Mobile orientation tracking with non-linear Kalman filters

Samu Kallio

School of Electrical Engineering

Bachelor's thesis

Espoo 11.5.2018

Supervisor

D.Sc. (Tech). Pekka Forsman

Advisors

M.Sc. Toni Karvonen

M.Sc. (Tech). Filip Tronarp

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the bachelor's thesis

Author Samu Kallio

Title Mobile orientation tracking with non-linear Kalman filters

Degree programme Automation and Information Technology

Major Automation and Control Engineering **Code of major** ELEC3014

Supervisor D.Sc. (Tech). Pekka Forsman

Advisors M.Sc. Toni Karvonen, M.Sc. (Tech). Filip Tronarp

Date 11.5.2018 **Number of pages** 38+6 **Language** English

Abstract
Mobile orientation tracking refers to the methods and technologies used in estimating the orientation of a vehicle or a mobile device. Accurate orientation estimation is a necessity in many technical applications. This thesis investigates two approaches to orientation tracking based on non-linear Kalman filters. The first approach, known as the multiplicative extended Kalman filter, is a quaternion-based algorithm that is widely used for orientation tracking. The second approach is a vector-based algorithm that is partly an original result of this thesis. Implementations of both approaches are tested using purpose-built sensor hardware. Testing suggests that both approaches can be used to achieve orientation estimates of comparable quality.

Keywords Kalman filter, orientation tracking, sensor fusion

Aalto University
School of Electrical Engineering

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Tekniikan kandidaatintyön tiivistelmä

Tekijä Samu Kallio

Työn nimi Mobiililaitteiden asennon seuranta epälineaarisilla Kalman-suotimilla

Koulutusohjelma Automaatio- ja informaatioteknologia

Pääaine Automaatio- ja systeemitekniikka **Pääaineen koodi** ELEC3014

Vastuuopettaja TkT Pekka Forsman

Työn ohjaajat FM Toni Karvonen, M.Sc. (Tech). Filip Tronarp

Päivämäärä 11.5.2018 **Sivumäärä** 38+6 **Kieli** Englanti

Tiivistelmä

Mobiililaitteiden asennon seurannalla tarkoitetaan menetelmiä, joilla liikkeessä olevan laitteen asentoa voidaan estimoida reaaliaikaisesti. Asennon seurannalla on runsaasti käytännön sovelluksia. Käyttökohteita ovat muun muassa lentokoneet, avaruusraketit, satelliitit, teollisuuden robotiikka, autotekniikka sekä viihde-elektroniikka kuten älypuhelimet, tabletit ja virtuaalitodellisuuslasit. Asennon seurantaan käytettävät menetelmät riippuvat käyttökohteesta. Tässä työssä keskityttiin erityisesti maan pinnalla käytettävien laitteiden asennon seurantaan kiihtyvyysanturin, gyroskoopin sekä magnetometrin eli kompassin avulla. Moderni mikrosysteemitekniikka mahdollistaa näiden anturien valmistuksen alhaisin kustannuksin ja hyvin pienessä koossa, minkä vuoksi niiden käyttö on viime vuosina yleistynyt erityisesti viihde-elektroniikassa.

Halpojen anturien käyttö asennon seurannassa korostaa tehokkaiden algoritmien tärkeyttä. Yksi mahdollinen lähestymistapa on sensorifuusio, jossa usean eri anturin mittausdataa hyödynnetään samanaikaisesti, jolloin yksittäisten anturien häiriöitä saadaan kompensoitua. Tässä työssä perehdyttiin asennon seurantaan kahdella epälineaariseen Kalman-suotimeen pohjautuvalla sensorifuusiomenetelmällä. Kalman-suodin on rekursiivinen stokastinen tilaestimaattori, joka kykenee estimoimaan dynaamisen järjestelmän tilaa siitä saatujen häiriöisten mittausten perusteella. Alkuperäinen Kalman-suodin soveltuu vain lineaaristen järjestelmien tilaestimointiin. Suotimen teoriaa on myöhemmin laajennettu kattamaan myös epälineaaristen järjestelmien tilaestimointia. Tässä työssä esiintyvien epälineaaristen tilamallien tilaestimointiin sovellettiin erityisesti suotimen vanhinta ja tunnetuinta epälineaarista laajennusta, niin kutsuttua laajennettua Kalman-suodinta (engl. extended Kalman filter, EKF).

Ensimmäinen työssä tarkastelluista menetelmistä on kvaterniopohjainen niin kutsuttu multiplikatiivinen laajennettu Kalman-suodin (engl. multiplicative extended Kalman filter, MEKF), jota on aikaisemmin sovellettu muun muassa satelliittien asennon seurannassa. Toinen menetelmä on vektoripohjainen algoritmi (tässä työssä nimellä vector Kalman filter, VKF), joka on osittain tämän työn alkuperäinen tulos. VKF-algoritmin pääasiallinen etu MEKF-algoritmiin verrattuna on helpompi ymmärettävyys ja toteutettavuus. Työn tavoitteena oli varmistaa sekä verrata menetelmien toimivuus käytännössä. Käytetyt sensorifuusiomallit ovat tarkoituksella pelkistettyjä, eivätkä sellaisenaan sovellu tuotantokäyttöön.

Työssä tehtiin lyhyt katsaus Kalman-suotimen teoriaan ja johdettiin sekä lineaarinen että laajennettu Kalman-suodin Bayesiläistä lähestymistapaa soveltaen. Asennon seurantaa varten johdettiin edellä mainittujen menetelmien mukaiset sensorifuusiomallit, joiden estimointiin sovellettiin laajennettua Kalman-suodinta. Tuloksena saadut algoritmit toteutettiin ohjelmallisesti, ja niiden toimintaa testattiin kokeellisesti tarkoitusta varten rakennetulla anturilaitteella. Algoritmien suorituskykyä arvioitiin laadullisesti. Kokeellisesti saadut tulokset viittaavat siihen, että molemmilla menetelmillä päästään laadullisesti yhtä hyvään lopputulokseen. Algoritmeissa käytetyt anturien kohinamallit ovat samat molemmissa menetelmissä, mutta algoritmien rakenteellisista eroista johtuen kohinamallien parametrit eivät ole keskenään vertailukelpoisia. Molemmat algoritmit täytyy siten virittää erikseen halutun lopputuloksen saavuttamiseksi.

Avainsanat Kalman-suodin, asennon seuranta, sensorifuusio

Contents

Symbols and abbreviations

Operators and symbols

$\mathbf{x}$	Column vector
$\mathbf{x}^\mathrm{T}$	Row vector
$\mathbf{x}^\mathrm{b}$	Representation of vector $\mathbf{x}$ in reference frame b
$\mathbf{I}$	Identity matrix
$\mathbf{A}^\mathrm{T}$	Matrix transpose
$\mathbf{A}^{-1}$	Matrix inverse
$\mathbf{A}^\mathrm{bn}$	Matrix of transform from reference frame n to b
$\mathrm{E}[x]$	Expected value (mean) of x
$\mathbf{x} \triangleq \mathbf{y}$	$\mathbf{x}$ is defined as $\mathbf{y}$
$\mathbf{x} \simeq \mathbf{y}$	$\mathbf{x}$ is assumed approximately equal to $\mathbf{y}$
$\mathbf{x} \sim p(\mathbf{x})$	$\mathbf{x}$ is distributed as $p(\mathbf{x})$
$p(\mathbf{x})$	Probability density function (PDF) of random vector $\mathbf{x}$
$p(\mathbf{x} \mid \mathbf{y})$	Conditional PDF of $\mathbf{x}$ given $\mathbf{y}$
$p_k(\mathbf{x})$	Conditional PDF of $\mathbf{x}$ given $\mathbf{y}_1, \ldots, \mathbf{y}_k$
$\mathcal{N}(\hat{\mathbf{x}}, \mathbf{P})$	Gaussian PDF with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}$
$\mathcal{N}(\mathbf{x} \mid \hat{\mathbf{x}}, \mathbf{P})$	Gaussian PDF with explicit argument $\mathbf{x}$
$\mathbf{p} \odot \mathbf{q}$	Hamilton quaternion product of $\mathbf{p}$ and $\mathbf{q}$
$\mathbf{p} \otimes \mathbf{q}$	Opposite quaternion product of $\mathbf{p}$ and $\mathbf{q}$
$[\mathbf{q}\,\odot]$	Matrix of quaternion product $\mathbf{q} \odot \bullet$
$[\mathbf{q}\,\otimes]$	Matrix of quaternion product $\mathbf{q} \otimes \bullet$
$[\mathbf{x}\,\times]$	Matrix of cross product $\mathbf{x} \times \bullet$
$\mathcal{Q}(\mathbf{x})$	Quaternion parameterized by axis-angle vector $\mathbf{x}$
$\mathcal{R}(\mathbf{q})$	Rotation matrix parameterized by quaternion $\mathbf{q}$

Notation

$\mathbf{x}(t)$	Continuous-time system state at time t
$\mathbf{x}_k$	Discrete-time system state at step k
$\mathbf{y}_k$	System response
$\mathbf{v}_k$	Process noise
$\mathbf{w}_k$	Measurement noise
$\mathbf{F}_k$	System matrix of a discrete-time state-space model
$\mathbf{H}_k$	Measurement matrix of a discrete-time state-space model
$\mathbf{Q}_k$	Process noise covariance matrix
$\mathbf{R}_k$	Measurement noise covariance matrix
$\hat{\mathbf{x}}_k$	State estimate at step k given $\mathbf{y}_1, \ldots, \mathbf{y}_k$
$\mathbf{P}_k$	State covariance at step k given $\mathbf{y}_1, \ldots, \mathbf{y}_k$
$\hat{\mathbf{x}}_k^-$	State estimate at step k given $\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}$
$\mathbf{P}_k^-$	State covariance at step k given $\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}$
$\tilde{\mathbf{y}}_k$	Measurement pre-fit residual
$\mathbf{S}_k$	Pre-fit residual covariance matrix
$\mathbf{K}_k$	Kalman filter gain
$\hat{\mathbf{q}}_k$	Nominal orientation quaternion
$\delta\mathbf{q}$	Orientation error quaternion

Abbreviations

EKF	Extended Kalman filter
IMU	Inertial measurement unit
MARG	Magnetic, angular rate, and gravity
MEKF	Multiplicative extended Kalman filter
MEMS	Microelectromechanical system
PDF	Probability density function
UI	User interface
USB	Universal serial bus
VKF	Vector Kalman filter

1 Introduction

Orientation tracking, also known as attitude and heading determination, refers to the methods and technologies used in estimating the orientation of an object such as a vehicle or a mobile device. Accurate orientation estimation is a necessity in many technical applications, especially in the field of aerospace engineering. In recent years, micro-electromechanical system (MEMS) technology has also enabled orientation tracking capabilities in a wide range of consumer devices such as smartphones, tablets, drones, game controllers, and virtual reality headsets [1, 2, 3].

The methods used for estimating orientation depend on the application. The focus of this thesis is on terrestrial orientation tracking using a magnetic field, angular rate, and gravity (MARG) sensor system. These systems are typically implemented as a combination of 3-axis inertial sensors (an accelerometer and a gyroscope) and a 3-axis magnetometer. MEMS technology enables low-cost fabrication of these sensors in a single miniaturized chip package [1]. Good orientation tracking algorithms are instrumental in compensating for the lesser performance of low-cost sensors.

A good orientation tracking algorithm makes optimal use of all available information in determining the orientation of a device. Accelerometer and magnetometer measurements can be used obtain reference directions for estimating orientation, but these measurements are noisy and subject to unwanted disturbances which make the resulting orientation estimate unstable. Gyroscopes can also be used to estimate orientation by integrating the angular rate measurements, but the resulting estimate will drift due to noise and biases [4]. Therefore, neither type of sensor alone is enough to form a stable and accurate estimate.

A *sensor fusion* approach to orientation estimation combines direction and angular rate measurements to compensate for the individual shortcomings of both. The celebrated Kalman filter, introduced by Rudolf Kálmán in the 1960s [5], is the basis of many sensor fusion algorithms in use today. The Kalman filter is a recursive linear-quadratic estimator that can be used to estimate the state of a system expressed as a stochastic state-space model. A non-linear extension of the Kalman filter, known as the extended Kalman filter (EKF) has been particularly useful for orientation estimation.

The purpose of this thesis is to investigate and compare two different sensor fusion algorithms for orientation estimation, a well-known quaternion-based estimator known as the multiplicative EKF (MEKF), and a vector-based estimator referred to in this thesis as the vector Kalman filter (VKF). The MEKF was originally developed for spacecraft attitude estimation, and has since been used aboard several NASA spacecraft [6]. The VKF is an extension by the author of the approach found in [7]. Its main advantage compared to the MEKF is that it is easier to understand and to implement. Both estimators are experimentally evaluated and their performance compared in a qualitative manner.

The use of Kalman filters in orientation estimation presents many possibilities and challenges. The scope of this thesis is limited to the implementation and comparison of the two estimators mentioned above. The stability of the resulting algorithms – a vast topic in itself – is not explored in this work. Non-real-time approaches such as Kalman smoothing are also omitted. However, formulating smoother algorithms based on the filter algorithms is relatively straightforward (see e.g. [8, 9]). Practical estimators are often augmented with real-time sensor bias estimation [10], but for the sake of simplicity, this is not done here. Lastly, the measurement models use simple Gaussian noise which is inaccurate especially for the gravity measured by an accelerometer.

The rest of this thesis is structured as follows. Chapter 2 introduces the discrete-time Bayesian filtering problem for time-varying linear and non-linear systems, and derives the linear and extended Kalman filter equations. Chapter 3 discusses the orientation tracking problem in detail, building on the results of Chapter 2 to establish two different filtering algorithms for orientation estimation using a MARG sensor system. Chapter 4 presents the experimental test device used to test implementations of the two filter algorithms derived in Chapter 3. Both algorithms are verified and the performance of both approaches is discussed in detail. Finally, Chapter 5 gives a summary of the work done and discusses the findings, limitations and future directions. Appendix A contains a brief synopsis of multivariate probability and the quaternions, which are used to establish the results in this thesis.

2 Bayesian filtering

This chapter gives a concise treatment of recursive Bayesian estimation in a stochastic state-space framework. The focus is on establishing the relevant results for this thesis, namely the linear and extended Kalman filters. A much more complete treatment can be found in [9].

Section 2.1 presents the linear stochastic state-space model that is the basis of the system models considered in this thesis. Section 2.2 treats the problem of recursively estimating the state of a state-space model with Gaussian probability distributions. Section 2.3 collects the results of Section 2.2 into a recursive algorithm that is the Kalman filter, and discusses its properties. Section 2.4 presents a general non-linear stochastic state-space model and discusses its properties in relation to state estimation. Section 2.5 derives the extended Kalman filter using a Taylor series approximation of the general non-linear state-space model.

2.1 Linear stochastic state-space model

Consider the discrete-time linear stochastic state-space model

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1}, \\ \mathbf{y}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k, \end{aligned} \tag{2.1}$$

where $\mathbf{F}_k$ is the system matrix at step k, $\mathbf{H}_k$ is the output matrix (also called the *sensitivity matrix*), $\mathbf{x}_k$ is the state, $\mathbf{y}_k$ is the measurement, $\mathbf{v}_k$ is the system noise sequence and $\mathbf{w}_k$ is the measurement noise sequence. We assume that $\mathbf{v}_k$ and $\mathbf{w}_k$ are Gaussian white noise, i.e. $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ and $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$, but this restriction can be relaxed [8]. Viewed independently, the equation for $\mathbf{x}_k$ is called the *dynamic model*, and the equation for $\mathbf{y}_k$ is called the *measurement model*. Figure 2.1 shows a diagram of the state-space model.

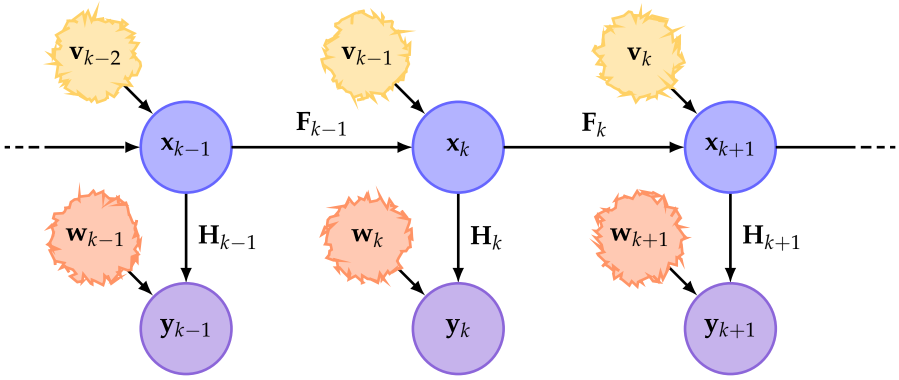

Figure 2.1: The discrete-time linear stochastic state-space model.

