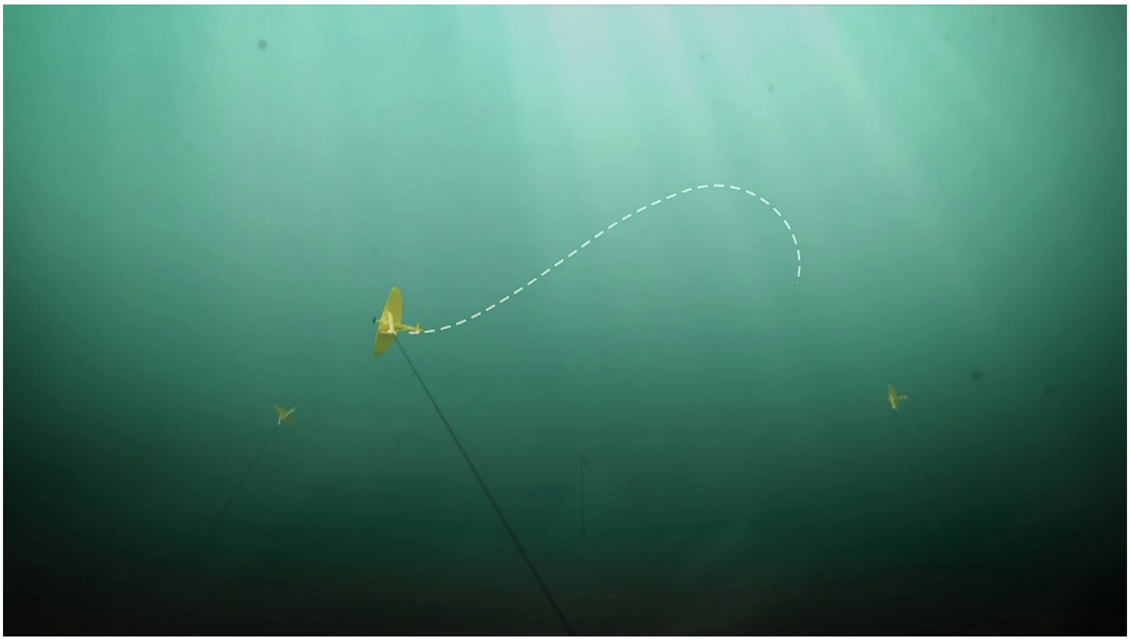
CHALMERS
UNIVERSITY OF TECHNOLOGY



# Position, Velocity and Orientation Estimation of Minesto's Crossflow Underwater Kite

LINN LYSTER

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020:49

# Position, Velocity and Orientation Estimation of Minesto's Crossflow Underwater Kite

LINN LYSTER

Department of Mechanics and Maritime Sciences
Chalmers University of Technology
Gothenburg, Sweden 2020

Position, Velocity and Orientation Estimation of Minesto's Crossflow Underwater Kite
LINN LYSTER



Supervisor: Björn Bergqvist, Minesto AB
Examiner: Rickard Bensow, Department of Mechanics and Maritime Sciences

Master's Thesis 2020:49
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Minesto's crossflow underwater kite Deep Green. Picture from Minesto [1].

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2020

Position, Velocity and Orientation Estimation of Minesto's Crossflow Underwater Kite
LINN LYSTER
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

# Abstract

Crossflow underwater kites have shown promising potential to generate green energy, with Minesto's Deep Green on the front line. The power output is optimized when the kite is following a figure-eight trajectory, where the motion is controlled by a rudder on the kite. This work focuses on providing the control system with improved input about position, velocity and orientation of the kite, with the use of inertial sensors and knowledge of depth, in order to steer the rudder optimally. The sensor signals were processed and filtered in order to handle problems of sensor and process noise.

An algorithm was designed that combined the different sensors to predict the pose and motion of the kite. This was done by first approximate the noise of the sensors, which were used as input, into an extended Kalman filter for orientation estimation, together with inertial measurements from a gyroscope and an accelerometer. After an initial guess on position, based mainly on depth of kite, a steady-state Kalman filter was applied in order to improve the position estimate and also obtain velocity.

The result show that the sensor fusion performed has potential in predicting the movement of the kite. However, the limited access to data prevents us from drawing too big conclusions. Even if there are some challenges regarding bias drift and robustness of the algorithm, it can be shown that the proposed algorithm produces realistic output when it comes to physical constraints due to the tether length but also in terms of periodicity of the orientation.

Keywords: pose estimation, cross-flow underwater kite, sensor fusion, Kalman Filter, extended Kalman Filter.

# Acknowledgements

I would like to thank all people at Minesto for making this project possible. A special thanks to Joakim Boodh for having great patience with all of my questions and for always taking time to discuss challenges and share ideas. I also want to thank my supervisor Björn Bergqvist for keeping me on the right track and provide appreciated input on my work. In addition, I would like to thank my examinator Rickard Bensow, for great support in the academic process.

Linn Lyster, Gothenburg, June 2020

# Contents

# 1 Introduction

The oceans contains a huge amount of energy. One example is the energy from tides, where currents are created. This energy resource is advantageous compared to other renewable energy resources when it comes to reliability, since tides are predictable. The generation of energy from ocean currents can be obtained by crossflow underwater kites. Those have showed great potential, with Minesto's Deep Green on the very cutting edge. The cross-current motion gives the kite a velocity significantly higher than the flow, increasing the power output compared to a stationary tidal plant.

First-generation tidal technologies have suffered from constraints in order to be commercially viable, due to the dependence of strong tidal streams and specific installation depth [2]. Crossflow underwater kites do not suffer from those constraints, since the crossflow motion makes it possible to operate at slow tidal streams cost-efficiently.

The kite is joined to a tether which usually is attached to the seabed. In order to maximise power output and avoid twisting of the tether, movement along a so called *figure-eight motion* is preferred [3]. This requires knowledge of the whereabouts of the kite, in order to be able to steer the kite along the desired path and in that way optimise the power output.

The need of the kite's position and orientation proposes us to use different sensors. In this project inertial sensors, such as accelerometer and gyroscope will be used together with a pressure measurement device, giving the depth. The sensors are combined in so called sensor fusion. Examples of sensor fusion can be found in many different areas, such as vehicle tracking systems with the use of camera and radar. The camera is good at estimating the angle at where an object is located but is bad at estimating distance to the object. However, the opposite holds for the radar - it is good at measuring distance but not angle. Combining these sensors is desired, giving a better estimate than from one unique sensor.

Another example is the case of estimating orientation with inertial sensors, where the gyroscope gives a good measure on the kite's angular velocity but is exposed to drift when integrated to obtain angles. In order to correct the drift, the measurement from an accelerometer is correcting the orientation by comparing the acceleration with the gravitational acceleration as a ground truth. The sensor signals are though subject to noise, whose impact is reduced by applying different filters.

## 1.1 Aim and Research Question

The control system steers the kite in a predetermined figure-eight trajectory. In order to steer the rudder optimally and efficiently, an estimation of the kite's position and orientation is needed. Today the kite's control system is based on position and heading of kite, but the estimation of those parameters is made without any compensation for noise or systematic errors. It is therefore a desire to implement more advanced filtering techniques. The position estimate used today is also dependent on a speed measurement device, which is desirable to remove.

The aim for the project is to develop an algorithm that improves the estimate of position and orientation of the kite and provides the kite's velocity. In order to successfully be able to extract those parameters, the data from the kites *inertial measurement unit* (IMU) is used together with a pressure sensor indicating the depth of the kite. This data is then processed and filtered in order to handle problems of sensor and process noise.

With better estimation of position and orientation the power production increases, by controlling the system to better follow the optimal trajectory. This gain in energy generation is naturally a positive contribution in cost-efficiency. Also, in the current system there is a speed measurement device mounted on the kite - which will be superfluous if we can successfully extract this variable from the other sensors. This device elimination is desirable in order to lower the production costs.

## 1.2 Delimitation

The project only focuses on estimating the input parameters to the control system. Any work related to other parts of the control system, for example trajectory planning or actuation control, will be disregarded.

There is some practical framework that limits the available data. Since there is a generator mounted on the kite, the information from the IMUs magnetometer must be discarded due to disturbances in the magnetic field when generator is operating.

## 1.3 Thesis Outline

The thesis starts with a background of ocean energy. The chapter also describes the field of crossflow kites as well as pose estimation. In chapter 3, the theory of filtering is presented, and the filters used in the report are described. Next chapter treats the model of the system, where coordinate frames, measurements and parametrization are described. Here also the system dynamics is presented. In chapter 5 the method for solving the pose estimation problem is discussed in detail. Finally, the results are presented, followed by a discussions and conclusions. Lastly future work is proposed.

# 2
# Background

This section will give a short summary over the potentials in ocean energy. An overview of crossflow kites will also be presented, with emphasis on Minesto's Deep Green and its functionality and structure. Also, the field of pose estimation will be introduced.

## 2.1 Ocean Energy

The ocean contains a tremendous amount of energy. Minesto's Deep Green is developed to efficiently produce energy out of tidal streams and ocean currents. Other ocean energy resources, for example waves and salinity gradients, will be disregarded in this report.

### 2.1.1 Tidal Streams and Ocean Currents

Tidal energy has a great potential for electricity generation in terms of reliability. Tide is the periodic motion of sea waters, which is the result from the gravitational pull from mainly the moon and the sun together with rotation of earth [4]. This creates tidal streams, that contain an enormous amount of energy. Those forces are not dependent on weather, but instead well-known cycles of the moon, sun and earth and can thus be predicted accurately. This also means that the tide's movement and stream speed are known with great precision. In other words, the tides are more reliable compared to other renewable resources, for example wind and sun-based power.

Tidal energy has traditionally struggled with fairly high costs and limited accessibility of suitable locations with adequately high stream speed and tidal ranges. However, over the last two decades there has been increased interest in this field [5], with many technical innovations and improvements. This constant development show that the aggregate accessibility of tidal energy might be greatly under estimated [6]. Although it is not used extensively today, it has great future potential [4], with economical and ecological viability on the horizon.

Ocean currents have a lot in common with tidal streams in terms of reliability, but are created by geographical differences in salinity and temperature along with the Coriolis effect [2]. In addition to the currents giving rise to continuous flow of water in the same direction, they are also influencing the climate widely, especially in Northern Europe.

In summary, the greatest advantages with tidal streams and ocean currents are the predictability and reliability. Tidal streams and ocean currents are also global energy resources, with minimal use of land [2].

### 2.1.2 Available power

The mass flow of a fluid trough a turbine of area $A$ can be derived from the continuity equation of fluid mechanics. This is a function of fluid density $\rho$ and fluid velocity $U$, according to equation (2.1). The power of the flow is given by equation (2.2). [7]

$$\frac{dm}{dt} = \rho A U \tag{2.1}$$

$$P = \frac{1}{2}\frac{dm}{dt}U^2 = \frac{1}{2}\rho A U^3 \tag{2.2}$$

As can be seen the power is a function of fluid speed, area of turbine and density of fluid. However, not all of the available power can be harvested and therefore a turbine power coefficient $c_p$, needs to be included [8]. Thus, the available power can be expressed as

$$P_{turbine} = c_p P = \frac{1}{2}c_p \rho A U^3. \tag{2.3}$$

It can also be noted that since water is a high-density fluid compared to for example air, it requires smaller turbines in order to extract the same amount of power from same velocity conditions.

## 2.2 Crossflow Underwater Kites

A crossflow underwater kite energy system is a solid undersea wing attached by a flexible tether to a support structure. The support structure can be located on the seabed or the ocean surface [9]. It is an alternative to the underwater stationary turbines, where the crossflow underwater kite has the advantage of higher potential power output. This is because the kite is controlled to travel in cross-current motions at a velocity significantly higher than the ocean flow [8]. The cross-current motion is described in section 2.2.1.

Underwater kite systems have not been studied as much as the related problem of generating power from wind by using kites. The concept of extracting power using kite systems was first proposed by Loyd in 1980 [10] and using automatically controlled tethered kites in order to generate power from high-altitude wind is still a growing field [11]. The concept is often referred to as an *Airborne Wind Energy* (AWE) system, where power output is given as a function of wind speed and the lift to drag ratio [9]. Those studies have advanced our knowledge in modelling and understanding kite-tether dynamics and design of critical system components, among other [3]. Especially well-advanced is the autonomous control system to handle kite trajectories for AWE systems, which is of great importance also in the underwater kite systems in order to optimize power output [3]. The great advantage of crossflow underwater kites compared to AWE systems is that they can be made almost weightless, due to the forces of the high-density fluid. This can be one reason why the AWE systems not yet have reached commercial potential.

Landberg [12] first proposed undersea kites similar to AWE systems, but generating power from water currents instead of wind. Since water is 832 times denser than air [13], operating in a different environment gives the possibility to capture more energy. This can be seen in equation (2.3). Landberg is the inventor of Minesto's Deep Green.

### 2.2.1 Figure-eight motion

It has been shown that power output is optimized when the kite moves in cross-fluid motions, i.e. roughly perpendicular to the flow. For underwater kite systems, this means cross-current figure-eight motion [3], described by a lemniscate. This motion also prevents the tether from twisting.

Ideally the kite motion is confined to the surface of a sphere, and thus its velocity is tangential to a sphere's surface [14], whose radius is defined by the tether length. The figure-eight motion can be expressed mathematically as consisting of two small circle sweeps and two great circle sweeps which connect the small circles [14], see figure 2.1.

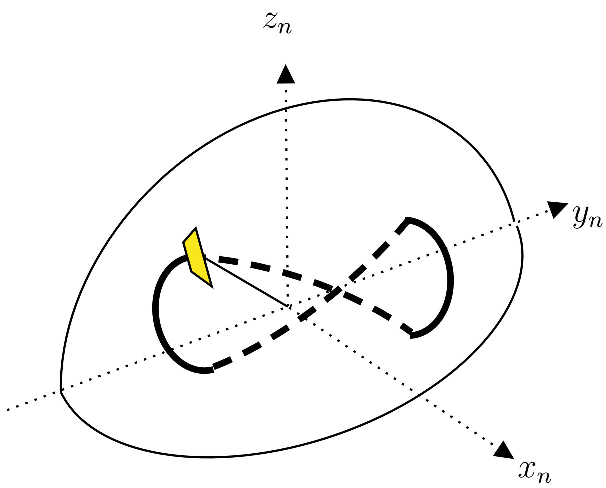



**Figure 2.1:** Figure-eight motion defined by the kite moving on the surface of a sphere in a motion built up by two small curves (solid) and two bigger curves (dashed).

