \( L(q) \) is used and (3.15) is rewritten in the alternative vector-matrix form

\[
\left[ \sum_{t=1}^{N} L(q)\zeta_t L(q)\varphi^T_t \right] \delta - \left[ \sum_{t=1}^{N} L(q)\zeta_t L(q)y_t \right] = 0 \tag{3.17}
\]

which has the analytical solution

\[
\hat{\delta}_{IV} = \left[ \frac{1}{N} \sum_{t=1}^{N} L(q)\zeta_t L(q)\varphi^T_t \right]^{-1} \left[ \frac{1}{N} \sum_{t=1}^{N} L(q)\zeta_t L(q)y_t \right] \tag{3.18}
\]

The dimension of the instrument \( \zeta_t \in \mathbb{R}^{n_\zeta} \) is identical to that of the regression vector \( \varphi_t \in \mathbb{R}^{n_\varphi} \). In some applications, an augmented instrument might be used where its dimension is selected as \( n_\zeta > n_\varphi \). In this case, an improved version of the basic IV method called the extended IV method is obtained, where the parameter vector \( \delta \) is estimated by solving

\[
\hat{\delta}_{IV} = \arg \min_{\delta} \frac{1}{N} \sum_{t=1}^{N} \| L(q)\zeta_t L(q)y_t - L(q)\zeta_t L(q)\varphi^T_t \delta \|^2_Q \tag{3.19}
\]

where \( \|x\|^2_Q = x^T Q x \), \( Q \geq 0 \) is a weighting matrix and \( L(q) \) is a stable prefilter.

One choice of the instrument is a pure lag of the input. Note that if \( u_t \) is persistently exciting and is uncorrelated with \( w_t \), the instrument variables will meet the two conditions above. However, the delayed lag cannot be chosen too large since it may result in \( \mathbb{E}[L(q)\zeta_t L(q)\varphi^T_t] \) becoming singular and hence the condition C.1 is not fulfilled [Ljung, 1999]. Another choice of the instrument is

\[
\bar{\zeta}_t = L(q)\left[ -\hat{y}_{t-1}, \ldots, -\hat{y}_{t-n_\varphi}, u_{t-n_\varphi}, \ldots, u_{t-n_\varphi-n_{\kappa}+1} \right]^T \tag{3.20}
\]

where the overline symbol \( \bar{\cdot} \) indicates the filtered signal and \( \hat{y}_t \) is computed as the output of the auxiliary model

\[
\hat{y}_t = \frac{\hat{B}(q)}{\hat{F}(q)}u_t \tag{3.21}
\]

in which \( \hat{B}(q) \) and \( \hat{F}(q) \) are the estimates of the \( B(q) \) and \( F(q) \) polynomials, respectively.

Since the standard implementation of the IV method including the basic and extended IV methods is an iteration procedure, \( \hat{B}(q) \) and \( \hat{F}(q) \) are updated iteratively. It indicates that the instrument depends on the parameters to be estimated. More specifically, at the \( j \)th iteration, the system model parameter vector \( \hat{\delta}_{IV}^{(j)} \) is estimated based on the estimate from the previous iteration \( \hat{\delta}_{IV}^{(j-1)} \).
Secondly, the choice of the prefilter \( L(q) \) is also important because it has a considerable effect on the covariance of the estimated parameters [Ljung, 1999]. If the true noise model structure is known, the covariance of the estimate can be minimized. It is well known that the Cramér-Rao inequality gives a lower bound for any unbiased estimate. If the parameter estimate has the normal distribution, the lower bound of the covariance matrix is [Ljung, 1999]

\[
P_{IV}^{opt} = \lambda_0 \left[ p \lim_{\delta \rightarrow 0} \frac{d\epsilon_t(\delta)}{d\delta} \frac{d\epsilon_t^T(\delta)}{d\delta} \right] 
\] (3.22)

where the gradients of the prediction errors evaluated at the true parameters are

\[
\frac{d}{d\delta} \epsilon(t, \delta) = \frac{d}{d\delta} \left[ \frac{1}{H(q, \eta)} \left( y_t - G(q, \delta)u_t \right) \right] = [F(q, \delta)H(q, \eta)]^{-1}q_t(\delta) 
\] (3.23)

which leads to the optimal choice of the prefilter

\[
L(q^{-1}) = \frac{1}{F(q^{-1})H(q^{-1})} 
\] (3.24)

and the weighting matrix \( Q = I \) where \( I \) is the identity matrix. The following example illustrates the benefit of using the optimal prefilter.

---

**Example 3.1**

Consider the open-loop system

\[
\dot{y}_t = \frac{1.0q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u_t \\
y_t = \dot{y}_t + \frac{1 + 0.7q^{-1}}{1 - 0.7q^{-1}} e_t 
\]

where \( u_t \) and \( e_t \) are zero mean white Gaussian processes with variance 1 and 0.25, respectively.

The open-loop IV method is applied with the optimal prefilter

\[
L_{opt}(q^{-1}) = \frac{1}{F(q^{-1})H(q^{-1})} = \frac{1 - 0.7q^{-1}}{1 - 0.8q^{-1} - 0.35q^{-2} + 0.49q^{-3}} 
\]

and the prefilter

\[
L_{OE}(q^{-1}) = \frac{1}{F(q^{-1})} = \frac{1}{1 - 1.5q^{-1} + 0.7q^{-2}} 
\]

that is used for output error (OE-like) systems. We suppose that the noise-free signals are available to construct the instruments. 1000 Monte Carlo runs with 4000 samples for each run are simulated.
Table 3.1: Comparison between the optimal and OE-like prefilters in the open-loop IV method.

<table>
<thead>
<tr>
<th>Prefilter</th>
<th>$a_1 = -1.5$</th>
<th>$a_2 = 0.7$</th>
<th>$b_1 = 1.0$</th>
<th>$b_2 = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OE</td>
<td>-1.4999</td>
<td>0.6999</td>
<td>0.9990</td>
<td>0.5008</td>
</tr>
<tr>
<td></td>
<td>±0.0048</td>
<td>±0.0043</td>
<td>±0.0184</td>
<td>±0.0240</td>
</tr>
<tr>
<td>Optimal</td>
<td>-1.5000</td>
<td>0.6999</td>
<td>0.9998</td>
<td>0.4999</td>
</tr>
<tr>
<td></td>
<td>±0.0034</td>
<td>±0.0032</td>
<td>±0.0075</td>
<td>±0.0092</td>
</tr>
</tbody>
</table>

As can be seen from Table 3.1, the estimates of the system are accurate in both cases. However, the optimal prefilter open-loop IV method provides estimates with lower variance than the OE-like prefilter IV method.

The optimal prefilter depends on the unknown system properties, i.e., the plant as well as the noise dynamics. Hence, the optimal accuracy cannot be achieved in practice. Without iterative refinements, the estimated covariance of the parameters for the basic IV method using the prefilter $L(q)$ is

$$\hat{P}_{IV} = \hat{\sigma}^2 R_{\xi \phi}^{T} R_{\xi \phi}^{-1} R_{\xi \phi}^{T} R_{\xi \phi}$$

and for the extended IV method

$$\hat{P}_{EIV} = \hat{\sigma}^2 \left( R_{\xi \phi}^{T} R_{\xi \phi}^{-1} R_{\xi \phi}^{T} R_{\xi \phi} \right)^{-1},$$

where $R_{\xi \phi} = \mathbb{E} \left[ \xi_t \phi_t^T \right] = \mathbb{E} \left[ L(q) \xi_t L(q) \phi_t^T \right]$, $R_{\xi \xi} = \mathbb{E} \left[ \xi_t \xi_t^T \right] = \mathbb{E} \left[ L(q) \xi_t L(q) \phi_t^T \right]$ and $\hat{\sigma}$ is the estimated standard deviation of the residual.

The open-loop refined instrumental variable method is summarized in Algorithm 1.

3.4.2 Closed-loop Refined Instrumental Variable method

In reality, experiments for system identification are often performed in closed-loop setups because of economic or safety reasons, or because the system is unstable [Forssell and Ljung, 1999]. The issue with closed-loop identification is that the disturbances fed back via the feedback mechanism correlate with the input signals, yielding biased estimates if no special strategy is introduced. Several methods have been proposed to overcome these issues of the closed-loop data such as the PEM method [Ljung, 1999] and the closed-loop subspace method [Chiuso and Picci, 2005a]. Moreover, for the linear time-invariant system, the instrumental variable (IV) method has been proposed to solve this problem [Gilson and Van den Hof, 2005, Gilson et al., 2011, Young, 2012].

Let us consider the closed-loop configuration shown in Figure 3.1,

$$u_t = C_c(q)(\delta_t - y_t)$$

$$y_t = G(q, \delta)u_t + H(q, \eta)e_t$$
Algorithm 1 Open-loop refined Instrumental Variable method

1: **Initialization:** Rewrite the system in the regression form (3.7) and use the LS method to estimate $\hat{\vartheta}^{(0)}$. This provides the initial estimate of $\hat{G}^{(0)}(q, \vartheta)$. Set $j = 1$.

2: **Estimating parameters:**

   (1) Generate the simulated noise-free output $\hat{y}_t$ from the auxiliary model
   \[
   \hat{y}_t = \frac{\hat{B}(q, \hat{\vartheta}^{(j-1)})}{\hat{F}(q, \hat{\vartheta}^{(j-1)})} u_t
   \]
   with the polynomial $\hat{B}$ and $\hat{F}$ based on the estimated parameter vector $\hat{\vartheta}^{(j-1)}$ obtained from the previous iteration.

   (2) Generate the residual $\varepsilon_t = y_t - \hat{y}_t$ and estimate the noise model $\hat{H}(q, \hat{\eta}^{(j)})$ from $\varepsilon_t = \hat{H}(q, \hat{\eta}^{(j)}) \hat{e}_t$.

   (3) Prefilter the input $u_t$, output $y_t$ and instrumental variable $\zeta_t$ using the filter
   \[
   L(q, \hat{\vartheta}^{(j-1)}, \hat{\eta}^{(j)}) = \frac{\hat{D}(q, \hat{\eta}^{(j)})}{\hat{C}(q, \hat{\eta}^{(j)}) \hat{F}(q, \hat{\vartheta}^{(j-1)})}
   \]
   where the polynomial $\hat{F}$ is based on the estimated parameter vector $\hat{\vartheta}^{(j-1)}$ obtained from the previous iteration and $\hat{\eta}^{(j)}$ obtained from Step (2).

   (4) Estimate $\hat{\vartheta}^{(j)}$ based on the filtered input, output and instrument using the basic IV method (3.17) or extended IV method (3.19), and increase $j$ by 1.

3: **Until:** $\hat{\vartheta}^{(j)}$ appears to have converged according to a stop criterion $\|\hat{\vartheta}^{(j)} - \hat{\vartheta}^{(j-1)}\|_2 / \|\hat{\vartheta}^{(j-1)}\|_2 + \|\hat{\eta}^{(j)} - \hat{\eta}^{(j-1)}\|_2 / \|\hat{\eta}^{(j-1)}\|_2$ below a threshold or if $j > j_{\text{max}}$.

4: **Covariance matrix estimate:** After convergence, compute the estimated covariance matrices of the parameters using (3.25) or (3.26).
where $C_c(q)$ is the controller and $\delta_t$ is the reference signal. Note that the controller $C_c(q)$ can be a nonlinear and/or time-varying but that it is assumed to ensure the stability of the closed-loop system according to some criterion, i.e., bounded input-bounded output (BIBO). The identification problem is then to estimate the parameters that characterize the model structure (3.27), based on the dataset

$$Z_N = \{\delta_t, u_t, y_t\}_{t=1}^N$$

where $N$ is the number of samples.

Starting similarly to the open-loop case, the output $y_t$ can be rewritten in regression form as

$$y_t = \phi_t^T \vartheta + w_t \quad (3.28)$$

that is identical to the open-loop case (3.7). However, the estimated parameters will be biased since $\phi_t$ and $w_t$ are correlated due to the feedback mechanism using (3.19)–(3.21). A solution is to use the closed-loop IV method [Gilson and Van den Hof, 2005] with a new formulation of the instruments. Then, the filtered instrument vector, in the closed-loop configuration, is computed using the auxiliary model from Figure 3.2 as follows

$$\zeta_t = L(q) \left[ -\hat{y}_{t-1}, \ldots, -\hat{y}_{t-n_a}, \hat{u}_{t-n_k-1}, \ldots, \hat{u}_{t-n_b-n_k+1} \right] \quad (3.29)$$

where

$$\hat{y}_t = \frac{\hat{B}(q, \hat{\vartheta})}{\hat{F}(q, \hat{\vartheta})} \hat{u}_t \quad (3.30a)$$

$$\hat{u}_t = C_c(q)(\delta_t - \hat{y}_t) \quad (3.30b)$$

In case of unknown controller $C_c(q)$, the instrument will be created based on the simulated noise-free signals as

$$\hat{u}_t = \hat{G}_{\delta u}(q, \hat{\vartheta}) \delta_t \quad (3.31a)$$

$$\hat{y}_t = \hat{G}_{\delta y}(q, \hat{\vartheta}) \delta_t \quad (3.31b)$$
where \( \hat{G}_{\delta u}(q, \hat{\vartheta}) \) and \( \hat{G}_{\delta y}(q, \hat{\vartheta}) \) are two estimated transfer functions from \( \delta_t \) to \( u_t \) and \( y_t \), respectively.

The structure of the closed-loop extended IV estimator is typically identical to that of the open-loop one. As a result, the choices of the prefilter \( L(q) \) and weighting matrix \( Q \) are similar and the estimated covariance matrix \( P_{IV} \) is also obtained using (3.25) for the basic IV method or (3.26) for the extended IV method. The following example shows in detail the benefit of using the optimal prefilter in the closed-loop setting.

---

**Example 3.2**

Consider a closed-loop system

\[
\begin{align*}
  u_t &= 0.5(\delta_t - y_t) \\
  y_t &= \frac{1.0q^{-1} + 0.5q^{-2}}{1 - 1.5q^{-1} + 0.7q^{-2}} u_t + \frac{1 + 0.7q^{-1}}{1 - 0.7q^{-1}} e_t
\end{align*}
\]

where \( \delta_t \) and \( e_t \) are zero mean white Gaussian processes with variance 1 and 0.25, respectively.

Here, two different prefilters are used. The first choice of the prefilter is the inverse of the output dynamic as

\[
L_{OE}(q^{-1}) = \frac{1}{F(q^{-1})} = \frac{1}{1 - 1.5q^{-1} + 0.7q^{-2}}
\]

while the optimal choice of the prefilter is

\[
L_{opt}(q^{-1}) = \frac{1}{F(q^{-1})H(q^{-1})} = \frac{1 - 0.7q^{-1}}{1 - 0.8q^{-1} - 0.35q^{-2} + 0.49q^{-3}}
\]

The noise-free signals are supposed to be available to construct the instruments. 1000 Monte Carlo runs are performed and the sample size is selected to 4000 for each run.
Table 3.2: Comparison between the optimal and OE-like prefilters in the closed-loop IV method.

<table>
<thead>
<tr>
<th>Prefilter</th>
<th>$a_1 = -1.5$</th>
<th>$a_2 = 0.7$</th>
<th>$b_1 = 1.0$</th>
<th>$b_2 = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OE</td>
<td>-1.5001</td>
<td>0.6997</td>
<td>1.0007</td>
<td>0.5006</td>
</tr>
<tr>
<td></td>
<td>±0.0170</td>
<td>±0.0175</td>
<td>±0.0329</td>
<td>±0.0347</td>
</tr>
<tr>
<td>Optimal</td>
<td>-1.5002</td>
<td>0.6999</td>
<td>1.0009</td>
<td>0.5001</td>
</tr>
<tr>
<td></td>
<td>±0.0076</td>
<td>±0.0083</td>
<td>±0.0139</td>
<td>±0.0141</td>
</tr>
</tbody>
</table>

As can be seen from Table 3.2, the optimal prefilter helps to reduce the variance of the estimated parameters. Furthermore, the closed-loop IV method provides accurate parameter estimates in both cases.

To summarize, the major steps in the implementation of the IV algorithms for the closed-loop setup are provided in Algorithm 2.

3.5 A comparison of closed-loop identification approaches

In the preceding sections some identification methods which are applicable for the open- and closed-loop identification are described. The PEM method can be used directly for the closed-loop input-output data as long as the model is flexible enough. For the Gaussian noise case, the PEM method is asymptotically efficient, implying that the lowest possible covariance of the parameter estimator is achieved. However, it suffers from the risk of converging to local minima due to the nonconvex loss function.

The least-squares and IV methods, on the other hand, avoid the use of a nonconvex loss function. However, while the least-squares method is applicable for some cases under strict assumptions, the IV method allows obtaining a consistent estimate of the parameters in a complex setting. In a closed-loop setting, it is not necessary to have prior knowledge of the feedback controller [Gilson and Van den Hof, 2005]. If the controller in the experiment is assumed to be known, the parameters are estimated iteratively with the instrument generated from the signals simulated using the controller and auxiliary plant. Otherwise, a higher order model can be used to ensure that the closed-loop model set contains the true system.

An advantage of the PEM method is that it can be applied directly to the closed-loop data and gives the smallest variance of the estimated parameter [Forsell and Chou, 1998]. It is natural since the whole input spectrum is used to reduce the variance while the IV method considers only the noise-free part of the input. As a consequence, the signal-to-noise ratio is reduced and the accuracy of the IV method will be sub-optimal in most cases. The following examples will
Algorithm 2 Closed-loop refined Instrumental Variable method

1: **Initialization:** Rewrite the system as the regression form (3.28) and use the LS method to estimate $\hat{\theta}^{(0)}$. This provides the initial estimate of $\hat{G}(q, \theta)$. Set $j = 1$.

2: **Estimating parameters:**

   (1) Generate the simulated noise-free input $\hat{u}_t$ and output $\hat{y}_t$ from the auxiliary model given in (3.30) based on $\hat{\theta}^{(j-1)}$ from the previous iteration if the controller $C_c(q)$ is known or from (3.31) in case of an unknown controller.

   (2) Generate the residual $\epsilon_t = y_t - \hat{B}(q, \hat{\theta}^{(j-1)}) \hat{F}(q, \hat{\theta}^{(j-1)}) u_t$ and estimate the noise model $\hat{H}(q, \hat{\eta}^{(j)})$ from $\epsilon_t = \hat{H}(q, \hat{\eta}^{(j)}) \hat{e}_t$.

   (3) Prefilter the input $u_t$, output $y_t$ and instrumental variable $\zeta_t$ by the filter

   $$L(q, \hat{\theta}^{(j-1)}, \hat{\eta}^{(j)}) = \frac{\hat{D}(q, \hat{\eta}^{(j)})}{\hat{C}(q, \hat{\eta}^{(j)}) \hat{F}(q, \hat{\theta}^{(j-1)})}$$

   with the polynomial $\hat{F}$ is based on the estimated parameter vector $\hat{\theta}^{(j-1)}$ obtained from the previous iteration and $\hat{\eta}^{(j)}$ obtained from Step (2).

   (4) Estimate $\hat{\theta}^{(j)}$ based on the filtered input, output and instrument using the basic IV method (3.17) or extended IV method (3.19), and increase $j$ by 1.

3: **Until:** $\hat{\theta}^{(j)}$ appears to have converged according to a stop criterion $\|\hat{\theta}^{(j)} - \hat{\theta}^{(j-1)}\|_2 / \|\hat{\theta}^{(j-1)}\|_2 + \|\hat{\eta}^{(j)} - \hat{\eta}^{(j-1)}\|_2 / \|\hat{\eta}^{(j-1)}\|_2$ below a threshold or if $j > j_{\text{max}}$.