This model has a number of important properties. It is:

i. *Discrete-time*. All variables take values at discrete time steps k instead of being continuous functions of time. Discrete-time models are useful because they can be readily implemented on a computer.

ii. *Linear time-variant*. The next state $\mathbf{x}_k$ and the measurement $\mathbf{y}_k$ are linear functions of the state, written as matrices $\mathbf{F}_k$ and $\mathbf{H}_k$ which depend on the time step k.

iii. *Autonomous*. The system model contains no explicit input into the state. In control applications in particular, an input term $\mathbf{G}_{k-1}\mathbf{u}_{k-1}$ is added into the equation for $\mathbf{x}_k$. We leave it out, as we are only interested in estimating the state of the system, not controlling it.

iv. *Markovian*. The state of the system at step k depends only on the preceding state and no other states or measurements (see Figure 2.1). This makes the estimation problem much simpler and permits us to develop a feasible recursive estimator where the computational work needed to incorporate each new measurement into the filtering estimate is constant [9].

When developing a new Kalman filter for an estimation problem, one often starts with a continuous-time model representing the dynamics of the estimated quantity as it behaves in the physical world. A linear continuous-time model can be written in linear state-space form as a differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{v}(t), \tag{2.2}$$

where $\mathbf{A}(t)$ is the continuous-time counterpart of $\mathbf{F}_k$. This equation can then be discretized to find a corresponding discrete-time model suitable for implementation. A common discretization strategy is to hold the time-varying system matrix $\mathbf{A}(t)$ constant over a discrete time interval and solve the resulting constant-coefficient equation within each interval. This gives an approximation of the continuous-time dynamics whose accuracy improves as the length of the interval decreases.

Suppose that $\mathbf{A}(t) = \mathbf{A}_{k-1}$ when $t \in [t_{k-1}, t_k]$. If we know the previous state of the system $\mathbf{x}(t_{k-1}) = \mathbf{x}_{k-1}$ at some discrete time step k, then the solution to the nonhomogeneous differential Equation (2.2) when $t \in [t_{k-1}, t_k]$ is

$$\mathbf{x}(t) = e^{\mathbf{A}_{k-1}(t-t_{k-1})}\mathbf{x}(t_{k-1}) + \int_{t_{k-1}}^{t} e^{\mathbf{A}_{k-1}(t-\tau)}\mathbf{v}(\tau)\, d\tau, \tag{2.3}$$

where $\mathbf{x}(t_{k-1})$ is the initial value [11]. Based on this, we set $\mathbf{x}_k = \mathbf{x}(t_k)$. Comparing the right side of Equation (2.3) with the discrete-time model of Equation (2.1), we obtain the discrete system matrix and noise sequence

$$\begin{aligned}\mathbf{F}_{k-1} &= e^{\mathbf{A}_{k-1}(t_k - t_{k-1})} = e^{\mathbf{A}_{k-1}\Delta t_{k-1}},\\ \mathbf{v}_{k-1} &= \int_{t_{k-1}}^{t_k} e^{\mathbf{A}_{k-1}(t_k-\tau)}\mathbf{v}(\tau)\,d\tau.\end{aligned} \tag{2.4}$$

Here, $\mathbf{F}_{k-1}$ is also known as the *state transition matrix* of the continuous system from step $k-1$ to k [8]. The discrete process noise covariance matrix of $\mathbf{v}_{k-1}$ is given by

$$\begin{aligned}\mathbf{Q}_{k-1} &= \mathrm{E}\left[\left(\int_{t_{k-1}}^{t_k} e^{\mathbf{A}_{k-1}(t_k-\tau)}\mathbf{v}(\tau)\,d\tau\right)\left(\int_{t_{k-1}}^{t_k} e^{\mathbf{A}_{k-1}(t_k-\tau)}\mathbf{v}(\tau)\,d\tau\right)^{\mathrm{T}}\right]\\ &= \int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{t_k} e^{\mathbf{A}_{k-1}(t_k-\tau)}\mathbf{Q}\delta(\tau-\mu)\left(e^{\mathbf{A}_{k-1}(t_k-\mu)}\right)^{\mathrm{T}} d\tau\,d\mu\\ &= \int_{t_{k-1}}^{t_k} e^{\mathbf{A}_{k-1}(t_k-\tau)}\mathbf{Q}\left(e^{\mathbf{A}_{k-1}(t_k-\tau)}\right)^{\mathrm{T}} d\tau,\end{aligned} \tag{2.5}$$

where we used the fact that $\mathbf{v}(\tau)$ is Gaussian white noise, and thus $\mathrm{E}\big[\mathbf{v}(\tau)\mathbf{v}(\mu)^{\mathrm{T}}\big] = \mathbf{Q}\delta(\tau-\mu)$ where δ is the Dirac delta function.

2.2 Stochastic state estimation

The problem of state estimation boils down to determining the probability distribution of $\mathbf{x}_k$ at each step k given the available information. In this case, the available information consists of the assumed system model (i.e. $\mathbf{F}_k$, $\mathbf{H}_k$, $\mathbf{v}_k$, and $\mathbf{w}_k$), the probability distribution (i.e. PDF) $p(\mathbf{x}_0)$ of the initial state $\mathbf{x}_0$, and all the measurements $\mathbf{y}_1, \ldots, \mathbf{y}_T$ obtained so far. Starting from the initial distribution, we wish to compute the conditional distributions $p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$ which constitute refined views of the true state $\mathbf{x}_k$ of the system based on evidence in the form of the measurements $\mathbf{y}_1, \ldots, \mathbf{y}_T$.

This approach to state estimation is quite general in that it is not limited to any particular type of probability distribution. However, dealing with general distributions is computationally intractable [9]. Therefore, we will restrict ourselves exclusively to Gaussian distributions. These distributions are fully determined by their mean and covariance, so that we may write

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) = \mathcal{N}(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k|T}, \mathbf{P}_{k|T}), \tag{2.6}$$

where $\hat{\mathbf{x}}_{k|T}$ is the mean and $\mathbf{P}_{k|T}$ is the covariance matrix of the distribution. The mean is also a mode of a Gaussian distribution which means that $\hat{\mathbf{x}}_{k|T}$ is the best estimate of $\mathbf{x}_k$ assuming that $p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$ represents our best knowledge of the distribution of $\mathbf{x}_k$ given $\mathbf{y}_1, \ldots, \mathbf{y}_k$.

Note that the time step k is separate from the number of measurements T. This gives rise to three distinct possibilities. If $k = T$, we have a *filtering* distribution which is the estimate at a given step k when all measurements up to and including that step have been observed. If $k < T$, we have a *smoothing* distribution, i.e. a refinement of a past estimate based on later measurements. If $k > T$, we have a *prediction* distribution, i.e. an extrapolation into the future based on the (assumed) system dynamics. [9]

We need to consider only filtering and prediction distributions when deriving the Kalman filter. In particular, we are only interested in the cases where $T \in \{k-1, k\}$. To reduce notational clutter, we denote these means and covariances by the shorthands

$$\begin{aligned} \hat{\mathbf{x}}_k &\triangleq \hat{\mathbf{x}}_{k|k}, \\ \mathbf{P}_k &\triangleq \mathbf{P}_{k|k}, \\ \hat{\mathbf{x}}_k^- &\triangleq \hat{\mathbf{x}}_{k|k-1}, \\ \mathbf{P}_k^- &\triangleq \mathbf{P}_{k|k-1}. \end{aligned} \tag{2.7}$$

We will also denote distributions conditional on measurements up to and including the step k by the shorthand

$$p_k(\mathbf{z}) \triangleq p(\mathbf{z} \,|\, \mathbf{y}_1, \ldots, \mathbf{y}_k). \tag{2.8}$$

2.2.1 State prediction

We now consider the problem of predicting the future state of the system given its current state. More concretely, we want to compute the prediction distribution $p_{k-1}(\mathbf{x}_k)$ given the filtering distribution $p_{k-1}(\mathbf{x}_{k-1})$ at some time step k. Assuming that the filtering distribution is Gaussian, we have

$$p_{k-1}(\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_{k-1} \,|\, \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}). \tag{2.9}$$

According to Equation (2.1), $\mathbf{x}_k$ is a linear combination of $\mathbf{x}_{k-1}$ and $\mathbf{v}_{k-1}$, which means that the conditional distribution of $\mathbf{x}_k$ given $\mathbf{x}_{k-1}$ is

$$p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) = p_{k-1}(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_{k-1} \,|\, \mathbf{F}_{k-1}\mathbf{x}_{k-1}, \mathbf{Q}_{k-1}). \tag{2.10}$$

The first equality is justified because the state-space model is Markovian, meaning that $\mathbf{x}_k$ is independent from the measurements if $\mathbf{x}_{k-1}$ is fixed. Equation (A8) then gives the prediction distribution

$$\begin{aligned} p_{k-1}(\mathbf{x}_k) &= \mathcal{N}(\mathbf{x}_k \mid \hat{\mathbf{x}}_k^-, \mathbf{P}_k^-), \\ \hat{\mathbf{x}}_k^- &\triangleq \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}, \\ \mathbf{P}_k^- &\triangleq \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}. \end{aligned} \tag{2.11}$$

This prediction distribution represents the best estimate of the state of the system in the absence of any new measurements. It is the result of propagating our previous estimate through the dynamics of the system as given by Equation (2.1) (see Figure 2.2 below).

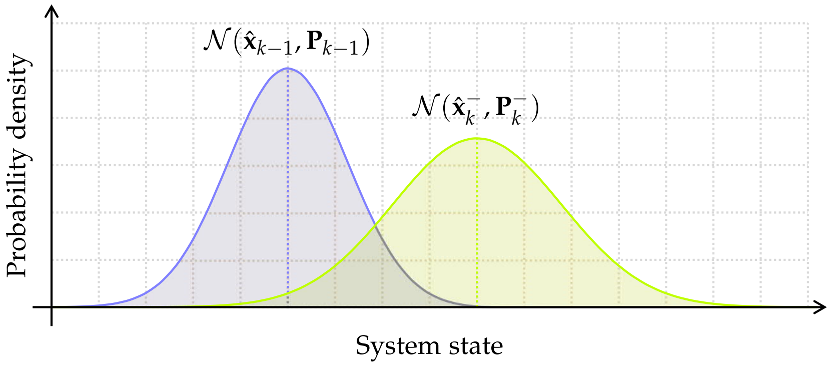

Figure 2.2: Propagation of the probability distribution of the state (blue) through the dynamic model. The covariance of the resulting distribution (green) is larger from the added uncertainty due to noise.

It is possible to predict the state further by iterating the above steps, resulting in distributions $p_{k-1}(\mathbf{x}_{k+n})$ for some $n \in \mathbb{N}$. Because the model incorporates a noise term at each step, our confidence in the estimate decreases the further ahead we predict, as evidenced by the term $\mathbf{Q}_{k-1}$ in Equation (2.11).

2.2.2 Measurement update

After predicting the new state of the system, a new measurement $\mathbf{y}_k$ is obtained which brings new information of the actual state of the system. We may then compute a new filtering distribution $p_k(\mathbf{x}_k)$ given the previously computed prediction distribution $p_{k-1}(\mathbf{x}_k)$ and this new measurement. Suppose that we have $p_{k-1}(\mathbf{x}_k)$ as given by Equation (2.11) at some time step k. According to Equation (2.1), $\mathbf{y}_k$ is a linear combination of $\mathbf{x}_k$ and $\mathbf{w}_k$, which means that the conditional distribution of $\mathbf{y}_k$ given $\mathbf{x}_k$ is

$$p(\mathbf{y}_k \mid \mathbf{x}_k) = p_{k-1}(\mathbf{y}_k \mid \mathbf{x}_k) = \mathcal{N}(\mathbf{y}_k \mid \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k), \tag{2.12}$$

where the first equality is justified because the measurement $\mathbf{y}_k$ is independent from all previous measurements when $\mathbf{x}_k$ is fixed. Equation (A8) then gives the new filtering distribution as

$$\begin{aligned} p_k(\mathbf{x}_k) &= p_{k-1}(\mathbf{x}_k \,|\, \mathbf{y}_k) = \mathcal{N}(\mathbf{x}_k \,|\, \hat{\mathbf{x}}_k, \mathbf{P}_k), \\ \hat{\mathbf{x}}_k &\triangleq \hat{\mathbf{x}}_k^- + \mathbf{P}_k^- \mathbf{H}_k^\mathrm{T} (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\mathrm{T} + \mathbf{R}_k)^{-1} (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-), \\ \mathbf{P}_k &\triangleq \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^\mathrm{T} (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\mathrm{T} + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^-. \end{aligned} \tag{2.13}$$

This distribution is merely a result of applying Bayes' theorem to invert the conditional probability distribution of the measurement model. Because of this, the prediction distribution $p_{k-1}(\mathbf{x}_k)$ is also called the *prior* distribution and the new filtering distribution $p_k(\mathbf{x}_k)$ the *posterior* distribution of $\mathbf{x}_k$ at time step k. The mean $\hat{\mathbf{x}}_k^-$ and covariance $\mathbf{P}_k^-$ are called the *a priori* state estimate and covariance. Likewise, the mean $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$ are called the *a posteriori* state estimate and covariance. Figure 2.3 illustrates the measurement update when $\mathbf{H}_k = \mathbf{I}$, meaning that the measurement and state spaces are the same.

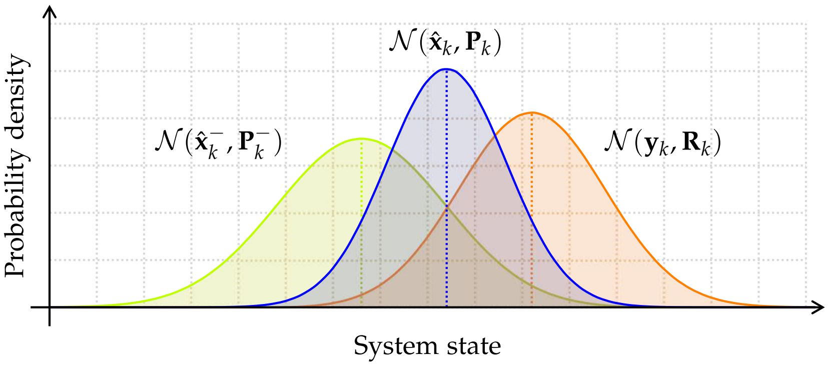

Figure 2.3: The measurement update. The posterior distribution (blue) is a tradeoff between the prior (green) and measurement (red) distributions such that the covariance (i.e. width) of the posterior is minimized.

2.3 Kalman filter

The Kalman filter now follows from the results of the previous section. Combining Equations (2.11) and (2.13) gives $p_k(\mathbf{x}_k)$ in terms of $p_{k-1}(\mathbf{x}_{k-1})$. If the distribution $p(\mathbf{x}_0)$ of the initial state is Gaussian, then it follows by induction that $p_k(\mathbf{x}_k)$ is Gaussian for all k. We can use this fact to establish an iterative algorithm. To make the equations clearer, we first define the auxiliary terms

$$\begin{aligned} \tilde{\mathbf{y}}_k &\triangleq \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-, \\ \mathbf{S}_k &\triangleq \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\mathrm{T} + \mathbf{R}_k, \\ \mathbf{K}_k &\triangleq \mathbf{P}_k^- \mathbf{H}_k^\mathrm{T} \mathbf{S}_k^{-1}, \end{aligned} \tag{2.14}$$

where $\tilde{\mathbf{y}}_k$ is known as the *residual*, $\mathbf{S}_k$ is the *residual covariance* and $\mathbf{K}_k$ is the *optimal Kalman gain*. Substituting these definitions into Equations (2.11) and (2.13) and rearranging, the state prediction and measurement update equations become

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1},\\ \mathbf{P}_k^- &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1},\\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k,\\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-.\end{aligned} \tag{2.15}$$

Examining the equation for $\hat{\mathbf{x}}_k$ reveals that the Kalman filter estimate is a linear combination of the predicted state and the residual which measures the deviation of the actual measurement from that given by the prediction. Intuitively, the filter balances between prediction and evidence from measurements at each time step, where the balance is controlled by the Kalman gain $\mathbf{K}_k$. The choice of $\mathbf{K}_k$ given by the maximum-a-posteriori (MAP) estimate of the updated distribution given by the Bayesian update rule is not the only possible one, but it is optimal in the sense that it minimizes the resulting covariance $\mathbf{P}_k$ [8].

Other, equivalent ways of formulating the Kalman filter exist [8, 9]. It should be noted that, in a computer implementation, the above expression for the a posteriori covariance $\mathbf{P}_k$ may result in a non-symmetric matrix due to rounding errors. An analytically equivalent expression which guarantees symmetry is

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}}, \tag{2.16}$$

commonly referred to as the *Joseph form* for the a posteriori covariance [8].

2.4 Non-linear state-space models

The discrete linear state-space model can be generalized into a non-linear model in a straightforward way. We write

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}),\\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k),\end{aligned} \tag{2.17}$$

where $\mathbf{f}_{k-1}$ and $\mathbf{h}_k$ are general non-linear functions of the state and noise, and the other variables are defined as in the linear case. With the exception of non-linearity, this model has all the same basic properties as the linear state-space model in that it is discrete-time, autonomous and Markovian.

The main difficulty in estimating non-linear models is that the probability distributions inevitably become non-Gaussian, which makes them difficult or impossible to compute with. Such distributions may be multimodal which makes the notion of "best estimate" problematic. The means of such distributions give no useful information about probabilities. In fact, the mean of a distribution resulting from a non-linear model might even be an impossible state with $p(\hat{\mathbf{x}}_k) = 0$.

