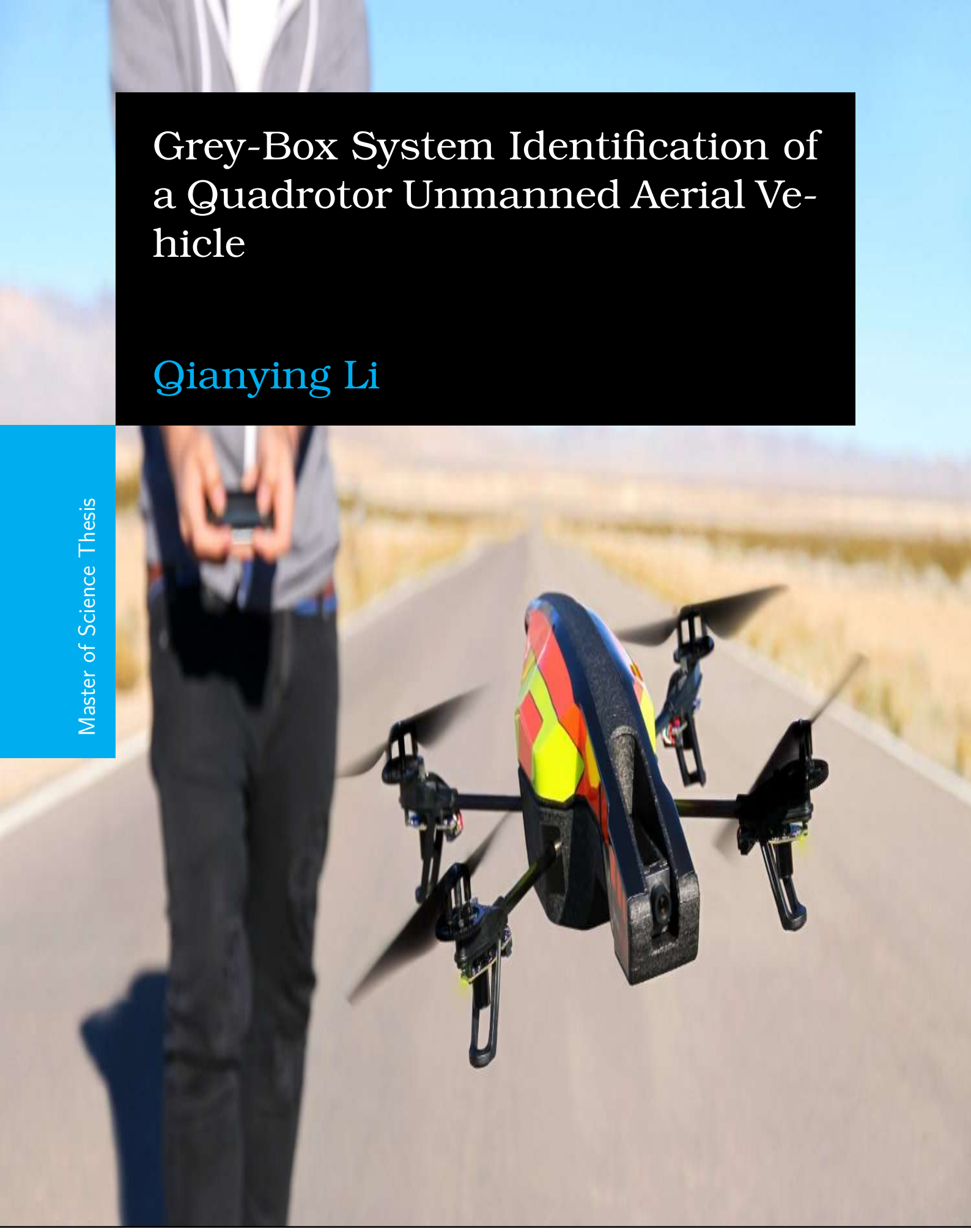
Grey-Box System Identification of a Quadrotor Unmanned Aerial Vehicle

Qianying Li

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Grey-Box System Identification of a Quadrotor Unmanned Aerial Vehicle

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft University of Technology

Qianying Li

August 20, 2014

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

This thesis focuses on the modelling and identification for auto-stabilization of a quadrotor (AR Drone 2) system through the implementation of a grey-box identification method for control purpose, which covers the results of the project addressing the grey-box identification of the nonlinear dynamic model for near-hover equilibrium. The methodology followed in this work addresses the specific issues of system modelling, model parameter estimation, simulation and bare-airframe system identification.

The nonlinear model of AR Drone 2 dynamics is derived from a Newton-Euler formulation and presented in East, North, Up (ENU) coordinates under Matlab/Simulink environment. Applying this model to real flight data requires obtaining the total thrust and torques as model inputs in advance thereby leading to the measurements of engine dynamics and aerodynamic coefficients of each propeller. The rest of the unknown parameters have been estimated by weighing and dimensioning each part of AR Drone 2. Then the approximation of the moment of inertia can be used as initial guess for Extended and Unscented Kalman filters (EKF and UKF) for parameter identification purposes. Meanwhile, the linearized partitioned system model structures were used in the fitting of a model to the real flight data which were collected with carefully devised experiments via Paparazzi. The prediction error method was used for the Box-Jenkins (BJ) polynomial models fitting.

On the one hand, although the EKF approach for parameter identification differed significantly when applied to real flight data and to simulation data, the UKF approach represented relatively reasonable results. It can be concluded that the parameter estimation from EKF and UKF is largely susceptible to the measurements of thrust and torque. In this project, the lack of or inaccuracy of measurements results in the less satisfactory performance of EKF and UKF for parameter estimation. On the other hand, the coupling effects owing to the "X-type" configuration did a disservice to roll and pitch channels. One simple solution is to change its configuration to "cross-type".

Acknowledgements

I am particularly grateful to the people for their timely help, otherwise there is no way to finish this project in time. I would like to do my best to fit all those to whom I own my gratitude.

Firstly, I would like to thank my supervisor dr.ir. M. Mazo Jr. and prof.dr. R. Babuska for their regularly assistance during this project. dr.ir. M. Mazo Jr. pointed out the right direction for me when the progress of the project was not quite smooth and encouraged me a lot.

Secondly, I would like to thank Dr.ir. C.C. de Visser, Dr.ir. G.C.H.E. de Croon and Ir. B.D.W. Remes for granting me the opportunity to study in Micro Air Vehicle Laboratory (MAVLab). Without the AR Drones and experimental setups available in the MAVLab, this process would have been extremely more difficult. Particularly, I would like to thank Erik van der Horst from the MAVlab for his assistance throughout the entire project, especially for the overtime help at night. Thanks to intern Clément Roblot and Ir. C. De Wagter at the MAVlab for their effort in developing the RPM sensor. I also acknowledge PhD candidates Sophie Armanini and Hann Woei Ho for their assistance during the whole flight tests. Without their help, I would have not been able to fly the AR Drone 2 at arena. Msc student Eric Poppe, thank you for selflessly sharing the experimental data with me when I had trouble in my own AR Drone 2. And of course I owe a great deal of thanks to the rest of the lab, they accompanied and helped me to get through the lonely hours of development in the lab.

Most of all I would like to thank my parents and family for their raising and being so supportive throughout my life, always believing in me and inspiring me, I would love you all forever.

Delft, University of Technology — Qianying Li
August 20, 2014

Contents

List of Figures

List of Tables

Glossary

List of Acronyms

SIM	Subspace Identification Method
PEM	Prediction Error Method
FEM	Filter Error Method
EEM	Equation Error Method
LTI	Linear Time-Invariant
MIMO	Multiple Input Multiple Output
SISO	Single Input Single Output
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
MREKF	Multi-Rate Extended Kalman Filter
MRUKF	Multi-Rate Unscented Kalman Filter
ENU	East-North-Up
GCS	Ground Control Station
RPM	Revolutions Per Minute
IMU	Inertial Measurement Unit
GPS	Global Position System
ECEF	Earth Center Earth Fixed
BJ	Box-Jenkins
OEM	Output Error Method
NASA	National Aeronautics and Space Administration

Chapter 1

Introduction

Micro aerial vehicles (MAVs) are small-scale aircraft which have been developing rapidly in recent years and are known as the best vehicles for the civil applications in the fields of surveillance, rescue missions, fire detection, agricultural spraying and reconnaissance. Thanks to their structural simplicity, relatively low cost and extraordinary manoeuvrability, great fascination and attention have been exerted on MAV design and control for development and research.

1-1 Motivation and objective

As one of the hottest areas for MAVs, the development of the cutting edge control system is still a challenging goal. To be a time-saving, cost effective approach, model-based control design is indispensable to success in the high-tech industries owing to its working efficiency; therefore, a model is required to synthesize the controller. The core of this work is to investigate the model of a quadrotor based on the grey-box system identification techniques for the control purpose. Grey-box system identification, the impetus behind the research presented in this thesis, is an effective approach for estimating the model.

1-2 Related work and proposed methods

System identification modeling obtains the actual quadrotor dynamics by using real flight data. In most cases, it is preferable to use black-box system identification methods due to the lack of knowledge about the hardware and software running on the quadrotor, or the lack of or inaccuracy of measurements. In Lotion's MSc thesis [20], Subspace Identification Method (SIM) was used under closed-loop conditions, but owing to the low fits, the model was considered not suitable for off-line controller design. Prediction Error Method (PEM) is also available for decoupled black-box submodels [10], the validation results seem valid for control purpose. In addition to the black box identification methods, the grey-box system identification method has become a popular approach to obtain practical Linear Time-Invariant (LTI) Multiple Input Multiple Output (MIMO) parametric models for quadrotors. Yuan and Katupitiya [27] proved Output Error Method (OEM) in SIDPAC as a useful tool for parameter estimation, and SIDPAC has been successfully used at National Aeronautics and Space Administration (NASA) Langley Research Center [27]; a model was identified in partitioned systems. Besides, an interesting comparison among the modern methods of aircraft parameter estimation [22] was investigated by using real flight data, which indicated that the performance of Filter Error

Method (FEM) prevailed over OEMs and Equation Error Method (EEM) in presence of turbulence and noise. Thus, the proposed system identification Methods in this work are the two Filter Error methods—Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), and this work mainly focuses on the nonlinear model identification.

1-3 Overview

The block diagram below shows the procedure for nonlinear model identification.

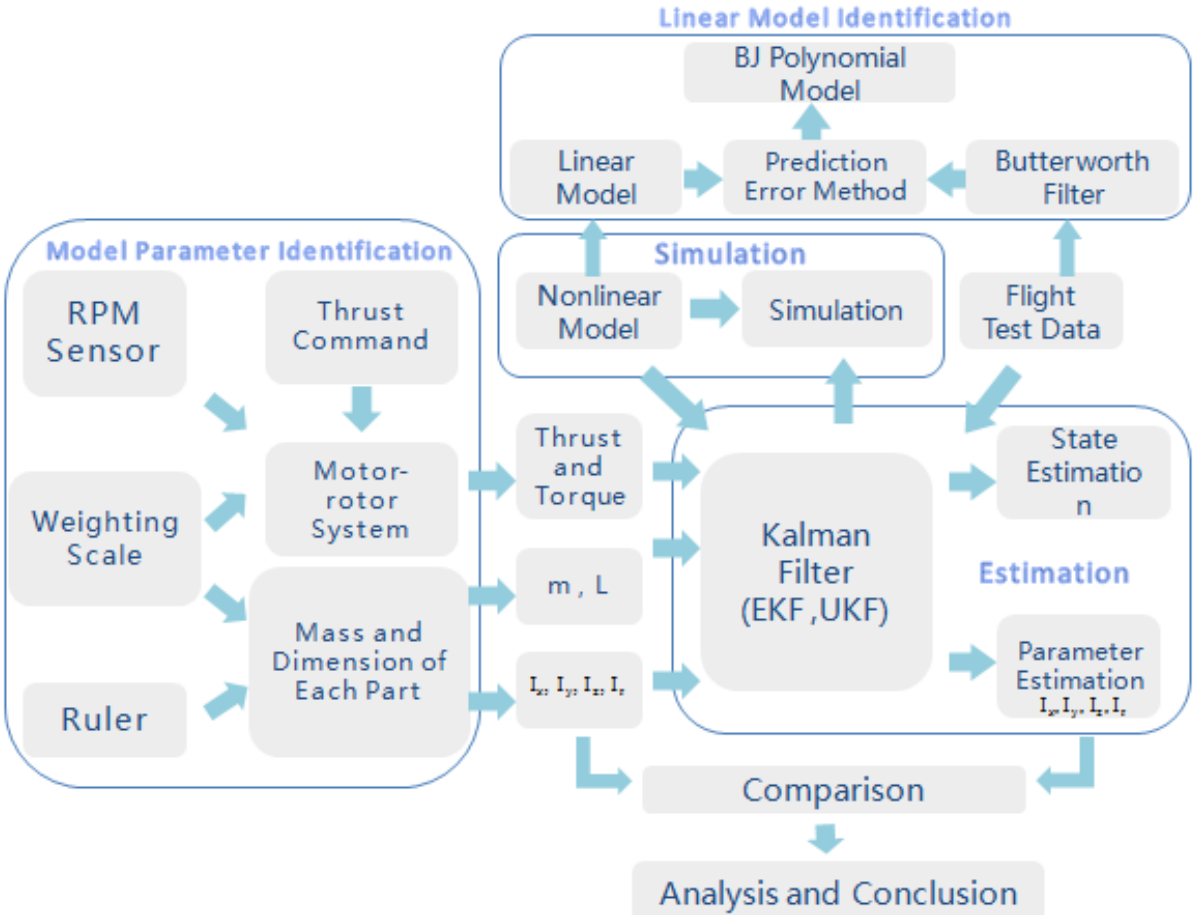


Figure 1-1: The block diagram of the identification procedure for nonlinear model

The whole block diagram can be divided into three categories, they are the model parameter identification, simulation and system identification, which are the main parts of this work and will be described in this thesis in more detail.

1-4 Outline

This thesis is structured as follows:

- Chapters 2 introduces the modeling of the quadrotor and provides the derivation of the nonlinear and linearized models based on Newton-Euler formalism, then followed by the introduction of the main experimental setups in Chapter 3.
- Chapter 4 estimates the model parameters; special attention is given to the motor-rotor system and basic measurements of AR Drone 2.

- Chapter 5 focuses on the identification algorithms needed to estimate unknown parameters. EKF and UKF as well as Multi-Rate Unscented Kalman Filter (MRUKF) are adopted in this chapter. Next chapter illustrates the quadrotor simulator based on a nonlinear model; the goal of this Matlab-Simulink program is to test the correctness and performance of the algorithms for parameter identification purpose.
- Chapter 7 shows the implementation of the proposed algorithms on real flight data. The experimental results of the identification algorithms are reported, and the constraints and uncertainties are discussed in depth. Chapter 8 concludes with the performance and the results of the work and proposes solutions to improve the accuracy of parameter estimation, thereby obtaining a more precise model.
- More information on this work can be found in literature report [19].

Chapter 2

System Modelling

Aero-elastic effects, internal dynamics of engine, flexibility of the wings and the whole set of changing variables render the modelling of the complete dynamics of AR Drone 2 (the quadrotor used in this project) highly challenging and unmanageable. Moreover, since AR Drone 2 has four rotors, which can be considered even more nonlinear than other rotorcraft despite its symmetrical mechanics. For the control purpose, a simplified model with main features must be considered and formed by the substantially decreased states and inputs. In the literature study, four models based on two different approaches (Euler-Lagrange and Newton-Euler) are discussed in depth. Consequently, this chapter is concise and only encompasses the models which are used in this project, more details on these can be found in [19].

2-1 Nonlinear and linear Newton-Euler model for an X-type AR Drone

This project introduces an "X-type" AR Drone 2 flying configuration, considering two frontal engines and two rear engines. The AR Drone 2 dynamical model equations are based on the Newton-Euler formalism, where the nonlinear dynamics are obtained in body-fixed and East-North-Up (ENU) inertial frames, see Fig. 2-1. The AR Drone 2 is controlled by the angular speeds of four separate electric motors. Each motor produces a thrust and torque, and the combination of these thrusts and torques generates the total thrust, the roll torque, the pitch torque and the yaw torque. Given that the motors of AR Drone 2 can only turn in a fixed direction, the produced thrust force T_i is always positive, the sub-index denotes the serial number of the rotor in Fig. 2-1 . In trimmed flight (near-hover), with the rotation arrangement in Fig. 2-1, both gyroscopic effects and aerodynamic torques have a tendency to cancel.

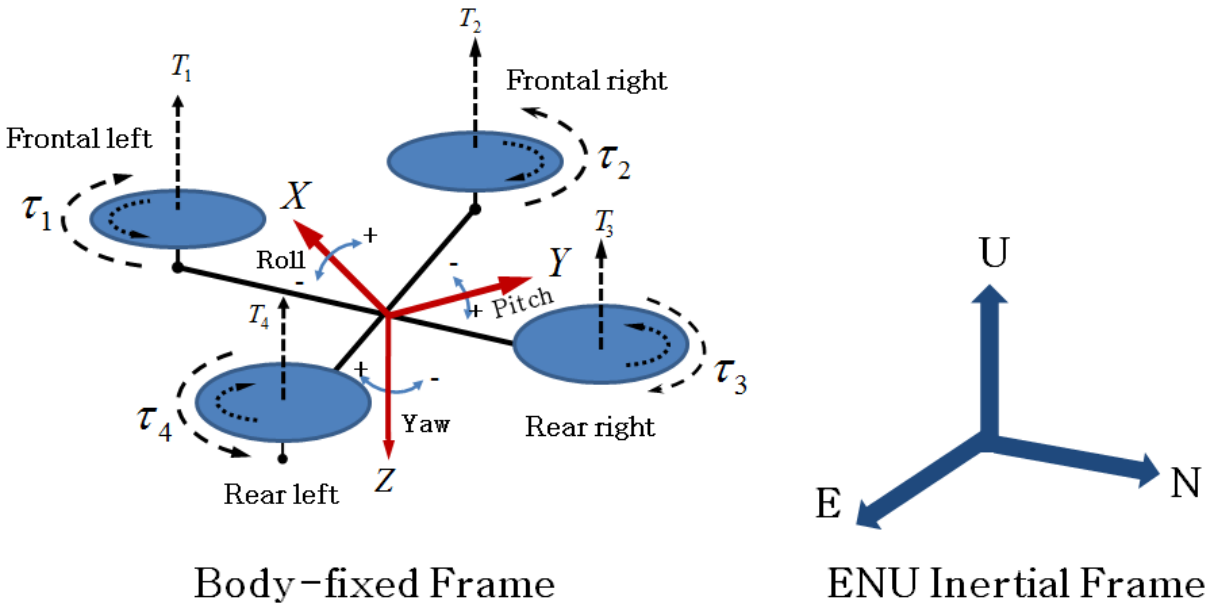


Figure 2-1: The quadrotor in an inertial frame. T_i and τ_i represent the thrust and torque of motor i

Let $\{X, Y, Z\}$ describe the body-fixed frame and $\{E, N, U\}$ represent the inertial reference frame. The position vector of the center of mass of the AR Drone is representing the position coordinates of the vehicle relative to the ENU, which is denoted by $\xi = \begin{pmatrix} x & y & z \end{pmatrix}^T$, the orientation vector with respect to the ENU is described as $\eta = \begin{pmatrix} \phi & \theta & \psi \end{pmatrix}^T$, where ϕ, θ and ψ are the roll, pitch, yaw Euler angles, respectively. The rotational speed of AR Drone relative to body-fixed frame is expressed by $\Omega = \begin{pmatrix} p & q & r \end{pmatrix}^T$. Then the full nonlinear dynamics of the AR Drone is given by [15]:

$$m\ddot{\xi} = -mgU + RF \tag{2-1}$$

$$I\dot{\Omega} = -\Omega \times I\Omega + \tau \tag{2-2}$$

where m and g are the total mass and gravitational constant respectively, R is the plane rotation matrix (see Appendix B) that associates the inertial frame with body-fixed one, F denotes the total force applied to the quadrotor and has one component in the Z direction. I and τ denote the inertia matrix and the total applied torque. The plane rotation matrix and the rotational velocities matrix can be found in Appendix B. Let $\begin{pmatrix} T_1 & T_2 & T_3 & T_4 \end{pmatrix}^T$ represent the thrust generated by each rotor and $\begin{pmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 \end{pmatrix}^T$ the corresponding torque. Assume the torque τ_i generated by each rotor is proportional to its thrust: $\tau_i = C_M T_i$. Then the input matrix for the nonlinear system can be expressed as:

$$\begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ L & -L & -L & L \\ L & L & -L & -L \\ -C_M & C_M & -C_M & C_M \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{2-3}$$

Where L denotes the distance between the center of mass and the center of the rotor. The model of AR Drone 2 is formed by considering all the significant effects and a minimum number number of inputs and states, but the main features are reserved for designing the control laws. The dynamical model of AR Drone 2 is formulated based on the following assumptions.

Assumptions:

1. The structure of AR Drone 2 is rigid and highly symmetrical.

2. The propellers are rigid, the thrust is parallel to the axis of the rotor, thus flexibility of the propellers is neglected .
3. The centre of gravity of AR Drone 2 and the origin of the body frame (cross center) are assumed to coincide.
4. No turbulence and airflow through the rotor indoors which guarantees that all the coefficients are constant (static trust and torque).
5. AR Drone. 2.0 flies higher than one rotor diameter, which ensures that the ground effect could not influence the flight data (in real flight tests, AR Drone 2 flies at an altitude of more than 2 meters except take-off and landing).
6. Last but not least, **the model holds for near hovering**, and neglects the blade flapping, change in advance ratio and effects of translational lift.