4: **Covariance matrix estimate:** After convergence, compute the estimated covariance matrices of the parameter using (3.25) or (3.26).

clarify this point. Similar analyses can be found in [Forssell and Chou, 1998, Forssell and Ljung, 1999]

---

**Example 3.3**

Consider a closed-loop Box-Jenkins system

$$y_t = \frac{B(q)}{F(q)} u_t + \frac{C(q)}{D(q)} e_t = G(q) u_t + H(q) e_t$$

where $e_t$ is a zero-mean white noise as $e_t \sim \mathcal{N}(0, \lambda)$. The system operates in a closed-loop setting where a stabilizing linear time-invariant controller is given by

$$u_t = C_c(q)(\delta_t - y_t) = \frac{S(q)}{R(q)} (\delta_t - y_t)$$

where $\delta_t$ is the reference signal. It is also assumed that the reference signal is independent of $e_t$. 
The input \( u_t \) and output \( y_t \) can be rewritten in terms of the reference and noise as

\[
\begin{align*}
  u_t &= \frac{C_c(q)}{1 + C_c(q)G(q)} \delta_t - \frac{C_c(q)}{1 + C_c(q)G(q)} H(q)e_t \\
y_t &= \frac{C_c(q)G(q)}{1 + C_c(q)G(q)} \delta_t + \frac{1}{1 + C_c(q)G(q)} H(q)e_t
\end{align*}
\]

which can be partitioned as

\[
\begin{align*}
u_t &= u_t^\delta + u_t^e \\
y_t &= y_t^\delta + y_t^e
\end{align*}
\]

The superscript \( \delta \) and \( e \) denote contributions to the signals from the reference and noise, respectively. Note that the system can be rewritten in a regression form as

\[
y_t = \varphi_t^T \delta + w_t
\]

where \( w_t = \frac{F(q)c(q)}{D(q)} e_t \). Similarly the regression vector \( \varphi_t \) can be split into two parts

\[
\varphi_t = \varphi_t^\delta + \varphi_t^e
\]

due to the reference \( \delta_t \) and noise \( e_t \).

The direct PEM method ignores the feedback and considers directly the measured input-output data. The asymptotic variance estimate of the parameters given by the direct PEM method is

\[
\sqrt{N}(\hat{\theta}_N - \theta) \in \text{As}(0, P_{\theta}^{\text{PEM}})
\]

\[
P_{\theta}^{\text{PEM}} = \lambda_e \left[ \mathbb{E} [\psi_t \psi_t^T] \right]^{-1}
\]

where the gradient vector is

\[
\psi_t = L^*(q)(y_t - \varphi_t^T \delta) + \frac{D(q)}{F(q)c(q)} \varphi_t = L^*(q) \frac{F(q)c(q)}{D(q)} e_t + \frac{D(q)}{F(q)c(q)} \varphi_t
\]

Note that the filter \( L^*(q) \) is used to emphasize certain ranges of the input-output spectrum that may differ from the optimal prefilter \( \frac{D(q)}{F(q)c(q)} \) in (3.24). The gradient vector can be partitioned due to the reference and noise as

\[
\psi_t = \psi_t^\delta + \psi_t^e
\]

in which

\[
\begin{align*}
  \psi_t^\delta &= \frac{D(q)}{F(q)c(q)} \varphi_t^\delta \\
  \psi_t^e &= L^*(q) \frac{F(q)c(q)}{D(q)} e_t + \frac{D(q)}{F(q)c(q)} \varphi_t^e
\end{align*}
\]
Furthermore, since $\delta_t$ and $e_t$ are assumed to be independent, the accuracy of the direct PEM method in this case is

$$P_{PEM}^\vartheta = \lambda_e \left[ \mathbb{E}[\psi_t^\delta \psi_T^\delta] + \mathbb{E}[\psi_T^e \psi_T^e] \right]^{-1}$$

The closed-loop basic IV method, on the other hand, uses an instrument $\zeta_t$ and prefilter $L(q)$ to estimate the parameters as

$$\hat{\vartheta}_{IV}^N = \left[ \frac{1}{N} \sum_{t=1}^N L(q) \zeta_t L(q) \phi^T_t \right]^{-1} \left[ \frac{1}{N} \sum_{t=1}^N L(q) \zeta_t L(q) y^T_t \right]$$

Since $\delta_t$ and $e_t$ are independent and if the instrument is chosen as $\zeta_t = \phi^\delta_t$ according to Algorithm 2, $\mathbb{E}[L(q)\zeta_t L(q)\phi^T_t] = \mathbb{E}[L(q)\zeta_t L(q)\zeta^T_t]$, the optimal prefilter chosen as $L(q) = \frac{D(q)}{F(q) C(q)}$ leads to the asymptotic covariance matrix $P_{IV}^\vartheta$ in (3.25)

$$P_{IV}^\vartheta = \lambda_e \left[ \mathbb{E}[L(q)\zeta_t L(q)\phi^T_t] \right]^{-1} \left[ \mathbb{E}[L(q)\zeta_t L(q)\zeta^T_t] \right]$$

$$\quad \quad = \lambda_e \left[ \mathbb{E}[L(q)\zeta_t L(q)\zeta^T_t] \right]^{-1} \left[ \mathbb{E}[\frac{D(q)}{F(q) C(q)} \phi^\delta_t \frac{D(q)}{F(q) C(q)} \phi^\delta_T] \right]^{-1}$$

$$\quad \quad = \lambda_e \left[ \mathbb{E}[\psi^\delta_t \psi^\delta_T] \right]^{-1}$$

It can be seen that $P_{IV}^\vartheta \geq P_{PEM}^\vartheta$. The equality holds in some special cases when the contribution of the noise to the asymptotic covariance estimate in the PEM method can be neglected.

---

Example 3.4

Consider a closed-loop Box-Jenkins system

$$y_t = \frac{1.0 q^{-1} + 0.5 q^{-2}}{1 - 1.5 q^{-1} + 0.7 q^{-2}} u_t + 1 + 0.7 q^{-1} e_t$$

$$u_t = \frac{1 - 0.5 q^{-1}}{1 + 0.5 q^{-1}} (\delta_t - y_t)$$

where $\delta_t$ and $e_t$ are both zero-mean Gaussian processes with variance $\delta_t \sim \mathcal{N}(0, 1)$ and $e_t \sim \mathcal{N}(0, 0.2)$. The PEM method is performed using the function $bj$ in MATLAB’s System Identification toolbox [Ljung, 1995], while the closed-loop IV method is carried out as described in Algorithm 2. 1000 realizations of the signals are simulated with 4000 samples for each run.

The results are shown in Table 3.3. It is obvious that two methods provide consistent parameter estimates in the closed-loop setting. However, the direct PEM method gives more accurate estimates than those of the refined closed-loop IV method.
Table 3.3: Comparison between the direct PEM and refined closed-loop basic IV methods in the closed-loop setting.

<table>
<thead>
<tr>
<th>Method</th>
<th>$a_1 = -1.5$</th>
<th>$a_2 = 0.7$</th>
<th>$b_1 = 1.0$</th>
<th>$b_2 = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>direct PEM</td>
<td>$-1.5003$</td>
<td>$0.7005$</td>
<td>$0.9998$</td>
<td>$0.4998$</td>
</tr>
<tr>
<td></td>
<td>$±0.0097$</td>
<td>$±0.0081$</td>
<td>$±0.0063$</td>
<td>$±0.0052$</td>
</tr>
<tr>
<td>IV</td>
<td>$-1.5004$</td>
<td>$0.7004$</td>
<td>$0.9999$</td>
<td>$0.4998$</td>
</tr>
<tr>
<td></td>
<td>$±0.0117$</td>
<td>$±0.0109$</td>
<td>$±0.0067$</td>
<td>$±0.0054$</td>
</tr>
</tbody>
</table>

An disadvantage of the direct PEM method is that the exact input is assumed to be available. If the noise is induced by the actuator dynamic on the input, or the exact input is not accessible and must be measured, the direct PEM method is not applicable. In this circumstance, a two-stage PEM method can be applied where the noise-free input is first estimated and used to replace the measured input [Chiuso and Picci, 2005b; Hjalmarsson, 2005]. However, the efficiency of the estimator in this case is also sub-optimal.

Note that all methods are offline but can be extended to online estimators. During online estimation, the model is sequentially updated as new measurements are available. Compared to batch estimators, recursive estimators require generally lower computational cost and demand less memory. One online estimator is the EKF estimator where the parameters and state vectors of the state space model (2.16) are augmented into a new state vector. This nonlinear model with respect to new augmented state vector is linearized based on the previous estimate of the state. However, the EKF method produces a locally optimal estimate and the stability of the estimate is not always ensured. Moreover, the EKF method is based on the assumption of identically distributed and independent process and measurement noises and the independence of the input to these noises. These strict assumptions are not always satisfied in the closed-loop setting.
Estimation of forward and inverse models

In this chapter, a particular use of the IV methods is considered where the signals $u_t$ and $y_t$ are measured but it is not obvious which signal is viewed as the input. The common assumption of the input is that it is known exactly. The output, on the other hand, is measured with the noise. The measurement noise can be correlated to the input if the system operates in a closed-loop setting. Despite the input-output choice may have a strong impact on the performance of some system identification methods [Jung and Enqvist, 2013], we will here show that the basic IV estimator with $u_t$ or $y_t$ as the input gives identical results in both cases.

4.1 Introduction

The input-output selection is essential in system identification. For a single-input single-output system, the standard way is to select a noise-free signal as the input and a measured signal as the output. However, the exact input signal is sometimes unknown and it has to be replaced with a noisy measurement, leading to an errors-in-variables (EIV) problem [Söderström, 2007]. In this case, both measurement noises of $u_t$ and $y_t$ can be white as in [Zheng, 1998] or the system output is corrupted by the colored noise that describes process disturbances and measurement noises as studied in [Liu and Zhu, 2017, Zheng, 2002]. Furthermore, it could be the case that there exists a dynamical relation between the available signals but no clear distinction between input and output.

This issue can be found in, for instance, a mechanical system where two sensors measure the movements at different places or in different directions but where the external force or torque that generates the movements is unknown. For example, this setup is common in vibration analysis [Devriendt and Guillaume, 2008, Maia et al., 2001] and the model estimation problem is sometimes called
sensor-only blind system identification [D'Amato et al., 2009].

Similar examples can be found in electrical power grids, communication systems, process industry applications, and biochemical reactions. Many of these complex systems can be modeled as dynamic networks where the system identification problem is to estimate a particular part of the network, e.g., the dynamical subsystem that connects two nodes in the network [Chiuso and Pillonetto, 2012, Dankers et al., 2015, Linder and Enqvist, 2017a,b, Van den Hof et al., 2013, Weerts et al., 2015]. Note that due to the effects of feedback loops and measurement noise, a direct prediction error method (PEM) often gives a biased estimate. In this setting, Instrumental Variable (IV) methods are commonly used since they provide a way to handle challenges concerning confounding variables and EIV settings, without estimating a noise model.

Our contributions are the following. First, we will show that the basic IV estimators give identical results when estimating the forward and inverse model. More specifically, no matter which signal $u_t$ or $y_t$ is selected as the input, the corresponding model estimators are equal. This property of the basic IV estimate is achieved for finite data. Second, it is observed that the result also holds for the extended IV method in an asymptotic sense. Third, simulation studies are performed to exemplify and validate the findings.

### 4.2 Problem formulation

The considered system in this chapter is a single-input single-output (SISO) system. The input and output of the system are measured such that both available signals contain noise terms.

#### 4.2.1 Forward model

The true system is a SISO system that has a noise-free scalar input $\hat{u}_t$ and a noise-free scalar output $\hat{y}_t$. Noisy measurements $u_t$ and $y_t$ of the two signals are available and the relationships between all signals can be written

\[
\begin{align*}
\hat{y}_t &= G(q)\hat{u}_t \\
u_t &= \hat{u}_t + T(q)v_t \\
y_t &= \hat{y}_t + H(q)e_t
\end{align*}
\]

We have the following assumptions

**Assumption 4.1.** The noises $v_t$ and $e_t$ are unknown zero mean white sequences affecting the input and output, respectively.

**Assumption 4.2.** $G(q), H(q)$ and $T(q)$ are rational transfer functions in the time
shift operator $q$ ($qy_t = y_{t+1}$) according to

$$G(q) = \frac{B(q)}{A(q)} = \frac{b_0 q^{-n_k} + \cdots + b_{n_b-1} q^{-n_b-n_k+1}}{1 + a_1 q^{-1} + \cdots + a_n q^{-n_s}}$$  \hspace{1cm} (4.2)

$$H(q) = \frac{C(q)}{D(q)} = \frac{1 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c}}{1 + d_1 q^{-1} + \cdots + d_{n_d} q^{-n_d}}$$  \hspace{1cm} (4.3)

$$T(q) = \frac{S(q)}{F(q)} = \frac{1 + s_1 q^{-1} + \cdots + s_{n_s} q^{-n_s}}{1 + f_1 q^{-1} + \cdots + f_{n_f} q^{-n_f}}$$  \hspace{1cm} (4.4)

**Assumption 4.3.** The transfer functions of the noise models $H(q)$ and $T(q)$ are assumed to be stable and coprime.

The aim of this work is to derive estimates of the model $G(q)$ and its inverse form $(G)^{-1}(q)$. We rewrite the forward model in a regression form as

$$y_t = \varphi_t^T \vartheta_F + w_t$$  \hspace{1cm} (4.5)

where

$$\varphi_t^T = [u_{t-n_k}, \ldots, u_{t-n_b-n_k+1}, -y_{t-n_a}, \ldots, -y_{t-1}]$$

$$\vartheta_F^T = [b_0, \ldots, b_{n_b-1}, a_{n_a}, \ldots, a_1] = [b_0, \tilde{\vartheta}_F^T]$$

The residual $w_t$ of the forward system is generated as

$$w_t = y_t - \varphi_t^T \vartheta_F$$  \hspace{1cm} (4.6a)

$$= A(q) \left[ - G(q) T(q) v_t + H(q) e_t \right]$$  \hspace{1cm} (4.6b)

$$= -B(q) T(q) v_t + A(q) H(q) e_t$$  \hspace{1cm} (4.6c)

which implies that $w_t$ can be modeled as $w_t = L(q)^{-1} \eta_{F,t}$, where $L(q)^{-1}$ is a stable function and the noise $\eta_{F,t}$ is white. Note that the transfer function $G(q)$ can be unstable and non-minimum phase.

### 4.2.2 Inverse model

We now consider the problem of rewriting the inverse model $G(q)^{-1}$ in regression form. First, we can write the inverse model relation as

$$u_t = G(q)^{-1} y_t + G(q)^{-1} \left[ G(q) T(q) v_t - H(q) e_t \right]$$

$$= G(q)^{-1} y_t + T(q) v_t - G(q)^{-1} H(q) e_t$$

Since the forward model $G(q)$ is rational proper, the inverse model will be non-causal. Under the assumption that $b_0 \neq 0$, this inverse model can be written on regression form as

$$u_{t-n_k} = \psi_t^T y_t + \epsilon_t$$  \hspace{1cm} (4.7)
where

$$\psi_t^T = [-u_{t-n_k-1}, \ldots, -u_{t-n_k-n_b+1}, y_{t-n_a}, \ldots, y_t] = [-\tilde{\varphi}_t^T, y_t]$$

$$\gamma_t^T = \begin{bmatrix} b_1 & \cdots & b_{n_b-1} & a_{n_a} & \cdots & a_1 & 1 \end{bmatrix} \frac{1}{b_0} = [\tilde{\varphi}_F^T, 1]$$

and the residual $\epsilon_t$ is generated as

$$\epsilon_t = u_{t-n_k} - \psi_t^T \gamma_t$$

$$= 1 \frac{1}{b_0} B(q) [T(q)v_t - G(q)^{-1} H(q)e_t]$$

$$= B(q) T(q) \frac{v_t}{b_0} - A(q) H(q) \frac{e_t}{b_0}$$

### 4.3 On equivalence of the estimation of the forward and inverse models

If the noise-free signals $\{\hat{u}_t, \hat{v}_t\}$ are unknown, but measured with the noises as in (4.1), the direct methods such as PEM provide biased results. The Instrumental Variable method [Young, 2012] can provide consistent estimates of a plant without estimating a noise model in an EIV setting. However, if the true noise model is assumed to be known, the estimated covariance of the parameters can be minimized compared to (3.24). This means that the filtered input $\hat{u}_t = L(q)u_t$, the filtered output $\hat{y}_t = L(q)y_t$ and the filtered instrument $\hat{\zeta}_t = L(q)\zeta_t$ are used in the estimator. In practice, an approximation of the true noise model is often used to define the prefilter.

In order to prove the equivalence of the estimators of the forward and inverse models, the following matrices are introduced using a finite dataset measurements from $t = 1$ to $N$

$$\Phi_N = \begin{bmatrix} \tilde{\varphi}_{p+1} & \tilde{\varphi}_{p+2} & \cdots & \tilde{\varphi}_N \end{bmatrix}^T$$

$$Z_N = \begin{bmatrix} \tilde{\zeta}_{p+1} & \tilde{\zeta}_{p+2} & \cdots & \tilde{\zeta}_N \end{bmatrix}^T$$

$$Y_N = \begin{bmatrix} \tilde{y}_{p+1} & \tilde{y}_{p+2} & \cdots & \tilde{y}_N \end{bmatrix}^T$$

$$U_N = \begin{bmatrix} \tilde{u}_{p-n_k+1} & \tilde{u}_{p-n_k+2} & \cdots & \tilde{u}_{N-n_k} \end{bmatrix}^T$$

where $p = \max\left\{n_a, n_b + n_k - 1\right\}$.

### 4.3.1 The basic IV method

Using a basic IV method, an estimate of the parameter vector $\tilde{\varphi}_F$ of the forward model can be obtained by solving

$$\tilde{\varphi}_F = \text{sol} \left[ \frac{1}{N} Z_N^T \Phi_{F,N} \tilde{\varphi}_F - \frac{1}{N} Z_N^T Y_N = 0 \right]$$
4.3 On equivalence of the estimation of the forward and inverse models

where \( \Phi_{F,N} = [\bar{U}_N, \bar{\Phi}_N] \).

Similarly, an estimate of the parameter vector \( \gamma_I \) of the inverse model can be obtained by solving

\[
\hat{\gamma}_I = \text{sol}_{\gamma_I} \left[ \frac{1}{N} \bar{Z}_N^T \Phi_{I,N} \gamma_I - \frac{1}{N} \bar{Z}_N^T \bar{U}_N = 0 \right]
\] (4.11)

where \( \Phi_{I,N} = [-\bar{\Phi}_N, \bar{Y}_N] \). Let \( \hat{\gamma}_I \) be the vector consisting of the first \( n_a + n_b - 1 \) elements of \( \hat{\gamma}_I \) and let \( \hat{\gamma}_{I,n_a+n_b} \) be the last element. If \( \hat{\gamma}_{I,n_a+n_b} \neq 0 \), an estimate \( \hat{\delta}_I \) of the forward model parameters can be obtained from \( \hat{\gamma}_I \) as

\[
\hat{\delta}_I = \frac{1}{\hat{\gamma}_{I,n_a+n_b}} \begin{bmatrix} 1 & \hat{\gamma}_I \end{bmatrix}^T
\] (4.12)

If the true input and output would have been known, one choice of filtered instrument \( \bar{\zeta}_t \) could have been

\[
\bar{\zeta}_t = \begin{bmatrix} \bar{u}_{t-n_k}, \ldots, \bar{u}_{t-n_b-n_k+1}, \bar{y}_{t-n_a}, \ldots, \bar{y}_{t-1} \end{bmatrix}^T
\] (4.13)

in which \( \bar{u}_t \) and \( \bar{y}_t \) are the filtered noise-free input and output, respectively. However, this is of course not the case in practice and approximations of the instruments in (4.13) are often used instead. Such approximations can sometimes be obtained using a known external signal, such as the reference signal in a closed-loop system.

For the basic IV method, the estimated covariance matrix \( \hat{P}_{\delta_F} \) for \( \delta_F \) is given by

\[
\hat{P}_{\delta_F} = \hat{\sigma}^2 \left[ \bar{Z}_N^T \Phi_{F,N} \right]^{-1} \left[ \bar{Z}_N^T \bar{Z}_N \right] \left[ \bar{Z}_N^T \Phi_{F,N} \right]^{-T}
\] (4.14)

where \( \hat{\sigma}^2 \) is the estimated variance of the model residual.

Since the available signals \( u_t \) and \( y_t \) can have very different signal-to-noise ratios, it seems relevant to consider whether the signal qualities should affect the choice of a forward or an inverse model estimator. However, it turns out that the basic IV estimators are equivalent.