A standard strategy for dealing with non-linearity is to approximate the resulting distributions by Gaussian distributions using a technique called *moment matching*. The mean and covariance of the resulting distribution are computed exactly or to an approximation, and the resulting approximation is then simply a Gaussian distribution with matching mean and covariance. These approximation techniques are collectively referred to as *general Gaussian filtering*. [9] The success of these techniques naturally depends on the properties of the non-linear model. A high degree of non-linearity may prevent use of Gaussian approximations altogether, in which case more sophisticated algorithms may be needed. However, Gaussian-approximated filters have found widespread success in many non-linear estimation tasks.

Perhaps the most immediately obvious approach to approximating the non-linear distributions is to linearize the functions **f** and **h** using their Taylor series expansions. The resulting filter formulation is known as the Extended Kalman Filter (EKF), and it was the first non-linear extension to the Kalman filter. Later variants include the Statistically Linearized Filter (SLF) and the Unscented Kalman Filter (UKF). Other variants based on numerical approximation include the Gauss-Hermite Kalman Filter (GHKF) and the Cubature Kalman Filter (CKF). [9] The non-linear tracking model considered in this thesis is suitable for the EKF which we derive in the next section. The EKF requires that the functions **f** and **h** be differentiable at appropriate points, which is the case for the models developed in this thesis.

2.5 Extended Kalman filter

We now consider the linearization of Equations (2.17). Suppose that

$$p_{k-1}(\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_{k-1} \,|\, \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}). \tag{2.18}$$

By linearizing the dynamic model at $(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) = (\hat{\mathbf{x}}_{k-1}, \mathbf{0})$, we obtain

$$\mathbf{x}_k \approx \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}) + \mathbf{F}_{\mathbf{x},k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{F}_{\mathbf{v},k-1}\mathbf{v}_{k-1}, \tag{2.19}$$

where $\mathbf{F}_{\mathbf{x},k-1}$ and $\mathbf{F}_{\mathbf{v},k-1}$ are the Jacobian matrices of $\mathbf{f}_{k-1}$ with respect to the state and noise respectively, evaluated at the linearization point. The conditional distribution of $\mathbf{x}_k$ given $\mathbf{x}_{k-1}$ is then approximated as $p_{k-1}(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}) \simeq \mathcal{N}(\mathbf{x}_k \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\begin{aligned} \boldsymbol{\mu} &\triangleq \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}) + \mathbf{F}_{\mathbf{x},k-1}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}), \\ \boldsymbol{\Sigma} &\triangleq \mathbf{F}_{\mathbf{v},k-1}\mathbf{Q}_{k-1}\mathbf{F}_{\mathbf{v},k-1}^{\mathrm{T}}, \end{aligned} \tag{2.20}$$

and Equation (A8) then gives the prediction $p_{k-1}(\mathbf{x}_k) \simeq \mathcal{N}(\mathbf{x}_k \,|\, \hat{\mathbf{x}}_k^-, \mathbf{P}_k^-)$, where

$$\begin{aligned} \hat{\mathbf{x}}_k^- &\triangleq \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{0}), \\ \mathbf{P}_k^- &\triangleq \mathbf{F}_{\mathbf{x},k-1}\mathbf{P}_{k-1}\mathbf{F}_{\mathbf{x},k-1}^{\mathrm{T}} + \mathbf{F}_{\mathbf{v},k-1}\mathbf{Q}_{k-1}\mathbf{F}_{\mathbf{v},k-1}^{\mathrm{T}}. \end{aligned} \tag{2.21}$$

The update step is handled in the same way. We linearize the measurement model at $(\mathbf{x}_k, \mathbf{w}_k) = (\hat{\mathbf{x}}_k^-, \mathbf{0})$, giving

$$\mathbf{y}_k \approx \mathbf{h}_k(\hat{\mathbf{x}}_k^-) + \mathbf{H}_{\mathbf{x},k}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \mathbf{H}_{\mathbf{w},k}\mathbf{w}_k, \tag{2.22}$$

where $\mathbf{H}_{\mathbf{x},k}$ and $\mathbf{H}_{\mathbf{w},k}$ are the Jacobian matrices of $\mathbf{h}_k$ with respect to the state and noise respectively, evaluated at the linearization point. The conditional distribution of $\mathbf{y}_k$ given $\mathbf{x}_k$ is then approximated as $p_{k-1}(\mathbf{y}_k \,|\, \mathbf{x}_k) \simeq \mathcal{N}(\mathbf{y}_k \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\begin{aligned} \boldsymbol{\mu} &\triangleq \mathbf{h}_k(\hat{\mathbf{x}}_k^-) + \mathbf{H}_{\mathbf{x},k}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-), \\ \boldsymbol{\Sigma} &\triangleq \mathbf{H}_{\mathbf{w},k}\mathbf{R}_k\mathbf{H}_{\mathbf{w},k}^{\mathrm{T}}, \end{aligned} \tag{2.23}$$

and Equation (A8) then gives the updated distribution $p_k(\mathbf{x}_k) \simeq \mathcal{N}(\mathbf{x}_k \,|\, \hat{\mathbf{x}}_k, \mathbf{P}_k)$, where

$$\begin{aligned} \hat{\mathbf{x}}_k &\triangleq \hat{\mathbf{x}}_k^- + \mathbf{P}_k^-\mathbf{H}_{\mathbf{x},k}^{\mathrm{T}}(\mathbf{H}_{\mathbf{x},k}\mathbf{P}_k^-\mathbf{H}_{\mathbf{x},k}^{\mathrm{T}} + \mathbf{H}_{\mathbf{w},k}\mathbf{R}_k\mathbf{H}_{\mathbf{w},k}^{\mathrm{T}})^{-1}[\mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-, \mathbf{0})], \\ \mathbf{P}_k &\triangleq \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_{\mathbf{x},k}^{\mathrm{T}}(\mathbf{H}_{\mathbf{x},k}\mathbf{P}_k^-\mathbf{H}_{\mathbf{x},k}^{\mathrm{T}} + \mathbf{H}_{\mathbf{w},k}\mathbf{R}_k\mathbf{H}_{\mathbf{w},k}^{\mathrm{T}})^{-1}\mathbf{H}_{\mathbf{x},k}\mathbf{P}_k^-. \end{aligned} \tag{2.24}$$

We may now substitute the auxiliary definitions of Equation (2.14) into Equations (2.21) and (2.24) to obtain the extended Kalman filter equations

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{0}), \\ \mathbf{P}_k^- &= \mathbf{F}_{\mathbf{x},k-1}\mathbf{P}_{k-1}\mathbf{F}_{\mathbf{x},k-1}^{\mathrm{T}} + \mathbf{F}_{\mathbf{v},k-1}\mathbf{Q}_{k-1}\mathbf{F}_{\mathbf{v},k-1}^{\mathrm{T}}, \\ \tilde{\mathbf{y}}_k &= \mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-, \mathbf{0}), \\ \mathbf{S}_k &= \mathbf{H}_{\mathbf{x},k}\mathbf{P}_k^-\mathbf{H}_{\mathbf{x},k}^{\mathrm{T}} + \mathbf{H}_{\mathbf{w},k}\mathbf{R}_k\mathbf{H}_{\mathbf{w},k}^{\mathrm{T}}, \\ \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}_{\mathbf{x},k}^{\mathrm{T}}\mathbf{S}_k^{-1}, \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k, \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_{\mathbf{x},k})\mathbf{P}_k^-. \end{aligned} \tag{2.25}$$

This is the most general form of the first-order extended Kalman filter for an autonomous (zero-input) system. Note that if the system functions **f** and **g** are linear in either state or noise, then the EKF coincides with the linear Kalman filter for those parts.

Although the extended Kalman filter is essentially a linear Kalman filter for the linearized system, there are important differences to consider. The expression for $\mathbf{P}_k$ in the linear case shows that it is independent of both the filter state and the measurements. As a result, relatively strong convergence and stability results are available for the linear Kalman filter [8, 12]. In contrast, the EKF uses Jacobian system and measurement matrices which depend on the linearization point and thus the filter state. This makes the stability of such filters depend on the initial state estimate, the measurements obtained from the system, and the accuracy of the linearization. The stability analysis of non-linear Kalman filters is in general difficult compared to linear filters [12].

3 Orientation tracking

This chapter forms the core of this thesis. Section 3.1 gives a mathematical formulation of the orientation tracking problem using coordinate frames. Section 3.2 presents the general outline of a quaternion-based state-space model for orientation tracking. Section 3.3 derives the extended multiplicative Kalman filter (MEKF) algorithm using the model of Section 3.2 as a base. Section 3.4 derives an alternative vector-based Kalman filter that also uses the model of Section 3.2 as a starting point.

3.1 Reference frames and rotations

Orientation tracking can be formulated as the problem of estimating the orientation (i.e. rotation) of a *reference frame* relative to an external reference frame in 3-dimensional Euclidean space. A reference frame (or just *frame*) is given by an origin point and three orthogonal axes (unit vectors). A reference frame attached to an object being tracked is commonly called the *body frame* [4, 10, 13]. The external frame is sometimes called the *Earth-fixed frame* [13] as it is commonly fixed relative to the Earth in terrestrial orientation tracking (see Figure 3.1 below).

Because orientation tracking is only concerned with relative orientation, the relative position of the frames is often irrelevant and can be ignored. The exception to this is when the relative position affects measurements of the relative orientation. For example, a compass has to be adjusted for magnetic declination in order to find true north, and the magnetic declination varies between locations on Earth. However, this effect is typically negligible in terrestrial orientation tracking at short distances and time scales. Therefore, an intermediate frame known as the *navigation frame* can be used in place of the Earth-fixed frame. The navigation frame is commonly defined with its origin point coincident with the origin of the body frame, the x-axis pointing east, the y-axis pointing to true north, and the z-axis pointing up [13] (see Figure 3.1).

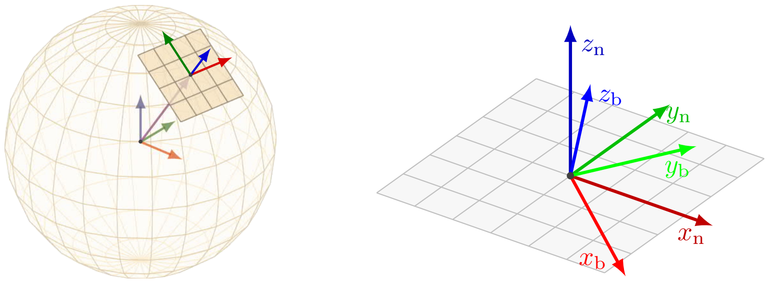

Figure 3.1: Reference frames. Left: Earth-Centered/Earth-Fixed (ECEF) frame (in the center) and the navigation frame (tangent to the surface). Right: navigation and body frames with coincident origins.

Suppose that $\mathbf{v}^{\mathrm{n}}, \mathbf{v}^{\mathrm{b}} \in \mathbb{R}^3$ are the coordinate representations of a vector $\mathbf{v}$ in the navigation and body frames, respectively. They are related by the equation

$$\mathbf{v}^{\mathrm{b}} = \mathbf{R}^{\mathrm{bn}}\mathbf{v}^{\mathrm{n}}, \tag{3.1}$$

where $\mathbf{R}^{\mathrm{bn}} \in \mathrm{SO}(3)$ is a rotation (i.e. a linear map) which takes the representation of a vector from the navigation frame into the body frame. This is known as the *passive* interpretation of a transform, where the vector stays the same but the frame is changed. This is contrasted with the *active* interpretation, where a new vector is produced in the same reference frame, e.g. $\mathbf{v}' = \mathbf{R}\mathbf{v}$. Where an active transform acts on a vector, the passive transform acts on the axes of a reference frame. In this sense, they are inverses of each other (see Figure 3.2). In the modeling that follows, we adopt the passive interpretation, as it is useful for dealing with reference frames.

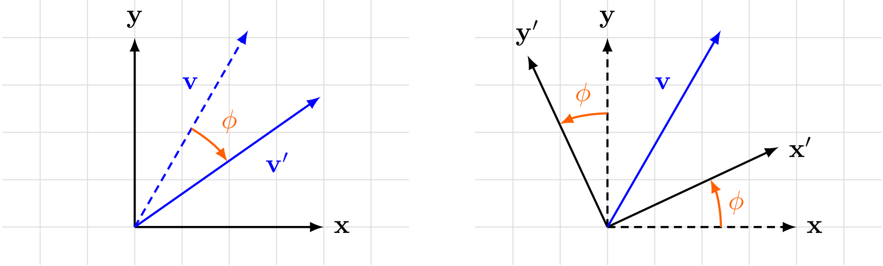

Figure 3.2: Active and passive interpretation of rotations.

The rotation $\mathbf{R}^{\mathrm{bn}}$ can be parameterized in several ways. The 3×3 rotation matrix is a common parameterization, but as a general representation of a linear transform it is cumbersome for manipulating rotations specifically. Minimum-degree parameterizations like the Euler angles, the Tait-Bryan angles or the axis-angle representation are better suited for rotations, but they suffer from singularities inherent to all 3-dimensional parameterizations. Quaternions are a minimal singularity-free representation. [10] Quaternions form the basis of most orientation tracking models, and we will use them for both the non-linear model and as a starting point for the linear model.

Based on the above definitions, we can solve the orientation tracking problem by determining $\mathbf{R}^{\mathrm{bn}}$. Suppose that we know the representation $\mathbf{g}^{\mathrm{n}}$ of the gravitational acceleration vector $\mathbf{g}$ which is constant in the navigation frame. If we can measure the gravity in the body frame, then the equation $\mathbf{g}^{\mathrm{b}} = \mathbf{R}^{\mathrm{bn}}\mathbf{g}^{\mathrm{n}}$ gives $\mathbf{R}^{\mathrm{bn}}$ up to a rotation about $\mathbf{g}$. One vector alone is not enough to determine the orientation fully: a second vector is needed that is not parallel to $\mathbf{v}_{\mathrm{g}}$. Suppose that we also know the representation $\mathbf{m}^{\mathrm{n}}$ of the geomagnetic field $\mathbf{m}$, also constant in the navigation frame. The orientation is then obtained by solving

$$\begin{aligned} \mathbf{g}^{\mathrm{b}} &= \mathbf{R}^{\mathrm{bn}}\mathbf{g}^{\mathrm{n}}, \\ \mathbf{m}^{\mathrm{b}} &= \mathbf{R}^{\mathrm{bn}}\mathbf{m}^{\mathrm{n}}, \end{aligned} \tag{3.2}$$

for the rotation $\mathbf{R}^{\mathrm{bn}}$. In the following sections, we derive Kalman filters capable of estimating a solution to Equation (3.2) when the measurements of $\mathbf{g}^{\mathrm{b}}$ and $\mathbf{m}^{\mathrm{b}}$ are affected by noise.

3.2 State-space representation for orientation tracking

We will now consider a state-space model for orientation tracking that is based on quaternions, as alluded to in Section 3.1. As a minimal representation of rotations that is also free of singularities, they are a good candidate for orientation tracking. Solving the orientation tracking problem involves finding (or approximating) a solution to Equations (3.2) in terms of a quaternion $\mathbf{q}$, which in turn requires knowledge of $\mathbf{g}^{\mathrm{b}}$ and $\mathbf{m}^{\mathrm{b}}$. However, in a real system, we can not determine $\mathbf{g}^{\mathrm{b}}$ and $\mathbf{m}^{\mathrm{b}}$ directly, but have to rely on inherently noisy sensor measurements.

The gravity vector $\mathbf{g}^{\mathrm{b}}$ is in practice measured with an accelerometer, which leads us to a measurement model of the form

$$\mathbf{y}_{\mathrm{g}} = \mathbf{a}^{\mathrm{b}} - \mathbf{g}^{\mathrm{b}} + \mathbf{w}_{\mathrm{a}} = -\mathbf{g}^{\mathrm{b}} + \mathbf{w}_{\mathrm{g}}, \tag{3.3}$$

where $\mathbf{a}^{\mathrm{b}}$ is acceleration due to forces other than gravity and $\mathbf{w}_{\mathrm{a}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathrm{a}})$ is the measurement noise of the sensor. The sign of $\mathbf{g}^{\mathrm{b}}$ is negative because the sensor effectively measures gravity as the acceleration due to the force supporting the sensor against gravity. Assuming that the non-gravity part of the measured acceleration is Gaussian white noise, $\mathbf{a}^{\mathrm{b}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathrm{a'}})$, we may combine it with the sensor noise, giving $\mathbf{w}_{\mathrm{g}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathrm{g}})$ where $\boldsymbol{\Sigma}_{\mathrm{g}} = \boldsymbol{\Sigma}_{\mathrm{a}} + \boldsymbol{\Sigma}_{\mathrm{a'}}$. Note that this model assumes that the sensor is well-calibrated with negligible constant bias, axis misalignment and scale errors.

The magnetic field $\mathbf{m}^{\mathrm{b}}$ can be measured using a magnetometer, which leads to a straightforward measurement model

$$\mathbf{y}_{\mathrm{m}} = \mathbf{m}^{\mathrm{b}} + \mathbf{w}_{\mathrm{m}}, \tag{3.4}$$

where $\mathbf{w}_{\mathrm{m}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathrm{m}})$. This model also assumes a well-calibrated sensor. Dynamic disturbances in the magnetic field are often dependent on the location of the sensor and other objects in the vicinity of the sensor; these are simply neglected in this model.

