TESINA FIN DE MÁSTER

Autor: Ricardo Sanz Díaz
Director: Dr. Pedro José García

IMPROVING ATTITUDE ESTIMATION AND CONTROL OF QUADROTOR SYSTEMS

departamento de ingeniería de sistemas y automática

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Contents

Chapter 1

Introduction

1.1 Unmanned Aerial Vehicles

In the past years there has been an increasing interest in Unmmaned Aerial Systems (UAS). It is important to mention that UAS regulations are becoming more strict worldwide, limiting their range of applications and requiring the pilots to obtain a license which guarantees they have the necessary skills to remotely operate the aircraft in a safe manner. In the case of Spain, the flight of UAVs for commercial and civil purposes was regulated the 5th of July of 2014.

Among the UAS, quadrotors are of special interest in control from both perspectives, theoretical and applied [1]. They have been used as testbed platforms for validation of non-linear [2], robust and predictive controllers [3]. Disregarding the control strategy, a high-performance attitude tracking subsystem is a requisite for developing any other high-level controlling task. A good example of this statement can be found in [4], where a full control (vision, collision avoidance, landing/taking-off) is developed relying on the attitude control.

Developing a quadrotor vehicle involves several challenges. They are unstable systems, governed by coupled nonlinear equations. Furthermore, they rotational dynamics is quite fast and thus, control algorithms have to be implemented with a small sample period. This enforces strong requirements on the architecture of the device (motors, drivers, communications, real-time operative systems).

Sensing is the first challenge when it comes to controlling quadrotors. The key state variables to be estimated are the attitude and the angular velocity, as they are the primary variables used in attitude control of the vehicle [5]. Inertial Measurement Units (IMUs), which are the core of lightweight robotic applications have experienced a proliferation, resulting in cheaper, and more accurate devices [6]. The emergence of cheaper IMUs makes it

possible to use UAS for civil purposes like ground traffic inspection [7], forest fire monitoring [8] or real-time irrigation control [9]. Researchers have also been able to push the limits beyond traditional control stabilization, achieving outstanding acrobatic maneuvers [10].

It is thus both a challenging and an interesting task to obtain a reliable attitude estimation using low-cost sensors. This is difficult due to the low performance of the sensors which restricts the quality of the resulting estimation. And it is definitely interesting because the problem of obtaining an accurate attitude estimation is crucial and it usually represents a large portion of the cost of an UAS [6].

The sensor fusion problem consists of obtaining an optimal estimation of the required vehicle state variables with the direct measurements from multiple sensors. There are many possible solutions to this problem, e.g., Kalman filters [11, 12] or complementary filters [13]. The major part of this thesis will be focused on the problem of attitude estimation.

Control of quadrotors involves also challenges such as parametric uncertainties, nonlinearity, coupling and external disturbances [14, 15]. A high-performance attitude tracking subsystem is a prerequisite for developing any other high-level control tasks [16, 4]. In practical applications, the attitude of an UAV is automatically stabilized via an on-board controller and its position is generally controlled by an operator through a remote control system [17]. A wide variety of attitude controllers already exist in the literature. Some of them are based on classical techniques, whilst others employ nonlinear and robust control strategies; see for example [3, 18, 15] and the references therein. The last part of this thesis will treat the problem of controlling qudarotor systems.

1.2 Thesis Organization and Contributions

This thesis summarizes the work carried out for a period of one year related to improving the estimation and control of quadrotor vehicles. Such work derived into several publications which are spread out along this thesis. The rest of it is structured as follows:

The preliminaries can be found in Chapter 2, where probability concepts and the Kalman Filter are reviewed. Also the mathematical representation of 3D-orientations of a rigid body is treated. An insight into MEMS (Micro-Electro-Mechanical Systems) sensors, which are the basis of modern attitude estimation devices for lightweight robotics, is also given. The chapter finishes with an explanation of the experimental platforms used to validate the algorithms presented along the rest of the thesis.

Chapter 3 contains two sections, each of them related to an individual publication. The first section is based on [19], presented at the *World Congress on Intelligent Control and*

Automation (WCICA) 2014, where a comparison between different estimation algorithms can be found. The second section is focused on an efficient attitude estimation algorithm [20] presented at the *International Conference of Unmanned Aircraft Systems (ICUAS) 2014*.

The problematic of time delays in attitude estimation is introduced in Chapter 5. The whole chapter is dedicated to [21], presented at the *International Conference on Information Fusion (FUSION) 2014*, where a method to compensate the delay of the attitude measurements was introduced.

Chapters 5 and 6 are related to control of quadrotors. In Chapter 4, a brief introduction to modeling of quadrotor systems is given, while the rest of the chapter is related to a technique called Uncertainty and Disturbance Estimator (UDE). A complete derivation of this method is given and extensive results in real experiments are presented. This chapter is a revisited version of [22], which has been already submitted to the *IEEE Transactions on Control Systems Technology*, and it is still pending of acceptance.

Similarly, the last chapter introduces the theory of adaptive controllers based on reference models. Simulations and successful experimental results are also presented. This chapter is derived from , which was also presented at the *WCICA 2014*.

Chapter 2

Preliminaries

2.1 Random variables

2.1.1 General definitions

A random variable is a variable whose value is subject to variations due to its nature. The possible values that a random variable can take (continuous or discrete) have an associated probability. In case the variable is discrete, one can speak about *probability distribution*, whereas the term *probability density function* is used for continuous variables.

Among the continuous probability functions, the so-called normal (or Gaussian) distribution is a very commonly occurring one. The normal distribution is

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.1}$$

The parameter μ is the mean or expected value of the distribution. Given an univariate random variable X defined in a domain $\mathcal{X}$ which follows a probability density function $f(x)$ (this is denoted by $X \sim f(x)$), the expected value is defined as

$$\mathbb{E}[X] = \int_{\mathcal{X}} x f(x) \mathrm{d}x \tag{2.2}$$

The parameter σ is the standard deviation, which is the square root of the variance. Although they are directly related, many statistical properties are easier to handle in terms of the variance rather than the standard deviation. The variance is defined as

$$\mathrm{Var}[X] = \sigma^2 = \int_{\mathcal{X}} (x - \mu)^2 f(x) \mathrm{d}x \tag{2.3}$$

Intuitively, the variance measures how far a set of numbers is spread out around the mean value. The following expression, which can be easily derived, relates the variance and the expected value

$$\mathrm{Var[X]} = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \tag{2.4}$$

There is one more measure, which will be regularly used in the rest of this chapter, which is called covariance. The covariance indicates how much two random variables change together. The covariance between two variables x and y is defined as

$$\mathrm{Cov}(x, y) = \mathbb{E}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])] \tag{2.5}$$

Similarly as it occurred with the variance, it is possible to simplify the expression to obtain

$$\mathrm{Cov}(x, y) = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y] \tag{2.6}$$

2.1.2 Random vectors and covariance matrices

A random vector or more formally, a multivariate random variable, is a collection of random variables. The individual variables are gathered because there may be correlations between them. The use of random vector is convenient in many cases.

The previous definitions will be extended to the case of random vectors. The expected value of a random vector $\mathbf{X}$ is a vector $\mathbb{E}[\mathbf{X}]$ whose elements are the expected values of the respective random variable.

The covariance matrix is a very useful representation, which is defined between two random vectors $\mathbf{X}$ and $\mathbf{Y}$ as

$$\mathrm{Cov}[\mathbf{X}, \boldsymbol{Y}] = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\boldsymbol{Y} - \mathbb{E}[\boldsymbol{Y}])^T] = \mathbb{E}[\mathbf{X}\boldsymbol{Y}^{\boldsymbol{T}}] - \mathbb{E}[\mathbf{X}]\mathbb{E}[\boldsymbol{Y}]^T \tag{2.7}$$

Let us consider a particular case that will commonly happen in the derivation of the Kalman Filter. Let $\mathbf{X}$ be a random vector with $\mathbb{E}[\mathbf{X}] = 0$, which means that every variable contained in the vector has null expectation. The covariance matrix of $\mathbf{X}$, which will be always symmetric, is given by

$$\mathrm{Cov}[\mathbf{X}, \mathbf{X}] = \mathbb{E}[\mathbf{X}\mathbf{X}^T] - \mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{X}]^T = \mathbb{E}[\mathbf{X}\mathbf{X}^T] \tag{2.8}$$

where the last equality follows from the fact that $\mathbb{E}[\mathbf{X}] = 0$. It is interesting to see that the diagonal of $\mathrm{Cov}[\mathbf{X}, \mathbf{X}]$ contains the variance of each element in the vector $\mathbf{X}$, while the off-diagonal terms are the covariances between every combination of them. If a random variable is characterized by a diagonal covariance matrix, that implies that the variables in the vector a independent (not correlated). The notation $\mathbf{X} \sim (\boldsymbol{M}, 0)$ is used, to describe a random vector $\mathbf{X}$ that follows a zero-mean normal distribution with covariance matrix $\boldsymbol{M}$.

2.2 The Kalman Filter

2.2.1 Introduction

The Kalman Filter [23] (and its variants) is one of the most popular data fusion algorithms in the field of information processing. The most famous early use of the Kalman Filter was in

the Apollo navigation computer that took Neil Armstrong to the moon. Today, Kalman filters are at work in every satellite navigation device, smart phone and many computer games.

From a theoretical point of view, the Kalman Filter is an algorithm permitting exact inference in a linear dynamical system. It is usually referred to as the optimal observer for linear systems. The following introduction will help placing the Kalman Filter in the context of linear observers.

Let us consider a generic LTI system

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{2.9}$$

where the output of the system $\boldsymbol{y}$ can be measured with some measurement error. Let us denote by $\hat{\boldsymbol{x}}$ and define $\boldsymbol{z} = \boldsymbol{y} - \boldsymbol{C}\hat{\boldsymbol{x}}$ as the innovation error. This variable shows the discrepancy between the model and the measurements. A traditional approach, the so-called Luenberger observer, consists of using the dynamic model to propagate the state estimation and adding a correction term based on the innovation error as follows

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\hat{\boldsymbol{x}}) \tag{2.10}$$

The matrix $\boldsymbol{L}$ is the observer gain which has to be designed. Defining the estimation error as $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}$, it can be seen that the error dynamics satisfy

$$\dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{e} \tag{2.11}$$

Thus the gain $\boldsymbol{L}$ can be designed using the pole placement method such that the desired error dynamics is achieved. A general rule of thumb suggests that the error dynamics should converge ten times faster than the system dynamics. However, an obvious question arises: is there a way to choose the observer gain in an optimal manner depending on the model reliability and the quality of the measurements?

The Kalman Filter solves this problem for linear systems in which the state and measured variables have a Gaussian distribution.

2.2.2 Kalman Filter Derivation

Let us consider a discrete-time dynamic model in which both the model and the measurements are corrupted by noise as follows

$$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k \mathbf{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \boldsymbol{w}_k \tag{2.12}$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \boldsymbol{v}_k \tag{2.13}$$

where $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ are assumed to be zero-mean Gaussian white-noise errors, which means that they are not correlated forward or backward in time so that

$$\mathbb{E}[\boldsymbol{v}_k \boldsymbol{v}_j^T] = \boldsymbol{R}_k \delta_{kj} \qquad \mathbb{E}[\boldsymbol{w}_k \boldsymbol{w}_j^T] = \boldsymbol{Q}_k \delta_{kj} \tag{2.14}$$

being δ_{jk} the Kronecker delta function. Furthermore, it is also assumed that both errors are uncorrelated so that $\mathbb{E}[\boldsymbol{v}_k \boldsymbol{w}_k^T] = 0$, $\forall k$.

The following estimator form, following the idea of the Luenberger's observer previously described, is considered

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\Phi}_k \hat{\mathbf{x}}^+ + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \tag{2.15}$$
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}^-) \tag{2.16}$$

where a distinction is made between the *a priori* state estimation, $\hat{\mathbf{x}}_k^-$, propagated using the dynamic model, and the *a posteriori* estimation, $\hat{\mathbf{x}}_k^+$, updated upon the arrival of a new measurement.

The following error covariances are defined

$$\boldsymbol{P}_k^- = \mathbb{E}[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-T}], \quad \boldsymbol{P}_{k+1}^- \equiv \mathbb{E}[\boldsymbol{e}_{k+1}^- \boldsymbol{e}_{k+1}^{-T}] \tag{2.17}$$
$$\boldsymbol{P}_k^+ = \mathbb{E}[\boldsymbol{e}_k^+ \boldsymbol{e}_k^{+T}], \quad \boldsymbol{P}_{k+1}^+ \equiv \mathbb{E}[\boldsymbol{e}_{k+1}^+ \boldsymbol{e}_{k+1}^{+T}] \tag{2.18}$$
$$\tag{2.19}$$

where

$$\boldsymbol{e}_k^- \equiv \hat{\mathbf{x}}_k^- - \mathbf{x}_k, \quad \boldsymbol{e}_{k+1}^- \equiv \hat{\mathbf{x}}_{k+1}^- - \mathbf{x}_{k+1} \tag{2.20}$$
$$\boldsymbol{e}_k^+ \equiv \hat{\mathbf{x}}_k^+ - \mathbf{x}_k, \quad \boldsymbol{e}_{k+1}^+ \equiv \hat{\mathbf{x}}_{k+1}^+ - \mathbf{x}_{k+1} \tag{2.21}$$

are the state errors in the prediction and update stages. The goal is to derive recursive expressions for the propagation and update of the error covariance matrices, and also, derive an optimal expression for the gain $\mathbf{K}_k$, which will be obtained by minimizing the error.

The derivation will start finding the expression for $\boldsymbol{P}_{k+1}^-$. Substituting the Equations (2.15) and (2.12) into (2.20) and using the definition of $\boldsymbol{e}_k^+$ in (2.21) yields

$$\boldsymbol{e}_{k+1}^- = \boldsymbol{\Phi}_k \boldsymbol{e}_k^+ - \boldsymbol{w}_k \tag{2.22}$$

$$\begin{aligned} \boldsymbol{P}_{k+1}^- &\equiv \mathbb{E}[\boldsymbol{e}_{k+1}^- \boldsymbol{e}_{k+1}^{-T}] \\ &= \mathbb{E}[\boldsymbol{\Phi}_k \boldsymbol{e}_k^+ \boldsymbol{e}_k^{+T} \boldsymbol{\Phi}_k^T] - \mathbb{E}[\boldsymbol{\Phi}_k \boldsymbol{e}_k^+ \boldsymbol{w}_k^T] - \mathbb{E}[\boldsymbol{w}_k \boldsymbol{e}_k^{+T} \boldsymbol{\Phi}_k^T] + \mathbb{E}[\boldsymbol{w}_k \boldsymbol{w}_k^T] \\ &= \boldsymbol{\Phi}_k \mathbb{E}[\boldsymbol{e}_k^+ \boldsymbol{e}_k^{+T}] \boldsymbol{\Phi}_k^T - \boldsymbol{\Phi}_k \mathbb{E}[\boldsymbol{e}_k^+ \boldsymbol{w}_k^T] - \mathbb{E}[\boldsymbol{w}_k \boldsymbol{e}_k^{+T}] \boldsymbol{\Phi}_k^T + \mathbb{E}[\boldsymbol{w}_k \boldsymbol{w}_k^T] \end{aligned} \tag{2.23}$$

The last equality in Equation (2.23) follows from the fact that $\boldsymbol{w}_k$ and $\boldsymbol{e}_k^+$ are uncorrelated and therefore $\mathbb{E}[\boldsymbol{e}_k^+ \boldsymbol{w}_k^T] = \mathbb{E}[\boldsymbol{w}_k \boldsymbol{e}_k^{+T}] = 0$. Using the definitions in Equations (2.14) and (2.18), Equation (2.23) is reduced to

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{\Phi}_k \boldsymbol{P}_k^+ \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k \tag{2.24}$$

The next step is to find an optimal expression for the updated covariance $\boldsymbol{P}_k^+$. Substituting Equations (2.16) and (2.13) into (2.21) and using the definition of $\boldsymbol{e}_k^+$ in (2.20) yields

$$\boldsymbol{e}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\boldsymbol{e}_k^- + \mathbf{K}_k \boldsymbol{v}_k \tag{2.25}$$

Then $\boldsymbol{P}_k^+$ is given by

$$\begin{aligned} \boldsymbol{P}_k^+ &\equiv \mathbb{E}[\boldsymbol{e}_k^+ \boldsymbol{e}_k^{+T}] \\ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbb{E}[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-T}](\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T \\ &+ (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbb{E}[\boldsymbol{e}_k^- \boldsymbol{v}_k^T]\mathbf{K}_k^T + \mathbf{K}_k \mathbb{E}[\boldsymbol{v}_k \boldsymbol{e}_k^{-T}](\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbb{E}[\boldsymbol{v}_k \boldsymbol{v}_k^T]\mathbf{K}_k^T \end{aligned} \tag{2.26}$$

Again, the last equality in Equation (2.26) follows from the fact that $\boldsymbol{v}_k$ and $\boldsymbol{e}_k^-$ are uncorrelated and therefore $\mathbb{E}[\boldsymbol{e}_k^- \boldsymbol{v}_k^T] = \mathbb{E}[\boldsymbol{v}_k \boldsymbol{e}_k^{-T}] = 0$. Using the definitions in Equations (2.14) and (2.17), Equation (2.23) is reduced to

$$\boldsymbol{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\boldsymbol{P}_k^-(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \boldsymbol{R}_k \mathbf{K}_k^T \tag{2.27}$$

In order to find the optimal value for the gain $\mathbf{K}_k$, the minimization problem is defined

$$\text{minimize} \quad J = \text{Tr}(\boldsymbol{P}_k^+) \tag{2.28}$$

Minimizing the trace of $\boldsymbol{P}_k^+$ is equivalent to minimizing the length of the estimation error vector. Using the following property, which holds for any symmetric matrix $\boldsymbol{M}$

$$\frac{\partial}{\partial \boldsymbol{X}}(\text{Tr}(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T) = 2\boldsymbol{X}\boldsymbol{M}, \tag{2.29}$$

it is possible to see that

$$\frac{\partial J}{\partial \mathbf{K}_k} = -2(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\boldsymbol{P}_k^- \mathbf{H}_k^T + 2\mathbf{K}_k \boldsymbol{R}_k \tag{2.30}$$

Setting Equation (2.30) equal to zero and solving for $\mathbf{K}_k$ yields

$$\mathbf{K}_k = \boldsymbol{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \boldsymbol{P}_k^- \mathbf{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{2.31}$$

Plugging Equation (2.31) into Equation (2.27), results in

$$\boldsymbol{P}_k^+ = (\mathbf{I} - K_k \mathbf{H}_k)\boldsymbol{P}_k^- \tag{2.32}$$

A detailed derivation of the Kalman filter and further considerations can be found in [24]. The Kalman filter equations can be summarized as

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1}^- &= \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \\ \boldsymbol{P}_{k+1}^- &= \boldsymbol{\Phi}_k \boldsymbol{P}_k \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k \end{aligned} \tag{2.33}$$

$$\begin{aligned} \boldsymbol{S}_k &= \boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k \\ \boldsymbol{K}_k &= \boldsymbol{P}_k^- \boldsymbol{H}_h^T \boldsymbol{S}_k^{-1} \\ \hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \mathbf{H}_k \hat{\boldsymbol{x}}_k^- \right) \\ \boldsymbol{P}_k^+ &= (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- \end{aligned} \tag{2.34}$$

2.2.3 Extended Kalman Filter

There is a large number of problems involving nonlinear models. The vast majority of nonlinear models are given in continuous time, as follows

$$\dot{\mathbf{x}}(t) = \boldsymbol{f}(\mathbf{x}(t), \boldsymbol{u}(t), t) + \boldsymbol{w}(t), \quad \boldsymbol{w}(t) \sim \mathcal{N}(0, \boldsymbol{Q}(t)) \tag{2.35}$$

$$\mathbf{y}(t) = \boldsymbol{h}(\mathbf{x}(t)) + \boldsymbol{v}(t), \quad \boldsymbol{v}(t) \sim \mathcal{N}(0, \boldsymbol{R}(t)) \tag{2.36}$$

The main inconvenient of nonlinear models is that the Gaussian properties are not preserved. In other words, a Gaussian input does not imply a Gaussian output. However, if the model is linearized, the nonlinear effects can be small in a certain neighborhood. The Extended Kalman Filter approach consists of linearizing the model at the last estimated state. However, the optimality of the Kalman Filter is lost when applied to nonlinear systems, and there are no guarantees of convergence either.

Let us consider the next discretized version of the model described by Equations (2.35) and (2.36)

$$\mathbf{x}_{k+1} = \boldsymbol{F}(\mathbf{x}_k, \boldsymbol{u}_k, t) + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k) \tag{2.37}$$

$$\mathbf{y}_k = \mathbf{H}(\mathbf{x}_k) + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{R}_k) \tag{2.38}$$

Then the equations of the EKF, which can be found also in [24], are summarized as

$$\boldsymbol{\Phi}_k = \left. \frac{\partial \boldsymbol{F}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-, \boldsymbol{u}_k} \qquad \boldsymbol{\Gamma}_k = \left. \frac{\partial \boldsymbol{F}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-, \boldsymbol{u}_k} \qquad \mathbf{H}_k = \left. \frac{\partial \boldsymbol{H}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-} \tag{2.39}$$

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1}^- &= \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \\ \boldsymbol{P}_{k+1}^- &= \boldsymbol{\Phi}_k \boldsymbol{P}_k \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k \end{aligned} \tag{2.40}$$

$$\begin{aligned} \boldsymbol{S}_k &= \boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k \\ \boldsymbol{K}_k &= \boldsymbol{P}_k^- \boldsymbol{H}_h^T \boldsymbol{S}_k^{-1} \\ \hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-) \right) \\ \boldsymbol{P}_k^+ &= (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- \end{aligned} \tag{2.41}$$

Since the convergence is not guaranteed, one has to be specially careful when initializing an EKF algorithm. The initial guess should be accurate in order avoid the filter to diverge. Although it is very difficult, even for some simple cases, to establish uncertainty bounds on the initial guess that guarantee the convergence, the analysis can be addressed with numerical simulations.

2.3 Attitude Parameterization

2.3.1 Introduction

In this section, different attitude representations are reviewed. Regardless of the method, describing the attitude of an aerial vehicle consists basically on describing the orientation of the body-fixed reference frame (attached to the principal axes of the vehicle) with respect to an external inertial reference frame.

Let us denote hereafter by $\mathcal{E}$ and $\mathcal{B}$ the inertial and body-fixed reference frames, respectively, each consisting of three orthogonal unit vectors $\{\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$ and $\{\boldsymbol{b}_x, \boldsymbol{b}_y, \boldsymbol{b}_z\}$. A superscript notation is used to indicate in which reference frame a vector is defined, e.g., the vector $\boldsymbol{v}^B$ is expressed in the $\mathcal{B}$ frame. It is possible to transform a vector between different reference frames using a rotation matrix. The following notation is used such that ${}^{\mathrm{B}}\boldsymbol{R}_E$ maps vectors of $\mathcal{E}$ onto $\mathcal{B}$, i.e., $\boldsymbol{v}^B = {}^{\mathrm{B}}\boldsymbol{R}_E \boldsymbol{v}^E$.