**Lemma 4.4.** Assume that the collected dataset with \( N \) input and output measurements and the chosen instrument vector \( \bar{\zeta}_t \in \mathbb{R}^{n_a+n_b} \) are such that the forward and inverse IV estimates in (4.10) and (4.11) are unique and \( \hat{b}_0 = \hat{\delta}_{F,1} \neq 0 \) and \( \hat{\gamma}_{I,n_a+n_b} \neq 0 \). Then, it holds that

\[
\hat{\delta}_F = \hat{\delta}_I
\] (4.15)

**Proof:** From (4.10), the estimate of \( \delta_F \) of the forward model is given as

\[
\hat{\delta}_F = \text{sol}_{\delta_F} \left[ \frac{1}{N} \bar{Z}_N^T \Phi_{F,N} \delta_F - \frac{1}{N} \bar{Z}_N^T \bar{Y}_N = 0 \right]
\] (4.16)

where \( \Phi_{F,N} = [\bar{U}_N, \bar{\Phi}_N] \), which implies

\[
\begin{bmatrix} \hat{\delta}_F \\ 1 \end{bmatrix} \in \text{Null}\{\bar{Z}_N^{-T}[-\Phi_{F,N}, \bar{Y}_N]\}
\] (4.17)
Dividing this vector with $\hat{b}_0$ gives
\[
\begin{bmatrix}
\hat{\delta}_E \\
\hat{b}_0 \\
1
\end{bmatrix}
\in \text{Null}[\hat{Z}_N^T [-\Phi_{E,N}, \hat{Y}_N]]
\]
(4.18)

With the notation $\hat{\delta}_E = [1, \hat{b}_1, \hat{b}_0, \ldots, \hat{b}_{nb-1}, \hat{b}_0, \ldots, \hat{b}_1]^T$, the previous expression can be rewritten as
\[
\begin{bmatrix}
1 \\
\hat{\delta}_E \\
\hat{b}_0 \\
1
\end{bmatrix}
\in \text{Null}[\hat{Z}_N^T [-\hat{U}_N, \Phi_{I,N}]]
\]
(4.19)

which implies that $\hat{\gamma}_F = [\hat{\delta}_E, 1, \hat{b}_0]^T$ is also the solution to the inverse IV problem. Since the solution to the inverse IV problem is unique, this implies that $\hat{\gamma}_I = \hat{\gamma}_F$. Hence, the result follows.

A straightforward consequence of this result is that the model residuals also are equal except for a scale factor.

**Corollary 4.5.** Under the same assumptions as in Lemma 4.4, it holds that
\[
\hat{\epsilon}_t = -\frac{1}{\hat{b}_0} \hat{\omega}_t
\]
(4.20)

where $\hat{\omega}_t$ and $\hat{\epsilon}_t$ are the forward and inverse model residuals, respectively.

**Proof:** The residual of the forward model is
\[
\hat{\omega}_t = y_t - \varphi_t^T \hat{\delta}_F = y_t - [u_{t-nk}, \varphi_t^T] \hat{\delta}_F
\]
\[
= y_t - [u_{t-nk}, \varphi_t^T] \begin{bmatrix}
\hat{b}_0 \\
\hat{\delta}_E
\end{bmatrix}
\]
On the other hand, the residual of the inverse model is
\[
\hat{\epsilon}_t = u_{t-nk} - \psi_t^T \hat{\gamma}_I = u_{t-nk} - [-\varphi_t^T, y_t] \hat{\gamma}_I
\]
\[
= u_{t-nk} - [-\varphi_t^T, y_t] \begin{bmatrix}
\hat{\delta}_E \\
\hat{b}_0 \\
1
\end{bmatrix}
= -\frac{1}{\hat{b}_0} y_t + u_{t-nk} + \varphi_t^T \hat{\delta}_F \frac{\hat{b}_0}{\hat{b}_0}
\]
\[
= -\frac{1}{\hat{b}_0} \hat{\omega}_t
\]
Remark 4.6. The residuals of the forward and inverse models can be modeled as

\[ w_t = -B(q)T(q)v_t + A(q)H(q)e_t = \left(L(q)\right)^{-1}\eta_{F,t} \]

\[ \epsilon_t = B(q)T(q)\frac{v_t}{b_0} - A(q)H(q)\frac{e_t}{b_0} = \left(L(q)\right)^{-1}\eta_{I,t} \]

where \( \eta_{F,t} \) and \( \eta_{I,t} \) are the driving white noises of the forward and inverse noise models, respectively. The difference between \( w_t \) and \( \epsilon_t \) is a factor \( -\frac{1}{b_0} \) which affects the variance of the noises \( \eta_{F,t} \) and \( \eta_{I,t} \). If an ARMA model estimator is used, the same estimated noise model \( \hat{L}(q) \) is obtained in the forward and inverse approaches. Hence, \( \hat{L}(q) \) can be used to filter the residuals to achieve estimates of the driving white noises \( \eta_{F,t} \) and \( \eta_{I,t} \). The variances of these signals are related as \( \text{Var}[\eta_{I,t}] = \left(\frac{1}{b_0}\right)^2 \text{Var}[\eta_{F,t}] \).

Remark 4.7. Since \( \hat{\theta}_F = \hat{\theta}_I \) for the basic IV method, it follows that these estimators also must have equal covariance matrices. Hence, the estimate of the covariance matrix for the forward IV approach can be used also for \( \hat{\theta}_I \). Note that the estimated covariance matrix \( \hat{P}_{\gamma I} \) of \( \hat{\gamma}_I \) could also be derived similarly compared to \( \hat{P}_{\theta F} \) in (4.14).

Remark 4.8. The estimated parameter vector \( \hat{\theta}_F \) using (4.10) is unique if the signals are persistently exciting (the matrix \( \bar{Z}_T^T\Phi_{F,N} \) is invertible) with the instrument as an approximation of (4.13). Therefore, this instrument vector can also be used to obtain \( \hat{\gamma}_I \) of the inverse model.

4.3.2 The extended IV method

An extended IV estimate of \( \theta_F \) of the forward model is obtained by generalizing the optimization problem by prefiltering the data and using an augmented instrument that leads to an overdetermined set of equation as

\[ \hat{\theta}_F = \arg \min_{\theta_F} \left\| \frac{1}{N} \bar{Z}_T^T\Phi_{F,N}\theta_F - \frac{1}{N} \bar{Z}_N \bar{Y}_N \right\|^2 \]  

(4.21)

where \( \Phi_{F,N} = [\bar{U}_N, \bar{\Phi}_N] \). The solution of (4.21) is

\[ \hat{\theta}_F = \left[ \Phi_{F,N}^T \bar{Z}_N \bar{Z}_N^T \Phi_{F,N} \right]^{-1} \left[ \Phi_{F,N}^T \bar{Z}_N \bar{Y}_N \right] \]  

(4.22)

Similarly, an estimate of the parameter vector \( \gamma_I \) of the inverse model using the extended IV method can be obtained by solving

\[ \hat{\gamma}_I = \arg \min_{\gamma_I} \left\| \frac{1}{N} \bar{Z}_N^T\Phi_{I,N}\gamma_I - \frac{1}{N} \bar{Z}_N^T\bar{U}_N \right\|^2 \]  

(4.23)

which has an analytical solution

\[ \hat{\gamma}_I = \left[ \Phi_{I,N}^T \bar{Z}_N \bar{Z}_N^T \Phi_{I,N} \right]^{-1} \left[ \Phi_{I,N}^T \bar{Z}_N \bar{U}_N \right] \]  

(4.24)

where \( \Phi_{I,N} = [-\bar{\Phi}_N, \bar{Y}_N] \).

If the model belongs to the model set, i.e., \( G \in \mathcal{G} \), the extended IV estimate give a consistent estimator of the parameters under the following three conditions.
• \( \mathbb{E}[\frac{1}{N} \bar{Z}_N^T \Phi_{F,N}] \) is non singular,
• \( \mathbb{E}[\frac{1}{N} \bar{Z}_N^T \bar{W}_N] = 0 \) where \( \bar{W}_N = \bar{Y}_N - \Phi_{F,N}^T \delta_F \).
• The signals involved are strongly ergodic.

where \( \mathbb{E}[\cdot] \) is the expectation and \( \delta_F = \lim_{N \to \infty} \hat{\delta}_F, \delta_F = [\bar{b}_0, \hat{\delta}_F]^T \).

**Lemma 4.9.** Assume that an instrument \( Z_N \) that satisfies the three above conditions is used, the estimates for the forward and inverse models given by the extended Instrumental Variable estimate are unique and \( \bar{b}_0 = \hat{\delta}_{F,1} \neq 0 \) and \( \check{y}_{1,n_a+n_b} \neq 0 \). Then, it holds that

\[ \hat{\delta}_F - \hat{\delta}_I \to 0 \quad \text{as} \quad N \to \infty \quad \text{with probability 1} \quad (4.25) \]

**Proof:** From the second condition \( \mathbb{E}[\frac{1}{N} \bar{Z}_N^T \bar{W}_N] = 0 \) where \( \bar{W}_N = \bar{Y}_N - \Phi_{F,N}^T \delta_F \) and \( \delta_F = [\bar{b}_0, \hat{\delta}_F]^T \) yields

\[
\mathbb{E}[\frac{1}{N} \bar{Z}_N^T (\bar{Y}_N - \bar{b}_0 \bar{U}_N)] = \lim_{N \to \infty} \frac{1}{N} \bar{Z}_N^T (\bar{Y}_N - \bar{b}_0 \bar{U}_N) = \lim_{N \to \infty} \frac{1}{N} \bar{Z}_N^T (\bar{W}_N + \bar{\Phi}_N \hat{\delta}_F) = 0 + \lim_{N \to \infty} \frac{1}{N} \bar{Z}_N^T \bar{\Phi}_N \hat{\delta}_F
\]

(4.26)

In the rest of the proof, the limitation \( \lim_{N \to \infty} \) is omitted for simplicity.

Multiplying both sides with \( (\bar{Y}_N + \bar{b}_0 \bar{U}_N)^T \bar{Z}_N \), we have

\[(\bar{Y}_N + \bar{b}_0 \bar{U}_N)^T \bar{Z}_N \bar{Z}_N^T (\bar{Y}_N - \bar{b}_0 \bar{U}_N) = (\bar{Y}_N + \bar{b}_0 \bar{U}_N)^T \bar{Z}_N \bar{Z}_N^T \bar{\Phi}_N \hat{\delta}_F \quad (4.27)\]

where \( \frac{1}{N} \) is removed since it appears on the both sides. Taking the terms out from the brackets in both sides leads to

\[ \bar{Y}_N^T \bar{Z}_N \bar{Z}_N^T \bar{Y}_N - \bar{b}_0^2 \bar{U}_N^T \bar{Z}_N \bar{Z}_N^T \bar{U}_N = \bar{Y}_N^T \bar{Z}_N \bar{Z}_N^T \bar{\Phi}_N \hat{\delta}_F + \bar{b}_0 \bar{U}_N^T \bar{Z}_N \bar{Z}_N^T \bar{\Phi}_N \hat{\delta}_F \quad (4.28)\]

Assuming \( \bar{b}_0 \neq 0 \) and dividing both sides with \( \bar{b}_0 \) and rearranging the terms yields

\[
\frac{1}{\bar{b}_0} \bar{Y}_N^T \bar{Z}_N \bar{Z}_N^T \bar{Y}_N - \bar{Y}_N^T \bar{Z}_N \bar{Z}_N^T \bar{\Phi}_N \hat{\delta}_F = \bar{b}_0 \bar{U}_N^T \bar{Z}_N \bar{Z}_N^T \bar{U}_N + \bar{U}_N^T \bar{Z}_N \bar{Z}_N^T \bar{\Phi}_N \hat{\delta}_F \quad (4.29)\]

From the extended IV solution (4.22) for the forward model, we already know that

\[
\begin{bmatrix}
\bar{U}_N^T \bar{Z}_N \bar{Z}_N^T \bar{U}_N & \bar{U}_N^T \bar{Z}_N \bar{Z}_N^T \bar{\Phi}_N \\
\bar{\Phi}_N \bar{Z}_N \bar{Z}_N^T \bar{U}_N & \bar{\Phi}_N \bar{Z}_N \bar{Z}_N^T \bar{\Phi}_N
\end{bmatrix}
\begin{bmatrix}
\bar{b}_0 \\
\hat{\delta}_F
\end{bmatrix}
= 
\begin{bmatrix}
\bar{U}_N^T \bar{Z}_N \bar{Z}_N^T \bar{Y}_N \\
\bar{\Phi}_N \bar{Z}_N \bar{Z}_N^T \bar{Y}_N
\end{bmatrix}
\quad (4.30)
\]

Considering the second subequation and dividing both sides with \( \bar{b}_0 \neq 0 \) yields

\[
\bar{\Phi}_N \bar{Z}_N \bar{Z}_N^T \bar{U}_N + \bar{\Phi}_N \bar{Z}_N \bar{Z}_N^T \bar{\Phi}_N \hat{\delta}_F = \frac{1}{\bar{b}_0} \bar{\Phi}_N \bar{Z}_N \bar{Z}_N^T \bar{Y}_N \quad (4.31)
\]
which can be rewritten as

\[
- \frac{1}{b_0} \tilde{\Phi}_N \tilde{Z}_N \tilde{Z}_N^T \tilde{Y}_N + \tilde{\Phi}_N \tilde{Z}_N \tilde{Z}_N^T \tilde{\delta}_F = -\tilde{\Phi}_N \tilde{Z}_N \tilde{Z}_N^T \tilde{U}_N
\]  

(4.32)

From (4.29) and the first subequation of (4.30), we get

\[
\frac{1}{b_0} \tilde{Y}_N^T \tilde{Z}_N \tilde{Z}_N^T \tilde{Y}_N - \tilde{Y}_N^T \tilde{Z}_N \tilde{Z}_N^T \tilde{\delta}_F \frac{1}{b_0} = \tilde{U}_N^T \tilde{Z}_N \tilde{Z}_N \tilde{Y}_N
\]  

(4.33)

Hence, \([\tilde{\delta}_F; \frac{1}{b_0}]\) is the solution of

\[
\begin{bmatrix}
\tilde{\Phi}_N^T \tilde{Z}_N \tilde{Z}_N^T \tilde{\delta}_F \\
-\tilde{\Phi}_N^T \tilde{Z}_N \tilde{Z}_N^T \tilde{\delta}_F
\end{bmatrix} = \begin{bmatrix}
\tilde{U}_N^T \tilde{Z}_N \tilde{Z}_N \tilde{Y}_N \\
0
\end{bmatrix}
\]  

(4.34)

The above solution is also the solution of the extended IV estimate of the inverse model (4.24), which is unique due to the first condition implying that

\[
\tilde{\gamma}_I = \begin{bmatrix}
\tilde{\delta}_F \\
\frac{1}{b_0}
\end{bmatrix}
\]  

(4.35)

Moreover, since \(\tilde{\delta}_I = \frac{1}{\tilde{\gamma}_I \eta_a + \eta_b} \begin{bmatrix} 1 & \tilde{\gamma}_I \end{bmatrix}^T\), then

\[
\tilde{\delta}_I - \hat{\delta}_F \rightarrow 0 \quad \text{as} \quad N \rightarrow \infty
\]  

(4.36)

The above proof shows the asymptotic relation between the estimates of the forward and inverse models. Another indication to quantify the performance of the method is based on the accuracy which can be described as confidence ellipsoids around the true forward and inverse system parameters. This is motivated as, under reasonable assumptions, as the number of measurements \(N\) tends to infinity, the random variable \(\sqrt{N}(\hat{\delta}_F - \delta_f)\) or \(\sqrt{N}(\hat{\gamma}_I - \gamma_I)\) converges to Gaussian random variables. Note that the relation between \(\tilde{\delta}_I\) and \(\gamma_I\) is given in Lemma 4.9 as \(\tilde{\delta}_I = f(\gamma_I)\). Assuming a sufficient smoothness of \(f\), yields

\[
\sqrt{N}(\hat{\delta}_I - \delta_I) \sim \mathcal{N}(0, \text{AsCov}(f(\hat{\gamma}_I))), \quad \text{as} \quad N \rightarrow \infty
\]  

(4.37)

The \(\text{AsCov}(f(\hat{\gamma}_I))\) is obtained using Gauss approximation method [Ljung, 1999] as

\[
\text{AsCov}(f(\hat{\gamma}_I)) = \nabla f^T(\hat{\gamma}_I) \hat{P}_{\hat{\gamma}_I} \nabla f(\hat{\gamma}_I)
\]  

(4.38)

where \(\nabla f\) indicates the partial derivative of \(f\) with respect to elements of \(\hat{\gamma}_I\).

\textbf{Remark 4.10.} Since \(\hat{\delta}_F \rightarrow \hat{\delta}_I\) as \(N \rightarrow \infty\), it has to be noted that the asymptotic covariance of \(\hat{\delta}_I\) will also be equivalent to that of \(\hat{\delta}_F\) or \(\text{AsCov}(\hat{\delta}_F) \rightarrow \text{AsCov}(f(\hat{\gamma}_I))\). Therefore, the asymptotic variance of the parameter of the inverse model can be omitted.
4.4 Simulation study

In this section, we present the results of two Monte Carlo simulations where the estimation of forward and inverse models using the basic IV method with the optimal instrument. Comparisons between the basic IV method and other methods are also carried out.

Example 4.11

In the first simulation, the data is generated using

\[ G(q) = \frac{0.01q^{-1}}{1 - 0.995q^{-1}} \]

\[ H(q) = \frac{1 + 0.9q^{-1}}{1 - 0.9q^{-1}} \]

and \( T(q) = 1 \). The signal-to-noise ratio (SNR) at the input and the SNR at the output are defined, respectively, as

\[ \text{SNRI} = 10 \log_{10} \frac{\sum_{t=1}^{N} \hat{u}_t^2}{\sum_{t=1}^{N} v_t^2} \approx 7.2 \text{dB}, \quad \text{SNRO} = 10 \log_{10} \frac{\sum_{t=1}^{N} \hat{y}_t^2}{\sum_{t=1}^{N} e_t^2} \approx 20 \text{dB}. \]

The Monte Carlo simulation is performed with 100 runs to create different realizations with data length 5000. The input is generated as

\[ u_t = \frac{1.0q^{-1}}{1 - 0.5q^{-1}} \delta_t \quad (4.39) \]

where \( \delta_t \sim \mathcal{N}(0, 1.0^2) \) can be considered a known external signal.

The results obtained from three methods are shown in Table 4.1 for the mean and standard deviation estimates of the parameters and Table 4.2 for the differences between these values, respectively. Note, the conventional least-squares method is applied directly to the measured input-output data while the two-stage PEM and refined basic IV methods also use the simulated noise-free signals as instruments. For the two-stage method, the two transfer functions from \( \delta_t \) to \( u_t \) and \( y_t \) are first estimated. The noise-free signals are simulated \( \hat{u}_t = \hat{G}_{\delta u}(q)\delta_t, \hat{y}_t = \hat{G}_{\delta y}(q)\delta_t \) and the forward and inverse models are then estimated based on these noise-free signals. In the two-stage approach, the models are assumed to be Box-Jenkins models which are estimated using the PEM method. On the other hand, the instrument vector is constructed based on the filtered simulated noise-free signals \( \hat{u}_t \) and \( \hat{y}_t \) for the forward model estimate. Once the refined basic IV method for the forward model converges, the instrument and prefiltered measured signals are used to estimate the parameters of the inverse model.

In Table 4.1, we can observe that the LS method provides biased results while the accurate estimates are obtained from the two-stage and basic IV methods. However, only the basic IV method gives estimates of \( \hat{\delta}_F \) and \( \hat{\delta}_I \) that are essentially identical. The small differences shown in Table 4.2 are insignificant and
Table 4.1: The estimates of the parameters of the forward and inverse models obtained from the first simulation study when the conventional least-squares, two-stage (PEM) and basic IV methods are used.

