NTNU – Trondheim
Norwegian University of
Science and Technology

Indoor Navigation System and Suspended Load Control for Multirotors

Recep Cetin

Master of Science in Cybernetics and Robotics
Submission date: June 2015
Supervisor: Thor Inge Fossen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics

MSC THESIS DESCRIPTION SHEET

Name:	Recep Cetin
Department:	Engineering Cybernetics
Thesis title (Norwegian):	Innendørs navigasjonssystem og kontrol av hengende last for multirotor systemer
Thesis title (English):	Indoor Navigation System and Suspended Load Control for Multirotors

Thesis Description: The purpose of the thesis is to design an accurate navigation system for indoor navigation for multirotors and develop a control system for multirotor suspended load system.

The following items should be considered:

1. Define the scope of the thesis and clarify what your contributions are.
2. Development and testing of indoor radio navigation system for multirotors.
3. Development and experimental testing of state estimators for accurate positioning.
4. Demonstrate that one multirotors can lift a small payload of limited weight. The controller should be based on the Pixhawk autopilot.
5. Design a swing reducing control scheme for a multirotor suspended load system.
6. Implement and test the swing reducing control scheme with a small payload utilizing the Pixhawk autopilot.
7. Conclude your results.

Start date:	2015-01-05
Due date:	2015-06-15

Thesis performed at:	Department of Engineering Cybernetics, NTNU
Supervisor:	Professor Thor I. Fossen, Dept. of Eng. Cybernetics, NTNU
Co-Supervisor:	Kristin Klausen, Dept. of Eng. Cybernetics, NTNU

Abstract

This thesis considers a high precision indoor navigation system and a suspended load control for multirotor systems. A complete overview of a multirotor type Unmanned Aerial Vehicle (UAV) with all the necessary hardware and software is presented. A local navigation system that is low-cost, light weight, easy to deploy, and works indoor is developed. The solution is based on Extended Kalman Filter (EKF) with sensor fusion approach to filtering using all available on-board sensory data, and uses the principle of trilateration utilizing range measurements from radio transceivers. The presented EKF algorithm is capable of outlier detection and rejection, and solves the divergence problem with adaptively weighted covariance based on Fuzzy Logic Adaptive System (FLAS). The swing problem of a suspended load system is solved with two separate controllers that are independent from the multirotor autopilot. The well known open-loop input shaping filter is used to avoid initial swing excitation in the transition between hover and forward flight, while a time-delayed feedback control damps the remaining oscillations on the load based on load angle measurements. The performance of the navigation system and the suspended load control is verified by simulations and in-flight experiments.

Sammendrag

(*Norwegian translation of the abstract*)

Denne masteroppgaven tar for seg et høypresisjons innendørs navigasjonssystem og kontroll av hengende last for multirotorsystemer. En komplett oversikt over et multirotorsystem med all nødvendig maskinvare og programvare er presentert. Et navigasjonssystem som er enkelt å sette opp, billig og lett, og som fungerer innendørs er utviklet. Løsningen er basert på Extended Kalman Filter (EKF), der avstandsmålinger fra radiomottakere og trilatereringsprinsippet er benyttet. Algoritmen er i stand til å oppdage og forkaste avvikende målinger, og løser divergensproblemet til EKF med adaptiv vekting av kovariansen basert på fuzzylogikk. Kontrollproblemet til hengende lastsystemet er løst ved å bruke to separate kontrollere som er uavhengig av autopiloten til multirotorsystemet. En foroverkopling i open sløyfe er brukt til å dempe sving eksitasjon, mens en tidsforsinket tilbakekoplingskontroller basert på lastvinkelen demper gjenværende svingninger. Ytelsen av både navigasjonssystemet og kontrolleren er verifisert med simuleringer og flyvninger.

Preface

This paper concludes my master thesis written for the Unmanned Aerial Vehicles Laboratory (UAV-Lab) at the Department of Engineering Cybernetics, NTNU. The project is initiated by *Centre for Autonomous Marine Operations and Systems* (AMOS) and part of the Autonomous UAV project. It is assumed that the target audience has prior basic knowledge in the field of systems and control.

I would like to thank my supervisor, Professor Thor I. Fossen for valuable feedback and guidance. A special thanks to my co-supervisor, PhD candidate Kristian Klausen for his never-ending support throughout the project. Next, PhD candidate Krzysztof Cisek from Dept. of Engineering Cybernetics, NTNU for cooperation on Atmel equipment and engineer Lars Semb from Dept. of Engineering Cybernetics, NTNU for piloting during in-flight experiments. I will also like to extend my gratitude to Professor Fredrik Gustafsson from Dept. of Electrical Engineering at Linköping University for initiating cooperation with their research group at Linköping University. Finally, i would like to thank my brother for his enthusiasm, encouragement and guidance throughout my studies.