### 2.2.2 Minesto's Deep Green

Deep green is Minesto's unique and patented tidal power plant, that can cost-efficiently produce electricity from slow tidal streams and ocean currents [15].

#### Functionality

Deep Green is a crossflow underwater kite power plant, with a wing that pushes the turbine trough the water in a figure-eight trajectory. The kite then sweeps a large

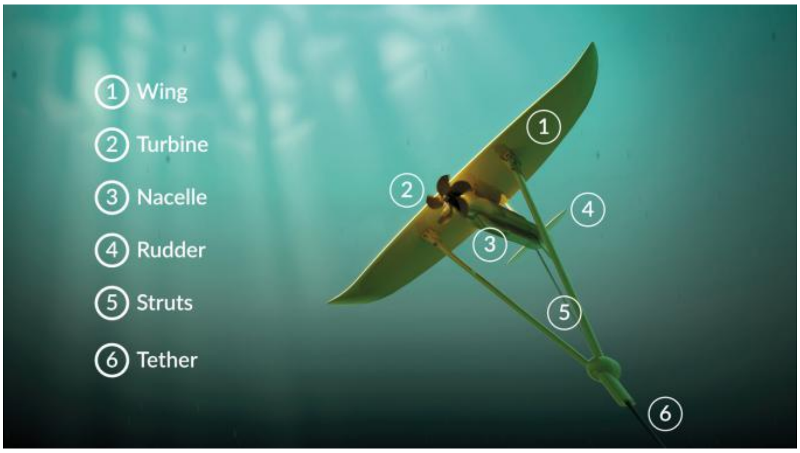



**Figure 2.2:** The different parts of the kite. Image from Minesto [1].

area at a relative speed several times the actual speed of the underwater current.

This property is what differs Deep Green from stationary marine turbines [8]. Where stationary plants need higher currents in order to be cost-efficient, Deep Green can also handle low-velocity streams. The power density is dependent on the cube of the velocity, see equation (2.3). This means that when the relative speed of the turbine increases compared to a stationary one, the electricity produced by the power plant is much greater.

When water flows through the turbine, the generator produces electricity. Electricity is then transferred trough a cable encapsulated in the tether. The cable continues on the seabed and then on to land, where the electricity is passed on to the grid [1].

### Kite Structure

A sketch of a kite can be seen in figure 2.2, where one can see the major parts of the kite. It is built up by a wing (1) and a turbine (2) which is connected to a generator in the nacelle (3). The control system and servo steer the rudder (4). The tether (6) is fixed to the kite by the struts (5) and is connected to a bottom joint at seabed. The tether carries cables for power distribution and communication.

### Kite Control Today

The control system aims to steer the kite in a optimal trajectory along a lemniscate. It is steered by a rudder and the rudder angle is provided by a track following algorithm. The algorithm needs input from the kite position, heading and speed.

Today those inputs are estimated without accounting for noise. First the azimuth and elevation angles are estimated, giving position of kite since constant tether length is assumed. The elevation is computed from the depth given by the pressure measurement device. The azimuth is dependent on kite heading and speed in order to find closest point on the lemniscate. However, this approximation is not very exact and the azimuth estimation is reset every now and then.

The heading is computed from the projection of acceleration vector, assuming only gravitational and centripetal acceleration. The centripetal acceleration due to circular motion, is subtracted from the acceleration measurement, by looking at change in heading and speed. Depending on the gravitation vector, the heading is estimated. For example, when gravitation is only in the kite's positive z-direction, the heading is assumed 0 and when the kite's acceleration is directed in only x-direction, 180 degrees is set.

The kite has multiple control modes and the control system acts differently depending on mode. In this report, we only focus on the run and park mode. Modes as in for example releasing the kite to the surface or steer the kite manually will not be considered.

In the run mode, the kite delivers electricity to the grid by letting the turbine spin. The control system takes input from the kite's state, such as position or angular velocity and tries to follow the optimal trajectory. In park mode, the kite focuses on stable hovering in water without loosing control.

## 2.3 Pose Estimation and Inertial Sensors

Applications of inertial sensors and inertial navigation has been rapidly growing in recent years. This can be explained by the development of micro electromechanical sensors and better computer performance [16]. The term inertial sensor is used to describe the combination of an accelerometer and a gyroscope, where devices containing these sensors often are referred to as *inertial measurement units* (IMUs) [17].

Pose estimation is the term describing the combined estimation of both position and orientation. In order to successfully estimate pose, inertial sensors are used. For example, those can be found in mobile devices, unmanned and autonomous aerial and underwater vehicles. These sensors are also common in order to track human body motions, often called motion capture [18]. Inertial sensor measurements can be sampled at high frequencies and can be integrated in order to obtain position, velocity and orientation. However, these measurements are typically suffering from bias and noise. Therefore, the need of fusing data from the different sensors is crucial in order to receive an optimal pose estimate, so called *sensor fusion*.

Lots of estimation algorithms fusing inertial and magnetic sensors have been implemented, where the most used approaches are the complementary filter and the extended Kalman filter (EKF) [18]. The Kalman filter (KF) and EKF are the most recognized and adopted approaches, which is going to be explained in depth in section 3.1.1. First, we start with an introduction to Bayesian statistics, which the filters applied in this project is based upon.

# 3

# Filtering

In order to be able to extract the kite's angular velocity and linear acceleration from the noisy measurements, filtering has to be performed. This section gives a brief review of the filtering methods used.

## 3.1 Bayesian Statistics and Filtering

Filtering can be defined as recursively estimate parameters of interest $\boldsymbol{x}_k$, also referred to as the state vector, based on measurements $\boldsymbol{y}_k$ at time $k$. The objective is to compute the posterior distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$, where $\boldsymbol{y}_{1:k}$ contains all measurements up to time $k$.

Most dynamical systems can be described on a state space model form, from where we often can find efficient algorithms to solve the filtering problems. A discrete-time state space model can be described by a motion model (3.1) and a sensor or measurement model (3.2),

$$\boldsymbol{x}_k = f_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{q}_{k-1}), \tag{3.1}$$

$$\boldsymbol{y}_k = h_k(\boldsymbol{x}_k, \boldsymbol{r}_k), \tag{3.2}$$

where $\boldsymbol{q}$ and $\boldsymbol{r}$ are motion and measurement noise, which are considered as additive gaussian noise, and $\boldsymbol{u}$ is the control vector. We also assume that the initial prior $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$ is known. The motion model (3.1) gives information about $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ and describes the dynamic of the system. The measurement model (3.2) provides information about distribution $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ and relate the observations to the state vector.

Moreover, it is assumed that the models possess the *Markov property*. That is for the state at time $k$ being conditionally independent on all the previous measurements and states, where all information up to time $k$ is included in the state [17].

From Bayesian statistics, the posterior density of the collection of all state vectors between start and time $k$ can be obtained by using Bayes' rule and the Markov property,

$$p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{1:k-1}) = \frac{p(\boldsymbol{y}_{1:k}|\boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k})}{p(\boldsymbol{y}_{0:k})} \propto p(\boldsymbol{x}_0)\prod_{i=1}^{k} p(\boldsymbol{y}_i|\boldsymbol{x}_i)p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}). \tag{3.3}$$

Further, in order to obtain information about current state $\boldsymbol{x}_k$, marginalization with respect to all previous states $\boldsymbol{x}_{0:k-1}$ is performed,

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \int p(\boldsymbol{x}_{0:k}|\boldsymbol{y}_{k-1})d\boldsymbol{x}_{0:k-1}. \tag{3.4}$$

However, with increasing $k$, (3.3) and (3.4) are demanding to solve. Therefore recursive filtering is proposed, for example Kalman filters, where $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ is computed from previous $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$. This is done by first predicting the next state, which is used as a prior in order to update the state with the use of the current measurement.

In the prediction step the aim is to predict $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$ from $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$. In order to get the prior distribution, the law of total probability is used to integrate over previous state $x_{k-1}$. This gives the Chapman-Kolmogorov equation,

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) &= \int p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1} \\ &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1}. \end{aligned} \tag{3.5}$$

Solving this integral gives the prediction step.

In the measurement update the state $\boldsymbol{x}_k$ is updated with the new measurement $\boldsymbol{y}_k$. More formally, computation of $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ from $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$ is performed,

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}), \tag{3.6}$$

where the normalization factor is omitted. As usual in Bayesian statistics, the posterior distribution is proportional to the likelihood and the prior distribution. The equations (3.5) and (3.6) are referred to as the filtering equations and are applicable to all filtering problems. [19]

### 3.1.1 Kalman Filtering

The filtering equations (3.5) and (3.6) are generally tedious to solve. One reason is that the posterior distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ can rarely be expressed analytically. However, linear and Gaussian models are one exception, with state space model expressed according to

$$\boldsymbol{x}_k = \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{q}_{k-1}, \tag{3.7}$$

$$\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{r}_k, \tag{3.8}$$

where $\boldsymbol{q}_{k-1}, \boldsymbol{r}_k$ and $\boldsymbol{x}_0$ are all Gaussian with covariances $\boldsymbol{Q}_{k-1}, \boldsymbol{R}_k$ and $\boldsymbol{P}_{0|0}$ respectively. $\boldsymbol{A}_{k-1}$ is the transition matrix, $\boldsymbol{B}_k$ is the control-input model matrix applied to the control vector $\boldsymbol{u}_k$ and $\boldsymbol{H}_k$ is the measurement model matrix. For those models the distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ is also Gaussian. More generally this also holds for

$$p(\boldsymbol{x}_m|\boldsymbol{y}_{1:n}) = \mathcal{N}(\boldsymbol{x}_m; \hat{\boldsymbol{x}}_{m|n}, \boldsymbol{P}_{m|n}) \quad \forall m, n, \tag{3.9}$$

where $\hat{\boldsymbol{x}}_{m|n}$ and $\boldsymbol{P}_{m|n}$ are the state estimate or mean and state covariances respectively. The notation $\hat{\boldsymbol{x}}_{n|m}$ represents the estimate of $\boldsymbol{x}$ at time $n$ given observations up to $m \leq n$.

The Kalman filter gives an analytic solution to the filtering equations (3.5), (3.6) for linear and Gaussian models. The goal of the Kalman filter is to compute the mean and covariance for $m = k$ and $n = k-1, k$ recursively for $k = 1, 2, 3...$ The Kalman filter equations are derived using the properties of linear combination of independent Gaussian variables together with conditional distribution of Gaussian variables.

The prediction step in the Kalman filter is straightforward, with mean and covariances computed according to

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}_k\boldsymbol{u}_k \tag{3.10}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{A}_{k-1}^T + \boldsymbol{Q}_{k-1}. \tag{3.11}$$

The update step of the state $\hat{\boldsymbol{x}}_{k|k}$ is given by the prior state together with a correction term depending on the new measurement. The covariance is updated by the prior covariance along with a term depending on how informative the new measurement is. This can be summarized in the following equations

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\tilde{\boldsymbol{y}}_k \tag{3.12}$$

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{S}_k\boldsymbol{K}_k^T, \tag{3.13}$$

where $\boldsymbol{K}_k, \tilde{\boldsymbol{y}}_k$, and $\boldsymbol{S}_k$ are the so called Kalman gain, innovation and innovation covariance respectively,

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T\boldsymbol{S}_k^{-1} \tag{3.14}$$

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{y}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_{k|k-1} \tag{3.15}$$

$$\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k. \tag{3.16}$$

### Steady-state Kalman Gain

Instead of recursively compute the Kalman gain $\boldsymbol{K}$ in the update step for Kalman filtering, there exists a simplification when the change in noise covariance matrix $\boldsymbol{Q}$ is expected to be approximately constant. We can then replace $\boldsymbol{K}_k$ and $\boldsymbol{P}_{k|k-1}$ with their steady-state values,

$$\boldsymbol{K} := \lim_{k\to\infty} \boldsymbol{K}_k \qquad\qquad \boldsymbol{P}_\infty := \lim_{k\to\infty} \boldsymbol{P} \tag{3.17}$$

The steady-state Kalman gain $\boldsymbol{K}$ can be computed only once, by solving the following equation,

$$\boldsymbol{K} = \boldsymbol{A}\boldsymbol{P}_\infty\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_\infty\boldsymbol{H}^T + \boldsymbol{R})^{-1}, \tag{3.18}$$

where $\boldsymbol{P}_\infty$ satisfies the *Algebraic Riccati Equation* (ARE),

$$\boldsymbol{P}_\infty = \boldsymbol{A}\boldsymbol{P}_\infty\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{P}_\infty\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}_\infty\boldsymbol{H}^T + \boldsymbol{R})^{-1}\boldsymbol{H}\boldsymbol{P}_\infty\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{Q}\boldsymbol{B}^T, \tag{3.19}$$

obtained by setting $\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_{k|k}$. [20][21] This naturally reduces the computational cost during operation, and using steady-state KF instead of the classical KF is faster. The reduced estimation time can be of importance in many real-time applications.

### Extended Kalman Filter

Sometimes dynamical systems cannot be modeled by linear state space models (3.7) and (3.8) and generally the motion or measurement models are non-linear functions where no analytical solution for the filtering equations exists. Therefore, it is common to approximate the non-linear models, so an analytical solution can be found. The *extended Kalman filter* (EKF) uses local linearization about the estimate of the mean and covariance, in order to overcome the problem.

Instead of conditioning the motion and measurement model to be linear functions, the state space model can be written as

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{q}_{k-1}, \tag{3.20}$$

$$\boldsymbol{y}_k = h_k(\boldsymbol{x}_k) + \boldsymbol{r}_k, \tag{3.21}$$

where $\boldsymbol{q}_{k-1}$ and $\boldsymbol{r}_k$ are the motion and measurement noise assumed Gaussian with covariance $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ respectively. The generally non-linear function $f$ can be used to compute the the predicted state estimate $\hat{\boldsymbol{x}}_{k|k-1}$, but the same does not hold for the predicted covariance estimate. Instead, the Jacobian of $f$ with respect to $\boldsymbol{x}$ is computed and evaluated at the last predicted state estimate. The same holds for the function $h$. The Jacobians of $f$ and $h$ are instead plugged into the Kalman filter equations, which linearizes the non-linear functions around the current estimate. The Jacobians can be expressed as

$$\boldsymbol{F}_k = \frac{\partial f(\boldsymbol{x}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{x}_k}\bigg|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_{k|k}}, \tag{3.22}$$

$$\boldsymbol{H}_k = \frac{\partial h(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k}\bigg|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_{k|k-1}}. \tag{3.23}$$

The prediction step can then be formulated as

$$\hat{\boldsymbol{x}}_{k|k-1} = f_{k-1}(\hat{\boldsymbol{x}}_{k-1|k-1}), \tag{3.24}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}_{k-1}^T + \boldsymbol{Q}_{k-1}, \tag{3.25}$$

and the update step similar to (3.12) and (3.13), but with $\boldsymbol{H}_k$ according to (3.23).