2.3.2 Direction Cosine Matrix

The Direction Cosine Matrix (DCM) is the most straightforward manner of representing the orientation. Given two coordinate systems, namely $\mathcal{E}$ and $\mathcal{B}$, there exist a matrix $\boldsymbol{R}$ that relates the two coordinate systems. More specifically, the columns of the rotation matrix ${}^{\mathrm{B}}\boldsymbol{R}_E$ are given by the basis vectors of $\mathcal{E}$ expressed in $\mathcal{B}$, i.e., ${}^{\mathrm{B}}\boldsymbol{R}_E = [\boldsymbol{e}_x^B, \boldsymbol{e}_y^B, \boldsymbol{e}_z^B]$.

This way of representing orientation is highly redundant. The columns of the DCM have several constraints. On one hand, they must be unitary, and, on the other hand, each column is orthogonal to the others. There are six constraints in total, which means that it is possible to obtain a minimal representation of the orientation using only three parameters.

2.3.3 Euler angles

Euler angles are a minimal representation of the orientation given only by three parameters. Since the order of rotation matters, there are up to twelve possible sequences of rotations. The common convention for aerial vehicles is the roll(ϕ) - pitch(θ) - yaw(ψ) sequence. This is an intrinsic sequence, which means that each rotation is defined over the successive rotated

reference frames.

It is easy to derive the rotation matrices for individual rotations around a single axis (roll, pitch or yaw). Due to the properties of the rotation matrices, the final orientation of the body is given by the multiplication of those individual rotations. Using the roll-pitch-yaw convention results in the following orientation matrix

$$ {}^{\mathrm{B}}\boldsymbol{R}_{E} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - c\phi c\psi & c\phi s\theta \end{bmatrix} \tag{2.42} $$

Let $\boldsymbol{\omega} = \boldsymbol{\omega}^{B}_{B/E} = [p, q, r]^T$ be the angular velocity of the aircraft with respect to $\{E\}$ expressed in the body frame $\{B\}$. The rotational kinematic relating these angular velocities to the Euler angles, $\boldsymbol{\eta} = [\phi\, \theta\, \psi]^T$, is expressed as

$$ \dot{\boldsymbol{\eta}} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \boldsymbol{\omega} \tag{2.43} $$

where ϕ, θ, and ψ denote the roll, pitch and yaw angle, respectively.

2.3.4 Quaternions

Quaternions are a convenient mathematical notation for representing orientations and rotations in three-dimensional spaces. Compared to Euler angles, they do not present singularities and are more efficient to compose, which means less computational effort in embedded applications.

According to Euler's rotation theorem, any sequence of rotations of a coordinate system is equivalent to a single rotation by a given angle θ about a fixed axis $\boldsymbol{u}$. Quaternions provide an efficient way to encode this information. They are defined as

$$ \boldsymbol{q} = \begin{bmatrix} q_0 \\ \tilde{\boldsymbol{q}} \end{bmatrix} \tag{2.44} $$

where $q_0 = \cos\frac{\theta}{2}$ and $\tilde{\boldsymbol{q}} = \boldsymbol{u}\sin\frac{\theta}{2}$ are the scalar and vector parts of the quaternion, respectively. When used to describe rotations, the quaternions are constrained to the surface of a unitary hyper-sphere which means that they always satisfy $\tilde{\boldsymbol{q}}^T\tilde{\boldsymbol{q}} + q_0^2 = 1$. This fact is used in real applications to avoid the propagation of numerical errors.

The rotation matrix is expressed in terms of the quaternion components as follows

$$ {}^{\mathrm{B}}\boldsymbol{R}_{E} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 - q_1q_4) & q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{2.45} $$

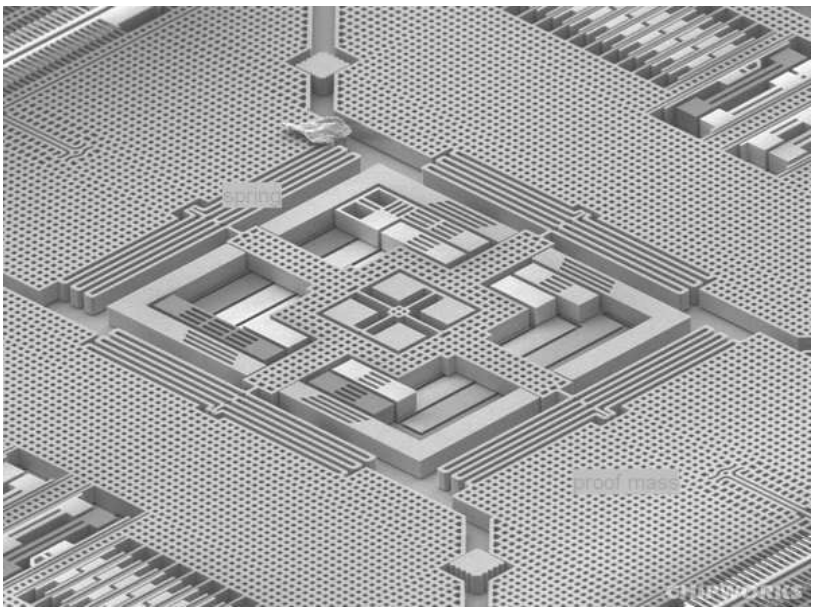

Figure 2.1: Detailed view of the iPhone 4s 3-axis gyroscope

2.4 Inertial MEMS Sensors

2.4.1 Introduction

Micro-Electro-Mechanical Systems (MEMS) is the integration of mechanical elements, sensors, actuators, and electronics on a common silicon substrate through microfabrication technology. The micromechanical components are fabricated by selectively etching away parts of the silicon wafer or adding new structural layers. The exploitation of the silicon as a mechanical material has been exceptionally successful in the production of 3D structures. A detailed view of the structure of a 3-axis gyroscope is shown in Figure 2.1.

2.4.2 MEMS Characterization

There are several technological companies like Crossbow, Analog Devices, InvenSense that focus on developing MEMS inertial sensors. When finding the most suitable device for an specific application, these are some of the important metrics to look at:

Scale Factors

These parameters are responsible for the deterministic behaviour of the sensor and they can be calibrated. A default calibration is provided by the manufacturer, which sometimes includes temperature compensation. This is commonly achieved by adjusting each coefficient to a low-order polynomial.

The inertial sensors are usually mounted in a tri-axial distribution on a plate. Any misalignment can give rise to *cross-axis coupling* effects. *Nonlinearities* also appear due to the nature of the components used to fabricate the sensors. These two effects are non-desirable and are usually upper-bounded in the manufacturer specifications.

Bias

The bias is the term referring to a nearly-constant drift in the measurement. Although this error can be accurately calibrated, the bias slowly drifts over time due to electro-mechanical factors. The *bias stability* is an important metric of the quality of an inertial sensor, usually expressed in deg/s/h. Similarly, the *bias repeatability* refers the variability of the bias every time the sensor is turned on.

Having a mathematical representation of the bias is useful when designing filters. The slow drift of the bias is described by a random walk process, which is the result of integrating a white noise signal. If the bias is denoted by β, then it evolves over time according to

$$\dot{\beta} = w, \qquad w \sim \mathcal{N}(0, \sigma^2) \tag{2.46}$$

where σ^2 is the variance of the underlying white noise and determines how fast the bias drifts.

Random-Walk Error

This is a measure of the noise of the sensor, and it is specially important when defining the quality of a gyroscope, where it is given in units of deg/$s/\sqrt{\text{Hz}}$ (*noise density*). The amount of noise will determine the drift of the angular estimation obtained by integrating the angular velocity. Because of that, some manufacturers provide an equivalent measure, the *angular rate walk (ARW)* in units of deg/$\sqrt{\text{h}}$, from which it is straightforward to obtain an approximation of the drift in the angular estimation over a certain period of time.

2.4.3 Gyroscopes

The MEMS gyroscopes are placed in a strapdown configuration, which means they are attached to the vehicle. This differs from traditional gyroscopes used in aviation which were mounted on gimbals. In a strapdown configuration, a 3-axis gyroscope measures the angular velocity of the body. Using the notation introduced in Section 2.3, the output of the gyroscope is the angular velocity of $\{\mathcal{B}\}$ with respect to $\{\mathcal{E}\}$, expressed in $\{\mathcal{B}\}$, which is denoted by $\boldsymbol{\omega} = \boldsymbol{\omega}^B_{B/E} = [p, q, r]^T$.

According to the characteristics previously described about the MEMS sensors, a suitable model for the gyroscope measurement is given by

$$\bar{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\beta}_\omega + \boldsymbol{\eta}_\omega, \qquad \mathbb{E}[\boldsymbol{\eta}_\omega] = \mathbf{0} \qquad \mathbb{E}[\boldsymbol{\eta}_\omega \boldsymbol{\eta}_\omega^T] = \sigma_\omega \mathbf{I}_3 \tag{2.47}$$

$$\dot{\boldsymbol{\beta}}_\omega = \boldsymbol{\eta}_\beta, \qquad \mathbb{E}[\boldsymbol{\eta}_\beta] = \mathbf{0} \qquad \mathbb{E}[\boldsymbol{\eta}_\beta \boldsymbol{\eta}_\beta^T] = \sigma_\beta \mathbf{I}_3 \tag{2.48}$$

where the angular velocity measurement $\bar{\boldsymbol{\omega}}$ is composed of its actual value $\boldsymbol{\omega}$, plus the bias $\boldsymbol{\beta}_\omega$ and noise in the measurement $\boldsymbol{\eta}_\omega$. Furthermore, the bias is modeled as a random walk process.

The measurements of the gyroscopes are directly related to the kinematics of the vehicle. Given a known initial attitude it is possible to use the information of the gyroscopes along with any of the equations XXXXXX to propagate over time. However, this will lead to a drift in the angular estimation that will depend on the noise characteristics of the sensors.

2.4.4 Accelerometers

The MEMS accelerometers in strapdown configuration measure the specific force action on the vehicle expressed in $\{\mathcal{B}\}$. According to the characteristics previously described about the MEMS sensors, a suitable model for the accelerometer measurements is given by

$$\bar{\boldsymbol{a}} = \boldsymbol{a} + \boldsymbol{\eta}_a, \qquad \mathbb{E}[\boldsymbol{\eta}_a] = \mathbf{0} \qquad \mathbb{E}[\boldsymbol{\eta}_a \boldsymbol{\eta}_a^T] = \sigma_a \mathbf{I}_3 \tag{2.49}$$

where the acceleration measurement $\bar{\boldsymbol{a}}$ is composed of its actual value $\boldsymbol{a}$ and noise in the measurement $\boldsymbol{\eta}_a$. There is also a bias in the accelerometers, but they are not critical as it will be explained next.

In contrast to the gyroscopes, the relation between the accelerometer readings and the attitude of the vehicle is not so straightforward. From XXXX, it can be seen that the specific force

$$\boldsymbol{a}^B = \frac{1}{m}\left(\boldsymbol{f}^B - {}^{\mathrm{B}}\boldsymbol{R}_E (mg)\hat{\boldsymbol{e}}_3\right) = \dot{\boldsymbol{v}}^B - {}^{\mathrm{B}}\boldsymbol{R}_E g \hat{\boldsymbol{e}}_3 \tag{2.50}$$

where $\dot{\boldsymbol{v}}^B$ is the acceleration vector due to the external forces expressed in $\{B\}$, m denotes the mass of the aerial vehicle and $\boldsymbol{f}^B$ represents the vector of external forces that act on the quadrotor. Since the accelerations in stable flight regimes are usually small compared to the gravity acceleration, neglecting the linear acceleration ($\dot{\boldsymbol{v}}^B = 0$) is a classical assumption [25]. Normalizing the vector of acceleration measurements facilitates to express the roll and pitch angles as

$$\boldsymbol{a} = \frac{\boldsymbol{a}^B}{|\boldsymbol{a}^B|} \approx -{}^{\mathrm{B}}\boldsymbol{R}_E \hat{\boldsymbol{e}}_3 = \begin{bmatrix} \sin\theta \\ -\sin\phi\cos\theta \\ -\cos\phi\cos\theta \end{bmatrix} \tag{2.51}$$

From Equation (2.51), it can be seen that any bias in the accelerometer measurements will result only in a small offset in the roll and pitch estimations with respect to the real horizontal

plane perpendicular to the gravity vector. However, these offsets will also appear due to any misalignment of the inertial sensors mounted on the vehicle. These errors can be calibrated once the whole system is built.

2.5 Experimental Platforms

Along this thesis, the following experimental platforms will be constantly referenced.

2.5.1 Quanser Quadrotor Laboratory

Most of the algorithms contained in this thesis have been first validated using the platform shown in Figure 2.2. It is thought of as a test bed platform of control algorithms for vertical lift off vehicles, so that the translational degrees of freedom are clamped for convenience.

The orientation of the vehicle is measured by means of optical encoders with an accuracy of 0.04 deg. These encoders provide a reliable pattern for the evaluation and comparison of the algorithms. In addition, a commercial IMU (3DM-GX2) and a low cost inertial sensor (MPU6050) were also included in the platform in order to validate and compare the measurements. The 3DM-GX2 runs at 200 Hz and outputs directly the orientation in Euler angles representation. The MPU6050 is composed of a 3-axis gyroscope and a 3-axis accelerometer. It does not provide the angles of the rigid body but only the raw measurements of the sensors.

Figure 2.2: Experimental platform

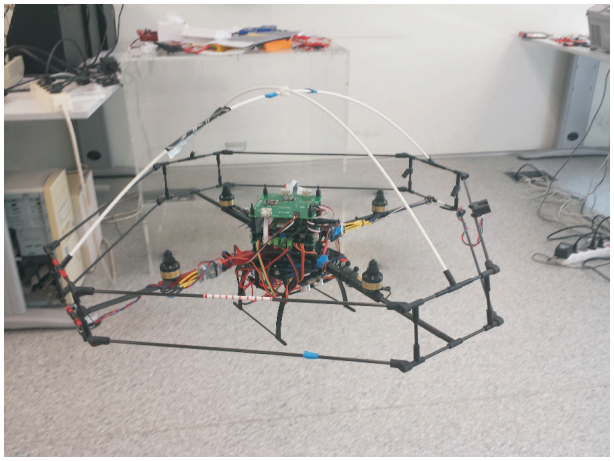

Figure 2.3: Quadrotor prototype

2.5.2 Quadrotor Prototype

Although the platform described above is a suitable scenario for numerical comparison, there are two handicaps to overcome in real flight, i.e., vibrations and linear accelerations. The proposed algorithm has been also validated in-flight using a small quadrotor specially built for this task, see Figure 2.3. It has a distance of 41 cm between rotors, it weights around 1.3 kg without battery, and it is outfitted with an IMU MicroStrain 3DM-GX2 and with the MPU6050, among other sensors.

The basic hardware consists of a MikroKopter frame, YGE 25i electronic speed controllers, RobbeRoxxy 2827-35 brushless motors and 10x4.5 plastic propellers. All the computations are made onboard using an Arduino Due which is based on an Atmel SAM3X8E ARM Cortex-M3 microcontroller running at 84 MHz, and an Igep v2 board running Xenomai real-time operating system at 1 GHz.

The Arduino Due is in charge of reading every sensor, running the Kalman filter algorithm for attitude estimation and the attitude control algorithm, controlling the motor's speed, and sending the data to the Igep board. The control algorithm consists of a PD controller with nested saturations. So far the Igep board is only used as Wifi bridge.

Further details about this platform can be found in [26].

Chapter 3

Attitude estimation

3.1 A Comparison of Attitude Estimation Algorithms

3.1.1 Introduction

This chapter aims at providing a comparative evaluation of attitude estimation algorithms using low-cost sensors (gyroscopes and accelerometers). Low-cost IMUs are considered those devices with a price less than 100 USD. These are very cheap indeed, and they are commonly referred to as hobbyist-level IMUs. These devices have lower performance in terms of bias stability, nonlinearities and signal-to-noise ratio than those on the market for industrial applications. A comparison of a wide range of IMUs can be found in [27].

Several aspects must be taken into account while choosing the most suitable approach for a given application: singularity existence, convergence guarantee, computational time, bias estimation, etc. The evaluation will be focused on Kalman filtering methods, as they provide a suitable framework for an easy integration in higher level localization techniques based on laser range finders, cameras or GPS [4].

The major contributions of this chapter are to provide a comparative evaluation of different algorithms in the literature and to propose a slightly modified algorithm in order to improve the angular velocity estimation. The results show that it is possible to obtain a performance with a hobbyist-grade IMU similar to that of an industrial-grade IMU.

3.1.2 Estimation algorithms

The algorithms considered in the comparison will be presented here. The discrete Kalman filtering approach is used in all of them, disregarding the formulation of the system. If the resulting system is non-linear then the Extended Kalman Filter (EKF) is applied.

Recall that the Kalman filter is derived for a system described by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w} \qquad & \mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T] &= \boldsymbol{Q} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v} \qquad & \mathbb{E}[\boldsymbol{v}\boldsymbol{v}^T] &= \boldsymbol{R} \end{aligned} \tag{3.1}$$

where $\boldsymbol{w}$ and $\boldsymbol{v}$ are the process and the measurement noises, respectively, which are assumed to be Gaussian, with covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. Observe that, it is thus necessary to estimate and correct the biases of the sensors as they are nearly constant errors which are not removed in the Kalman filtering process. Moreover, it is convenient to identify the stochastic model of the sensors for achieving a good performance.

In a first group of formulations, the elements of the DCM are manipulated directly, leading to simpler formulations. We refer to them as DCM-based. Within this group, different variants are distinguished depending on which variables are estimated. The last of this variants is a new one and it has been proposed in this thesis with the intention of obtaining a smoother estimation of the angular velocity. A second group is based on the application of the EKF for non-linear systems using both Euler angles and quaternions.

DCM-based variant #1 [DCM #1]

Algorithms based on DCM directly update the components of the matrix, avoiding the representation by means of Euler angles. Although the whole matrix could be updated, the last column (denoted by $\boldsymbol{r}_3$) suffices at providing information about ϕ and θ, as it can be seen in (2.42). The following formulation can be found in [28]. Using the common notation in control theory, the vector of estimated states is denoted by $\boldsymbol{x} = \boldsymbol{r}_3$, and the system is described by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\xi} \qquad & \mathbb{E}[\boldsymbol{\xi}\boldsymbol{\xi}^T] &= \boldsymbol{Q} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v} \qquad & \mathbb{E}[\boldsymbol{v}\boldsymbol{v}^T] &= \boldsymbol{R} \end{aligned} \tag{3.2}$$

where $\boldsymbol{A} = [\tilde{\boldsymbol{\omega}}]_\times; \quad \boldsymbol{C} = -\boldsymbol{I}_3; \quad \boldsymbol{Q} = q_p\boldsymbol{I}_3; \quad \boldsymbol{R} = \boldsymbol{\Sigma}_a$

If one assumes that $\boldsymbol{\Sigma}_a$ can be determined characterizing the sensors, then there is only one tuning parameter q_p, which determines the relative importance of the state propagation with respect to the information recovered form the accelerometers.

Notice that in this case the resultant system is linear and observable. Thus the convergence of the Kalman filter is guaranteed. However, this methodology would imply the need of zero-velocity updates in practice, as the bias of the gyroscopes is not explicitly estimated.

DCM-based variant #2 [DCM #2]

The previous filter can be modified to include the bias estimation [29]. According to the measurement model in (2.47), the bias-free angular velocity is given by $\tilde{\boldsymbol{\omega}} = \bar{\boldsymbol{\omega}} - \boldsymbol{\beta}_\omega$. Denoting

by $\boldsymbol{x} = [\boldsymbol{r}_3^T,\, \boldsymbol{\beta}_\omega^T]^T$ the vector of estimated variables, the resulting system can be expressed as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{\xi} & \mathbb{E}[\boldsymbol{\xi}\boldsymbol{\xi}^T] &= \boldsymbol{Q} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v} & \mathbb{E}[\boldsymbol{v}\boldsymbol{v}^T] &= \boldsymbol{R} \end{aligned} \tag{3.3}$$

where

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} [\bar{\boldsymbol{\omega}} - \boldsymbol{\beta}_\omega]_\times & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} -\boldsymbol{I}_3 & \boldsymbol{0} \end{bmatrix}$$
$$\boldsymbol{Q} = \begin{bmatrix} q_p \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & q_\beta \boldsymbol{I}_3 \end{bmatrix} \qquad \boldsymbol{R} = \boldsymbol{\Sigma}_a$$

There is an extra parameter q_β accounting for the bias noise. It has a physical meaning as it describes the random walk model for the bias. Although it is used as a tuning parameter, it has to be small enough to allow only slow variations of the bias, as it is known it happens in reality.

New DCM-based #3 [DCM #3]

In order to improve the angular velocity estimation, the above algorithm can be extended by including the angular velocity in the vector of estimated variables, $\boldsymbol{x} = [\boldsymbol{r}_3^T,\, \boldsymbol{\beta}_\omega^T,\, \boldsymbol{\omega}^T]^T$. The process model for the angular velocities is assumed to be constant, so they are only modified in the updating phase of the Kalman filter. In practice, this will work as a low-pass filter, controlled by the parameter q_ω. In this case, the matrices describing the system are

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & [\boldsymbol{r}_3]_\times \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} -\boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_3 & \boldsymbol{I}_3 \end{bmatrix}$$
$$\boldsymbol{Q} = \begin{bmatrix} q_p \boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & q_\beta \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & q_\omega \boldsymbol{I}_3 \end{bmatrix} \qquad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{\Sigma}_a & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_\omega \end{bmatrix}$$

Euler EKF [EKF]

In this algorithm, six variables are estimated: three Euler angles and three biases of the gyroscopes. The vector of estimated states can be written as $\boldsymbol{x} = [\boldsymbol{\eta}^T\, \boldsymbol{\beta}^T]^T$. The non-linear equations are described in a state-space form as,

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{f}(x) + \boldsymbol{\xi} & \mathbb{E}[\boldsymbol{\xi}\boldsymbol{\xi}^T] &= \boldsymbol{Q} \\ \boldsymbol{y} &= \boldsymbol{h}(x) + \boldsymbol{v} & \mathbb{E}[\boldsymbol{v}\boldsymbol{v}^T] &= \boldsymbol{R} \end{aligned}$$

where

$$\boldsymbol{Q} = \begin{bmatrix} q_\eta \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & q_\beta \boldsymbol{I}_3 \end{bmatrix} \qquad \boldsymbol{R} = \boldsymbol{\Sigma}_a$$

The process equation is derived from (2.43) by incorporating directly the measurements from the gyroscopes as follows

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} (\bar{\omega}_x - \beta_{\omega_x}) + (\bar{\omega}_y - \beta_{\omega_y}) \sin\phi + (\bar{\omega}_z - \beta_{\omega_z}) \tan\theta \\ (\bar{\omega}_y - \beta_{\omega_y}) \cos\phi - (\bar{\omega}_z - \beta_{\omega_z}) - \sin\phi \\ (\bar{\omega}_y - \beta_{\omega_y}) \frac{\sin\phi}{\cos\theta} + (\bar{\omega}_z - \beta_{\omega_z}) \frac{\cos\phi}{\cos\theta} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

while the measurement model is given by (2.51)

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \sin\theta \\ -\sin\phi\cos\theta \\ -\cos\phi\cos\theta \end{bmatrix} \tag{3.4}$$

In this case, the parameter q_η determines how reliable the state propagation (based on gyroscopes measurements) is, with respect to the observations made by the accelerometers.