Sincerely

Recep Cetin

Contents

List of Figures

List of Tables

List of Abbreviations

BBB BeagleBone Black.

CO Center of Origin.

CRC Cyclic Redundancy Check.

DARE Discrete-time Algebraic Ricatti Equation.

DoF Degrees of Freedom.

DQF Distance Quality Factor.

DUNE DUNE: Uniform Navigational Environment.

EKF Extended Kalman Filter.

ESC Electronic Speed Controller.

FLAC Fuzzy Logic Adaptive Controller.

FLAS Fuzzy Logic Adaptive System.

GCS Ground Control Station.

GLONASS Globalnaya Navigatsionnaya Sputnikovaya Sistema (Global Navigation Satellite System).

GNSS Global Navigation Satellite System.

GPS Global Positioning System.

GUI Graphical User Interface.

HITL Hardware in the Loop.

IMC Inter Module Communication.

IMU Inertial Measurement Unit.

INS Inertial Navigation Systems.

KF Kalman Filter.

LP Low Pass.

LSTS Laboratório de Sistemas e Tecnologias Subaquáticas.

LTI Linear Time Invariant.

MIMO Multiple Input Multiple Output.

NED North-East-Down reference frame.

NTNU Norwegian University of Science and Technology.

P Position model.

PF Particle Filter.

PMU Phase difference Measurement Unit.

PV Position-Velocity model.

PVA Position-Velocity-Acceleration model.

PVT Position/Velocity/Time.

PWM Pulse-Width Modulation.

RC Radio Controller.

RF Radio Frequency.

RTB Ranging Toolbox provided by Atmel for REB233SMAD-EK Kit.

RTK Real Time Kinematics.

SBP Swift Nav Binary Protocol.

SITL Software in the Loop.

UART Universal Asynchronous Receiver/Transmitter.

UAV Unmanned Aerial Vehicle.

UAV-Lab Unmanned Aerial Vehicles Laboratory.

UKF Unscented Kalman Filter.

USB Universal Serial Bus.

WP Waypoint.

ZV Zero Vibration Shaper.

ZVD Zero Vibration and Derivative Shaper.

Chapter 1

Introduction

1.1 Background and Motivation

An unmanned aerial vehicle (UAV) is an aircraft without a human pilot on board. It can be controlled either by the remote control of a pilot on the ground or autonomously by onboard computer. An UAV consisting of multiple fixed-pitch propellers attached to a simple mechanical construction is often referred to as multirotor system or simply multirotors.

The popularity of multirotors have risen in the last years due to technological advances. Advanced chip fabrication have made it possible to put accelerometer and gyroscopes directly on silicon chips, thus reducing the cost and size of the inertial sensing platform radically. Moreover, low-cost, power efficient, miniature and advanced microcontrollers that is capable of high speed data generation and acquisition have made it possible to embed many sensors in a single system. In addition, improved lithium-polymer battery technology have improved the flight time remarkably.

Multirotors can be used to carry vital payloads, such as sensors, in routine civil and military operations, where utilization of ground based equipment is impractical or impossible. More importantly, multirotors are particularly suited for emergency and rescue operations, where the environment is hazardous.

Pick-up and placement applications are delicate operations, where high precision positioning systems are required. Therefore, traditional navigation based on Global Navigation Satellite System (GNSS) may not be suitable. The carrier phase differential Global Positioning System (GPS), known as Real Time Kinematics (RTK) navigation, can deliver the needed accuracy at a centimetre level (Svendsen 2001). However, these solutions are traditionally expensive, and the reliability of the low-cost alternatives has so far proven to be poor (Takasu and Yasuda 2004; Røli 2015).

The robustness of carrier phase GPS may be reduced in extreme environments, eg., some marine, arctic, or urban applications. More importantly, these solutions do not work for indoor applications due to lack of satellite coverage. Therefore a low-cost, light-weight and easy-to-deploy high precision local positioning system that works regardless of its environments and satellite coverage is desirable.

In addition, the load attached to a multirotor is often suspended and has a pendulum like motion. The pendulous motion, if left uncontrolled, can damage the load or cause damage to its environments. The induced external forces due to the oscillation of the suspended load can make the multirotor system unstable or even uncontrollable. Consequently, these problems slow down or even prevent accurate pick-up and placement operations. Therefore, a control system able to account for these effects is necessary.