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>( a_1 )</th>
<th>( b_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS</td>
<td>Forward</td>
<td>(-0.9889 \pm 0.0009)</td>
<td>(0.0909 \pm 0.0032)</td>
</tr>
<tr>
<td></td>
<td>Inverse</td>
<td>(-0.9754 \pm 0.0063)</td>
<td>(0.5665 \pm 0.1238)</td>
</tr>
<tr>
<td>Two-stage(PEM)</td>
<td>Forward</td>
<td>(-0.9950 \pm 0.0003)</td>
<td>(0.0999 \pm 0.0018)</td>
</tr>
<tr>
<td></td>
<td>Inverse</td>
<td>(-0.9957 \pm 0.0107)</td>
<td>(0.0999 \pm 0.0016)</td>
</tr>
<tr>
<td>Basic IV</td>
<td>Forward</td>
<td>(-0.9950 \pm 0.0004)</td>
<td>(0.1000 \pm 0.0022)</td>
</tr>
<tr>
<td></td>
<td>Inverse</td>
<td>(-0.9950 \pm 0.0004)</td>
<td>(0.1000 \pm 0.0022)</td>
</tr>
</tbody>
</table>

Table 4.2: The differences of the parameter estimates obtained from the first simulation study using the basic IV method: \( \delta_F - \delta_I \) and \( \text{std}(\delta_F) - \text{std}(\delta_I) \). The results obtained from the conventional LS and two-stage methods are omitted since the differences seen from Table 4.1 are obviously significant.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean estimate</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 = -0.995 )</td>
<td>(0.2673 \times 10^{-7})</td>
<td>(0.2051 \times 10^{-7})</td>
</tr>
<tr>
<td>( b_1 = 0.01 )</td>
<td>(0.3467 \times 10^{-7})</td>
<td>(0.1405 \times 10^{-7})</td>
</tr>
</tbody>
</table>

Example 4.12

In order to verify our findings, we perform a second simulation study with a transfer function

\[
G(q) = \frac{0.5q^{-1} + 0.4q^{-2}}{1 - 1.5q^{-1} + 0.8q^{-2}}
\]

The measurement noises sequences at the input and output are low-pass filtered signals generated by using \( T(q) = \frac{1.0}{1-0.75q^{-1}} \) and \( H(q) = \frac{1+0.9q^{-1}}{1-0.9q^{-1}} \), respectively. The driving noise variances are chosen to achieve that \( \text{SNRI} \approx 10 \text{ dB} \), \( \text{SNRO} \approx 20 \text{ dB} \) and 100 Monte Carlo simulations are conducted with different noise realizations. The input is generated as

\[
u_t = \frac{1.0q^{-1}}{1 - 0.5q^{-1}} \delta_t \tag{4.40}
\]

where \( \delta_t \sim \mathcal{N}(0, 1.0^2) \). For each Monte Carlo run, 5000 samples are simulated.
The estimates of the parameters of the forward and inverse models obtained from three methods are shown in Table 4.3 while Table 4.4 shows the differences between these estimates. It can be seen that the conventional LS estimates obviously exhibit a bias. The two-stage PEM and basic IV methods provide the desired results. However, the differences between the estimated forward and inverse models using the two-stage method are obvious and thus this method is omitted in Table 4.4. Again, it can be seen that the mismatch between the two estimates of $\theta_F$ and $\theta_I$ using the basic IV method is small.
Table 4.3: The estimates of the parameters of the forward and inverse models obtained from the second simulation study when the conventional least-squares, two-stage (PEM) and basic IV methods are used.

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>Parameter</th>
<th>Mean estimate</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LS</td>
<td>Forward</td>
<td>$a_1 = -1.5$</td>
<td>$-1.5516 \pm 0.0047$</td>
<td>$-0.0623 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_2 = 0.8$</td>
<td>$0.8014 \pm 0.0044$</td>
<td>$-0.0465 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>Inverse</td>
<td>$b_1 = 0.5$</td>
<td>$0.4166 \pm 0.0090$</td>
<td>$0.0557 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_2 = 0.4$</td>
<td>$1.5035 \pm 0.0538$</td>
<td>$0.0091 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>Parameter</th>
<th>Mean estimate</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-stage(PEM)</td>
<td>Forward</td>
<td>$a_1 = -1.5$</td>
<td>$-1.4999 \pm 0.0021$</td>
<td>$-0.1050 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_2 = 0.8$</td>
<td>$0.8001 \pm 0.0021$</td>
<td>$-0.3318 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>Inverse</td>
<td>$b_1 = 0.5$</td>
<td>$0.4992 \pm 0.0053$</td>
<td>$-0.1332 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_2 = 0.4$</td>
<td>$0.4998 \pm 0.0047$</td>
<td>$-0.1332 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Model</th>
<th>Parameter</th>
<th>Mean estimate</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic IV</td>
<td>Forward</td>
<td>$a_1 = -1.5$</td>
<td>$-1.4996 \pm 0.0043$</td>
<td>$-0.1050 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$a_2 = 0.8$</td>
<td>$0.7996 \pm 0.0048$</td>
<td>$-0.3318 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td>Inverse</td>
<td>$b_1 = 0.5$</td>
<td>$0.4996 \pm 0.0065$</td>
<td>$-0.1332 \times 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$b_2 = 0.4$</td>
<td>$0.4996 \pm 0.0065$</td>
<td>$-0.1332 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 4.4: The differences of the parameter estimates obtained from the second simulation study: $\hat{\vartheta}_F - \hat{\vartheta}_I$ and std($\hat{\vartheta}_F$) - std($\hat{\vartheta}_I$). The results obtained from the conventional LS and two-stage methods are omitted since the differences seen from Table 4.3 are obviously significant.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean estimate</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1 = -1.5$</td>
<td>$-0.0623 \times 10^{-6}$</td>
<td>$-0.0427 \times 10^{-6}$</td>
</tr>
<tr>
<td>$a_2 = 0.8$</td>
<td>$0.0465 \times 10^{-6}$</td>
<td>$-0.0352 \times 10^{-6}$</td>
</tr>
<tr>
<td>$b_1 = 0.5$</td>
<td>$0.0557 \times 10^{-6}$</td>
<td>$0.0091 \times 10^{-6}$</td>
</tr>
<tr>
<td>$b_2 = 0.4$</td>
<td>$-0.1050 \times 10^{-6}$</td>
<td>$0.1332 \times 10^{-6}$</td>
</tr>
</tbody>
</table>
Choosing an appropriate model is the first step to describe quadcopter behaviour with desired accuracy. The model can be derived based on the expressions of the kinematics and kinetics of the rigid body. The kinematics describe the relation between position and velocities (or orientation and angular velocities) while the kinetics define the relations between forces and moments and the momentum. These forces and moments rely on the well-known theory of aerodynamics applied to the propellers of the quadcopter. Moreover, these relations can be combined with Newton’s laws of motion, for example, $F = m\ddot{v}$ for the linear motion, to formulate the complete nonlinear equations of motions. For more details on the nonlinear equations of motions, covering all the aerodynamics of the quadcopter, see, e.g., [Bangura and Mahony, 2012, Bristeau et al., 2009, Hoffmann et al., 2007, Martin and Salaun, 2010, Svacha et al., 2017].

In this thesis, we will focus on a couple of subsystems of the quadcopter since the complete dynamic model is too complex. Specifically, the equations of motions will be projected to a single axis to create the transfer functions and state space models appropriate for estimation purposes.

### 5.1 Quadcopter dynamics

We consider a quadcopter as shown in Figure 5.1. To derive and understand the dynamic behavior of the quadcopter, two reference frames are defined: the inertial and body-fixed frames.

**Definition 5.1 (The inertial frame).** The inertial coordinate frame is an earth-fixed coordinate system with its origin at the defined home location. As shown in Figure 5.1, the unit vector $x$ is supposed to point east, $y$ is directed north, and $z$ is pointing in the opposite direction to the center of the earth.
Definition 5.2 (The body-fixed frame). The origin of the body-fixed frame is assumed to coincide with the center of mass of the quadcopter. The unit vector $x_b$ in this frame points toward to the front, $y_b$ points to the left, and $z_b$ points downward, as shown in Figure 5.1.

5.1.1 Representation of rigid body motions

The relation between these two frames can be described by the state variables of the quadcopter. In general, to develop the equations of motions for a quadcopter, twelve state variables corresponding to six degrees of freedom will be introduced. There are three position states and three linear velocity states related to the translational motion of the quadcopter. In the inertial frame, the position of the quadcopter is defined as $\xi = [x \ y \ z]^T$. Similarly, the rotational motion of the quadcopter can be defined using three angular positions, the roll, pitch and yaw angles. These angles are commonly known as Euler angles and collected in $\eta = [\phi \ \theta \ \psi]^T$. Euler angles are commonly used because they provide an intuitive means for representing the orientation of a body in three dimensions. The rotation sequence $\phi - \theta - \psi$ is commonly used for aircraft and is just one of several Euler angle systems in use [Beard and McLain, 2012].

Euler angle representations have a mathematical singularity that can cause computational instabilities [Beard and McLain, 2012, Gustafsson, 2010]. For the $\phi - \theta - \psi$ Euler angle sequence, there is a singularity when the pitch angle $\theta$ is $\pm 90^\circ$, in which case the yaw angle is undefined. This singularity is commonly referred to as gimbal lock. A common alternative to Euler angles is the quaternion framework. While the quaternion orientation representation lacks the intuitive appeal of Euler angles, they are free of mathematical singularities and are computationally more efficient.

The linear velocities $V_b = [u \ v \ w]^T$ and the angular velocities as $v = [p \ q \ r]^T$ of the quadcopter are defined with respect to the body-fixed frame.
5.1 Quadcopter dynamics

5.1.2 Kinematic relations

The translational velocity $V_b$ of the quadcopter is commonly expressed in terms of the velocity components along each of the axes in the body-fixed coordinate frame. The components $u$, $v$, and $w$ correspond to the inertial velocity of the vehicle projected onto the $x_b$, $y_b$, and $z_b$ axes, respectively. On the other hand, the position of the quadcopter is usually measured and expressed in the inertial reference frame. Relating the translational velocity and position requires differentiation and a rotational transformation. Hence, the rotation matrix describing the relationship between the translational velocities in the body-fixed frame and those in the inertial frame is given by

$$
R = \begin{bmatrix}
C_{\phi}C_{\psi} & S_{\phi}S_{\theta}C_{\psi} - C_{\phi}S_{\psi} & C_{\phi}S_{\theta}C_{\psi} + S_{\phi}S_{\psi} \\
C_{\phi}S_{\psi} & S_{\phi}S_{\theta}S_{\psi} + C_{\phi}C_{\psi} & C_{\phi}S_{\theta}S_{\psi} - S_{\phi}C_{\psi} \\
-S_{\phi} & S_{\phi}C_{\theta} & C_{\phi}C_{\theta}
\end{bmatrix}
$$

(5.1)

in which $S_x = \sin x$ and $C_x = \cos x$. The rotation matrix $R$ is orthogonal since $R^{-1} = R^T$ where $R^{-1}$ is the rotation matrix from the inertial frame to the body-fixed frame. This is a kinematic relation in that it relates the derivative of position to velocity: forces or accelerations are not considered.

The relationships between angular positions $\phi$, $\theta$ and $\psi$ and angular rates $p$, $q$, and $r$ is also complicated by the fact that these quantities are defined in different coordinate frames. The body-fixed frame angular rates can be expressed in terms of the derivatives of the Euler angles, provided that the proper rotational transformations are carried out as follows. The transformation matrix for the angular velocities from the inertial frame to the body-fixed frame is

$$
\begin{bmatrix}
p \\
q \\
r
\end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_{\theta} \\ 0 & C_{\phi} & S_{\phi}C_{\theta} \\ 0 & -S_{\phi} & C_{\phi}C_{\theta}
\end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi}
\end{bmatrix}
$$

(5.2)

in which $S_x = \sin x$. The transformation matrix is invertible if $\theta \neq (2k-1)\pi/2, (k \in \mathbb{Z})$.

Inverting this expression yields

$$
\begin{bmatrix}
\dot{\phi} \\ \dot{\theta} \\ \dot{\psi}
\end{bmatrix} = \begin{bmatrix} 1 & S_{\phi}T_{\theta} & C_{\phi}T_{\theta} \\ 0 & C_{\phi} & -S_{\theta} \\ 0 & S_{\phi}/C_{\theta} & C_{\phi}/C_{\theta}
\end{bmatrix} \begin{bmatrix} p \\ q \\ r
\end{bmatrix}
$$

(5.3)

which expresses the derivatives of the three angular position states in terms of the angular positions $\phi$, $\theta$ and $\psi$ and the body rates $p$, $q$, and $r$.

5.1.3 Kinetic relations

To derive the dynamic equations of motion for the quadcopter, we will apply Newton’s second law first to the translational degrees of freedom and then to the rotational degrees of freedom. Newton’s laws hold in the inertial reference frame,
meaning the motion of the body of interest must be referenced to a fixed (i.e., inertial) frame of reference.

The dynamics of a generic 6 degrees of freedom rigid body takes into account the mass of the body $m$ and its inertia matrix $I$. The dynamics of a rigid body under external forces applied to the center of mass and expressed in the body-fixed frame are in Newton-Euler formalism

$$m\ddot{V}_b + \nu \times (mV_b) = F$$

$$I \ddot{\nu} + \nu \times (I\nu) = \tau$$

where $\dot{\nu}$ is the angular acceleration vector with respect to the body-fixed frame while $\dot{V}_b$ is the linear acceleration vector in the same frame, $\tau$ is the torque acting on the body, $F$ is the forces acting on the body and $m$ is the mass of the body. The moment of inertia $I$ with respect to the center of rotation (CoR) is approximated as a diagonal matrix

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

due to the symmetric structure of a quadcopter. The generalized force vector $\Delta$ can be defined according to

$$\Delta = [F^T \quad \tau^T]^T$$

Therefore it is possible to rewrite (5.4) in a matrix form

$$M_B \begin{bmatrix} \dot{V}_b \\ \dot{\nu} \end{bmatrix} + C_B(\nu, V_b)\nu = \Delta$$

where $M_B$ is a diagonal and constant inertia matrix and $C_B(\nu, V_b)$ is the Coriolis-centripetal matrix with respect to the body-fixed frame. $\Delta$ can be divided into four components according to the nature of the quadcopter aerodynamic.

$$\Delta = G_B(\eta) + O_B(\nu, \Omega) + E_B(\Omega) + D_B(V_b)$$

The first contribution is the gravitational effect $G_B(\eta)$ that is proportional to the mass by the gravitational constant $g$. This force acts at the center of mass and along the z-axis with respect to the inertial frame. Since there is no moment produced by gravity, the gravity force acting on the center of mass is given in the inertial frame by

$$G_B(\eta) = \begin{bmatrix} F_G^I \\ 0 \end{bmatrix}$$

where $F_G^I = [0 \quad 0 \quad mg]^T$. It could also be transformed into the body-fixed frame by the rotational matrix as

$$F_G^B = R_B^BR_G^I = \begin{bmatrix} -mg \sin \theta \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{bmatrix}$$
The second contribution is due to the gyroscopic effects produced by the propeller rotation. A common structure of a quadcopter is that two propellers rotate clockwise and the other two counterclockwise. Hence, there is an overall imbalance when the sum of the propeller angular speeds is not equal to zero. This nonzero propeller rates in combination with nonzero roll or pitch rates create a gyroscopic torque as

\[
O_B(v, \Omega) = \left[ 0_{3 \times 1} \right] - \sum_{i=1}^{4} J_O \left[ \nu \times 0 \right] (-1)^i \Omega_i \tag{5.11}
\]

where \( O_B(v) \) is the gyroscopic propeller matrix, \( 0_{3 \times 1} \) is a 3 \( \times \) 1 zero matrix, \( J_O \) is the total rotational moment of inertia around the propeller axes and \( \Omega_i \) is the angular speed of the \( i^{th} \) propeller. It can be seen that the contribution of the gyroscopic effect to the dynamics of the quadcopter is a linear torque which is proportional to the angular velocities of the quadcopter along the \( x \) and \( y \) axes in the body-fixed frame.

The third contribution is due to the thrust and torques produced from the propellers. As a quadcopter passes through the air, the interaction between the air and propellers create a thrust in the \( z \) axis and torques in the three rotational axes in the body-fixed frame. A simple model is obtained by assuming that the thrust and torques generated by the propellers are proportional to the squared propeller speeds according to

\[
E_B(\Omega) = \begin{bmatrix} 0 \\ 0 \\ T_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ bl(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ bl(\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{5.12}
\]

where \( T_z \) is the thrust produced in \( z \) axis of the body-fixed frame, \( \tau_x \), \( \tau_y \) and \( \tau_z \) are the generated torques in the corresponding axes of the body-fixed frame, respectively. Furthermore, \( b \) is the thrust coefficient, \( l \) is the distance from the center of mass to the center of a propeller and \( d \) is the torque coefficient. Better thrust and torque models explaining the underlying physics as the air passes through the propellers are given in the next section [Bristeau et al., 2009, Svacha et al., 2017].

The fourth contribution \( D_B(V_b) \) is the presence of a force acting along the propeller plane [Bangura and Mahony, 2012]. This force is proportional to the translational velocities of the quadcopter in the body-fixed frame and the scale factor is a unique aerodynamic property of the quadcopter. The fourth contribution is

\[
D_B(V_b) = \begin{bmatrix} -\lambda_1 \mu \\ -\lambda_1 \nu \\ 0 \end{bmatrix} \tag{5.13}
\]
where $\lambda_1$ is the drag coefficient, similar in both $x$ and $y$ axes in the body-fixed frame thanks to the symmetric structure of the quadcopter. A more detailed study of the drag force is given in the next sections.

To summarize, it is possible to describe the quadcopter dynamics considering these four contributions according to the equation

$$M \dot{V} + C_B(\nu, V_b)\nu = G_B(\eta) + O_B(\nu, \Omega) + E_B(\Omega) + D_B(V_b)$$

which can be rearranged as

$$m \dot{V}_b + m\nu \times V_b = mR_T g + E^F_B(\Omega) + D^F_B(V_b)$$

$$I \dot{\nu} + \nu \times (I \nu) = O^T_B(\nu, \Omega) + E^\tau_B(\Omega)$$

where the superscript $(.)^F$ and $(.)^\tau$ indicate the translational and rotational components.

### 5.2 Quadcopter submodels

The actual behavior of the quadcopter is well described in (5.15) in the previous section. However, it is often too complex to consider the full model of the quadcopter for estimation purposes. Instead, it is useful to consider particular submodels of the quadcopter.

#### 5.2.1 Longitudinal model

The right hand side of (5.14) contains two key aspects of the quadcopter’s dynamics. The second term is the main thrust, which is perpendicular to the rotor plane, and it does not affect the motion of the quadcopter in that frame. The last term is a drag force in the $x_{b}-y_{b}$ plane that is caused mainly by a phenomenon called blade flapping [Mahony et al., 2012]. The blade-flapping effect is due to the flexibility of the rotors and occurs primarily when the quadcopter is moving freely in the air. The effect of the relative speed of the blades with respect to free air divides the operating region of a propeller into two areas: a retreating and an advancing blade. The advancing blade has a higher relative velocity than the retreating one, which creates a force imbalance between the two areas. This results in a drag force acting in the opposite direction compared to the motion of the quadcopter’s body. Luckily, the mathematical expression is simple and a single term is sufficient to represent this effect. This term carries information about the horizontal linear velocities. Hence, the measurements of the acceleration can be used to estimate the linear velocities.

The above analysis reveals a possibility to design an estimator in order to be able to track mass changes of the quadcopter. Projecting (5.14) onto the $x_{b}-y_{b}$
5.2 Quadcopter submodels

Figure 5.2: The blade flapping effect. After tilting, the quadcopter starts moving horizontally and the drag force is accumulated acting along the frame of the quadcopter.