Combining the above with the Equations (3.2) gives the sensor measurement models

$$\begin{aligned} \mathbf{y}_{\mathrm{g}} &= -\mathbf{R}^{\mathrm{bn}}\mathbf{g}^{\mathrm{n}} + \mathbf{w}_{\mathrm{g}}, \\ \mathbf{y}_{\mathrm{m}} &= \mathbf{R}^{\mathrm{bn}}\mathbf{m}^{\mathrm{n}} + \mathbf{w}_{\mathrm{m}}, \\ \mathbf{R}^{\mathrm{bn}} &= (\mathbf{R}^{\mathrm{nb}})^{\mathrm{T}} = \mathcal{R}(\mathbf{q})^{\mathrm{T}}, \end{aligned} \tag{3.5}$$

where $\mathcal{R}(\mathbf{q})$ is a rotation matrix parameterized by $\mathbf{q}$ (see Section A.2). Note that we take $\mathbf{q}$ to represent the rotation from the body frame into the navigation frame (and not vice versa). The measurements $\mathbf{y}_{\mathrm{g}}$ and $\mathbf{y}_{\mathrm{m}}$ can be used to directly approximate a solution to Equations (3.5), but they alone are not enough to make use of a Kalman filter; a dynamic model for $\mathbf{q}$ is needed. To that end, we assume that a noisy measurement of the angular velocity about the body frame axes is available,

$$\mathbf{y}_{\omega} = \boldsymbol{\omega} + \mathbf{w}_{\omega}, \tag{3.6}$$

where $\mathbf{w}_{\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\omega})$. The kinematic equation for $\mathbf{q}$ can then be written as

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \odot \boldsymbol{\omega} = \frac{1}{2}\mathbf{q} \odot (\mathbf{y}_{\omega} - \mathbf{w}_{\omega}), \tag{3.7}$$

which is a continuous-time dynamic model for the state. Discretizing Equation (3.7) with a zero-order hold $\boldsymbol{\omega}_k = \boldsymbol{\omega}(t_k)$ gives

$$\mathbf{q}_{k+1} = \exp\left(\frac{1}{2}[\boldsymbol{\omega}_k \otimes]\Delta t\right)\mathbf{q}_k = \mathbf{q}_k \odot \mathcal{Q}(\boldsymbol{\omega}_k \Delta t), \tag{3.8}$$

where $\mathcal{Q}$ is a quaternion parameterized by the rotation vector $\boldsymbol{\omega}_k \Delta t$.

It is possible to formulate a Kalman filter based on Equations (3.5) – (3.7) and the resulting filter has been used in practice [10]. However, the fact that the quaternion representation is not strictly minimal poses difficulties. To map a 4-dimensional quaternion to a 3-dimensional rotation, we either have to restrict the state space to unit quaternions (norm invariant), or use a non-injective map (the ray representation, see e.g. [10]). It is evident from the measurement update equation of the EKF that it does not in general preserve the norm invariant. Ad-hoc remedies such as brute-force normalization at each time step are possible, but may lead to other problems [4, 10]. In the ray representation the norm of the quaternion is unobservable, which may lead to problems with numerical precision [10].

A more robust strategy involves representing the orientation quaternion as a combination of a nominal quaternion and an error quaternion and using a Kalman filter to estimate only the error part. This type of filter is known as an *error-state Kalman filter*, owing to the fact that it uses the error of a nominal quantity as its state variable [14]. Concretely, we express the full state $\mathbf{x}$ as the combination of a nominal (reference) state $\hat{\mathbf{x}}$ and an error $\delta\mathbf{x}$ by writing

$$\mathbf{x} = \hat{\mathbf{x}} \oplus \delta\mathbf{x}, \tag{3.9}$$

where $\oplus$ denotes a generic composition operation in the state space. The Kalman filter is then formulated for $\delta\mathbf{x}$ instead of $\mathbf{x}$. Here, $\hat{\mathbf{x}}$ is not treated as a random variable itself but rather as a reference point for the error $\delta\mathbf{x}$. The standard predict-update cycle is augmented with a *reset* operation that moves the error $\delta\mathbf{x}$ into $\hat{\mathbf{x}}$ and resets $\delta\mathbf{x}$ to zero after each measurement update step. The initial estimate is $\hat{\mathbf{x}}_0$ and the initial error $\delta\mathbf{x}_0 = 0$, and thus the filter state is always zero between iterations.

3.3 Quaternion error-state filtering

We now consider an error-state filter for estimating the quaternion $\mathbf{q}$ known as the multiplicative extended Kalman filter (MEKF) (see e.g. [10]). We begin by splitting the quaternion estimate into a nominal part and an error part. Unit quaternions representing rotations naturally compose by multiplication, which gives

$$\mathbf{q} = \hat{\mathbf{q}} \odot \delta\mathbf{q}, \tag{3.10}$$

where $\delta\mathbf{q}$ is an error quaternion representing the orientation error in the body frame. The error could equivalently be defined in the navigation frame, but a choice has to be made because 3-dimensional rotations are not commutative.

The error-state filter strives to keep the error $\delta\mathbf{q}$ small at all times. In particular, it strives to keep the error from reaching an orientation that would be problematic for a representation with singularities. This feature permits us to further parameterize the 4-dimensional unit error quaternion by a 3-dimensional *rotation vector*,

$$\delta\mathbf{q} \triangleq \mathcal{Q}(\mathbf{x}) = \begin{bmatrix} \cos\left(\frac{\|\mathbf{x}\|}{2}\right) \\ \frac{\mathbf{x}}{\|\mathbf{x}\|}\sin\left(\frac{\|\mathbf{x}\|}{2}\right) \end{bmatrix}, \tag{3.11}$$

where $\mathbf{x}$ becomes the actual state vector of the filter. Reducing the dimensionality of the filter state solves the problems with quaternion norm outlined in Section 3.2. Moreover, the computational overhead over the filter is reduced by the fact that the filter now only involves inversions of 3-by-3 matrices. Figure 3.3 shows an overview of the operation of the resulting MEKF algorithm.

3.3.1 Prediction step

To formulate an error-state filter using Equation (3.10), the kinematic model of Equation (3.7) needs to be transformed into a dynamic model for the error rotation vector $\mathbf{x}$. We begin by considering the kinematic equation for the quaternion error

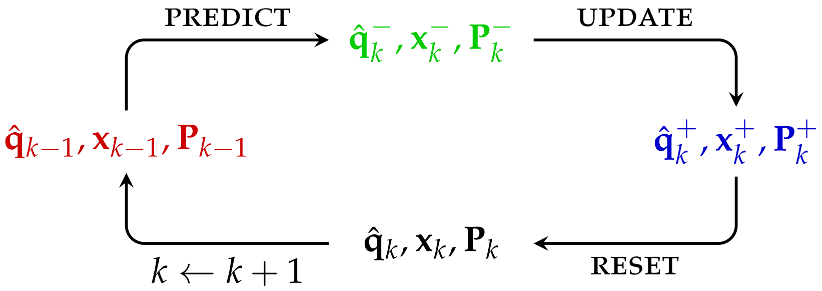
PREDICT
$\hat{\mathbf{q}}_k^-, \mathbf{x}_k^-, \mathbf{P}_k^-$
UPDATE
$\hat{\mathbf{q}}_{k-1}, \mathbf{x}_{k-1}, \mathbf{P}_{k-1}$
$\hat{\mathbf{q}}_k^+, \mathbf{x}_k^+, \mathbf{P}_k^+$
$\hat{\mathbf{q}}_k, \mathbf{x}_k, \mathbf{P}_k$
$k \leftarrow k+1$
RESET

Figure 3.3: Overview of the multiplicative extended Kalman filter algorithm.

$\delta\mathbf{q}$. Differentiating Equation (3.10) with respect to time gives

$$\dot{\mathbf{q}} = \dot{\hat{\mathbf{q}}} \odot \delta\mathbf{q} + \hat{\mathbf{q}} \odot \delta\dot{\mathbf{q}}, \tag{3.12}$$

which is the kinematic equation of $\mathbf{q}$ in terms of the kinematic equations for the nominal and error quaternions. Note that to fulfill Equation (3.7), it is enough to rotate either the nominal or the error quaternion. In the interest of keeping the error small (and thus making the linearization more accurate), we rotate the nominal quaternion and set $\delta\dot{\mathbf{q}} = 0$. The kinematic model of the nominal quaternion then becomes

$$\dot{\hat{\mathbf{q}}} = \dot{\mathbf{q}} \odot \delta\mathbf{q}^{-1}. \tag{3.13}$$

Substituting Equations (3.7) and (3.10) into the above, we obtain

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}} \odot \delta\mathbf{q} \odot (\mathbf{y}_\omega - \mathbf{w}_\omega) \odot \delta\mathbf{q}^{-1}. \tag{3.14}$$

Note that $\dot{\hat{\mathbf{q}}}$ is a random variable, because $\mathbf{w}_\omega$ and $\delta\mathbf{q}$ are random. We assume that $\mathbf{w}_\omega$ and $\delta\mathbf{q}$ are uncorrelated, and that $\delta\mathbf{q}$ represents a symmetric deviation from the null rotation. The expected value of $\dot{\hat{\mathbf{q}}}$ is then

$$\mathrm{E}[\dot{\hat{\mathbf{q}}}] = \frac{1}{2}\hat{\mathbf{q}} \odot \mathbf{y}_\omega. \tag{3.15}$$

We will simply denote this expected value by $\dot{\hat{\mathbf{q}}}$ and use it as the kinematic equation for the nominal quaternion.

Although the error quaternion itself is not affected by the dynamic model, its covariance matrix is. It is the only place where the uncertainty of the prediction step can be accounted for, because the nominal quaternion $\hat{\mathbf{q}}$ is by definition not random. Solving for $\delta\dot{\mathbf{q}}$ in Equation (3.12) and substituting the kinematic model equations for the true and nominal quaternions, we obtain

$$\begin{aligned}\delta\dot{\mathbf{q}} &= \hat{\mathbf{q}}^{-1} \odot \dot{\mathbf{q}} - \hat{\mathbf{q}}^{-1} \odot \dot{\hat{\mathbf{q}}} \odot \delta\mathbf{q} \\ &= \frac{1}{2}\Big(\hat{\mathbf{q}}^{-1} \odot \mathbf{q} \odot \boldsymbol{\omega} - \hat{\mathbf{q}}^{-1} \odot \hat{\mathbf{q}} \odot \mathbf{y}_\omega \odot \delta\mathbf{q}\Big) \\ &= \frac{1}{2}(\delta\mathbf{q} \odot \boldsymbol{\omega} - \mathbf{y}_\omega \odot \delta\mathbf{q}) \\ &= \frac{1}{2}(\delta\mathbf{q} \odot \mathbf{y}_\omega - \mathbf{y}_\omega \odot \delta\mathbf{q}) - \frac{1}{2}\delta\mathbf{q} \odot \mathbf{w}_\omega,\end{aligned} \tag{3.16}$$

which is the non-linear dynamic model for the error quaternion $\delta\mathbf{q}$. Writing the quaternion multiplications in matrix form, we get

$$\begin{aligned}\delta\dot{\mathbf{q}} &= \frac{1}{2}([\mathbf{y}_\omega \otimes] - [\mathbf{y}_\omega \odot])\delta\mathbf{q} - \frac{1}{2}[\delta\mathbf{q} \odot]\begin{bmatrix} 0 \\ \mathbf{w}_\omega \end{bmatrix} \\ &= \begin{bmatrix} 0 & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{3\times 1} & -[\mathbf{y}_\omega \times] \end{bmatrix}\delta\mathbf{q} - \frac{1}{2}[\delta\mathbf{q} \odot]\begin{bmatrix} 0 \\ \mathbf{w}_\omega \end{bmatrix}.\end{aligned} \tag{3.17}$$

In a minor departure from the procedure of linearizing a discrete-time model (see Section 2.5), we linearize this model in continuous time and then discretize the linear model. Because we are ultimately after the dynamic model for $\mathbf{x}$ instead of $\delta\mathbf{q}$, we first linearize the above equation with respect to $\mathbf{x}$. Assuming that the initial error at some time t_k is zero, the linearization point for the EKF prediction step is $(\mathbf{x}, \mathbf{w}_\omega) = (\mathbf{0}, \mathbf{0})$. Therefore

$$\delta\mathbf{q} \approx \begin{bmatrix} 1 \\ \frac{\mathbf{x}}{2} \end{bmatrix}, \qquad \text{and} \qquad \delta\dot{\mathbf{q}} \approx \begin{bmatrix} 0 \\ \frac{\dot{\mathbf{x}}}{2} \end{bmatrix}. \tag{3.18}$$

Substituting these into Equation (3.17) then gives

$$\begin{bmatrix} 0 \\ \frac{\dot{\mathbf{x}}}{2} \end{bmatrix} \approx \begin{bmatrix} 0 & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{3\times 1} & -[\mathbf{y}_\omega \times] \end{bmatrix}\begin{bmatrix} 1 \\ \frac{\mathbf{x}}{2} \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 1 & -\frac{\mathbf{x}^{\mathrm{T}}}{2} \\ \frac{\mathbf{x}}{2} & \mathbf{I}_3 + \frac{1}{2}[\mathbf{x} \times] \end{bmatrix}\begin{bmatrix} 0 \\ \mathbf{w}_\omega \end{bmatrix}, \tag{3.19}$$

from which we can extract the dynamic equation for $\mathbf{x}$ as

$$\begin{aligned}\dot{\mathbf{x}} &\approx -\mathbf{y}_\omega \times \mathbf{x} - \mathbf{w}_\omega - \frac{1}{2}\mathbf{x} \times \mathbf{w}_\omega \\ &\approx -\mathbf{y}_\omega \times \mathbf{x} - \mathbf{w}_\omega,\end{aligned} \tag{3.20}$$

where the cross product $\mathbf{x} \times \mathbf{w}_\omega$ vanishes entirely at the linearization point. Discretizing Equation (3.20) according to Equation (2.4) with a zero-order hold

$\mathbf{y}_{\omega,k} = \mathbf{y}_\omega(t_k)$ then gives

$$\begin{aligned}\mathbf{F}_{k-1} &= e^{-[\mathbf{y}_{\omega,k}\times]\Delta t},\\ \mathbf{v}_{k-1} &= \int_{t_{k-1}}^{t_k} e^{-[\mathbf{y}_{\omega,k}\times](t_k-\tau)}\mathbf{w}_\omega(\tau)\,d\tau,\end{aligned} \tag{3.21}$$

and the discrete-time covariance matrix of the process noise becomes

$$\mathbf{Q}_{k-1} = \int_{t_{k-1}}^{t_k} e^{-[\mathbf{y}_{\omega,k}\times](t_k-\tau)}\mathbf{S}_\omega\left(e^{-[\mathbf{y}_{\omega,k}\times](t_k-\tau)}\right)^{\mathrm{T}} d\tau, \tag{3.22}$$

where $\mathbf{S}_\omega$ is the continuous-time covariance matrix of the white noise process $\mathbf{w}_\omega$. Assuming that the gyroscope noise has the same covariance for each axis and is uncorrelated between the axes, i.e. $\mathbf{S}_\omega = \sigma_\omega^2\mathbf{I}_3$, the above simplifies to

$$\mathbf{Q}_{k-1} = \sigma_\omega^2 \int_{t_{k-1}}^{t_k} e^{-[\mathbf{y}_{\omega,k}\times](t_k-\tau)}\left(e^{-[\mathbf{y}_{\omega,k}\times](t_k-\tau)}\right)^{\mathrm{T}} d\tau = \sigma_\omega^2\Delta t\,\mathbf{I}_3, \tag{3.23}$$

because the matrix exponential of $-[\mathbf{y}_{\omega,k}\times]$ is a rotation matrix and therefore orthogonal.

Using the dynamic model given by the above equations, we end up with the error-state Kalman filter prediction equations

$$\begin{aligned}\hat{\mathbf{q}}_k^- &= \hat{\mathbf{q}}_{k-1} \odot \mathcal{Q}(\mathbf{y}_{\omega,k}\Delta t),\\ \hat{\mathbf{x}}_k^- &= \hat{\mathbf{x}}_{k-1} = \mathbf{0},\\ \mathbf{P}_k^- &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1},\end{aligned} \tag{3.24}$$

where the equation for $\hat{\mathbf{q}}_k^-$ is just the discretization of Equation (3.15) with the zero-order hold $\mathbf{y}_{\omega,k}$.