## 3.2 Exponential Moving Average

In order to smooth out the short-term fluctuations in signals, a moving average is commonly used. This can be viewed as applying a low-pass filter. Using an *exponential moving average* (EMA), also known as exponential weighted average, decreases the weight on previous measurements exponentially. The EMA for measurements Y can be described by

$$\hat{Y}_t = \alpha \hat{Y}_{t-1} + (1 - \alpha) Y_t, \tag{3.26}$$

where $Y_t$ is the measured signal at time $t$ and $\hat{Y}_t$ is the EMA. The coefficient $\alpha$ is the weighting decrease factor, representing how much weight to be put on earlier measurements. The weighting coefficient $\alpha$ ranges between $[0, 1]$, where high values of $\alpha$ indicate more weight on the past.

## 3.3 Other Filters

The EKF is common when estimating orientation. But the implementation of EKF introduces a linearization error in the KF and increases the computational complexity [22], which proposes using less complex filters if same end result could be obtained. Therefore, other less complex filters were also considered in the orientation estimation. The ones considered are going to be briefly explained below.

### 3.3.1 Complementary Filter

The accelerometer gives a good estimate of orientation in static conditions whereas the gyroscope is good in dynamic conditions. This is used by the complementary filter (CF). It needs input from two different sources - one from high-frequency noise and the other from low-frequency noise [23]. In the case of orientation estimation the filter combines the slow moving signal from the accelerometer and the fast moving signal from the gyroscope.

The simplest implementation found of this is the following equation

$$\begin{bmatrix}\psi_k\\ \theta_k\\ \phi_k\end{bmatrix} = \big(1-\alpha\big)\left(\begin{bmatrix}\psi_{k-1}\\ \theta_{k-1}\\ \phi_{k-1}\end{bmatrix} + T\begin{bmatrix}\omega_{x,k}\\ \omega_{y,k}\\ \omega_{z,k-1}\end{bmatrix}\right) + \alpha\begin{bmatrix}a_{x,k}\\ a_{y,k}\\ a_{z,k}\end{bmatrix}, \tag{3.27}$$

where the parameter $\alpha$ decides how much impact the different sensors should have on the estimation. If $\alpha$ is small, more weight is given on the gyro measurement. The CF implemented in the comparison is a bit more complex. Details can be found in [23], but builds upon the same idea of weighting the sensors differently. Note the need of choosing the tuning parameter $\alpha$.

### 3.3.2 Mahony Filter

Mahony filter[24] is a so called explicit complementary filter. It uses a proportional-integral controller to estimate the gyro bias [22]. This is done by computing the error between the estimated orientation and the orientation obtained by looking at the accelerometer. The error then corrects the gyroscope bias. The measured angular velocity is updated according to

$$\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + K_p\boldsymbol{e}(t) + K_i\int \boldsymbol{e}(t')dt', \tag{3.28}$$

where $K_p$ and $K_i$ are the coefficients of the proportional and integral term respectively, which have to be decided. This is then used in order to obtain the change in orientation dynamics,

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q}\otimes\boldsymbol{\omega},$$

explained more in section 4.1.3.

### 3.3.3 Madgwick Filter

The Madgwick filter [25] is built upon a gradient descent based algorithm [26], which ensures good attitude estimation at low computational cost [27]. It uses the orientation error obtained by a gradient descent algorithm to provide a gyroscope drift compensation.

As in the CF, first the orientation is estimated from the gyro by integration. Then a corrective step based on the gradient decent algorithm is obtained by minimizing the orientation deviation from the assumed gravitational acceleration given by the accelerometer. This correction is then subtracted from the estimated orientation, scaled with a factor $\beta$. More details can be found in for example [26] and [25].

In [26] it is proposed to use particle swarm optimization to determine the control parameters in the Madgwick and Mahony filters.

# 4

# Model

The section defines important notions and concepts, such as the coordinate frames, the measurement outputs and how the dynamics can be described.

## 4.1 System Overview

### 4.1.1 Coordinate Frames

We start by defining the different coordinate frames. First, the body frame $b$ is moving with the kite and origins at the center of the accelerometer. The navigation frame $n$ is the stationary frame with its origin at the tether's attachment to the ground. The aim is to estimate the position and orientation of frame $b$ with respect to the navigation frame $n$. We also have the inertial frame $i$, originating at the center of earth. It is with respect to frame $i$ that linear acceleration from the IMU is measured. A sketch of the coordinate system can be seen in figure 4.2.

In figure 4.1, the different orientation angles pitch $\theta$, yaw $\phi$ and roll $\psi$ are defined. The $x_b$-axis is aligned with the kite's forward direction, the $y_b$-axis points to the right wing and the $z_b$-axis refers to the upward direction of the kite, where the subscript $b$ indicates the body frame. For example, the position and velocity in the navigation frame are denoted $\boldsymbol{p^n}$ and $\boldsymbol{v}^n$.

Also, the kite's position can be described by spherical coordinates, with tether length $r$, azimuth $\phi_{az}$ and elevation angle $\theta_{el}$ as in figure 4.3. The angle $\theta_{el} \in [0, \frac{\pi}{2}]$ is the angle between the $x_n y_n$-plane and vector $\boldsymbol{r}$ and $\phi_{az}$ is the angle between the $x_n$ axis and the projection of $\boldsymbol{r}$ onto the $x_n y_n$-plane.

### 4.1.2 Measurements

The angular velocity from the gyro and accelerometer measurement are extracted from the IMU. A pressure measuring device mounted on the kite gives the depth of the kite and furthermore information about the z-position in navigation frame $p_z^n$.

#### Angular Velocity

The angular velocity is measured by a gyroscope, more specifically angular velocity of the body frame $b$ with respect to the inertial frame $i$. Expressed in body frame $b$ the notion is then $\boldsymbol{\omega}_{ib}^b$ and can be expressed by

$$\boldsymbol{\omega}_{ib}^b = \mathbb{R}^{bn} \boldsymbol{\omega}_{in}^n + \boldsymbol{\omega}_{nb}^b, \tag{4.1}$$

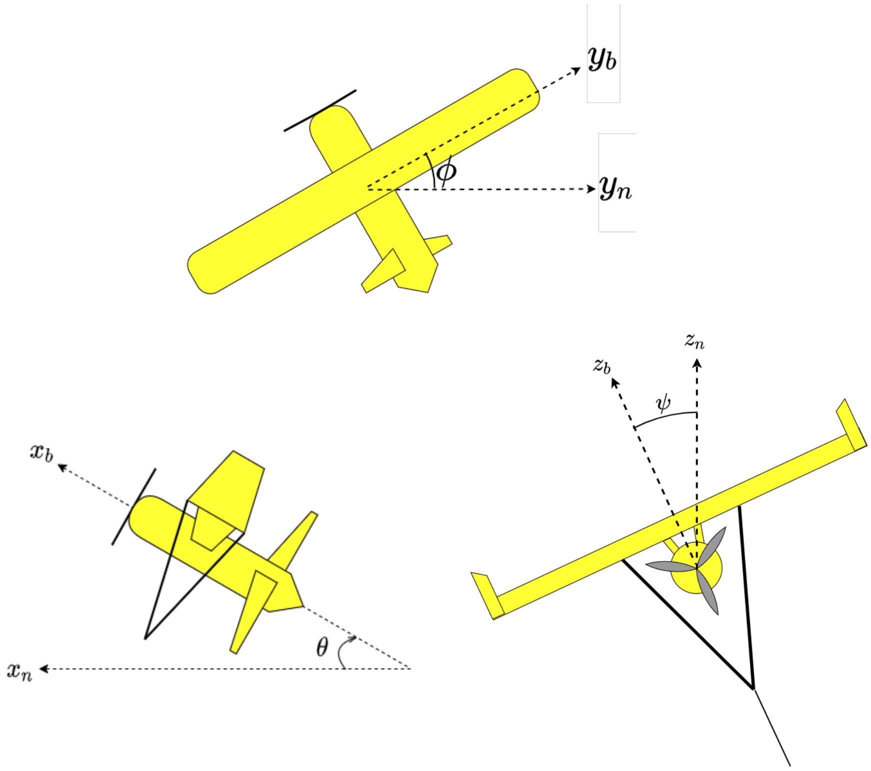



**Figure 4.1:** The rotation around $x_b$-, $y_b$- and $z_b$ are defined as the roll $\psi$, pitch $\theta$ and yaw $\phi$ respectively.

where $\mathbb{R}^{bn}$ is the rotation matrix from frame $n$ to $b$ [17]. The so called *earth rate* $|\boldsymbol{\omega}_{in}| = 7.29 \cdot 10^{-5}$ rad/s, depends on how the earth rotates around its own axis. It is assumed negligible since it is rather small. This assumption leads to that $\boldsymbol{\omega}_{ib}^b \approx \boldsymbol{\omega}_{nb}^b$.

The gyroscope measurement $\hat{\boldsymbol{\omega}}$ is corrupted by slowly time-varying bias $\delta_\omega(t)$ and measurement noise $\eta_\omega(t)$, assumed Gaussian. Thus, the gyro measures

$$\hat{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}_{nb}^b(t) + \delta_\omega^b(t) + \eta_\omega^b(t). \tag{4.2}$$

The slowly time-varying bias is commonly modelled as a random walk, where $\eta_{\delta_\omega}^b$ is Gaussian noise,

$$\delta_\omega^b(t+1) = \delta_\omega^b(t) + \eta_{\delta_\omega}^b(t), \tag{4.3}$$

but in many practical scenarios this bias is varying so slow that it can be approximated constant.

### Acceleration

The accelerometer measures the specific force $f$ in the body frame $b$. The acceleration due to the kite's motion with respect to navigation frame, is obtained by subtracting the offset due to gravity. The specific force can be expressed in terms of the linear acceleration and the gravity vector as follows

$$\boldsymbol{f}^b = \mathbb{R}^{bn}(\boldsymbol{a}_{ii}^n - \boldsymbol{g}^n), \tag{4.4}$$

where $\boldsymbol{a}_{ii}$ denotes the linear acceleration of the sensor and $\boldsymbol{g}^n$ is the gravity vector, both expressed in the navigation frame $n$. The subscripts of acceleration $\boldsymbol{a}$ denotes

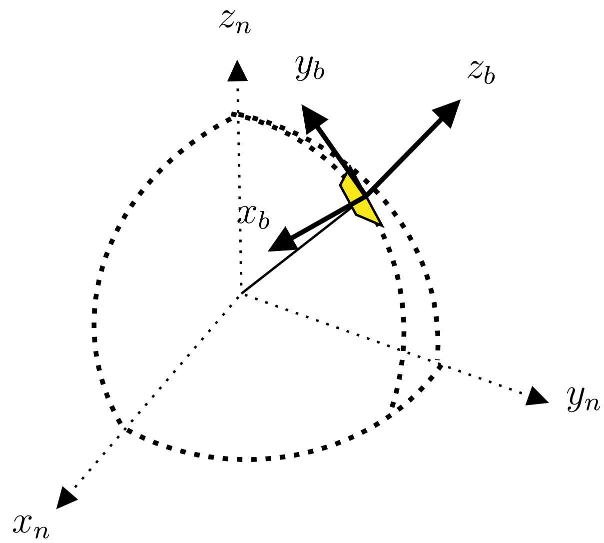



**Figure 4.2:** Definition of the navigation and body frame. Navigation frame origins at tether's attachment to the ground and the body frame origin at the center of the IMU.

which frame the differentiation is performed. Thus, in order to get the position in the navigation frame $n$, denoted $\boldsymbol{p}^n$, we are interested in an expression for $\boldsymbol{a}^n_{nn}$. From [17] a relation between $\boldsymbol{a}_{ii}$ and $\boldsymbol{a}_{nn}$ is derived, see equation (4.5), where effects due to the Earth's rotation relative to the inertial frame $i$ are considered,

$$\boldsymbol{a}^n_{ii} = \boldsymbol{a}^n_{nn} + 2\boldsymbol{\omega}^n_{in} \times \boldsymbol{v}^n_n + \boldsymbol{\omega}^n_{in} \times \boldsymbol{\omega}^n_{in} \times \boldsymbol{p}^n. \tag{4.5}$$

The second term in equation (4.5) is the Coriolis acceleration and the third term is the centrifugal acceleration due to earth's rotation. The latter is dependent on the earth rate and considered negligible. The magnitude of Coriolis acceleration is relatively small compared to the linear acceleration and is also going to be assumed negligible [17].

Similar to the gyroscope, also the accelerometers measurements are affected by slowly time-varying bias $\delta_a(t)$ modelled as a random walk, and Gaussian noise $\eta_a(t)$. In summary, we can approximate the measurement from the accelerometer according to

$$\hat{\boldsymbol{a}}(t) = \mathbb{R}^{bn}(\boldsymbol{a}^n_{nn} - \boldsymbol{g}^n) + \delta_a(t) + \eta_a(t). \tag{4.6}$$

### Pressure

The pressure $p$ increases linearly with depth, according to $\Delta p = \Delta h \rho g$, where $\Delta h$ is the depth below the ocean surface and $\rho$ is the density of the fluid, in this case seawater. Providing the information of where the bottom joint is placed, either on seabed or ocean surface, the difference in $z_n$ can be provided. In other words, the distance from origo at the bottom joint to the kite in z-direction of the navigation frame, $p^n_z$. The details on the conversions from pressure to depth is left out here.