The drag effect, shown in Figure 5.2 is significant in all quadcopters, regardless of size [Faessler et al., 2018]. We can use this property to compute feedforward control terms directly from a reference trajectory to be tracked. The obtained feedforward terms are then used in a cascaded, nonlinear feedback control law that enables accurate agile flight with quadcopters, i.e., see [Abeywardena et al., 2013, Hoffmann et al., 2007, Leishman et al., 2014, Martin and Salaun, 2010]. Moreover, the dynamical model of a quadcopter subjected to linear rotor drag effects is shown to be differential flatness in its position and heading [Faessler et al., 2018]. Hence, the desired forces and torques can be computed directly from the predefined trajectory, simplifying the path planning problem.

plane in the body-fixed frame, i.e., assuming $r \approx 0$ and $w \approx 0$, yields

\[
\dot{u} = -g \sin(\theta) - \frac{\lambda_1}{m} u \tag{5.16a}
\]
\[
\dot{v} = g \cos(\theta) \sin(\phi) - \frac{\lambda_1}{m} v \tag{5.16b}
\]

where $\lambda_1$ is the drag coefficient. Interestingly, the above model does not have a standard input such as the thrust or control signal, which might require a non-standard way to address the estimation problem. Furthermore, since the quadcopter is designed symmetrically, the lateral dynamic in the $y_b$ axis is similar to the longitudinal one in $x_b$ axis. Hence, it is sufficient to consider only the roll motion of the quadcopter.

\[
\Sigma T
\]
\[
D
\]
\[
\Phi
\]
\[
mg
\]

\[
\text{Direction of body Motion}
\]

\[
\text{Blade Flapping}
\]
Figure 5.3: The blade-element momentum theory for the vertical dynamics. During vertical climbs, the drag effect can be neglected and the vertical thrust is due to the interaction between the propellers and the air flow.

5.2.2 Vertical model

Projecting (5.14) onto the $z_b$ axis yields

$$\dot{w} = -\frac{T_z}{m} - \frac{k_w}{m} w + g \cos \theta \cos \phi$$  \hspace{1cm} (5.17)

where $k_w$ is the drag coefficient.

The quadcopter has fixed-pitch propellers and the motors are assumed to produce the force

$$T_b = \begin{bmatrix} 0 & 0 & T_z \end{bmatrix}^T$$  \hspace{1cm} (5.18)

where $T_z$ is computed as $T_z = \sum_{i=1}^{4} T_i$. The standard model of the thrust produced by the $i^{th}$ propeller is given by $T_i = k_\Omega \Omega_i^2$, where $\Omega_i$ is the angular speed of the $i^{th}$ propeller and $k_\Omega$ is a positive constant. A better dynamic model that more accurately explains the underlying physics using blade-element momentum theory is given in [Hoffmann et al., 2007, Svacha et al., 2017] and is described by

$$T_i = c_1 \Omega_i^2 \left( c_2 (1 + \frac{3}{2} \mu_i^2) - \lambda_i \right)$$  \hspace{1cm} (5.19)

where $c_1$ and $c_2$ are positive constants which depend on the propeller’s structure. The advance ratio and inflow ratio shown in Figure 5.3 are

$$\mu_i = \frac{V_{hi} + v_{hi}}{R \Omega_i}$$  \hspace{1cm} (5.20a)

and

$$\lambda_i = \frac{V_{zi} + v_{zi}}{R \Omega_i}$$  \hspace{1cm} (5.20b)
respectively, where $R$ is the radius of a propeller, $V_{hi}$ and $V_{zi}$ are the horizontal and vertical velocities of the $i^{th}$ rotor in the body-fixed frame, and $v_{hi}$ and $v_{zi}$ are the horizontal and vertical induced velocities of the air stream through the $i^{th}$ rotor, respectively.

During vertical flight maneuvers, the quadcopter is controlled to fly vertically with zero roll and pitch angles. It is assumed that the horizontal speed $V_{hi} \approx 0$ and the horizontal induced velocity $v_{hi} \approx 0$ while the vertical speeds are assumed to be constant. This implies that the force from the $i^{th}$ motor can be approximated by

$$T_i \approx \bar{c}_1 \Omega_i^2 + \bar{c}_2 \Omega_i$$  \hspace{1cm} \text{(5.21)}$$

Assuming that the dynamics of the rotors are fast enough to be neglected, the relation between the calculated control signal and the angular speed, that is associated with the electrical control unit (ECU), is given by $\Omega_i = k_c u_{ci}$. Moreover, all rotors are assumed to be identical, which implies that the vertical force is given by

$$T_z = \sum_{i=1}^{4} k_1 u_{ci}^2 = k_1 u_t^2 + k_2 u_t$$  \hspace{1cm} \text{(5.22)}$$

where $k_1$ and $k_2$ are positive and unknown constants. In this work, we will also consider the standard thrust model given by $T_z = \sum_{i=1}^{4} k_1 u_{ci}^2 = k_1 u_t^2$.

### 5.3 Inertial measurement sensor

Recent developments of micro-electro-mechanical systems (MEMS) technology such as small but accurate Inertial Measurement Unit (IMU) sensor have helped to increase the maneuverability and flexibility of small and lightweight quadcopters. In this section, the accelerometer and gyroscope sensors are described, which are used intensively in this thesis.

The IMU provides measurements in a sensor-fixed coordinate system of the angular velocities and the linear accelerations in three dimensions. For simplicity, we assume that the sensor-fixed coordinate frame coincides with the body-fixed frame of the quadcopter except for a 180° rotation around the $x_b$ axis.

The gyroscope measures the rotational velocities of the body with respect to the inertial frame, and for instance, the sampled roll rate measurement can be modeled as

$$\dot{\phi}_t = \dot{\phi}_s + b_{\dot{\phi},t} + e_{\dot{\phi},t}$$  \hspace{1cm} \text{(5.23)}$$

where $\dot{\phi}_t$ is the roll rate to be sampled, $b_{\dot{\phi},t}$ is the gyroscope bias which is assumed to be constant during short flights and $e_{\dot{\phi},t}$ is zero mean measurement noise.

The accelerometers measure a combination of the inertial and gravitational accelerations. To understand this phenomenon, let us consider a mass that lies on a table. The forces acting on the mass will be the gravity pointing downward, and an equal and opposite force acting upward. These two forces cancel each other, which makes the total acceleration zero. However, the accelerometer will measure an acceleration equal to one $g$. When the table is removed, the mass will
fall freely and the accelerometer will measure zero acceleration. A more detailed explanation is described in an accelerometer tutorial [Leishman et al., 2014].

Denoting $a_{acc,t}$ as the acceleration that would be measured by an accelerometer, the sampled acceleration measurement is

$$a_{acc,t} = \dot{V}_b + \nu \times (V_b) - R^T g + e_{acc,t}$$  \hspace{1cm} (5.24)

where $\dot{V}_b$ is the acceleration of the body of the quadcopter in the body-fixed frame, $\nu \times V_b$ is the Coriolis acceleration, $R^T g$ is the gravitational acceleration projected to the body-fixed frame and $e_{acc,t}$ is the zero mean measurement noise. Another way to present (5.24) is in a component form as

$$a_x = \dot{u} + qw - rv + g \sin \theta + e_{ax}$$  \hspace{1cm} (5.25a)

$$a_y = -\dot{v} - ru + pw + g \cos \theta \sin \phi + e_{ay}$$  \hspace{1cm} (5.25b)

$$a_z = -\dot{w} - pv + qu + g \cos \theta \cos \phi + e_{az}$$  \hspace{1cm} (5.25c)

For small roll angle $\phi$ and $\theta \approx 0$, $p = r \approx 0$ in a horizontal flight, the lateral acceleration is given by

$$a_y = -\dot{v} + g \cos(\theta) \sin(\phi) + e_{ay} = \frac{\lambda_1}{m} v + e_{ay}$$  \hspace{1cm} (5.26)

which is proportional to the lateral velocity.

During the vertical flight, the quadcopter experiences the pitch-fixed thrust along the $z$-axis in the body-fixed frame and the gravity. Therefore, with $\phi = \theta \approx 0$, $\psi = \psi_0$ and $p = q = r \approx 0$, the acceleration measurement is

$$a_z = -\dot{w} + g = \frac{T_z}{m} + \frac{k_w}{m} w + e_{az}$$  \hspace{1cm} (5.27)

which also contains a linear term of the vertical velocity.

### 5.4 Pulse-width modulation

Pulse-width modulation (PWM) is one way to control the power supplied to the electrical motors in a quadcopter. For simplicity, the four propellers are assumed to be identical so that a single model can be used for all rotors’ dynamics. Without losing the main features of the rotor, the back EMF effect voltage $V_{emf}$ and output torque $T_r$ of $i^{th}$ rotor can be described as

$$V_{emf} = k_{emf} \Omega_i$$  \hspace{1cm} (5.28a)

$$T_r = k_r I_r$$  \hspace{1cm} (5.28b)

where $k_{emf}$ is the back EMF constant, $\Omega_i$ is the angular rate of $i^{th}$ rotor and $k_r$ is the torque constant. In a stationary condition, the mechanical power is equivalent to the electrical power dissipated by the back electromotive force (EMF) which means that $V_{emf} I_r = T_r \Omega_i$ or $k_r = k_{emf}$. 
Moreover, one of many possible choices of modeling the rotor dynamics is

\[ I_r = \frac{V - k_r \Omega_i}{R} \]  
(5.29a)

\[ T_r = J \frac{d\Omega_i}{dt} + b_r \Omega_i + T_L \]  
(5.29b)

where \( V \) is the applied voltage, \( J \) is the moment of inertia of the engine, \( T_L \) is the load torque, \( R \) and \( b_r \) are the armature resistance and damping coefficient, respectively. The transfer function from the applied voltage \( V \) to the angular velocity of a propeller is

\[ \frac{\Omega_i}{V} = \frac{k_r}{k_r + R(Jp + b_r)} \]  
(5.30)

where \( p \) is the differential operator.

The desired speeds of the rotors are obtained by adjusting the supply voltage \( V \) using an electronic speed controller (ESC). This ESC will receive the back EMF signal from the motor windings and send the PWM signal to adjust the effective voltage. This voltage is then proportional to the PWM signal, implying that the supplied voltage is linearly dependent on the PWM signal. Note that in a small and lightweight quadcopter, sensors to measure the angular velocities of the propellers are not available. However, it could be assumed that the dynamics of the rotors are fast enough to be neglected compared to that of the quadcopter, which implies

\[ \Omega_i \sim V \sim PWM_i \]  
(5.31)

Hence, the relation between the \( PWM_i \) and \( \Omega_i \) can be written as

\[ \Omega_i = k_{PWM} PWM_i \]  
(5.32)

where the coefficient \( k_{PWM} \) is assumed to be identical for all propellers.
In this chapter, the dynamic model (5.15) will be projected to the $x-y$ plane in the body-fixed frame and a longitudinal submodel is obtained. This submodel relates linearly the lateral acceleration and roll rate and describes a unique aerodynamic property of a quadcopter. The challenge of the estimation problem is due to the presences of measurement noises in both input and output. Furthermore, it can also be unclear which signal should be selected as the input or output.

6.1 Introduction

Payload is a crucial factor of quadcopters. If the quadcopter has a higher payload, more equipment can be attached on the quadcopter, which is useful in many applications. In the design phase of the quadcopter, the payload limitation has to be considered carefully since it depends on several factors, i.e., the number of onboard sensors, and the frame material. Furthermore, if the quadcopter is operated outdoors in a hazardous environment, its maximum payload also depends on the forces and torques caused by turbulence. Too much payload may cause the quadcopter to crash. It can therefore be interesting to monitor the payload, to allow the quadcopter to land safely if needed. In [Mellinger et al., 2011], the authors have found a way to estimate the payload of a quadcopter. The main contribution of this work is to design an aerial grasper and estimate the mass when the quadcopter uses the grasper to carry objects between different positions. Naturally, the mass of the quadcopter system changes when the quadcopter grasps different objects. This change is estimated based on the dynamic equations of the force-linear acceleration and the torque-angular acceleration relations of the quadcopter. The measurements are taken as the quadcopter is perturbed slightly around its hovering position. Recursive least squares estimation is then used to detect any payload change.
An alternative to the approach in [Mellinger et al., 2011] is to use an enhanced model of the quadcopter. In [Martin and Salaun, 2010], a drag-force enhanced model is derived. The drag force is created from the interaction of the propeller with the air stream. Fundamental blade element theory is used to derive the entire dynamic model of a quadcopter. However, the experimental validation of the model is only done for the force model describing the relations between the rotation angles and the translational velocities. Two controllers based on the drag-force enhanced model are proposed. The resulting hardware implementations are only evaluated qualitatively by pointing out that the systems are much easier to fly with these new controllers.

Furthermore, the rotor drag effect can also be used to enhance the trajectory tracking accuracy [Faessler et al., 2018, Mellinger and Kumar, 2011]. It has been proven that the dynamic model subjected to linear drag effects is differentially flat. Hence, the desired thrust and torque inputs can be computed reversibly from the flat outputs as the quadcopter position and heading. Based on this property, controls can be designed to improve the trajectory tracking performance with respect to prior unknown trajectories and estimated coefficients.

The work of [Martin and Salaun, 2010] is extended in [Leishman et al., 2014]. An observer is presented and the estimation problem is approached from a different viewpoint: improving the translational velocity and attitude estimates using Inertial Measurement Unit (IMU) measurements. The improvement is due to a better dynamic model that correctly explains the physics related to the measured acceleration. In principle, it is shown that the accelerometers directly measure the translational velocity. However, the model parameters are based on the estimated values of the payload as well as the drag coefficient which are typically obtained using least squares optimization. In order to achieve consistent estimates, this method requires accurate measurements of signals from a Vicon (Motion Capture) system. Therefore, this approach is usually not applicable when the Vicon system is missing, for example when the quadcopter is operating in outdoor environments.

Our contribution is to design another estimator based on the noisy measurements from an IMU and pilot commands to detect the change of the mass of a quadcopter. The estimator exploits the linear relationship between the linear speed and the angular velocity of the quadcopter. Due to the closed-loop effect, the colored measurement noises from accelerometer and gyroscope are fed back to the controller, which complicates the parameter estimation.

### 6.2 Approximate roll model of a quadcopter

Recall the model (5.16) of the quadcopter in $x$-$y$ plane of the body-fixed frame as

\[
\dot{u} = -g \sin(\theta) - \frac{\lambda_1}{m} u \\
\dot{v} = g \cos(\theta) \sin(\phi) - \frac{\lambda_1}{m} v
\]
Assuming small angles \((\sin(\phi) \approx \phi\) and \(\cos(\theta) \approx 1\)), the horizontal model can be linearized

\[
\dot{u} = -g\theta - \frac{\lambda_1}{m}u + \bar{\tau}_u \tag{6.2a}
\]

\[
\dot{v} = g\phi - \frac{\lambda_1}{m}v + \bar{\tau}_v \tag{6.2b}
\]

where \(\bar{\tau}_u\) and \(\bar{\tau}_v\) represent process noise and unmodeled dynamics. Note that the lateral dynamic on the \(y_b\) axis is similar to the longitudinal one. Hence, it is sufficient to consider only the roll motion (6.2b) of the quadcopter.

In the roll model (6.2b), the roll rate gyroscope provides \(\dot{\phi}\) which can be integrated directly to provide the input \(\phi\). The IMU also provides the lateral acceleration measurement \(a_y\) given in (5.26) which is linearly dependent on the lateral velocity. It has to be noted that \(a_y\) provides information about \(v\), which we can view as the output of the model (6.2b). However, both measurements from the accelerometer and gyroscope are noisy, which leads to the errors-in-variables problem. Another issue is that the disturbances in \(a_y\) and \(\dot{\phi}\) are correlated with each other as a consequence of the closed-loop control.

Combining (6.2b) and the roll rate measurement \(\dot{\phi}_s = \dot{\phi}_t + e_{\dot{\phi},t}\), given by the measurement model of an IMU, yields the transfer function

\[
a_y = \frac{\lambda_1}{m}v + e_{a_y} = \frac{\lambda_1 g}{p(p + \frac{\lambda_1}{m})} (\dot{\phi}_s - e_{\dot{\phi}}) + e_{a_y} + \bar{\tau}_v \tag{6.3}
\]

where \(p\) is the differential operator. Moreover, it should be noted that the total noise term \(e\) is colored such that a noise model might be required in the estimation step.

In fact, the measurements are taken in the discrete time domain and they need to be related to the model. Here, the transfer function is discretized using the Bilinear transformation \(p = \frac{2}{T_s} \frac{p - 1}{p + 1}\) which gives

\[
a_y = \beta_1 q^2 + 2q + 1 \dot{\phi}_s + e, \tag{6.4}
\]

where

\[
\beta_1 = \frac{\lambda_1 g T_s^2}{4 + 2 \frac{\lambda_1}{m} T_s} \quad \alpha_1 = -\frac{8}{4 + 2 \frac{\lambda_1}{m} T_s} \quad \alpha_2 = \frac{4 - 2 \frac{\lambda_1}{m} T_s}{4 + 2 \frac{\lambda_1}{m} T_s} \tag{6.5}
\]

and \(q\) is the forward shift operator and \(T_s\) is the sampling time.

Another way to represent the discrete-time lateral model (6.4) is to rewrite it in state-space form, taking only the lateral acceleration \(a_y\) as the output. With the state defined as \(x_t = [a_{y,t-1} \quad a_{y,t}]^T\), the input as \(u_t = \dot{\phi}^s_{t+1} + 2\dot{\phi}^s_t + \dot{\phi}^s_{t-1}\) and
the output as \( y_t = a_{y,t} \), we get

\[
\begin{bmatrix}
    x_{1,t+1} \\
    x_{2,t+1}
\end{bmatrix} = \begin{bmatrix}
    x_{2,t} \\
    -\alpha_1 x_{2,t} - \alpha_2 x_{1,t} + \beta_1 u_t
\end{bmatrix} + \bar{\tau}_x
\]

\( y_t = x_{2,t} + e_y \) \hfill (6.6a)

One of the main issues in estimating the parameter of the system (6.4) is the identifiability of the drag coefficient \( \lambda_1 \) and the mass \( m \). Since these two parameters are combined into a ratio, one dataset \( \{ a_{y,s}, \dot{\phi}_s \} \) can provide an estimate only of \( \frac{\lambda_1}{m} \). If a priori information of \( \lambda_1 \) is provided, the mass \( m \) can indeed be estimated using only one dataset. Another approach to overcome the identifiability problem is to use multiple datasets. In this case, several datasets will be collected. In the first dataset, the mass \( m \) is known and the physical parameter \( \lambda_1 \) is estimated. Based on that estimate of \( \lambda_1 \), the change of the mass can be computed using an estimate of the ratio \( a = \frac{\lambda_1}{m} \) obtained from the second dataset.

### 6.3 Numerical verification

In this section, we provide two experimental results. The first one is based on simulated data and the second one is based on experimental data collected from an AR Drone.

#### 6.3.1 Simulation study

In order to evaluate the abilities of the estimation approaches defined in Chapter 3, a Monte Carlo (MC) simulation has been conducted for the quadcopter’s lateral dynamics (6.4). The true value of the drag coefficient is chosen as 0.36. For each individual MC simulation, three datasets corresponding to the nominal mass (\( m = 455 \) g) and two additional masses (\( m = 510 \) g and \( m = 582 \) g) have been generated in Simulink with the sampling rate 200 Hz. Each dataset contains 11000 samples simulating 55 s of flight.

The simulation has been repeated 100 times with different noise realizations for \( \dot{\phi}_s \) and \( a_{y,s} \). Note that in the linear framework, the input and output are generated partly by the reference as well as the measurement/process noises in the closed-loop systems such that the model could be ARMAX for the white noise case or Box-Jenkins for the colored noise. Each realization is created by feeding a white noise signal with the standard deviation of 0.1 through a filter of order 6. The coefficients of the noise models of \( \dot{\phi}_s \) and \( a_{y,s} \) are chosen to capture approximately the dominant frequencies estimated from the spectrum of the experimental signals and are given in Table 6.1. The order of the noise model is selected to be 6.

Figure 6.1 shows the signals in a dataset. The reference signal \( \delta_t \) is created in such a way that the quadcopter is supposed to move similarly to the experiments described later in Section 6.3.2. Our goal is to assess the performance of the least-squares, refined IV and Extended Kalman filter of the roll model. Concerning
Algorithm 2, the true controllers are supposed to be unknown, and the transfer functions from $\delta_t$ to $a_y$ and $\phi_s(t)$ ($\hat{G}_\delta\hat{\phi}$ and $\hat{G}_{\hat{\delta}a_y}$) are estimated with excessive model orders to ensure that the model set contains the true system.