Quaternions

In this algorithm, six variables are estimated: four quaternions and two biases of the gyroscopes. The vector of estimated states can be written as $\boldsymbol{x} = [\boldsymbol{q}^T, \boldsymbol{\beta}^T]^T$. The non-linear equations are described in a state-space form as,

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{\xi} & \mathbb{E}[\boldsymbol{\xi}\boldsymbol{\xi}^T] &= \boldsymbol{Q} \\ \dot{\boldsymbol{y}} &= \boldsymbol{h}(x) + \boldsymbol{v} & \mathbb{E}[\boldsymbol{v}\boldsymbol{v}^T] &= \boldsymbol{R} \end{aligned} \tag{3.5}$$

where

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\Omega}(\bar{\boldsymbol{\omega}}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} -\boldsymbol{I}_3 & \boldsymbol{0} \end{bmatrix}$$
$$\boldsymbol{Q} = \begin{bmatrix} q_q \boldsymbol{I}_4 & \boldsymbol{0} \\ \boldsymbol{0} & q_\beta \boldsymbol{I}_3 \end{bmatrix} \qquad \boldsymbol{R} = \boldsymbol{\Sigma}_a$$

The measurement model is given by the third column of (2.45),

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} 2(q_1 q_3 + q_2 q_4) \\ 2(q_2 q_3 - q_1 q_4) \\ q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{3.6}$$

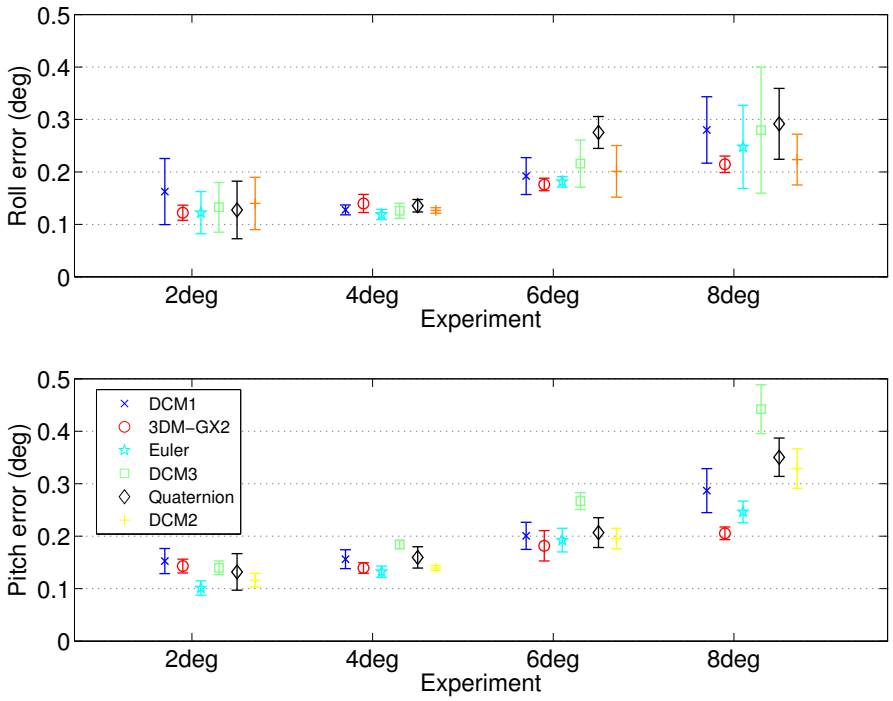

Figure 3.1: Root mean square errors and standard deviations.

3.1.3 Conlusions

The results show that it is possible to obtain a high-performance attitude estimation using hobbyist-level sensors, in an almost-true non-accelerated experimental platform. A new proposed algorithm provides very fast and smooth velocity estimation, which may be beneficial for the attitude control of UAS.

3.2 A High-Performance Computationally-Efficient Attitude Estimation Algorithm

3.2.1 Proposed Algorithm

The kinematics of an aerial vehicle and the measurement model can be expressed by (2.43) and (2.51), respectively. An advantage of the Euler formulation is that the yaw angle can be removed from the equations. Let us denote the state vector of estimated variables by

$\boldsymbol{x} = [\phi,\, \beta_x,\, \theta,\, \beta_y]$. Thus, the filter can be written as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} (\bar{\omega}_x - \beta_x) + (\bar{\omega}_y - \beta_y)\sin\phi\tan\theta \\ 0 \\ (\bar{\omega}_y - \beta_y)\cos\phi \\ 0 \end{bmatrix} + \boldsymbol{w} \\ \boldsymbol{y} &= \begin{bmatrix} \sin\theta \\ -\sin\phi\cos\theta \\ -\cos\phi\cos\theta \end{bmatrix} + \boldsymbol{v} \end{aligned} \tag{3.7}$$

Moreover, it is reasonable to simplify the equations by assuming small angles approximations, $\sin\alpha \approx \alpha$ and $\cos\alpha \approx 1$, with $\alpha = \{\phi,\, \theta\}$, which leads to the following linear equations

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{u} + \boldsymbol{w} \\ \boldsymbol{y} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{v} + \boldsymbol{v} \end{aligned} \tag{3.8}$$

where the input vector consists of the angular velocity measurements $\boldsymbol{u} = [\bar{\omega}_x,\, \bar{\omega}_y]$, and $\boldsymbol{y} = [\bar{a}_x,\, \bar{a}_y]$ contains the acceleration measurements. The third equation of the measurement model which involves the third accelerometer axis has been removed, as it has very low sensitivity with respect to the roll-pitch orientation for small angles.

These simplifications result in a smaller-size linear system thus reducing the computational load substantially. Furthermore, it will be shown next how the structure of the matrices can be also exploited to reduce the Kalman filter to a set of simple equations. The continuous-time system (3.8) can be discretized with sample time T, assuming zero-order hold of the input, as follows

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{w}_k & \mathbb{E}[\boldsymbol{w}_k\boldsymbol{w}_k^T] &= \boldsymbol{Q}_k \\ \boldsymbol{y}_k &= \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k & \mathbb{E}[\boldsymbol{v}_k\boldsymbol{v}_k^T] &= \boldsymbol{R}_k \end{aligned} \tag{3.9}$$

where

$$\boldsymbol{A}_k = e^{\boldsymbol{A}T} = \begin{bmatrix} 1 & -T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{B}_k = \int_0^T e^{\boldsymbol{A}\tau}\mathrm{d}\tau = \begin{bmatrix} T & 0 \\ 0 & 0 \\ 0 & T \\ 0 & 0 \end{bmatrix}$$

$$\boldsymbol{H}_k = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}$$

Notice that, as a consequence of the simplifications, the resulting system is decoupled. Therefore, one can implement two different Kalman filters separately for roll and pitch. Also, one can take advantage of the fact that the matrix $\boldsymbol{P}$ is symmetric, so that only three of its entries need to be stored in memory. Using (2.33) and (2.34), the following equations for the Kalman filter of the roll angle can be derived

$$\begin{aligned}
\phi_k^- &= \phi_{k-1} + T(\bar{\omega}_x - \beta_{x_k}) \\
\beta_{x_k}^- &= \beta_{x_{k-1}} \\
p_{11_k}^- &= p_{11_{k-1}} - 2Tp_{12_{k-1}} + T^2 p_{22_{k-1}} + q_{11_k} \\
p_{12_k}^- &= p_{12_{k-1}} - Tp_{22_{k-1}} \\
p_{22_k}^- &= p_{22_{k-1}} + q_{22_k}
\end{aligned} \tag{3.10}$$

$$\begin{aligned}
\phi_k^+ &= (1-\alpha_\phi)\phi_k^- - \alpha_\phi \bar{a}_y \\
\beta_{x_k}^+ &= \beta_{x_k}^- - \gamma_\phi(\bar{a}_y + \phi_k^-) \\
p_{11_k}^+ &= (1-\alpha_\phi)p_{11_k}^- \\
p_{12_k}^+ &= (1-\alpha_\phi)p_{12_k}^- \\
p_{22_k}^+ &= -\gamma_\phi p_{12_k}^- + p_{22_k}^-
\end{aligned} \tag{3.11}$$

where

$$\alpha_\phi = \frac{p_{11_k}^+}{p_{11_k}^+ + r_{11}} \qquad \gamma_\phi = \frac{p_{12_k}^+}{p_{11_k}^+ + r_{11_k}} \tag{3.12}$$

In a very similar way, the derivation of the Kalman filter equations for the pitch angle leads to the following set of equations

$$\begin{aligned}
\theta_k^- &= \theta_{k-1} + T(\bar{\omega}_x - \beta_{x_k}) \\
\beta_{y_k}^- &= \beta_{y_{k-1}} \\
p_{11_k}^- &= p_{11_{k-1}} - 2Tp_{12_{k-1}} + T^2 p_{22_{k-1}} + q_{33_k} \\
p_{12_k}^- &= p_{12_{k-1}} - Tp_{22_{k-1}} \\
p_{22_k}^- &= p_{22_{k-1}} + q_{44_k}
\end{aligned} \tag{3.13}$$

$$\begin{aligned}
\theta_k^+ &= (1-\alpha_\phi)\theta_k^- + \alpha_\phi \bar{a}_y \\
\beta_{y_k}^+ &= \beta_{y_k}^- + \gamma_\phi(\bar{a}_x - \theta_k^-) \\
p_{11_k}^+ &= (1-\alpha_\theta)p_{11_k}^- \\
p_{12_k}^+ &= (1-\alpha_\theta)p_{12_k}^- \\
p_{22_k}^+ &= -\gamma_\theta p_{12_k}^- + p_{22_k}^-
\end{aligned} \tag{3.14}$$

where

$$\alpha_\theta = \frac{p_{11_k}^+}{p_{11_k}^+ + r_{22_k}} \qquad \gamma_\theta = \frac{p_{12_k}^+}{p_{11_k}^+ + r_{22_k}} \tag{3.15}$$

3.2.2 Numerical Validation

The Kalman filter algorithm proposed for attitude estimation has been first validated using the platform shown in Figure 2.2. It is thought of as a test bed platform of control algorithms for vertical lift off vehicles, so that the translational degrees of freedom are clamped for convenience.

The orientation of the vehicle is measured by means of optical encoders with an accuracy of 0.04 deg. These encoders provide a reliable pattern for the evaluation and comparison of the algorithms. In addition, a commercial IMU (3DM-GX2) and a low cost inertial sensor (MPU6050) were also included in the platform in order to validate and compare the measurements. The 3DM-GX2 runs at 200 Hz and outputs directly the orientation in Euler angles representation. The MPU6050 is composed of a 3-axis gyroscope and a 3-axis accelerometer. It does not provide the angles of the rigid body but only the raw measurements of the sensors. The characteristics of the both devices are given in table 3.1.

Table 3.1: IMUs specifications

	Microstrain 3DM-GX2	MPU-6050
Size	$63 \times 41 \times 32$	$21 \times 17 \times 2$
Weight	50 g	6 g
Gyro range	± 75 to ± 1200 deg/s	± 250 to ± 2000 deg/s
Gyro bias	± 0.2 deg/s	± 20 deg/s
Gyro nonlinearity	0.2 %	0.2%
Gyro noise performance	0.17 deg/s (rms)	0.025 deg/s/ (rms)
Accel. bias	± 5 mg	± 50 mg
Accel. nonlinearity	0.2 %	0.5 %
Accel. noise performance	0.6 mg (rms) [1]	1.3 mg (rms)

A PD controller was used to stabilize the system to constant references of pitch and roll. The system was then perturbed applying disturbances by hand. All data was collected at 333 Hz and the algorithm described in Section 3.2.1 was computed offline using Matlab. A trial and error tuning process resulted in the following covariance matrices

$$\boldsymbol{Q}_k = \begin{bmatrix} 0.94 \cdot 10^{-6} & 0 & 0 & 0 \\ 0 & 0.91 \cdot 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{R}_k = \begin{bmatrix} 0.37 & 0 \\ 0 & 0.39 \end{bmatrix}$$

The estimation obtained by means of this procedure can be seen in Figure 3.2. At first sight, it can be noticed that the proposed algorithm performs fairly well, thus validating the simplifications made on its derivation. As it is difficult to visually evaluate the quality

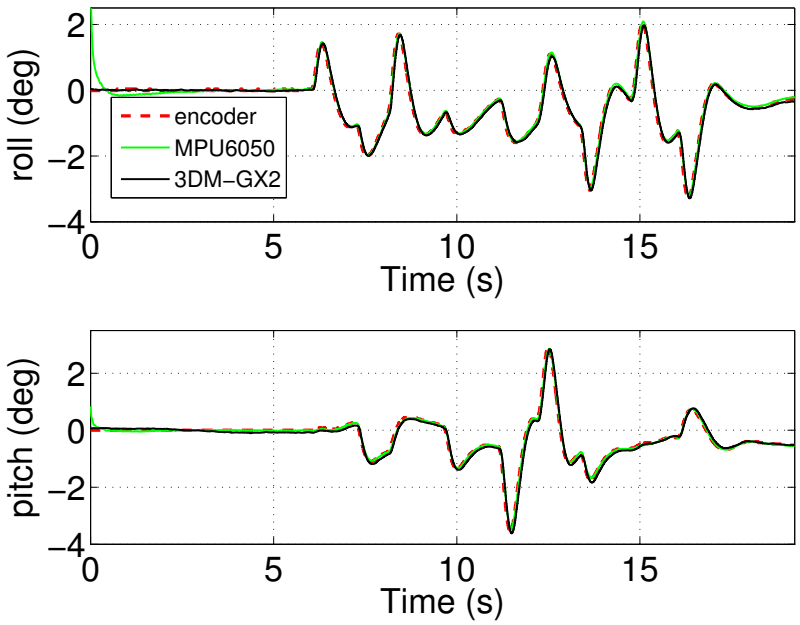

Figure 3.2: Attitude estimation

of both estimations, some performance indexes were chosen, i.e., the root mean squared error, the maximum absolute error and the delay, all of them computed with respect to the estimation given by the encoders. Table 3.2 gathers the information of these indexes for an experiment of several minutes. One can see how the proposed algorithm performs even better than the 3DM-GX2.

Table 3.2: Performance indexes

		rmse	$\|\text{error}\|_{\max}$	delay
3DM-GX2	roll	0.3 deg	1.56 deg	25 ms
	pitch	0.27 deg	1.46 deg	
MPU 6050	roll	0.14 deg	0.72 deg	15 ms
	pitch	0.19 deg	0.91 deg	

Figure 3.3 shows the evolution of the bias estimation. The real bias of the gyroscopes was computed averaging the first few seconds during which the system remains steady. It is possible to see how the estimated bias converges to the real value within a few seconds.

The accelerations measured by the MPU6050 are depicted in Figure 3.4 along with the ideal measurements, which were built by computing (2.51) using the angular measurements

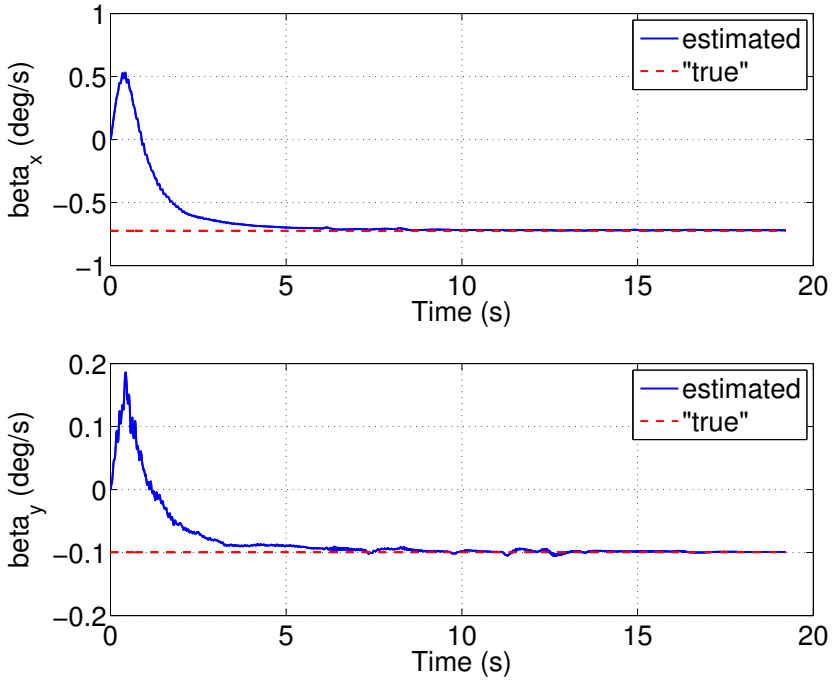

Figure 3.3: Bias estimation

from the encoders and adding Gaussian noise. It is pointed out that the accelerometers are easily affected by the vibrations of the motors.

3.2.3 Flight tests

The quadrotor was controlled in roll and pitch angles using the estimated values, $\hat{\phi}, \hat{\theta}$ and $\dot{\hat{\phi}}, \dot{\hat{\theta}}$, computed using the proposed algorithm. The yaw angle was stabilized with the measurement of the Microstrain sensor.

In the lack of a motion capture system, the in-flight attitude estimation of the proposed algorithm is compared to the 3DM-GX2. Figure 3.5 shows the attitude estimation during one minute of flight. One can see that both estimations are very similar. Although the 3DM-GX2 is not a fully reliable pattern, it can be seen that the proposed algorithm provides a fast, noise-free and drift-free estimation. Furthermore, it must be also noticed that the control is computed with the attitude estimation of the proposed algorithm. The small oscillations around the equilibrium point evidence that the attitude and velocity estimations lead to a very good control performance.

The angular velocities are shown in Figure 3.6. The estimated angular velocities consist

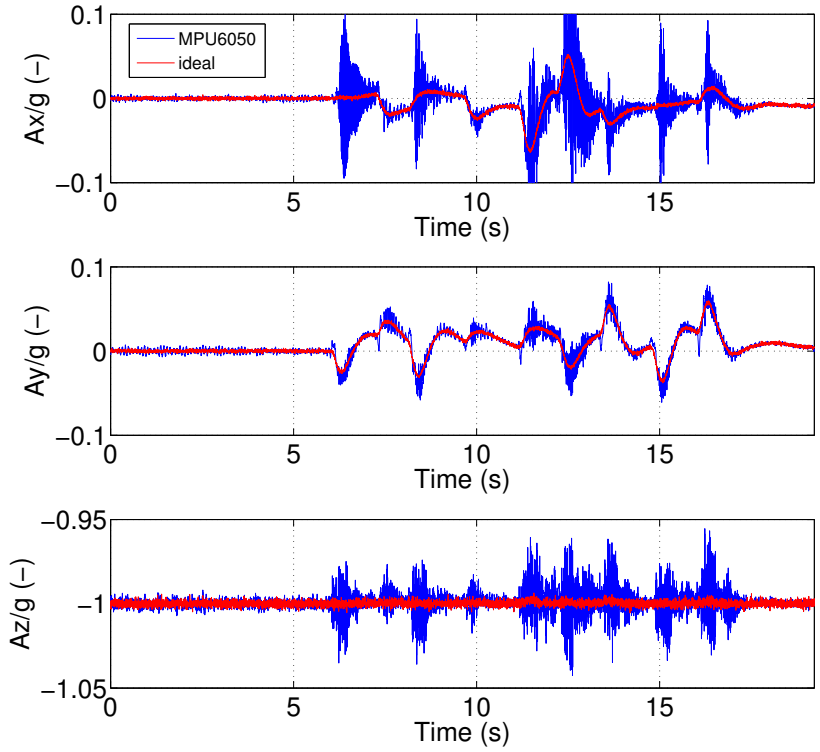

Figure 3.4: Accelerometer responses

of the raw gyroscope measurements corrected with the estimated biases. The estimation of the bias avoids the need of correcting the offset of the gyroscopes prior to each flight and allows operation over long periods of time.

3.2.4 Conclusions

A simplified algorithm for attitude estimation based on the Kalman filter has been proposed and validated in-flight. The simplifications in both the dynamic and measurement models result in a very computationally-efficient algorithm.

Despite the simplifications, a comparison carried out in an experimental platform with a reliable pattern provided by optical encoders showed that the proposed algorithm exhibits even better performance than a commercial IMU, the Microstrain 3DM-GX2. Further validation in-flight showed that the proposed algorithm performs also very well under strong vibrations and linear accelerations.

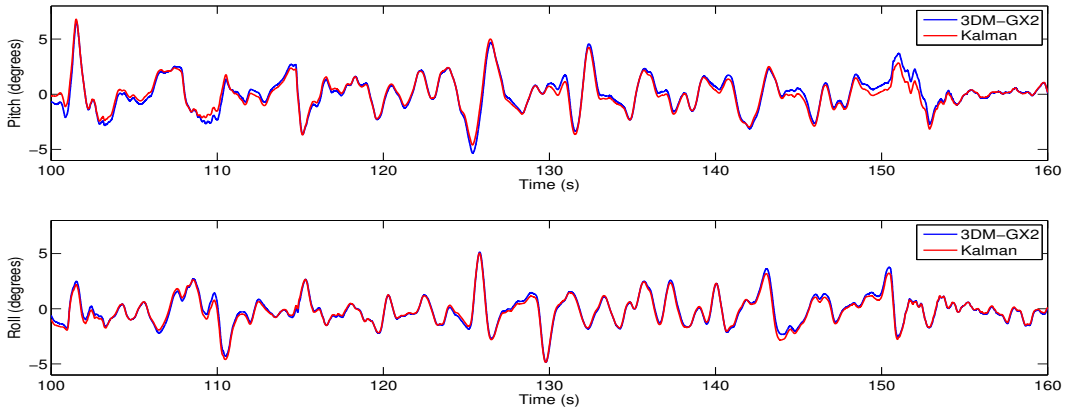

Figure 3.5: Comparison of the attitude estimations in-flight

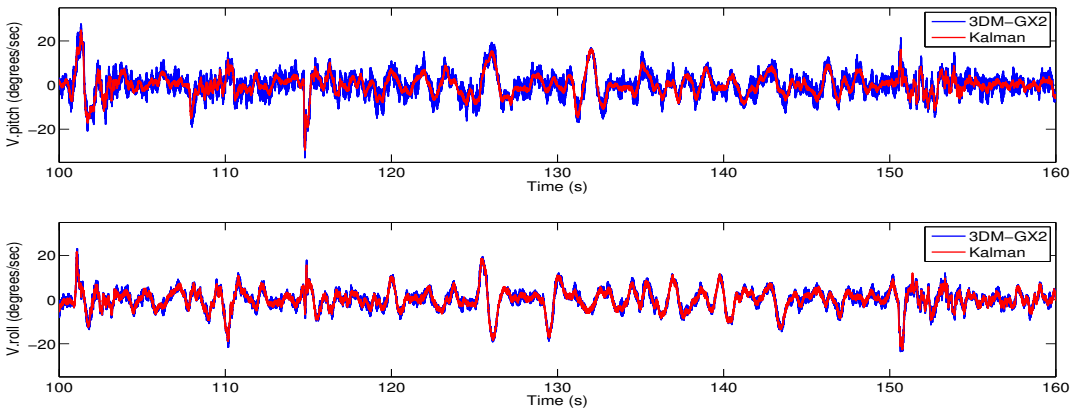

Figure 3.6: Comparison of the angular velocity estimations in-flight.