3.3.2 Update step

The update step works much like the normal extended Kalman filter update. The error is updated and the nominal quaternion is kept constant. We begin by deriving the measurement model of the filter. Based on the error-state representation of Equation (3.10), the rotation matrix of Equation (3.5) becomes

$$\mathbf{R}^{\mathrm{bn}} = \mathcal{R}(\mathbf{q})^{\mathrm{T}} = \mathcal{R}(\hat{\mathbf{q}} \odot \delta\mathbf{q})^{\mathrm{T}} = \mathcal{R}(\delta\mathbf{q})^{\mathrm{T}}\mathcal{R}(\hat{\mathbf{q}})^{\mathrm{T}}, \tag{3.25}$$

where $\mathcal{R}(\hat{\mathbf{q}})^{\mathrm{T}}$ is a constant matrix. The error rotation matrix $\mathcal{R}(\delta\mathbf{q})^{\mathrm{T}}$ has a non-linear relationship with the error state $\mathbf{x}$. Linearizing at $\mathbf{x} = 0$, we obtain

$$\mathcal{R}(\delta\mathbf{q})^{\mathrm{T}} \approx \mathbf{I}_3 - [\mathbf{x}\times]. \tag{3.26}$$

The measurement model equation for the accelerometer then becomes

$$\begin{aligned}\mathbf{y}_{\mathrm{g}} &= -(\mathbf{I}_3 - [\mathbf{x}\,\times])\mathcal{R}(\hat{\mathbf{q}})^{\mathrm{T}}\mathbf{g}^{\mathrm{n}} + \mathbf{w}_{\mathrm{g}} \\ &= -\mathcal{R}(\hat{\mathbf{q}})^{\mathrm{T}}\mathbf{g}^{\mathrm{n}} + [\mathbf{x}\,\times]\mathcal{R}(\hat{\mathbf{q}})^{\mathrm{T}}\mathbf{g}^{\mathrm{n}} + \mathbf{w}_{\mathrm{g}} \\ &= -\mathcal{R}(\hat{\mathbf{q}})^{\mathrm{T}}\mathbf{g}^{\mathrm{n}} - \left[\mathcal{R}(\hat{\mathbf{q}})^{\mathrm{T}}\mathbf{g}^{\mathrm{n}}\,\times\right]\mathbf{x} + \mathbf{w}_{\mathrm{g}},\end{aligned} \tag{3.27}$$

and the measurement model equation for the magnetometer is similar with the exception of the sign. Collecting both accelerometer and magnetometer measurements into one equation and noting the time step k, we get the filtering measurement model

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_{\mathrm{g},k} \\ \mathbf{y}_{\mathrm{m},k} \end{bmatrix} = \hat{\mathbf{y}}_k^- + \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k, \tag{3.28}$$

where $\mathbf{H}_k$, $\mathbf{w}_k$, and $\hat{\mathbf{y}}_k^-$ are

$$\begin{aligned}\mathbf{H}_k &= \begin{bmatrix} -[\mathcal{R}(\hat{\mathbf{q}}_k^-)^{\mathrm{T}}\mathbf{g}^{\mathrm{n}}\,\times] \\ [\mathcal{R}(\hat{\mathbf{q}}_k^-)^{\mathrm{T}}\mathbf{m}^{\mathrm{n}}\,\times] \end{bmatrix}, \\ \mathbf{w}_k &= \begin{bmatrix} \mathbf{w}_{\mathrm{g},k} \\ \mathbf{w}_{\mathrm{m},k} \end{bmatrix}, \\ \hat{\mathbf{y}}_k^- &= \begin{bmatrix} -\mathcal{R}(\hat{\mathbf{q}}_k^-)^{\mathrm{T}}\mathbf{g}^{\mathrm{n}} \\ \mathcal{R}(\hat{\mathbf{q}}_k^-)^{\mathrm{T}}\mathbf{m}^{\mathrm{n}} \end{bmatrix}.\end{aligned} \tag{3.29}$$

Expanding the Kalman filter equations at the linearization point $\hat{\mathbf{x}}_k^- = \mathbf{0}$ gives the measurement update equations

$$\begin{aligned}\hat{\mathbf{q}}_k^+ &= \hat{\mathbf{q}}_k^-, \\ \hat{\mathbf{x}}_k^+ &= \mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k)^{-1}(\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}}(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k)^{-1}\mathbf{H}_k\mathbf{P}_k^-,\end{aligned} \tag{3.30}$$

where the measurement noise covariance matrix $\mathbf{R}_k$ is

$$\mathbf{R}_k = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathrm{g},k} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\mathrm{m},k} \end{bmatrix}. \tag{3.31}$$

Note that we have written the a posteriori nominal quaternion $\hat{\mathbf{q}}_k^+$, error $\hat{\mathbf{x}}_k^+$, and covariance $\mathbf{P}_k^+$ with superscripts to accommodate the final reset step. The reset step is needed because the measurement update step results in a state $\hat{\mathbf{x}}_k^+ \neq \mathbf{0}$ that is incompatible with the prediction step assumption that $\hat{\mathbf{x}}_k = \mathbf{0}$.

3.3.3 Reset step

The reset step has to move the error into the nominal quaternion without affecting the actual orientation estimate (i.e. their product). This means preserving the invariant of Equation (3.10), which gives

$$\hat{\mathbf{q}}_k \odot \delta\mathbf{q}_k = \hat{\mathbf{q}}_k^+ \odot \delta\mathbf{q}_k^+. \tag{3.32}$$

Setting the post-reset error $\hat{\mathbf{x}}_k$ to zero, i.e. $\delta\mathbf{q}_k = \mathbf{I}_{\mathrm{q}}$, we get

$$\begin{aligned} \hat{\mathbf{q}}_k &= \hat{\mathbf{q}}_k^+ \odot \delta\mathbf{q}_k^+, \\ \hat{\mathbf{x}}_k &= \mathbf{0}. \end{aligned} \tag{3.33}$$

Whether or not the reset operation affects the error covariance is contested [10]. Assuming that it does not, then $\mathbf{P}_k = \mathbf{P}_k^+$. Otherwise, we model effect of the reset step on the covariance by computing the Jacobian of the reset step [4]. Suppose that the reset step rotates the a posteriori error $\delta\mathbf{q}_k^+$ by a small amount given by $\delta\tilde{\mathbf{q}}$. The resulting post-reset error $\delta\mathbf{q}_k$ is then given by

$$\delta\mathbf{q}_k = \delta\tilde{\mathbf{q}} \odot \delta\mathbf{q}_k^+. \tag{3.34}$$

Since the error (and hence the reset step) is always small, we may write the above in terms of the underlying state vector by using the linear approximation

$$\begin{bmatrix} 1 \\ \frac{\mathbf{x}_k}{2} \end{bmatrix} \approx \begin{bmatrix} 1 & -\frac{\tilde{\mathbf{x}}^{\mathrm{T}}}{2} \\ \frac{\tilde{\mathbf{x}}}{2} & \mathbf{I}_3 + \frac{1}{2}[\tilde{\mathbf{x}}\times] \end{bmatrix} \begin{bmatrix} 1 \\ \frac{\mathbf{x}_k^+}{2} \end{bmatrix}, \tag{3.35}$$

from which we can extract

$$\mathbf{x}_k \approx \tilde{\mathbf{x}} + \left(\mathbf{I}_3 + \frac{1}{2}[\tilde{\mathbf{x}}\times]\right)\mathbf{x}_k^+. \tag{3.36}$$

Given that $\mathbf{x}_k^+ \sim \mathcal{N}(\hat{\mathbf{x}}_k^+, \mathbf{P}_k^+)$, then by Equation (3.36) we have that $\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_k, \mathbf{P}_k)$, where the post-reset estimate $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$ are given by

$$\begin{aligned} \hat{\mathbf{x}}_k &= \tilde{\mathbf{x}} + \left(\mathbf{I}_3 + \frac{1}{2}[\tilde{\mathbf{x}}\times]\right)\hat{\mathbf{x}}_k^+, \\ \mathbf{P}_k &= \left(\mathbf{I}_3 + \frac{1}{2}[\tilde{\mathbf{x}}\times]\right)\mathbf{P}_k^+\left(\mathbf{I}_3 + \frac{1}{2}[\tilde{\mathbf{x}}\times]\right)^{\mathrm{T}}. \end{aligned} \tag{3.37}$$

Assuming that the error is reset to zero, $\hat{\mathbf{x}}_k = \mathbf{0}$, means that $\tilde{\mathbf{x}} = -\hat{\mathbf{x}}_k^+$, which gives the updated covariance

$$\mathbf{P}_k = \left(\mathbf{I}_3 - \frac{1}{2}[\hat{\mathbf{x}}_k^+\times]\right)\mathbf{P}_k^+\left(\mathbf{I}_3 - \frac{1}{2}[\hat{\mathbf{x}}_k^+\times]\right)^{\mathrm{T}}. \tag{3.38}$$

3.4 Independent estimation of direction vectors

An alternative to estimating the orientation directly, as the MEKF does, is to use Kalman filters to estimate the direction vectors $\mathbf{g}^{\mathrm{b}}$ and $\mathbf{m}^{\mathrm{b}}$ separately. The estimates can then be used to approximate a a solution to Equation (3.2). This approach has the advantage that it does not rely on quaternions, which sidesteps the issues with quaternions discussed earlier. It is also much simpler to implement and easier to understand than the MEKF. The formulation presented in this section follows the general strategy described in [7], while extending it to multiple vectors. In this thesis, this approach is referred to as the *vector Kalman filter* (VKF) to distinguish it from the MEKF. Figure 3.4 gives an overview of the VKF algorithm.

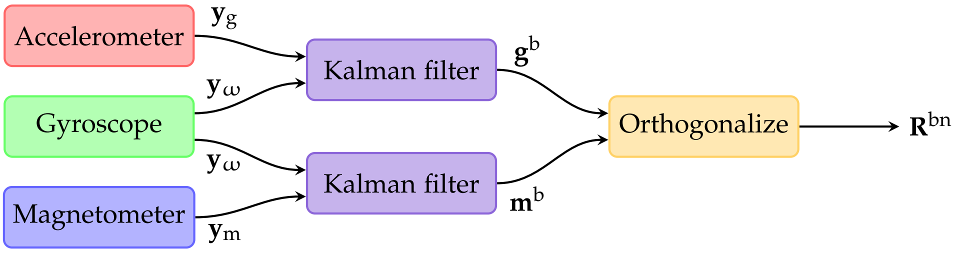
Accelerometer
Gyroscope
Magnetometer
$\mathbf{y}_{\mathrm{g}}$
$\mathbf{y}_{\omega}$
$\mathbf{y}_{\omega}$
$\mathbf{y}_{\mathrm{m}}$
Kalman filter
Kalman filter
$\mathbf{g}^{\mathrm{b}}$
$\mathbf{m}^{\mathrm{b}}$
Orthogonalize
$\mathbf{R}^{\mathrm{bn}}$

Figure 3.4: Schematic of the vector Kalman filter algorithm.

3.4.1 Rotational kinematics of a vector

Suppose that $\mathbf{x} = \mathbf{x}^{\mathrm{b}}$ is the vector to be estimated, where $\mathbf{x}^{\mathrm{n}}$ is constant. Using Equation (3.2) with the rotation $\mathbf{R}^{\mathrm{bn}} = \mathcal{R}(\mathbf{q})^{\mathrm{T}}$, we have by the definition of the quaternion rotation matrix that

$$\mathbf{x} = \mathcal{R}(\mathbf{q})^{\mathrm{T}}\mathbf{x}^{\mathrm{n}} = \mathcal{R}(\mathbf{q}^*)\mathbf{x}^{\mathrm{n}} = \mathbf{q}^* \odot \mathbf{x}^{\mathrm{n}} \odot \mathbf{q}, \tag{3.39}$$

where we identify the ordinary vectors in $\mathbb{R}^3$ with pure quaternions. Differentiating with respect to time and substituting the kinematic equation for $\mathbf{q}$ from Equation (3.7) gives

$$\begin{aligned}\dot{\mathbf{x}} &= \dot{\mathbf{q}}^* \odot \mathbf{x}^{\mathrm{n}} \odot \mathbf{q} + \mathbf{q}^* \odot \mathbf{x}^{\mathrm{n}} \odot \dot{\mathbf{q}} \\ &= \left(-\frac{1}{2}\boldsymbol{\omega} \odot \mathbf{q}^*\right) \odot \mathbf{x}^{\mathrm{n}} \odot \mathbf{q} + \mathbf{q}^* \odot \mathbf{x}^{\mathrm{n}} \odot \left(\frac{1}{2}\mathbf{q} \odot \boldsymbol{\omega}\right) \\ &= \frac{1}{2}(\mathbf{x} \odot \boldsymbol{\omega} - \boldsymbol{\omega} \odot \mathbf{x}).\end{aligned} \tag{3.40}$$

Writing the above in matrix form gives

$$\dot{\mathbf{x}} = \frac{1}{2}([\boldsymbol{\omega} \otimes] - [\boldsymbol{\omega} \odot])\mathbf{x} = -[\boldsymbol{\omega} \times]\mathbf{x}. \tag{3.41}$$

Substituting the gyroscope measurement model equation for $\boldsymbol{\omega}$ and rearranging then gives the dynamic model

$$\dot{\mathbf{x}} = -[\mathbf{y}_\omega \times]\mathbf{x} - [\mathbf{x} \times]\mathbf{w}_\omega, \tag{3.42}$$

which is linear in state $\mathbf{x}$ but non-linear in noise. Note that the cross product of the noise with $\mathbf{x}$ makes the noise contribution perpendicular to $\mathbf{x}$. This makes sense, as the rotational kinematics of the state vector $\mathbf{x}$ cannot possibly change its length. Linearizing this equation at the point $(\mathbf{x}, \mathbf{w}_\omega) = (\hat{\mathbf{x}}, \mathbf{0})$ gives

$$\dot{\mathbf{x}} \approx -[\mathbf{y}_\omega \times]\mathbf{x} - [\hat{\mathbf{x}} \times]\mathbf{w}_\omega. \tag{3.43}$$

Discretizing the above with a zero-order hold $\mathbf{y}_{\omega,k} = \mathbf{y}_\omega(t_k)$ gives

$$\begin{aligned} \mathbf{F}_{k-1} &= e^{-[\mathbf{y}_{\omega,k} \times]\Delta t}, \\ \mathbf{v}_{k-1} &= -\int_{t_{k-1}}^{t_k} e^{-[\mathbf{y}_{\omega,k} \times](t_k-\tau)}[\hat{\mathbf{x}}_{k-1} \times]\mathbf{w}_\omega(\tau)\, d\tau. \end{aligned} \tag{3.44}$$

The discrete process noise covariance is then given by

$$\mathbf{Q}_{k-1} = \int_{t_{k-1}}^{t_k} e^{-[\mathbf{y}_{\omega,k} \times](t_k-\tau)}[\hat{\mathbf{x}}_{k-1} \times]\mathbf{S}_\omega[\hat{\mathbf{x}}_{k-1} \times]^{\mathrm{T}}\left(e^{-[\mathbf{y}_{\omega,k} \times](t_k-\tau)}\right)^{\mathrm{T}} d\tau, \tag{3.45}$$

assuming that $\mathbf{w}_\omega$ is Gaussian white noise. Assuming further that the noise is uniform and uncorrelated across the gyroscope axes, that is, $\mathbf{S}_\omega = \sigma_\omega^2\mathbf{I}_3$, and also that $t_k - \tau \leq \Delta t$ is small, we obtain the approximation

$$\mathbf{Q}_{k-1} \approx \int_{t_{k-1}}^{t_k} [\hat{\mathbf{x}}_{k-1} \times]\sigma_\omega^2[\hat{\mathbf{x}}_{k-1} \times]^{\mathrm{T}}\, d\tau = -\sigma_\omega^2\Delta t[\hat{\mathbf{x}}_{k-1} \times]^2. \tag{3.46}$$

3.4.2 Kalman filter for vector estimation

Two vector estimates are needed to determine orientation uniquely, as discussed in Section 3.1. The following filter formulation uses the same gravity and magnetic field vector measurements as the MEKF. From the perspective of the Kalman filter, the estimates are completely independent and can be implemented as a pair of concurrent Kalman filters. However, for the sake of presentation, we combine them into a single filter with the 6-dimensional state vector

$$\mathbf{x} \triangleq \begin{bmatrix} \mathbf{g}^{\mathrm{b}} \\ \mathbf{m}^{\mathrm{b}} \end{bmatrix}. \tag{3.47}$$

The above derivations form the dynamic model of a state-space representation where the state is the estimated vector $\mathbf{g}^{\mathrm{b}}$ or $\mathbf{m}^{\mathrm{b}}$. Because both vectors are driven by the same dynamics, the dynamic model derived above is simply duplicated, giving

$$\begin{aligned}\mathbf{F}_{k-1} &\triangleq \begin{bmatrix} e^{-[\mathbf{y}_{\omega,k}\times]\Delta t} & \mathbf{0} \\ \mathbf{0} & e^{-[\mathbf{y}_{\omega,k}\times]\Delta t} \end{bmatrix}, \\ \mathbf{Q}_{k-1} &\triangleq -\sigma_\omega^2 \Delta t \begin{bmatrix} [\hat{\mathbf{x}}_{k-1}\times]^2 & \mathbf{0} \\ \mathbf{0} & [\hat{\mathbf{x}}_{k-1}\times]^2 \end{bmatrix}. \end{aligned} \tag{3.48}$$

The sensor measurements are now a direct measurement of the state (save for a difference in sign for the gravity measurement). Therefore, the measurement model is trivially given by combining Equations (3.3) and (3.4), resulting in

$$\mathbf{H}_k \triangleq \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix}, \qquad \mathbf{R}_k \triangleq \begin{bmatrix} \boldsymbol{\Sigma}_{\mathrm{g}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\mathrm{m}} \end{bmatrix}. \tag{3.49}$$

The above definitions can be used as-is with Equation (2.15) to form the Kalman filter for tracking the gravity and magnetic field vectors.