To evaluate the quality of the parameter estimates we use a benchmark criterion which corresponds to the Root Mean Square Error (RMSE) computed as

$$RMSE = \frac{1}{N_n} \sum_{n=1}^{N_n} \| \hat{\delta}_n - \delta \|$$

(6.7)

where $n$ is the index of the Monte Carlo run and $N_n$ is the number of Monte Carlo simulations. Additionally, we compute the change of the mass due to the cross-validation of several datasets. In fact, six mass estimation setups are obtained using three datasets corresponding to three different mass setups of the quadcopter. For each combination, the mass $\hat{m}_c$ and its standard deviation are computed using estimates of two ratios $\hat{a}_{ref} = \frac{\lambda_{1,mref}}{m_{ref}}$ and $a_c = \frac{\lambda_{1,mc}}{m_c}$. With a known $m_{ref}$, the mass $\hat{m}_c$ is estimated as $\hat{m}_c = \frac{\lambda_{1,mref}}{a_c}$ where $\hat{\lambda}_{1,mref}$ and $\hat{a}_c$ are the estimated values of $\lambda_{1,mref}$ and $\frac{\lambda_{1,mc}}{m_c}$, respectively. The approximated covariance $P_{\hat{m}_c}$ of $\hat{m}_c$ is computed using the Gauss approximation formula, $P_{\hat{m}_c} = (P_{\lambda_{1,mref} a_c}^2 + \lambda_{1,mref}^2 P_{\hat{a}_c})/\hat{a}_c^4$.

The estimates of $\lambda_1$ are given in Table 6.2. It turns out that the values obtained from the LS method are far from the true values. For the EKF method, since it is

### Table 6.1: The coefficients of the two noise models $H_{\phi_s}(q)$ and $H_{a_{y,s}}(q)$.

<table>
<thead>
<tr>
<th>Order</th>
<th>$H_{\phi_s}(q)$</th>
<th>$H_{a_{y,s}}(q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Numerator</td>
<td>Denominator</td>
</tr>
<tr>
<td>$q^0$</td>
<td>0.449</td>
<td>1</td>
</tr>
<tr>
<td>$q^{-1}$</td>
<td>0.277</td>
<td>1.604</td>
</tr>
<tr>
<td>$q^{-2}$</td>
<td>0.02359</td>
<td>0.818</td>
</tr>
<tr>
<td>$q^{-3}$</td>
<td>0.8013</td>
<td>0.4226</td>
</tr>
<tr>
<td>$q^{-4}$</td>
<td>0.4817</td>
<td>0.2457</td>
</tr>
<tr>
<td>$q^{-5}$</td>
<td>-0.03697</td>
<td>0.03886</td>
</tr>
<tr>
<td>$q^{-6}$</td>
<td>0.08703</td>
<td>0.03612</td>
</tr>
</tbody>
</table>

### Table 6.2: The average of the estimated $\lambda_1$ with standard deviation based on the MC simulations. The first column indicates the mass setup while the estimation results are shown in the 2nd – 4th columns.

<table>
<thead>
<tr>
<th>True mass</th>
<th>LS</th>
<th>EKF</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>455 g</td>
<td>818.3 ± 20.34</td>
<td>0.3598 ± 0.0543</td>
<td>0.3605 ± 0.0043</td>
</tr>
<tr>
<td>510 g</td>
<td>916.9 ± 22.77</td>
<td>0.3596 ± 0.0639</td>
<td>0.3605 ± 0.0052</td>
</tr>
<tr>
<td>582 g</td>
<td>1043.7 ± 25.88</td>
<td>0.3593 ± 0.0769</td>
<td>0.3611 ± 0.0062</td>
</tr>
</tbody>
</table>
Figure 6.1: An example of the simulated data. The reference signal $\delta_t$ is found in the top sub-figure, the roll rate $\dot{\phi}_s$ in the middle one and the bottom sub-figure shows the lateral acceleration $a_{y,s}$.

A recursive approach for estimating the state and parameters simultaneously, it is natural that it gives a different result compared to the LS and IV methods where the whole batch of measurements is used to estimate the interesting parameters. However, since the physical parameter $\lambda_1$ is supposed to be constant during the experiment, the last estimated state vector and state covariance from the EKF estimator in each MC simulation are used to compute the mean and covariance of $\lambda_1$. It turns out that the obtained value of $\lambda_1$ from the EKF is rather similar to the IV result for the three different mass setups.

The simulation results can also be analyzed using the root mean square error (RMSE) of the estimated $\lambda_1$ in Table 6.3. The smallest RMSE in all three mass setups are achieved by the IV estimator. Table 6.4 shows the estimates of the mass using the ratio $\frac{\lambda_1}{m}$. In detail, the LS method provides a biased estimate of $\lambda_1$ that results in a failure to detect the variation of the masses. The EKF and IV methods, on the other hand, provide accurate mass estimates. However, since the standard deviation of the $\lambda_1$ estimate affects that of the mass estimate, the IV estimator also provides the smallest standard deviation of the mass estimates in all combinations. However, it should be noted that the standard deviations provided by the EKFs can be changed by retuning the filters without affecting the actual estimators.

The RMSE of the mass estimates is shown in Table 6.5. The LS method gives the worst results due to the inaccurate estimate of $\lambda_1$. The IV method, on the
Table 6.3: The RMSE of the estimates of $\lambda_1$ based on the MC simulations. The first column indicates the mass setup while the estimation results are shown in the 2nd – 4th columns.

<table>
<thead>
<tr>
<th>True mass</th>
<th>LS</th>
<th>EKF</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>455 g</td>
<td>818.2</td>
<td>0.0043</td>
<td>0.0043</td>
</tr>
<tr>
<td>510 g</td>
<td>916.5</td>
<td>0.0051</td>
<td>0.0047</td>
</tr>
<tr>
<td>582 g</td>
<td>1043.4</td>
<td>0.0053</td>
<td>0.0047</td>
</tr>
</tbody>
</table>

Table 6.4: The average of the estimated masses and the approximated standard deviation using the ratio $\frac{\lambda_1}{m}$ based on the MC simulations. For each MC simulation, the mass $\hat{m}_{c,i}$ is estimated as $\hat{m}_{c,i} = \frac{\lambda_1 m_{\text{ref},i}}{a_{c,i}}$ and the covariance of $\hat{m}_{c,i}$ is approximated with $P_{\hat{m}_{c,i}} = (P_{\lambda_1 m_{\text{ref},i}} a_{c,i} + \lambda_1^2 P_{\lambda_1 m_{\text{ref},i}} a_{c,i}^2) / \hat{a}_{c,i}^4$ for the LS and IV methods.

<table>
<thead>
<tr>
<th>$m_{\text{ref}}$</th>
<th>$m_c$</th>
<th>$\hat{m}_c$ (LS)</th>
<th>$\hat{m}_c$ (EKF)</th>
<th>$\hat{m}_c$ (IV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>455 g</td>
<td>510 g</td>
<td>455.2 ± 16.0 g</td>
<td>510.4 ± 119.1 g</td>
<td>510.1 ± 9.6 g</td>
</tr>
<tr>
<td>510 g</td>
<td>582 g</td>
<td>456.4 ± 16.1 g</td>
<td>583.0 ± 152.8 g</td>
<td>581.1 ± 12.2 g</td>
</tr>
<tr>
<td>582 g</td>
<td>455 g</td>
<td>510.0 ± 17.9 g</td>
<td>454.8 ± 106.0 g</td>
<td>455.1 ± 8.5 g</td>
</tr>
<tr>
<td>582 g</td>
<td>510 g</td>
<td>511.4 ± 17.9 g</td>
<td>582.6 ± 162.1 g</td>
<td>581.1 ± 13.2 g</td>
</tr>
<tr>
<td>582 g</td>
<td>455 g</td>
<td>580.5 ± 20.4 g</td>
<td>454.4 ± 119.1 g</td>
<td>455.8 ± 9.6 g</td>
</tr>
<tr>
<td>582 g</td>
<td>510 g</td>
<td>580.6 ± 20.4 g</td>
<td>509.7 ± 141.8 g</td>
<td>510.9 ± 10.6 g</td>
</tr>
</tbody>
</table>

The other hand, gives a somewhat smaller RMSE of the mass estimate than the EKF.

6.3.2 Experimental results

In this section, the goal is to apply the estimation approaches to the real flight data of a quadcopter. Due to its inherent instability, the quadcopter must be operated in closed loop.

AR Drone

The Parrot AR Drone 2 is a popular quadcopter platform for research and education. It consists of a carbon-fiber structure, a plastic body, four high-efficiency brushless motors, sensors and a control board, two cameras and a removable hull, see Figure 6.2. The maximum translational speed of the quadcopter is 5 m/s and its battery provides enough energy for up to 13 minutes of continuous flight. The AR Drone is capable to communicate with the ground workstation via Wi-Fi, which lets the user control the vehicle with an external device. Moreover, the Wi-Fi communication provides access to preprocessed sensor measurements and images from onboard cameras stored in the transmitted messages.
Table 6.5: The RMSE of the mass estimates based on the MC simulations. For each MC simulation, the mass $\hat{m}_{c,i}$ is computed as $\hat{m}_{c,i} = \frac{\lambda_{1,m_{\text{ref}},i}}{\hat{a}_{c,i}}$ and $\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{m}_{c,i} - m_c)^2}$, where $N$ is the number of MC simulations.

<table>
<thead>
<tr>
<th>$m_{\text{ref}}$</th>
<th>$m_c$</th>
<th>RMSE (LS)</th>
<th>RMSE (EKF)</th>
<th>RMSE (IV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>455 g</td>
<td>510 g</td>
<td>55.7</td>
<td>9.4</td>
<td>9.0</td>
</tr>
<tr>
<td></td>
<td>582 g</td>
<td>129.0</td>
<td>10.3</td>
<td>9.8</td>
</tr>
<tr>
<td>510 g</td>
<td>455 g</td>
<td>56.1</td>
<td>8.3</td>
<td>8.0</td>
</tr>
<tr>
<td></td>
<td>582 g</td>
<td>71.3</td>
<td>11.7</td>
<td>10.5</td>
</tr>
<tr>
<td>582 g</td>
<td>455 g</td>
<td>126.2</td>
<td>8.0</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>510 g</td>
<td>71.6</td>
<td>10.3</td>
<td>9.3</td>
</tr>
</tbody>
</table>

Figure 6.2: AR Drone.

Experimental result

During the experiment, the first seconds of measurements are used to estimate the bias of the IMU when the AR Drone is on the ground. Afterwards, the AR Drone is flying and excited from the hovering position to move left-right with constant altitude and zero pitch angle. Figure 6.3 illustrates a typical dataset collected in the particular experiments. In the figure, the bias estimates are already subtracted from the measurements.

Three datasets have been collected, one with the nominal mass and two with different additional masses. The nominal mass of the AR Drone is 455 g while the new masses (nominal + additional) are 510 g and 582 g. These small additional masses are separated equally into four pieces placed underneath the propellers to ensure that they influence the aerodynamics of the quadcopter as little as possible.

Table 6.6 shows the estimate of $\lambda_1$ with its standard deviation. Note that $\lambda_1$ is an unknown parameter. However, due to the small deviations of the $\lambda_1$ estimates and the mass estimates in Table 6.7 for the IV algorithm, it is reasonable to believe that the obtained value of $\lambda_1$ from the IV method is actually close to the truth.
6.3 Numerical verification

Figure 6.3: An example of the experimental data. The top plot shows the reference signal $\delta_t$ while the lower plots show the roll angular speed $\dot{\phi}_s$ and the lateral acceleration $a_{ys,t}$, respectively.

In contrast, the LS method gives very large values of the drag coefficient, which is unrealistic. The EKF method provides a result similar to the IV method in the first two datasets. However, the estimate of $\lambda_1$ in the third dataset is not close to the first two even though the tuning parameters are kept as $R = 0.05$ and $Q = \text{diag}(0.000125, 0.9031, 0.000125) \times 10^{-6}$ for the three datasets.

The estimate of $\lambda_1$ is then used to detect the mass change of the quadcopter. Table 6.7 shows the mass estimate with its standard deviation in six different setups. For each setup, two estimated ratios of $a_{ref} = \frac{\lambda_1,m_{ref}}{m_{ref}}$ and $a_c = \frac{\lambda_1,m_c}{m_c}$ are used to compute $\hat{m}_c$ and its standard deviation. Note that the LS estimator provides inaccurate estimates of $\lambda_1$ in the three datasets. Hence, it fails to detect the mass change of the quadcopter. The EKF estimator is also unable to detect the changes in mass. However, the performance of the EKF estimator depends on the tuning of process and measurement noise variances, indicating that a better result might be obtained using another tuning. The best performance is achieved with the IV estimator in which the mass changes are detected successfully in all six combinations.
Table 6.6: The estimated $\lambda_1$ with standard deviation using experimental data. The first column indicates the mass setup while the estimation results are shown in the 2nd – 4th columns.

<table>
<thead>
<tr>
<th>True mass</th>
<th>EKF</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>455 g</td>
<td>0.3589 ± 0.1147</td>
<td>0.3599 ± 0.0021</td>
</tr>
<tr>
<td>510 g</td>
<td>0.3620 ± 0.1447</td>
<td>0.3641 ± 0.0017</td>
</tr>
<tr>
<td>582 g</td>
<td>0.5440 ± 0.1482</td>
<td>0.3607 ± 0.0009</td>
</tr>
</tbody>
</table>

Table 6.7: The estimated masses and the approximated standard deviation using the ratio $\lambda_1 m$ and the experimental data. The mass $\hat{m}_c$ is estimated as $\hat{m}_c = \frac{\lambda_1 m_{\text{ref}}}{d}$ and the covariance of $\hat{m}_c$ is approximated with $P_{\hat{m}_c} = (P_{\lambda_1 m_{\text{ref}}} \hat{\alpha}_c^2 + \lambda_1^2 m_{\text{ref}} P_{\hat{\alpha}_c}^4)/\hat{\alpha}_c^4$ for the LS and IV methods.

<table>
<thead>
<tr>
<th>$m_{\text{ref}}$</th>
<th>$m_c$</th>
<th>$\hat{m}_c$ (LS)</th>
<th>$\hat{m}_c$ (EKF)</th>
<th>$\hat{m}_c$ (IV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>455 g</td>
<td>510 g</td>
<td>1362.5 ± 54.9 g</td>
<td>505.6 ± 258.8 g</td>
<td>504.1 ± 3.9 g</td>
</tr>
<tr>
<td></td>
<td>582 g</td>
<td>2126.2 ± 78.9 g</td>
<td>384.4 ± 161.2 g</td>
<td>580.9 ± 3.8 g</td>
</tr>
<tr>
<td>510 g</td>
<td>455 g</td>
<td>170.3 ± 6.9 g</td>
<td>458.9 ± 234.8 g</td>
<td>460.3 ± 3.4 g</td>
</tr>
<tr>
<td></td>
<td>582 g</td>
<td>795.8 ± 25.7 g</td>
<td>387.3 ± 187.3 g</td>
<td>587.5 ± 3.2 g</td>
</tr>
<tr>
<td>582 g</td>
<td>455 g</td>
<td>124.5 ± 4.6 g</td>
<td>689.7 ± 289.6 g</td>
<td>456.1 ± 3.0 g</td>
</tr>
<tr>
<td></td>
<td>510 g</td>
<td>373.1 ± 12.1 g</td>
<td>766.4 ± 370.7 g</td>
<td>505.2 ± 2.8 g</td>
</tr>
</tbody>
</table>

6.4 Estimating the center of gravity of a quadcopter

The roll model can also be used to estimate the center of gravity (CoG) of a quadcopter. Ideally, the CoG of a quadcopter is designed to coincide with the intersection of its frame arms and the onboard IMU is supposed to be placed close to the CoG. Therefore, during any aggressive maneuver of the quadcopter, the second derivative of the Euler angles will not contribute significantly to the IMU measurements, which is beneficial since such contributions could cause side effects in inertial-based navigation methods.

However, due to the maneuverability and capability of quadcopters, they are used in a variety of applications, e.g., carrying various payloads such as cameras or other items. In these applications, the CoG is shifted into a new position and it can be useful to be able to estimate this shift.

We consider a quadcopter as in Figure 6.4. The position of the quadcopter in the inertial frame is defined as $\xi = [x \ y \ z]^T$. The roll, pitch and yaw angles $\phi$, $\theta$ and $\psi$ denote the orientation of the quadcopter. These Euler angles are collected in $\eta = [\phi \ \theta \ \psi]^T$.

The distance from the IMU to the shifted CoG which is denoted by $d$ is to be estimated. The translational model of the quadcopter is projected to the $x$-$y$
6.4 Estimating the center of gravity of a quadcopter

The inertial and the body coordinate frames of the quadcopter. The quadcopter is carrying a payload, which is modeled as a point mass.

plane which gives

\[ \dot{v} = g \sin \phi - \frac{\lambda}{\bar{m}}v \]  \hspace{1cm} (6.8)

where \( v \) is the velocity in the \( y \) direction of the body-fixed frame, \( \bar{m} = M + m \) with \( m \) as the mass of the quadcopter and \( M \) as the mass of the load.

This model contains a drag force that is linearly dependent on the velocity of the quadcopter, which has been used earlier in [Ho et al., 2017a] for mass estimation purposes. The measurements from the IMU are

\[ \dot{\phi}_m = \dot{\phi} + e_{\dot{\phi}}, \]

\[ a_y = g \sin \phi - \dot{v} + d\ddot{\phi} + e_{a_y} \]  \hspace{1cm} (6.9)

The first term in the acceleration measurement \( a_y \) is due to the gravity, the second term is the contribution from the lateral acceleration and the third term is the angular acceleration around the \( x_b \) axis.

The lateral model (6.8) can be linearized under a small angle assumption \( \sin(\phi) \approx \phi \) and \( \cos(\theta) \approx 1 \) which gives

\[ \dot{v} = g\phi - \frac{\lambda_1}{\bar{m}}v + \ddot{\tau} \]  \hspace{1cm} (6.10)
Estimation of the roll submodel

Figure 6.5: The experimented data. The reference signal $\delta_t$ is found in the top, the lateral acceleration $a_{y,t}$ in the middle and the roll rate $\dot{\phi}_t$ in the bottom.

where $\bar{\tau}$ represents both process noise and unmodeled dynamics.

Combining (6.9) and (6.10) yields

$$
\dot{\phi}_m - e_{\dot{\phi}} = \frac{p^2 + \frac{1}{m} p}{dp^3 + d\frac{1}{m} p^2 + g\frac{1}{m}} (a_y - e_{a_y}) + \bar{\tau} \quad (6.11)
$$

from the lateral acceleration measurement to the measured roll rate, where $p$ is the differential operator. This model can also be written as

$$
\dot{\phi}_m = \frac{p^2 + \frac{1}{m} p}{dp^3 + d\frac{1}{m} p^2 + g\frac{1}{m}} a_y + e_F = G_F(p) a_y + e_F \quad (6.12)
$$

Note that the total noise term $e_F$ typically is colored that it might thus be beneficial to include a noise model or a prefilter in the estimator. In fact, the measurements are taken in the discrete time domain and they need to be related to the model. Here, the transfer function $G_F(p)$ is discretized using $p = q^{-1}$ which gives

$$
G_F(q) = \frac{\alpha_1 q^{-1} + \alpha_2 q^{-2} + \alpha_3 q^{-3}}{1 + \beta_1 q^{-1} + \beta_2 q^{-2} + \beta_3 q^{-3}} \quad (6.13)
$$
Table 6.8: The estimated $\hat{\vartheta}_F$ of the forward model and $\hat{\vartheta}_I$ of the inverse model with their standard deviation using (4.14) using the basic IV method.

<table>
<thead>
<tr>
<th>Par</th>
<th>Forward ($\hat{\vartheta}_F$)</th>
<th>Inverse ($\hat{\vartheta}_I$)</th>
<th>std Inverse ($\hat{\gamma}_I$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_1$</td>
<td>0.1103</td>
<td>0.1104</td>
<td>$\frac{1}{\hat{\alpha}_1} = 9.0546$</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>-0.2204</td>
<td>-0.2206</td>
<td>$\frac{1}{\hat{\alpha}_2} = -1.9975$</td>
</tr>
<tr>
<td>$\alpha_3$</td>
<td>0.1101</td>
<td>0.1102</td>
<td>$\frac{1}{\hat{\alpha}_3} = 0.9975$</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>-2.9917</td>
<td>-2.9917</td>
<td>$\frac{1}{\hat{\beta}_1} = -27.0882$</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>2.9834</td>
<td>2.9834</td>
<td>$\frac{1}{\hat{\beta}_2} = 27.0132$</td>
</tr>
<tr>
<td>$\beta_3$</td>
<td>-0.9917</td>
<td>-0.9917</td>
<td>$\frac{1}{\hat{\beta}_3} = -8.9794$</td>
</tr>
</tbody>
</table>

Table 6.9: The estimates of the CoG $d$ with standard deviations obtained from the forward model and the inverse model, respectively.