Chapter 4

Delays in the Attitude Estimation Process

4.1 Introduction

Different approaches to the attitude estimation problem have been reported in literature, e.g., Kalman filters [11, 12] or complementary filters [13]. Although its convergence is not guaranteed, the Extended Kalman Filter (EKF) has been the workhorse of real-time spacecraft attitude estimation for quite some time [30]. On the other hand, some processes are internally performed before the observer algorithms are computed. For example, during the data acquisition process in an IMU, the signals are low-pass filtered to remove noise and avoid aliasing effects. The filter introduces a time delay in the measurements, which results in an attitude estimation which is also delayed by the same amount of time. One of the unavoidable sources of delay is the low-pass filtering before sampling. The other one is the computational time required to run the estimation algorithm.

It is well-known that measurement delays decrease the phase margin and can even lead to the instability of the controlled process [31]. The incorporation of delayed measurements into the Kalman filter while preserving optimality is far from being trivial. When the delay consists only of a few sample periods, the problem can be handled optimally by augmenting the state vector [32]. However, for larger delays, the computational burden of this approach becomes too large. This topic has been investigated in [33]. More recent work on this topic has been done in [34], where a general delayed Kalman filter framework is derived for linear-time invariant systems.

Dead-time compensation techniques are frequently used in the control of time-delay systems [35, 36, 37]. In [38] a discrete predictor for continuous-time plants with time delay is proposed and the closed-loop stability is proved. Later, the proposed predictor has been explored to perform in different scenarios [39, 40, 41, 42, 43, 44]. Nowadays, almost any con-

trol system application is implemented by using a computer, discrete-time predictor-based control schemes increase their internes in practical applications [45, 46].

The goal of this paper is to improve the estimation of the pitch and roll angle with a low-cost IMU by proposing an observer-predictor algorithm (OP-A). The proposed scheme uses a KF and a discrete-time predictor to fuse the measurements coming from this sensor. The KF estimates the roll and pitch angles and corrects the bias of the gyroscopes, while the predictor counteracts the inherent delay in the estimated states.

In practice, it is observed that attitude estimation obtained by applying fusing algorithms to the inertial sensors measurements exhibit a delay with respect to their real value. For illustration purposes, the angular measurements coming from a commercial IMU (the Mircrostrain 3DM-GX2) are compared with those of set of encoders. It is well know that the encoders are faster and more accurate than any IMU. Thus, the delayed measurement of the commercial IMU is represented in Figure 4.1 along with the ideal value measured by encoders. The experimental platform that allows taking these measurements is described in detail in Section 2.5.1.

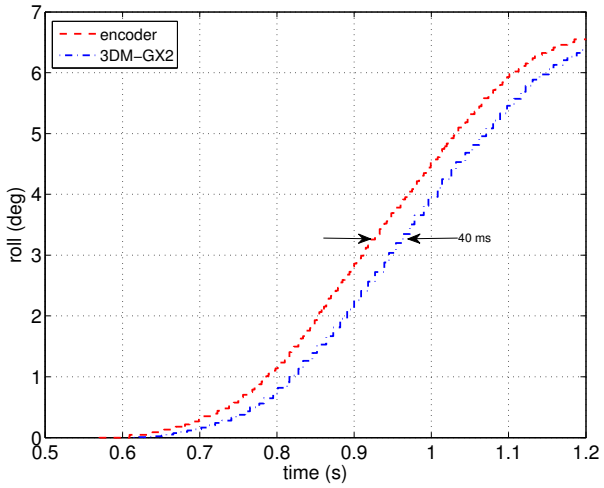

Figure 4.1: Delay of the 3DM-GX2, a commercial IMU

One of the unavoidable sources of delay is the low-pass filtering before sampling. During the data acquisition process in an IMU, the signals are low-pass filtered to remove noise and avoid aliasing effects. The other one is the computational time required to run the estimation algorithm, which is often carried out in an on-board microcontroller. In addition, it is well-known that measurement delays decrease the phase margin and can even lead to the instability of the controlled process.

4.2 Time-Delay Compensation Using Inertial Measurement Systems

4.2.1 Observer-Predictor Scheme

Kalman filter

The following dicrete-time filter derived in Section 3.2.1 will be used hereafter

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1} &= \begin{bmatrix} 1 & -T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -T \\ 0 & 0 & 0 & 1 \end{bmatrix} \hat{\boldsymbol{x}}_k + \begin{bmatrix} T & 0 \\ 0 & 0 \\ 0 & T \\ 0 & 0 \end{bmatrix} \boldsymbol{u}_k + \boldsymbol{w}_k \\ \hat{\boldsymbol{y}}_k &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \hat{\boldsymbol{x}}_k + \boldsymbol{v}_k \end{aligned} \tag{4.1}$$

where $\hat{\boldsymbol{x}}_k = [\hat{\phi}_k,\, \hat{\beta}_{x_k},\, \hat{\theta}_k,\, \hat{\beta}_{y_k}]^T$ and $\hat{\boldsymbol{y}}_k = [\hat{a}_{x_k},\, \hat{a}_{y_k}]^T$ are the discrete state and output vectors, $\boldsymbol{u}_k = [\bar{\omega}_{x_k},\, \bar{\omega}_{y_k}]^T$ defines the system input which consists of the measured angular velocities, and $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ represent the discrete process and measurement vectors, respectively.

h-step ahead Predictor

The discrete-time predictor algorithm used to improve the KF estimation is described in this part. The predictor algorithm compensates the delays in the estimated variables improving considerably the closed-loop stability.

The state of the plant is fully accessible but there is a known constant transmission delay τ, which is assumed to be a multiple of the sampling period[1] T, i.e., $\tau = Td$. The measured state can be thus written as

$$\bar{\boldsymbol{x}}_k = \boldsymbol{x}_{k-d} \tag{4.2}$$

An h-step ahead predicted state $\tilde{\boldsymbol{x}}_{k+h}$, with $h \in \mathbb{Z}^+$ being a design parameter, is computed using the discrete-time model of the plan [38] in order to counteract the delay

$$\tilde{\boldsymbol{x}}_{k+h} = \boldsymbol{A}^h \bar{\boldsymbol{x}}_k + \sum_{i=0}^{h-1} \boldsymbol{A}^{h-i-1} \boldsymbol{B} \boldsymbol{u}_{k+i-h} \tag{4.3}$$

where $\tilde{\boldsymbol{x}}, \bar{\boldsymbol{x}} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^m$, $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times m}$.

The proposed algorithm consists of applying the predictor to the Kalman estimates, i.e., $\bar{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k$. The resulting algorithm can be considered as a self-contained predictor-based observer, which is depicted in Figure 4.2.

[1] As the control structure should be robust under model parameters uncertainty, the round-off of the fractional delay will not be a problem [38].

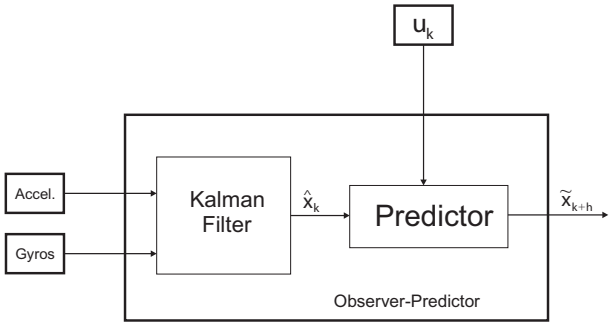

Figure 4.2: Observer-predictor scheme diagram

4.2.2 Simulations

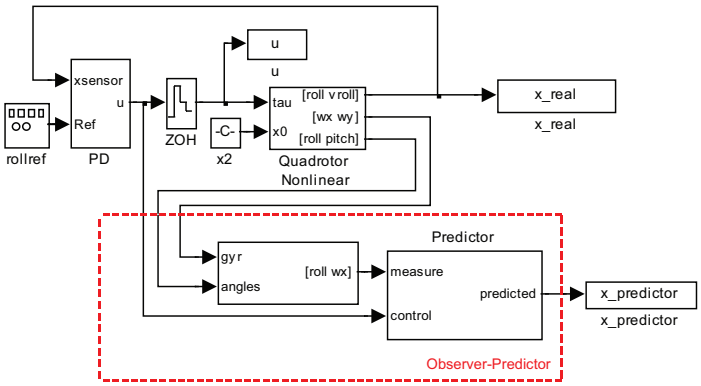

Figure 4.3: Simulink model

For the sake of brevity and without loss of generality, only the roll axis of the quadrotor is considered in what follows. Therefore, the state of the plant is given by $\boldsymbol{x} = [\phi,\, \dot{\phi}]^T$ while the dynamic model is a double integrator. Thus

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ b \end{bmatrix} u \tag{4.4}$$

where $u = \tau_\phi$ represents the external torque in the roll axis. A zero-order hold discretization of (4.4) leads to

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}u_k \tag{4.5}$$

with

$$\boldsymbol{A} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 0 \\ T \end{bmatrix} \tag{4.6}$$

The parameter h is chosen to be equal to the number of delayed sample periods d. In simulations, d is known, whereas in the experiments it has to be measured.

An h-step ahead prediction given by (4.3) is proposed to compensate the delay in the system. Simulations were carried out using the simulink model depicted in Figure 4.3. The nonlinear quadrotor model in (5.1) is used to represent the plant. For the sake of simplicity, only references in the roll angle are applied while pitch and yaw are driven to zero using PD controllers. The predictor is applied to the estimates given by the Kalman filter. The results are shown in Figure 4.4. Notice that in this figure the predictor algorithm improves the estimated value and compensates the delay.

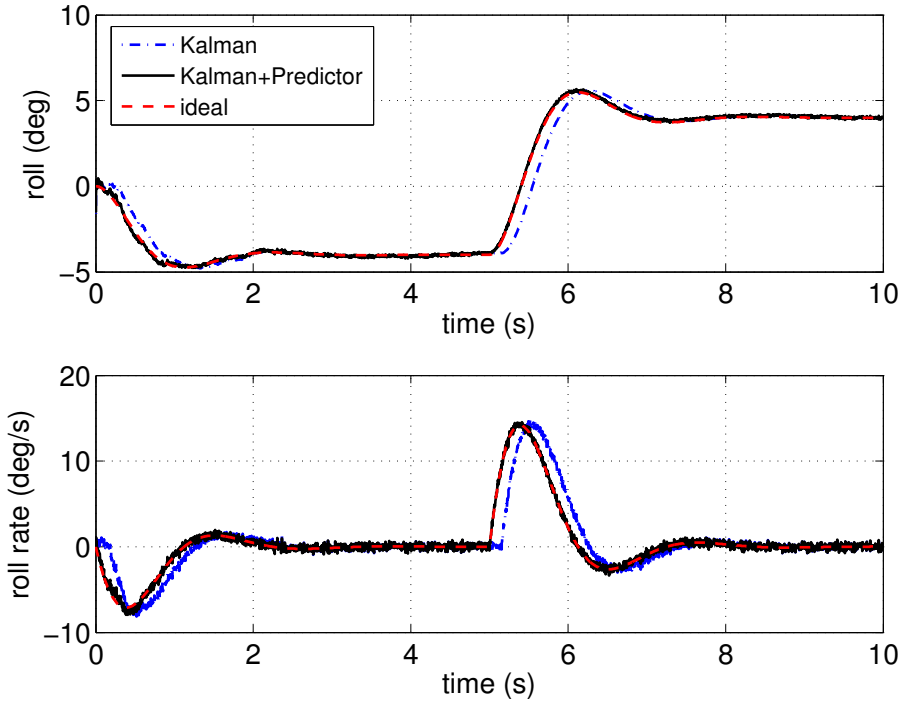

Figure 4.4: Simulated measurements

As aforementioned, a delayed measurement decreases the performance of a given controller. A simple state-feedback controller with reference tracking was used

$$u_k = [\phi_k^*,\, 0]^T - \boldsymbol{K}\boldsymbol{x}_k = k_p(\phi_k^* - \phi_k) - k_d\dot{\phi}_k \tag{4.7}$$

Figure 4.5 shows the output of the closed-loop system and the control action when the different state measurements are fed to the controller. Notice how oscillations arise when the delayed measurement from the Kalman filter is used. However, the use of the predictor improves the performance substantially, and the response gets very close to that of the system when using a non-delayed measurement.

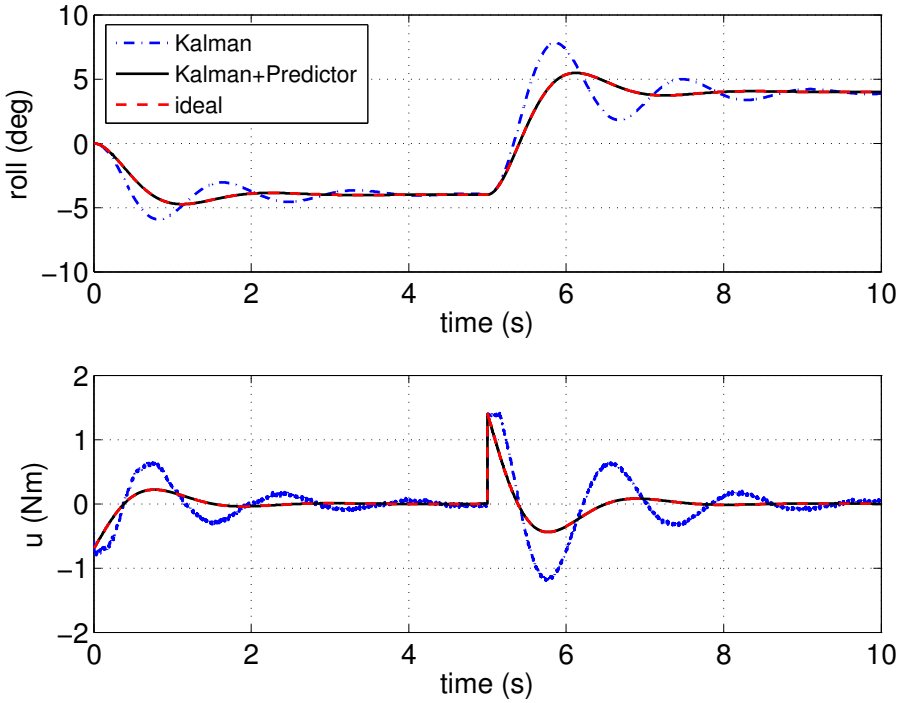

Figure 4.5: Simulated closed-loop response

4.2.3 Experiments

Some experiments were carried out using the platform shown in Figure 2.2. It is thought of as a test bed platform of control algorithms for vertical lift off vehicles, so that the translational degrees of freedom are clamped for convenience. The orientation of the vehicle is measured by means of optical encoders with an accuracy of 0.04 deg. These encoders provide almost-true non-delayed angular measurements in three axis. The angular rate was computed offline from the encoder measurement by using central difference approximation and filtering. The same controller structure as in the simulations (4.7) was used for the experiments.

In order to illustrate the performance of the proposed algorithm, two experiments are carried out. First, the system is controlled via state feedback, according to (4.7), using the measurements coming from the 3DM-GX2. The different state estimations are shown in Figure 4.6. A detail of the rising phase of the response can be seen in Figure 4.7. The delay of the 3DM-GX2 is quantified as 40 ms, while the delay if the proposed OP-A is used is almost negligible.

In the second experiment, the benefits of using the measurement obtained with the OP-A are analyzed. For this purpose, the OP-A is implemented in real-time. The system is brought to marginal stability by increasing the gain of the controller, and a step reference of 8 deg is applied. The result is shown in Figure 4.8. Notice that, for a given controller, the system becomes unstable when the measurement of the 3DM-GX2 is used. However, if the

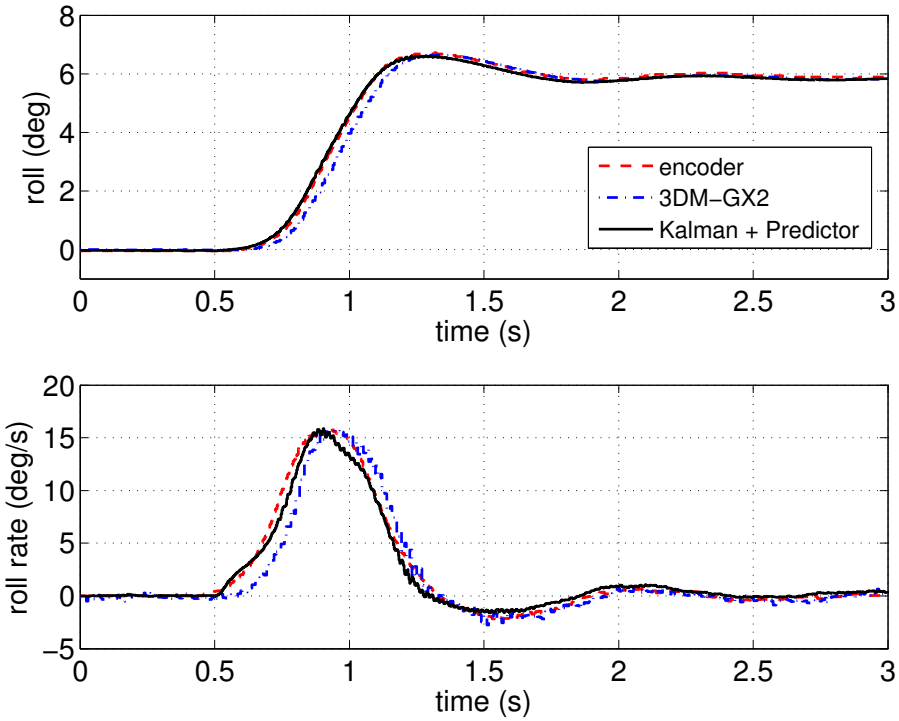

Figure 4.6: Experimental state measurements

measurement obtained by the OP-A is used, the system remains stable.

4.2.4 Conclusions

A new attitude estimation approach for quadrotor vehicles based on an observer-predictor algorithm is presented. The scheme consists of a Kalman filter that estimates the desired states and an h-step ahead predictor that improves the estimated measurement. Several simulations were carried out to validate the proposed schema and some graphs were selected to illustrate its behavior. In addition, real-time validation was also carried out. Experimental results show that the proposed algorithm improves significantly the measurements of a commercial IMU. Finally, closed-loop experiments evidence the importance of having a non-delayed measurement in fast unstable system such as quadrotors. For a given state-feedback controller, the delayed measurements of the commercial IMU resulted in an unstable response while the measurements obtained with the proposed algorithm succeeded in stabilizing the system.

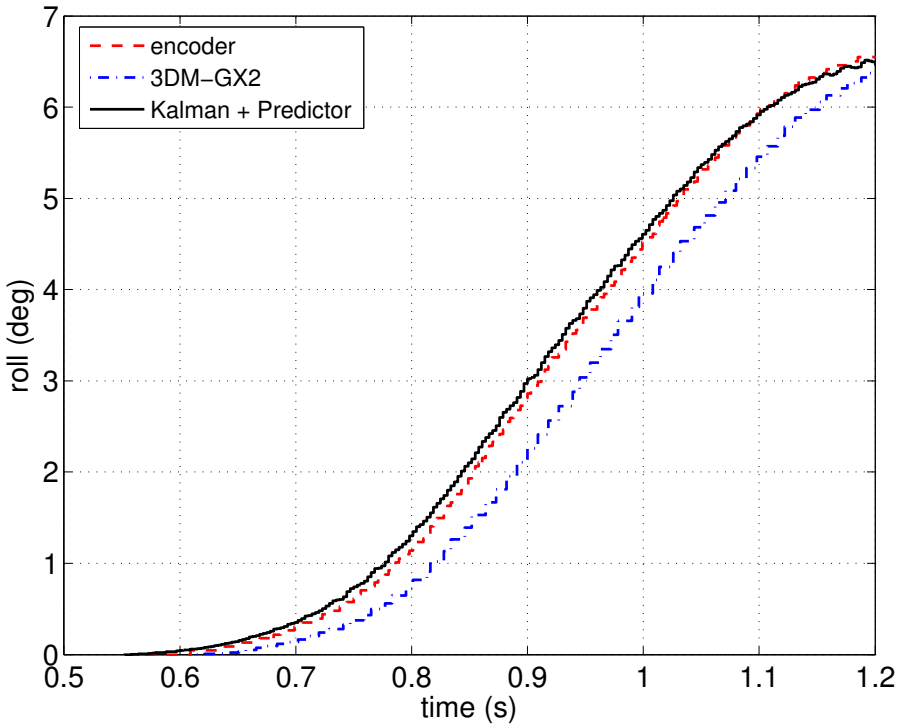

Figure 4.7: Time-delay comparison when estimating the angular position using an encoder and the 3DM-GX2 commercial IMU

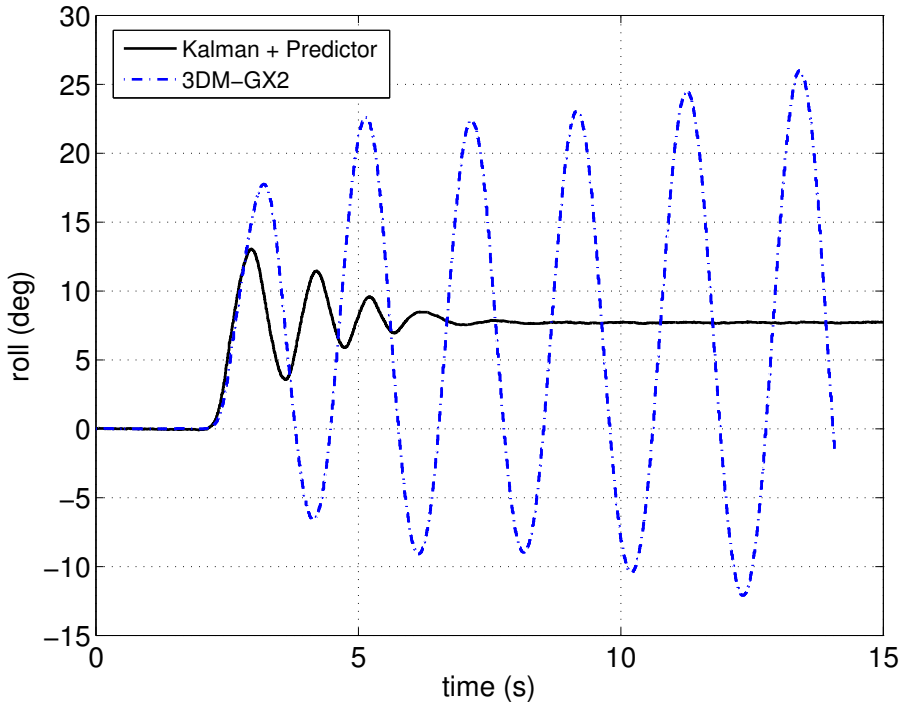

Figure 4.8: Experimental closed-loop response

Chapter 5

UDE-based Control

5.1 Introduction

Although many linear and nonlinear attitude control strategies have been proposed in the literature, very few of these controllers have been validated in real flight tests and the most popular techniques are still based on the classical PID-based control strategy with nested saturation algorithms [47, 48, 49, 1, 50, 51, 52, 53]. This is mainly due to the fact that, in practical applications, the attitude is stabilized via an on-board controller [54]. The unstable nature of quadrotors and their fast rotational dynamics impose strong requirements on the attitude controller. As the controllers must run typically at very high frequencies [16], algorithms with low computational cost are needed. Also, only few of them explicitly address the robustness issue with respect to uncertainties of the physical parameters and external disturbances [18], which is critical for UAVs because the aerodynamic effects are extremely hard to be accurately modeled [55]. Similarly, in outdoor applications, a UAV is constantly perturbed by wind gusts [56]. Therefore, good disturbance rejection capability is also demanded.