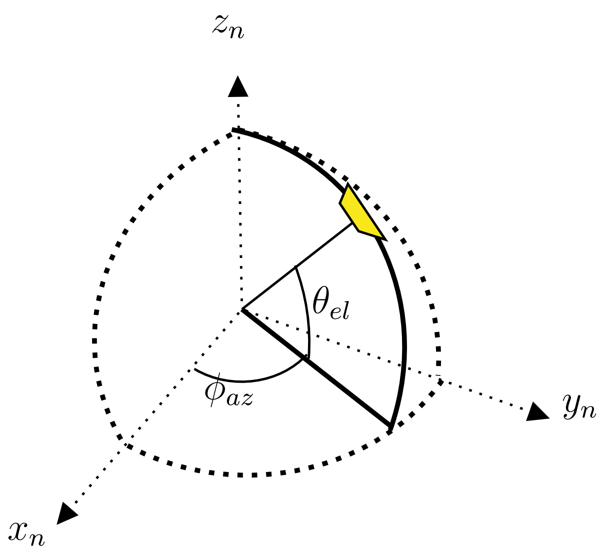



**Figure 4.3:** Definition of the spherical angles $\phi_{az}$ and $\theta_{el}$.

### 4.1.3 Quaternion Parametrisation

Orientation can be described by different parameterizations, for examples rotation matrices or Euler angles. However, in pose estimation algorithms, a quaternion-based representation is widely used [17]. Quaternion $\boldsymbol{q}$ is a vector in $\mathcal{R}^4$ that represent the rotation relation between the body frame $b$ and the navigation frame $n$ [18].

Quaternion $\boldsymbol{q}$ is defined by

$$\boldsymbol{q} = (q_0 \quad q_1 \quad q_2 \quad q_3)^T = \begin{pmatrix} q_0 \\ \boldsymbol{q}_v \end{pmatrix}, \tag{4.7}$$

where $||\boldsymbol{q}||_2 = 1$ is the definition of an unit quaternion. A vector $\boldsymbol{v}$ in $\mathcal{R}^3$ can be written as a pure quaternion by $\boldsymbol{v}_q = (0\, v_x\, v_y\, v_z)$. It can be rotated by a quaternion $\boldsymbol{q}$ using

$$\boldsymbol{v}_q^n = \boldsymbol{q} \otimes \boldsymbol{v}_q^b \otimes \boldsymbol{q}^*. \tag{4.8}$$

This is equivalent to a rotation by a rotation matrix $\boldsymbol{v}^n = \mathbb{R}^{nb}\boldsymbol{v}^b$. Note that the symbol $\otimes$ represent quaternion multiplication and $\boldsymbol{q}^*$ is the quaternion conjugate given by

$$\boldsymbol{q}^* = (q_0 \quad -q_1 \quad -q_2 \quad -q_3) = \begin{pmatrix} q_0 \\ -\boldsymbol{q}_v \end{pmatrix}. \tag{4.9}$$

The time derivative of a rotation of an object with angular velocity $\boldsymbol{\omega}$ is given by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega}. \tag{4.10}$$

It can be noted that the derivative of a quaternion is itself a quaternion. The derivative can be represented in matrix form by (4.11).

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}\begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4.11}$$

defining the matrices above as $\mathbb{S}(\boldsymbol{\omega})$ and $\bar{\mathbb{S}}(\boldsymbol{q})$,

$$\mathbb{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{4.12}$$

$$\bar{\mathbb{S}}(\boldsymbol{q}) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{4.13}$$

The advantages with using quaternions instead of for example Euler representation is that the number of parameters is decreased as well as the avoidance of singlarity configurations. Those are some of the reasons why quaternion representation is used in most of the recent sensor fusion algorithms for orientation estimation [28].

## 4.2 Dynamics

The discrete representations of the dynamical models are presented below.

### 4.2.1 Orientation Dynamics

In order to discretize the motion model in (4.11), we use that $t \in (t_{k-1}, t_k]$ with time step $T$, along with the derivation in appendix B. This gives the discrete expression with first order Taylor expansion,

$$\boldsymbol{q}_k = \exp\Big(\frac{T}{2}\mathbb{S}(\hat{\boldsymbol{\omega}}_{k-1})\Big)\boldsymbol{q}_{k-1} \approx \Big[\mathcal{I} + \frac{T}{2}\mathbb{S}(\hat{\boldsymbol{\omega}}_{k-1})\Big]\boldsymbol{q}_{k-1}, \tag{4.14}$$

where $\mathbb{S}$ is defined in (4.12). It has previously been discussed that the gyroscope measurement $\hat{\boldsymbol{\omega}}_{k-1}$ contains zero-mean Gaussian noise $\eta_{\omega,k-1}$. With separation of noise [22] one can write the motion model as

$$\begin{aligned} \boldsymbol{q}_k &= \Big[\mathcal{I} + \frac{T}{2}\mathbb{S}(\boldsymbol{\omega}_{k-1} + \boldsymbol{\eta}_{k-1})\Big]\boldsymbol{q}_{k-1} \\ &= \Big[\mathcal{I} + \frac{T}{2}\mathbb{S}(\boldsymbol{\omega}_{k-1})\Big]\boldsymbol{q}_{k-1} + \Big[\frac{T}{2}\bar{\mathbb{S}}(\boldsymbol{q}_{k-1})\Big]\boldsymbol{\eta}_{\omega,k-1}, \end{aligned} \tag{4.15}$$

where the last equality is obtained from (4.11) and (4.13). The process noise can then be formulated as

$$\boldsymbol{\nu}_{k-1} = \Big[\frac{T}{2}\bar{\mathbb{S}}(\boldsymbol{q}_{k-1})\Big]\boldsymbol{\eta}_{\omega,k-1}.$$

### 4.2.2 Position and Velocity Dynamics

In order to describe the movement of the kite in the navigation frame, an estimation of the linear acceleration in the navigation frame $\hat{\boldsymbol{a}}^n_{nn}$, or more compact $\hat{\boldsymbol{a}}^n$, is assumed to be known. This can be obtained by rotating the accelerometer measurement vector from the kite body frame to the navigation frame with the estimated orientation, according to equation (4.8).

The estimate of the acceleration can be considered as

$$\hat{\boldsymbol{a}}^n(t) = \boldsymbol{a}^n(t) + \boldsymbol{\eta}_{a,t}, \tag{4.16}$$

where $\boldsymbol{\eta}_{a,t} \in \mathbb{R}^3$ is the estimation error. The estimate $\hat{\boldsymbol{a}}^n$ can be treated as input to a steady-state Kalman filter, where the state space model can be described by a discretized linear system [29]. In one dimension and if acceleration is assumed constant between samples with sampling time $T$, we have that

$$\begin{aligned} p_k &= p_{k-1} + Tv_{k-1} + \frac{T^2}{2} a_{k-1} \\ v_k &= v_{k-1} + Ta_{k-1}. \end{aligned}$$

In 3-D matrix form, the motion model can be described by

$$\begin{bmatrix} \boldsymbol{p}^n_k \\ \boldsymbol{v}^n_k \end{bmatrix} = \begin{bmatrix} \mathcal{I}_3 & T\mathcal{I}_3 \\ 0_3 & \mathcal{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^n_{k-1} \\ \boldsymbol{v}^n_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2}\mathcal{I}_3 \\ T\mathcal{I}_3 \end{bmatrix} \hat{\boldsymbol{a}}^n_{k-1}, \tag{4.17}$$

where $k$ is the discrete time variable, $\mathcal{I}_3$ is the identity matrix and $0_3$ is the null matrix, both in three dimensions.

In [29] it is assumed that an estimate of position is known, obtained by for example an GPS or line angle sensor. For the case of Minesto's Deep Green those sensors are not available and an estimate of position has to be obtain by other techniques, discussed in section 5.4.1. If we assume that we have an estimate of the position $\hat{\boldsymbol{p}}^n_k$ at $k$, the measurement model can be defined as follow,

$$\hat{\boldsymbol{p}}^n_k = [\mathcal{I}_3 \quad 0_3] \begin{bmatrix} \boldsymbol{p}^n_k \\ \boldsymbol{v}^n_k \end{bmatrix} + \boldsymbol{\eta}_{\boldsymbol{p},\boldsymbol{k}}. \tag{4.18}$$

The noise $\boldsymbol{\eta}_{\boldsymbol{a},\boldsymbol{k}}$ and $\boldsymbol{\eta}_{\boldsymbol{p},\boldsymbol{k}}$ are assumed to be independent Gaussian processes with covariance $\boldsymbol{Q}$ and $\boldsymbol{R}$ respectively. Assuming the process noise change slowly, the steady-state Kalman filter approach can be chosen, which makes the algorithm faster. Translating this to the Kalman equations from section 3.1.1, we have that

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{p}^n_k \\ \boldsymbol{v}^n_k \end{bmatrix}, \quad \boldsymbol{A}_k = \begin{bmatrix} \mathcal{I}_3 & T\mathcal{I}_3 \\ 0_3 & \mathcal{I}_3 \end{bmatrix}, \quad \boldsymbol{B}_k = \begin{bmatrix} \frac{T^2}{2}\mathcal{I}_3 \\ T\mathcal{I}_3 \end{bmatrix}, \quad \boldsymbol{H}_k = [\mathcal{I}_3 \quad 0_3]. \tag{4.19}$$

Those equations are then used in the steady state Kalman filter.

# 5 Method

This chapter include an overview of the workflow, before going more into details. Explanation of how to handle estimation of initial conditions, orientation, spherical angle and lastly position and velocity will be given.

## 5.1 Overview

The approach to obtain an estimation of the kite's pose and motion can be seen in figure 5.1. The sensors utilised are the gyroscope, accelerometer and pressure measurement device. The conversion from the pressure measurement output to depth is provided by Minesto, giving information about the variable $p_z^n$. This estimation can be considered rather accurate, since pressure sensors can provide resolution of 2mm [30].

The signals from the gyroscope and the accelerometer are first filtered. This is done by applying an exponential running mean, which can be seen as a lowpass filter. The parameter $\alpha$, responsible for how much weight should be put on previous samples, is set to $\alpha = 0.5$ for both gyroscope and accelerometer signal.

Initial orientation is estimated when kite is in parking mode. In parking mode, only gravitational acceleration is assumed and no significant linear acceleration due to movement of kite. When initial orientation is found and flight mode is entered, the estimate of orientation is given by an EKF with quaternion states. First, the approximated linear dependence on angular velocity is updating the quaternion states before measurement from the accelerometer is correcting the predicted state, depending on direction of gravitational vector.

When the orientation is estimated, we can use the result in order to split up the measurement from the accelerometer into gravitational and linear acceleration. The linear acceleration is used as input to a Kalman filter (KF), whose goal is to estimate the kite's position and velocity. Also, the KF needs a prior guess of the kite's position. Due to that only the depth is given by the pressure sensor, an algorithm for estimating position in all three Cartesian components is developed and used as input to the KF. The KF outputs position and velocity given the linear acceleration and the prior position estimate.

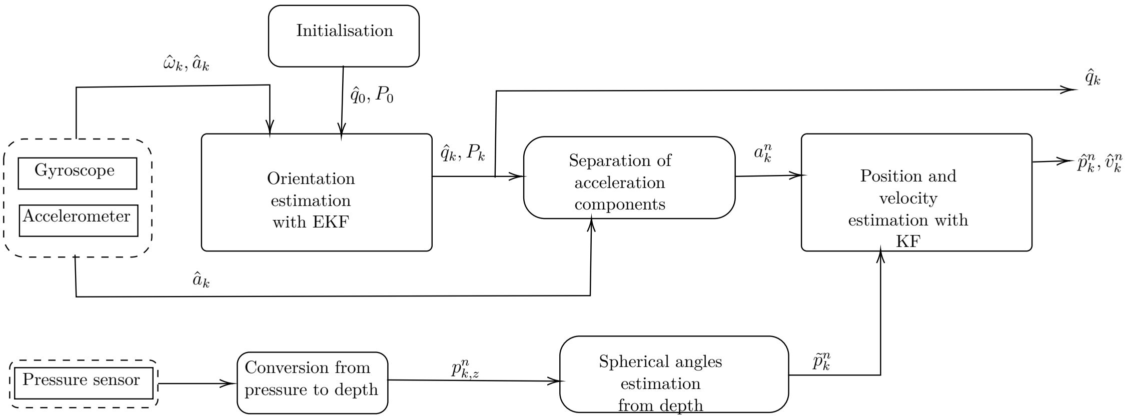



**Figure 5.1:** Block diagram of filter design.

### 5.1.1 Sensor Signals

Sensor signals from the accelerometer and gyroscope are modelled as

$$\hat{\boldsymbol{a}} = \mathbb{R}^{bn}(\boldsymbol{a}^n_{nn} - \boldsymbol{g}^n) + \boldsymbol{\eta}_a \tag{5.1}$$

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}^b_{nb} + \boldsymbol{\eta}_\omega, \tag{5.2}$$

where the noise $\boldsymbol{\eta}_a, \boldsymbol{\eta}_\omega$ are assumed Gaussian with covariance $\Sigma_a, \Sigma_\omega$ respectively. Every sensor is also influenced by a time varying bias. However, since those can be assumed to change slowly [31], they are not taken into account here.

### 5.1.2 Research Methodology

The choice to implement an EKF for the orientation estimation instead of other filters, was founded by comparing the performance with three other common filters - the complementary, Mahony and Madgwick filter. This was done by using the gyroscope and accelerometer measurement from an Android device, collected online by an app available at Google Play, called *Sensor Fusion* and developed by Linköpings University [32]. The data was streamed to a PC where the different filters were implemented in Matlab. Also the orientation given by the estimation implemented in the Android device was streamed and used as a ground truth.

By comparing the output of the filters visually with the ground truth, when moving the Android device around it was clear that the best performance was obtained by the EKF. However, one of the shortcomings of the comparison with the other filters was that they had control parameters that needed to be tuned before applying the algorithms [26]. This was not rigorously investigated, but a span of parameters were tried, all pointing in favor for the EKF. Note also that the computational time was not considered here, which could be an disadvantages with the EKF, see for example in [22]. The choice of the KF filter in the position and velocity estimate was a natural choice considering the straight forward implementation.

## 5.2 Initial Condition and Noise Characteristics

The kite has different operational modes. In parking mode, we treat the kite as approximately stationary, with no linear acceleration ($\boldsymbol{a}_{nn}^n \approx 0$). The accelerometer is under this assumption only showing the specific force from the gravitational acceleration. This means that the pitch angle $\theta$ can be determined by using that the gravitational vector in navigation frame is $\boldsymbol{g}^n = [0,\, 0,\, -1]^T$. When in parking mode, the initial orientation is updated once every second, so that it has not drifted when entering running mode.