<table>
<thead>
<tr>
<th></th>
<th>Forward model ($d_F$)</th>
<th>Inverse model ($d_I$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$4.5312 \pm 0.00696$</td>
<td>$4.5273 \pm 0.00695$</td>
<td></td>
</tr>
</tbody>
</table>

where

$$\alpha_1 = \frac{T_s}{d}, \quad \alpha_2 = (-2 + T_s \frac{\lambda}{m}) \frac{T_s}{d}, \quad \alpha_3 = (1 - T_s \frac{\lambda}{m}) \frac{T_s}{d}$$

$$\beta_1 = -3 + T_s \frac{\lambda}{m}, \quad \beta_2 = 3 - 2T_s \frac{\lambda}{m}, \quad \beta_3 = -1 + T_s \frac{\lambda}{m} + g \frac{\lambda T_s^3}{md}$$

The corresponding inverse model has the roll rate measurement as input and lateral acceleration measurement as output and can be written

$$a_y = \frac{\frac{1}{\hat{\alpha}_1} \beta_1 q^{-1} + \frac{\beta_2}{\hat{\alpha}_1} q^{-2} + \frac{\beta_3}{\hat{\alpha}_1} q^{-3}}{1 + \frac{\alpha_2}{\hat{\alpha}_1} q^{-2} + \frac{\alpha_3}{\hat{\alpha}_1} q^{-3}} q \hat{\phi}_m + e_I \quad (6.14)$$

Figure 6.5 shows the signals collected from an experiment with an AR Drone quadcopter. The reference signal $\delta_t$ is created in such a way that the AR Drone moves left and right with a constant altitude and zero pitch angle. The measured signals have been collected with a sampling time $T_s = 0.005s$.

Table 6.8 shows the estimate of $\hat{\vartheta}_F$ and $\hat{\vartheta}_I$ with their standard deviations. As expected, the estimates of $\hat{\vartheta}_F$ and $\hat{\vartheta}_I$ are similar. Furthermore, an estimate of $\hat{d}$ can be obtained using the estimate of $\hat{\alpha}_1,F$ or $\hat{\alpha}_1,I$. Using the estimated $\hat{\alpha}_1,F$ and its variance $\hat{P}_{\alpha_1,F}$, the estimate of $\hat{d}$ is $\hat{d}_F = \frac{T_s}{\hat{\alpha}_1,F}$ and its variance is computed as $\hat{P}_{d,F} = \frac{\hat{d}_F^4}{T_s^2} \hat{P}_{\alpha_1,F}$. A similar estimate can also be obtained from $\hat{\alpha}_1,I$ and $\hat{P}_{\alpha_1,I}$ as $\hat{d}_I = \frac{T_s}{\hat{\alpha}_1,I}$ and $\hat{P}_{d,I} = \frac{\hat{d}_I^4}{T_s^2} \hat{P}_{\alpha_1,I}$. These estimates are shown in Table 6.9, and as can
be seen there, the estimates of $d$ obtained from the forward and inverse models are similar.

Finally, if the position and mass of the load are known, the CoG of the unloaded quadcopter can also be computed. This estimate will not depend on the choice of a forward or inverse model either.
Estimation of the vertical submodel

In this chapter, the vertical dynamic model of a quadcopter is considered. Due to the nonlinearity of the quadcopter’s propellers, the generated thrust is a quadratic function of the angular speed of the propeller in the hovering stage. However, the operating quadcopter creates more complex aerodynamics and a refined thrust model is needed. It is also shown that the nonlinearity and closed-loop setting can be handled by using the instrumental variable method.

7.1 Introduction

Various types of more challenging flight scenarios have been reported in recent years, e.g., aggressive flight maneuvers, such as spins and flips [Lupashin et al., 2010] and dancing in the air [Dinh et al., 2017]. During these aggressive flights, the quadcopters are affected by various aerodynamic effects, such as blade flapping [Martin and Salaun, 2010] and thrust variation [Bristeau et al., 2009] due to changes in the induced velocities. Hence, performing extreme maneuvers result in high power consumption peaks and overall high power consumption, which drains the battery rapidly. The impacts of aerodynamics and battery lifetime need to be taken into consideration in the design of the electronic control unit (ECU). In particular, any change in the thrust generation has to be detected and compensated for. More precisely, to allow autonomous aerial vehicles to deal with unexpected incidents, it is necessary to detect changes/faults in the system before they lead to a complete system breakdown [Avram et al., 2017, Freddi et al., 2010, Sharifi et al., 2010]. For instance, it is necessary to track the parameter changes in the thrust model, which reflect the actuator lock and other potential actuator problems [Rupp et al., 2005].

In the previous chapter, a drag-force enhanced dynamic model is used to detect changes in the mass and the center of gravity (CoG) of the quadcopter. This
model includes the flapping effect, which describes how the propeller interacts with the air. The approach only focuses on the horizontal flight, which can occur during particular flight segments of a mission. However, a mission usually does not only contain horizontal flight between waypoints with constant altitude, but also segments of vertical flight with stationary attitude. During these segments of the maneuver, system identification algorithms could also be applied to estimate the physical parameters of the quadcopter [Ljung, 1999, Tischler and Remple, 2006]. This chapter considers the latter case. First, a refined version of the standard lumped parameters model for the propeller aerodynamic of the quadcopter is derived. Furthermore, the standard model and a refined model of the thrust are compared. The parameters of the two models are estimated based on the collected data using low-cost onboard sensors. This limited sensor setup complicates the estimation problem compared to the work of [Svacha et al., 2017] where several additional sensors such as motor angular speed sensors are available. Finally, the possibilities to perform battery and weight diagnosis by detecting changes in the parameters of the model are investigated.

7.2 Approximate model of the vertical dynamics of a quadcopter

Recall the vertical dynamic model (5.27) of a quadcopter

\[
\dot{w} = -\frac{T_z}{m} - \frac{k_w}{m} w + g \cos \theta \cos \phi
\]  

(7.1)

where \(k_w\) is the drag coefficient. The generated thrust is given in (5.22) as

\[T_z = \sum_{i=1}^{4} k_1 u_{ci}^2 + k_2 u_{ci} = k_1 u_t^2 + k_2 u_t\]

(7.2)

Combining two equations, and assuming small angles \((\cos(\phi) \approx \cos(\theta) \approx 1)\), the transfer function from \(u_{in1,t}\) and \(u_{in2,t}\) to \(a_{z,t}\) is given by

\[
a_{z,t} = \frac{p}{p + \frac{k_w}{m}} \left( \frac{k_1}{m} u_t^2 + \frac{k_2}{m} u_t \right) + \frac{\frac{k_w g}{m}}{p + \frac{k_w}{m}} + e_{z,t}
\]  

(7.3)

where \(p\) is the differential operator. The noise term \(e_{z,t}\) might be colored and correlated to the inputs \(u_t\) due to the closed-loop control. The model is discretized using the bilinear transformation \(p = \frac{2(q-1)}{T_s(q+1)}\) which gives

\[
a_{z,t} = \frac{2(q-1)}{(2 + \frac{k_w}{m} T_s)q - (2 - \frac{k_w}{m} T_s)} \left( \frac{k_1}{m} u_t^2 + \frac{k_2}{m} u_t \right) + \frac{\frac{k_w g}{m} T_s(q+1)}{(2 + \frac{k_w}{m} T_s)q - (2 - \frac{k_w}{m} T_s)} g + v_t
\]  

(7.4)
where \( v_t \) is noise, \( q \) is the forward shift operator and \( T_s \) is the sampling time. The discrete time model can now be rewritten in a regression form as

\[
a_{z,t} = q^T \delta + v_t
\]  

where \( q = [-a_{z,t-1}, u_t^2 - u_{t-1}^2, u_t - u_{t-1}, g] \). The elements of the parameter vector \( \delta = [\alpha, \beta_1, \beta_2, \beta_3]^T \) is given by

\[
\alpha = \frac{2 - k_w m T_s}{2 + \frac{k_w}{m} T_s}, \quad \beta_1 = \frac{2 k_1}{2 + \frac{k_w}{m} T_s}, \quad \beta_2 = \frac{2 k_2}{2 + \frac{k_w}{m} T_s}, \quad \beta_3 = \frac{2 k_w T_s}{2 + \frac{k_w}{m} T_s}
\]  

7.3 Numerical studies

In this section, we evaluate the applicability of the refined thrust model in two different scenarios. The first scenario studies the application of the IV method to estimate a nonlinear system in closed loop using simulated data. Experimental data collected from a AR drone is used in the second scenario.

7.3.1 Simulation study

In order to evaluate the possibility to use the instrumental variable method to estimate the parameters of the nonlinear vertical dynamic model, a Monte Carlo simulation with a closed-loop setup is performed. Three different noise levels related to the variance \( \sigma^2 \) are considered and 100 runs are performed for each noise level, see Table 7.1. For each run, 10000 samples have been simulated. Since the contribution of the gravity is just a constant offset to the measurements, we then will use the linear part of the simplified vertical dynamic model as

\[
G(p) = \frac{B(p)}{A(p)} = \frac{p}{p + \frac{k_w}{m} T_s}
\]  

where \( k_w = 0.6, m = 1 \). The parameters of the PI controller are selected to be \( k_p = 1 \) and \( k_i = 0.5 \), corresponding to the transfer function \( C(p) = \frac{p+0.5}{p} \). The nonlinear vertical dynamic model is

\[
y_t = G(p)(0.5u_t^2 + 1.5u_t) + e_t
\]  

The additive noise has a zero mean normal distribution \( e_t \sim N(0, \sigma^2) \) and the pilot command reference is chosen as a square wave with uniformly distributed amplitude between \([-1, 1]\).

Figure 7.1 illustrates an example of the identification data and the estimated parameters of the vertical dynamic models are shown in Table 7.1. It shows that the IV method provides accurate parameter estimates in three different noise variance scenarios.
Figure 7.1: An example of the simulation dataset with $\sigma^2_\varepsilon = 0.05^2$. The first subplot shows the pilot command and the control signal $u_t$ while the measured and simulated outputs are shown in the second subplot.

7.3.2 Experimental study

Several experiments with vertical flight maneuvers have been carried out with the AR Drone quadcopter shown in Figure 6.2. The IMU measurements and calculated rotor control signals are streamed wirelessly from the AR Drone to a ground computer via Wi-Fi connection. This ground computer sent the pilot command to control the quadcopter to perform a vertical flight with increasing/decreasing altitude, zero roll/pitch and constant yaw angles. All maneuvers are carried out indoors in order to reduce the effect of the turbulence to the quadcopter’s states. In detail, the first seconds of vertical acceleration measurements are used to estimate the bias of the IMU when the AR Drone is stationary. In total nine datasets have been collected, three datasets each for the nominal mass 455 g, the mass 530 g, and 586 g. Figure 7.2 shows a typical dataset. In this figure, the bias estimates are already subtracted from the measurements.

The performance of the methods are evaluated and validated by calculating the FIT of the simulated output, given in percent by

$$\text{FIT} = 100 \left( 1 - \frac{\|y_t - \hat{y}_t\|^2}{\|y_t - \mathbb{E}(y_t)\|^2} \right)$$

where $\|\cdot\|$ indicates the 2– norm of a signal, $\mathbb{E}$ is the expectation, $y_t$ and $\hat{y}_t$ are the measured and simulated outputs, respectively.
Table 7.1: Estimated continuous-time parameters $k_w$, $k_1$ and $k_2$ and their estimated standard deviations in the refined model using the IV method for three simulated measurement noise variances.

<table>
<thead>
<tr>
<th>Param</th>
<th>$\sigma_e^2 = 0.1^2$</th>
<th>$\sigma_e^2 = 0.05^2$</th>
<th>$\sigma_e^2 = 0.03^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_w$ (0.6)</td>
<td>0.6003 ± 0.0024</td>
<td>0.6001 ± 0.0013</td>
<td>0.5999 ± 0.0007</td>
</tr>
<tr>
<td>$k_1$ (1.5)</td>
<td>1.5004 ± 0.0049</td>
<td>1.5002 ± 0.0023</td>
<td>1.4999 ± 0.0013</td>
</tr>
<tr>
<td>$k_2$ (0.5)</td>
<td>0.4997 ± 0.0032</td>
<td>0.4999 ± 0.0017</td>
<td>0.5000 ± 0.0011</td>
</tr>
</tbody>
</table>

Table 7.2: Estimated continuous-time parameters $k_w$ and $k_1$ and their estimated standard deviations in the standard model using LS and IV methods for three different sets of data with different masses.

<table>
<thead>
<tr>
<th>Param</th>
<th>Mass 455 g</th>
<th>Mass 530 g</th>
<th>Mass 586 g</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_w$</td>
<td>LS 0.2588 ± 0.0870</td>
<td>0.3088 ± 0.1784</td>
<td>0.1812 ± 0.1477</td>
</tr>
<tr>
<td></td>
<td>IV 0.3040 ± 0.1340</td>
<td>0.4051 ± 0.0573</td>
<td>0.3121 ± 0.1047</td>
</tr>
<tr>
<td>$k_1$</td>
<td>LS 0.0437 ± 0.0380</td>
<td>0.0295 ± 0.0651</td>
<td>0.0612 ± 0.0525</td>
</tr>
<tr>
<td></td>
<td>IV 2.8344 ± 0.0148</td>
<td>2.6975 ± 0.0015</td>
<td>2.5099 ± 0.0039</td>
</tr>
</tbody>
</table>

For each mass case, three datasets have been used simultaneously to estimate the parameters of the standard or refined models. Tables 7.2 and 7.3 show the results using the LS and IV methods for both models, respectively. Note for the standard model in Table 7.2, both methods give highly uncertain parameter estimates.

The estimated parameters with standard deviations of the refined model are shown in Table 7.3. The LS method still gives very large estimated standard deviations, which indicates that the parameter values are unreliable. On the other hand, the IV estimates vary less between the datasets and the variations seem to match the estimated standard deviations. Hence, the IV method is a promising approach to estimate the parameters of the refined model in this closed-loop setup.

Figure 7.3 shows the simulated vertical acceleration using the estimated parameters of the standard and refined models obtained from the IV method for one typical validation dataset. The estimated parameters $\vartheta = [\alpha, \beta_1, \beta_2, \beta_3]^T$ obtained from the third set of data (586 g) are used to compute the coefficients of the discrete-time model associated with the first set of data (455 g), taking into account the mass difference. According to Figure 7.3, the refined model (57.10% model fit) gives a more accurate estimate of the vertical dynamics of the quadcopter than the standard model (33.30% model fit). Note that the result is selected among the best for both models using only one dataset. More detailed results can be found in Table 7.4, where the value in each row in the third or fourth columns shows the average of the fitting percentages for three validation
Estimation of the vertical submodel

Figure 7.2: An example of an experimental dataset. The first subplot shows the pilot command while $u_t$ and $a_z$ are shown in the second and third subplots, respectively. The $u_t$ is already normalized with factor $1/255$ since the maximum value of $u_{ci}$ is $255$ (8 bits memory $[0 – 255]$).

datasets $m_{val}$. It is clear that the refined model can capture the dynamics of the quadcopter better than the standard model in all cases.

If a model has been estimated from a reference dataset where the mass is known, the system identification approach from the previous section can be used to monitor changes in the mass. Table 7.5 shows the estimates of the masses using both the standard and refined models. In fact, six mass estimation setups are obtained using three datasets corresponding to three different masses of the quadcopter. For each combination, a mass estimate $m_c$ could be obtained using estimates of two ratios of $\beta_{1,\,ref} = \frac{k_{1,\,ref}}{m_{ref}}$ and $\beta_{1,\,c} = \frac{k_{1,c}}{m_c}$. With a known $m_{ref}$, the mass $\hat{m}_c$ is estimated as $\hat{m}_c = \frac{\hat{k}_{1,\,ref}}{\hat{\beta}_{1,\,c}}$ where $\hat{k}_{1,\,ref}$ and $\hat{\beta}_{1,\,c}$ are the estimated values of $k_{1,\,ref}$ and $\beta_{1,\,c} = \frac{k_{1,c}}{m_c}$, respectively. Performing similar calculations for the ratios $\alpha$ and $\beta_2$ gives three mass estimates. These mass estimates are averaged to achieve a combined mass estimate. As Table 7.5 shows, the refined model provides more accurate mass estimates than the standard model in all combinations.

7.4 Comparison with the horizontal dynamic model

An alternative approach to estimate of the mass change of a quadcopter is to use a lateral dynamic model. During a flight mission, there could be intervals where
Table 7.3: Estimated continuous-time parameters $k_w$, $k_1$ and $k_2$ and their estimated standard deviations in the refined model using LS and IV methods for three different sets of data with different masses.

<table>
<thead>
<tr>
<th>Param</th>
<th>Mass 455 g</th>
<th>Mass 530 g</th>
<th>Mass 586 g</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_w$</td>
<td>LS 0.2590 ± 0.0848</td>
<td>0.3068 ± 0.1475</td>
<td>0.1713 ± 0.1473</td>
</tr>
<tr>
<td></td>
<td>IV 0.3040 ± 0.0063</td>
<td>0.2904 ± 0.0083</td>
<td>0.3052 ± 0.0022</td>
</tr>
<tr>
<td>$k_1$</td>
<td>LS 0.1217 ± 0.1298</td>
<td>−0.1067 ± 0.3205</td>
<td>0.4957 ± 0.2078</td>
</tr>
<tr>
<td></td>
<td>IV 0.5198 ± 0.0482</td>
<td>0.5165 ± 0.0833</td>
<td>0.4921 ± 0.0217</td>
</tr>
<tr>
<td>$k_2$</td>
<td>LS −0.0988 ± 0.1248</td>
<td>0.1870 ± 0.2326</td>
<td>−0.6443 ± 0.1893</td>
</tr>
<tr>
<td></td>
<td>IV 1.5115 ± 0.0305</td>
<td>1.5574 ± 0.0565</td>
<td>1.5247 ± 0.0171</td>
</tr>
</tbody>
</table>

Table 7.4: Model fit values for validation data. The estimated parameters obtained from the IV method using $m_{ref}$ are projected to those using $m_{val}$ for both models and three sets of data $m_{val}$ are used for validation. The values on each row in the third and fourth columns show the average fitting percentages for three sets of data $m_{val}$ of the standard and refined models, respectively.

<table>
<thead>
<tr>
<th>$m_{ref}$</th>
<th>$m_{val}$</th>
<th>Standard model</th>
<th>Refined model</th>
</tr>
</thead>
<tbody>
<tr>
<td>455 g</td>
<td>530 g</td>
<td>8.83%</td>
<td>56.62%</td>
</tr>
<tr>
<td></td>
<td>586 g</td>
<td>−5.69%</td>
<td>55.90%</td>
</tr>
<tr>
<td>530 g</td>
<td>455 g</td>
<td>22.54%</td>
<td>56.02%</td>
</tr>
<tr>
<td></td>
<td>586 g</td>
<td>5.71%</td>
<td>55.80%</td>
</tr>
<tr>
<td>586 g</td>
<td>455 g</td>
<td>31.71%</td>
<td>56.95%</td>
</tr>
<tr>
<td></td>
<td>530 g</td>
<td>26.31%</td>
<td>56.89%</td>
</tr>
</tbody>
</table>

the quadcopter’s movements are either dominantly horizontal or vertical. Hence, it could be interesting to be able to estimate the mass from both horizontal and vertical excitations. In Chapter 6, an estimator has been designed to estimate the change of the mass of the quadcopter from horizontal movements. It is shown that the mass estimate error is about 8 g for all different experimental mass setups. Moreover, it can be seen from Table 7.5 that the use of the estimator based on the refined vertical model provides an estimate with similar accuracy. Hence, these two estimators can combine to estimate changes in the mass of the quadcopter during a large part of a flight maneuver.

7.5 Diagnosis and discussion

Any fault representing malfunctioning of a component of the quadcopter, for example as a result of a locked actuator, affects the performance of the overall
Figure 7.3: Measured (grey) and simulated outputs from the standard model (solid red) and refined model (solid blue).