The UDE-based control algorithm, which was originally proposed in [57], is a robust control strategy that is devoted to handling uncertainties and disturbances. Based on the assumption that a signal can be recovered by passing it through a filter with the appropriate bandwidth, all uncertainties and disturbances can be treated as a lumped signal to be estimated and compensated in the controller straight away. The UDE strategy has demonstrated excellent performance in handling uncertainties and disturbances and has been successfully applied to robust input-output linearization [58], linear and nonlinear systems with state delays [59, 60], combined with sliding-model control [61], and uncertain nonlinear systems [62]. Some applications of UDE can be also found in [63, 64, 65].

In this chapter, the UDE-based control is applied to quadrotor systems. At first, the UDE-based control strategy is revisited and further developed to highlight the decoupling of the reference tracking from disturbance rejection and the trade-off between disturbance

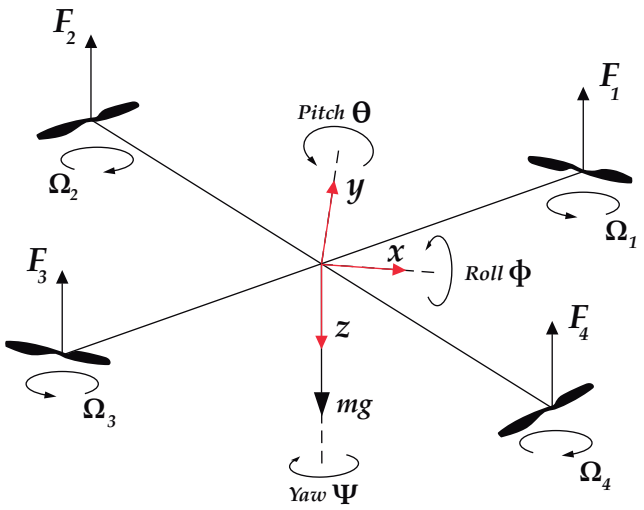

Figure 5.1: Sketch of a 6-DoF quadrotor

rejection and noise attenuation. The desired tracking performance can be guaranteed via selecting an appropriate reference model. The trade-off between the disturbance rejection and noise attenuation can be achieved via tuning one parameter in the UDE control law, which makes the real-time implementation very straightforward. Then, the dynamic model of quadrotors is formulated as a set of double integrators with non-linear uncertainties so that the UDE-based control strategy can be applied. A numerical example is presented to show that, for the same degree of robustness, the UDE has much better disturbance rejection than an equivalent 2-DoF PID controller. The easy tuning process and the advantage of the UDE are validated with a 3 DoF quadrotor laboratory setup and then with a 6 DoF quadrotor in real flight tests. The proposed UDE-based controller offers much better performance than a widely-used PID controller, which is illustrated by indexes like the integral absolute error (IAE) and the root mean squared error (RMSE).

5.2 Modeling of Quadrotor Systems

The key variables involved in the formulation of the quadrotor dynamics are illustrated in Figure 5.1. A fairly accurate model of a quadrotor is given by the following set of nonlinear

equations [47]

$$\begin{aligned}
\ddot{\phi} &= \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J}{I_x}\dot{\theta}\Omega + \frac{\tau_\phi}{I_x}, \\
\ddot{\theta} &= \frac{I_z - I_x}{I_y}\dot{\psi}\dot{\phi} + \frac{J}{I_y}\dot{\phi}\Omega + \frac{\tau_\theta}{I_y}, \qquad (5.1) \\
\ddot{\psi} &= \frac{I_x - I_y}{I_z}\dot{\theta}\dot{\phi} + \frac{\tau_\psi}{I_z}, \\
\ddot{z} &= g - \frac{\Gamma}{m}\cos\phi\cos\theta, \\
\ddot{x} &= -\frac{\Gamma}{m}\cos\phi\sin\theta, \qquad (5.2) \\
\ddot{y} &= \frac{\Gamma}{m}\sin\phi, \qquad (5.3)
\end{aligned}$$

where ϕ, θ, and ψ are the roll, pitch and yaw Euler angles, and I_i, $i = \{x, y, z\}$ are the moments of inertia, $\Omega = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3$ with Ω_i being the rotor speed of the ith motor, τ_i, $i = \{\phi, \theta, \psi\}$ are the input torques along the axes of a body-fixed reference frame, z is the coordinate along the $z-$axis of the body-fixed frame which points downwards, g is the gravity acceleration, m is the mass of the quadrotor and $\Gamma = \sum_{i=1}^{4} F_i$ is the total thrust. Since I_x and I_y are almost symmetric and only differ slightly because of construction tolerance, the roll and the pitch axes have very similar dynamics.

5.3 Control Based on Uncertainty and Disturbance Estimation

5.3.1 Revisit of the UDE-based Robust Control Strategy

General case

Consider the following class of nonlinear single-input multiple-output (SIMO) systems

$$\dot{\mathbf{x}}(t) = \mathbf{A}_p\mathbf{x}(t) + \mathbf{B}_p u(t) + \mathbf{f}(\mathbf{x}, u, t) + \mathbf{d}(t) \qquad (5.4)$$

where $\mathbf{x}(t) \in \boldsymbol{R}^n$ and $u(t) \in \boldsymbol{R}$ are the state and control variable respectively, $\mathbf{f}(\mathbf{x}, u, t) : \boldsymbol{R}^n \times \boldsymbol{R} \times \boldsymbol{R}^+ \to \boldsymbol{R}^n$ is a possibly unknown non-linear function, and $\mathbf{d}(t) : \boldsymbol{R}^+ \to \boldsymbol{R}^n$ is the vector of unknown disturbances. The full state is accessible and measured. The state matrix can be split as $\mathbf{A}_p = \mathbf{A} + \Delta\mathbf{A} \in \boldsymbol{R}^{n\times n}$, where $\mathbf{A}$ is the known state matrix and $\Delta\mathbf{A}$ is the unknown part of the state matrix, and similarly with the control matrix, $\mathbf{B}_p = \mathbf{B} + \Delta\mathbf{B} \in \boldsymbol{R}^n$, where $\mathbf{B}$ is the known control matrix and $\Delta\mathbf{B}$ the unknown control matrix.

Moreover, assume that the desired dynamics of the closed-loop system are given in terms of the linear model described by

$$\dot{\mathbf{x}}_m(t) = \mathbf{A}_m\mathbf{x}_m(t) + \mathbf{B}_m r(t), \qquad (5.5)$$

where $\mathbf{A}_m \in \boldsymbol{R}^{n\times n}$, $\mathbf{B}_m \in \boldsymbol{R}^n$, $\mathbf{x}_m \in \boldsymbol{R}^n$ and $r(t) \in \boldsymbol{R}$ is a piecewise continuous and uniformly bounded command to the system.

The control objective is to derive a control law such that the state error between the actual plant and the reference model, defined as

$$\boldsymbol{e} = \mathbf{x}_m - \mathbf{x}, \tag{5.6}$$

is stable and satisfies the dynamic equation

$$\dot{\boldsymbol{e}} = (\mathbf{A}_m + \mathbf{K})\boldsymbol{e} \tag{5.7}$$

where $\mathbf{K}$ is the error feedback gain with appropriate dimensions.

Combining the Equations (5.4)-(5.7), it results in

$$\mathbf{A}_m x + \mathbf{B}_m r - (\mathbf{A} + \Delta\mathbf{A})\mathbf{x} - (\mathbf{B} + \Delta\mathbf{B})u - \mathbf{f} - \mathbf{d} = \mathbf{K}\boldsymbol{e}. \tag{5.8}$$

Then, the control input signal $u(t)$ should satisfy

$$\mathbf{B}u = \mathbf{A}_m\mathbf{x} + \mathbf{B}_m r - \mathbf{A}\mathbf{x} - \boldsymbol{u}_d - \mathbf{K}\boldsymbol{e} \tag{5.9}$$

where

$$\boldsymbol{u}_d = -\Delta\mathbf{A}\mathbf{x} - \Delta\mathbf{B}u - \mathbf{f} - \mathbf{d} \tag{5.10}$$

denotes the unknown terms in Equation (5.8). According to the system dynamics described by Equation (5.4), it is possible to rewrite the Equation (5.10) as

$$\boldsymbol{u}_d = \dot{\mathbf{x}} - \mathbf{A}\mathbf{x} - \mathbf{B}u \tag{5.11}$$

which indicates that the unknown dynamics and disturbances can be estimated from the known dynamics of the systems and control signal. Following the procedures provided in [57, 60, 63], the signal given by (5.11) can be accurately represented in the frequency domain by[1]

$$U_d(s) = G_f(s)\left(\dot{\mathbf{X}}(s) - \mathbf{A}\mathbf{X}(s) - \mathbf{B}U(s)\right) \tag{5.12}$$

where the filter $G_f(s)$ is assumed to be a strictly proper low-pass filter with unitary steady-state gain and zero phase shift over the spectrum of u_d [57, 66]. Using Equations (5.9) and (5.12) and, after some algebraic manipulation, the UDE-based control law can be derived as

$$\begin{aligned} U(s) = \frac{1}{1-G_f}\mathbf{B}^+ [&\mathbf{A}_m\mathbf{X} + \mathbf{B}_m R - \mathbf{A}\mathbf{X}(1-G_f) \\ &-\mathbf{K}\mathbb{E} - sG_f\mathbf{X}] \end{aligned} \tag{5.13}$$

where $\mathbf{B}^+ = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$ is the pseudoinverse of $\mathbf{B}$. As shown in [67, 57], Equation (5.13) is always satisfied if the system is described in the controllable canonical form.

[1]The Laplace transformation is introduced to facilitate the manipulation of expressions.

The case with a SISO system

Assume that the system under consideration is a controllable SISO system. It can be represented in the controllable canonical form, i.e., the matrices involved admit the following partitioning

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{n-1} \\ & \mathbf{A}_1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{n-1} \\ b \end{bmatrix};$$
$$\mathbf{A}_m = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{n-1} \\ & \mathbf{A}_{1m} \end{bmatrix} \quad \mathbf{B}_m = \begin{bmatrix} \mathbf{0}_{n-1} \\ b_m \end{bmatrix};$$

where $\mathbf{A}_1 = [-a_1, -a_2, \ldots, -a_n]$ and $\mathbf{A}_{1m} = [-a_{m1}, -a_{m2}, \ldots, -a_{mn}]$. Then the transfer functions from the input to the first state variable X_1 (taken as the output Y) for both the known plant model and the reference model are given by $G(s) = \frac{b}{P(s)}$ and $G_m(s) = \frac{b_m}{P_m(s)}$, respectively, with $P(s) = s^n + \sum_{i=1}^{n} a_i s^{i-1}$ and $P_m(s) = s^n + \sum_{i=1}^{n} a_{im} s^{i-1}$. Substituting the above matrices into (5.13), then the UDE-based control law for system (5.4) is derived as:

$$\begin{aligned} U = \frac{1}{(1-G_f)b} \Bigg(& b_m R + \sum_{i=1}^{n} (a_i - a_{im}) \mathbf{X}_i - G_f \sum_{i=1}^{n} a_i \mathbf{X}_i \\ & - s G_f \mathbf{X}_n + \sum_{i=1}^{n} k_i \mathbb{E}_i \Bigg) \end{aligned} \tag{5.14}$$

Assume that the frequency range of the unknown system dynamics and the external disturbances is limited by ω_f. Then, $G_f(s)$ can be approximately chosen as a low-pass filter

$$G_f(s) = \frac{1}{Ts+1}$$

with $T = 1/\omega_f > 0$. In this case, $\frac{1}{1-G_f}$ is a PI controller, denoted as

$$PI(s) = \frac{Ts+1}{Ts}.$$

Assume $K = 0$ for simplicity. Then the control law (5.14) can be rewritten, after some algebraic manipulations, as

$$U(s) = PI(s) \left(\frac{b_m}{b} R(s) - M(s) Y(s) \right) \tag{5.15}$$

with

$$M(s) = \frac{b_m}{b} G_m^{-1}(s) - PI^{-1}(s) G^{-1}(s).$$

Denoting by $G_p(s)$ the actual transfer function of the plant and considering the nominal case, i.e., $G_p(s) = G(s)$, then the relevant closed-loop transfer functions for reference tracking, disturbance rejection and noise attenuation are, respectively,

$$\frac{Y(s)}{R(s)} = \frac{b_m}{b}\frac{PI(s)G_p(s)}{1 + PI(s)M(s)G_p(s)} = G_m(s), \tag{5.16}$$

$$\frac{Y(s)}{D(s)} = \frac{G_p(s)}{1 + PI(s)M(s)G_p(s)} = \frac{b}{b_m}G_m(s)\frac{Ts}{Ts+1}, \tag{5.17}$$

$$\frac{Y(s)}{N(s)} = \frac{-PI(s)G_p(s)M(s)}{1 + PI(s)M(s)G_p(s)} = 1 - \frac{b}{b_m}\frac{G_m(s)}{G_p(s)}\frac{Ts}{Ts+1}. \tag{5.18}$$

It is evident that the UDE-based control structure allows decoupling between reference tracking and disturbance rejection. The parameters of the UDE-based control are "automatically" tuned by choosing the reference model which defines the closed-loop tracking performance. But the most interesting fact is that there is only one parameter left, the time constant of the filter T, to adjust the disturbance decay rate. This parameter should be chosen as small as possible to improve the disturbance rejection and, theoretically, it is only limited by the sampling period. However, in practice, reducing T would amplify higher frequency noise from measurements to enter into the system, as can be seen from (5.18). Hence, the choice of T is a compromise between fast disturbance rejection and acceptable noise attenuation.

Because of these structural properties of the UDE-based control strategy, it is very straightforward to design a control system and tune the parameters to meet the requirements: after selecting the reference model to meet the desired reference tracking performance, the time constant T can be either reduced or increased to reach a trade-off between disturbance rejection and noise attenuation. This allows an easy on-line tuning of the implemented controller.

5.3.2 Quadrotor equations

Although there are 6 DoF considered in the model (5.1), only the first four variables are critical when controlling quadrotors because the displacements on the $xy-$plane can be controlled by commanding the roll and pitch angles, as explained in [14]. The first four variables in the model (5.1) can be written in the form of (5.4) as

$$\begin{aligned} \ddot{\phi} &= f_1(\dot{\theta}, \dot{\psi}) + u_\phi \\ \ddot{\theta} &= f_2(\dot{\psi}, \dot{\phi}) + u_\theta \\ \ddot{\psi} &= f_3(\dot{\theta}, \dot{\phi}) + u_\psi \\ \ddot{z} &= f_4(\theta, \phi) + u_z \end{aligned} \tag{5.19}$$

with the control inputs given by

$$u_\phi = \frac{\tau_\phi}{I_x}, \quad u_\theta = \frac{\tau_\theta}{I_y}, \quad u_\psi = \frac{\tau_\psi}{I_z} \quad \text{and} \quad u_z = g - \frac{\Gamma}{m}$$

and

$$
\begin{aligned}
f_1(\dot{\theta},\dot{\psi}) &= \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} - \frac{J}{I_x}\dot{\theta}\Omega,\\
f_2(\dot{\psi},\dot{\phi}) &= \frac{I_z - I_x}{I_y}\dot{\psi}\dot{\phi} + \frac{J}{I_y}\dot{\phi}\Omega,\\
f_3(\dot{\theta},\dot{\phi}) &= \frac{I_x - I_y}{I_z}\dot{\theta}\dot{\phi},\\
f_4(\theta,\phi) &= \frac{\Gamma}{m}(1-\cos\phi\cos\theta).
\end{aligned}
$$

Each of the equations in (5.19) represents a single-input dynamic system of a double integrator. They share the same known matrix $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and the difference is that they have different matrices $\mathbf{B}$. Note that thanks to the excellent capability of UDE to handle uncertainties and disturbances, there is no need to know the parameters of the quadrotor.

5.3.3 The Case with a 3D Hover System

In this section, a 3D hover system shown in 2.2, is investigated and validated with simulations and experiments.

Controller design

In order to clearly demonstrate the effectiveness of the UDE-based controller, only the roll and pitch control will be considered and the following 2-DoF PID controller with the weighted set-point

$$U(s) = \left(\epsilon K_p + \frac{K_i}{s}\right) R(s) - \left(K_p + \frac{K_i}{s} d + K_d s\right) \theta(s)$$

with $\epsilon = 0.6956$, $K_p = 90$, $K_d = 50$ and $K_i = 39.2$ is designed as a base for comparison. The corresponding closed-loop transfer function is

$$G_c(s) \;=\; \frac{6.26}{s^2 + 4.37s + 6.26}.$$

For the UDE-based control, the reference model for the roll and pitch can be chosen as the following second-order system in the canonical form

$$\dot{\mathbf{x}}_m = \begin{bmatrix} 0 & 1 \\ -6.26 & -4.37 \end{bmatrix} \mathbf{x}_m + \begin{bmatrix} 0 \\ 6.26 \end{bmatrix} r \tag{5.20}$$

with $r = \{\theta^{ref}, \phi^{ref}\}$, θ^{ref} and ϕ^{ref} being the reference commands for the pitch and roll angles, respectively. T is chosen as $T = 0.28$s to achieve the same robustness index as the

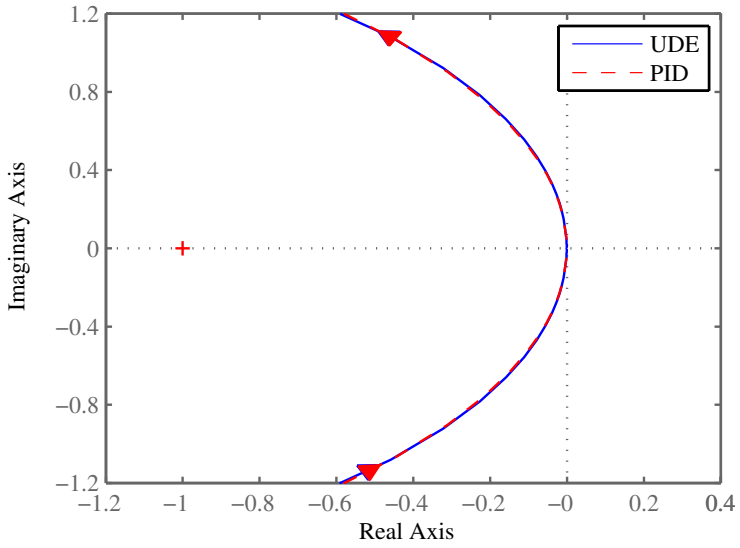

Figure 5.2: Nyquist plots of the system with the PID controller and the UDE-based controller with $T = 0.28$s

PID controller, as shown in the Nyquist plots given in Figure 5.2. The resulting control law is

$$U_{ude}(s) = \frac{0.28s+1}{0.28s}\left(\frac{6.25}{0.1}C(s) - 6.25X(s) \right.$$
$$\left. -4.5\dot{X}(s) - \frac{s}{0.28s+1}\dot{X}(s)\right).$$

Simulation results

The simulation results for the roll axis are shown in Figure 5.3. It is shown that the UDE-based controller performs better in disturbance rejection while maintaining similar robustness. Note also that the set-point response is not affected by the choice of T, as it is decoupled, and the system has better disturbance rejection when T is decreased. When compared with the PID controller, it is shown that the UDE-based controller performs better in disturbance rejection while maintaining similar robustness.

The output responses with the effect of white noise in the measurement are shown in Figure 5.4. It can be seen that the UDE-based controller does not need much extra control effort to obtain better performance in disturbance rejection while maintaining similar performance in noise attenuation.