In every update except from the first one, averaging over previous acceleration measurements is performed. This is done in order to reduce the impact of fluctuations from noise. The averaging length is dependent on how long the kite has been parked. A maximum history of averaging is set to 2000 samples, or around 8 seconds with a sampling frequency of 250Hz. If for instance the kite has been parked for only 100 samples, averaging is done only over those.

The averaged accelerometer history in parked mode is denoted $\bar{\boldsymbol{a}}_{parked}$. In Euler notation, the start pitch angle can then be found as

$$\theta_0 = -\arctan\Big(\frac{\bar{a}_x^{parked}}{\bar{a}_z^{parked}}\Big). \tag{5.3}$$

Unfortunately, the roll and yaw angles are not related to the sensors at hand. If the magnetometer was not too disturbed by the generator, the direction of the magnetic north could have been extracted. This would give the initial yaw angle, also referred as heading. With two known axes, the last orthogonal axis could be obtained. However, for the time being the roll and yaw are both set to 0. It should therefore be emphasised that the orientation is the relative orientation from this initial orientation. The initial quaternion $\boldsymbol{q}_0$ is then computed from the Euler angels.

The noise characteristics of the sensors are also determined when the kite is in parked mode. First of all, the different axes of the sensors are assumed uncorrelated, giving an diagonal covariance matrices with squared variances,

$$\Sigma_\omega = \mathcal{I}_3\,[\sigma_{\omega,x}^2 \quad \sigma_{\omega,y}^2 \quad \sigma_{\omega,z}^2]^T,$$

and similarly with $\Sigma_a$. The variances are found by looking for the deviation around mean of every sensor axis measurement in park mode.

## 5.3 Orientation Estimation

In order to estimate orientation, an EKF is designed, with 4-D quaternion state vector and 3-D accelerometer measurement vector. The motion model is derived in 4.2.1 and its first order Taylor approximation can be summarized as

$$\boldsymbol{q}_k = \Big[\mathcal{I}_4 + \frac{T}{2}\mathbb{S}(\boldsymbol{\omega}_{k-1})\Big]\boldsymbol{q}_{k-1} + \boldsymbol{\nu}_{k-1}, \tag{5.4}$$

where $\mathcal{I}_4$ is the identity matrix in four dimensions and $\boldsymbol{\nu}_{k-1}$ is the motion noise. The matrix $\mathbb{S}$ is defined in (4.12). In the orientation estimation, linear acceleration is neglected, and the measurement model in the EKF can be expressed as

$$\boldsymbol{y}_{EKF,k} = \boldsymbol{y}_{a,k} = -\mathbb{R}^{bn}\boldsymbol{g}^n + \boldsymbol{\eta}_{a,k}. \tag{5.5}$$

It can be noted that the state space model is nonlinear. However, if using an EKF local linearization can be performed around the state estimate. For that the Jacobians (3.22) and (3.23) need to be computed,

$$\boldsymbol{F}_k = \frac{\partial f(\boldsymbol{q}_k)}{\partial \boldsymbol{q}_k} = \mathcal{I}_4 + \frac{T}{2}\mathbb{S}(\boldsymbol{\omega_k}), \tag{5.6}$$

$$\boldsymbol{H}_k = \frac{\partial h(\boldsymbol{q}_k)}{\partial \boldsymbol{q}_k} = \frac{\partial \mathbb{R}^{bn}_{k|k-1}}{\partial \boldsymbol{q}_{k|k-1}}\boldsymbol{g}^n. \tag{5.7}$$

The derivation of $\boldsymbol{H}_k$ can be found in appendix A. The EKF is performed in two steps, described in section 3.1.1. The time update is given by

$$\hat{\boldsymbol{q}}_{k|k-1} = \left[\mathcal{I}_4 + \frac{T}{2}\mathbb{S}(\hat{\boldsymbol{\omega}}_k)\right]\hat{\boldsymbol{q}}_{k-1|k-1}, \tag{5.8}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}^T_{k-1} + \boldsymbol{Q}_{k-1}. \tag{5.9}$$

The noise covariance [31] is

$$\boldsymbol{Q}_{k-1} = \mathbb{E}[\boldsymbol{\nu}_{k-1}\boldsymbol{\nu}^T_{k-1}] = \frac{T^2}{4}\bar{\mathbb{S}}(\hat{\boldsymbol{q}}_{\boldsymbol{k-1|k-1}})\Sigma_\omega\bar{\mathbb{S}}(\hat{\boldsymbol{q}}_{\boldsymbol{k-1|k-1}})^T, \tag{5.10}$$

where $\Sigma_\omega$ is the covariance matrix of the gyro measurements noise. The update step is performed as described in 3.1.1 with $\boldsymbol{H}_k$ according to (5.7). The measurement covariance $\boldsymbol{R}_k$ is given by the acceleration measurement noise $\boldsymbol{R}_k = \Sigma_a$. In order to achieve a stable filter [22], the covariance matrix $\boldsymbol{P}_0$ should be chosen a large positive value, initially set to $\boldsymbol{P}_0 = 10 \cdot \mathcal{I}_4$.

However, when updating the quaternion state, it was found that too much weight was put on the accelerometer measurement when in flight mode. The assumption of only acceleration from gravity was considered not completely true, so an extra parameter $\gamma = 0.01$ was added in the quaternion update, to give less impact on the acceleration correction. The quaternion measurement update was then adjusted to

$$\hat{\boldsymbol{q}}_{k|k} = \hat{\boldsymbol{q}}_{k|k-1} + \gamma\boldsymbol{K}_k\tilde{\boldsymbol{y}}_k. \tag{5.11}$$

From the EKF, the estimated orientation $\hat{\boldsymbol{q}}_k$ is extracted. The accelerometer measurement can then be rotated to the navigation frame, from where the gravitational acceleration can be extracted. This is done by first expressing the measurement as a quaternion, $\hat{\boldsymbol{a}}_{q,k} = (0 \quad \hat{\boldsymbol{a}}_k)$ and then rotate with the orientation,

$$\hat{\boldsymbol{a}}^n_{nn,k} = \hat{\boldsymbol{q}}_k \otimes \hat{\boldsymbol{a}}_{q,k} \otimes \hat{\boldsymbol{q}}_k. \tag{5.12}$$

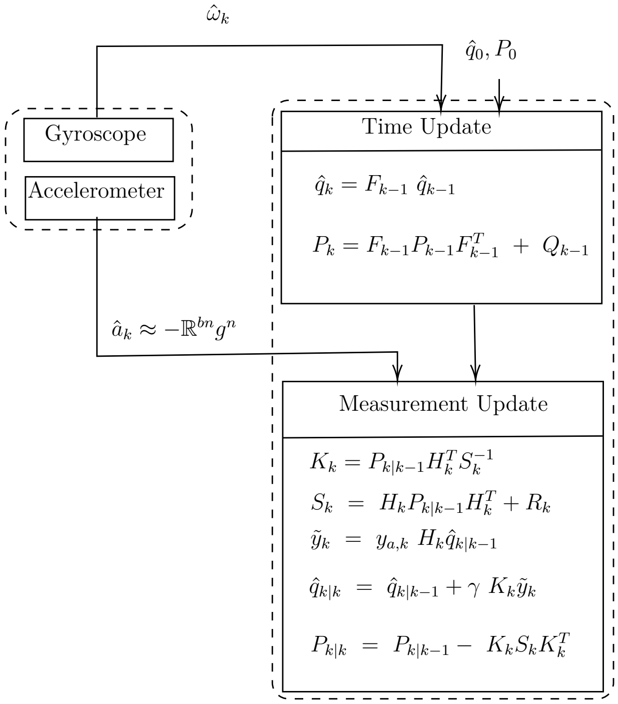



**Figure 5.2:** An overview of the extended Kalman filter algorithm used for orientation estimation.

## 5.4 Estimation of Position and Velocity

Note that the position and velocity estimations are only performed when the kite is in running mode. When it is parked, the position is assumed to be $\boldsymbol{p^n} = [0 \quad 0 \quad p_z^n]^T$ and the kite is assumed stationary $\boldsymbol{v}^n = [0 \quad 0 \quad 0]^T$.

This section describes how the spherical azimuth angle is estimated in order to obtain an approximate position of the kite. The Kalman filter uses this position estimate and the linear acceleration.

### 5.4.1 Spherical Angle Estimation

The kites position can be expressed in the navigation frame by position vector $\boldsymbol{p}^n = [p_x^n \quad p_y^n \quad p_z^n]^T$. It can also be expressed by spherical coordinates $r, \theta_{el}, \phi_{az}$, see figure 4.3.

The tether length is assumed to be constant. This means that the length of vector $|\boldsymbol{p}^n|$ from origin of navigation frame to the kite can be assumed to have constant length

$$|\boldsymbol{p}^n| = \sqrt{(p_x^n)^2 + (p_y^n)^2 + (p_z^n)^2} = r = (L + l). \tag{5.13}$$

The tether length is denoted $L$ and $l$ is the distance between the top joint and the center of the IMU.

From the pressure sensor, the position $p_z^n$ is indirectly provided. Since it is assumed a constant tether length, the spherical elevation angle $\theta_{el}$ can be calculated,

$$\theta_{el} = \arcsin \frac{p_z^n}{\sqrt{(p_x^n)^2 + (p_y^n)^2 + (p_z^n)^2}} = \arcsin \frac{p_z^n}{r}. \tag{5.14}$$

The aim is to be able to give a position input to the Kalman filter and thus obtain $p_x^n$ and $p_y^n$. For this, we need the azimuth angle $\phi_{az}$. The relation between the azimuth angle and the Cartesian coordinates can be described by

$$\phi_{az} = \arctan \frac{p_y}{p_x}. \tag{5.15}$$

In order to estimate the azimuth $\phi_{az}$ we study the figure-eight motion of the kite. The subscript is dropped for convenience, but should not be mixed with yaw angle of kite.

It can be noted that the kite passes the average depth, $\bar{z}_{mean}$, 4 times during a loop. Assuming kite is moving to the left in figure 5.3, it is first passing $\phi_{mean}$, then continuing to $\phi_{min}$, where it turns back and again crosses $\phi_{mean}$. At last it passes $\phi_{max}$ before reaching the starting point, ready for next loop.

The illustration shows that when $\bar{z}_{mean}$ is passed, three possible azimuth angles can be assumed, either $\phi_{min}, \phi_{mean}$ or $\phi_{max}$. The details how to decide which of these possibilities to choose will be presented below.

First, a start value of $\bar{z}_{0,mean}$ is set by the user, denoted as the target elevation. After some loops, the mean depth $\bar{z}_{k,mean}$ is updated by averaging over the actual measures of $p_z^n$.

### Choosing angle at mean depth

When the kite is close enough to mean depth $\bar{z}_{mean}$, within a tolerance of $\delta = 0.015$m, it is considered that one of the three extremes of $\phi$ is reached, denoted $\phi_i$. In order to decide which of the three angles is reached, we first look on how the depth is changing. We know that around $\phi_{mean}$ the kite is on its way down, meaning a negative change, $\Delta z < 0$. The opposite holds for $\phi_{min}$ and $\phi_{max}$. The depth change is the difference between the current depth and the depth $\Delta k$ samples ago, $\Delta z = p_{z,k} - p_{z,k-\Delta k}$, where $\Delta k = 20$ is chosen to be enough to avoid fluctuations due to noise. If depth change is positive at $p_z^n \approx \bar{z}_{mean}$, either $\phi_i = \phi_{min}$ or $\phi_i = \phi_{max}$.

In order to differentiate between those two extremes, we use the linear acceleration in y-direction. Intuitively, when the kite is approaching $\phi_{min}$, the maximum value of $p_y^n$ is reached before turning, meaning that the kite's acceleration is pointed in the negative y-direction. The opposite holds for $\phi_{max}$. We could therefore distinguish the two extremes by looking at the acceleration in y. Since the orientation estimation introduces estimation error, the mean acceleration is not approximately zero as expected over time. Instead of comparing in terms of negative and positive y-direction, relation to the mean of the acceleration in y, denoted $\bar{a}_y$, is used. Note that the mean only is calculated over data point when in running mode. In conclusion, if $a_y^n < \bar{a}_y$, we assumed to have reached $\phi_i = \phi_{min}$ and otherwise $\phi_i = \phi_{max}$.

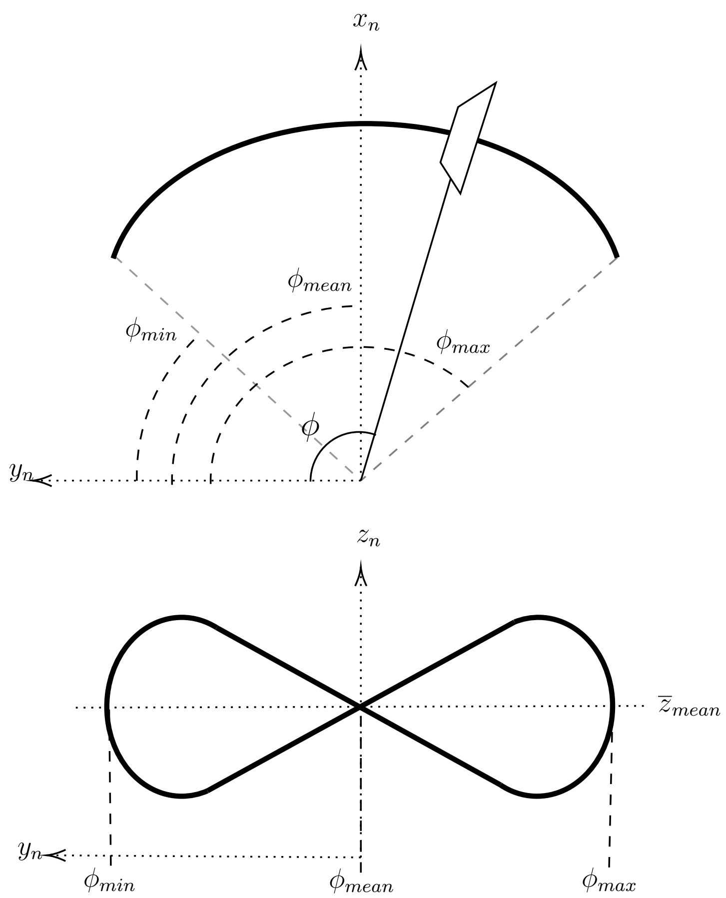



**Figure 5.3:** Projection of the kites figure-eight motion on the *xy*-plane and *yz*-plane, where different values of $\phi_{az}$ are marked.