Table 7.5: The average estimates of the masses using the ratios $\frac{k_1}{m}$, $\frac{k_2}{m}$, and $\frac{k_w}{m}$. The mass estimate $\hat{m}_c$ is the average of three estimates obtained from three ratios for the refined model and two ratios for the standard model.

<table>
<thead>
<tr>
<th>$m_{ref}$</th>
<th>$m_c$</th>
<th>$\hat{m}_c$ (Standard)</th>
<th>$\hat{m}_c$ (Refined)</th>
</tr>
</thead>
<tbody>
<tr>
<td>455 g</td>
<td>530 g</td>
<td>596.1 g</td>
<td>526.3 g</td>
</tr>
<tr>
<td></td>
<td>586 g</td>
<td>560.3 g</td>
<td>578.1 g</td>
</tr>
<tr>
<td>530 g</td>
<td>455 g</td>
<td>419.1 g</td>
<td>458.6 g</td>
</tr>
<tr>
<td></td>
<td>586 g</td>
<td>508.9 g</td>
<td>582.6 g</td>
</tr>
<tr>
<td>586 g</td>
<td>455 g</td>
<td>478.5 g</td>
<td>461.6 g</td>
</tr>
<tr>
<td></td>
<td>530 g</td>
<td>618.2 g</td>
<td>534.1 g</td>
</tr>
</tbody>
</table>

system. The fault has to be estimated using fault detection and isolation (FDI) and then fault tolerant control (FTC) can be used to retain the quadcopter’s maneuverability. In particular, with respect to the refined model, there are three different fault/change scenarios that could be investigated as shown in Table 7.6 and as described as follows:

- The first fault/change scenario may be the decrease in the battery voltage. In detail, the thrust coefficients $k_1$ and $k_2$ depend on the battery characteristics. The longer the flight time is, the more the battery state of charge decreases, which will result in a decrease of the ratios $\frac{k_1}{m}$ and $\frac{k_2}{m}$. On the other hand, the drag ratio $\frac{k_w}{m}$ will typically not be affected. An estimator could then be designed to monitor the state of the battery to alert the quadcopter to land automatically before the battery is completely drained.
### Table 7.6: The relations of the system changes and the ratios $\frac{k_w}{m}$, $\frac{k_1}{m}$, and $\frac{k_2}{m}$.

The two ratios $\frac{k_1}{m}$ and $\frac{k_2}{m}$ vary similarly with respect to different fault/change scenarios.

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Battery efficiency ↑</th>
<th>Drag parameter ↑</th>
<th>Mass variation ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{k_w}{m}$</td>
<td>–</td>
<td>↑</td>
<td>↓</td>
</tr>
<tr>
<td>$\frac{k_1}{m}$</td>
<td>↓</td>
<td>–</td>
<td>↓</td>
</tr>
<tr>
<td>$\frac{k_2}{m}$</td>
<td>↓</td>
<td>–</td>
<td>↓</td>
</tr>
</tbody>
</table>

- The dynamic model of the quadcopter might be changed when the quadcopter carries objects with a significant size between different waypoints. One possibility is that the two ratios $\frac{k_1}{m}$ and $\frac{k_2}{m}$ remain constant while the ratio $\frac{k_w}{m}$ varies. This should also be possible to detect and the controller can then use the knowledge of the parameter change to adapt to the overall system’s behavior.

- It can be seen that all three ratios $\frac{k_1}{m}$, $\frac{k_2}{m}$, and $\frac{k_w}{m}$ vary similarly when the mass of the quadcopter is changed. Hence, mass changes during and between missions should be possible to detect.

It is also possible to consider combinations of the scenarios mentioned above. In this case, it might be necessary to combine several approaches, e.g., the vertical modeling approach presented here and the horizontal modeling approach from Chapter 6.
In this thesis, we have dealt with aerodynamic modeling and estimation problems in terms of the lateral and vertical dynamics of a quadcopter. The lateral model describes the relationship between the roll rate and the lateral acceleration due to the interaction of the propeller blades and the air stream along the quadcopter’s frame. The model parameters are estimated using only measurements from an onboard Inertial Measurement Unit (IMU). The vertical model derived is a Hammerstein model with known nonlinear basis functions, which are derived based on a refined thrust equation that can capture the system dynamics better than a standard thrust equation.

The challenge of the roll submodel estimation problem, in Chapter 6, is the data fusion of the roll rate and lateral acceleration in order to determine the current state of the quadcopter. The state is fed back to the controller to stabilize the quadcopter, which creates correlations between these measured signals. The analysis shows that the estimation problem is similar to an error-in-variable closed-loop single-input single-output (EIV-CL-SISO) problem. The second application is to estimate the parameters of the vertical submodel, in Chapter 7. Hence, the framework from Chapter 6 is extended to study the multiple-input single-output (MISO) problem. In this application, the input is assumed to be exactly known while it has to be measured using an IMU in the first application. Furthermore, the theoretical aspects of the Instrumental Variable (IV) method in estimating the forward and inverse models of a linear system are studied. The main observation is that these estimates are equal, except for small numerical errors, also for finite data.

Several comparisons between the closed-loop refined IV, the least-squares (LS), and the extended Kalman Filter (EKF) methods have been carried out. Both simulations and the experiments described in Chapters 6 and 7 show that the IV method provides more accurate estimates of the parameters than the LS and EKF
methods in these closed-loop setups. However, the performance of the IV estimator depends on the choice of the instruments and the prefilter. Future work includes improving these components and using the IV method to recursively estimate the changes in parameters of the quadcopter, taking into consideration the flexibility of the noise model.

Since a nonlinear system can be approximated by a linear parameter-varying (LPV) model, more complex nonlinear models such as Wiener-Hammerstein or Hammerstein-Wiener structures can also be studied. Even though the recursive IV method is applied to the open-loop Hammerstein model in previous studies, its applicability on the closed-loop nonlinear block-oriented models requires further investigation. Finally, the fault detection and isolation application, see Section 7.5, can be developed further in the combination with several modeling approaches. This framework might also be extended to fixed-wing unmanned aerial vehicles (UAVs).
Aerodynamics of a quadcopter

A.1 Propeller aerodynamic

The quadcopter aerodynamic effects relate to the interaction between the quadcopter and the air. These effects can only be ignored at slow velocities. Already at moderate velocities, the impact of these effects resulting from the variation in airspeed is significant. More specifically, the overall aerodynamic lift force \( L \) and the drag force \( D \) can be derived by integrating the lift per surface increment \( dL \) and the drag per surface increment \( dD \), along with each rotor blade. At a point \( l \), \( dL \) and \( dD \) can be expressed as

\[
\begin{aligned}
dL &= \frac{1}{2} \rho U(l)^2 C_{La} \alpha(l) c dl \\
dD &= \frac{1}{2} \rho U(l)^2 (C_{D0} + C_{Di} \alpha(l)^2) c dl
\end{aligned}
\]

where \( \rho \) is the air density, \( U \) is the airspeed, \( c \) is the chord, \( dl \) is the surface increment, \( C_{La} \) is the lift coefficient, \( C_{D0} \) is the parasitic drag coefficient, \( C_{Di} \) is the lift-induced drag coefficient, and \( \alpha \) is the local angle of attack [Bangura and Mahony, 2012, Bristeau et al., 2009].

The lift force produced by a rotor is then obtained by integrating the lift per surface increment as

\[
LF = \rho c R^3 \Omega_r^2 C_{La} \left( \frac{a_0}{3} - \frac{\hat{w} + L(\epsilon_1 \omega_y - \epsilon_2 \omega_x)}{2R |\Omega_r|} \right)
\]

where \( R \) is the radius of the rotor blade, \( a_0 \) is the rotor’s average pitch angle at rest, \( \Omega_r \) is the rotation speed of the rotor, \( \omega_x \) and \( \omega_y \) are the angular velocities of the quadrotor about the x and y-axes respectively, \( \hat{w} \) is the sum of the induced velocity and the wind speed in the z-axis body-fixed frame of reference subtracted
by the velocity of the quadrotor in the $z$-axis body-fixed frame of reference, and $\epsilon_i$ takes on value $-1$, $0$, or $1$ depending on the rotor under consideration [Bristeau et al., 2009].

In a similar manner to the lift force, the drag moment has been derived by integrating the drag moment per surface increment as

$$DM = -\text{sgn}(\Omega_r)\rho cR^4\Omega_r^2\left(\frac{C_{D0}}{4} + C_{Di}\alpha \frac{2\bar{w}}{3R|\Omega_r|}\right) +$$

$$-\text{sgn}(\Omega_r)\rho cR^4\Omega_r^2\left(\frac{C_{La}}{R|\Omega_r|}\left(\alpha \frac{2\bar{w}}{3R|\Omega_r|}\right)\right)$$

(A.3)

whereas integrating the drag per surface increment produces the drag force, which has been decomposed into two main parts: a parasitic drag term

$$DF_0 = \left[\frac{1}{2}\rho cR^2C_{D0}\left(|\Omega_r|\bar{u} - \text{sgn}(\Omega_r)\bar{\Omega}_r\omega_x\right)\right]$$

$$\left[\frac{1}{2}\rho cR^2C_{D0}\left(|\Omega_r|\bar{v} - \text{sgn}(\Omega_r)\bar{\Omega}_r\omega_y\right)\right]$$

(A.4)

and an induced drag term

$$DF_i = \left[\frac{\rho cR^3C_{Di}\alpha\left(\frac{\bar{w}}{R^2}\bar{u} + \frac{\bar{\omega}}{3}\omega_x\right)\right]$$

$$\left[\frac{\rho cR^3C_{Di}\alpha\left(\frac{\bar{w}}{R^2}\bar{v} + \frac{\bar{\omega}}{3}\omega_y\right)\right]$$

(A.5)

where $\bar{u}$ and $\bar{v}$ is the difference between the wind velocity and vehicle velocity in the body-fixed frame $x$ and $y$ axes respectively and $\alpha_i$ is the rotor blade angle of attack when taking the induced velocity into account. More precisely, the parasitic drag is incurred as a result of the nonlifting surfaces of the quadcopter that is significant at high speeds [Bangura and Mahony, 2012].

In addition, a third drag force caused by the lift forces is also derived from the lift force per section increment integration. This force is the proportional to the speed $u$, $v$ and acts along the quadcopter’s body-fixed frame $x$ and $y$ axes. The expression of the force is

$$DF_L = \left[\frac{1}{2}\rho cC_{La}\left(R\alpha\bar{w}\bar{u} + \left(\text{sgn}(\Omega_r)R^2\bar{w} - \frac{R^3}{3}\alpha\bar{\Omega}_r\right)\omega_x\right)\right]$$

$$\left[\frac{1}{2}\rho cC_{La}\left(R\alpha\bar{w}\bar{v} + \left(\text{sgn}(\Omega_r)R^2\bar{w} - \frac{R^3}{3}\alpha\bar{\Omega}_r\right)\omega_y\right)\right]$$

(A.6)

Finally, the moment produced by the lift force of a rotor can be derived

$$LM = \left[\frac{\rho cR^4C_{La}}{8}\left(\frac{\bar{\omega}}{4R^2}\bar{u} + \bar{\Omega}_r\right)\right]$$

$$\left[\frac{\rho cR^4C_{La}}{8}\left(\frac{\bar{\omega}}{4R^2}\bar{v} + \bar{\Omega}_r\right)\right]$$

(A.7)

Since the quadcopter’s rotors are designed with fixed pitch, the lift force acts along the body-fixed frame $z$ axis, the drag moment also acts about the body-fixed frame $z$-axis, and the lift moment has the side effect on the longitudinal and lateral axes. The above propeller model was developed by applying two dimensional incompressible flow concepts in three dimensions.
As the quadcopter moves along the rotor frame, the advancing blade of the propeller has the higher relative speed than the preceding blade. This imbalance of velocities cause a net force against the motion of the quadcopter, which has been described in Chapter 5. The effect is called blade flapping since the difference in force will tilt the rotor plane slightly. This effect appears also in the small quadcopters that are very commonly used among researchers [Abeywardena et al., 2013, Faessler et al., 2018, Leishman et al., 2014, Martin and Salaun, 2010]. This effect can be modeled with the equation

\[ T_{def} = T \sin \kappa_1 \] (A.8)

where \( T \) is the thrust vector produced by a rotor, \( T_{def} \) is the thrust generated in the direction opposing the motion of the vehicle along the rotor plane, and \( \kappa_1 \) is the angle of the deflected thrust vector. The additional effect of the blade flapping is to create a moment about the hub of the rotor

\[ M_{bf} = \kappa_2 \kappa_1 \] (A.9)

where \( \kappa_2 \) is the stiffness of the blade [Hoffmann et al., 2007]. Note that in helicopter theory, several of the drag effects due to the variation of the airspeed such as the induced drag, the translational drag and so on, are combined into a single term called the rotor drag or H-force drag. [Bristeau et al., 2009] model this term for quadrotors using the equation

\[ F_H = \lambda V_p^\perp \] (A.10)

where \( F_H \) is the H-force drag exerted on the rotor, \( \lambda \) is a constant, and \( V_p^\perp \) is the velocity along the plane of the rotor [Martin and Salaun, 2010]. This H-force drag model enables the design of new control algorithms [Faessler et al., 2018] or state observers [Abeywardena et al., 2013].

## A.2 Other aerodynamic effects

The quadcopter can perform complex maneuvers in which the airspeed and angle of attack could change rapidly. As the result, two aerodynamic effects are observed: the ground effect and the vertical descent.

### A.2.1 Ground effect

The ground effect happens when the quadcopter is operating close (within 1/2 rotor diameter) to the ground. More precisely, the induced velocity is reduced which results in less induced drag and more vertical lift force. Therefore, the required supplied power to the rotors is also reduced, implying that the rotor efficiency increases.
A.2.2 Vertical descent

Depending on the vertical velocity in the descent phase, the flight behavior varies from normal operating state to Vortex Ring State (VRS), Turbulent Wake State (TWS) and eventually Windmill Brake State (WBS). It should be noted that the blade momentum theory cannot be used in these flight regimes due to the energy dissipation in the unsteady airflow.

The normal working state is that the quadcopter hovers or climbs axially. The tip vortices follow smoothly the quadcopter trajectories and the flow is periodic. The turbulence in this regime is assumed to be ignored.

For the vortex ring state, the rate of descent is equal to half the induced velocity at hover \( \approx 0.5v_h \). For slightly higher descent velocities, the air recirculates through the blade periodically in the rotor plane. To recover from this state, more power needs to be supplied to the rotors that helps to reduce the descent speed and blow away the vortex.

The turbulent wake state happens when the descent rate is higher compared to that in vortex ring state. If the descent rate equals the induced velocity \( \approx v_h \), there is no net airflow through the rotor disc. The tip vortices in this state look like wake behind the rotor that could be more turbulent and aperiodic. To exit this state, additional power from the rotors is required to ensure that the descent rate is decreased and the vortex is blown away.

The windmill wake state occurs when the rate of descent is higher, as much as more than twice the induced velocity \( \approx 2v_h \). The airflow in this state is well behaved (flow upward) and there is no trapped vortices or wake. The rotor can then rotate with any applied power, implying that the power will transfer to the air unlike the previous two states.
B.1 Quaternion rotations

A common way to represent the attitude of a quadcopter is the unit quaternion representation. The quaternions use a four-dimensional description of the orientation which is more difficult to visualize than Euler angles [Beard and McLain, 2012]. However, there are mathematical advantages to the quaternion representation. The Euler angle representation has a singularity where the roll and yaw angles are indistinguishable when the pitch angle $\theta$ is $\pm 90^\circ$. This issue is called the gimbal lock. More specifically, the inverse transformation from the rotation matrix $R$ to the Euler angles in this case cannot be determined uniquely since one degree of freedom in the rotation is lost, i.e., $\cos \theta = 0$ as $\theta = \pm 90^\circ$. This singularity problem is avoided using the quaternion representation which helps the orientation to be determined in all flight conditions, including extreme maneuvers such as rapid descent. Another advantage of the quaternion presentation is its computational efficiency. While the Euler angle formulation contains nonlinear trigonometric functions, the quaternion expression results in much simpler linear and algebraic equations.

In general, a quaternion $e$ can be represented as a 4-dimensional vector as

$$e = \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

(B.1)

where $e_0$, $e_1$, $e_2$ and $e_3$ are scalars. When a quaternion is used to represent an orientation, it is constrained to be a *unit quaternion*, or in other words, $\|e\| = 1$.

The unit quaternion vector $e$ is commonly separated into a scalar $e_0$ and a
vector \( e_1 i + e_2 j + e_3 k \),

\[
e_0 = \cos \frac{\Theta}{2} \tag{B.2}
\]

and

\[
v \sin \frac{\Theta}{2} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{B.3}
\]

where the rotation angle \( \Theta \) is defined around the axis specified by the unit vector \( v \). The scalar term \( e_0 \) of the quaternion vector then defines the amplitude of the rotation and the vector part defines the axis of the rotation.

With this brief description of the quaternion, we can see how a quadcopter orientation can be represented. The orientation from the inertial frame to the body-fixed frame using the quaternion framework can be described as a single rotation whereas the Euler representation needs three rotation about the \( x \), \( y \), and \( z \) axes.

### B.2 Quadcopter kinematic and kinetic equations

Using a unit quaternion to represent the quadcopter attitude, the rotation matrix \( R \) from the inertial frame to body-fixed frame and the quaternion \( e \) are related as

\[
R = \begin{bmatrix}
e_1^2 + e_0^2 - e_2^2 - e_3^2 & 2(e_1 e_2 - e_3 e_0) & 2(e_1 e_3 + e_2 e_0) \\
2(e_1 e_2 + e_3 e_0) & e_2^2 + e_0^2 - e_1^2 - e_3^2 & 2(e_2 e_3 - e_1 e_0) \\
2(e_1 e_3 - e_2 e_0) & 2(e_2 e_3 + e_1 e_0) & e_3^2 + e_0^2 - e_1^2 - e_2^2
\end{bmatrix} \tag{B.4}
\]

which leads to a reformulated expression of the kinetic and kinematic relations

\[
m \ddot{V}_b = -v \times (m V_b) + F \tag{B.5a}
\]

\[
\begin{bmatrix}
\dot{e}_0 \\
\dot{e}_1 \\
\dot{e}_2 \\
\dot{e}_3
\end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\
p & 0 & r & -q \\
q & -r & 0 & p \\
r & q & -p & 0
\end{bmatrix} \begin{bmatrix} e_0 \\
e_1 \\
e_2 \\
e_3
\end{bmatrix} \tag{B.5b}
\]

\[
I \dot{\nu} = -\nu \times (I \nu) + \tau \tag{B.5c}
\]

It can be seen that the quaternion representation will only modify the kinematic relations of the quadcopter.

### B.3 Conversion between Euler angles and quaternions

The Euler angles can be calculated from a quaternion vector as

\[
\phi = \arctan2\left(2(e_0 e_1 + e_2 e_3), \left(e_0^2 + e_3^2 - e_1^2 - e_2^2\right)\right) \tag{B.6a}
\]

\[
\theta = \arcsin\left(2(e_0 e_2 - e_1 e_3)\right) \tag{B.6b}
\]

\[
\psi = \arctan2\left(2(e_0 e_3 + e_1 e_2), \left(e_0^2 + e_1^2 - e_2^2 - e_3^2\right)\right) \tag{B.6c}
\]
where the two-argument arctangent operator \(\text{arctan2}(y, x)\) defines the value of the arctangent of \(y/x\) in the range \([-\pi, \pi]\) [Beard and McLain, 2012]. Only a single argument is required for the \(\text{arcsin}\) operator since the pitch angle is only defined in the range \([-\pi/2, \pi/2]\).

On the other hand, the quaternion elements are computed from the yaw, pitch, and roll Euler angles \((\psi, \phi, \theta)\) using

\[
\begin{align*}
 e_0 &= \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \quad \text{(B.7a)} \\
 e_1 &= \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \quad \text{(B.7b)} \\
 e_2 &= \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \quad \text{(B.7c)} \\
 e_3 &= \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \quad \text{(B.7d)}
\end{align*}
\]
Bibliography


Bibliography


Licentiate Theses
Division of Automatic Control
Linköping University