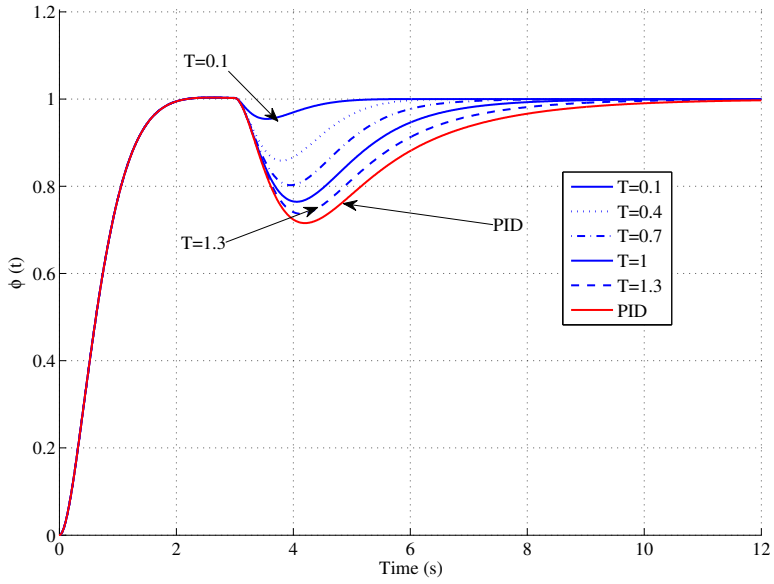

Figure 5.3: Influence of the parameter T in the disturbance rejection performance

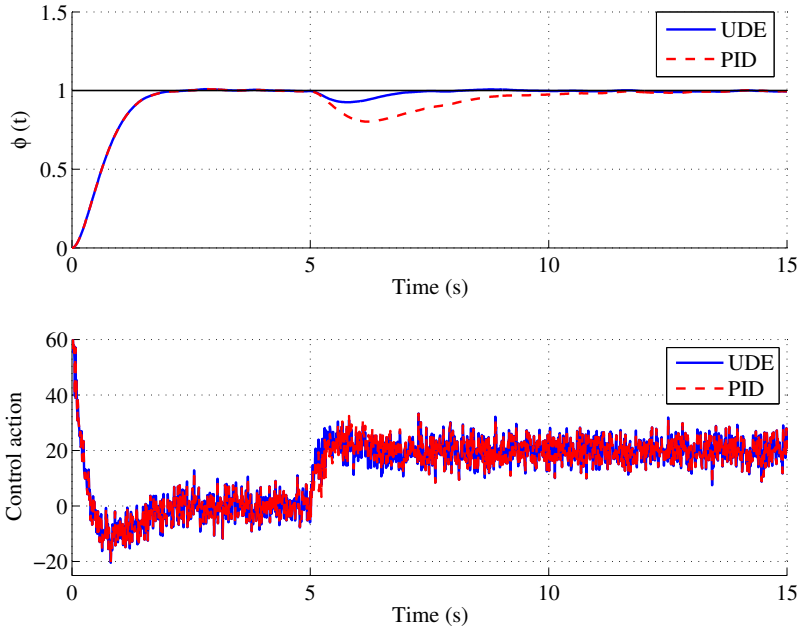

Figure 5.4: Nominal system responses with a white-noise measurement and a -10 step load disturbance at t = 5s

Experimental results

Since the state of the plant is fully accessible, the controllers were implemented in the following form for both the roll and pitch axes, using the same controller used in the simulations

$$U_{pid}(s) = \left(62.6 + \frac{39.2}{s}\right) C(s) - \left(90 + \frac{39.2}{s}\right) X(s) - 50\dot{X}(s),$$

where $X(s) = \{\Theta(s), \Phi(s)\}$ and $C(s) = \{\Theta^{ref}(s), \Phi^{ref}(s)\}$, depending on the axis under control. Two experiments were carried out, with one to demonstrate the performance of disturbance rejection and the other to demonstrate the performance of robustness against additional model uncertainties.

Figure 5.5 shows the output of the system from the first experiment, in which a square input signal of ± 5 deg was used as the reference signal. An input disturbance was applied at around $t = 33$s, generated by software to offset the control signal before sending to the motors. The UDE results in smaller output perturbations and faster disturbance rejection, as expected from the simulations in the previous section.

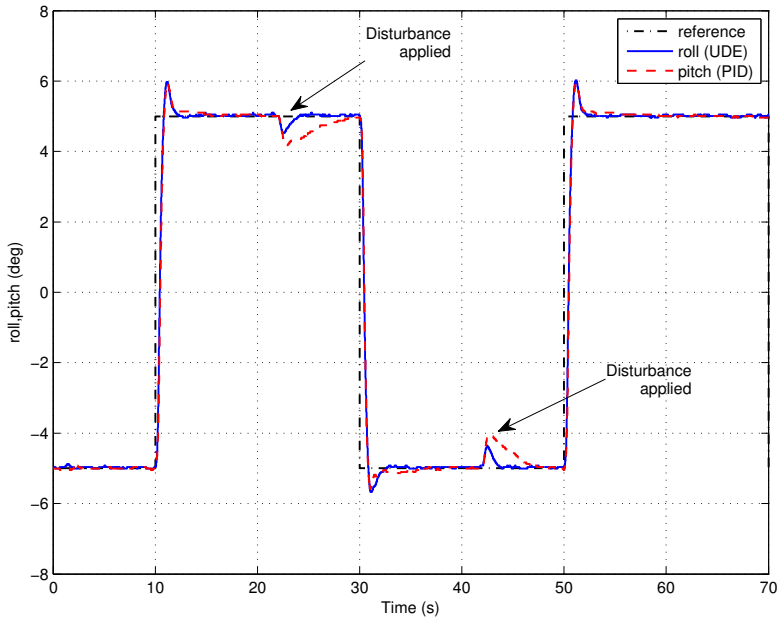

Figure 5.5: Disturbance rejection comparison for similar reference tracking performance. PID (roll), UDE (pitch)

The second experiment was carried out to test the robustness with additional model uncertainties. In this case, a weight of about 21g was placed at one end of the platform, which affects the moment of inertia of both axes and hence additional model uncertainties were added. The results of this experiment are shown in Figure 5.6. The UDE reacts faster than the PID when the weight was applied.A video of this experiment is available at https://vimeo.com/100991794.

5.3.4 Real Flight Tests with a Quadrotor

Although the platform described above is a suitable test-bed to validate the real-time control algorithms, there are some handicaps to overcome in real flights, e.g., large model uncertainties and disturbances generated by vibrations of the motors, linear accelerations, flapping and

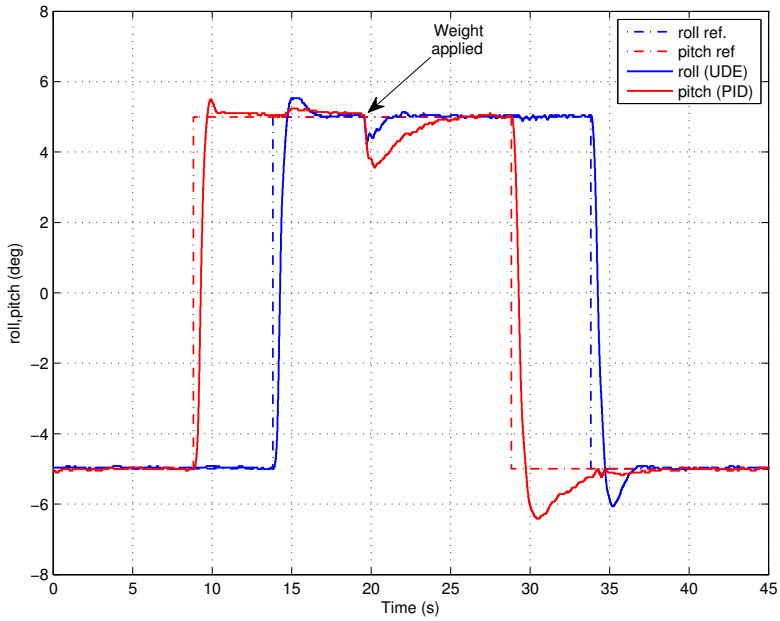

Figure 5.6: Robustness comparison for similar reference tracking performance

ground effects, noise in the sensors measurements, unknown perturbations as wind gusts, etc.

In this section, the experimental results from several real flight tests carried out on the quadrotor shown in Figure 2.3 are presented to further illustrate the control strategy, including the full control of the attitude (ϕ, θ and ψ) and the altitude (z) of the quadrotor. The proposed strategy is compared with the following control law that is widely used in real-time implementations [52, 68]:

$$u_x = \sigma(k_{p_x}\bar{x},\, p_x) + \sigma(k_{d_x}\dot{x},\, d_x) + \sigma(k_{i_x}\int \bar{x}\mathrm{d}t,\, i_x) \tag{5.21}$$

where $\sigma(\cdot,\,\cdot)$ is the saturation function defined as

$$\sigma(x,\, b) = \begin{cases} -b & \text{if } x < -b, \\ x & \text{if } -b \leq x \leq b, \\ b & \text{if } x > b, \end{cases} \tag{5.22}$$

k_{p_x}, k_{d_x} and k_{i_x} are the proportional, derivative and integral gains, respectively, and $\bar{x} = x_d - x$ is the tracking error with x_d the desired value. The same control law is applied to the roll and pitch axes, i.e., $x = \{\phi,\, \theta\}$, with the parameters given in Table 5.1.

The UDE-based controllers for each DoF were tuned using the double integrator model, as appearing in Equation (5.19), which implies $a_1 = a_2 = 0$. The parameters of the UDE control

law for each DoF are given in Table 5.2. Note that, in both cases, the roll and pitch axes are tuned to obtain the same performance, which makes it possible to carry out experiments for comparison by applying UDE and PID controllers to different axes, respectively, at the same time.

Table 5.1: Parameters of the PID controller used in real flight tests

DoF	k_{p_x}	k_{d_x}	k_{i_x}	p_x	d_x	i_x
Roll ϕ	3.3	1.2	0.03	100	50	50
Pitch θ	3.3	1.2	0.03	100	50	50
Altitude z	150	1.2	50	120	60	200

Table 5.2: Parameters of the UDE controller used in real flight tests

DoF	b	T	a_{m2}	a_{m1}	b_m
Roll ϕ	1.5	0.6	3.2	4	4
Pitch θ	1.5	0.6	3.2	4	4
Yaw ψ	1	5	7.6	16	16
Altitude z	1.7	0.6	3	2.25	2.25

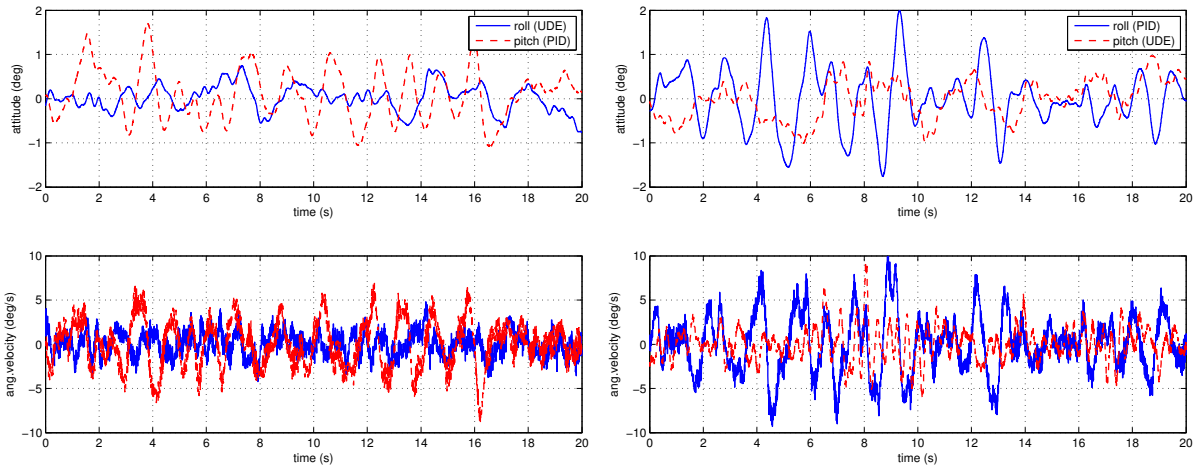

Figure 5.7: Real flight test of hovering: UDE for roll and PID for Pitch (left) and PID for roll and UDE for pitch (right).

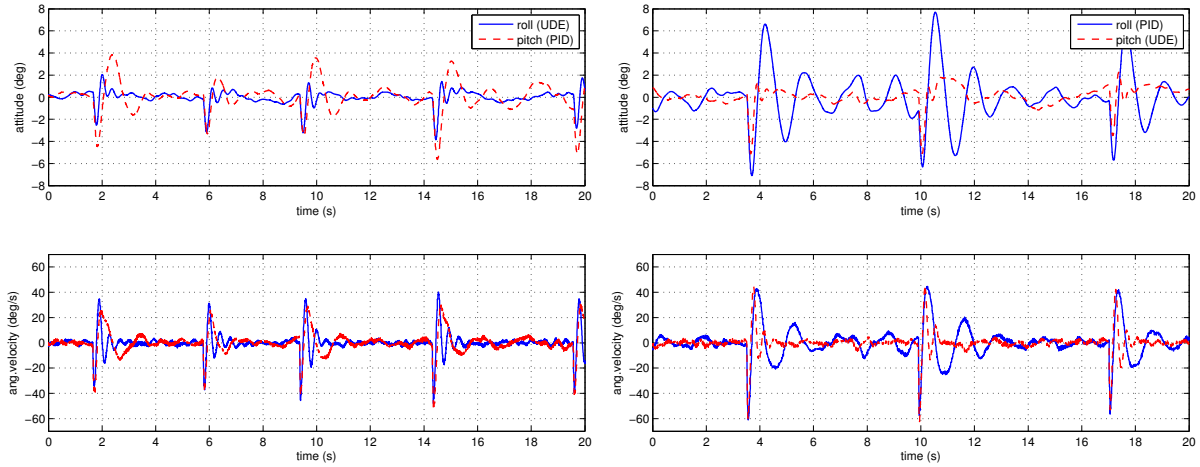

Figure 5.8: Disturbance rejection: UDE for roll and PID for Pitch (left) and PID for roll and UDE for pitch (right).

First flight test: Hovering (Figure 5.7)

In this experiment, the UDE-based control law was implemented for one of the roll and pitch axes, while the other axis was controlled using the PID control law in (5.21). Data was recorded while the quadrotor was hovering, i.e., flying freely with zero reference inputs in both axis. The experiment was repeated after swapping the controllers for each axis, in order to provide non-biased results. The results from both experiments are shown in Figure 5.7. In both cases, the axis controlled with the UDE-based strategy exhibits less deviation with respect to the reference input. This can be observed in both the angular position and the angular velocity. Integral Absolute Error (IAE) and Root Mean Squared Error (RMSE) are presented in Table 5.3 for comparison. It can be seen that the UDE-based controller outperforms the widely-used PID control law significantly.

Table 5.3: IAE and RMSE of the results in Figures 5.7 and 5.8

		Hovering (Fig. 5.7)		Disturbances (Fig. 5.8)	
		IAE	RMSE	IAE	RMSE
Pitch	UDE	7.3	0.47	11.3	0.89
	PID	9.8	0.59	17.8	1.37
Roll	UDE	4.7	0.30	7.9	0.66
	PID	10.5	0.69	30.2	2.2

Second flight test: Disturbance rejection (Figure 5.8)

The goal of this experiment is to provide comparative results of the disturbance rejection capability of both control strategies. The experiments were carried out via applying disturbances to the quadrotor while it was hovering, hitting the quadrotor by hand at an intermediate point between both axes. Due to the construction of the prototype, this is easy because such points can be any of the corners of the squared protection frame. Similarly, the experiment was performed twice, with different combinations of controllers. The results are shown in Figure 5.8. The UDE-based control strategy results in much faster performance in disturbance rejection than the PID controller. This can also be seen from the corresponding IAE and RMSE shown in Table 5.3. Indeed, the UDE-based controller outperforms the widely-used PID control law significantly.

Third flight test: Altitude control (Figure 5.9)

This experiment is slightly different from the previous ones. In this case, an altitude reference input pattern which consists of alternating steps of different magnitude is fed to to the controller. One experiment was performed to test each control strategy. The results of both experiments are shown together in Figure 5.9. Both control strategies performed similarly. Oscillations occurred when the quadrotor was close to the ground from a high altitude. This is due to the well-known "ground effect", which is an aerodynamic effect that increases the lift of the propellers. The "ground effect" acts as a spring making the system more oscillatory. The UDE-based control strategy is better at handling this effect.

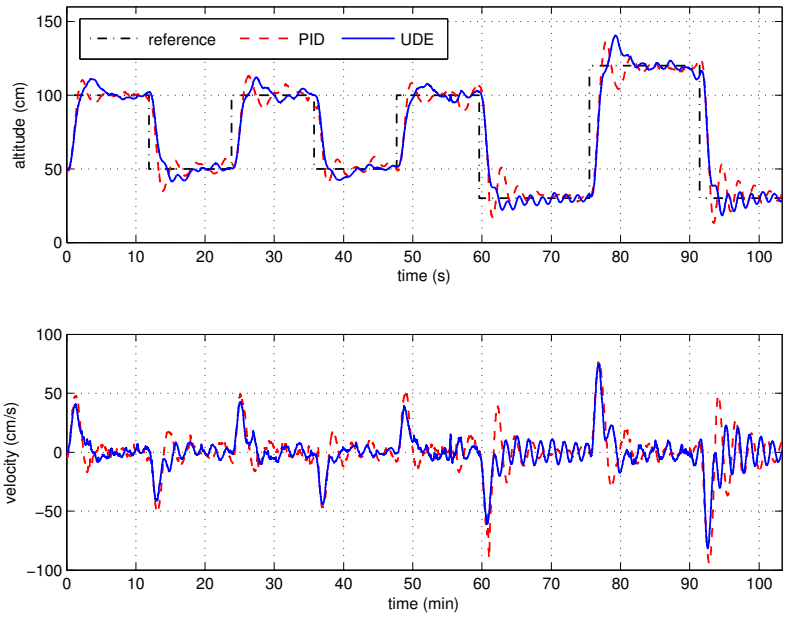

Figure 5.9: Altitude UDE vs. PID control

Fourth flight test: Full control (Figure 5.10)

The purpose of this experiment is to give an overall view of the behavior of a quadrotor fully controlled with the proposed UDE-based control strategy. The design parameters of the UDE-based controllers are given in Table 5.2. The roll and pitch references were set at zero and step changes in yaw and altitude were applied. The results are shown in Figure 5.10. Note that, even though the quadrotor is a coupled non-linear system, the UDE-based control strategy could deal with it fairly well. No visible coupling effects appeared in the roll and the pitch while the yaw and altitude references were changed. The RMSE of the roll and pitch angles are 0.39 deg and 0.46 deg, respectively. Note that experiments performed using the platform in Fig. 2.2 have revealed that the attitude measurements have an accuracy of 0.15 deg (RMSE), but the measurements are highly degraded in real flight tests due to the vibrations induced by the motors and the presence of lateral accelerations [20]. Hence, it is remarkable for the proposed UDE control strategy to achieve 0.39 deg and 0.46 deg RMSE for the roll and pitch angles.

A video clip of the quadrotor in real flight tests, including hovering and disturbance rejection is available at http://vimeo.com/101082533, which shows the excellent disturbance rejection capability under the full UDE-based control. A series of very large disturbances were applied but the quadrotor responded extremely well.

5.3.5 Conclusion

A robust control scheme based on the Uncertainty and Disturbance Estimator (UDE) has been developed for quadrotor systems. It has been shown analytically that the UDE control law can be easily tuned to achieve the desired reference tracking performance via selecting an appropriate reference model. At the same time, the trade-off between disturbance attenuation and noise attenuation can be easily met through the tuning of only one parameter. A numerical example has been presented to show that, for the same degree of robustness, the UDE can offer much better disturbance rejection than an equivalent 2-DoF PID controller. The proposed UDE-based control strategy has been extensively validated with a 3 DoF quadrotor and then a 6 DoF quadrotor in real flight tests. The UDE-based controller has demonstrated much better performance than the PID-based controller, as evidenced by indexes like the integral absolute error (IAE) and the root mean squared error (RMSE).

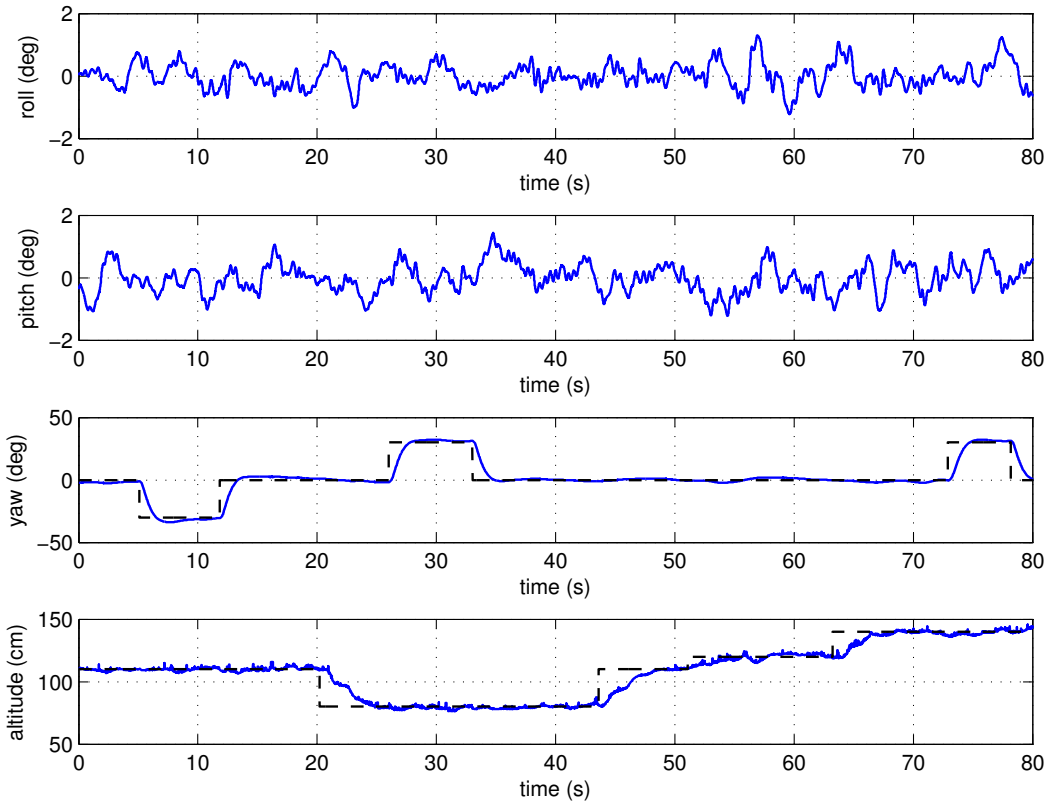

Figure 5.10: Full flight test results with the proposed UDE-based control

Chapter 6

Model Reference Adaptive Control

6.1 Introduction

Several approaches have been developed to design an adaptive controller, a controller with adjustable parameters and a mechanism for adjusting the parameters [69], to cope with changing plant dynamics. They are usually classified as direct adaptive control, where a control design method is combined with the control parameter adjust mechanism, and indirect adaptive control, where the control design method is combined with a plant parameter identification mechanism.

Applying adaptive techniques to the control of aerial unmanned vehicles has been a topic of increasing research interest, as adaptive control is a non linear control strategy well suited to approach this kind of specific plant structure. Different adaptive control design strategies have been proposed, like R,S,T two degree of freedom polynomial controller design methods, fuzzy controllers, backstepping [70], [71], [72], [73], [74].

In this chapter, a Lyapunov based direct model reference adaptive control is proposed for both plant input disturbance rejection and real-time tuning of a two degree state feedback control. The open-loop plant may be stable or unstable, but without zeros. This is a model frequently used when approximately modeling plants, suitable for unmanned aerial vehicles (UAV), [75], where many parameters are negligible and the model is reduced to a chain of integrators. The design is proved to stabilize an unstable electromechanical system, showing good simulation and experimental results.

6.2 A two degree of freedom state feedback MRAC

A model reference adaptive control strategy combines a fast regulatory loop with a slow parameter tuning loop. In this case, a continuous time two degree of freedom state feedback controller design is considered, as shown in Fig. 6.1.