Using Eq. 2-1 to Eq. 2-3, the nonlinear dynamical model can be represented as:

$$\begin{cases} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{z} &= v_z \\ \dot{v_x} &= -(\sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi))\frac{T}{m} \\ \dot{v_y} &= -(\cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi))\frac{T}{m} \\ \dot{v_z} &= \cos(\theta)\cos(\phi)\frac{T}{m} - g \\ \dot{\phi} &= p + \sin(\phi)\tan(\theta)q + \cos(\phi)\tan(\theta)r \\ \dot{\theta} &= \cos(\phi)q - \sin(\phi)r \\ \dot{\psi} &= sin(\phi)\sec(\theta)q + \cos(\phi)\sec(\theta)r \\ \dot{p} &= \frac{I_y - I_z}{I_x}qr + \frac{I_r}{I_x}q\Omega + \frac{\tau_\phi}{I_x} \\ \dot{q} &= \frac{I_z - I_x}{I_y}pr - \frac{I_r}{I_y}p\Omega + \frac{\tau_\theta}{I_y} \\ \dot{r} &= \frac{I_x - I_y}{I_z}qp + \frac{\tau_\psi}{I_z} \end{cases} \tag{2-4}$$

where $\Omega = \omega_{rot,2} + \omega_{rot,4} - \omega_{rot,1} - \omega_{rot,3}$ is the counter clockwise residual Revolutions Per Minute (RPM), resulting in rotor gyroscopic effect, especially, in the case of aggressive manoeuvres. $\omega_{rot,i}$ is the angular velocity for the i-th rotor. Linearising this model under the hypothesis of small angular values gives the linearised model:

$$\begin{cases} \ddot{z} &\approx \frac{T}{m} - g \\ \ddot{\phi} &\approx \frac{\tau_\phi}{I_x} \\ \ddot{\theta} &\approx \frac{\tau_\theta}{I_y} \\ \ddot{\psi} &\approx \frac{\tau_\psi}{I_z} \end{cases} \tag{2-5}$$

Both the nonlinear and linear models are formed in the equations. For the better understanding of the dynamic system, the state-space models are introduced afterwards.

2-2 The state-space model description

The state-space structure of the AR Drone can be written as:

$$\dot{x} = Ax + Bu + Lw \tag{2-6}$$

$$y = Cx + Mv \tag{2-7}$$

where x is the system state and u is the control input, while w and v are the process and measurement noise. The state vector x is $x = [x\ y\ z\ v_x\ v_y\ v_z\ \phi\ \theta\ \psi\ p\ q\ r]^T$ and the control vector is $u = [T\ \tau_\phi\ \tau_\theta\ \tau_\psi]^T$. This leads to a state-space structure of the following dynamics, controller matrices as well as measurement matrix representing the bare airframe model dynamics.

Nonlinear state space model

$$\dot{x} = \begin{bmatrix} 0_{3\times3} & 1_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & L_1 \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & L_2 \end{bmatrix} x + \begin{bmatrix} 0_{3\times1} & 0_{3\times3} \\ L_3 & 0_{3\times3} \\ 0_{3\times1} & 0_{3\times3} \\ 0_{3\times1} & L_4 \end{bmatrix} u - g \begin{bmatrix} 0_{3\times1} \\ L_5 \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} \tag{2-8}$$

where

$$L_1 = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix}, \quad L_2 = \begin{bmatrix} 0 & \frac{I_y - I_z}{I_x} r + \frac{I_r}{I_x}\Omega & 0 \\ \frac{I_z - I_x}{I_y} r - \frac{I_r}{I_y}\Omega & 0 & 0 \\ \frac{I_x - I_y}{I_z} q & 0 & 0 \end{bmatrix} \tag{2-9}$$

$$L_3 = \begin{bmatrix} -(\sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi))\frac{1}{m} \\ -(\cos(\psi)\sin(\theta)\cos(\phi) + \sin(\psi)\sin(\phi))\frac{1}{m} \\ \cos(\theta)\cos(\phi)\frac{1}{m} \end{bmatrix}, \quad L_4 = \begin{bmatrix} \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix}, \quad L_5 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2-10}$$

For the measurement matrix, the responses of AR Drone for a sequence of flight, starting from slow speed motors rotation to high speed motors rotation, followed by take-off, hover, doublet (can be replaced by other manoeuvre) and landing are analyzed. The measurement data for positions, velocities, Euler angles and angular velocities are read by sensors and then recorded. The exhaustive illustration about the working principle of each sensor and the corresponding measurement data can be found in the next chapter. Thus the measurement matrix is only a twelve by twelve diagonal identity matrix.

$$\begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \\ \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} \tag{2-11}$$

Linear state space model

According to Eq. 2-5, the linear state-space model can be obtained in a simple form; therefore, this model is not described in this part.

In this chapter, the AR Drone dynamical model equations are modelled based on a Newton-Euler formalism. It is also possible to use the Euler-Lagrange approach to obtain the same set of equations [21].

Chapter 3

Experimental Setup

In this project, the AR Drone 2, the sensors and associated codes are supported by Micro Air Vehicle laboratory (MAVlab). Due to the help of MAVlab, all the flight tests and measurement can be accomplished in time. In this chapter, the AR Drone's main sensors are presented, which are fundamental to identify the robot's position, rotational speed, translational speed and attitude. Moreover, it is worth noting the importance of Paparazzi. Without this software, the data cannot be read and logged.

3-1 AR Drone 2

Figure 3-1: Parrot AR.Drone 2 without protective indoor hull

The Parrot AR Drone 2 is one of the most widely used commercially available quadcopters and represents the latest development of the renowned high tech quadcopter, which is easily controlled by Wi-Fi using a tablet or smartphone. It is a self-stabilising four bladed helicopter mounting two HD cameras on board through which the surrounding environment can be viewed while flying in real-time. Depending on the Wi-Fi signal strength, the flying range

is about 50 meters, and the lifetime of supplied lithium battery is of 16 minutes. For the further reading about Parrot AR. Drone 2, the product specification can be found on [2].

3-2 MAV-Laboratory and Paparazzi

3-2-1 MAV-Laboratory

The MAV-Laboratory is part of the Delft Robotics Institute, the goal of which is to study and create small autonomous air vehicles. MAVlab provided the AR Drone to do the flight test and gave help in developing and performing the parameter estimation technically. All the available measurement setups are created by MAVlab for the research purpose.

3-2-2 Paparazzi

Paparazzi is a free and open-source software and hardware project encompassing an exceptionally powerful and versatile autopilot system for fixed-wing aircraft as well as multicopters. This powerful open source has already been adapted as an autopilot for use with the Parrot AR Drone. Hence, it is able to perform GPS-based autonomous indoor and outdoor flight. Paparazzi can set the airframe, create the personalized flight plan, and on top of that, the real-time data can be logged, it is also free to add the sensors and write the scripts. Thanks to the convenience of Paparazzi, the project could be processed smoothly. In this work, the simple flight plan is written for the flight test. Since the flight test is not very aggressive, this can be done directly in Paparazzi Ground Control Station (GCS) by dragging the waypoint to the desired place. The logged data in 'xml' file format is in a Matlab friendly format which can be read expediently. For more information about how to use Paparazzi, an official website is available to learn [3].

3-3 Sensors

In our setup the exploited sensors are an accelerometer and a gyroscope, other sensors are a Global Position System (GPS) module (connected with camera system), a RPM sensor (soldered and used), and a magnetometer (soldered). The used sensors will be described explicitly in the following subsections.

3-3-1 RPM sensor

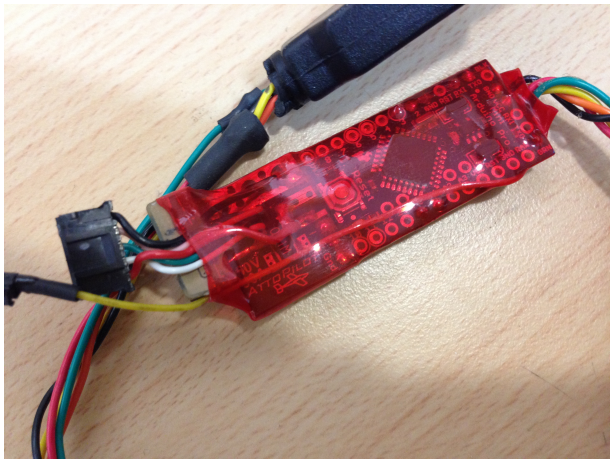


Figure 3-2: Arduino Pro Mini - 5V/16MHz (ATMEGA328)

The Arduino microcontroller is used to measure the time between pulses caused by the rotation of the engine, from which the RPM of the engine can be deduced. Specifically, the engines of AR Drone are controlled by power, not by voltage. The voltage is constant and equal to the battery voltage, but the power depends on how much of the time the engine is "on", which is called pulse width modulation. These pulses are supplied to engines and can be tracked. The way in this project to track the number of these pulses is simply to use a wire and connect it to a pin of the engine controller that is read out by the Arduino microcontroller. The value read out by Arduino microcontroller is a pulse that is measured once or several times during every round of the engine. Essentially speaking, this value is the ticks of CPU, which runs at 8MHz. By the calculation, the number of the pulses measured per round of engine is assumed to be four, the reason is that the RPM computed by four then matches the specification of the AR Drone, which is 28500 RPM in hover. Consequently, the relation between rotational speed and the measured value is given by:

$$\omega_{eng}\ (rad/s) = 2000000/(value\ from\ RPM\ sensor) * 2 * pi \tag{3-1}$$

with ω_{eng} the engine speed.

3-3-2 IMU

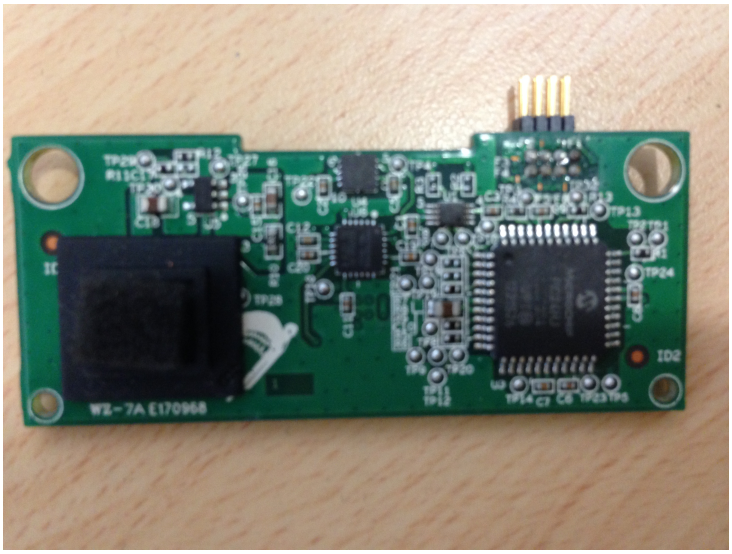


Figure 3-3: Navigation board

IMU is the acronym of Inertial Measurement Unit (IMU), which identifies a sensor capable of measuring the orientation of a body through inertial sensors [14]. In particular, fast and highly dynamic motions can be precisely estimated over short periods of time by fusing rotational velocity and linear acceleration measurements provided by IMU. The IMU provides measurements of the airframe accelerations in body frame and angular rates (3 axis accelerometer +/- 50mg precision and 3 axis gyroscope 2000 degree/second precision). The IMU collects the accelerometer data via a dedicated interface, while synchronizing data sampling at a user defined rate [4].

3-3-3 Vision system and GPS

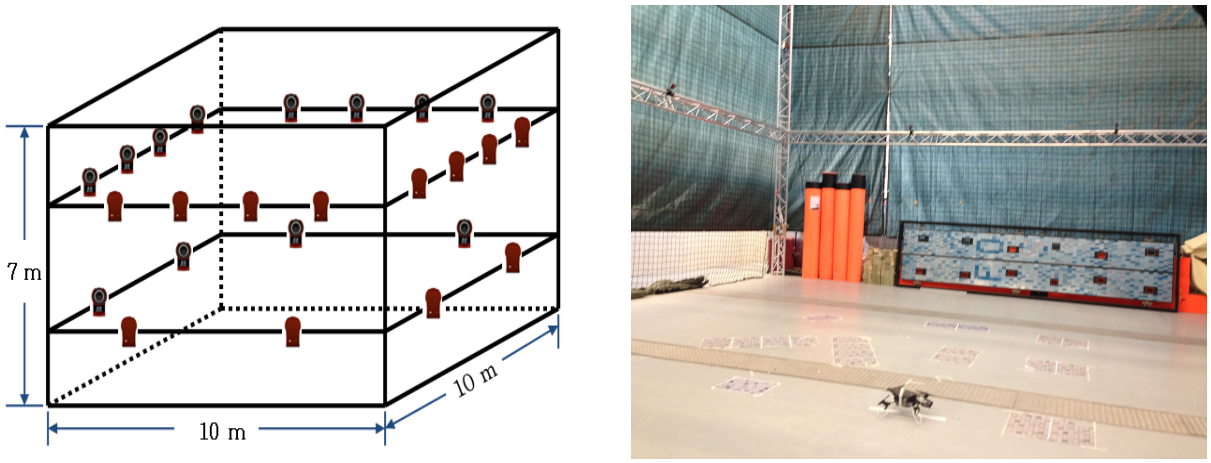


Figure 3-4: The distribution of cameras in the flight arena

In the figures above is a large (10m*10m*7m) arena for heterogeneous autonomous robots, including the flying and ground platforms. This space is equipped with twenty-four cameras with a motion capture systems that cover almost the entire volumetric arena. From each flight test, an enormous quantity of data can be collected, which includes the motion capture data and the telemetry information from each robot, etc. The motion capture software used is OptiTrack's unified platform Motive, and the type of OptiTrack camera is Flex 13, which is a medium volume motion capture camera with excellent precision . The table below shows the specifications of Flex 13 [5].

Table 3-1: Specifications of the Flex cameras. FOV and FPS stand for Field Of View and for frames per second.

Type	Resolution	Frame Rate	Horizontal FOV	Filter Switcher	Interface	No. of LEDs
Flex 13	1280×1024	120 FPS	$42°, 56°$	Optional	USB 2.0	28

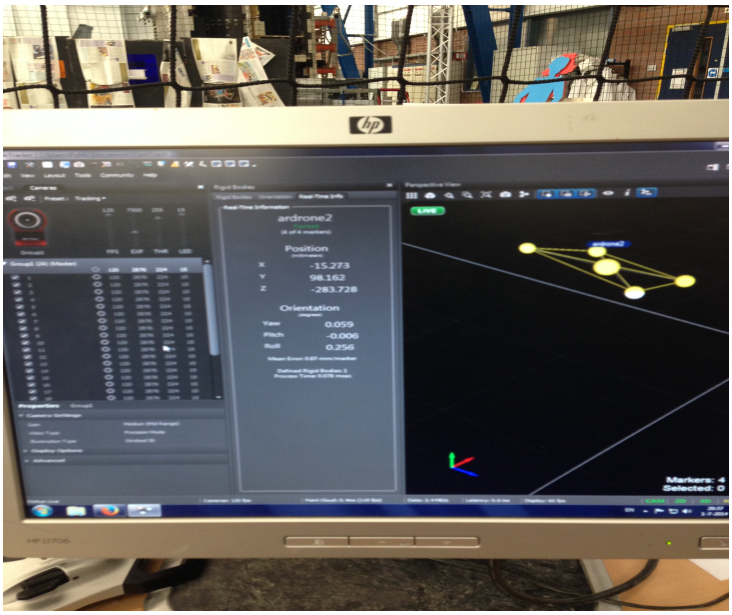


Figure 3-5: Motion capture software: Motive

Getting accurate position of objects in the observable OptiTrack can be achieved as follows. Using the attached twenty-eight infrared LEDs, the pulsed infrared light is sent out by Flex 13, and is then be reflected by markers (LED lights or reflexible markers) on AR Drone 2 and detected by Flex 13. Afterwards, the 3D video of the objects will be streamed to Motive. It should be noted that the bad quality and the small number of markers will do the tracking a disservice. Even regardless of the reflection quality of markers, the tracking quality is largely susceptible to the light from outdoors through the windows as well. Knowing the position of those markers from the perspective of several cameras, the actual 3D position of the AR Drone 2 in the arena can be computed using triangulation. This information of 3D position is then sent via multicast to Paparazzi by using the NatNet protocol, in this fashion, the GPS's data is collected .

3-4 Sensor fusion

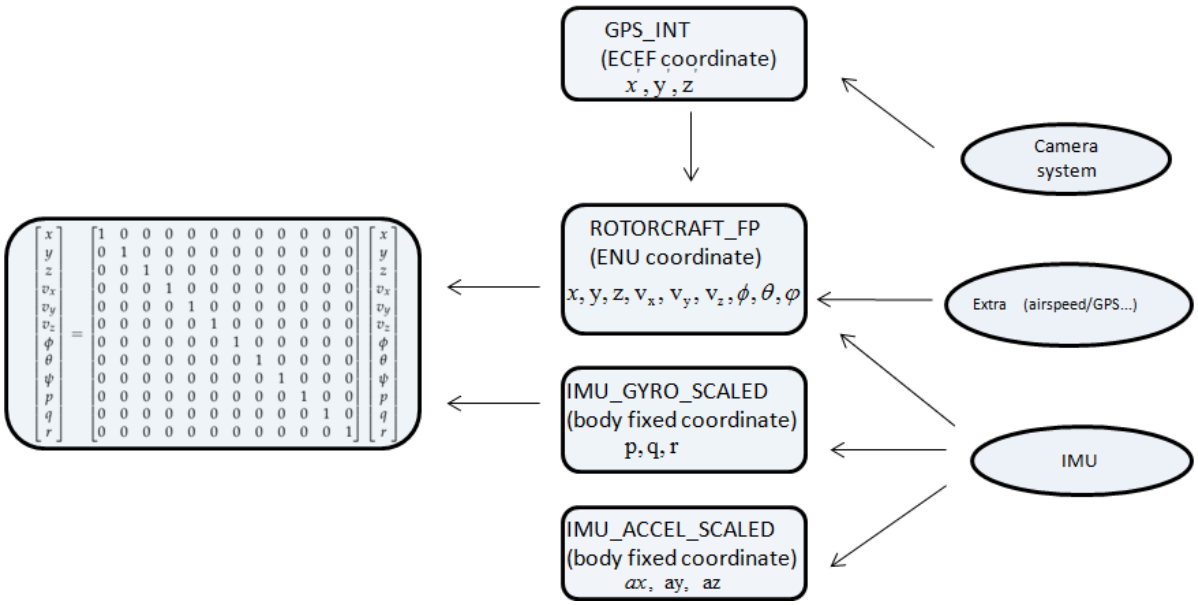


Figure 3-6: Connection between sensors and measurement matrix

In chapter 2, it is noted that the measurement matrix is a twelve by twelve diagonal identity matrix. In Fig. 3-6, ROTORCRAFT_FP, IMU_GYRO_SCALED, IMU_ACCEL_SCALED and GPS_INT are the message names in the message bar in Paparazzi, which are the measurements of the true position, attitude, translational speed and rotational speed of the AR Drone 2 with measurement noise and bias at each time step . Moreover, the errors caused due to the integration of the bias and noise in the IMU can be significantly reduced by processing observations to markers detected in camera system. In fact, the measurement noise in GPS_INT is much smaller than that in ROTORCRAFT_FP. By using the transformation matrix, it is easy to transform from Earth Center Earth Fixed (ECEF) coordinates to East-North-Up (ENU) coordinates. In appendix B, one can find more information with regard to the transformation matrix and ECEF coordinates. Summarily, the measurement needed for EKF and UKF by the sensors can be described as:

- GPS: x', y', z' (measured in ECEF coordinates)
- Gyro: p, q, r (measured in body-fixed coordinates)
- Sensor fusion information of IMU and extra: v_x, v_y, v_z (measured in ENU coordinates), ϕ, θ, ψ

The measurement listed above are used for the nonlinear model Eq. 2-4.

Chapter 4

Model Parameter Identification

The implementation of EKF and UKF requires a good initial guess of moment of inertia and accurate inputs. The variables of the model are estimated using a ruler and weighting scale. Even with all these variables, the inputs (thrust and torque) are unknown. Obtaining these inputs requires identifying the motor-rotor system in advance, which leads to the preliminary identification of engine dynamics and propeller aerodynamics. Some rudimentary measurements and tests should be carried out before the real flight test. Fig. 4-1 accurately describes the relations between the thrust command and thrust, thrust command and torque of each propeller.