### Interpolation

The numerical values of $\phi_{min}, \phi_{mean}$ and $\phi_{max}$ are obtained from *computational fluid dynamics* (CFD) models of the kite, provided by Minesto. In those computations, the angles of interest are approximated to $-\frac{\pi}{8}, 0$ and $\frac{\pi}{8}$ respectively.

When the kite has passed an extreme or is outside of the tolerance interval $\delta$ around the mean, the azimuth angle needs to be interpolated. This is done by looking at the two previous reached extreme points. If the most recent extreme point passed was $\phi_i$, the table 5.1 shows the next predicted extreme $\phi_{i+1}$, along with the total change between the points.

One problem is that the number of samples varies between every extreme point, since the change of $\phi$ depends on the velocity and the distance travelled varies. However, for simplicity constant step size $\Delta$ is chosen between two extreme points. The number of samples between two extremes, referred to as the interval length $I_\phi$, is first set to a default value of $I_{\phi,0} = 600$. It is though continuously updated by the mean interval when more extremes have been reached.

The step size can then be expressed as

$$\Delta = \frac{\Delta\phi}{I_\phi}, \tag{5.16}$$

which then gives $\phi_k = \phi_{k-1} + \Delta$. When a new extreme is reached, the angle is set to that value, $\phi_k = \phi_i$.

An indication of the algorithm correctness is performed by comparing the output compared with CFD data, seen in figure 5.4. As can be seen, the initial interval length does not match the true interval length completely, giving a discontinuity when a new extreme point is found. This evens out when averaging over more interval lengths and the discontinuities shrink. It can be noted that the estimate starts when the first extreme point is found. When $\phi_k = \phi_{az,k}$ is found, conversion of position from spherical to Cartesian coordinates can be performed,

$$\begin{aligned} p_x^n &= r \cos\theta_{el} \cos\phi_{az} \\ p_y^n &= r \cos\theta_{el} \sin\phi_{az} \\ p_z^n &= r \sin\theta_{el}. \end{aligned} \tag{5.17}$$

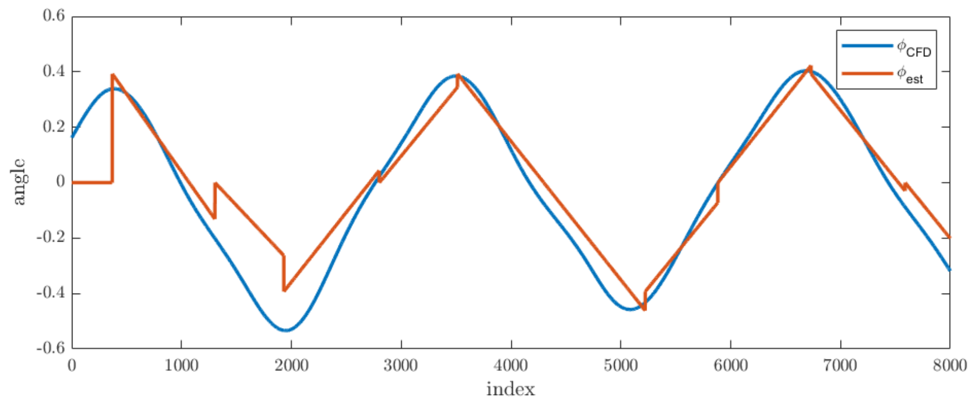



**Figure 5.4:** The estimated azimuth angle $\phi_{est}$ and azimuth angle extracted from CFD data.

**Table 5.1:** Scheme of how to predict next extreme point so that interpolation step size can be chosen.

| $\phi_{i-1}$ | $\phi_i$ | $\phi_{i+1}$ | $\Delta\phi$ |
|---|---|---|---|
| mean | max | mean | $-\pi/8$ |
| mean | min | mean | $+\pi/8$ |
| min | mean | max | $+\pi/8$ |
| max | mean | min | $-\pi/8$ |

### 5.4.2 Kalman Filter Design

From the EKF, the orientation is extracted and the acceleration is rotated and the linear acceleration is separated from gravitation. As discussed in the model description, the acceleration can be used as an input to a KF togheter with the estimated position from the spherical angle estimation.

A block diagram of the process can be seen in figure 5.5, where the assumption of a constant covariance matrix $\boldsymbol{Q}$ is made [20]. We can then use the steady state Kalman gain $\boldsymbol{K}$ and state covariance $P_\infty$.

The covariance matrix $\boldsymbol{Q}$ is the covariance of the accelerometer, assumed to vary slowly over time. This can be confirmed when looking closer at the noise variance in park mode. The position estimation covariance matrix $\boldsymbol{R}$ is designed manually. Considering that the pressure sensor is giving a good estimate of $p_z^n$, the variance of this signal should be relatively small. The estimations of $p_x^n, p_y^n$ on the other hand, come from the azimuth angle estimate and should not be trusted as much, indicating higher variances. After some tuning, where the constraints of constant tether length is weighted against not trusting the azimuth angle estimation too much, $\boldsymbol{R}$ is set to

$$\boldsymbol{R} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & (0.02)^2 \end{bmatrix} \tag{5.18}$$

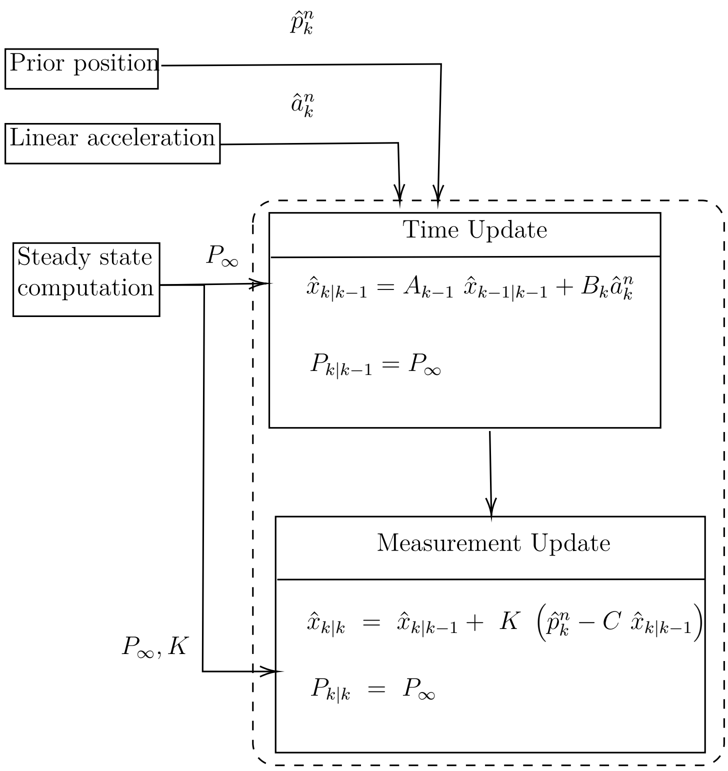



**Figure 5.5:** Overview of the KF used for estimating position and velocity.

# 6
# Results

This section provides the results from the estimation algorithm, together with some interpretation of the results. The data was collected in weak flow in February 2020 by Minesto, with the use of an old test kite. The use of an old test kite in combination with weak tide lead to lower kite velocities, but still serves as valuable data to perform this analysis on. In the logs analysed, the tether was attached to a barge on the ocean surface instead of a bottom joint on the seafloor. In the log presented, kite is in parking mode the first 4 minutes before it enters flight mode for approximately 20 minutes. The tether length was approximately 16 m.

## 6.1 Orientation Estimation

The resulting orientation estimation can be seen in figure 6.1 and is presented in Euler angles. The start pitch was set to approximate 60°. As can be noted, all three angles drift slightly over time. With the current approach, the gyro bias and noise are treated as process noise instead of measurement noise, since the gyro data is treated as input. The chosen approach is reducing the state vector and leading to efficient filter implementation [33], but leads to less exact drift estimation. This shows that it can be favorable to also add the gyro's drift in the state vector and implement a 7-D state vector in the EKF, done in for example [34].

In figure 6.2 an arbitrary smaller interval have been chosen. It can be seen that all angles follow the elevation in some sense. The maximum and minimum of the roll $\psi$ can be found to coincide roughly with maximum of elevation. The pitch follows the same periodicity as the elevation, but slightly in advance.

The yaw is the angle which has the greatest span, between approximately $\pm 120°$. It can also be seen that the yaw $\phi \approx 0$ at approximately $\bar{z}_{mean}$, which according to the changes in elevation indicate that this happens at $\phi_{az,mean}$, in other word at the center of the loop. This is in line with the assumptions in the control system today - the yaw should be zero in the middle of the eight and also according to CFD data. The maximum and minimum yaw angles $\phi$ are also found at $\bar{z}_{mean}$, in spherical coordinates at $\phi_{az,min}$ and $\phi_{az,max}$. This is also in line with CFD data.

## 6.2 Linear Acceleration

The result of the linear acceleration can be seen in figure 6.3. One can note that when the kite is in park mode, the acceleration is approximately zero, which is expected if the gravitational acceleration compensation has been done successfully. When in

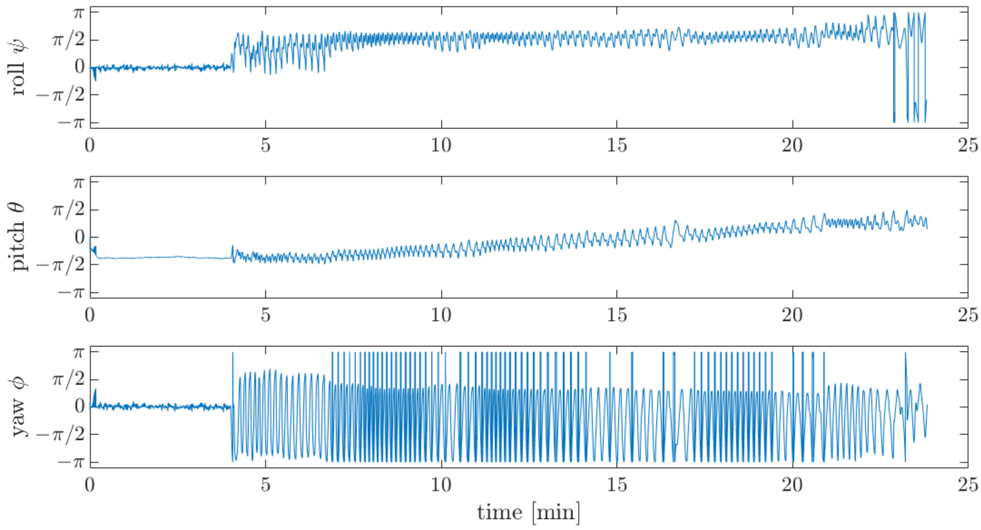



**Figure 6.1:** Orientation in euler angles roll $\psi$, pitch $\theta$ and yaw $\phi$.

flight mode, the mean of the acceleration in the log is $\bar{\boldsymbol{a}} \approx [0.04, 0.29, 0.00]$, but some drift can be seen. The drift can be partly explained by the noise of the accelerometer, but also due to the orientation error. The error in the orientation propagates to the linear acceleration when compensating for gravitational acceleration.

Note that the centripetal acceleration has not been compensated for. However, this should not explain the drift. The centripetal acceleration give rise to varying errors depending on location along the loop and velocity.

## 6.3 Position and Velocity Estimation

First, the result of the azimuth angle estimate is shown, followed by the obtained position and velocity.

### 6.3.1 Azimuth Angle Estimate

The azimuth angle estimation can be seen in figure 6.4, over the whole time span and a time interval between the minutes $5-10$. In this smaller interval we can see that the azimuth estimation is behaving as expected, going from $\phi_{max}$, passing $\phi_{mean}$ and then continuing to $\phi_{min}$. However, at earlier and later time instances, there are some issues keeping this pattern, where $\phi_{max}$ and $\phi_{min}$ are mixed up.

At $t < 5$ min the angle estimation performs poorly. This can be explained by assuming that the mean elevation $\bar{z}_{mean}$ has not yet settled to the target elevation. Also there is an initial phase before the cross-current motion stabilises. At later time instances, approximately at $t > 17$ min, we can see a change in behaviour. It seems like the estimation algorithm tends to have problems in distinguishing between $\phi_{max}$ and $\phi_{min}$.

This can be linked with the slowly varying bias affecting $a_y^n$ to drift and that the assumption of $a_y^n > \bar{a}_y : \phi_i = \phi_{max}$ and otherwise $\phi_i = \phi_{min}$, is not robust

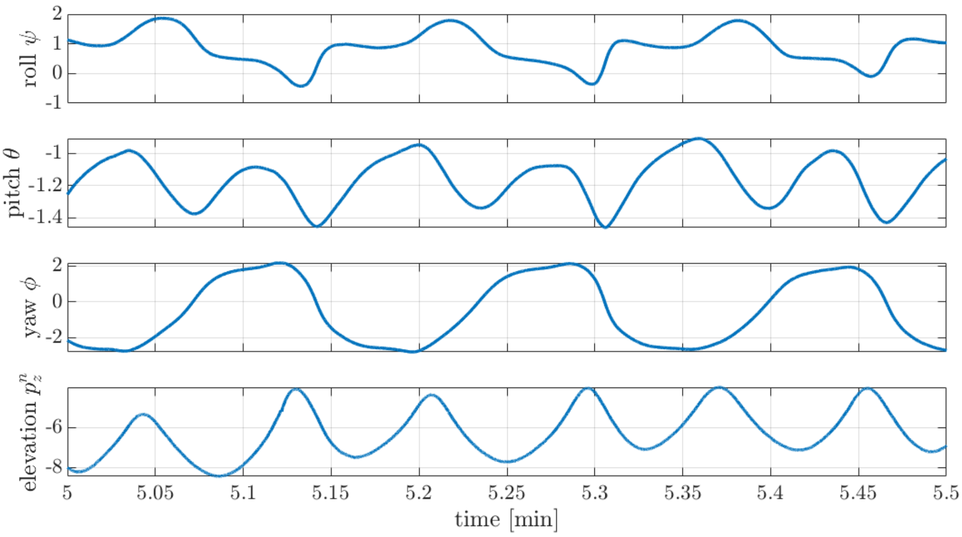



**Figure 6.2:** Orientation in Euler angles roll $\psi$, pitch $\theta$ and yaw $\phi$. Extraction of approximately three loops during approximately half a minute. Here the elevation is also included to be able to relate to position on the loop.

enough for bias. Different strategies to overcome this issue have been investigated without any improvement reached. The best result obtained has been the approach explained here.