Consider a SISO plant (6.1) with input u and output y

$$y(s) = G(s)u(s); \quad \rightarrow G(s) = \frac{b}{a(s)} \tag{6.1}$$

where

$$a(s) = a_1 + a_2 s + \ldots + a_n s^{n-1} + s^n \tag{6.2}$$

and $\frac{b}{a_1}$ is the static gain of the plant, if stable. For this system, the controllable canonical state space representation (A, B, C) is given by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{6.3}$$

where $x \in R^n$ and

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_1 & -a_2 & -a_3 & \cdots & -a_n \end{bmatrix}$$

$$B^T = \begin{bmatrix} 0 & 0 & \ldots & b \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix}$$

(and $D = 0$, non direct output/input coupling). Assume a plant model reference (6.4) with input u_c and output $\bar{y}$

$$\bar{y}(s) = \bar{G}(s)u_c(s); \quad \rightarrow \bar{G}(s) = \frac{\bar{b}}{\bar{a}(s)} \tag{6.4}$$

with characteristic polynomial

$$\bar{a}(s) = \bar{a}_1 + \bar{a}_2 s + \ldots + \bar{a}_n s^{n-1} + s^n \tag{6.5}$$

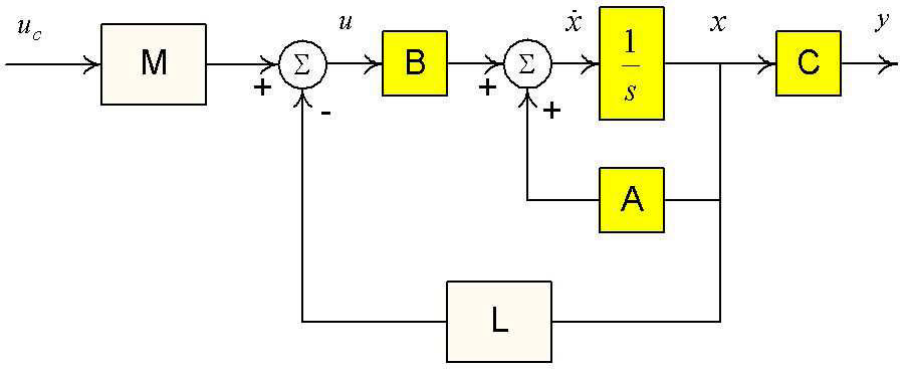

Figure 6.1: Continuous time 2DOF state feedback control.

The internal representation $(\bar{A}, \bar{B}, \bar{C})$ of the model reference (6.4), using the same structure that (6.3) is obtained with $\bar{a}_i$ instead of a_i, $\bar{b}$ instead of b and state vector $\bar{x}$.

For systems with no zeros[1] perfect tracking of the model reference state vector $\bar{x}$ can be obtained by using a two degree of freedom (2DOF) state feedback controller

$$u = Mu_c - Lx \tag{6.6}$$

where M is a scalar and $L \in R^n$ is a row vector. From (6.3) and (6.6), the state space closed loop dynamics is given by

$$\dot{x} = (A - BL)x + BMu_c = \bar{A}x + \bar{B}u_c \tag{6.7}$$

Thus, assuming full knowledge of the plant model (6.3), the ideal parameters L and M of (6.6) matching the closed loop model reference (6.4) are given by

$$\begin{aligned} L^* &= \left[\tfrac{\bar{a}_1 - a_1}{b} \quad \tfrac{\bar{a}_2 - a_2}{b} \quad \ldots \quad \tfrac{\bar{a}_n - a_n}{b}\right] \\ M^* &= \frac{\bar{b}}{b} \end{aligned} \tag{6.8}$$

yielding

$$\bar{A} = A - BL^*; \quad \bar{B} = BM^* \tag{6.9}$$

Obviously, $\bar{A}$ defines the stable closed loop dynamics to follow, whereas $\bar{b}/\bar{a}_1$ defines the static gain of the reference model.

6.2.1 Adaptation law

Note that $\bar{a}_i,\ \bar{b}$ are design parameters and $a_i,\ b$ are assumed to be known. But usually, the plant state matrix A as well as the parameter b representing the product of the actuator and output sensor gain can be only estimated. One approach to obtain model reference state perfect tracking with partial or without plant knowledge is to apply an adaptive strategy that automatically brings to zero the steady state tracking error. Here a Lyapunov based adaptive strategy is developed to tune the 2DOF state feedback controller (6.6) for unknown plant parameters. As will be shown later, the required knowledge is the plant gain sign as well as the accessibility to the plant state x, the model reference state $\bar{x}$ and the input command signal u_c.

The adaptive tuning goal is to obtain the (L, M) parameters dynamically for both, convergence to the "true" parameters and zero steady state tracking error vector, that is to obtain $L \to L^*$, $M \to M^*$ and $\lim_{t\to\infty} \|z(t)\| = 0$, $\forall t \geq 0$ from any initial condition $\|z(0)\| \neq 0$, where $z = x - \bar{x}$ is the tracking error vector. This goal can be achieved by using the Lyapunov stability theory, as follows.

[1]It is well-known [76] that the plant poles can be assigned by state feedback but there is no action on the zeros position.

Let $V(z, K)$ be a Lyapunov candidate function,

$$V(z,K) = \frac{1}{2}\left(\gamma z^T P z + (K-K^*)^T(K-K^*)\right) \tag{6.10}$$

where z is the previously defined tracking error vector, $K = [L\ M]^T$ is the parameter vector of dimension $(n+1)\times 1$, and P is a symmetric positive definite matrix ($P > 0$). Clearly, the continuously differentiable candidate function (6.10) meets the Lyapunov necessary conditions: $V(z,K) > 0 \quad \forall z \neq 0, \forall K \neq K^* = [L^*\ M^*]^T$ and $V(0,K^*) = 0$. In addition to these conditions, Lyapunov global asymptotically stability is assured if and only if $\dot{V}(z,K) < 0$. In that case $z = 0$ and K^* will be globally asymptotically stable equilibrium points.

The tracking error vector $z(t)$ dynamics can be expressed as (6.11)

$$\dot{z} = \dot{x} - \dot{[]}x = (A - BL)x - \bar{A}\bar{x} + (BM - \bar{B})u_c \tag{6.11}$$

Adding and subtracting the term BL^*x and using (6.9), this equation can be arranged as

$$\begin{aligned} \dot{z} &= \bar{A}z + B[-(L-L^*)x + (M-M^*)u_c] \\ &= \bar{A}z + B[K-K^*]^T \begin{bmatrix} -x \\ u_c \end{bmatrix} \end{aligned} \tag{6.12}$$

Due to the special form of B, it can be simplified as

$$\dot{z} = \bar{A}z + \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ & b[K-K^*]^T & \end{bmatrix} \begin{bmatrix} -x \\ u_c \end{bmatrix} \tag{6.13}$$

which is equivalent to

$$\begin{aligned} \dot{z} &= \bar{A}z + b \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ & \begin{bmatrix} -x^T & u_c \end{bmatrix} & \end{bmatrix} [K-K^*] \\ &= \bar{A}z + \Gamma[K-K^*] \end{aligned} \tag{6.14}$$

where $\Gamma = \Gamma(x, u_c)$ is a matrix of dimension $n \times (n+1)$

$$\Gamma = b \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ & \begin{bmatrix} -x^T & u_c \end{bmatrix} & \end{bmatrix} \tag{6.15}$$

with a gain being adapted in the estimation process $b = \bar{b}/M^*$.

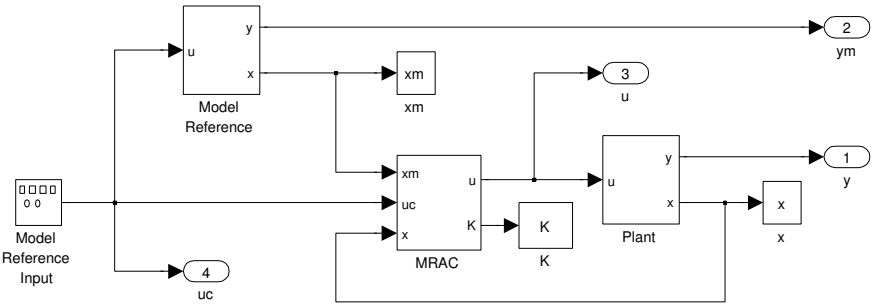

Figure 6.2: 2DOF state space MRAC scheme.

Differentiating (6.10) and combining it with the error vector dynamics (6.14) yields to

$$\dot{V}(z,K) = \frac{1}{2}\gamma z^T(\bar{A}^T P + P\bar{A})z + (K - K^*)^T(\dot{K} + \gamma\Gamma^T P z) \tag{6.16}$$

From this result, two conditions must be fulfilled to obtain global asymptotic stability

i) $\bar{A}^T P + P\bar{A} = -Q$, for any given $Q > 0$

ii) $\dot{K} = -\gamma\Gamma^T P z$, for any given $\gamma > 0$ ($b > 0$)

Condition i), known as the Lyapunov equation, is always satisfied for a stable model reference, and condition ii) represents the parameter adaptation law which requires the recursive computation of the non square matrix Γ. With both conditions $\dot{V}(z,K) = -\frac{1}{2}\gamma z^T Q z < 0$ is obtained, where γ defines the convergence velocity in a trade-off with the control effort.

The proposed adaptive strategy is shown in Fig. 6.2

6.2.2 2DOF MRAC simulation examples

Two examples are used to illustrate the performance of the proposed adaptive scheme. The first one considers an unstable third order plant and the second one is a double integrator SISO plant with input disturbances.

2DOF adaptive control of an unstable third order plant

Let us consider an unstable third order plant with transfer function

$$G(s) = \frac{1}{s^3 + 2s^2 + s + 10}$$

and assume the desired trajectory be represented by

$$\bar{G}(s) = \frac{8}{(s+2)^3}$$

That is

$$\left.\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -10 & -1 & -2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x \end{aligned}\right\} \tag{6.17}$$

$$\left.\begin{aligned}\dot{\bar{x}} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -8 & -12 & -6 \end{bmatrix} \bar{x} + \begin{bmatrix} 0 \\ 0 \\ 8 \end{bmatrix} u_c \\ \bar{y} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \bar{x} \end{aligned}\right\} \tag{6.18}$$

Using the design method for the 2DOF state feedback controller (6.8), yields

$$K^* = \begin{bmatrix} L^* & M^* \end{bmatrix}^T = \left[\begin{bmatrix} \frac{\bar{a}_1 - a_1}{b} & \frac{\bar{a}_2 - a_2}{b} & \frac{\bar{a}_3 - a_3}{b} \end{bmatrix} \quad \frac{\bar{b}}{b} \right]^T \tag{6.19}$$

That is, the elements of the nominal control law are $L_1^* = -2;\ L_2^* = 11;\ L_3^* = 4; M^* = 8;$.

Assuming full plant state accessibility, the tracking error vector dynamics can be obtained as $\dot{z} = \bar{A}z + \Gamma\left(K - K^*\right)$, where

$$\Gamma = b \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -x_1 & -x_2 & -x_3 & u_c \end{bmatrix} \tag{6.20}$$

As the state matrix $\bar{A}$ is stable, then $\exists P > 0$ solution to $\bar{A}^T P + P\bar{A} = -Q$ for any given $Q > 0$, for which the parameter adaptation law is obtained as $\dot{K} = -\gamma \Gamma^T P z$. Notice that the plant gain b in Γ should be estimated ($b = \bar{b}/M^*$) but it can be included into the selected adaptation gain γ. In this way the adaptation law just requires to know the sign of b, accessibility to the plant state x, the model reference state $\bar{x}$ and the input command signal u_c. The output error result is shown in Fig. 6.3 for $\bar{b} = 8$, $\bar{a}_1 = 8$, $\bar{a}_2 = 12$, $\bar{a}_3 = 6$ and $\gamma = 25$. The Lyapunov equation has been solved for $Q = I_{3\times 3}$, yielding to

$$P = \begin{bmatrix} 1.9063 & 1.2344 & 0.0625 \\ 1.2344 & 2.0938 & 0.1445 \\ 0.0625 & 0.1445 & 0.1074 \end{bmatrix}$$

The parameter adaptive law converges to the "true" values K^* with almost zero tracking error vector from $t = 100[s]$ onwards as shown in Fig. 6.4.

2DOF adaptive control of a disturbed double integrator

Consider a double integrator plant with an input disturbance and the proposed stable model reference (6.18) with $b, \bar{b}, \bar{a}_1, \bar{a}_2 > 0$.

$$\left.\begin{aligned}\dot{\bar{x}} &= \begin{bmatrix} 0 & 1 \\ -\bar{a}_1 & -\bar{a}_2 \end{bmatrix} \bar{x} + \begin{bmatrix} 0 \\ \bar{b} \end{bmatrix} u_c \\ \bar{y} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \bar{x} \end{aligned}\right\} \tag{6.21}$$

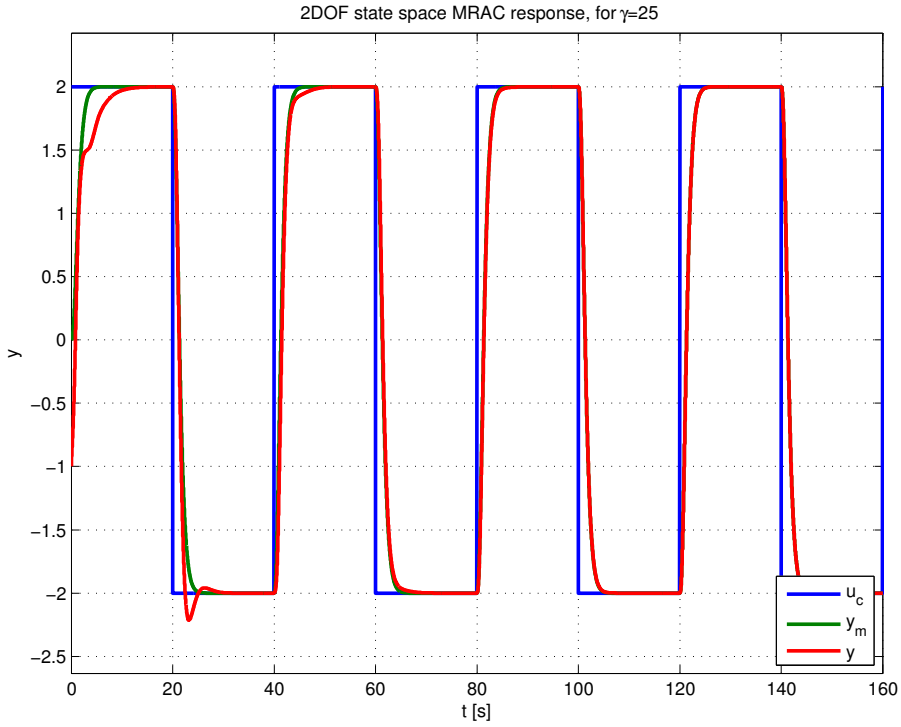

Figure 6.3: 2DOF state space MRAC response.

Following the previous approach, the Lyapunov equation is solved for $Q = I_{2\times 2}$, yielding to

$$P = \begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

The output error result is shown in Fig. 6.5 for $\bar{b} = 1$, $\bar{a}_1 = 1$, $\bar{a}_2 = 2$ and $\gamma = 2.5$. The parameter adaptive law converges to the "true" values $K^* = [2\ 4\ 2]^T$ with zero tracking error vector from $t = 35[s]$ onwards as shown in Fig. 6.6.

Now, a periodic disturbance d_i signal is added to the plant input u almost without affecting the output error, as shown in Fig. 6.7. In this case, the plant state tends to the model reference state, but the parameters (L, M) are changed to compensate the periodic input disturbance, as shown in Fig. 6.8.

6.3 Experimental results

The proposed adaptive schema is now experimentally tested on two axis of the previously depicted in Figure 2.2.

A PC running Linux-RT, a soft RTOS distributed with a GNU GPLv2 license is provided to implement the control algorithms, on top of an Ubuntu installation. The communications between the PC and the quadrotor platform were made with a PMC I/O target. Linux RT

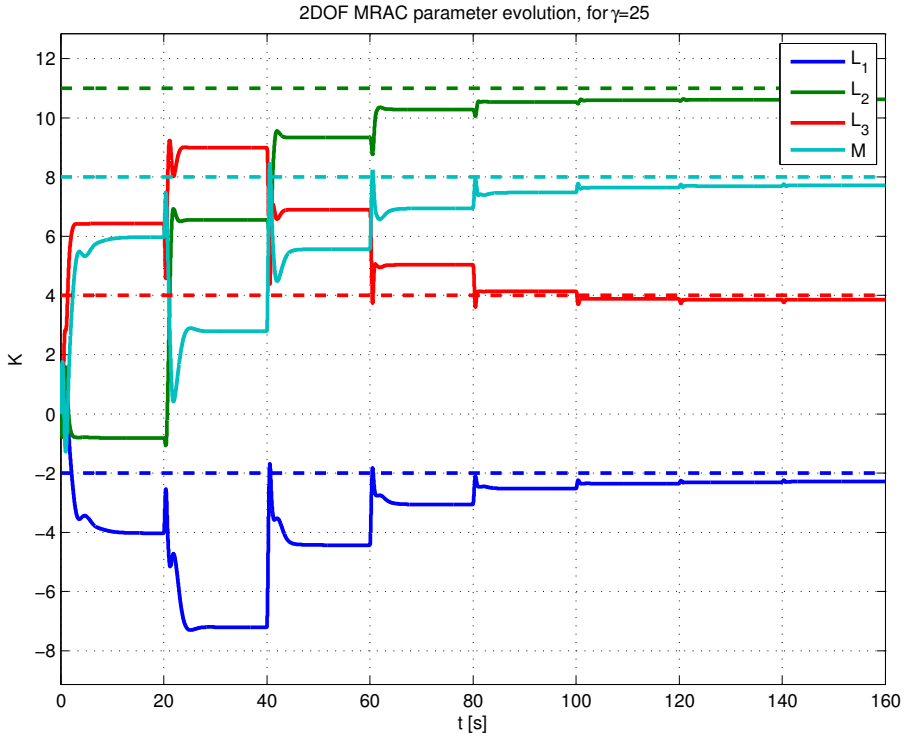

Figure 6.4: Parameter error evolution.

is a Linux O.S. with a patch whose objective is to minimize the amount of kernel code that is nonpreemptible. In this way, faster sampling periods with more reliable real-time guarantees (reduced sampling period jitter) can be implemented. The controller has been implemented in C++, using the newmat matrix library available in Ubuntu repositories. With the above operating system and an Intel I3 processor at 3.3 GHz, the computing power is ample enough to execute the MRAC adaptive control loop with a sampling period of 5 ms.

The model reference adaptive control scheme is applied to the roll and pitch axis, keeping the yaw angle constant. In this way, the MIMO model (5.1) linearized around the equilibrium point is reduced to two decoupled double integrator subsystems, one for each axis [75]:

$$\ddot{\phi} = u_1$$
$$\ddot{\theta} = u_2$$

The same reference model is used in both axis, which is defined using $\bar{a}_1 = 4.34$, $\bar{a}_2 = 2.3$ and $\bar{b} = 4.34$. Solving $\bar{A}^T P + P\bar{A} = -Q$ for $Q = I_{2\times 2}$ yields to

$$P = \begin{bmatrix} 1.4258 & 0.1152 \\ 0.1152 & 0.2675 \end{bmatrix} \tag{6.22}$$

This P matrix is then included in the adaptive control law $\dot{K} = -\gamma \Gamma^T P z$, for Γ given in (6.15). The experimental response to a square wave reference input u_c and $\gamma = 0.1$ is shown

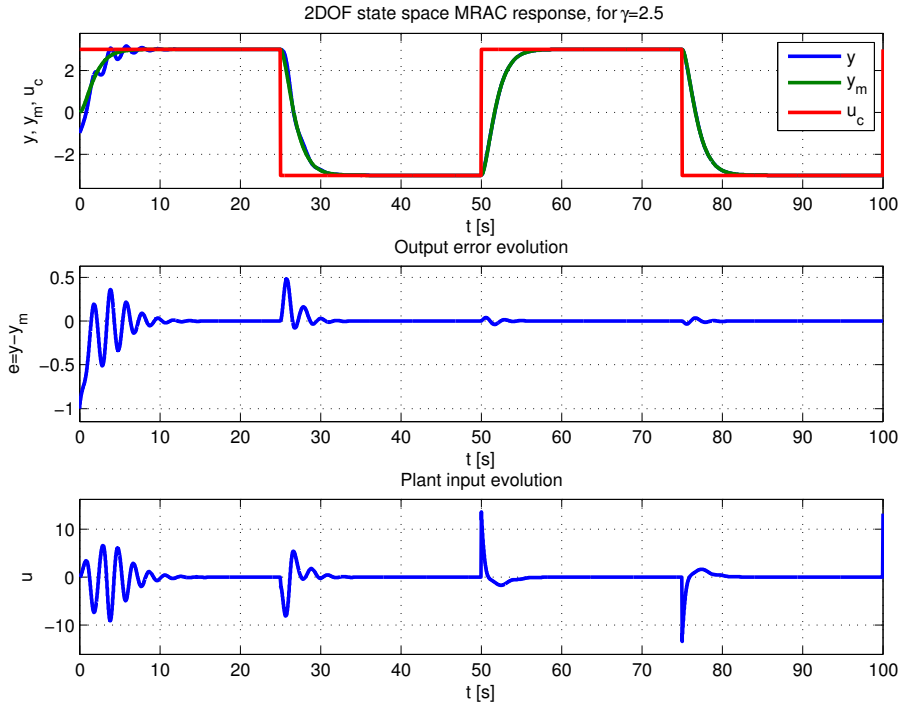

Figure 6.5: 2DOF state space MRAC response.

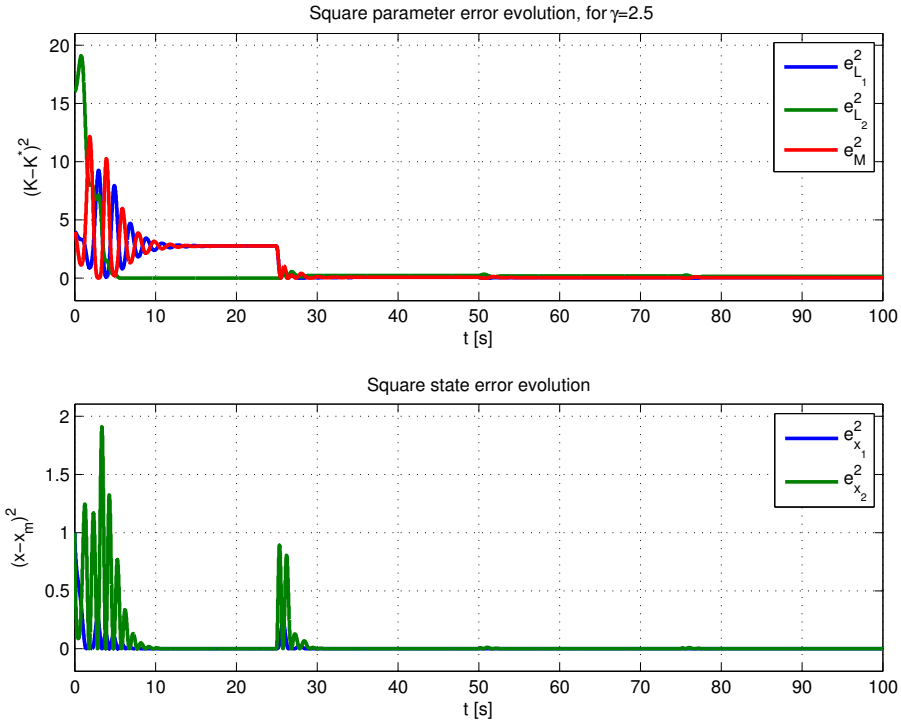

Figure 6.6: Parameter and state error evolution.