It is interesting to note the differences between this estimator which has a 6-dimensional state space, and the MEKF which has a 3-dimensional state space. The 3 additional dimensions of the vector estimator can be thought of as accounting for 3 additional degrees of freedom: the magnitude of both vectors as well as the angle between them. These quantities were assumed static and known beforehand in the MEKF, possibly determined by an offline calibration process. This additional flexibility in the vector estimator means that the quantities do not need to be determined or assumed beforehand. However, it can also be a source of problems. For example, in places with a steep magnetic inclination or a strong magnetic field disturbance, the magnetic field vector estimate may become almost parallel with the gravity vector estimate, which may cause instability in orientation estimates derived from them.

Also worth noting is the fact that this filter formulation is nearly linear, with the only non-linearity being due to the way the gyroscope noise couples to the state. It is in fact possible to use a simple linear additive noise term instead of the non-linear one used here, which makes the state-space model entirely linear (see [7]). This makes the algorithm computationally even cheaper compared to the MEKF as well as simplifying analysis of the filter itself.

3.4.3 Estimating orientation from a pair of vectors

To complete the algorithm, we need to find an estimate of the orientation matrix $\mathbf{R}^{\mathrm{bn}}$ based on the estimates of $\mathbf{g}^{\mathrm{b}}$ and $\mathbf{m}^{\mathrm{b}}$. *Wahba's problem* [15] formulates the task as a non-linear regression problem with the cost function

$$C(\mathbf{R}^{\mathrm{bn}}) = \frac{1}{2}\sum_{i=1}^{n} a_i \left\| \mathbf{u}_i^{\mathrm{b}} - \mathbf{R}^{\mathrm{bn}}\mathbf{u}_i^{\mathrm{n}} \right\|^2, \tag{3.50}$$

where $\mathbf{u}_i$ are unit vectors and a_i are weights expressing the relative importance of each vector. In the case of $\mathbf{g}^{\mathrm{b}}$ and $\mathbf{m}^{\mathrm{b}}$, the cost function becomes

$$C(\mathbf{R}^{\mathrm{bn}}) = \frac{a_{\mathrm{g}}}{2}\left\| \mathbf{g}^{\mathrm{b}} - \mathbf{R}^{\mathrm{bn}}\mathbf{g}^{\mathrm{n}} \right\|^2 + \frac{a_{\mathrm{m}}}{2}\left\| \mathbf{m}^{\mathrm{b}} - \mathbf{R}^{\mathrm{bn}}\mathbf{m}^{\mathrm{n}} \right\|^2. \tag{3.51}$$

However, in the case of only two vectors, it is possible to use a simpler procedure based on orthonormalizing the frame made of $\mathbf{g}$, $\mathbf{m}$, and a third vector perpendicular to both $\mathbf{g}$ and $\mathbf{m}$. Going by the definition of the navigation frame axis directions, we choose the z axis to point upwards, y to point toward the magnetic north and x to point toward the east. Then, the positive z direction of the orientation frame is opposite to gravity and is given by

$$\mathbf{e}_{\mathrm{z}} \triangleq -\frac{\mathbf{g}^{\mathrm{b}}}{\left\|\mathbf{g}^{\mathrm{b}}\right\|}, \tag{3.52}$$

the positive x direction is east, orthogonal to $\mathbf{m}^{\mathrm{b}}$ and $\mathbf{e}_{\mathrm{z}}$, and is given by

$$\mathbf{e}_{\mathrm{x}} \triangleq \frac{\mathbf{m}^{\mathrm{b}} \times \mathbf{e}_{\mathrm{z}}}{\left\|\mathbf{m}^{\mathrm{b}} \times \mathbf{e}_{\mathrm{z}}\right\|}, \tag{3.53}$$

and lastly the positive y direction is given by the right hand rule as

$$\mathbf{e}_{\mathrm{y}} \triangleq \mathbf{e}_{\mathrm{z}} \times \mathbf{e}_{\mathrm{x}}. \tag{3.54}$$

The axes $\mathbf{e}_{\mathrm{x}}$, $\mathbf{e}_{\mathrm{y}}$ and $\mathbf{e}_{\mathrm{z}}$ are orthonormal and form the column vectors of a rotation matrix $\mathbf{R}^{\mathrm{bn}}$ representing a change from the navigation frame to the body frame. Therefore

$$\mathbf{R}^{\mathrm{bn}} \triangleq \begin{bmatrix} \mathbf{e}_{\mathrm{x}} & \mathbf{e}_{\mathrm{y}} & \mathbf{e}_{\mathrm{z}} \end{bmatrix} = \begin{bmatrix} -\frac{\mathbf{m}^{\mathrm{b}}\times\mathbf{g}^{\mathrm{b}}}{\|\mathbf{m}^{\mathrm{b}}\times\mathbf{g}^{\mathrm{b}}\|} & \frac{\mathbf{g}^{\mathrm{b}}\times(\mathbf{m}^{\mathrm{b}}\times\mathbf{g}^{\mathrm{b}})}{\|\mathbf{g}^{\mathrm{b}}\|\|\mathbf{m}^{\mathrm{b}}\times\mathbf{g}^{\mathrm{b}}\|} & -\frac{\mathbf{g}^{\mathrm{b}}}{\|\mathbf{g}^{\mathrm{b}}\|} \end{bmatrix}, \tag{3.55}$$

which is a solution to Equation (3.2) when the gravity and magnetic field vectors in the navigation frame are defined as

$$\mathbf{g}^{\mathrm{n}} \triangleq \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}, \qquad \mathbf{m}^{\mathrm{n}} \triangleq \begin{bmatrix} 0 \\ B\cos\theta \\ -B\sin\theta \end{bmatrix}, \tag{3.56}$$

where g is the gravitational acceleration, B is the magnetic field strength, and θ is the magnetic field inclination.

4 Implementation and testing

This chapter covers the implementation and testing of the Kalman filter models derived in Chapter 3. Section 4.1 gives a general overview of the hardware and software used to implement and test the algorithms in this thesis. Section 4.2 gives a brief description of the issues related to sensor calibration and how those issues were addressed in the testing. Section 4.3 presents the test setup and discusses the results.

4.1 Overview of the hardware and software

4.1.1 Sensor board

The experimental test device consists of an MPU-9250 inertial measurement unit (IMU) on a breakout board mounted on a solderless breadboard along with the necessary interface circuitry (see Figure 4.1). The MPU-9250 is a microelectromechanical (MEMS) IMU developed by InvenSense Inc. which incorporates a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer [16, 17]. Small, lightweight and affordable, it can be found in many mobile devices, including smartphones and tablets. The MPU-9250 is an integrated solution that also incorporates a proprietary sensor fusion algorithm for orientation estimation. Low-pass filtering for raw measurement data is also supported [16]. For the purposes of testing out the Kalman filter algorithms, all of these advanced capabilities were disabled in order to gain access to the raw noisy measurement data.

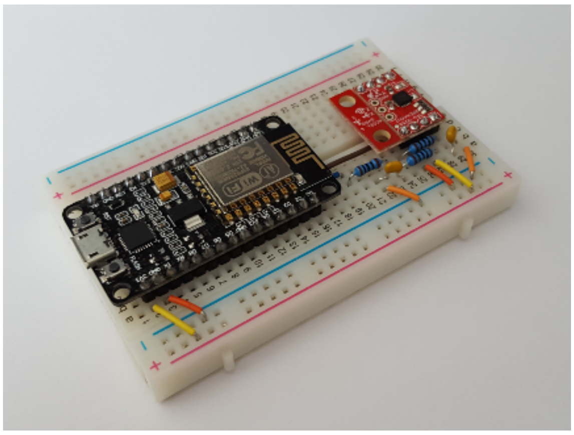

Figure 4.1: Experimental test device. On the left, a NodeMCU ESP8266 breakout board with an integrated Micro-USB connector. On the right, a SparkFun breakout board of the InvenSense MPU-9250 MEMS IMU.

The sensor board software running on the ESP8266 microcontroller is responsible for initializing the MPU-9250 sensor and polling the sensor for measurements at a configurable sampling rate. The measurements are formatted as text and transmitted in real time over a USB serial line to the host computer, where they may be processed. The communication topology is shown in Figure 4.2. The board itself does no processing on the measurements other than applying factory-calibrated scaling factors to magnetometer measurements. The ESP8266 needs to be reprogrammed only when the sampling rate or sensor sensitivity settings are altered. Using a text-based interface between the sensor board and host computer simplifies operations, as the sensor measurements can be accessed and manipulated using standard command line tools.

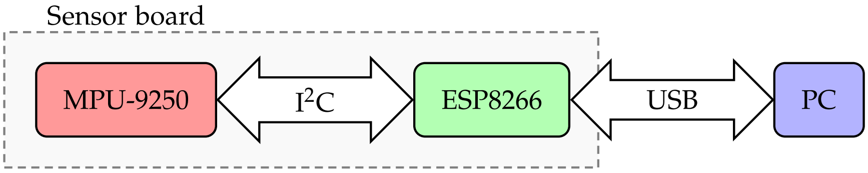

Figure 4.2: Hardware communication topology.

4.1.2 Kalman filtering and visualization UI

A real-time filtering and visualization application was developed to aid the testing and tuning of the filter algorithms. A real-time display helps to develop intuition about the behavior of the filters through immediate feedback. Figure 4.3 shows a screenshot of the graphical user interface of the application. On the right is a 3-by-3 grid of graphs showing real-time raw measurement data from the acceleration, magnetic field, and angular rate sensors. On the left are coordinate system widgets which present the axes of a coordinate system (red, green, and blue for x, y, and z, respectively).

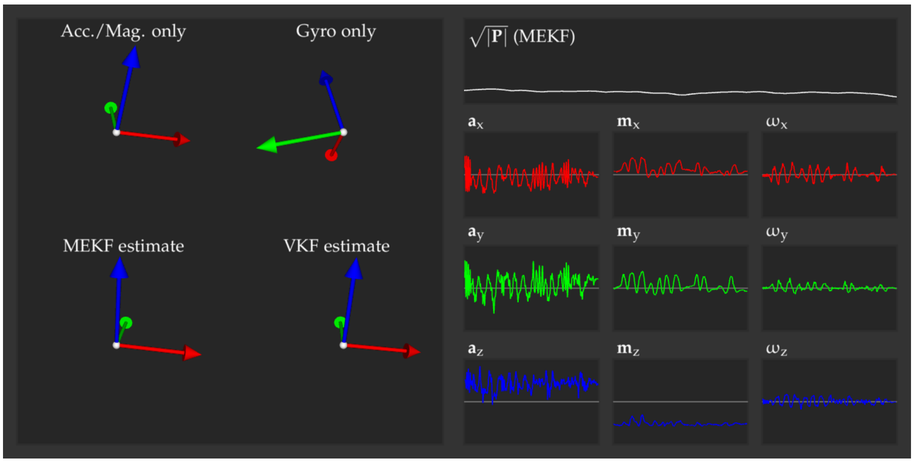

Figure 4.3: Screenshot of the orientation tracking application UI.

Four widgets corresponding to four different orientation estimates are shown. In the top left corner is the orientation estimate derived from raw accelerometer and magnetometer measurements according to the orthonormalization procedure described in Section 3.4. In the top right corner is the estimate obtained by directly integrating gyroscope measurements. The bottom row shows the orientation estimates obtained with the two approaches described in this thesis. In addition, the white graph on the top right shows a measure of the error covariance of the MEKF.

4.2 Sensor calibration

Measurements made by real sensors are be subject to a number of effects that affect their quality, including noise, bias, drift, non-linear response, hysteresis, and temperature sensitivity [18]. The measurement models introduced so far include only a noise term. To make best use of the sensor measurements, the remaining effects must also be compensated for. Static effects, such as a constant bias or non-linear response curve, are usually compensated for by an offline calibration procedure. Dynamic effects like a drifting bias or hysteresis require active compensation. Practical Kalman filters used for orientation estimation are often augmented with a bias estimate for this reason [10].

Multi-axis sensors present further challenges. Due to manufacturing inaccuracies, the sensitive axes of the sensor triad are not completely orthogonal, nor are their sensitivities exactly the same. These give rise to relative scaling errors and cross-axis sensitivity in the measurements [18]. These effects are typically treated as static, and can be expressed as a linear map on a 3-axis measurement triplet determined by an offline calibration process [19].

When fusing data from multiple sensors, it is also important to consider the relative alignment (i.e., orientation) of the different sensor triads with respect to each other and the body frame of the device where the sensors are mounted. Several algorithms have been developed to calibrate the orientation of magnetometer and gyroscope frames with respect to a calibrated accelerometer frame [19, 20]. Compensating for the alignment of the sensor frames with respect to the body frame is particularly important in control applications, where the position of the enclosing device in the surrounding space is important.

Magnetometer calibration is further complicated by the presence of environmental effects, as it is used to measure the relatively weak geomagnetic field. Magnetic fields are perturbed both by external field sources and surrounding objects. In the context of magnetometer calibration, these are known as hard-iron and soft-iron errors, respectively. When the sources of these perturbations are fixed relative to the sensor triad, it is possible to compensate for them. If the sources are not fixed, then the problem becomes difficult or even impossible. [19]

Due to time constraints, the approach to calibration in this experimental test was cursory. Only constant bias and scaling errors were calibrated for in the case of the accelerometer and magnetometer using a standard ellipsoid-fit approach. Gyroscope calibration involved finding a time-averaged estimate of a constant bias, with the device held completely still. Axis misalignment was neglected. Relative alignment errors between sensor triads were also neglected. However, part of the reason for choosing the MPU-9250 was that it integrates all three sensors in one package, providing better confidence in correct alignment than with discrete sensors.

4.3 Testing and results

Both of the implemented algorithms were evaluated using 4 different test cases, a rotation test, a stability test, a convergence test, and a swing test. All tests were performed using the same measurement data and noise variances for both algorithms. Two experimentally determined sets of noise parameters were used for the accelerometer and magnetometer. Although it is possible to obtain values for the noise variances from the datasheet or by statistical analysis of actual measurements, testing shows that an empirical approach seems to give more satisfactory results.

Table 1 gives the parameters used for each sensor. The sampling rate of the sensors was 50 Hz, and the update rate of the Kalman filters was equal to the sampling rate. The MEKF is parameterized by the assumed gravity and magnetic field vectors in the navigation frame. These measurements were normalized to unity by the calibration procedure. The assumed angle between the vectors was set to 17°, derived from the reported magnetic field inclination of approximately 73° for Espoo, Finland [21] where the tests were performed.

Parameter	Magnetometer	Accelerometer	Gyroscope
Range	±4912 µT	$\pm 4\,g$	±1000 °/s
Resolution	150 nT	$122\,\mu g$	0.03 °/s
RMS noise (A)	4.8 µT	$0.1\,g$	0.8 °/s
RMS noise (B)	48 µT	$1.0\,g$	0.8 °/s

Table 1: Sensor parameters used in the test ($g = 9.81\,\mathrm{m/s^2}$).

Testing indicates that both algorithms are capable of producing stable orientation estimates with no major differences in the quality. The MEKF appears to naturally favor the gyroscope more, whereas the VKF is more sensitive to the accelerometer and magnetometer. However, comparable results can be achieved by individually tuning the measurement model covariances for both algorithms.

The following subsections describe each test in more detail. For convenience, the orientation given from raw accelerometer and magnetometer measurements is referred to as the *reference* in the text. All plots are of the components of the axis-angle rotation vector corresponding to the orientation estimates. The x, y, and z components of the vector are shown in red, green, and blue, respectively. The vertical axis is in radians and the horizontal axis is in seconds.

4.3.1 Rotation test

In this test, the device was rotated 90 degrees about its x axis and then rotated back to its initial orientation. The rotations were repeated for the y and z axes. The sensor was rotated as close to its center as possible in order to avoid distortion due to linear acceleration. This test was used to verify that both algorithms are functioning correctly, and to highlight differences between the estimates in close to ideal conditions. Finally, the test demonstrates the value of Kalman filtering compared to estimating orientation from raw accelerometer and magnetometer measurements.

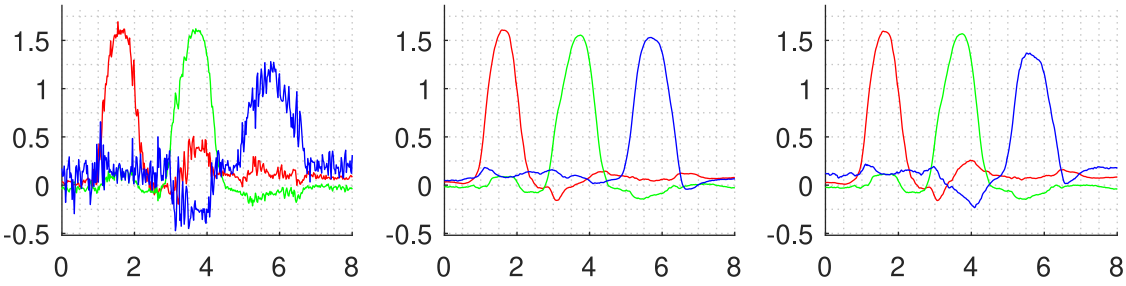

Figure 4.4: Components of the rotation vector in the rotation test using smaller measurement noise variances. Left: Raw accelerometer/magnetometer estimate. Middle: MEKF estimate. Right: VKF estimate.