### 6.3.2 Position Estimate

The position estimated by the steady state Kalman filter can be seen in figure 6.5. It can be noted that there is a transient when the kite switches from park mode to flight mode and that the position in especially x-direction drifts slightly.

After approximately 17 minutes, the position in y-direction change behaviour. This is related to when the azimuth angle estimation is bad differentiating between $\phi_{max}$ and $\phi_{min}$, discussed earlier.

In figure 6.6 a 3-D plot of the kite can be seen, around time $t \approx 10$ min. From the figure it can be seen that the kite is following a figure eight-up motion, even though some drift occur, especially in x-direction. However, it is promising that we have been able to imitate the expected behaviour of the kite - namely that the kite is moving in a figure-eight pattern.

### 6.3.3 Velocity Estimate

The velocity estimate can be seen in figure 6.7. Like the position estimate, there is a transient when kite switches from park to flight mode and the velocity drift slightly especially in x-direction. Since velocity is directly correlated to position change, this is not surprising, since also position in x drifts.

Figure 6.8 shows the direction of the velocity vector at different positions at one randomly chosen loop in the time interval $t = 460 - 468$s. According to CFD data, the power output is maximum when kite is on its way to the middle, and

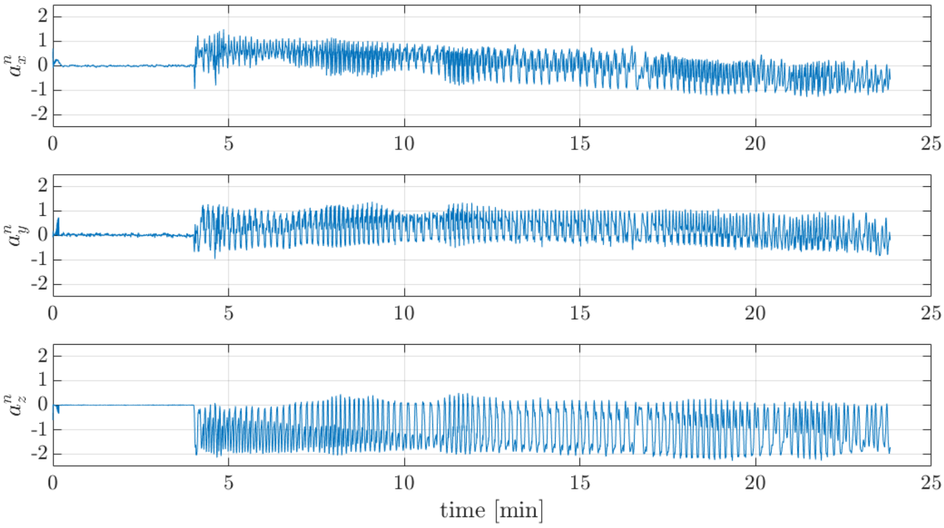



**Figure 6.3:** Acceleration in navigation frame with gravitational component extracted.

thus the same with the speed. But this result show that the maximum velocity is obtained in the outer parts of the pattern. This can be partly explained by the fact that the centripetal acceleration has not been compensated for. However, even if the magnitude of the velocity vector needs to be further looked into, the direction seems realistic.

Figure 6.9 shows the absolute velocity of the estimation and speed from an external speed measurement device mounted on the kite[1]. It can be noted that the trend is that the estimated velocity is slightly smaller than measured. However, one important distinction between the two is that the device measures speed *through water* and not speed *over ground*, which is what $|\boldsymbol{v}_n|$ measures. This means that an offset due to the flow velocity is expected.

Except from the offset, by looking at a smaller interval in figure 6.9, it can be seen that the peaks are correlated. However, the estimated speed is fluctuating more than the measured speed.

[1]The low kite speed seen in figure is due to use of an old test kite in combination with weak tide.

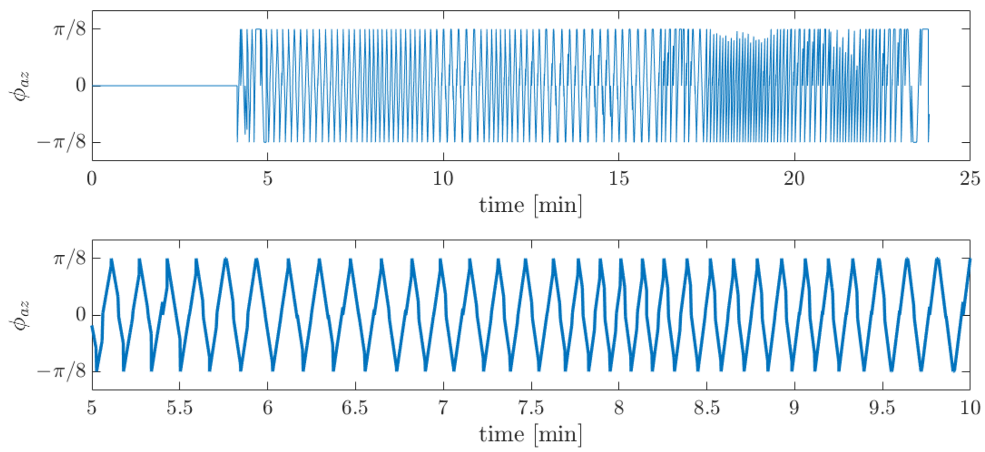



**Figure 6.4:** The estimated azimuth angle from $p_z^n$ from pressure sensor and estimated acceleration in y-direction $\hat{a}_y^n$. Note the two different x axes in figures, where bottom figure shows a smaller time interval.

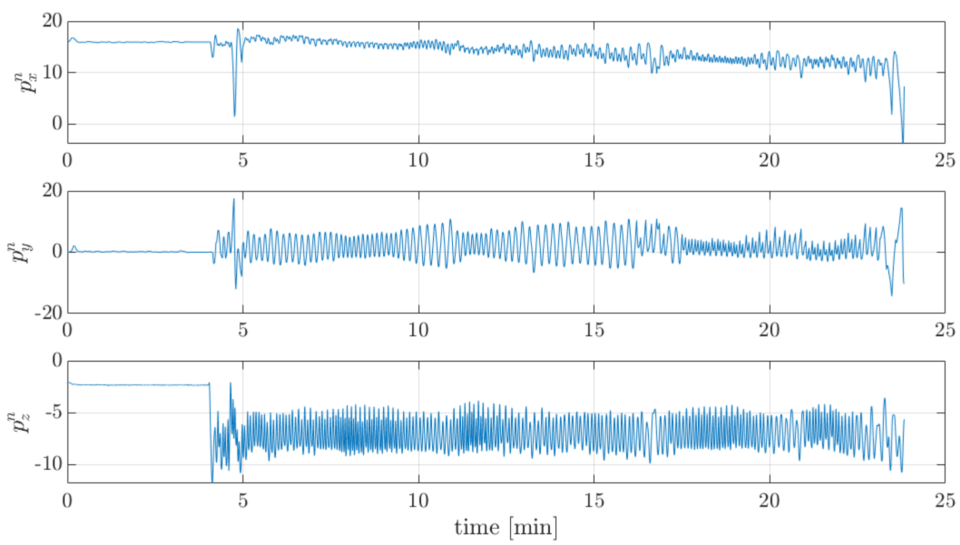



**Figure 6.5:** The kite's position in navigation frame.

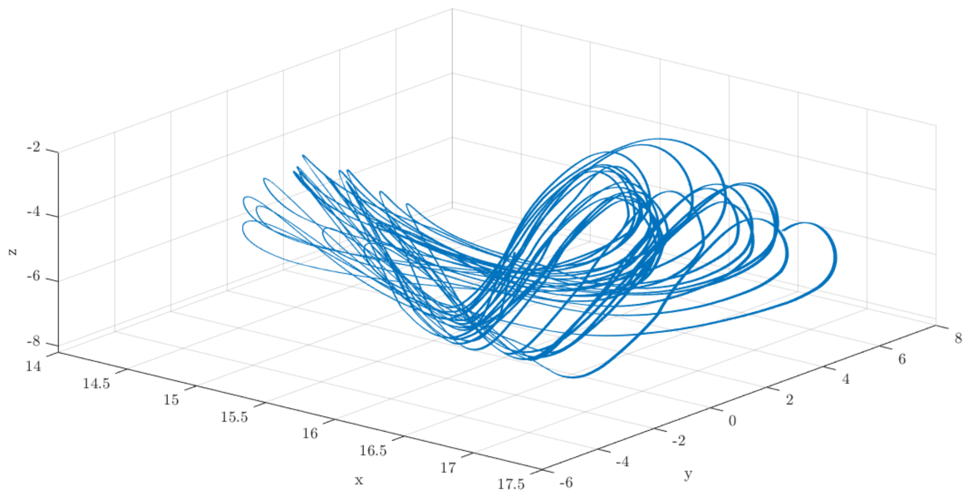



**Figure 6.6:** The kite's position in navigation frame, seen from 3-D perspective.

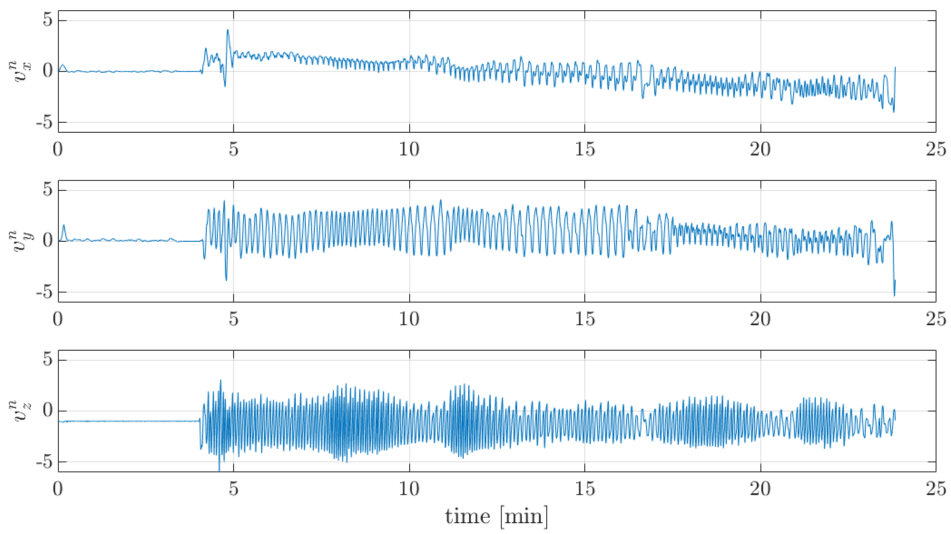



**Figure 6.7:** The kite's velocity in navigation frame.

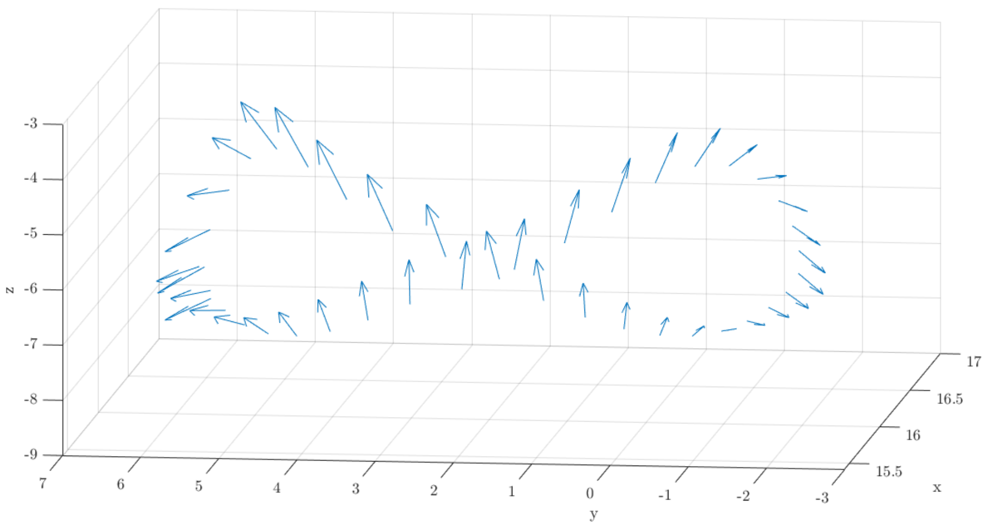



**Figure 6.8:** The kite's velocity vector during a loop with weak flow in navigation frame.

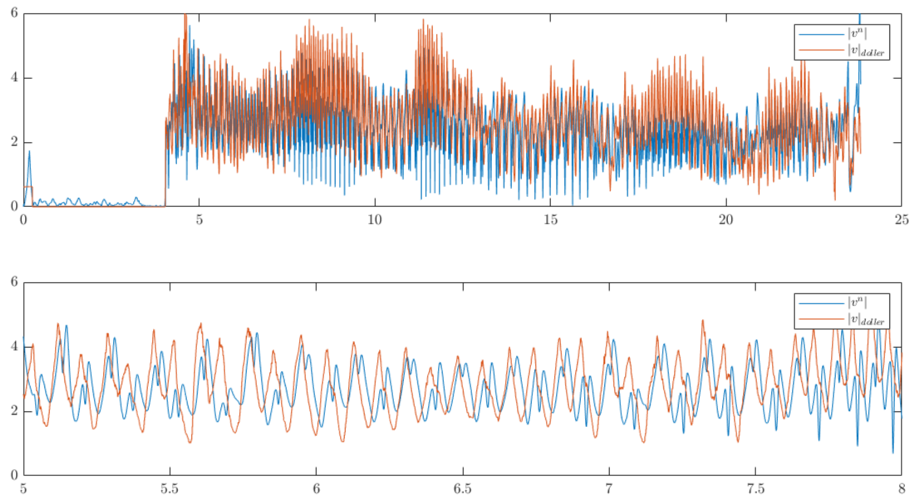


**Figure 6.9:** The kite speed and the measured speed mounted on the kite at weak flow. Note the two different time scales, where the bottom plot is a zoom.

# 7

# Closure

## 7.1 Conclusion

The result is promising when it comes to estimating pose and velocity of the kite. It can be seen that the position and velocity estimates are constrained and even if there are biases in the measurement, the estimations are held within a realistic interval. If position and velocity would be obtained directly by integration of acceleration, those values would drift way more, several meters every minute. The same holds for the orientation, where the EKF give rise to much less drift than just integrating the gyro measurement. This proves the necessity of using filters in order to handle the noise.