The work performed in this thesis is a part of the cooperative object lifting and high precision indoor navigation for UAV systems research and efforts at the Department of Engineering Cybernetics, NTNU. It is foreseen that some of the fundamentals established in the present work will be utilized in future efforts at department.

1.2 Previous Work

1.2.1 Indoor Navigation

Conventional GNSS technology relies on the line of sight of satellites. Consequently, it is not suited nor intended for indoor applications. Fortunately, the field of indoor navigation is developing rapidly with significant contributions in the recent years. Alternative indoor navigation methods have been developed, such as vision based high precision navigation systems. However, the cost, limited scalability, and inconvenient set-up and transport are significant practical challenges (Engedy and Horváth 2009; Martinez, Jimenez-Gonzalez, and Ollero 2011; Biswas and Veloso 2012).

On the other hand, beacon positioning systems for indoor applications have been developed and implemented, such as RF beacon (Pelka, Bollmeyer, and Hellbr 2014), novel radio interferometric positioning system (Dil and Havinga 2011), IR-photonic beacon (Chae et al. 2005) and cricket beacon localization systems (Zhou, Feng, and Yuan 2009). The popularity of such solutions have increased due to their low-cost, easy implementation and reliability.

Beside the different hardware implementations, the beacon positioning systems rely on similar algorithms. There are numerous publications comparing and discussing these alogorithms. In (Li et al. 2011) centroid algorithm, trilateration algorithm, and maximum likelihood estimation as a localization algorithm is compared, while Extended Kalman Filter (EKF) algorithm is used to improve the results. (St-pierre and Gingras 2004; Cotugno et al. 2013) compares the Unscented Kalman Filter (UKF), Particle Filter (PF) and EKF for the position estimation of a localization system, while (Gustafsson and Hendeby 2012) discusses their relation.

However, common for the above algorithms is that information about the noise characteristics is readily available. Measurements performed with low-cost sensors tend to be sensitive to their environments, thus varying noise characteristics. This challenge is attempted to be addressed in (Sasiadek and Wang 1999) by modifying the noise characteristics in a EKF with Fuzzy Logic Adaptive System (FLAS) achieving better positioning performance in a 3-D environment.

1.2.2 Suspended Load Control

The study of helicopter dynamics with suspended load received great attention in the late 1960s. The reason was increased interest in external load operations in the Vietnam war and the heavy-lift helicopter project [1]. However, the main attention at the time was directed towards the effect of a suspended load on the helicopter dynamics (Cicolani and Kanning 1992) and control of the helicopter. Therefore, other relevant topics, such as the swing damping suspended load control, are still rare in the literature. The interest in the subject has been renewed with new technological possibilities, such as small UAVs for landmine detection (Bisgaard 2007) and cooperative object lifting (Klausen, Fossen, and Johansen 2014).

There are several techniques to deal with the suspended load problem. For instance, (Ivler, Tischler, and Powell 2011) suggests a feedback controller utilizing the load angle from the suspended load motions in hover/low-speed operations, while (Bisgaard 2007; Omar 2012) proposes a delayed feedback controller to damp oscillations on the load.

Others have adopted approaches from similar engineering problems. (Singh and Singhose 2002) suggests a time delayed open-loop control technique for minimal residual vibrations as in gantry cranes. (Potter, Singhose, and Costello 2011) proposes a helicopter suspended load system which is based on an adaptive input shaping filter that yields minimal residual vibrations in three-dimensional overhead cranes published by (Stergiopoulos and Tzes 2007).

A more recent contribution is optimal trajectory generation that minimizes the swinging suggested in (Palunko, Fierro, and Cruz 2012). However, it should be noted that all of the above approaches are demanding with respect to actuation and navigation, i.e. the vehicles ability to follow reference trajectories and monitoring of the position and the orientation of the vehicles.

1.3 Contribution and Scope

The overall goal of this thesis is to lay the foundation of a complete multirotor system capable of solving tasks with suspended loads. The main focus is implementing and testing feasibility of a low-cost, high precision indoor navigation subsystem and a control system capable of handling suspended loads.

The proposed navigation system solves the scalability limitation of indoor positioning, with easy set-up and long range. Outdoor applications are not excluded since the positioning data can be fused together with low-cost local navigation solutions, e.g. an RTK system. The suggested control system is independent of the multirotor autopilot implementation, thus increasing modularity. Consequently, the control system can be easily implemented to any system with suspended loads, with the assumption of sufficiently small and light load. The presented system, with additional improvements in the future, may be utilized for solving cooperative lifting tasks as presented in (Klausen, Fossen, and Johansen 2014).