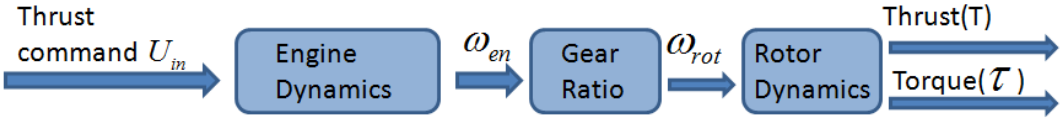


Figure 4-1: Block diagram of the thrust and torque generation process

4-1 Mass and Moment of Inertia

4-1-1 Moment of inertia

For accuracy, AR Drone 2 is disassembled into small parts. Assuming each part as cylindrical or bar shapes permits computing the local moments of inertia by weighing and dimensioning each part, the equations below are used to calculate the disassembled parts. One can refer to [14] for more specific information.

$$I_{cyl} = \frac{1}{2} m_i r_i^2 \tag{4-1}$$

$$I_{bar} = \frac{1}{12} m_i (l_i^2 + w_i^2) \tag{4-2}$$

with r_i the radius of cylinder, l_i and w_i the part length and width of the bar. By knowing the inertia of each part as well as the distance to the central axis $r_{i,c}$, the moment of inertia along the x, y and z-axis can be easily calculated.

$$I_x = \sum I_{x,i} + m_i r_{i,c}^2 \tag{4-3}$$

$$I_y = \sum I_{y,i} + m_i r_{i,c}^2 \tag{4-4}$$

$$I_z = \sum I_{z,i} + m_i r_{i,c}^2 \tag{4-5}$$

4-1-2 Rotor inertia

The nonlinear model also encompasses rotor gyroscopic effect when the residual rotor RPM is not perfectly equal to zero, it is necessary to calculate the rotor inertia. Assuming the rotor as point mass, a rod and a disc gives certain bounds to estimate the moment of inertia of the rotor. The following parameters of each rotor have been computed in advance.

$$m = 3.55 \cdot 10^{-3}\, kg, \quad D = 198 \cdot 10^{-3}\, m, \quad R = 99 \cdot 10^{-3}\, m$$

Modelling by point masses

The simplest model assumes the mass point of each blade is located at the center of gravity of each blade. Cut a rotor into two pieces, the center of gravity of each piece is roughly at 40% of the radius. Then the moment of inertia of each rotor can be determined as follows:

$$I = 2 \cdot \frac{m}{2} \cdot (0.4R)^2 = 3.55 \cdot 10^{-3} \cdot (0.4 \cdot 99 \cdot 10^{-3})^2 = 5.566968 \cdot 10^{-6}\, kg \cdot m^2 \tag{4-6}$$

Since the outer regions of the rotors are not modelled, the result obtained by this method gives a lower bound of the true value [6].

Modelling by a rod

The rotor can also be replaced by a solid rod of equal mass. The result below gives the upper bound on the grounds that the distribution of mass is not even along the radius.

$$I = \frac{2}{3} \cdot m \cdot R^2 = \frac{2}{3} \cdot 3.55 \cdot 10^{-3} \cdot (99 \cdot 10^{-3})^2 = 2.31957 \cdot 10^{-5}\, kg \cdot m^2 \tag{4-7}$$

Modelling by a disc

Assuming the rotor mass is evenly distributed over the disc area, a solid thin disk of equal mass can be used to replace the rotor.

$$I = \frac{1}{2} \cdot m \cdot R^2 = \frac{1}{2} \cdot 3.55 \cdot 10^{-3} \cdot (99 \cdot 10^{-3})^2 = 1.7396 \cdot 10^{-5}\, kg \cdot m^2 \tag{4-8}$$

According to the accurate pendulum experiment [6], it is sufficient enough to take the mean value of the rod and the disc model. The final estimated moment of inertia is:

$$I_r = 2.02 \cdot 10^{-5}\, kg \cdot m^2 \tag{4-9}$$

Table 4-1: The table illustrates the calculated and measured parameters of AR Drone 2.

Parameters	Value	Unit	Remark
g	9.80665	m/s^2	Gravity acceleration
m	0.429	kg	Mass of AR Drone 2
L	0.1785	m	Distance between the center of the mass and the center of the rotor
I_x	0.002237568	$kg \cdot m^2$	Moment of inertia for x-axes
I_y	0.002985236	$kg \cdot m^2$	Moment of inertia for y-axes
I_z	0.00480374	$kg \cdot m^2$	Moment of inertia for z-axes
I_r	$2.029585 \cdot 10^{-5}$	$kg \cdot m^2$	Moment of inertia of each rotor

In the appendix A, Table A-1, A-2 and A-3 illustrate the masses and dimensions of all the disassembled parts, one can refer to this table for the further reading.

4-2 Engine dynamics

For simplicity, the engine model can be modelled in a reduced form, without losing its main features. The back emf effect voltage and the output torque of engine can be described as:

$$\begin{aligned} v_b &= k_v \omega \\ T_e &= ki \end{aligned}$$

with k_v the back emf constant, ω the angular velocity of engine and k the torque constant. Leaving out the electromagnetic losses (magnetic field is constant) gives rise to the equivalence between the mechanical power and the electrical power dissipated by back emf effect. In SI units, the motor torque and back emf constants are same; therefore, k is used to represent both the motor torque constant and the back emf constant. Neglecting the engine inductance l, the dynamic equation of the engine is given by:

$$i = \frac{V - k\omega}{R} \tag{4-10}$$

$$T_e = J\frac{\mathrm{d}\omega}{\mathrm{d}t} + b\omega + T \tag{4-11}$$

where V is applied voltage, T is the torque, R and b are the armature resistance and damping coefficient respectively, and J is the engine inertia. With further substitution, T is given by:

$$T = \frac{k}{R}(V - k\omega) - J\dot{\omega} - b\omega \tag{4-12}$$

With Laplace transform, the equation is described in the form of transfer function.

$$\frac{\omega}{V} = \frac{k}{k^2 + R(Js + b)} \tag{4-13}$$

The inputs are the roll, pitch, yaw and thrust coefficients [3], Paparazzi has the set-point generator which computes and combines these inputs to the motor mixing commands. In order to avoid using the input signal which is included in certain calculations by Paparazzi, the motor mixing command is extracted from script, and is the input directly to each engine. The parameters of the engine dynamics model are identified by giving different step thrust input commands to the motor controller and measuring the value of the rotation speed by using RPM sensor. With the usage of Matlab command `lsqnonlin` based on the nonlinear least squares method, the optimal value of the unknown parameters of the engine model results in the following parameters:

$$k = 1.3014, \quad b = 3.7400, \quad R = 0.6029[\Omega] \quad J = 0.1215[kg \cdot m^2], \quad \omega_{en} = \omega + 1141.7[rad/s]$$

with ω_{en} the engine speed. 1141.7 $[rad/s]$ is engine minimum rotational speed when the engine is "on". The measured and estimated responses can be compared in Fig. 4-2.

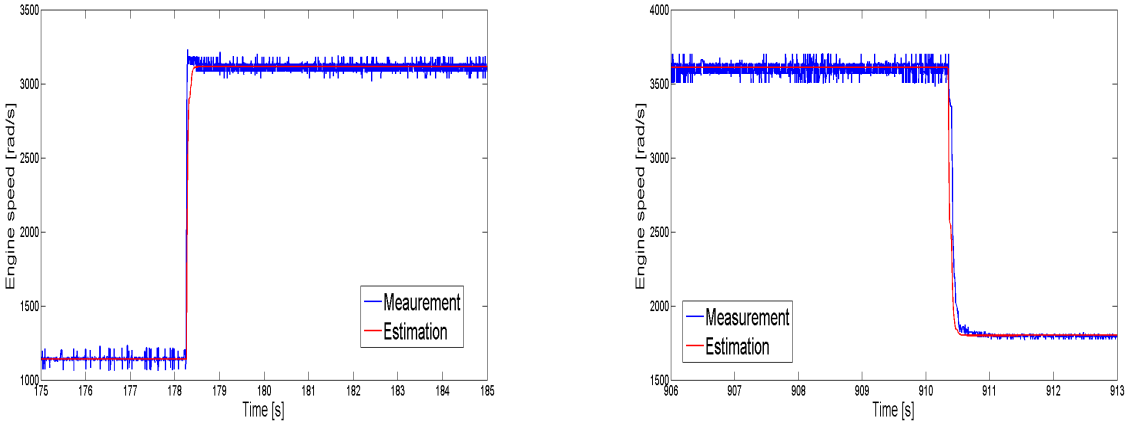


Figure 4-2: Comparison of the real spin-up and spin-down and the corresponding modelled engine dynamics

4-3 Aerodynamic coefficients identification

The thrust factor b and drag factor d of rotors are two important aerodynamic coefficients. From Blade Element Theory, both the thrust and the torque created by a rotor with a linearly twisted blade are proportional to rotor rotation speed squared [21].

$$T_i = b\omega_i^2 \qquad \tau_i = d\omega_i^2$$

The static thrust and torque can be measured by different methods, in which the wind tunnel test is the most precise one. Due to the limitation of the time and experimental setup, two simple experiment setups are created to compute the coefficients.

4-3-1 Thrust and torque measurement

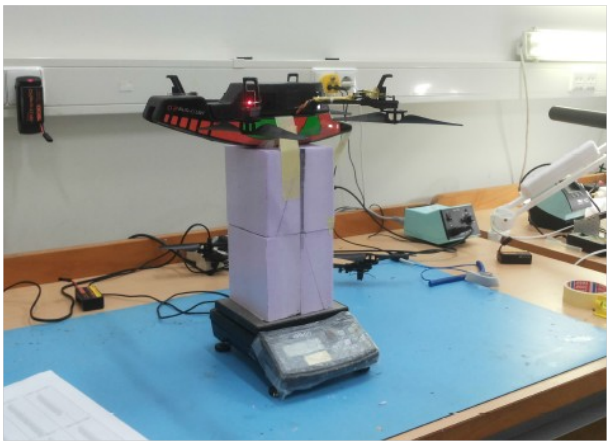


Figure 4-3: The thrust measurement setup

AR Drone 2 is placed upside down on a scale while using a foam pillar to get rid of the ground effect. To avoid the disturbance and noise as well as the inaccuracy of setup, five to ten sets of data are measured for each step thrust command. The derived data set shows a high consistency for each step thrust command. Same step thrust command is given to each engine, thus the total thrust measured by weighing scale is divided by four, by means of which the thrust for each rotor can be calculated. The data relative to the thrust measurement can be found in appendix A, Table A-4.

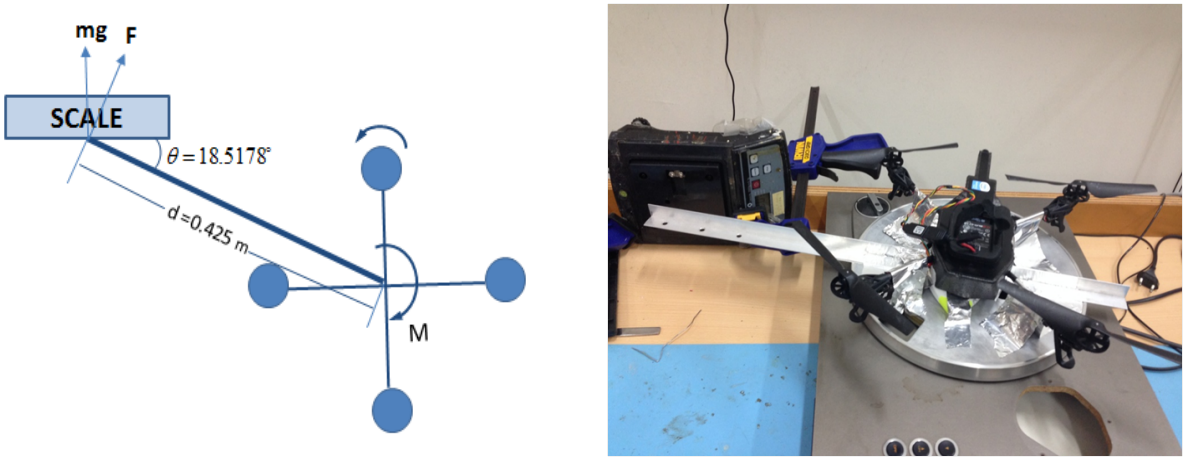


Figure 4-4: Setup for rotor torque identification experiment

In Fig. 4-4, the AR Drone 2 is placed on a rotating disk which is assumed without friction by consequence guarantees the right measurement of the rotor torque. A rigid bar with length d is attached to the AR Drone's center of mass and the other end of the bar is applied to the weighing scale, so that the torque M caused on the AR Drone's airframe due to the rotation of one rotor can be estimated as:

$$M = \frac{mg}{1000}\sec(\theta)d \quad [N\ m] \tag{4-14}$$

Compared with the rotor thrust measurement, a lower consistency of the different data sets for each thrust command is observed. In order to make the torque measurement more reliable, twenty-five sets of data for each thrust command are tested. The mean of 24 datasets for each step command is used to fit the model, and the last set is used to validate the model. The huge sets of measurement data can be found in appendix A, Table A-5.

4-3-2 Thrust and drag coefficients

For control purpose, the quadratic term should be sufficient to describe the relation between the angular velocity of each rotor and the corresponding torque and thrust. However, a better fit appears by adding a linear and constant term. Considering that the basis of the accurate identified model highly depends on the precision of the input and output data, both the quadratic relation and the quadratic term with added linear and constant term relation are considered and compared in details.

Thrust coefficient

By using the least square curve fitting method, the relation becomes:

$$T_{1,i} = 8.048 \times 10^{-6}\omega_{rot,i}^2 \tag{4-15}$$

$$T_{2,i} = 8.386 \times 10^{-6}\omega_{rot,i}^2 - 3.723 \times 10^{-5}\omega_{rot,i} - 0.03818 \tag{4-16}$$

Fig. 4-5 gives the mean of measured thrust values of several data sets and the estimated thrust values obtained by Eq. 4-15 and Eq. 4-16. It can be seen that adding a linear and constant term does improve the fit in the range from minimum to maximum rotational speed. Thus for identification purpose, this model will be utilized for obtaining high identification performance.

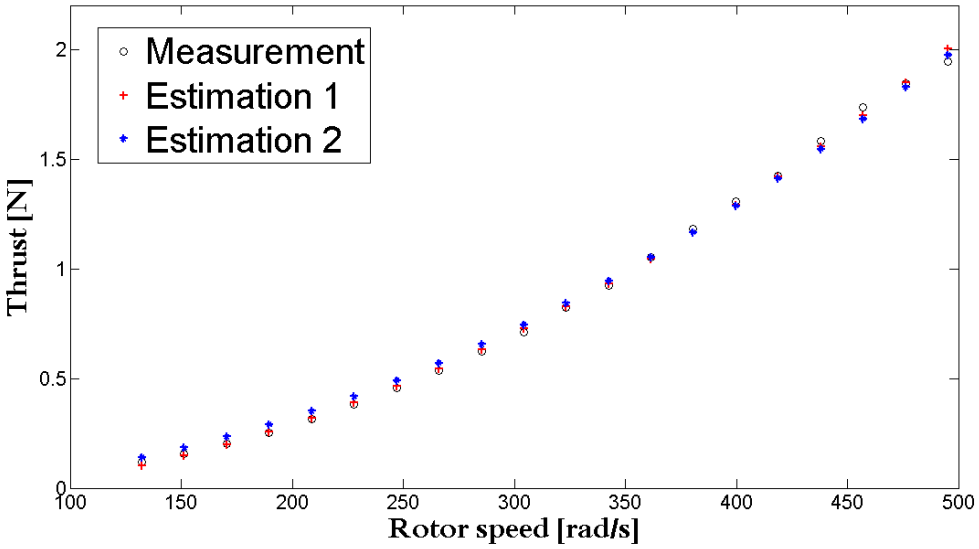


Figure 4-5: A series of thrust measurements compared to the estimated values based on the rotor speeds, estimation 2 is the model with only quadratic term while estimation 1 is the one including linear and constant term.

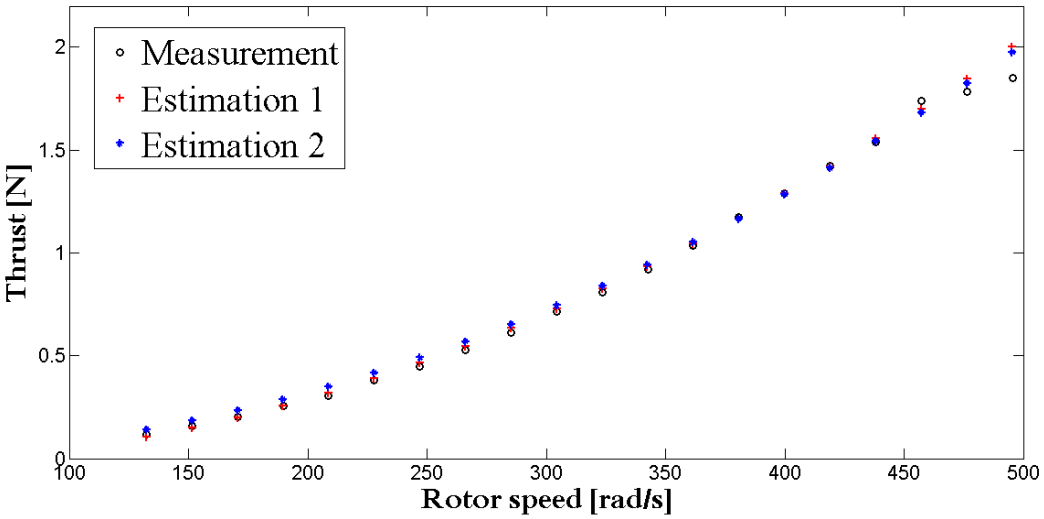


Figure 4-6: Another set of thrust measurements as validation data compared to the estimated values based on the rotor speeds, estimation 2 is the model with only quadratic term while estimation 1 is the one including linear and constant term.

Drag coefficient

The same method is implemented to derive the drag coefficient, the relation is given by:

$$\tau_{1,i} = 2.423 \times 10^{-7} \omega_{rot,i}^2 \tag{4-17}$$

$$\tau_{2,i} = 2.903 \times 10^{-7} \omega_{rot,i}^2 - 3.035 \times 10^{-5} \omega_{rot,i} + 0.04466 \tag{4-18}$$

Figure 4-7 gives the mean of the measured torque values of twenty-four data sets and the estimated torque values obtained by the above equations. In stark contrast to the thrust estimation it can be found that the torque estimation is relatively imprecise due to the inaccurate and rough measurement instrument, though the error has already been decreased significantly by taking the mean of numerous sets of data. Similarly, it can be also seen that adding a linear and

constant term does improve the fit in the range from minimum to maximum rotational speed, thus for identification purpose, this model will be utilized for high identification performance.

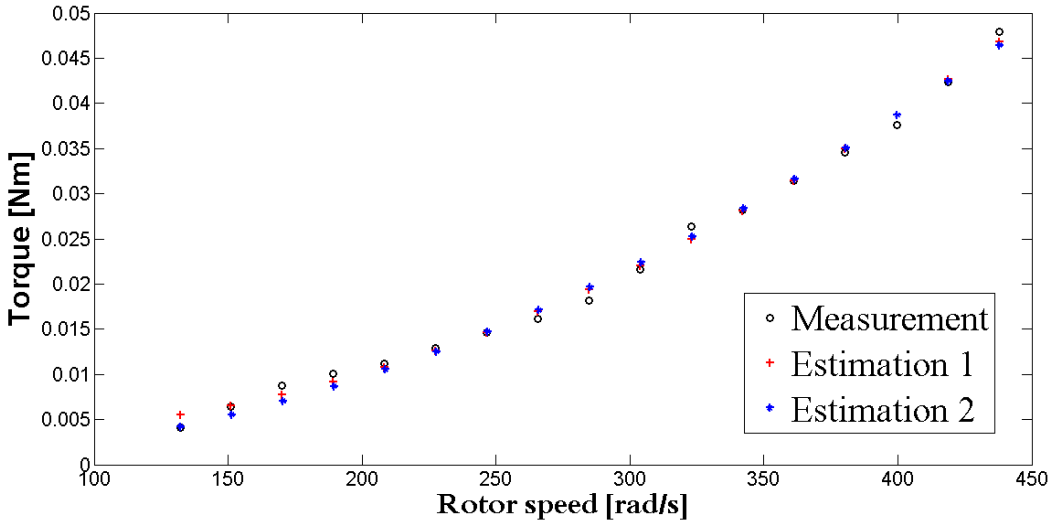


Figure 4-7: A series of torque measurements compared to the estimated values based on the rotor speeds, estimation 2 is the model with only quadratic term while estimation 1 is the one including linear and constant term.

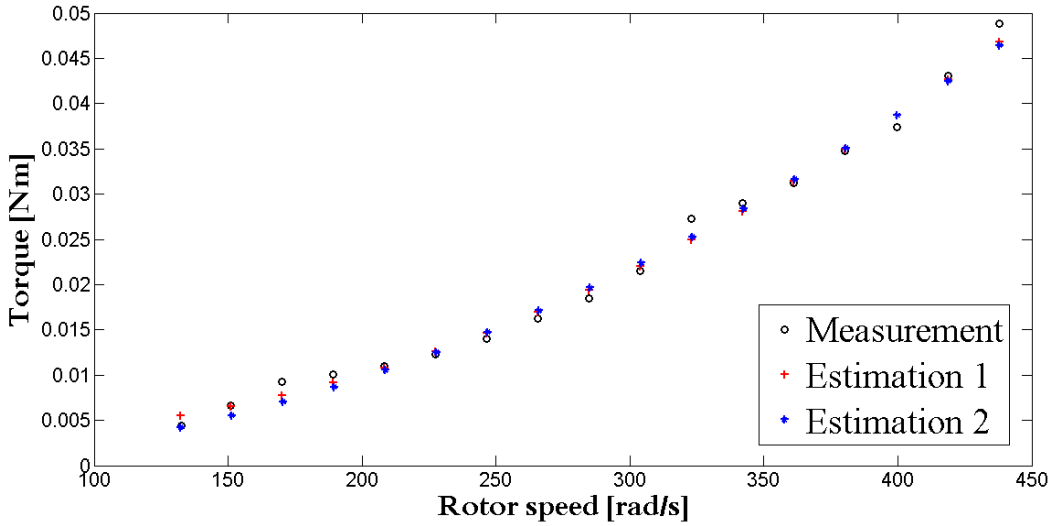


Figure 4-8: Another set of torque measurements as validation data compared to the estimated values based on the rotor speeds, estimation 2 is the model with only quadratic term while estimation 1 is the one including linear and constant term.