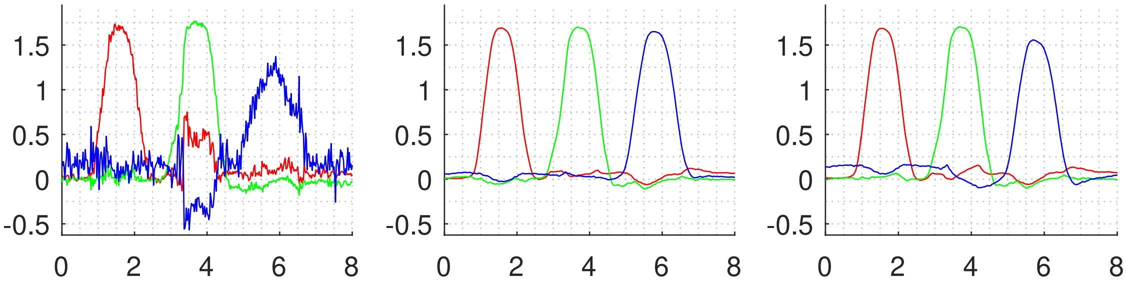

Figure 4.5: Components of the rotation vector in the rotation test using larger measurement noise variances. Left: Raw accelerometer/magnetometer estimate. Middle: MEKF estimate. Right: VKF estimate.

Results of the rotation test are shown in Figures 4.4 and 4.5. It is immediately apparent that the quality of the orientation estimate is drastically improved in the filtering approaches. The rotation around the z axis is especially problematic due to the high magnetic inclination, reaching only a peak of around 65 degrees in the raw estimate, when the actual rotation was close to 90 degrees. The MEKF is able to track orientation quite well despite the poor reference. The VKF also performs decently but is more sensitive to the reference, as seen from the slight deviations at 4 seconds and the slightly depressed z rotation peak at around 6 seconds in Figure 4.4. This problem is somewhat mitigated by increasing the measurement variances, as shown by Figure 4.5.

4.3.2 Stability test

This test consists of simply holding the device stationary to investigate the stability of the orientation estimate. The properties investigated were noise in the orientation estimate, as well as long-term stability. Stability was demonstrated by comparing the filter estimates to the estimate obtained by integrating gyroscope measurements.

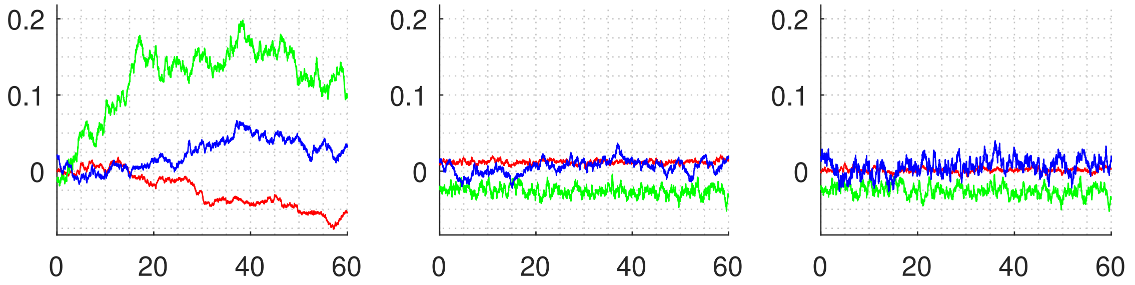

Figure 4.6: Components of the rotation vector in the stability test using smaller measurement noise variances. Left: Raw gyroscope estimate. Middle: MEKF estimate. Right: VKF estimate.

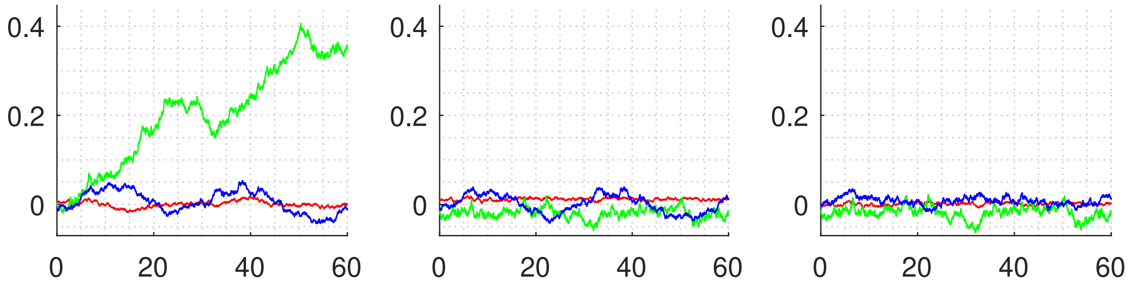

Figure 4.7: Components of the rotation vector in the stability test using larger measurement noise variances. Left: Raw gyroscope estimate. Middle: MEKF estimate. Right: VKF estimate.

Figures 4.6 and 4.7 show the results of the stability test. The gyroscope estimate shows significant drift after a few seconds. Both the MEKF and the VKF are able to keep the estimate from drifting, although there is noticeable noise in the estimate. The z axis of the rotation vector is particularly a victim of the gyroscope drift locally. The z axis noise of the VKF is higher than the MEKF, as it seems to be more sensitive to the noise in the reference. The other axes are nearly identical. Increasing the measurement variances visibly improves the stability of the VKF estimate and reduces noise in both algorithms, although in the MEKF estimate the z component is now heavily influenced by the gyroscope noise.

4.3.3 Convergence test

The convergence test compares the response speed of the algorithms when the current estimate is suddenly brought out of alignment with the reference given by accelerometer and magnetometer measurements. Starting from alignment with the navigation frame, the device was rotated 90 degrees counterclockwise around the x axis and then 90 degrees counterclockwise around the rotated y axis. The device was kept still in this orientation. Then, both filter states were reset to identity (alignment with navigation frame) without altering their covariance matrices. The test demonstrates the difference in sensitivity of both algorithms to changes in the reference.

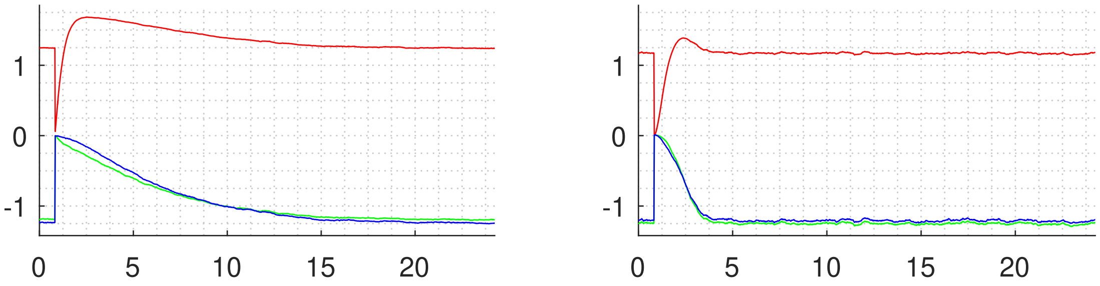

Figure 4.8: Components of the rotation vector in the convergence test using smaller measurement noise variances. Left: MEKF estimate. Right: VKF estimate.

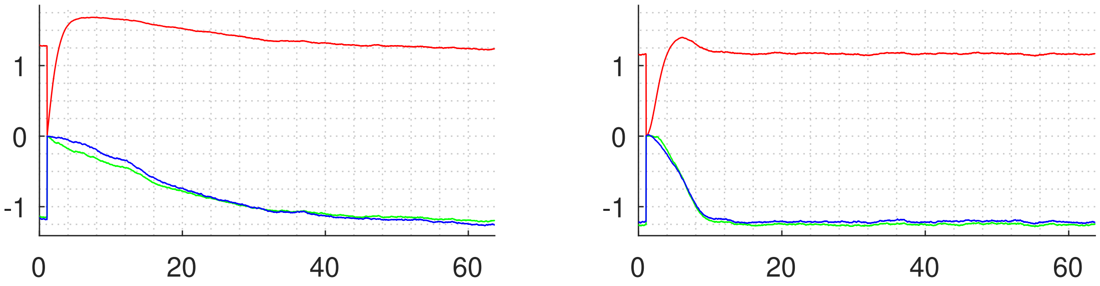

Figure 4.9: Components of the rotation vector in the convergence test using larger measurement noise variances. Left: MEKF estimate. Right: VKF estimate.

The results are shown in Figures 4.8 and 4.9. As expected from the above tests, the vector-based filter converges relatively quickly compared to the MEKF. The

MEKF estimate rotates 90 degrees around the x axis quickly which corrects the attitude error, but it takes significantly longer to correct for the heading error. The convergence is slower in both algorithms as the measurement variances are increased, but otherwise similar.

4.3.4 Swing test

This test involved suspending the device by a wire and swinging it like a pendulum. This test is particularly challenging for the accelerometer, as the device is in free fall except for the radial tension force exerted by the wire. Therefore, the only acceleration measured by the device is due to the tension force which has a constant direction in the body frame, and the direction of gravity is not available. The test rewards reliance on the gyroscope. The test was performed quickly in order to avoid excessive drift in the gyroscope estimate that was used as a baseline.

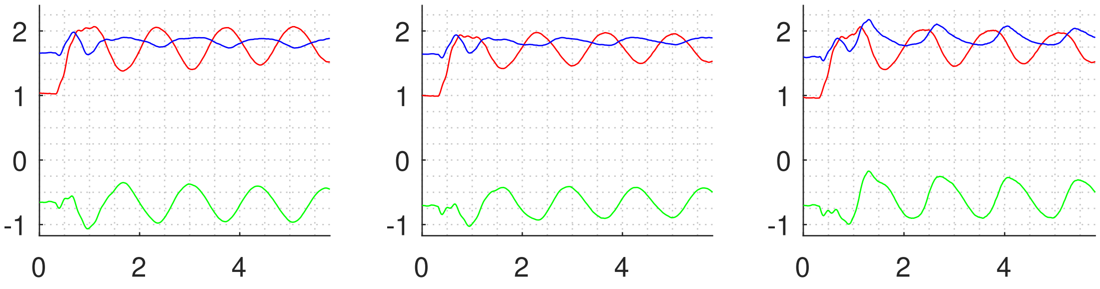

Figure 4.10: Components of the rotation vector in the swing test using smaller measurement noise variances. Left: Raw gyroscope estimate. Middle: MEKF estimate. Right: VKF estimate.

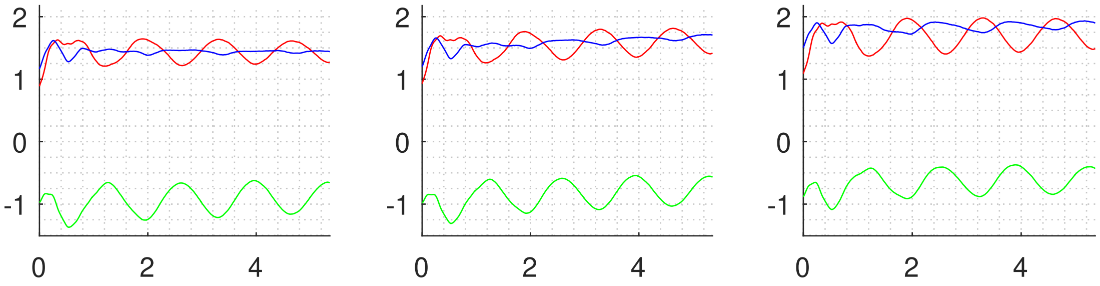

Figure 4.11: Components of the rotation vector in the swing test using larger measurement noise variances. Left: Raw gyroscope estimate. Middle: MEKF estimate. Right: VKF estimate.

Results of the swing test are shown in Figures 4.10 and 4.11. At small measurement variances, the MEKF does noticeably better than the VKF. The VKF eagerly follows the reference which in this case is giving completely bogus information about the orientation. This is causes noticeable precession in the orientation estimate, as shown by the wave-like pattern of the z component in the VKF estimate. Once again, increasing the measurement variance helps somewhat in reducing reliance on the reference.

5 Conclusions

In this thesis, two Kalman filter based algorithms for orientation tracking were introduced, implemented, and experimentally evaluated. The first algorithm, known as the multiplicative extended Kalman filter (MEKF), uses an indirect quaternion-based method. The second algorithm, referred to in this thesis as the vector Kalman filter (VKF) uses a vector-based method. Both algorithms implement a sensor fusion approach to estimating orientation that utilizes measurements from a magnetic field, gravity, and angular rate sensor system.

The focus of this thesis was on establishing the essentials of Kalman filter based orientation tracking from the ground up, starting with the derivation of the Kalman filter itself. Therefore, it was necessary to leave out many potentially interesting topics such as Kalman smoothing, non-Gaussian noise models, parameter estimation, and advanced sensor calibration, among others. Many of these topics include challenges that need to be addressed in the design of practical orientation tracking systems.

The implemented algorithms were tested using a custom-built sensor device. Experimental testing shows that both algorithms are capable of producing stable orientation estimates that are considerably better than those of a naive approach relying only on raw accelerometer and magnetometer measurements. The VKF appears more responsive than the MEKF to changes in the reference directions given by the accelerometer and magnetometer measurements when using identical sensor measurement model variances. By tuning the noise variances of both algorithms independently, it appears possible to attain comparable results from both.

Although both algorithms solve the same problem, there are notable differences between them. The MEKF assumes prior knowledge of the magnitude of the gravity and magnetic field vectors, as well as the angle between them, which depends on the magnetic inclination. Therefore, these quantities must be determined beforehand, for example as part of a calibration procedure. The VKF, owing to its construction, is capable of estimating these quantities as part of its normal operation. Furthermore, although both the MEKF and VKF are technically non-linear, the only non-linearity in the VKF is due to the coupling of the gyroscope noise with the state in the dynamic model. A fully linear filter model is possible with only a minor sacrifice in the modeling of the process noise.

The presented theory provides a rich basis for further explorations in the subject of orientation tracking. Of particular interest is the use of non-Gaussian noise models to better approximate the real world behavior of sensor measurements. Some of the more recent developments in non-linear Kalman filtering theory could prove useful in this endeavor. More advanced sensor calibration techniques including inter-triad misalignment compensation and online calibration could be used to improve the algorithms.

References

[1] M. Perlmutter and L. Robin. "High-performance, low cost inertial MEMS: A market in motion!" In: *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*. Apr. 2012, pp. 225–229. DOI: 10.1109/PLANS.2012.6236884.

[2] G. Lammel. "The future of MEMS sensors in our connected world". In: *2015 28th IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*. Jan. 2015, pp. 61–64. DOI: 10.1109/MEMSYS.2015.7050886.

[3] V. M. N. Passaro et al. "Gyroscope technology and applications: A review in the industrial perspective". In: *Sensors (Switzerland)* 17.10 (2017). ISSN: 14248220. DOI: 10.3390/s17102284.

[4] M. Kok. "Probabilistic modeling for positioning applications using inertial sensors". Licentiate's Thesis. Linköping University, Linköping, Sweden, 2014.

[5] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[6] J. L. Crassidis, F. L. Markley, and Y. Cheng. "Survey of nonlinear attitude estimation methods". In: *Journal of Guidance, Control, and Dynamics* 30.1 (2007), pp. 12–28. ISSN: 07315090. DOI: 10.2514/1.22452.

[7] S. Särkkä et al. "Adaptive Kalman filtering and smoothing for gravitation tracking in mobile systems". In: *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Oct. 2015, pp. 1–7. DOI: 10.1109/IPIN.2015.7346762.

[8] D. Simon. *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*. Hoboken, NJ, USA: Wiley-Interscience, 2006, p. 526. ISBN: 9780471708582.

[9] S. Särkkä. *Bayesian filtering and smoothing*. Institute of Mathematical Statistics textbooks, vol. 3. Cambridge, UK: Cambridge University Press, 2013, p. 232. ISBN: 9781107030657.

[10] F. L. Markley and J. L. Crassidis. *Fundamentals of spacecraft attitude determination and control*. New York, NY, USA: Springer New York, 2014, p. 486. ISBN: 9781493908028. DOI: 10.1007/978-1-4939-0802-8.

[11] M. W. Hirsch, S. Smale, and R. L. Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier Inc., 2013. ISBN: 9780123820105.

[12] T. Karvonen. "Stability of linear and non-linear Kalman filters". Master's Thesis. University of Helsinki, Finland, 2014.

[13] A. Noureldin, T. B. Karamat, and J. Georgy. *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Berlin Heidelberg, 2013, p. 313. ISBN: 9783642304668. DOI: 10.1007/978-3-642-30466-8.

[14] J. Solà. "Quaternion kinematics for the error-state Kalman filter". In: *Computing Research Repository* abs/1711.02508 (2017).

[15] G. Wahba. "A Least Squares Estimate of Satellite Attitude". In: *SIAM Review* 7.3 (1965), pp. 409–409. DOI: 10.1137/1007077.

[16] InvenSense Inc. *MPU-9250 Product Specification. MPU-9250 datasheet*. Dec. 2013. Revised June 2016.

[17] Asahi Kasei Microdevices. *3-axis Electronic Compass. AK8963 datasheet*. Oct. 2013.

[18] J. G. Webster. *The measurement, instrumentation, and sensors handbook*. Boca Raton, FL, USA: CRC Press, 1999. ISBN: 0849383471.

[19] M. Kok and T. B. Schon. "Magnetometer calibration using inertial sensors". In: *IEEE Sensors Journal* 16.14 (2016), pp. 5679–5689. ISSN: 1530437X. DOI: 10.1109/JSEN.2016.2569160.