Given that there is no ground truth data available, it is hard to quantify the result. In the end, power output is the best measure in order to obtain a result of the performance. If the developed pose estimation is giving higher power output than the current system, it can be stated that the algorithm is more successful in following the optimal trajectory and thus better in estimation pose. However, the algorithm has not yet been implemented and tested during operation, so no such comparison could be done.

Even if the result only could be evaluated on a small amount of data, the orientation is periodic and follows the elevation in an expected manner. When rotating the accelerometer according to the estimated orientation and compensating for gravitational acceleration, the resulting acceleration is close to zero mean, which is expected if orientation estimate has been done successfully.

However, drift occur over time. In order to battle this drift, also including the bias in the state vector is proposed, with the drawback of more computational heavy algorithm. Access to more data is also crucial in order to evaluate the robustness of the algorithm along with better estimate of how the noise of the sensors can be determined.

When it comes to position and velocity estimate, after the initial transient when entering flight mode, the result is that the kite is moving in a figure-eight pattern, where the velocity vector is directed in a realistic manner, approximately tangential to the figure eight pattern. However, the azimuth angle estimation performed is not robust enough to be able to distinguish the azimuth angle extreme points for a longer time period. When the azimuth angle estimation is performing badly, the position and velocity estimates are directly affected. This means that in order to get a more robust position and velocity estimation, more effort need to be done with developing the azimuth angle estimate.

## 7.2 Future work

This work should be seen as an introduction to sensor fusion in Minesto's Deep Green and more work needs to be done in order to get an ideal pose and motion estimation. This includes investigating the azimuth angle estimation more in depth to make it more robust against noisy data. Also evaluating the system on more data to see if there are some common pitfalls and problems with the system that could not be found by the limited data available.

Also, in this work the magnetometer data has been fully discarded. Even if the generator is influencing and disturbing the magnetic field greatly in flight mode, it could possibly be used in parking mode in order to estimate the initial heading. The magnetometer can ideally give information of the magnetic north, the same way as the gravitational acceleration gives information of the direction of the center of earth.

As mentioned earlier, it would be interesting to see if including the gyro bias in the state vector would reduce the drift in orientation. Including the bias estimation is crucial in order to get a better orientation estimation. Since the orientation error propagates to the other estimates, the overall performance could improve significantly due to reduced drift.

Another source of error in the orientation estimation besides the bias, is the linearization error when applying the EKF. A first-order linearization of the nonlinear system is performed, which introduces errors. An alternative approach is instead to use a *Unscented Kalman filter* (UKF), which reduces the linearization error by achieving third-order accuracy. Instead of using derivatives, the state distribution is represented by so called sigma points, a minimal set of carefully chosen samples [35]. For the interested reader, further discussions of the differences between the EKF and UKF can be found in [36].

# Bibliography


[1] Minesto AB, "The future of renewable energy," 2019. https://minesto.com/our-technology (Last accessed 2020-02-07).

[2] Minesto AB, "Ocean Energy," 2020. https://minesto.se/ocean-energy (Last accessed 2020-04-13).

[3] D. J. Olinger and Y. Wang, "Hydrokinetic Energy Harvesting Using Tethered Undersea Kites," *Renewable Sustainable Energy*, vol. 043114, 2015.

[4] C. W. F. Roger H. Charlier, *Current from Tidal Current*, vol. 54. Springer, 1984.

[5] A. Bahaj and L. Myers, "Analytical Estimates of the Energy Yield Potential From the Alderney Race (Channel Islands) Using Marine Current Energy Converters," *Renewable Energy*, vol. 29, no. 12, pp. 1931 – 1945, 2004.

[6] V. Khare, C. Khare, S. Nema, and P. Baredar, "Introduction of Tidal Energy," in *Tidal Energy Systems* (V. Khare, C. Khare, S. Nema, and P. Baredar, eds.), pp. 41 – 114, Elsevier, 2019.

[7] J. Manwell, J. Mcgowan, and A. Rogers, *Wind Energy Explained: Theory, Design and Application, Second Edition*, vol. 30, pp. 24–35. John Wiley & Sons, Incorporated, 2006.

[8] *Modeling and Simulation of Tethered Undersea Kites*, vol. 1 of *Energy Sustainability*, 06 2016. V001T07A001.

[9] H. Li, D. J. Olinger, and M. A. Demetriou, "Control of a Tethered Undersea Kite Energy System Using a Six Degree of Freedom Model," *54th IEEE Conference on Decision and Control (CDC)*, pp. 688–693, 02 2015.

[10] M. L. Loyd, "Crosswind Kite Power," *J. Energy*, vol. 4, no. 3, pp. 106–111, 1980.

[11] M. Erhard and H. Strauch, "Control of Towing Kites for Seagoing Vessels," *IEEE Transactions on Control Systems Technology*, vol. 21, p. 1629–1640, Sep 2013.

[12] M. Landberg, "A Submersible Plant." Patent US 8 246 293, 21 Aug 2012.

[13] Ocean Energy Europe, "Tidal current," 2020. https://www.oceanenergy-europe.eu/ocean-energy/tidal-energy/ (Last accessed 2020-04-13).

[14] G. M. Dadd, D. A. Hudson, and R. A. Shenoi, "Determination of Kite Forces Using Three-Dimensional Flight Trajectories for Ship Propulsion," *Renewable Energy*, vol. 36, no. 10, pp. 2667–2678, 2011.

[15] Minesto AB, "Deep green: unik teknologi som låser upp en hittills outnyttjad naturresurs," 2020. https://minesto.com/investor/varfor-investera-i-minesto/2-deep-green (Last accessed 2020-04-11).

[16] D. Titterton and J. Weston, "Strapdown Inertial Navigation Technology," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 7, pp. 33–34, 2005.

[17] M. Kok, J. D. Hol, and T. B. Schön, "Using Inertial Sensors for Position and Orientation Estimation," *Foundations and Trends in Signal Processing*, vol. 11, no. 1-2, pp. 1–153, 2017.

[18] L. Wang, Z. Zhang, and P. Sun, "Quaternion-Based Kalman Filter for AHRS Using an Adaptive-Step Gradient Descent Algorithm," *International Journal of Advanced Robotic Systems*, vol. 12, no. 9, p. 131, 2015.

[19] L. Svensson, "Course in Sensor Fusion and Nonlinear Filtering (SSY345)." Online Lectures, September 2014. (Last accessed 2020-05-12).

[20] E. Bachmann Mehammer, M. Føre, T. Sauder, V. Bruno Chabaud, and T. Parisini, "Kalman Estimation of Position and Velocity for ReaTHM Testing Applications," *Journal of Physics: Conference Series*, vol. 1104, 10 2018.

[21] B. Friedland, "Kalman Filters: Optimum Observers," in *Control System Design - An Introduction to State-Space Methods*, ch. 11, Dover Publications, 1986.

[22] K. Feng, J. Li, Z. Xiaoming, C. Shen, Y. Bi, T. Zheng, and J. Liu, "A New Quaternion-Based Kalman Filter for Real-Time Attitude Estimation Using the Two-Step Geometrically-Intuitive Correction Algorithm," *Sensors*, vol. 17, p. 2146, 09 2017.

[23] M. B. Del Rosario, N. H. Lovell, and S. J. Redmond, "Quaternion-Based Complementary Filter for Attitude Determination of a Smartphone," *IEEE Sensors Journal*, vol. 16, no. 15, pp. 6008–6017, 2016.

[24] R. Mahony, T. Hamel, and J.-m. Pflimlin, "Non-linear Complementary Filters on the Special Orthogonal Group," *IEEE Transactions on Automatic Control*, 06 2008.

[25] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *IEEE International Conference on Rehabilitation Robotics*, 2011.

[26] S. Ludwig, "Optimization of Control Parameter for Filter Algorithms for Attitude and Heading Reference Systems," pp. 1–8, 07 2018.

[27] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs," pp. 19302–19330, 2015.

[28] A. Al-Fahoum and M. Abadir, "Design of a Modified Madgwick Filter for Quaternion-Based Orientation Estimation Using AHRS," vol. 10, pp. 174–186, 10 2018.

[29] L. Fagiano, K. Huynh, B. Bamieh, and M. Khammash, "On Sensor Fusion for Airborne Wind Energy Systems," *IEEE Transactions on Control Systems Technology*, vol. 22, 11 2012.

[30] Blue Robotics Inc, "Bar30 High-Resolution 300m Depth/Pressure Sensor," 2020. `https://bluerobotics.com/store/sensors-sonars-cameras/sensors/bar30-sensor-r1/` (Last accessed 2020-05-12).

[31] J. K. Lee, E. J. Park, and S. N. Robinovitch, "Estimation of Attitude and External Acceleration Using Inertial Sensor Measurement During Various Dynamic Conditions," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 8, pp. 2262–2273, 2012.

[32] G. Hendeby, "Orientation Estimation using Smartphone Sensors," March 2019. https://www.control.isy.liu.se/student/tsrt14/file/orientation.pdf (Last accessed 2020-05-26).

[33] A. M. Sabatini, "Kalman-Filter-Based Orientation Determination Using Inertial/Magnetic Sensors: Observability Analysis and Performance Evaluation," *Sensors*, no. 11, 2011.

[34] M. Hale, P. Vergez, M. Meerman, and Y. Hashida, "Kalman Filtering and the Attitude Determination and Control Task," p. 21, 09 2004.

[35] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, 2000.

[36] R. Kandepu, B. Foss, and L. Imsland, "Applying the unscented kalman filter for nonlinear state estimation," *Journal of Process Control*, vol. 18, no. 7, pp. 753 – 768, 2008.

[37] J. D. Hol, T. Schön, F. Gustafsson, and B. Bernhardsson, *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. Linköping: Linköping University Electronic Press, 2011. Diss. Linköping : Linköpings universitet, 2011.

# A

# Derivation of Jacobians in EKF

Quaternion $\boldsymbol{q}$ can be converted to a rotation matrix $\mathbb{R}$ by the following relation

$$\mathbb{R} = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix} \tag{A.1}$$

Differentation of this expression gives,

$$Q_0 = \frac{\partial \mathbb{R}}{\partial q_0} = 2 \begin{bmatrix} 2q_0 & -q_3 & q_2 \\ q_3 & 2q_0 & -q_1 \\ -q_2 & q_1 & 2q_0 \end{bmatrix}, \tag{A.2}$$

$$Q_1 = \frac{\partial \mathbb{R}}{\partial q_1} = 2 \begin{bmatrix} 2q_1 & q_2 & q_3 \\ q_2 & 0 & -q_0 \\ q_3 & q_0 & 0 \end{bmatrix}, \tag{A.3}$$

$$Q_2 = \frac{\partial \mathbb{R}}{\partial q_2} = 2 \begin{bmatrix} 0 & q_1 & q_0 \\ q_1 & 2q_2 & q_3 \\ -q_0 & q_3 & 0 \end{bmatrix}, \tag{A.4}$$

$$Q_3 = \frac{\partial \mathbb{R}}{\partial q_3} = 2 \begin{bmatrix} 0 & -q_0 & q_1 \\ q_0 & 0 & q_2 \\ q_1 & q_2 & 2q_3 \end{bmatrix}. \tag{A.5}$$

The expression $\boldsymbol{H}$ used in the EKF then becomes

$$\boldsymbol{H}_k = \frac{\partial}{\partial q_k} \mathbb{R}_k^{bn} \boldsymbol{g}^n = \frac{\partial \mathbb{R}_{k|k-1}^{bn}}{\partial q_{k|k-1}} \boldsymbol{g}^n = \begin{bmatrix} Q_0 & Q_1 & Q_2 & Q_3 \end{bmatrix}^T \boldsymbol{g}^n. \tag{A.6}$$

# B

# Orientation Dynamics

The expression (4.10) can be integrated in order to obtain the orientation at time step $t+T$, with the assumption that the angular velocity is constant in the interval $(t, t+T)$[37]. The result is then the following expression,

$$q_{t+1} = q_t \otimes \exp\Big(\frac{T}{2}\omega_t\Big). \tag{B.1}$$

This expression is going to be derived and approximated next. The continuous orientation change is described by equation (4.10) or on matrix form by (4.11). In order to discretize the model we assume that $t \in (t_{k-1}, t_k]$ and thus starts with (B.2),

$$\dot{q}(t) = \frac{1}{2}S(\omega_t)q(t) = Aq(t). \tag{B.2}$$

The solution to this equation is a non-linear function, meaning that in general we have no analytical solution to the filtering equations given above. However, we can construct an approximate solution that is linear. By multiplying the expression with $\exp(-At)$ and moving equality one can simplify the expression using the chain rule in reverse,

$$\exp(-At)\dot{q}(t) - \exp(-At)Aq(t) = 0 \quad \Rightarrow \quad \frac{d}{dt}\exp(-At)q(t) = 0, \tag{B.3}$$

and when integrating the expression one gets,

$$\begin{aligned}\int_t^{t+T} \frac{d}{dt}\exp(-At)q(t)dt &= \Big[\exp\Big(-A(\tau)\Big)q(\tau)\Big]_{\tau=t}^{\tau=t+T} \\ &= \exp\Big(-A(t+T)\Big)q(t+T) - \exp(-At)q(t) = 0.\end{aligned} \tag{B.4}$$

By multiplication of $\exp(At)$ and rearranging the terms along with first order Taylor expansion, we get the following expression,

$$q(t+T) = \exp(AT)q(t) \approx (I + AT)q(t). \tag{B.5}$$

We now have a discrete approximated linear model. Measurements $k = 1, ..., N$ from the gyro is assumed to be obtained every $T$ time step, which means that $t + T = k$ and $t = k - 1$. It has previously been discussed that the measurements from the gyroscope contains process noise $\nu$. At step $k$ the gyro signal therefore consists of $\omega_k + \nu_k$. Inserting this into (B.5) gives,

$$\begin{aligned}q_k &= \Big[I + AT\Big]q_{k-1} = \Big[I + \frac{T}{2}S(\omega_{k-1} + \nu_{k-1})\Big]q_{k-1} \\ &= \Big[I + \frac{T}{2}S(\omega_{k-1})\Big]q_{k-1} + \frac{T}{2}\bar{S}(q_{k-1})\nu_{k-1},\end{aligned} \tag{B.6}$$

where the last equality is obtained from (4.11).