Observe that the maximum rotor speed is highly susceptible to the battery voltage under load, apparently in Fig. 4-6 and 4-8. The voltage drops while flying; therefore, the estimation of rotor aerodynamics for high rotor speed is invariably less accurate. However, this issue poses no effect on identification since the flight test of AR Drone is not aggressive.

4-4 Battery consumption

The full battery voltage in rest is 12.65 Volts, the minimum voltage under load to maintain the normal flight is 10 Volts theoretically; nevertheless, the AR Drone shuts down the engine

somewhere between 10 to 10.5 Volts. For the sake of safety, low battery pack is not suitable to do the flight test. Meanwhile, considering the influence of the maximum rotor speed, it is also important to identify the relation between the voltage under load and voltage in rest. The relation is assumed to be quadratic and static and is identified by logging the measured battery voltage after it has settled when the series of step inputs are applied to AR Drone. Four different relations are calculated as,

$$V_{load} = V_{rest} - 8.089 \times 10^{-9} U^2 - 1.649 \times 10^{-5} U \quad (4\text{-}19)$$

$$V_{load} = V_{rest} - 8.104 \times 10^{-9} U^2 - 1.416 \times 10^{-6} U V_{rest} \quad (4\text{-}20)$$

$$V_{load} = V_{rest} - 1.022 \times 10^{-8} U^2 - 0.001817 V_{rest} \quad (4\text{-}21)$$

$$V_{load} = 0.9996 V_{rest} - 8.342 \times 10^{-9} U^2 - 1.231 \times 10^{-6} U V_{rest} \quad (4\text{-}22)$$

where U is the average thrust command to four engine, V_{load} and V_{rest} are the voltage under load and in rest respectively. The table below shows the results by applying Eq. 4-19, 4-20, 4-21 and 4-22, where the data set is randomly selected among several battery and input levels.

Table 4-2: Compare the measured load and rest voltages with the calculated voltage under load of AR Drone 2, the estimated error 1, 2, 3 and 4 are relevant to Eq. 4-19, 4-20, 4-21 and 4-22, respectively.

Rest voltage [V]	Load Voltage [V]	Estimated Error 1 [V]	Estimated error 2 [V]	Estimated error 3 [V]	Estimated error 4 [V]
11.15	11.15	0	0	0.0203	0.0039
11.15	11.10	-0.0254	-0.0261	-0.0195	-0.0240
11.15	11.10	0.0153	0.0140	0.0112	0.0147
11.15	11.00	-0.0277	-0.0297	-0.0377	-0.0298
11.15	10.95	-0.0046	-0.0072	-0.0162	-0.0077
11.10	10.85	0.0347	0.0312	0.0257	0.0308
11.10	10.70	-0.0100	-0.0319	-0.0118	-0.0138
11.00	10.50	0.0118	0.0061	0.0209	0.0074

Eq. 4-19 gives the smallest sum of the absolute value of estimated error, which signals that this model is the most suitable model to describe the relation.

Chapter 5

Identification Algorithm

The identification algorithms tested in this project are presented in this chapter. The unscented Kalman filter (UKF) was first published in 1995 and proposed by Julier and Ulhman [16][23][17]. The early attempt to adapt this Kalman filter to nonlinear models was accomplished using the Extended Kalman Filter (EKF) [18]. In terms of the state estimation, both of them are able to track the state successfully for aircraft [9], [8] and [25]. However, it is also interesting to explore their application for the parameter estimation, since the calculated moments of inertia in chapter 4 are not accurate. In contrast with state estimation, less literature can be found relative to the parameter estimation for the quadrotor [8]. In addition to EKF and UKF, multi-rate Kalman filter is another interesting algorithm to explore, especially in the case of the multi-sensor problem with different sampling rates. This chapter is organized as follows. Section 5.1 gives a general description of EKF and UKF, respectively, section 5.2 discusses the multi-rate Kalman filter in depth, including the nonlinear multi-rate model description described by input-hold mechanism, Multi-Rate Extended Kalman Filter (MREKF) and multi-rate unscented Kalman filter concisely. Next,the augmented state description for parameter estimation is discussed.

5-1 Extended and Unscented Kalman Filter

Both EKF and UKF have already been described at large in the literature report [19] and thus will be introduced concisely in this section.

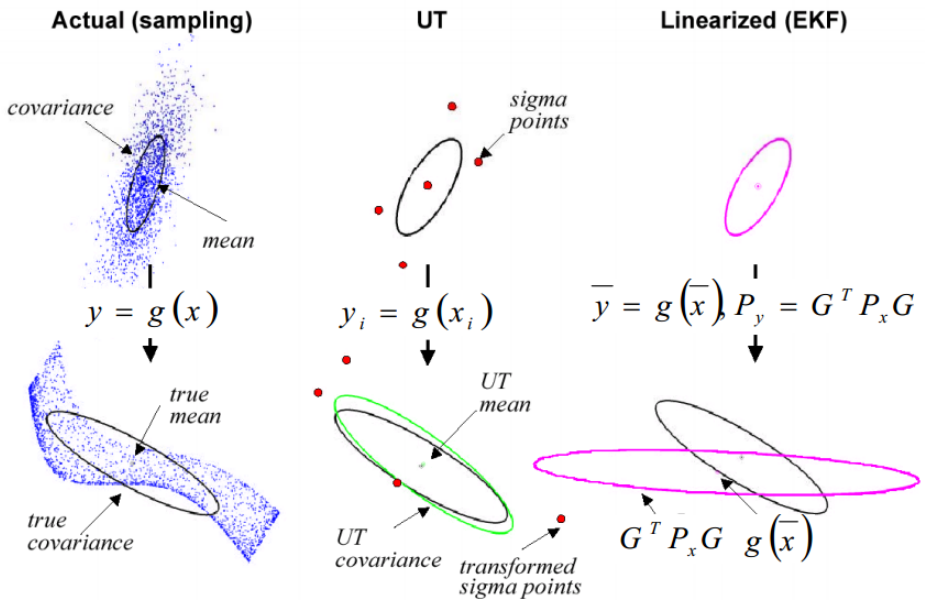


Figure 5-1: Propagation of distributions for EKF and UKF [25]

The basic idea for UKF is based on the principle that it is easier to approximate through a probability distribution than through a nonlinear transformation. Table 5-1 lists the comparison of the working procedure between EKF and UKF.

Table 5-1: Propagation procedure of UKF and EKF

UKF	EKF
1. UT transform and create sigma points	1. Linearize the model by computing the Jacobin matrix
2. Propagate the sigma points through process model	2. Compute the predicted mean and covariance of the linearized model
3. Calculate the transformed mean and covariance	3. Compute the Kalman gain
4. Compute the Kalman gain	4. Update the state estimate with measurements
5. Update the state estimate with measurements	

Unlike its linear counterpart, EKF is more sensitive to the initial estimate of the state and covariance matrix, and is found not to be an optimal estimator. If one of the initial estimates is incorrect, the filter is not able to converge and track the state. On the contrary, as an improvement over EKF, UKF tends to have the better performance in terms of accuracy and robustness. For a deep understanding of the working principle, the algorithms of EKF and UKF are described in the following subsections.

5-1-1 Extented Kalman filter

The system model and measurement model of continuous-discrete filtering problem are expressed as [7]:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) + w(t), \quad w(t) \sim N(0, Q(t)) \\ z_k &= h(x_k) + v_k, \qquad\qquad v_k \sim N(0, R_k) \end{aligned}$$

Where $x_k = x(t_k)$, $w(t)$ and v_k are the process and measurement zero mean Gaussian noises with covariance $Q(t)$ and R_k respectively.

Initialize

$$\begin{aligned} \hat{x}_{0|0} &= E[x(t_0)] \\ P_{0|0} &= Var[x(t_0)] \end{aligned}$$

Predict

Predicted state estimate: $\begin{cases} \dot{\hat{x}}(t) &= f(\hat{x}(t), u(t)) \\ \hat{x}(t_{k-1}) &= \hat{x}_{k-1|k-1} \end{cases}$, thus $\hat{x}_{k|k-1} = \hat{x}(t_k)$
Predicted covariance estimate: $\begin{cases} \dot{P}(t) &= F(t)P(t)+P(t)F(t)^T+Q(t) \\ P(t_{k-1}) &= P_{k-1|k-1} \end{cases}$, thus $P_{k|k-1} = P(t_k)$
where $F(t) = \frac{\partial f}{\partial x}|_{\hat{x}(t),u(t)}$ is the Jacobian matrix of f.

Update

Kalman gain: $K_k = P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1}$
Updated state estimate: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - h(\hat{x}_{k|k-1}))$
Updated covariance estimate: $P_{k|k} = (I - K_kH_k)P_{k|k-1}$
where $H_k = \frac{\partial h}{\partial x}|_{\hat{x}_{k|k-1}}$.

5-1-2 Unscented Kalman filter

The standard UKF algorithm can be summarized in the following steps:

1. Initialization with:

$$\begin{aligned} \hat{x}_0 &= E[x_0] \\ P_0 &= E[(x_0 - \hat{x_0})(x_0 - \hat{x}_0)^T] \\ \hat{x}_0^a &= E[x^a] = \begin{bmatrix} \hat{x}_0^T & 0 & 0 \end{bmatrix}^T \\ P_0^a &= E[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix} \end{aligned}$$

2. Update
Calculate the $(2L+1)$ sigma points, where L is the dimension of state.

$$\mathcal{X}_{k-1}^a = \begin{bmatrix} \hat{x}_{k-1}^a & \hat{x}_{k-1}^a \pm \sqrt{(L+\lambda)P_{k-1}^a} \end{bmatrix}$$

Time update:

$$
\begin{aligned}
\mathcal{X}^x_{k|k-1} &= f(\mathcal{X}^x_{k-1}, \mathcal{X}^v_{k-1}) \\
\hat{x}_{k|k-1} &= \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}^x_{i,t|t-1} \\
P_{k|k-1} &= \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}^x_{i,k|k-1} - \hat{x}_{k|k-1}][\mathcal{X}^x_{i,k|k-1} - \hat{x}_{k|k-1}]^T \\
\mathcal{Y}_{k|k-1} &= h(\mathcal{X}^x_{k|k-1}, \mathcal{X}^n_{k-1}) \\
\hat{y}_{k|k-1} &= \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1}
\end{aligned}
$$

Measurement update:

$$
\begin{aligned}
P_{\tilde{y}_k \tilde{y}_k} &= \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1}][\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1}]^T \\
P_{x_k y_k} &= \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{x}_{k|k-1}][\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1}]^T \\
K_k &= P_{x_k y_k} P^{-1}_{\tilde{y}_k \tilde{y}_k} \\
\hat{x}_k &= \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \\
P_k &= P_{k|k-1} - K_k P_{\tilde{y}_k \tilde{y}_k} K_k^T
\end{aligned}
$$

where λ is scaling parameter, Q and R are the process covariance and measurement covariance matrices respectively, K is the Kalman gain and W are the weights. The augmented state is expressed as: $x^a = \begin{bmatrix} x^T & v^T & n^T \end{bmatrix}^T$, v^T and n^T are the mean of the process and measurement noise. And more remarkably, these two algorithms deal with the problem when all the data have the same sampling rate. If the condition is getting complex and more than two sampling rates appear, the multi-rate Kalman filter is then taken into account.

5-2 Nonlinear Multi-rate Kalman Filter

Tracking is utilized in many important applications especially in "real-time" systems. For the accurate tracking purpose, numerous sensors have been used such as GPS, vision system sensors, IMU, etc.

In this project, in order to obtain more accurate data, the multi-rate Kalman filter (MRKF) is used to fuse the complementary characteristics of visual and inertial sensors in simulation, as it is suitable to integrate the high number of sensors. On the one hand, at fast motions, inertial sensing is able to track precisely and at the same time drift to some extent at slow motions. On the other hand, visual sensing tracks accurately at low velocities, which makes them the perfect combination for accurate data. Although the sampling period for each sensor can be random and asynchronous, for a discrete system a periodicity can be extracted between input and output sampling. From a sequence of inputs sampled at a slow sampling rate $\hat{T}$ (known as frame-period), a continuous signal is discretized at a high sampling rate (commonly known as base period) T. The base-period is N times faster than the frame-period, which is:

$$T = N\hat{T}$$

The multi-rate system can be described using an output-hold mechanism or input-hold mechanism. The output-hold mechanism is defined when an output is not measured and the auxiliary state vector is used to hold the value of the last measured output while the input-hold mechanism is corresponding to the case when an input is not measured [12].

5-2-1 Multi-rate system description

The multi-rate system description based on a size-varying output equation and input-hold mechanism is proposed. The non-linear multi-rate inputs are described as [12],

$$\begin{aligned} v_u(k) &= [1-\Delta_u(k-1)]v_u(k-1)+\Delta_u(k-1)u(k-1) \\ \hat{u}(k-1) &= [1-\Delta_u(k-1)]v_u(k-1)+\Delta_u(k-1)u(k-1) \end{aligned}$$

where $v_u(k)$ is the slack variable. The estimated inputs $\hat{u}(k-1)$ take the value of the slack variables or the updated input $u(k-1)$ ($\Delta_u(k-1)=0$ if not updated). Accordingly, a non-linear multi-rate model is described as,

$$\begin{aligned} x(k) &= f[x(k-1)]+g[x(k-1)]\hat{u}(k-1)+\gamma[x(k-1)]w(k-1) \\ y\Delta_y(k) &= h_{\Delta_y}[x(k)]+v_{\Delta_y}(k) \end{aligned}$$

5-2-2 Multi-rate Extended Kalman Filter

Based on the non-linear multi-rate model, the multi-rate Extended Kalman filter is obtained [12],

$$\begin{aligned} \hat{x}(k|k-1) &= f[\hat{x}(k-1|k-1)]+g[x(k-1)]\hat{u}(k-1) \\ P(k|k-1) &= F(k-1)P(k-1|k-1)F^T(k-1)+\Gamma(k-1)Q(k)\Gamma^T(k-1) \\ S_{\Delta_y}(k|k-1) &= H_{\Delta_y}(k)P(k|k-1)H^T_{\Delta_y}(k)+R_{\Delta_y}(k) \\ K_{\Delta_y}(k) &= P(k|k-1)H^T_{\Delta_y}(k)S^{-1}_{\Delta_y}(k|k-1) \\ z_{\Delta_y}(k) &= y_{\Delta_y}(k)-h_{\Delta_y}[\hat{x}(k|k-1)] \\ \hat{x}(k|k) &= \hat{x}(k|k-1)+K_{\Delta_y}(k)z_{\Delta_y}(k) \\ P(k|k) &= P(k|k-1)-K_{\Delta_y}(k)H_{\Delta_y}(k)P(k|k-1) \end{aligned}$$

The working principle is the same as for the normal EKF, note that the size of the vectors and matrices with sub-index Δ_y varies depending on the multi-rate sampling of the output vector, which shows the periodic change.

5-2-3 Multi-rate Unscented Kalman Filer

The sample set (sigma points) is chosen to have the same mean and covariance as the distribution of $x(k)$, which is computed with the Unscented Transform. The detailed steps are not shown in this subsection. The only difference between the MRUKF and normal UKF is the

estimated output equations, which depend heavily on the multi-rate sampling [12].

$$\begin{aligned}
\mathscr{X}_i(k|k-1) &= f[\mathscr{X}_i(k-1|k-1)]+g[\mathscr{X}_i(k-1|k-1)]\hat{u}(k-1)+\gamma[\mathscr{X}_i(k-1|k-1)]\mathscr{W}_i(k-1)\\
\hat{x}(k|k-1) &= \sum_{i=0}^{2n} W_i \mathscr{X}_i(k|k-1)\\
P(k|k-1) &= \sum_{i=0}^{2n} W_i \left\{\mathscr{X}_i(k|k-1)-\hat{x}(k|k-1)\right\}\left\{\mathscr{X}_i(k|k-1)-\hat{x}(k|k-1)\right\}^T\\
\mathscr{Y}_{\Delta_y,i}(k|k-1) &= h_{\Delta_y}[\mathscr{X}_i(k|k-1)]+\Psi_{\Delta_y,i}(k)\\
\hat{y}_{\Delta_y}(k|k-1) &= \sum_{i=0}^{2n} W_i \mathscr{Y}_{\Delta_y,i}(k|k-1)\\
P_{\Delta_y,yy}(k|k-1) &= \sum_{i=0}^{2n} W_i \left\{\mathscr{Y}_{\Delta_y,i}(k|k-1)-\hat{y}_{\Delta_y}(k|k-1)\right\}\left\{\mathscr{Y}_{\Delta_y,i}(k|k-1)-\hat{y}_{\Delta_y}(k|k-1)\right\}^T\\
P_{\Delta_y,xy}(k|k-1) &= \sum_{i=0}^{2n} W_i \left\{\mathscr{X}_i(k|k-1)-\hat{x}(k|k-1)\right\}\left\{\mathscr{Y}_{\Delta_y,i}(k|k-1)-\hat{y}_{\Delta_y}(k|k-1)\right\}^T
\end{aligned}$$

where $P_{\Delta_y,yy}(k|k-1)$ denotes the output covariance matrix which is equivalent to $S_{\Delta_y}(k|k-1)$ in Extended Kalman filter. The state estimation and the corresponding covariance are described using the subset of the sampled outputs which are given by:

$$\begin{aligned}
K_{\Delta_y}(k) &= P_{\Delta_y,xy}(k|k-1)\cdot P^{-1}_{\Delta_y,yy}(k|k-1)\\
\hat{x}(k|k) &= \hat{x}(k|k-1)+K_{\Delta_y}(k)\cdot(y_{\Delta_y}(k)-\hat{y}_{\Delta_y}(k|k-1))\\
P(k|k) &= P(k|k-1)-K_{\Delta_y}(k)\cdot P_{\Delta_y,yy}(k|k-1)\cdot K^T_{\Delta_y}(k)
\end{aligned}$$

5-3 The Augmented State Description

As stated before, **the object of this project is the state and parameter estimation of nonlinear models for control purpose**. Thus, it is of great significance to investigate how to incorporate unknown parameters into Kalman filter. The Eq. 5-1 represents the system dynamics of the continuous-discrete filtering problem [16].

$$\begin{aligned}
\dot{x}(t) &= f[x(t),u(t),\beta]+Fw(t), \quad x(t_0)=x_0\\
y(t) &= g[x(t),u(t),\beta]\\
z(k) &= y(k)+Gvk)
\end{aligned}$$

with F the time-invariant additive state matrix and G the time-invariant measurement noise matrix. The unknown parameter vector β is considered to be constant and then the augmented state vector is defined as:

$$x_a=\begin{bmatrix}x\\ \beta\end{bmatrix} \qquad \dot{\beta}=0$$

The extended system dynamics are then given by:

$$\begin{aligned}
\dot{x}_a(t) &= f_a[x_a(t),u(t)]+F_aW_a(t)\\
&= \begin{bmatrix}f[x(t),u(t),\beta]\\ 0\end{bmatrix}+\begin{bmatrix}F & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}w(t)\\ 0\end{bmatrix}\\
y(t) &= g_a[x_a(t),u(t)]\\
z(k) &= y(k)+Gv(k)
\end{aligned}$$

where the subscript "a" denotes the augmented variables. Applying this dynamical model to a Kalman filter, the dimensions of the observation vector y and the measurement vector z are unchanged. The changed parts in the Kalman filter are, obviously, the state vector, covariance matrix P and process noise matrix Q, and the size of these vector and matrices are increased accordingly after adding the unknown parameter vector.

Chapter 6

Simulation

In this chapter, the algorithms listed in chapter 5 will be tested by implementing an AR Drone simulator. The simulator has been modelled with Matlab tool Simulink before the model parameter estimation experiment, so the motor dynamics and control inputs, in Fig. 6-1, used in Simulink are reasonably cited from references, the bare-airframe in simulator is, however, identical to the one in chapter 2. Owing to the constraint of time, the simulation results will not be used as real-time simulator and compared with the real-time flight data afterwards. The purpose in simulation is to verify the applicability of the algorithms, and simultaneously, gives some clues how to tune the EKF and UKF, and to what extent the initial guess and noise matrices could influence the simulation results for each algorithm.