[20] D. Tedaldi, A. Pretto, and E. Menegatti. "A robust and easy to implement method for IMU calibration without external equipments". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. May 2014, pp. 3042–3049. DOI: 10.1109/ICRA.2014.6907297.

[21] National Oceanic and Atmospheric Administration. *NCEI Geomagnetic Calculators*. 2018. URL: https://www.ngdc.noaa.gov/geomag-web/#igrfwmm (visited on 2018-04-27).

[22] W. R. Hamilton. "XXII. On quaternions; or on a new system of imaginaries in algebra". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 25 (1844), pp. 10–13.

A Mathematical preliminaries

This appendix contains some of the mathematical results that are relevant to the derivations presented in this thesis.

A.1 Multivariate Gaussian distributions

Multivariate probability distributions arise naturally when considering the simultaneous probability involving two or more scalar variables. This includes joint distributions of conceptually separate quantities as well as distributions of naturally vector-like quantities. This section gives a brief introduction to multivariate Gaussian distributions and presents some results that are useful in the derivation of the Kalman filter.

A.1.1 Random vectors

We introduce the extension of continuous random variables in $\mathbb{R}$ into random vectors in the Euclidean space $\mathbb{R}^n$. Consider a set of jointly distributed random variables $x_i : \Omega \to \mathbb{R}$ where $i = 0, 1, ..., n$. We may define the vector

$$\mathbf{x} \triangleq \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \tag{A1}$$

where $\mathbf{x} : \Omega \to \mathbb{R}^n$ is a *multivariate random variable*. Intuitively, given expected values $\mathrm{E}[x_i] = \mu_i$, the expected value of $\mathbf{x}$ is

$$\boldsymbol{\mu} \triangleq \mathrm{E}[\mathbf{x}] = \begin{bmatrix} \mathrm{E}[x_1] \\ \vdots \\ \mathrm{E}[x_n] \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix}. \tag{A2}$$

Since x_i are jointly distributed, in order to define the variance of $\mathbf{x}$, we need to consider the covariances $\sigma_{ij} = \mathrm{cov}(x_i, x_j)$ in addition to the variances σ_{ii} of the individual components. We define the *covariance matrix* $\boldsymbol{\Sigma}$ as

$$\begin{aligned}\boldsymbol{\Sigma} &\triangleq \mathrm{E}\Big[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\Big] \\ &= \begin{bmatrix} \mathrm{E}\big[(x_1-\mu_1)^2\big] & \cdots & \mathrm{E}\big[(x_1-\mu_1)(x_n-\mu_n)\big] \\ \vdots & \ddots & \vdots \\ \mathrm{E}\big[(x_n-\mu_n)(x_1-\mu_1)\big] & \cdots & \mathrm{E}\big[(x_n-\mu_n)^2\big] \end{bmatrix} \\ &= \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma_{nn} \end{bmatrix}.\end{aligned} \tag{A3}$$

Noting that $\sigma_{ij} = \sigma_{ji}$, we see that $\boldsymbol{\Sigma}$ is symmetric. We can now write $\mathbf{x} \sim (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for $\mathbf{x}$ distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, and $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ if $\mathbf{x}$ is Gaussian-distributed. The probability density function of the multivariate Gaussian distribution is

$$\mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\bigg(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\bigg). \tag{A4}$$

A.1.2 Marginal and conditional distributions

Suppose that $\mathbf{x} \sim p(\mathbf{x})$ and that $\mathbf{z}$ depends on $\mathbf{x}$ such that $\mathbf{z} \,|\, \mathbf{x} \sim p(\mathbf{z} \,|\, \mathbf{x})$. Then the unconditional distribution of $\mathbf{z}$ is the marginal distribution of the joint distribution of $\mathbf{x}$ and $\mathbf{z}$,

$$p(\mathbf{z}) = \int p(\mathbf{x}, \mathbf{z}) \, d\mathbf{x} = \int p(\mathbf{z} \,|\, \mathbf{x}) p(\mathbf{x}) \, d\mathbf{x}, \tag{A5}$$

and the conditional distribution of $\mathbf{x}$ given $\mathbf{z}$ is given by Bayes' theorem

$$p(\mathbf{x} \,|\, \mathbf{z}) = \frac{p(\mathbf{z} \,|\, \mathbf{x}) p(\mathbf{x})}{p(\mathbf{z})}. \tag{A6}$$

If $p(\mathbf{x})$ and $p(\mathbf{z} \,|\, \mathbf{x})$ are Gaussian distributions of the form

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x} \,|\, \mathbf{m}, \mathbf{P}), \\ p(\mathbf{z} \,|\, \mathbf{x}) &= \mathcal{N}(\mathbf{z} \,|\, \mathbf{A}\mathbf{x} + \mathbf{b}, \boldsymbol{\Sigma}), \end{aligned} \tag{A7}$$

then $p(\mathbf{z})$ and $p(\mathbf{x} \,|\, \mathbf{z})$ are also Gaussian and of the form

$$\begin{aligned} p(\mathbf{z}) &= \mathcal{N}(\mathbf{z} \,|\, \mathbf{A}\mathbf{m} + \mathbf{b}, \mathbf{A}\mathbf{P}\mathbf{A}^{\mathrm{T}} + \boldsymbol{\Sigma}), \\ p(\mathbf{x} \,|\, \mathbf{z}) &= \mathcal{N}(\mathbf{x} \,|\, \mathbf{m}', \mathbf{P}'), \end{aligned} \tag{A8}$$

where $\mathbf{m}'$ and $\mathbf{P}'$ can be written as

$$\begin{aligned} \mathbf{m}' &= \mathbf{m} + \mathbf{P}\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{P}\mathbf{A}^{\mathrm{T}} + \mathbf{\Sigma})^{-1}(\mathbf{z} - \mathbf{A}\mathbf{m} - \mathbf{b}), \\ \mathbf{P}' &= \mathbf{P} - \mathbf{P}\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{P}\mathbf{A}^{\mathrm{T}} + \mathbf{\Sigma})^{-1}\mathbf{A}\mathbf{P}. \end{aligned} \tag{A9}$$

Proof that Equation (A5) results in a Gaussian distribution for Gaussian $p(\mathbf{x})$ and $p(\mathbf{z}\,|\,\mathbf{x})$ can be found by recognizing the integral as a convolution and by using the convolution theorem with the well-known Fourier transform of the Gaussian distribution. For Equation (A6), it suffices to note that the Gaussian distribution is fully determined by the argument of the exponent. Completing the square in the argument of the exponential for the product of the two distributions and using the matrix inversion lemma gives the above result [9].

A.2 Quaternions and rotations

The quaternions are the numbers of a 4-dimensional hypercomplex algebra first discovered by William Rowan Hamilton in 1843 [22]. Since their discovery, they have found use in many areas of mathematics and engineering, particularly as a representation for 3-dimensional rotations. However, despite their popularity, the algebraic notation of quaternions in literature has not been standardized. This section presents a brief synopsis of the quaternions with the conventions used in this thesis.

A.2.1 Definitions and properties

A quaternion $q \in \mathbb{H}$ is defined as the 4-dimensional hypercomplex number

$$q \triangleq a + bi + cj + dk, \tag{A10}$$

where $a, b, c, d \in \mathbb{R}$, and the imaginary units i, j and k obey the identities

$$i^2 = j^2 = k^2 = ijk = -1. \tag{A11}$$

A quaternion can be written in vector form as

$$\mathbf{q} = \begin{bmatrix} a & b & c & d \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} q_{\mathrm{r}} & \mathbf{q}_{\mathrm{v}} \end{bmatrix}^{\mathrm{T}}, \tag{A12}$$

where $q_{\mathrm{r}} \in \mathbb{R}$ is called the *scalar part* and $\mathbf{q}_{\mathrm{v}} \in \mathbb{R}^3$ the *vector part*. A *pure quaternion* has $q_{\mathrm{r}} = 0$, analogous to purely imaginary complex numbers. The *identity quaternion* $\mathbf{I}_{\mathrm{q}}$ is a quaternion with $q_{\mathrm{r}} = 1$ and $\mathbf{q}_{\mathrm{v}} = \mathbf{0}$. Quaternion multiplication is associative and distributive. Using the definitions (A10) and (A11) and writing in vector form, we define the *Hamilton product* of two quaternions as

$$\mathbf{p} \odot \mathbf{q} \triangleq \begin{bmatrix} p_{\mathrm{r}} q_{\mathrm{r}} - \mathbf{p}_{\mathrm{v}}^{\mathrm{T}} \mathbf{q}_{\mathrm{v}} \\ p_{\mathrm{r}} \mathbf{q}_{\mathrm{v}} + q_{\mathrm{r}} \mathbf{p}_{\mathrm{v}} + \mathbf{p}_{\mathrm{v}} \times \mathbf{q}_{\mathrm{v}} \end{bmatrix}. \tag{A13}$$

Quaternion multiplication is not commutative in general, as is evident from the cross product in Equation (A13). Therefore, it is sometimes useful to use the opposite product, given by

$$\mathbf{p} \otimes \mathbf{q} \triangleq \begin{bmatrix} p_{\mathrm{r}} q_{\mathrm{r}} - \mathbf{p}_{\mathrm{v}}^{\mathrm{T}} \mathbf{q}_{\mathrm{v}} \\ p_{\mathrm{r}} \mathbf{q}_{\mathrm{v}} + q_{\mathrm{r}} \mathbf{p}_{\mathrm{v}} - \mathbf{p}_{\mathrm{v}} \times \mathbf{q}_{\mathrm{v}} \end{bmatrix} = \mathbf{q} \odot \mathbf{p}. \tag{A14}$$

It is easy to see that the identity quaternion is the identity element for both quaternion products. The inverse of a quaternion closely resembles that of a complex number and is given by

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|^2} = \frac{1}{\|\mathbf{q}\|^2} \begin{bmatrix} q_{\mathrm{r}} \\ -\mathbf{q}_{\mathrm{v}} \end{bmatrix}, \tag{A15}$$

where $\mathbf{q}^*$ is the *conjugate* of $\mathbf{q}$, and $\|\cdot\|$ is the standard Euclidean norm in $\mathbb{R}^4$. Note that for unit length quaternions, the conjugate is the inverse. The conjugate and norm of a product have the properties that

$$(\mathbf{p} \odot \mathbf{q})^{-1} = \frac{(\mathbf{p} \odot \mathbf{q})^*}{\|\mathbf{p} \odot \mathbf{q}\|} = \frac{\mathbf{q}^* \odot \mathbf{p}^*}{\|\mathbf{q}\| \|\mathbf{p}\|} = \mathbf{q}^{-1} \odot \mathbf{p}^{-1}. \tag{A16}$$

As the quaternion multiplication is linear in either argument, it can be written in matrix form as $\mathbf{p} \odot \mathbf{q} = [\mathbf{p}\, \odot]\mathbf{q} = [\mathbf{q}\, \otimes]\mathbf{p} = \mathbf{q} \otimes \mathbf{p}$, where

$$\begin{aligned} [\mathbf{p}\, \odot] &\triangleq \begin{bmatrix} p_{\mathrm{r}} & -\mathbf{p}_{\mathrm{v}}^{\mathrm{T}} \\ \mathbf{p}_{\mathrm{v}} & p_{\mathrm{r}} \mathbf{I}_3 + [\mathbf{p}_{\mathrm{v}} \times] \end{bmatrix}, \\ [\mathbf{q}\, \otimes] &\triangleq \begin{bmatrix} q_{\mathrm{r}} & -\mathbf{q}_{\mathrm{v}}^{\mathrm{T}} \\ \mathbf{q}_{\mathrm{v}} & q_{\mathrm{r}} \mathbf{I}_3 - [\mathbf{q}_{\mathrm{v}} \times] \end{bmatrix}, \end{aligned} \tag{A17}$$

and where $[\mathbf{p}_{\mathrm{v}} \times]$ denotes the cross product matrix of $\mathbf{p}_{\mathrm{v}}$. Vectors $\mathbf{v} \in \mathbb{R}^3$ are sometimes identified with pure quaternions in order to simplify notation, writing $\mathbf{v} \equiv \mathbf{q}$ where $q_{\mathrm{r}} = 0$ and $\mathbf{q}_{\mathrm{v}} = \mathbf{v}$. This also gives meaning to notation such as $\mathbf{p} \odot \mathbf{v}$, $[\mathbf{v}\, \otimes]$, and so on.

A.2.2 Quaternions as rotations

Unit quaternions (known as *versors*) can be used to encode rotations of 3-dimensional space. Starting with a rotation vector (i.e. axis-angle vector) $\boldsymbol{\theta}$ whose direction is the rotation axis and magnitude is the angle of rotation in radians counterclockwise with the axis pointing toward the viewer, a quaternion

$\mathbf{q}$ representing the same rotation is given by

$$\mathbf{q} = \mathcal{Q}(\boldsymbol{\theta}) \triangleq \begin{bmatrix} \cos\left(\frac{\|\boldsymbol{\theta}\|}{2}\right) \\ \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|}\sin\left(\frac{\|\boldsymbol{\theta}\|}{2}\right) \end{bmatrix}. \tag{A18}$$

Note that $\|\mathbf{q}\| = 1$ and thus $\mathbf{q}^{-1} = \mathbf{q}^*$. Rotation of a vector $\mathbf{v} \in \mathbb{R}^3$ is then accomplished by

$$\mathbf{v}' = \mathbf{q} \odot \mathbf{v} \odot \mathbf{q}^{-1} = \mathbf{q} \odot \mathbf{v} \odot \mathbf{q}^*, \tag{A19}$$

which is sometimes called the "sandwich product" due to the way that it encloses its operand. Note that $\mathbf{v}$ and $\mathbf{v}'$ are identified with the pure quaternions, and that the sandwich product with a pure quaternion always yields a pure quaternion. Applying a further rotation by a quaternion $\mathbf{p}$ to $\mathbf{v}'$ gives

$$\begin{aligned} \mathbf{v}'' &= \mathbf{p} \odot \mathbf{v}' \odot \mathbf{p}^* \\ &= \mathbf{p} \odot \mathbf{q} \odot \mathbf{v} \odot \mathbf{q}^* \odot \mathbf{p}^* \\ &= (\mathbf{p} \odot \mathbf{q}) \odot \mathbf{v} \odot (\mathbf{p} \odot \mathbf{q})^*, \end{aligned} \tag{A20}$$

which means that quaternion rotations compose by Hamilton product. The rotation action of Equation (A19) is encoded by the rotation matrix

$$[\mathbf{q}\,\odot][\mathbf{q}^*\otimes] = \begin{bmatrix} q_\mathrm{r}^2 + \mathbf{q}_\mathrm{v}^\mathrm{T}\mathbf{q}_\mathrm{v} & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{3\times 1} & \mathbf{q}_\mathrm{v}\mathbf{q}_\mathrm{v}^\mathrm{T} + q_\mathrm{r}^2\mathbf{I}_3 + 2q_\mathrm{r}[\mathbf{q}_\mathrm{v}\,\times] + [\mathbf{q}_\mathrm{v}\,\times]^2 \end{bmatrix}, \tag{A21}$$

which, when restricted to $\mathbb{R}^3$, gives the rotation matrix parameterization

$$\mathcal{R}(\mathbf{q}) \triangleq \mathbf{q}_\mathrm{v}\mathbf{q}_\mathrm{v}^\mathrm{T} + q_\mathrm{r}^2\mathbf{I}_3 + 2q_\mathrm{r}[\mathbf{q}_\mathrm{v}\,\times] + [\mathbf{q}_\mathrm{v}\,\times]^2. \tag{A22}$$

It follows from Equation (A20) that the rotation matrix of the Hamilton product of two quaternions is the product of their rotation matrices,

$$\mathcal{R}(\mathbf{p} \odot \mathbf{q}) = \mathcal{R}(\mathbf{p})\mathcal{R}(\mathbf{q}). \tag{A23}$$

A.2.3 Quaternion kinematics

The continuous-time kinematics of a quaternion rotating according to an angular rotation rate vector $\boldsymbol{\omega} = \dot{\boldsymbol{\theta}}$ is given by

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \odot \boldsymbol{\omega} = \frac{1}{2}[\boldsymbol{\omega}\,\otimes]\mathbf{q}. \tag{A24}$$

If $\boldsymbol{\omega}$ is constant, then the solution to this equation is

$$\mathbf{q}(t) = \exp\left(\frac{t}{2}[\boldsymbol{\omega}\,\otimes]\right)\mathbf{q}(0) = \exp\left(\frac{1}{2}[\boldsymbol{\theta}\,\otimes]\right)\mathbf{q}(0), \tag{A25}$$

where $\theta = \omega t$. On the other hand, the above is simply a rotation of $\mathbf{q}(0)$ by $\|\boldsymbol{\theta}\|$ radians around the axis of ω, which can be written as

$$\mathbf{q}(t) = \mathbf{q}(0) \odot \mathcal{Q}(\theta) = [\mathcal{Q}(\theta) \otimes]\mathbf{q}(0). \tag{A26}$$

Therefore, we have for a quaternion $\mathbf{q}$ and rotation vector θ the identity

$$\mathbf{q} \odot \mathcal{Q}(\theta) = \exp\left(\frac{1}{2}[\theta \otimes]\right)\mathbf{q}. \tag{A27}$$