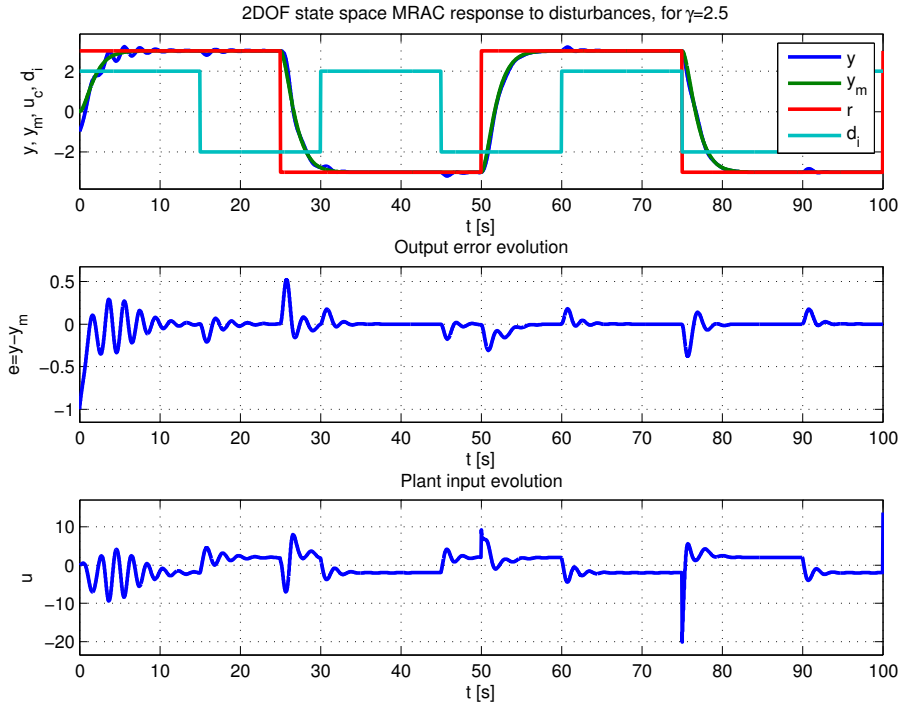

Figure 6.7: 2DOF state space MRAC input disturbance response.

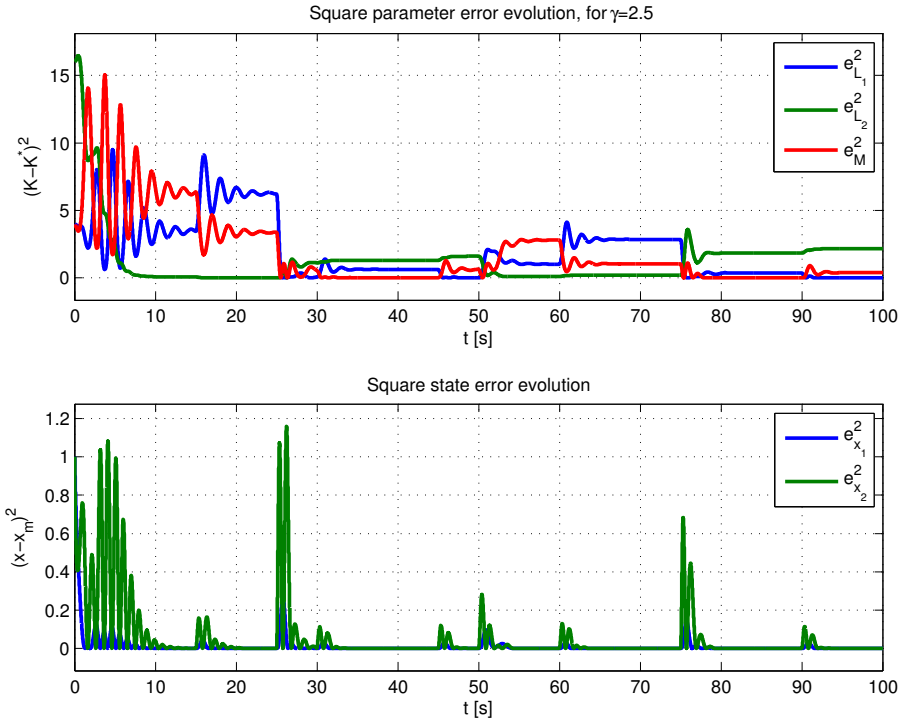

Figure 6.8: Parameter and state error evolution.

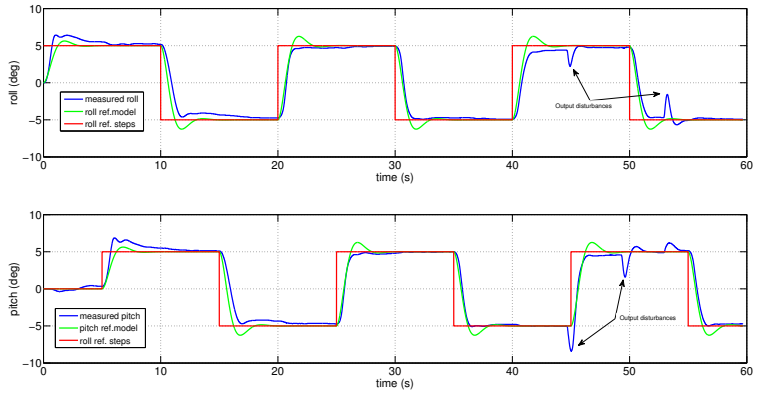

Figure 6.9: ML MRAC roll and pitch angles

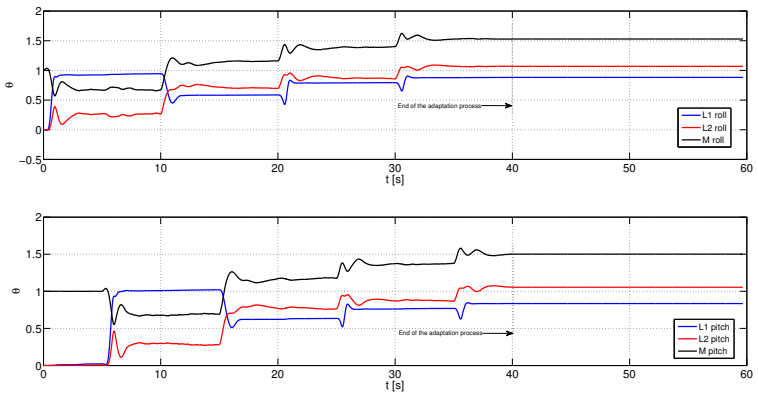

Figure 6.10: ML MRAC parameters

in Fig. 6.9. The measured output is close to the reference model although they do not match perfectly. However, the system is able to follow the references fairly well. The evolution of the adaptation parameters is shown in Fig. 6.10. Notice that the parameters convergence is slower than in the simulation examples (here, $\gamma = 0.1$), but an attempt to increase the convergence rate will saturate the quadrotor actuators. The adaptation algorithm was active only during the first two step references. After that moment, the parameters were kept constant. Notice that all the parameters were initialized with a null value (L) and unity (M), i.e., the system starts from an open-loop situation.

6.4 Conclusions

The MRAC control strategy has been applied for a special kind of systems, those without transmission zeros. This kind of models is frequent in practical applications and it is ob-

tained when linearized electromechanical systems are considered, as it has been illustrated for a lab quadrotor. An adaptive state feedback control with adjusted input/output gain has been derived, its convergence being proved and experimentally illustrated when applied to the quadrotor.

The adaptation mechanism allows dealing with uncertain plant models and the algorithm provides the estimated controller parameters. They converge to the ideal ones, in simulation. The convergence rate can be tuned by a design parameter (γ), requiring a tradeoff between convergence speed and control action. In practical applications, the control action amplitude is limited and thus, the parameters convergence is slow. If this speed is increased, the actuators saturate and the stability is not guaranteed.

Next step will be to extend the results to a more general class of systems and to investigate the implementation constraints.

Conclusions and Future Work

Although quadrotors are available commercial products nowadays, they remain an open topic for the research community. The computational efficiency of the algorithms is an strong requirement that restricts the number of techniques which can be implemented in real applications.

Despite of that, in this thesis some improvements in both estimation and control have been reported. Regarding the attitude estimation, an observer-predictor scheme was used to compensate the small delays present in the data acquisition and processing. A smaller delay allows for higher gains in the controller, which results in a higher performance. On the other hand, a novel control approach based on the uncertainty and disturbance estimator (UDE), has been successfully applied in real flights.

Future work will consist on exploiting the advantages of both techniques, in order to obtain a more robust control with higher performances. It could be also interesting to explore further the capabilities of the predictor as a technique for improving the performance of a given controller.

Developing new control strategies, focusing on robustness and disturbance rejection, is also a matter of interest. The UDE-based has shown promising results and future work following the same philosophy may lead to interesting results. The adaptive controller performed also fairly well in the laboratory platform, and its application in real flight is still pending.

Bibliography

[1] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *IEEE Control Systems Magazine*, vol. 25, no. 6, pp. 45–55, 2005.

[2] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 2247–2252, IEEE, 2005.

[3] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An integral predictive/nonlinear h infinity control structure for a quadrotor helicopter," *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.

[4] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*, pp. 153–158, IEEE, 2007.

[5] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *Robotics & Automation Magazine, IEEE*, vol. 19, pp. 20–32, 2012.

[6] H. Chao, Y. Cao, and Y. Chen, "Autopilots for small unmanned aerial vehicles: a survey," *International Journal of Control, Automation and Systems*, vol. 8, no. 1, pp. 36–44, 2010.

[7] L. Mejias, J. F. Correa, I. Mondragón, and P. Campoy, "Colibri: a vision-guided uav for surveillance and visual inspection," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2760–2761, IEEE, 2007.

[8] D. W. Casbeer, R. Beard, T. McLain, S.-M. Li, and R. K. Mehra, "Forest fire monitoring with multiple small uavs," in *American Control Conference, 2005. Proceedings of the 2005*, pp. 3530–3535, IEEE, 2005.

[9] H. Chao, M. Baumann, A. Jensen, Y. Chen, Y. Cao, W. Ren, and M. McKee, "Band-reconfigurable multi-uav-based cooperative remote sensing for real-time water management and distributed irrigation control," in *IFAC World Congress, Seoul, Korea*, 2008.

[10] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1649–1654, IEEE, 2010.

[11] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Wiley. com, 2006.

[12] S.-G. Kim, J. L. Crassidis, Y. Cheng, A. M. Fosbury, and J. L. Junkins, "Kalman filtering for relative spacecraft attitude and position estimation," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 133–143, 2007.

[13] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1203–1218, 2008.

[14] P. C. Garcia, R. Lozano, and A. E. Dzul, *Modelling and control of mini-flying machines*. Springer, 2006.

[15] H. Liu, G. Lu, and Y. Zhong, "Robust LQR attitude control of a 3-DOF laboratory helicopter for aggressive maneuvers," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 10, pp. 4627–4636, 2013.

[16] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 46–56, 2012.

[17] A. Franchi, C. Secchi, M. Ryll, H. Bulthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 57–68, 2012.

[18] B. Zheng and Y. Zhong, "Robust attitude regulation of a 3-DOF helicopter benchmark: theory and experiments," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 2, pp. 660–670, 2011.

[19] R. Sanz, L. Rodenas, P. Garcia, and P. Albertos, "Attitude estimaton using low-cost sensors: A comparative analysis," in *WCICA 2014, World Congress on Intelligent Control and Automation, June 29 – July 4, Shenyang, China*, 2014.

[20] R. Sanz, L. Rodenas, P. Garcia, and P. Castillo, "Improving attitude estimation using inertial sensors for quadrotor control systems," in *ICUAS 2014, The 2014 International Conference on Unmanned Aircraft Systems, May 27-30, Orlando, Florida, USA*, 2014.

[21] R. Sanz, L. Rodenas, P. Garcia, and P. Albertos, "Time-delay compensation using inertial measurement sensors for quadrotor control systems," in *FUSION 2014, International Conference on Information Fusion, 2014, July 7-10, Salamanca, Spain*, 2014.

[22] P. Garcia, R. Sanz, Q.-C. Zhong, and P. Albertos, "Robust control of quadrotors based on the uncertainty and disturbance estimator," *Control Systems Technology, IEEE Transactions on*, Submitted.

[23] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[24] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*, vol. 24. CRC press, 2011.

[25] P. Martin and E. Salaun, "The true role of accelerometer feedback in quadrotor control," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1623–1629, IEEE, 2010.

[26] L. Rodenas, R. Sanz, P. Albiol, A. Castillo, D. Verdu, and P. Garcia, "Plataforma para la implementación y validación en algoritmos de control de tiempo real en mini-helicópteros de varios rotores," in *Jornadas de Automática, 2014, Valencia, Spain*, 2014.

[27] H. Chao, C. Coopmans, L. Di, and Y. Chen, "A comparative evaluation of low-cost imus for unmanned autonomous systems," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*, pp. 211–216, IEEE, 2010.

[28] H. Rehbinder and X. Hu, "Drift-free attitude estimation for accelerated rigid bodies," *Automatica*, vol. 40, no. 4, pp. 653–659, 2004.

[29] C. Liu, Z. Zhou, and X. Fu, "Attitude determination for mavs using a kalman filter," *Tsinghua Science & Technology*, vol. 13, no. 5, pp. 593–597, 2008.

[30] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[31] J. Guzmán, P. García, T. Hägglund, S. Dormido, P. Albertos, and M. Berenguel, "Interactive tool for analysis of time-delay systems with dead-time compensators," *Control Engineering Practice*, vol. 16, no. 7, pp. 824–835, 2008.

[32] E. Kaszkurewicz and A. Bhaya, "Discrete-time state estimation with two counters and measurement delay," in *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, vol. 2, pp. 1472–1476, IEEE, 1996.

[33] T. D. Larsen, N. A. Andersen, O. Ravn, and N. K. Poulsen, "Incorporation of time delayed measurements in a discrete-time kalman filter," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 4, pp. 3972–3977, IEEE, 1998.

[34] Q. Ge, T. Xu, X. Feng, and C. Wen, "Universal delayed kalman filter with measurement weighted summation for the linear time invariant system," *Chinese Journal of Electronics*, vol. 20, no. 1, pp. 67–72, 2011.

[35] Z. J. Palmor, "Time delay compensation- smith predictor and its modifications, in the control handbook, (W.S. Levine, Eds)," *CRSC Press*, pp. 224–237, 1996.

[36] J. Richard, "Time-delay systems: an overview of some recent advances and open problems," *automatica*, vol. 39, no. 10, pp. 1667–1694, 2003.

[37] J. Normey-Rico and E. F. Camacho, "Dead-time compensators: A survey," *Control Engineering Practice*, vol. 16, no. 4, pp. 407–428, 2008.

[38] R. Lozano, P. Castillo, P. Garcia, and A. Dzul, "Robust prediction-based control for unstable delay systems: Application to the yaw control of a mini-helicopter," *Automatica*, vol. 40, no. 4, pp. 603–612, 2004.

[39] P. Garcia, P. Castillo, R. Lozano, and P. Albertos, "Robustness with respect to delay uncertainties of a predictor-observer based discrete-time controller," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC)*, pp. 199–204, IEEE, 2006.

[40] P. Albertos and P. García, "Predictor–observer-based control of systems with multiple input/output delays," *Journal of Process Control*, vol. 22, no. 7, pp. 1350–1357, 2012.

[41] A. Gonzalez, A. Sala, P. Garcia, and P. Albertos, "Robustness analysis of discrete predictor-based controllers for input-delay systems," *International Journal of System Science*, no. In Press, 2011.

[42] A. Cuenca, P. García, P. Albertos, and J. Salt, "A non-uniform predictor-observer for a networked control system," *International Journal of Control, Automation and Systems*, vol. 9, no. 6, pp. 1194–1202, 2011.

[43] A. González, "Robust stabilization of linear discrete-time systems with time-varying input delay," *Automatica*, vol. 49, no. 9, pp. 2919–2922, 2013.

[44] I. Karafyllis and M. Krstic, "Robust predictor feedback for discrete-time systems with input delays," *International Journal of Control*, vol. 86, no. 9, pp. 1652–1663, 2013.

[45] J. Normey-Rico and E. Camacho, *Control of dead-time processes*. Springer, 2007.

[46] A. Gonzalez, P. Garcia, P. Albertos, P. Castillo, and R. Lozano, "Robustness of a discrete-time predictor-based controller for time-varying measurement delay," *Control Engineering Practice*, vol. 20, no. 2, pp. 102–110, 2012.

[47] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2451–2456, IEEE, 2004.

[48] H. Lim, J. Park, D. Lee, and H. J. Kim, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 33–45, 2012.

[49] R. Lozano, P. Castillo, P. Garcia, and A. Dzul, "Robust prediction-based control for unstable delay systems: application to the yaw control of a mini-helicopter," *Automatica*, vol. 40, no. 4, pp. 603–612, 2004.

[50] J. Escareno, S. Salazar-Cruz, and R. Lozano, "Embedded control of a four-rotor UAV," in *American Control Conference, 2006*, pp. 6–pp, IEEE, 2006.

[51] F. Kendoul, D. Lara, I. Fantoni, and R. Lozano, "Real-time nonlinear embedded control for an autonomous quadrotor helicopter," *Journal of guidance, control, and dynamics*, vol. 30, no. 4, pp. 1049–1061, 2007.

[52] A. Sanchez, P. Garcia, P. C. Garcia, and R. Lozano, "Simple real-time stabilization of vertical takeoff and landing aircraft with bounded signals," *Journal of guidance, control, and dynamics*, vol. 31, no. 4, pp. 1166–1176, 2008.

[53] I. González, S. Salazar, J. Torres, R. Lozano, and H. Romero, "Real-time attitude stabilization of a mini-UAV quad-rotor using motor speed feedback," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 93–106, 2013.

[54] R. Lozano, *Unmanned aerial vehicles: Embedded control*. John Wiley & Sons, 2013.

[55] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proc. of the AIAA Guidance, Navigation, and Control Conference*, vol. 2, 2007.

[56] S. L. Waslander and C. Wang, "Wind disturbance estimation and rejection for quadrotor position control," in *AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference, Seattle, WA*, 2009.

[57] Q.-C. Zhong and D. Rees, "Control of uncertain lti systems based on an uncertainty and disturbance estimator," *Journal of dynamic systems, measurement, and control*, vol. 126, no. 4, pp. 905–910, 2004.

[58] S. Talole and S. Phadke, "Robust input–output linearisation using uncertainty and disturbance estimation," *International Journal of Control*, vol. 82, no. 10, pp. 1794–1803, 2009.

[59] A. Kuperman and Q.-C. Zhong, "Control of uncertain linear systems with a state delay based on an uncertainty and disturbance estimator," in *Proceedings of the Sixth IFAC Symposium on Robust Control Design*, pp. 279–283, 2009.

[60] A. Kuperman and Q.-C. Zhong, "Robust control of uncertain nonlinear systems with state delays based on an uncertainty and disturbance estimator," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 1, pp. 79–92, 2011.

[61] S. Talole and S. Phadke, "Model following sliding mode control based on uncertainty and disturbance estimator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 130, no. 3, p. 034501, 2008.

[62] V. Deshpande and S. Phadke, "Control of uncertain nonlinear systems using an uncertainty and disturbance estimator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 2, p. 024501, 2012.

[63] B. Ren and Q.-C. Zhong, "UDE-based robust control for a class of non-affine nonlinear systems," in *ASME 2013 Dynamic Systems and Control Conference*, pp. V003T34A004–V003T34A004, American Society of Mechanical Engineers, 2013.

[64] J. P. Kolhe, M. Shaheed, T. Chandar, and S. Talole, "Robust control of robot manipulators based on uncertainty and disturbance estimation," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 1, pp. 104–122, 2013.

[65] A. Kuperman, "UDE-based robust voltage control of DC-DC power converters," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2013 5th International Congress on*, pp. 140–145, IEEE, 2013.

[66] Q.-C. Zhong, A. Kuperman, and R. Stobart, "Design of UDE-based controllers from their two-degree-of-freedom nature," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 17, pp. 1994–2008, 2011.

[67] K. Youcef-Toumi and O. Ito, "A time delay controller for systems with unknown dynamics," *Journal of dynamic systems, measurement, and control*, vol. 112, no. 1, pp. 133–142, 1990.

[68] G. Sanahuja, P. Castillo, and A. Sanchez, "Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 10, pp. 1129–1139, 2010.

[69] K. J. Åström and B. Wittenmark, *Adaptive Control*. Massachusetts: Addison-Wesley, 2nd ed., 1995.

[70] Y. Morel and A. Leonessa, "Direct adaptive tracking control of quadrotor aerial vehicles," in *ASME 2006 Dynamic Systems and Control Conference,*, (Chicago, Illinois, USA), pp. 155–161, 2006.

[71] M. Mohammadi and A. M. Shahri, "Adaptive nonlinear stabilization control for a quadrotor UAV: Theory, simulation and experimentation," *Journal of Intelligent Robot Systems*, vol. 72, pp. 105–122, 2013.

[72] C. Coza and C. Macnab, "A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization," in *NAFIPS 2006 Annual meeting of the North American Fuzzy Information Processing Society*, (Montreal, Que. Canada), pp. 454–458, June 2006.

[73] K. Tanaka, H. Ohtake, and H. O. Wang, "A practical design approach to stabilization of a 3-DOF rc helicopter," *IEEE Transactions on Control Systems Technology*, vol. 12, 2, pp. 315–325, 2004.

[74] C. Nicol, C. Macnab, and A. Ramirez-Serrano, "Robust adaptive control of a quadrotor helicopter," *Mechatronics*, vol. 21, no. 6, pp. 927–938, 2011.

[75] P. Castillo, R. Lozano, and A. E. Dzul, *Modelling and Control of Mini-flying Machines*. Springer: Berlin, 2005.

[76] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems. Theory and Design*. Upper Saddle River, NJ: Prentice-Hall, 3rd ed., 1997.