6-1 Blocks implementation and 3D visualization

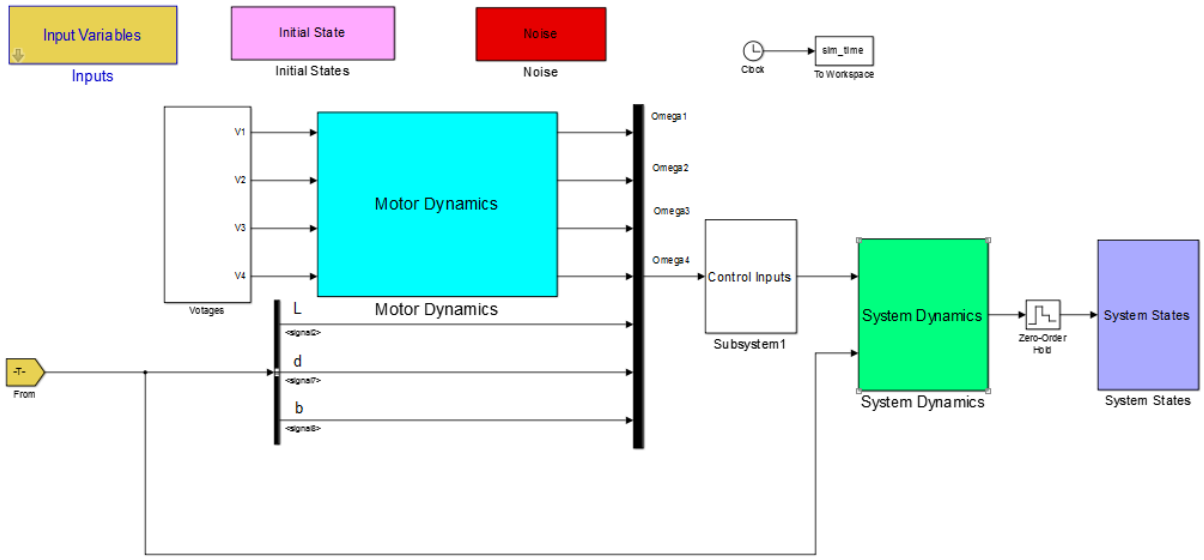


Figure 6-1: The block diagram of system structure

Fig. 6-1 shows the block diagram of the system dynamics, where different input signals are given to sufficiently excite the system. In the block diagram, the input variables are the unknown parameters measured in chapter 4. It should be noted that in the simulation the inputs

are the voltage signals given to the motors, whereas in reality the engines of the AR Drone are controlled by power, not voltage. Nevertheless, this is by no means affecting the validity of the simulation results. Furthermore, the noise block represents the process noise added to each state. In contrast to measurement noise, the most different part of tuning Kalman filter is to tune the process noise matrix and initial covariance matrix. Normally, both of them can be tuned by trial and error, and to the best of the author's knowledge, there is no way to symmetrically or empirically tune them; therefore, more attention is given to process noise so as to gain more insight into the world of Kalman filter. The last one is the initial state; indeed, it suffices to set all states to 0.

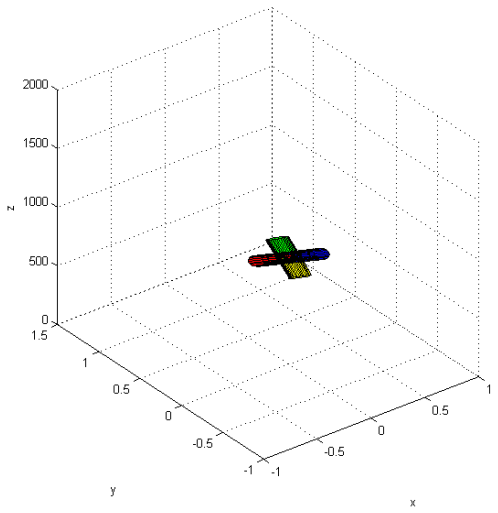


Figure 6-2: 3D animation

Fig. 6-2 demonstrates the 3D animation for the quadrotor and allows to check the trajectory and analyze the position and attitude of the quadrotor. Now the simulator is ready and the data from simulator is available to test the algorithms and compare the performance.

6-2 Simulation results

6-2-1 Simulation Analysis

The UKF, MRUKF, EKF and MREKF are examined and compared for parameter and state estimation. Noted that the multi-rate Kalman filter can be used under the assumption that, for simplicity, the sampling time for MREKF and MRUKF is $T = 2\hat{T}$ without losing generality. The period (base-period) of IMU and the input sampled period are half of GPS period. Of course, it is possible to set the common multiple period for the sensors as any value. Fig. 6-3, 6-4 and 6-5 represent the results of the reconstruction of the moment of inertia by UKF, MRUKF and EKF algorithms along three axes as well as the moment of inertia for the rotor.

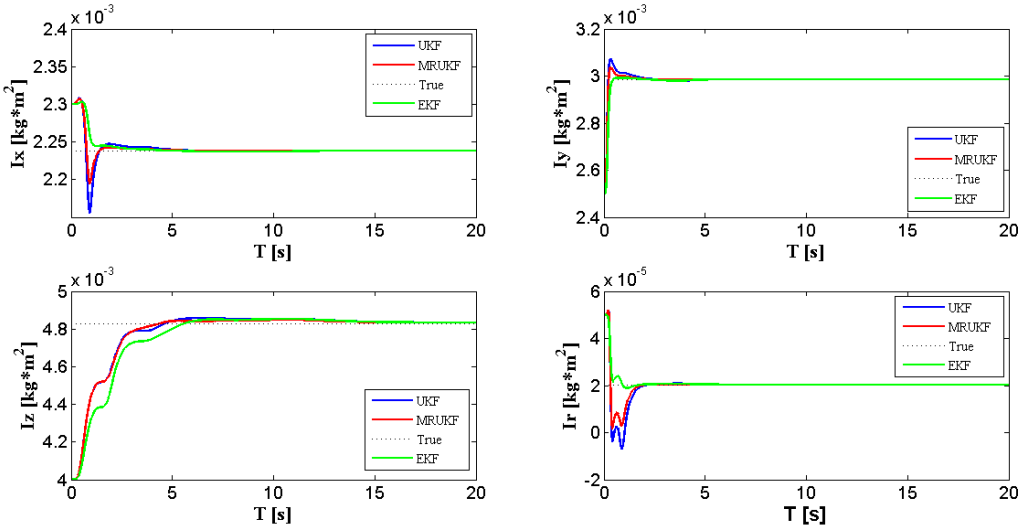


Figure 6-3: Reconstruction of moment of inertia for three axes and rotor without process noise

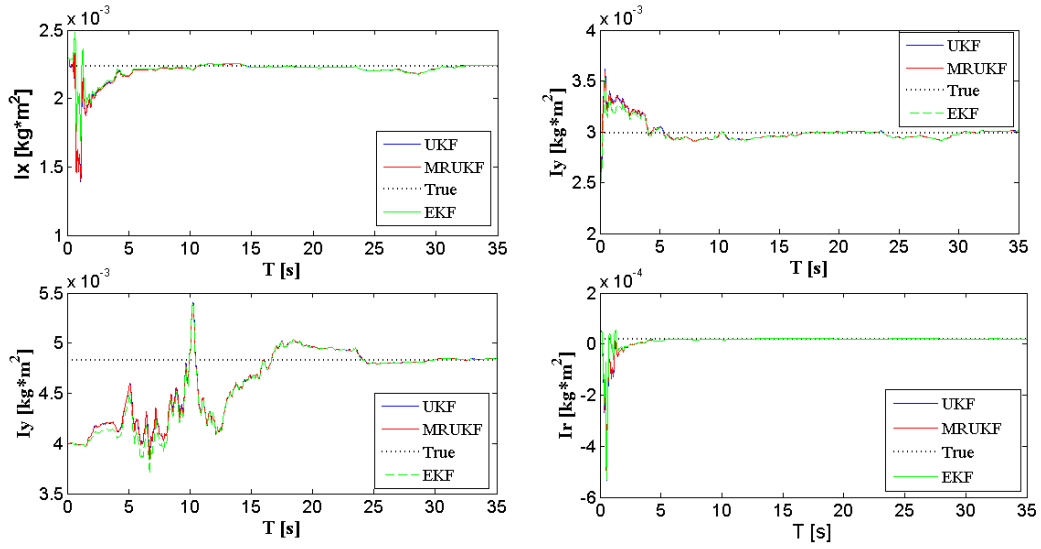


Figure 6-4: Reconstruction of moment of inertia for three axes and rotor with process noise

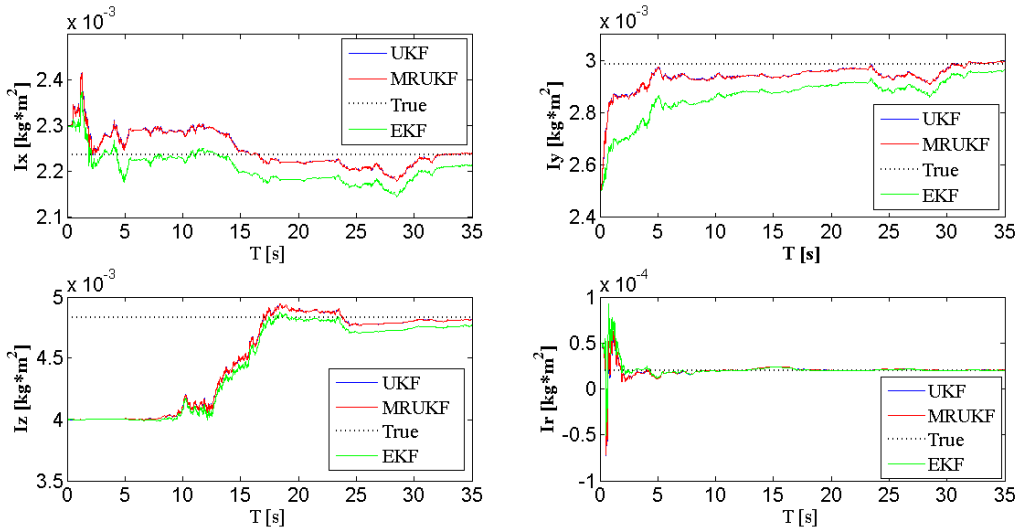


Figure 6-5: Reconstruction of moment of inertia for three axes and rotor with inconsistent estimate of process noise matrix.

It can be observed that UKF and EKF show a virtually identical performance under the condition of the consistency in initial guess and noise matrix between the simulator and Kalman filter. What is more, the convergence speed of the EKF in Fig. 6-3 along the x axis goes so far as to prevail over that of the UKF; however, if the process noise matrix of the Kalman filter is inconsistent with that in simulator, in Fig. 6-5, the performance of UKF is largely superior to EKF. From the analysis above it can be seen that the EKF is more sensitive to the initial estimate of the state or process noise owing to its linearization. The following subsections give the details of the analysis based on the Fig. 6-3.

Precision

MREKF is not shown in the plots since MREKF is not able to converge to the true value. The error caused by the proposed input-hold mechanism propagates through the linearized model and accumulates during the time. The estimation results are shown in Table 6-1.

Table 6-1: Comparison between estimated and true parameters, the error is included as bracketed points

Approach	UKF	MRUKF	EKF
Ix	0.002238347 $(7.8 * 10^{-7})$	0.002238313 $(7.45 * 10^{-7})$	0.002238332 $(7.64 * 10^{-7})$
Iy	0.002985760 $(5.24 * 10^{-7})$	0.002985782 $(5.46 * 10^{-7})$	0.002985745 $(5.09 * 10^{-7})$
Iz	0.004834382 $(5.364 * 10^{-6})$	0.004833439 $(5.421 * 10^{-6})$	0.004834336 $(5.318 * 10^{-6})$
Ir	0.00002031540 $(1.955 * 10^{-8})$	0.00002032108 $(2.523 * 10^{-8})$	0.00002031713 $(2.128 * 10^{-8})$

Convergence issue

Convergence properties of the three methods can be judged according to the number of iterations required to arrive to the same tolerance criteria. $\pm 5\%$ is selected as the tolerance level. Table 6-2 shows the iteration numbers of the three approaches.

Table 6-2: The number of required iterations based on three algorithms

Approach	UKF	MRUKF	EKF
Ix	987	822	784
Iy	442	372	324
Iz	3157	2946	3145
Ir	438	408	355

Computation load

The EKF based method requires less computational power than UKF since UKF needs to propagate $(2n + 1)$ sigma points through nonlinear model. Compared with MRUKF, UKF is also computationally expensive.

Conclusion

From precision and convergence perspective, the three algorithms show more or less same performance, it goes without saying that EKF is more efficient than UKF and MRUKF to some extent especially for computation load. However, EKF is more sensitive to the initial guess, noise and disturbance. For instance, if the estimated process noise in Kalman filer is inconsistent with the one in simulator to a certain extent, EKF is no longer able to estimate parameters and state due to the error accumulation, while UKF is more robust to the wrong initial guess of the state and process noise. Thus UKF is proposed as good backup plans. However, it is unrealistic to have such a high consistency for the model, the estimate of initial guess and noise matrix in practice. For this reason, this analysis has mostly a theoretical value. However, it gives a meaningful hint on how to tune the Kalman filter practically, and paves the way for a real-life implementation, as presented in the next chapter.

On the other hand, MRUKF outperforms UKF in almost every aspect. Obviously, MRUKF needs less computational power and converges fast, though it is ambiguous to compare the precision. On the top of that, MRUKF permits to effectively use different sensors and to implement sensor fusion. Undoubtedly, it clearly benefits from the assumption ($T = 2\hat{T}$) for the multi-rate Kalman filter, and the performance will be definitely changed with the change of this assumption. Assume $T = 4\hat{T}$, the performance of MRUKF is then expressed as:

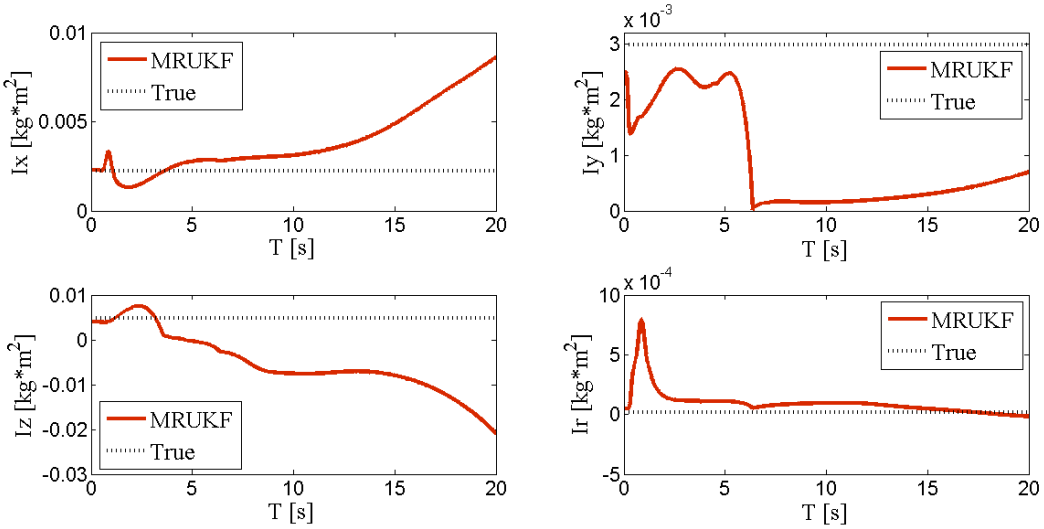


Figure 6-6: Reconstruct of moment of inertia for three axes and rotor when the assumption for MRUKF is $T = 4\hat{T}$

Obviously, under this assumption, the performance of MRUKF is drastically changed and not able to converge. Due to this limitation, MRUKF will not be tested for parameter estimation on the real flight test.

Chapter 7

System Identification

The unknown parameters of the bare-airframe can be obtained using different approaches. In chapter 4, the unknown parameters have been calculated and thereby used as initial guess for Kalman filter. In this chapter, the proposed Kalman filter will be applied to real flight data. The linear input-output transfer function can be considered as a fair approximation as long as under the small angle perturbation assumption, which will be analyzed in section 7-1. Sect. 7-2 introduces the core of how to implement the Kalman filter for the real flight test. For the purpose of understanding the constraints of implementing the Kalman filter, the corresponding flight test will be described, and the analysis based on the identification results will be described in section 7-3 . It is noteworthy that due to the high nonlinearities of the real flight test data, the nonlinear model identification is the main part of this work.

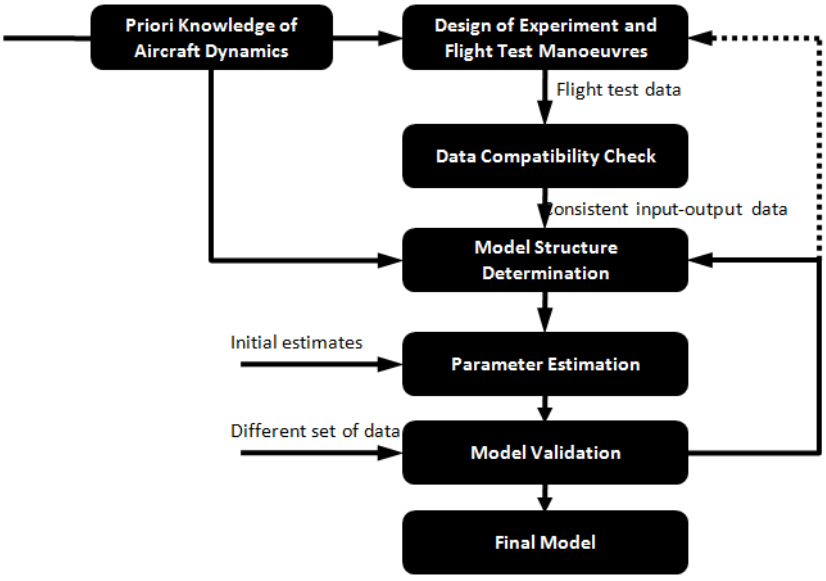


Figure 7-1: System identification procedure [19]

7-1 Linear Model Transfer Function Identification

The proposed approach for linear model identification in literature study in this project is the two-stage method, a combination of the subspace identification method and prediction error

method. The advantages of this approach have been analysized in literature report [19] and the approach has been applied successfully to sample data to verify the validity of the method [24]. However, the unsuccessful implementation for a linear state-space model using the simulated and real flight data proves the impracticable usage of the proposed method; the fit percentage is too low, even negative for the majority of the channels, owing to the coupling effect and nonlinearity of the model in the simulator. In practice, the nonlinear factors far surpass those in simulator, the aero-elastic effects and flexibility of the wings render the nonlinearity more complex. Considering all the factors mentioned above, the proposed method is not an optimal choice for linear state-space model identification. Consequently, the linear input-output transfer function is considered as a backup plan for linear model identification. For each transfer function, the procedure of identification is expressed as:

- The real flight data have been divided into identification and validation data, and a fourth-order low pass Butterworth filter is used to get rid of the noise;
- A Box-Jenkins (BJ) polynomial model based on the prediction error minimization method (PEM) which minimizes the prediction errors by estimate of the parameters of the BJ model. For more information relative to the PEM method and BJ model in details, see [26];
- The obtained transfer function is tested on the different sets of validation data using the system identification toolbox;
- The order of the transfer function is changed if the validation results are not satisfactory until the best numerical fit appears.

The identification for the four transfer functions will be reported and discussed as follows.

7-1-1 Throttle transfer function identification

The BJ polynomial model is given by:

$$y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}e(t) \tag{7-1}$$

where $\begin{bmatrix} n_b & n_c & n_d & n_f & n_k \end{bmatrix}$ defines the order of the polynomials used for estimation. Based on Eq. 2-5, the initial guess of the system is assumed to be a 2^{th} order system. After testing $2^{th}, 3^{th}, 4^{th}$ order BJ models with various samples delay, a second order discrete-time transfer function with one sample delay is obtained in the following equations:

$$\begin{aligned} B(z) &= -6.632e^{-5}z^{-1} - 2.395e^{-5}z^{-2} \\ C(z) &= 1 + 1.988z^{-1} + z^{-2} \\ D(z) &= 1 - 1.989z^{-1} + 0.989z^{-2} \\ F(z) &= 1 - 1.979z^{-1} + 0.9788z^{-2} \end{aligned}$$

The table lists the fits between the identified transfer function output and the identification data, and identified transfer function output and the validation data.

Table 7-1: Fits for the identification and validation data

BJ22221	Identification data	Validation data
Fits	67.6%	50% $\sim$ 65%

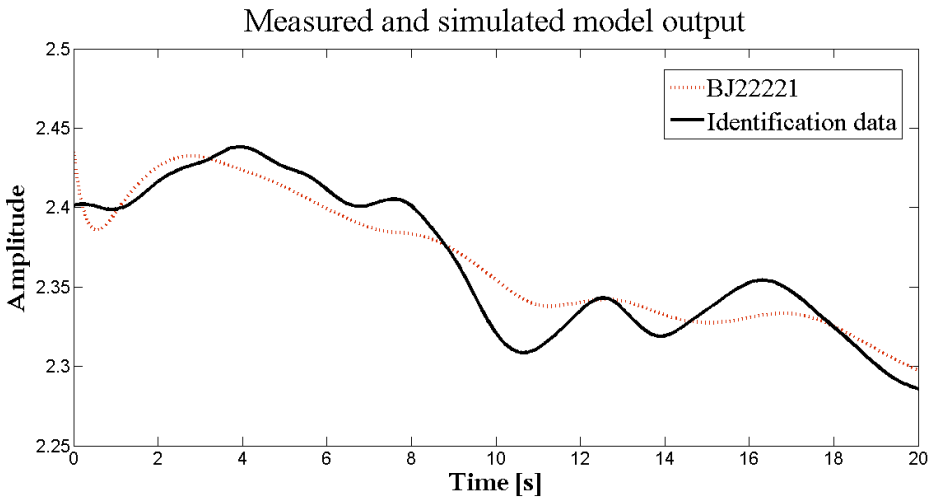


Figure 7-2: Fit of 67.6% between BJ model output (dashed) and identification data (solid)

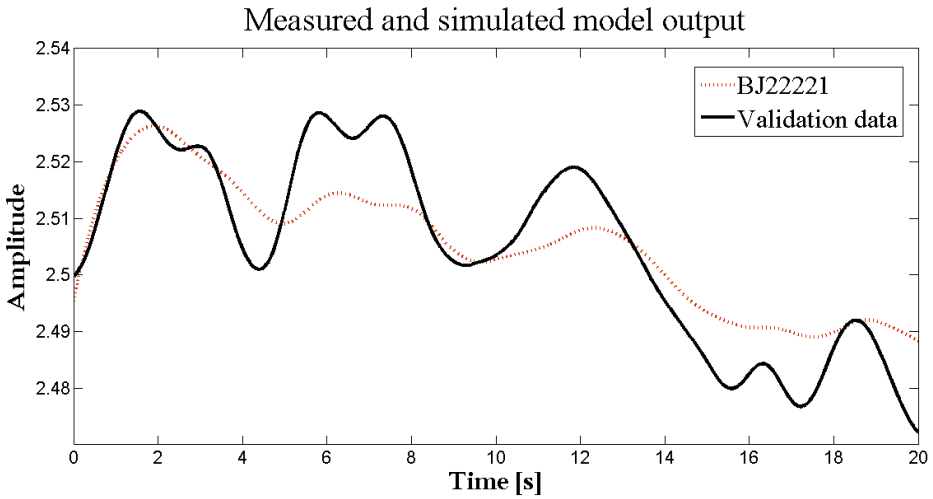


Figure 7-3: Fit of 51.85% between BJ model output (dashed) and validation data (solid)

BJ22221 is the best polynomial model obtained. Trials with higher order BJ, OE and ARMAX models are conducted, however, it does not yield more desirable outcomes. Same trials are conducted for the rest of the transfer function identification as well to obtain the most satisfactory result.

7-1-2 Yaw transfer function identification

A second order discrete-time transfer function with one sample delay provides a visible concordance between inputs and outputs. Even though the third order BJ model shows somewhat improvement over the second order model, a lower order system is better than a higher order system when it comes to the more or less same best fits results.

$$\begin{aligned} B(z) &= -91.99z^{-1} + 272.6z^{-2} - 269.5z^{-3} + 88.84z^{-4} \\ C(z) &= 1 - 0.9363z^{-1} + 0.01623z^{-2} \\ D(z) &= 1 - 1.994z^{-1} + 0.9939z^{-2} \\ F(z) &= 1 - 1.986z^{-1} + 0.9857z^{-2} \end{aligned}$$

The table lists the fits between the identified transfer function output and the identification data, and identified transfer function output and the validation data.

Table 7-2: Fits for the identification and validation data

BJ42221	Identification data	Validation data
Fits	88.29%	70% $\sim$ 88%

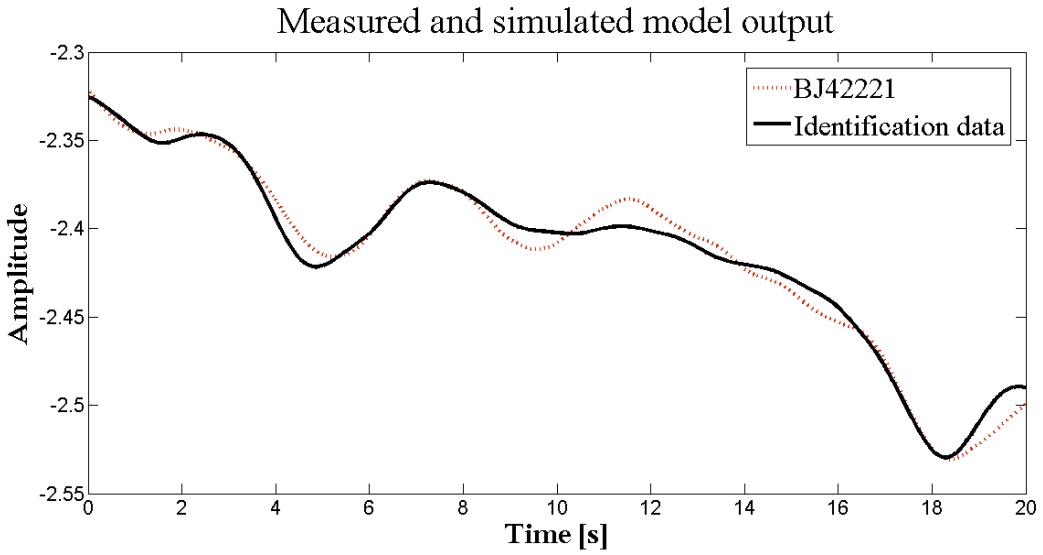


Figure 7-4: Fit of 88.29% between BJ model output (dashed) and identification data (solid)

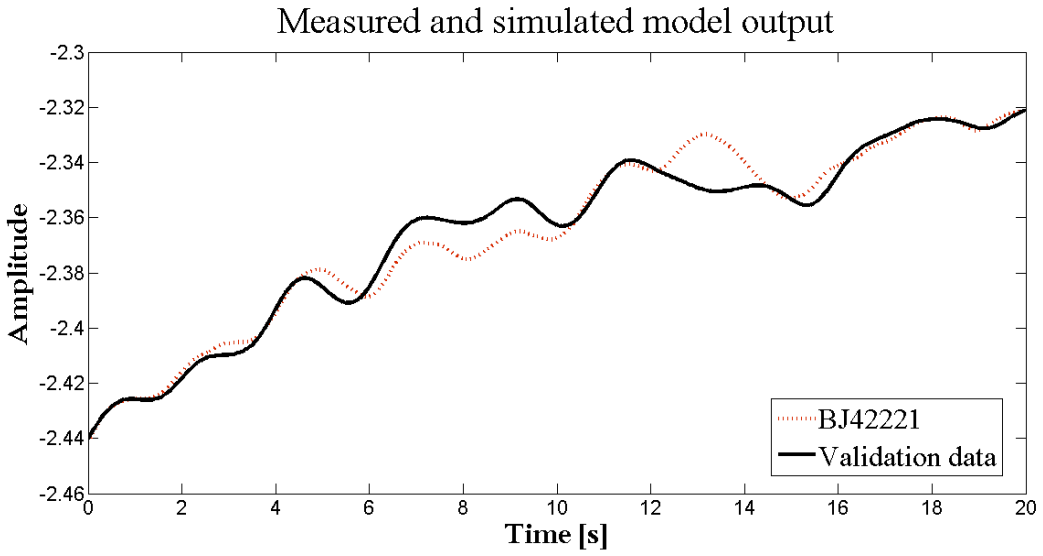


Figure 7-5: Fit of 78.41% between BJ model output (dashed) and validation data (solid)

BJ42221 is the best polynomial model obtained with excellent concordance between inputs and outputs, owing to the smaller coupling effect existing in yaw channel.

7-1-3 Pitch and roll transfer function identification

Due to the X-type quadrotor configuration, the coupling effect of pitch and roll channels plays the pivotal role in the transfer function identification, which gives rise to the less accurate estimation results since the coupling effect has been overlooked under the small angle assumption. Tables 7.3 and 7.4 illustrate the fits of the pitch and roll channels between the identified transfer

function output and the identification data, identified transfer function output and the validation data, respectively.

The pitch channel equations may be presented as:

$$\begin{aligned} B(z) &= 0.005255z^{-1} - 0.01175z^{-2} + 0.00658z^{-3} \\ C(z) &= 1 + 0.6075z^{-1} + 0.999z^{-2} \\ D(z) &= 1 - 1.998z^{-1} + 1.001z^{-2} \\ F(z) &= 1 - 1.993z^{-1} + 0.9934z^{-2} \end{aligned}$$

Table 7-3: Fits for the identification and validation data

BJ32221	Identification data	Validation data
Fits	45.3%	$10\% \sim 45\%$

The roll channel equations may be presented as:

$$\begin{aligned} B(z) &= 0.001148z^{-1} - 0.00116z^{-2} \\ C(z) &= 1 + 0.7265z^{-1} - 0.2721z^{-2} \\ D(z) &= 1 - 1.995z^{-1} + 0.9947z^{-2} \\ F(z) &= 1 - 1.991z^{-1} + 0.9908z^{-2} \end{aligned}$$

Table 7-4: Fits for the identification and validation data

BJ22221	Identification data	Validation data
Fits	43.67%	$8.518\% \sim 43\%$

Obviously, the fits of these two channels show the relatively poor performance in marked contrast to the throttle and yaw channels. Besides, the value of the fits between the output of the identified BJ polynomial model and the validation data to a large extent depend on the choice of validation data set and fluctuates strongly. This can be fixed in essence by changing the configuration of AR Drone from X-type to cross-type configuration, thereby reducing the coupling effect.

7-2 Kalman Filter Estimation

In simulation, both EKF and UKF are easy to tune as long as there is a guarantee of the concordance between the simulator and the model in Kalman filter, including the initial guess, noise matrices and the model structure. But in reality, there is no guarantee for tuning them correctly except for the measurement covariance matrix and the initial guess for the state. Besides, it can be found that too many uncertainties and constraints appear, which render the parameter estimation extremely complex. All these constraints and uncertainties will be explained in detail in Sect. 7-3. To begin with, all the relevant flight tests, the corresponding noise matrices and initial guess are introduced, in the next, the performance of EKF and UKF are presented.

7-2-1 Flight tests

The flight tests have been done by stages.

1. **First stage:** the data has been collected only at hover.
   Problem: there is no excitation; the parameter is not able to be estimated under such circumstance (p, q, r are basically equal to zero).

2. **Second stage:** enlightened by the failure of the parameter estimation at first stage, the excitation for the system is required: in the meantime, hovering is a matter of great account since the nonlinear model is described under the near-hover assumption. Therefore, the most commonly used flight test manoeuvre for rotorcrafts, the "doublet" is proposed, which refers to two opposing movements in quick succession.
   Problem: no sufficient excitation.

3. **Third stage:** unfortunately, a doublet was not sufficient enough to excite all the channels. It can be observed, in Eq. 7-2, that for the good estimation of parameters, the excitation of angular rate is important. Thus the third flight test is to excite the roll, pitch, and yaw one after another. Noted that there is no direct method to give the explicit roll, pitch, yaw commands in Paparazzi, changing the control law is one way to obtain the relatively aggressive angular rate. The full flight test encompasses a sequence of flight missions, including take-off, hover, roll, pitch, yaw movement, doublet, and landing. The sampling rate is 100 Hz with near 400 seconds full flight test, 40000 iterations can be obtained to make a good implementation of the algorithms.

$$\begin{cases} \dot{p} &= \frac{I_y - I_z}{I_x} qr + \frac{I_r}{I_x} q\Omega + \frac{\tau_\phi}{I_x} \\ \dot{q} &= \frac{I_z - I_x}{I_y} pr - \frac{I_r}{I_y} p\Omega + \frac{\tau_\theta}{I_y} \\ \dot{r} &= \frac{I_x - I_y}{I_z} pq + \frac{\tau_\psi}{I_z} \end{cases} \tag{7-2}$$

7-2-2 Measurement noise variance matrix

The measurement noise variance matrix is derived by computing the variance of each sensor, the matrix is expressed as:

$$\begin{aligned} R = diag[&1.0407 \times 10^{-3} \quad 1.1986 \times 10^{-3} \quad 1.5536 \times 10^{-3} \quad 7.2926 \times 10^{-4} \quad 1.938 \times 10^{-4} \\ &5.3423 \times 10^{-4} 2.7063 \times 10^{-5} \quad 3.1965 \times 10^{-5} \quad 1.4916 \times 10^{-4} \quad 3.5826 \times 10^{-2} \\ &2.8408 \times 10^{-2} \quad 8.0080 \times 10^{-2}] \end{aligned} \tag{7-3}$$

For the better performance, a little adjustment over the calculated measurement noise variance matrix has been tuned based on the same order of magnitude and fluctuating around the calculated matrix. In terms of the initial state, the calculated moment of inertia can be regarded as a good initial guess. Last but not least, there is still no practical or theoretical guidance to tune the process noise variance matrix Q and the initial covariance matrix P_0, trial and error method is the only way to achieve the goal.

7-2-3 UKF

Since UKF tends to be more robust while EKF is different to implement and tune, it is more preferable to start from tuning UKF in the first place. The initial state is selected unchangeably; hence, the core of tuning lies in the trade-off between the noise matrices and the initial covariance matrix. Although, theoretically speaking, it is not difficult to tune, the state estimation can be obtained fairly easy while the parameter estimation is extremely hard, and the reason will be analyzed at length afterwards.

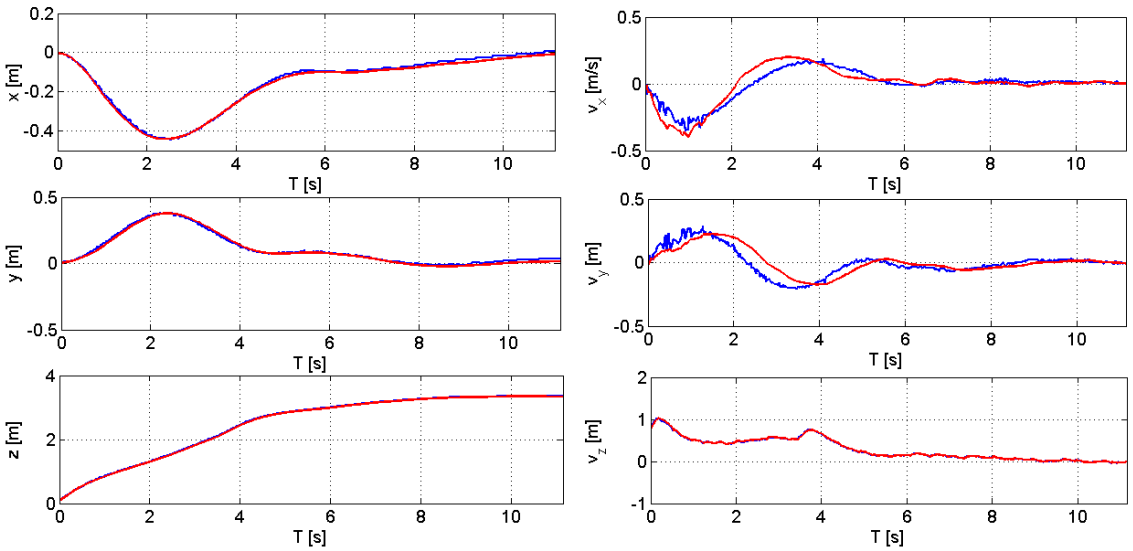


Figure 7-6: UKF state filtering: measured (blue) and estimated (red) for position and velocities at x, y, and z-axes respectively.

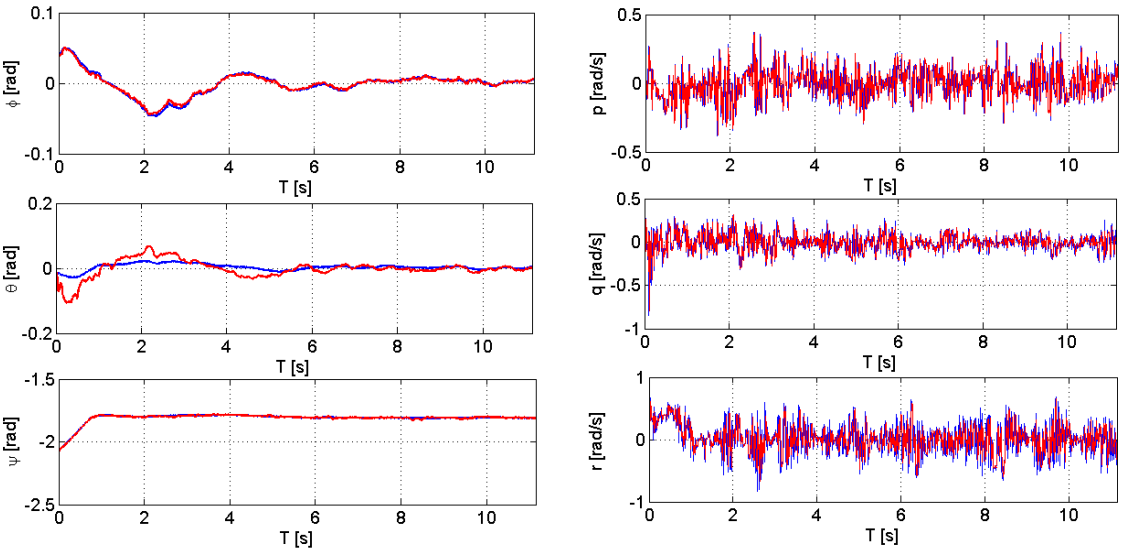


Figure 7-7: UKF state filtering: measured (blue) and estimated (red) for Euler angles and angular velocities at x, y, and z-axes respectively.

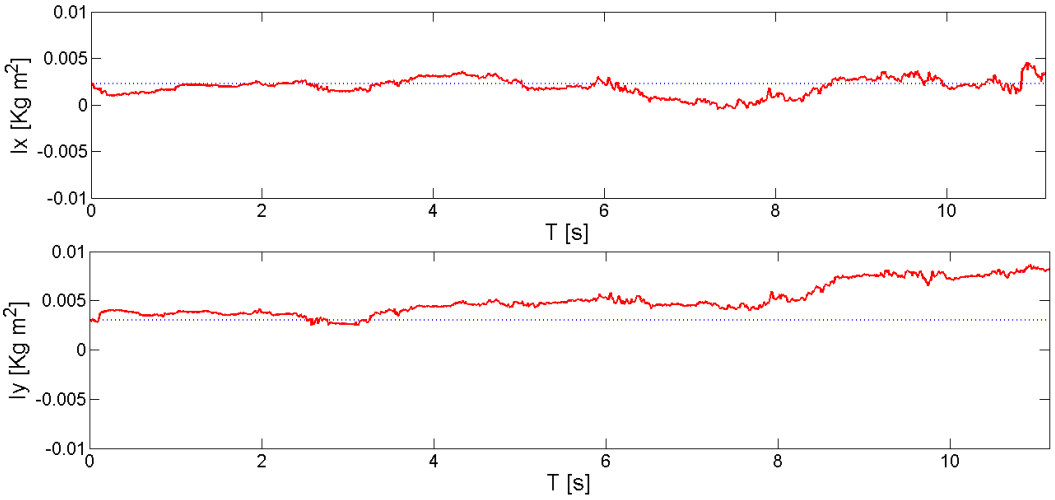


Figure 7-8: Compare the reconstruct of moment of inertia (red) by UKF algorithm with calculated value (dotted blue line) for x and y axes respectively.

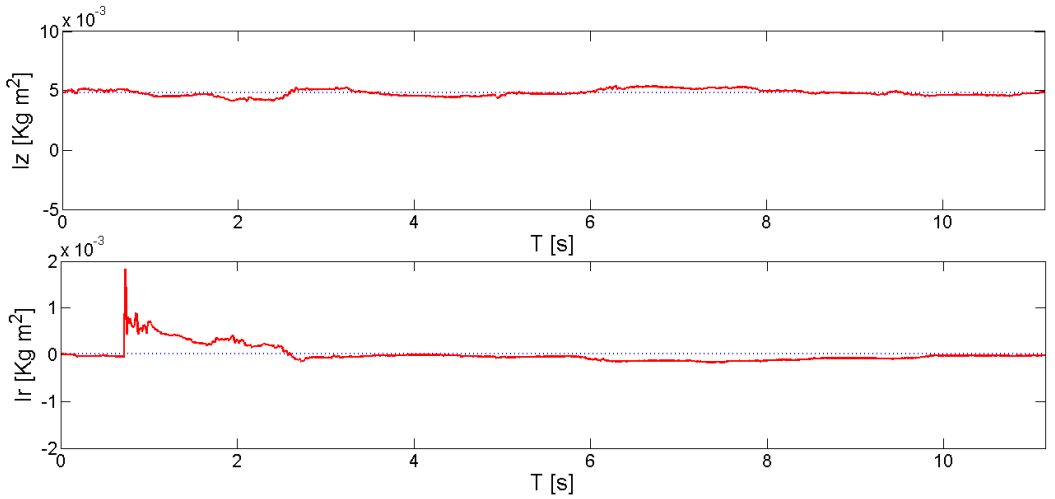


Figure 7-9: Compare the reconstruct of moment of inertia (red) for z axes and moment of initial for each rotor (red) by UKF algorithm with calculated value (dotted blue line)

The state filtering performance of UKF is shown in Fig. 7-6, 7-7, 7-8 and 7-9, showing that the reconstruction values for moment of inertia around the x and z-axes represent a high concordance with the calculated data in Table 4-1 while the reconstruction values for the moment of inertia around the y-axes and the moment of inertia of the rotor represent a relatively low concordance.

The Table illustrates the performance of the parameter estimation of UKF and the error of the estimation.

Table 7-5: The table illustrates the performance of the parameter estimation of UKF

Parameters	Unit	Calculated value	Reconstructed value	Mean square error
I_x	$kg \cdot m^2$	$2.238 \cdot 10^{-3}$	$2.619 \cdot 10^{-3}$	$9.5581 \cdot 10^{-7}$
I_y	$kg \cdot m^2$	$2.985 \cdot 10^{-3}$	$7.589 \cdot 10^{-3}$	$6.7202 \cdot 10^{-6}$
I_z	$kg \cdot m^2$	$4.805 \cdot 10^{-3}$	$4.815 \cdot 10^{-3}$	$8.0534 \cdot 10^{-8}$
I_r	$kg \cdot m^2$	$2.0296 \cdot 10^{-5}$	$0.9788 \cdot 10^{-7}$	$3.3682 \cdot 10^{-8}$

The reconstructed values in Table 7-5 are the mean of the moments of inertia when they are stable. The calculated values can be treated as the close approximation of the true values, thereupon the estimation of I_x and I_z is considered to be reasonable. Accordingly, the estimation of I_r seems not valid due to the wrong order of magnitude, which is even smaller than one hundredth of the calculated one. As for I_y, the estimation is imprecise as well owing to the coupling effect of I_r and I_y. Even so, it can be also concluded that UKF is a practical tool for the parameter estimation under the condition of the appropriate identification model and accurate data.

7-2-4 EKF

Due to the sensitivity to the initial guess and noise covariance matrix, EKF is much harder to tune. Furthermore, it is less robust to model uncertainty and disturbance from the outside world; therefore, more strict requirements for the concordance between the nonlinear model and AR Drone as well as for the measurement of sensors are necessary for EKF.

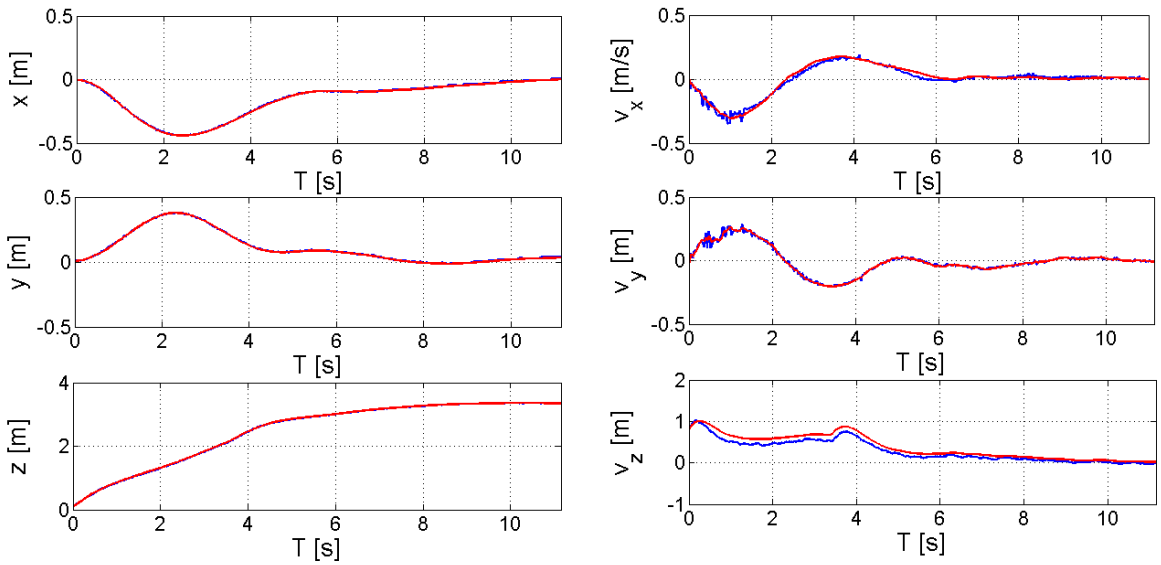


Figure 7-10: EKF state filtering: measured (blue) and estimated (red) for position and velocities at x, y, and z-axes respectively.

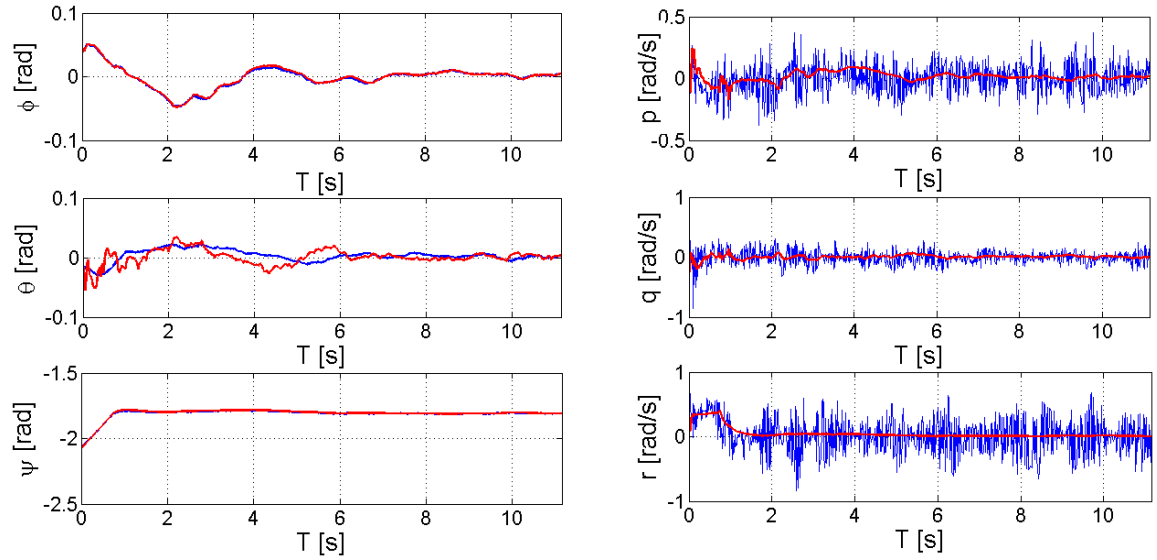


Figure 7-11: EKF state filtering: measured (blue) and estimated (red) for Euler angles and angular velocities at x, y, and z-axes respectively.

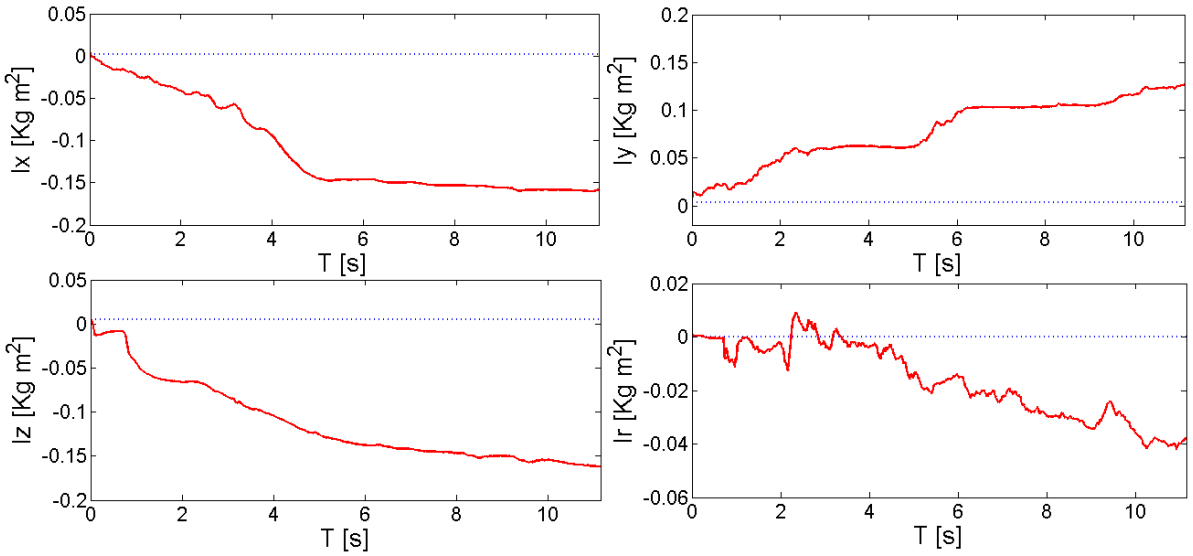


Figure 7-12: Compare the reconstruct of moment of inertia (red) by EKF algorithm with calculated value (dotted blue line) for x, y and z axes respectively as well as the moment of inertia of rotor

The performances of EKF for system identification purposes applied on a AR Drone 2 have been illustrated in Fig. 7-10, 7-11 and 7-12. Obviously, EKF is unable to converge, let alone converge to the true values. The attempts of using the linear Kalman filter to predict the next estimation may cause the filter to quickly diverge, which is clearly shown in Fig. 7-12.

7-3 Results Comparison and Analysis

In conclusion, it is clear that the performance of both EKF and UKF allows them to be practical tool for state estimation, which is manifest in chapter 5 and 6 theoretically. When applying these algorithms on the real flight data, EKF shows less computational cost, whereas less robust and accurate than the UKF in terms of parameter estimation. As an improvement over EKF, the performance of UKF proves itself to be one of the practical tools for parameter estimation using the real-time flight data. Although parts of the results are not satisfactory, this is reasonable and can be explained by the following constraints and uncertainties:

- **Lack of or inaccuracy of measurements of the thrust and torque;**

  In chapter 4, the relations between the rotor speed and thrust, the rotor speed and torque, are identified by using the setups in the Fig. 4-3 and 4-4. Theoretically, the static weighting scale readings are used as the measured static torque and thrust values. But in practice, the values are not settled during manual recording, the fluctuation of the weighting scale reading is particularly apparent for the torque measurement due to the small weighting scale reading (normally less than 10 grams). Take a look on the following equation,

$$\begin{aligned} \dot{p} &= \frac{I_y - I_z}{I_x} qr + \frac{I_r}{I_x} q\Omega + \frac{\tau_\phi}{I_x} \\ \dot{q} &= \frac{I_z - I_x}{I_y} pr - \frac{I_r}{I_y} p\Omega + \frac{\tau_\theta}{I_y} \\ \dot{r} &= \frac{I_x - I_y}{I_z} pq + \frac{\tau_\psi}{I_z} \end{aligned}$$

  Obviously, according to the equations above, it is theorized with a high degree of probability that the estimation of τ_ϕ, τ_θ and τ_ψ might be imprecise, which leads to the inaccuracy of parameter estimation of I_y and I_r. To be more specific, if one of the parameters

cannot be estimated correctly, the rest of the unknown parameters cannot be identified either owing to the coupling effect. In summary, based on all the analysis above, both EKF and UKF can be used as a practical tool for parameter estimation. UKF can be a better algorithm since it is more robust and easier to tune with appropriate model and data. In this work, the input data is deemed to be less precise as a result of the inaccurate measurement of experimental instruments.

- **Sensor bias drift;**

  The bias drift refers to the variation of the bias over time. This drift is basically caused by the self heating of the sensors (especially for gyro), and their associated electrical and mechanical components.

- **The assumption of near-hover is not true;**

  The nonlinear dynamical model is based on several assumptions; the near hover assumption is the most important one since the model neglects the aerodynamics effect, for instance the hub force when the AR Drone undergoes the translational movement; in the point of fact, this assumption is not true.

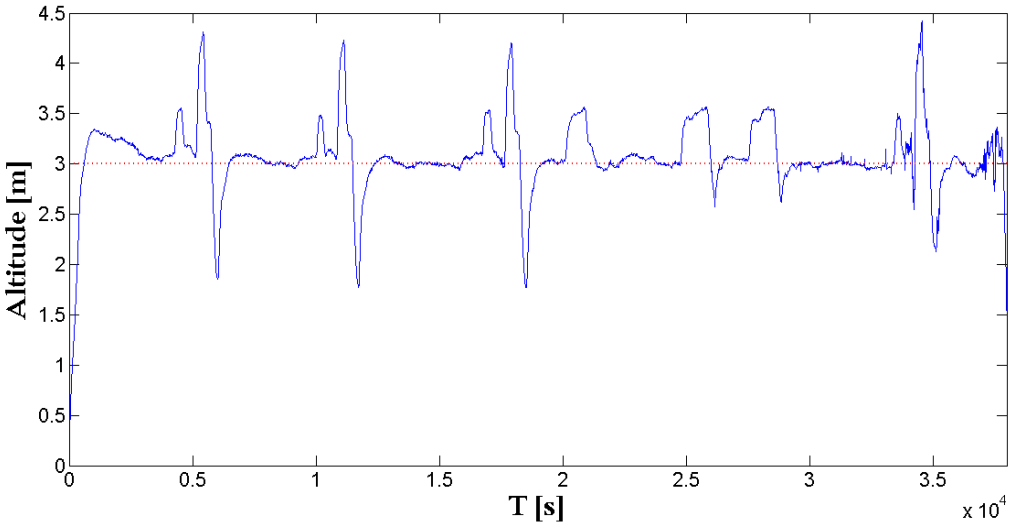


Figure 7-13: The red dotted line is near hover condition of three meters, the blue line is the real flight altitude of AR Drone 2

Fig. 7-13 represents the altitude of the real flight data of AR Drone 2 which is randomly picked from massive data sets. The height of hovering is set as three meters; it can be observed that the fluctuation of the hovering height appears throughout the whole flight test. By calculating, the error fluctuates and can be even largely up to 0.2 meters. Thus this assumption has been shown to be not accurate enough.

- **Different AR Drone 2 is used throughout the whole project.**

  There is no fixed AR Drone 2 available in MAVLab, the platforms used for engine test, thrust and torque experiment and the real flight tests are not same. Seeing that the AR Drone 2 was broken or being used by others, even for the real flight tests, a different AR Drone 2 was used during different time periods. This will lead to inconsistencies of the physical parameters of AR Drone 2 for each flight test and experiment. As the matter of fact, the AR Drone 2 can be used and changed by anyone in MAVLab, however, it is unpractical to do the model parameter estimation (the same procedure as in chapter 4) after the new AR Drone 2 is available. Based on all the factors mentioned above, there is no definite way to assure the concordance of the model and estimated parameters for the different AR Drone 2.

In general, the implementation of UKF and EKF for parameter identification purposes applying on an AR Drone 2 requires the appropriate identification model and identification data. The main factors which might be the reason of the less satisfactory performance for parameter identification have been discussed at large, according to which, it is most likely that the inaccurate measurement of the thrust and torque of each rotor gives rise to the inexact inputs for the nonlinear dynamical model, thereby leading to an imprecise calculation of the moment of inertia. In addition to the main factors, many assumptions seem to be not quite realistic, for instance, the AR Drone 2 is assumed to be structurally symmetrical and rigid, however, it turns out that the AR Drone 2 is only symmetrical about xz panel and the components of AR Drone 2 are not rigid apparently.

As for the linear transfer function identification, changing the configuration of AR Drone will largely improve the accuracy of identification performance for roll, pitch channels. After regathering the flight data, there is a high probability that it will be possible to implement controllers for this linear model.

Chapter 8

Conclusion and Future Work

The previous chapters have described the dynamics of the AR Drone, the results of modelling, parameter estimation, simulation and identification of AR Drone 2, it can be concluded that, according to the simulation results, EKF, UKF and MRUKF can be treated as the practical tools for system identification. However, the model uncertainty and inaccuracy of measurements make it more difficult to tune and severely impact the performance of EKF and UKF. The main conclusions and recommendations in this work are discussed in this chapter in more detail.

8-1 Concluding Remarks

In chapter 2, the nonlinear dynamical model and the linearized model are discussed, however, the model inputs, the total thrust and roll, pitch and yaw movement, cannot be measured directly, which gives rise to parameter estimation of the engines and rotor aerodynamics in chapter 4. Constrained by the experimental setups, the measurement of thrust and torque for each rotor is imprecise and it is hard to compensate the error. This will lead to the inaccurate inputs for the nonlinear dynamical model; it is especially recommended that the engine dynamics and the aerodynamics should be investigated precisely.

Next, the algorithms of proposed methods are introduced and the simulator is designed (before the model parameter estimation) to verify the effectiveness of EKF, UKF, MREKF and MRUKF for parameter identification purposes. In addition to MREKF, the rest of algorithms manage to identify parameters precisely when the same initial guess and noise matrices are given. They have proved themselves to be practical tools in the field of system identification theoretically. Additional information on the successful implementation of the algorithms emphasizes that the performance of EKF, UKF and MRUKF state filtering for parameter identification purpose applied on AR Drone 2 largely depends on consistency of the model (including appropriate model, precise initial estimate of the state and covariance matrix, noise covariance matrices) and accuracy of input-output data. An inappropriate model or data will end up with the incorrect estimation of parameters.

As for linear model identification, this work approximates the system as decoupled Single Input Single Output (SISO) systems rather than identifying a Multiple Input Multiple Output (MIMO) system. The coupling effects between the systems are assumed to be the disturbances acting on those systems. The BJ polynomial models have been identified based on the prediction error minimization method, and the low fits for roll and pitch channels indicate that the coupling effects are severely significant by virtue of the X-type configuration. These

coupling effects can be successfully rejected to some extent by changing the AR Drone configuration to cross-type in theory. However, this has not yet been verified.

As the core of this work, the nonlinear system identification based on the EKF and UKF is tested by using the recorded real flight data. It can be found that EKF is more sensitive to the initial guess of the state and noise matrices; therefore, it is unable to converge, let alone converge closely to the calculated value. On the contrary, as an improvement over EKF, UKF is capable of converging, it proves itself as a more robust and accurate tool of system identification than EKF. Despite all this, the estimation of rotor inertia seems to be incorrect apparently owing to the wrong measurements of thrust and torque. The wrong measurements of thrust and torque result in the wrong estimation of the input data, thereby computing the imprecise I_r and I_y .

In conclusion, with appropriate identification model and data, both EKF and UKF can be used as practical tools for system identification, at mean time, UKF tends to be more robust and accurate in contrast to EKF. In this work, constraint of experimental instruments led to the less satisfying performance of system identification. This means the implementation of EKF and UKF for system identification purposes calls for the more accurate measurements and setups. If there is no guarantee for the accuracy, the tuning of EKF and UKF tends to be more time-consuming and less reliable. Under this circumstance, it is highly recommended to implement other identification techniques.

8-2 Recommendations and future work

The following future works are recommended;

1. Specialized equipment (for instance, wind tunnel) can be utilized to correctly measure the thrust and torque, which assures the appropriate input data for the nonlinear model.
2. Since the near hover condition is not satisfied in practice, the more complex model is taken into account (please refer to model 1 [19]), which considers all the aerodynamics and friction. It might be a big challenge to identify this model due to its complexity, and this model has been proposed for principle analysis and simulation[13].
3. As AR Drone 2 is neither rigid nor symmetrical; it is more preferable and applicable to identify the moment of inertia by using professional software, for instance AutoCAD.
4. In terms of system identification, it is always recommended to use the same machine, on the ground that, even different machines of the same type are unlikely to have identical unknown parameters.
5. It is also recommended to tune EKF on line.

Appendix A

Data Sets for Model Parameter Estimation

This appendix presents the data sets collected with the designed experimental setups for model parameter estimation purposes.

Table A-1: Measurements of weight, dimensions and distance of the center of gravity, the moment of inertial along x, y, z-axes can be computed

Part	Number	Weight (g)	X-distance [m]	Y-distance [m]	Z-distance [m]
Propeller gear	4	2.36	0.1262	0.1262	-0.0125
Propeller	4	3.55	0.1262	0.1262	-0.024
Engine	4	17.7	0.1128	0.1128	0.008
Propeller axis	4	0.88	0.1128	0.1128	-0.014
Carbon rod	4	10.7	0.1146	0.1146	0
Engine holder	4	4.3	0.0668	0.0668	0
Engine cable	4	1.9	0.0495	0.0495	0
Center cross	4	8.775	0.0160	0.0160	0
Battery	1	119.1	-0.0120	0	-0.0290
Main board	1	32	-0.0245	0	0.0125
Sonar	1	8.34	0.0500	0	0.0190
Camera	1	2.92	0.1900	0	-0.0150
Front board	1	6.36	0.0110	0	0.0125
Cover sheet	1	6.34	0	0	0.0270
Top foam	1	30.8	0.0270	0	-0.0350
Plastic square	1	2.06	-0.0125	0	0.0090
Battery holder	1	19.5	-0.0120	0	0.0160
Bottom loam	1	23.24	0.0490	0	0

Table A-2: Measurements of weight, dimensions and distance of the center of gravity, the moment of inertial along x, y, z-axes can be computed

Part	Shape	X-dimension/ radius[m]	Y-dimension/ radius[m]	Z-dimension/ height[m]
Propeller gear	Circle plate	0.0350	0	0.0025
Propeller	Circle plate	0.1980	0	0.0100
Engine	Cylinder	0.0120	0	0.0300
Propeller axis	Line	0	0	0.0360
Carbon rod	Bar	0.1050	0	0
Engine holder	Box	0.0400	0.0150	0.0200
Engine cable	Line	0.1400	0	0
Center cross	Bar	0.0370	0	0
Battery	Bar	0.0720	0.0390	0.0240
Main board	Square plate	0.0590	0.0450	0.0050
Sonar	Bar	0.0160	0.0450	0.0150
Camera	Point	0	0	0
Front board	Square plate	0.0240	0.0510	0.0050
Cover sheet	Square plate	0.1200	0.0850	0
Top foam	Bar	0.3500	0.0550	0.0700
Plastic square	Line square	0.0710	0.0590	0
Battery holder	Bar	0.0970	0.0200	0.0200
Bottom loam	Bar	0.2700	0.0250	0.0250

Table A-3: Measurements of weight, dimensions and distance of the center of gravity, the moment of inertial along x, y, z-axes can be computed

Part	I_x (local) $[kg \cdot m^2]$	I_y (local) $[kg \cdot m^2]$	I_z (local) $[kg \cdot m^2]$
Propeller gear	1.8192E-07	1.8192E-07	3.6138E-07
Propeller	8.7280E-06	8.7280E-06	1.7397E-05
Engine	1.4868E-06	1.4868E-06	3.1860E-07
Propeller axis	9.5040E-08	9.5040E-08	0
Carbon rod	6.9513E-06	6.9513E-06	9.8306E-06
Engine cable	2.1944E-06	2.1944E-06	3.1033E-06
Center cross	7.0787E-07	7.0787E-07	1.0011E-06
Battery	2.0813E-05	5.7168E-05	6.65471E-05
Main board	5.4667E-06	9.3493E-06	1.4683E-05
Sonar	1.5638E-06	3.3430E-07	1.5853E-06
Camera	-	-	-
Front board	1.3918E-06	3.1853E-07	1.6838E-06
Cover sheet	3.81721E-06	7.6080E-06	1.1425E-05
Top foam	2.0341E-05	3.2699E-04	3.2218E-04
Battery holder	1.300E-06	1.59340E-05	1.5940E-05
Bottom loam	2.4208E-06	1.4239E-04	1.4239E-04

Table A-4: Measurements of thrust by using the weighting scale and the corresponding thrust command to each engine

Input [-]	Thrust tests [grams]									
	1	2	2	4	5	6	7	8	9	10
0	48.1	49	47	48	49	48	47	-	48	-
500	63.3	65	63	64	-	-	-	-	-	-
1000	82.5	82	82	85	81	83	81	85	82	83
1500	102.5	105	103	105	-	-	-	-	-	-
2000	128.6	130	127	131	128	133	128	125	126	125
2500	151	156	157	155	-	-	-	-	-	-
3000	186	185	187	184	186	190	182	185	185	182
3500	220	215	220	215	-	-	-	-	-	-
4000	252	256	255	251	256	250	251	261	253	249
4500	291	285	295	292	-	-	-	-	-	-
5000	343	335	328	335	335	333	-	335	336	330
5500	378	382	374	370	-	-	376	-	-	-
6000	423	430	420	435	430	431	428	425	433	423
6500	480	485	478	478	-	-	-	-	-	-
7000	528	540	535	535	530	530	532	535	529	525
7500	580	570	590	581	-	-	-	585	-	580
8000	651	635	645	635	650	-	645	655	628	640
8500	710	710	690	710	-	720	-	710	-	710
9000	780	745	765	775	730	746	729	740	-	735
9500	790	785	812	785	-	785	-	760	-	755

Table A-5: Measurements of torque by using the weighting scale and the corresponding thrust command to each engine

Input [-]	Thrust tests [grams]												
	1	2	2	4	5	6	7	8	9	10	11	12	13
0	1.1	0.9	0.9	0.9	0.9	1.2	1.2	0.9	0.9	0.9	0.9	0.8	0.8
500	1.5	1.5	1.6	1.6	1.4	1.6	1.5	1.6	1.45	1.4	1.35	1.3	1.3
1000	2.1	1.8	2.0	2.1	2.0	2.3	2.4	2.2	1.8	2.0	1.7	1.7	1.8
1500	2.4	2.4	2.3	2.4	2.3	2.5	2.6	2.4	2.2	2.4	2.1	2.1	2.0
2000	2.7	2.6	2.7	2.7	2.8	2.6	2.7	2.8	2.3	2.4	2.5	2.3	2.4
2500	3.2	2.7	2.8	2.9	2.9	3.0	2.9	3.2	3.0	3.1	2.8	2.7	2.8
3000	3.4	3.1	3.2	3.5	3.2	3.4	3.4	3.6	3.2	3.5	3.0	3.3	3.3
3500	3.8	3.4	3.4	3.7	3.4	3.5	3.6	3.8	3.4	4.0	3.5	3.5	3.5
4000	4.1	4.1	3.8	4.1	3.8	4.0	4.0	4.4	4.0	4.3	3.8	3.9	4.1
4500	4.5	5.0	4.8	5.1	4.3	4.6	4.7	5.0	4.4	5.2	4.6	4.8	4.8
5000	6.1	5.5	6.0	5.4	6.0	6.1	6.1	6.2	6.3	5.7	6.2	5.0	6.4
5500	6.5	5.8	6.2	6.0	6.1	6.4	6.3	6.5	6.5	6.0	6.4	6.2	6.8
6000	7.0	6.8	6.8	6.2	6.8	6.9	7.3	7.3	7.5	6.9	7.7	7.4	7.3
6500	7.4	7.2	7.9	7.5	7.4	8.0	8.0	8.2	7.8	7.9	8.1	8.0	7.7
7000	7.8	7.7	8.2	8.2	8.3	8.4	9.2	8.8	8.2	8.4	9.0	8.5	8.2
7500	8.6	8.3	8.8	9.7	9.8	9.3	9.7	9.9	9.4	9.5	9.9	9.4	9.6
8000	10.0	10.0	10.5	10.5	10.8	10.8	11.0	11.0	10.7	10.7	11.0	11.0	11.3

Input [-]	Thrust tests [grams]											
	1	2	2	4	5	6	7	8	9	10	11	12
0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	1.0	1.0	1.0	0.9	0.9
500	1.5	1.4	1.5	1.5	1.5	1.5	1.5	1.4	1.5	1.4	1.5	1.5
1000	2.2	1.9	1.8	1.9	2.2	1.9	2.0	1.9	2.1	1.9	2.1	2.1
1500	2.3	2.0	2.4	2.1	2.5	2.3	2.3	2.3	2.3	2.2	2.3	2.4
2000	2.5	2.2	2.6	2.5	2.8	2.4	2.5	2.3	2.5	2.7	2.8	2.6
2500	2.8	2.7	2.9	2.9	3.1	3.2	2.9	2.9	2.8	2.9	3.2	3.1
3000	3.1	3.2	3.3	3.5	3.3	3.6	3.3	3.4	3.2	3.2	3.6	3.5
3500	3.6	3.7	3.7	3.9	3.8	4.0	3.7	3.7	3.7	3.9	3.8	3.7
4000	4.2	4.6	4.0	4.2	4.4	4.8	4.2	4.3	4.2	4.1	4.2	4.0
4500	4.9	5.0	4.8	5.1	5.4	5.3	5.2	5.4	4.9	5.1	5.2	5.0
5000	5.8	5.9	6.1	5.9	6.1	6.2	6.1	6.0	6.2	6.2	6.2	6.2
5500	6.5	6.1	6.5	6.3	6.4	6.4	6.7	6.6	6.6	6.9	6.9	6.5
6000	7.2	7.0	7.1	7.1	7.0	7.4	7.5	7.3	7.1	7.5	7.6	7.3
6500	8.0	7.8	7.9	7.7	7.6	8.1	7.8	8.2	7.9	8.3	8.1	8.0
7000	8.8	8.4	8.7	8.3	8.8	8.7	8.6	9.2	8.5	9.3	8.9	8.8
7500	9.5	10.1	10.0	10.3	10.1	9.9	10.0	9.8	9.8	9.9	10.0	9.7
8000	10.6	1.6	11.2	11.2	10.9	10.9	11.1	11.1	11.1	11.6	11.6	11.3

Appendix B

Transform Matrices

B-1 Plane rotation matrix R and T

The plane rotation matrix R associates the inertial frame with body-fixed frame and transforms from body-fixed frame to inertial frame.

$$\begin{aligned} R &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\psi\cos\theta & (-\sin\psi\cos\phi+\cos\psi\sin\phi\sin\theta) & (\sin\psi\sin\phi+\cos\psi\sin\theta\cos\phi) \\ \sin\psi\cos\theta & (\cos\psi\cos\phi+\sin\psi\sin\theta\sin\phi) & (-\cos\psi\sin\phi+\sin\psi\sin\theta\cos\phi) \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \end{aligned} \tag{B-1}$$

If the Euler angles ϕ, θ and ψ of AR Drone 2 are given, the conversion of a velocity or acceleration vector in the body-fixed frame to inertial frame can be fulfilled by multiplying it with the rotation matrix R and vice versa. The transform matrix T for rotational velocities which links the two frames is given by:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{B-2}$$

B-2 From ECEF to ENU coordinates

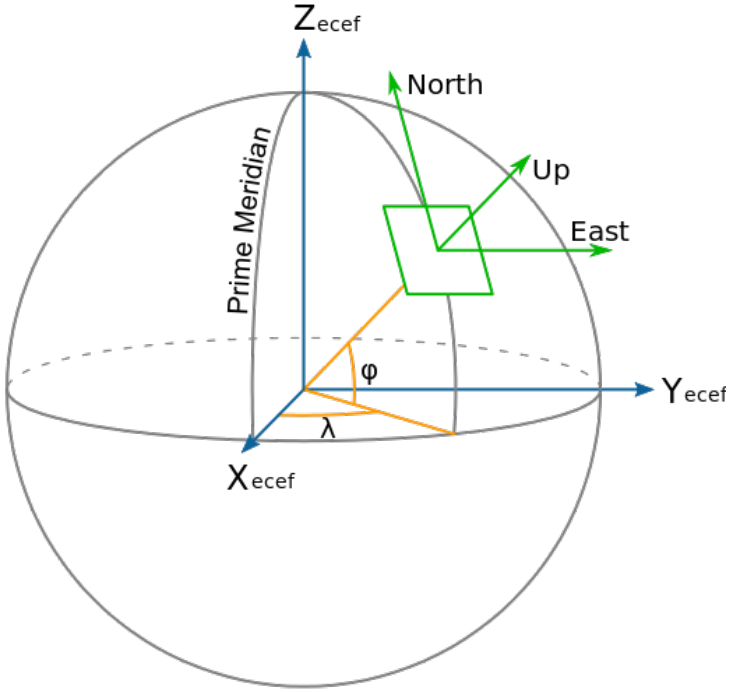


Figure B-1: Earth Centred Earth Fixed (ECEF) and East, North, Up (ENU) coordinates [1].

From the figure, it follows that the ENU coordinates can be transformed to the $\begin{bmatrix} x & y & z \end{bmatrix}$ ECEF by the two rotations:

1. A clockwise rotation over east-axis by an angle $(90° - \varphi)$ (here φ is latitude of the location of flight arena) to align the up-axis with the z-axis.
2. A clockwise rotation over the z-axis by an angle $(90° + \lambda)$ (λ is the longitude of the location of flight arena) to align the east-axis with the x-axis.

Then the transformation matrix from ECEF to ENU coordinates is given by [11];

$$\begin{bmatrix} E \\ N \\ U \end{bmatrix} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\cos\lambda\sin\varphi & -\sin\lambda\sin\varphi & \cos\varphi \\ \cos\lambda\cos\varphi & \sin\lambda\cos\varphi & \sin\varphi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{B-3}$$

Appendix C

State Estimation

Without tuning, another data set is randomly selected to test the performance of UKF and EKF.

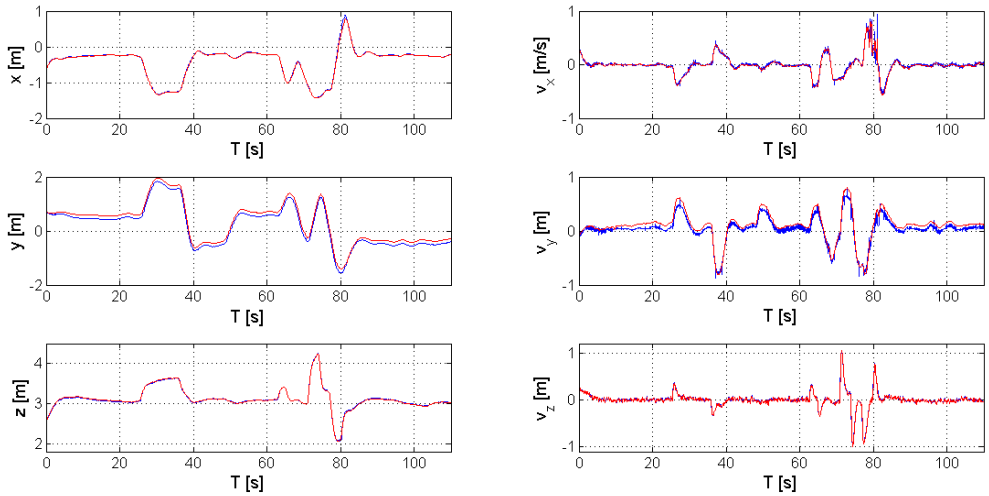


Figure C-1: UKF state filtering: measured (blue) and estimated (red) for position and velocities at x, y, and z-axes respectively.

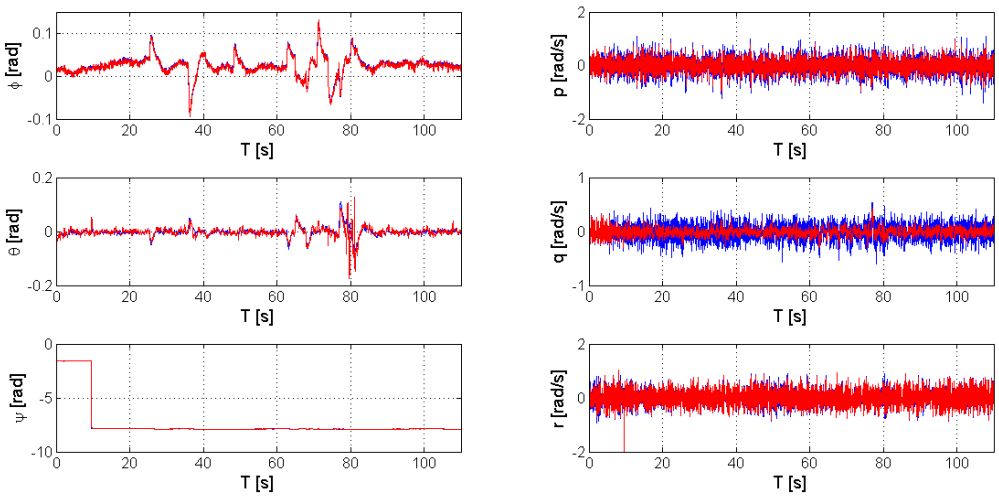


Figure C-2: UKF state filtering: measured (blue) and estimated (red) for Euler angles and angular velocities around the x, y, and z-axes respectively.

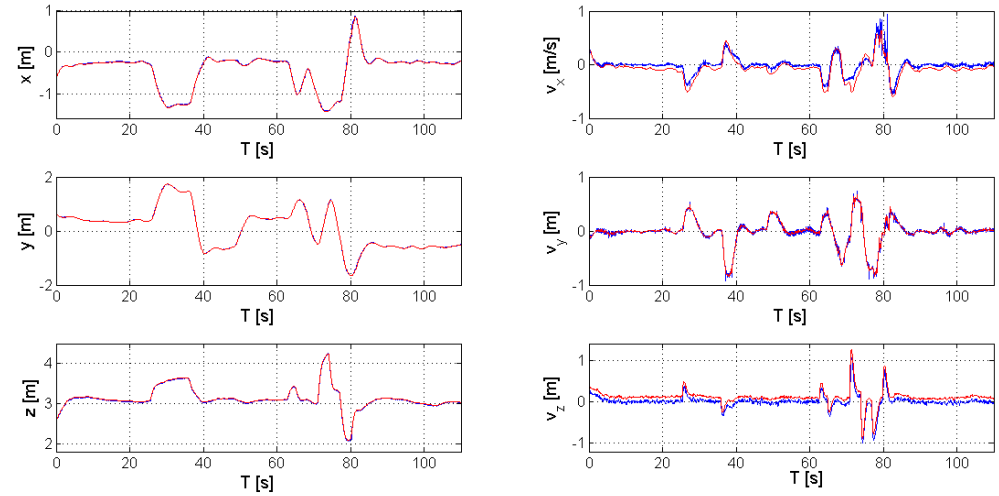


Figure C-3: EKF state filtering: measured (blue) and estimated (red) for position and velocities at x, y, and z-axes respectively.

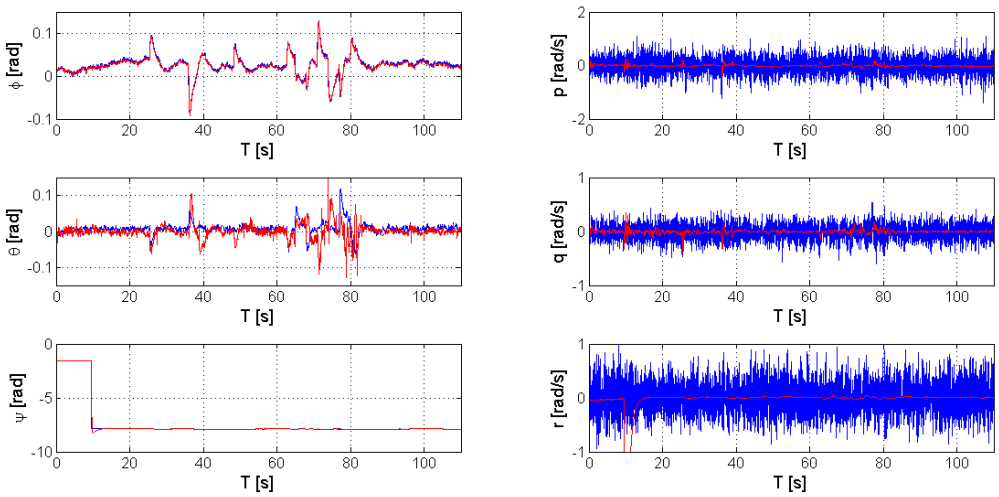


Figure C-4: EKF state filtering: measured (blue) and estimated (red) for Euler angles and angular velocities at x, y, and z-axes respectively.

Regardless of the small bias, the state estimation can be treated as accurate without changing Q, R, X_0 and P_0. From this point of view, the on-line or off-line controller can be designed based on Kalman filter.

Bibliography

[1] http://en.wikipedia.org/wiki/Geodetic_datum.

[2] https://abstract.cs.washington.edu/~shwetak/classes/ee472/notes/ARDrone_SDK_1_6_Developer_Guide.pdf.

[3] http://wiki.paparazziuav.org/.

[4] https://www.sparkfun.com/datasheets/Sensors/IMU/ps-imu-3000a-00-01.1.pdf.

[5] https://www.naturalpoint.com/optitrack/products/flex-13/.

[6] http://www.mhaerotools.deairfoilsprop_precession_english.htm.

[7] http://en.wikipedia.org/wiki/Extended_Kalman_filter.

[8] N. Abas, A Legowo, and R. Akmeliawati. Parameter identification of an autonomous quadrotor. In *Mechatronics (ICOM), 2011 4th International Conference On*, pages 1–8, May 2011.

[9] N. Abas, A. Legowo, Z. Ibrahim, N. Rahim, and A.M. Kassim. Modeling and system identification using extended kalman filter for a quadrotor system. *Applied Mechanics and Materials*, 313:976–981, 2013.

[10] R. Andreas. Dynamics identification and validation, and position control for a quadrotor. Swiss Federal Institude of Technology Zurich, Spring Term 2010.

[11] G. Aoude. *Threat Assessment for Safe Navigation in Environments with Uncertainty in Predictability*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, July 2011.

[12] L. Armesto, S. Chroust, M. Vincze, and J. Tornero. Multi-rate fusion with vision and inertial sensors. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 193–199 Vol.1, April 2004.

[13] S. Bouabdallah. Design and control of quadrotors with application to autonomous flying, 2007.

[14] T. Bresciani. Modelling, identification and control of a quadrotor helicopter. Master's thesis, Lund University, October 2008.

[15] L.R.G.C. Carrillo, A.E.D. Lopez, R. Lozano, and C. Pegard. Modeling the quad-rotor mini-rotorcraft. In *Quad Rotorcraft Control*, Advances in Industrial Control, pages 23–34. Springer London, 2013.

[16] G. Chowdhary and R. Jategaonkar. Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter. *Aerospace Science and Technology*, 14(2):106 – 117, 2010.

[17] T. Flanzer and A. Ning. Online parameter estimation for the adaptive control of unmanned aerial vehicles, Dec 2009.

[18] J.P. Jensen. Ensemble kalman filtering for state and parameter estimation on a reservoir model. Master's thesis, Norwegian University of Science and Technology, June 2007.

[19] Q. Li. The grey-box system identification of a quadrotor unmanned aerial vehicle, 2014.

[20] C.M. Lotion. Identification and control of a quadrotor in hover. Master's thesis, Delft University of Technology, November 2011.

[21] D.S. Miller. Open loop system identificaiton of a micro quadrotor helicopter from closed loop data. Master's thesis, University of Maryland, 2011.

[22] N.K. Peyada, A. Sen, and A.K. Ghosh. Aerodynamic characterization of hansa-3 aircraft using equation error, maximum likelihood and filter error methods. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume II, Hong Kong, March 2008.

[23] P. Sekhavat, M. Karpenko, and I. M. Ross. Ukf-based spacecraft parameter estimation using optimal excitation. In *Guidance, Navigation, and Control Conference*, August 2009.

[24] M.Q. Shao, G.W. Li, and H. Xu. Two-stage parameters identification method of unmanned helicopter based on subspace and pem. *Flight Dynamics*, April 2013. In Chinese.

[25] E.A Wan and R.V.D. Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158, 2000.

[26] L. Wang and G. Hugues. *System Identification, Environmental Modelling, and Control System Design.* 1st Edition. Springer, October 2011.

[27] W. Yuan and J. Katupitiya. A time-domain grey-box system identification procedure for scale model helicopters. In *Proceedings of the 2011 Australasian Conference on Robotics and Automation*, Sydney NSW 2052, Australia, 2011.