[1] The heavy-lift helicopter project was designed for the United States Army by Boeing Rotorcraft Systems (formerly Boeing Vertol).

The navigation system will be a sensor fusion approach to filtering, using all relevant onboard sensory data, along with range measurements from radio transceivers based on the principle of trilateration. The Extended Kalman Filter (EKF) will be used to solve the nonlinear localization problem. The algorithm will include outlier detection and rejection methods, and the divergence problem will be addressed with adaptive weighting of the covariance based on fuzzy logic.

The control system for the suspended load problem will focus on two techniques; input shaping filter and delayed feedback controller. The former is an open-loop zero-vibration command generation technique that reduces initial swing excitation, while the latter is a time-delayed feedback of the load angle. The output of the time-delayed feedback is the displacement of the load, which will be added to the longitudinal and lateral directions of the multirotor trajectory. Consequently, both controllers will be independent of the multirotor autopilot design and should not require any modification on the autopilot software.

1.3.1 Objectives

The partial objectives of this thesis can be summarized by the following points,

Multirotor Platform
Identify the necessary software and hardware to build a multirotor platform suitable for cooperative- and suspended load control.

Indoor Navigation System
Develop a navigation system for indoor positioning based on radio transceivers. The positioning system should be low-cost, light-weight, easy to deploy. The system should have sub-meter accuracy.

Suspended Load Control
Design a swing reducing control scheme for a multirotor suspended load problems. The control scheme should have high modularity, i.e., be independent of the multirotor autopilot system. The robustness and performance of the proposed controller will be investigated.

1.3.2 Individual Contributions

The development of the multirotor platform in Chapter 3, as well as the execution of all outdoor flight experiments, was done in collaboration with fellow master student Jon-Håkon Bøe Roli.

The payload design and implementation of the Piksi navigation system was done by fellow master student Jon-Håkon Bøe Roli.

The implementation of high precision indoor positioning system based on OptiTrack was done in collaboration with fellow master student Andreas Lindahl Flåten.

The author has assembled all the necessary hardware together, including soldering of the components and installed and developed the necessary software tools. An indoor navigation

system based on range measurements from radio transceivers and suspended load control was developed. The necessary hardware and software was integrated into the existing DUNE platform. All necessary software and developed applications can be found in the git respiratory of the Unmanned Aerial Vehicles Laboratory, NTNU [2]

1.4 Organization of This Thesis

Chapter 2 provides some important background theory for this thesis. First, the linear and the nonlinear Kalman filter with outlier detection and fuzzy logic adaptive system are presented. Next, the theory of fuzzy logic is explained, followed by reference model for trajectory tracking. Lastly, the principle of the zero-vibrational command generation, convolution in discrete-time domain and design of a delayed feedback controller are presented.

The multicopter platform and its interaction with the navigation subsystem are presented in Chapter 3. This involves the different hardware components and introduction to the necessary software tools.

Chapter 4 introduces the indoor navigation system. The provided ranging application and integration with the rest of the multicopter system is discussed first, followed by experiments with ranging measurements. Next, a solution to the localization problem is presented, using an EKF with adaptive covariance weithting based on Fuzzy Logic Adaptive Controller (FLAC). Results from simulations, indoor- and outdoor in-flight experiments are presented and discussed. The solutions include outlier detection and rejection algorithms, sensor fusion techniques, Inertial Measurement Unit (IMU) integration and Fuzzy Logic Adaptive Controller (FLAC) to weight covariances.

The suspended load control is presented in Chapter 5. First, the suspended load problem is introduced, followed by a control design for the multirotor system. Next, both feedforward and feedback swing damping controllers are presented. The results from MATLAB simulations, Software in the Loop (SITL) simulation, Hardware in the Loop (HITL) experiments and in-flight experiments are presented and discussed.

Chapter 6 summarizes the results and presents a general conclusion of the thesis. A short discussion of future work is also given.

Appendix A provides more detailed specifications of the main hardware components for the multicopter system.

[2] http://uavlab.itk.ntnu.no/

Chapter 2

Background Theory

This chapter presents the necessary background theory used in this thesis.

2.1 Kalman Filter

The Kalman Filter (KF) was first published by Rudolf E. Kalman in the 60s (Kalman 1960). The Kalman Filter algorithm estimates the unknown states in a linear state space model using noisy measurements over time. The KF is optimal, meaning that the estimates are unbiased and have minimum covariance. The optimal linear KF and the nonlinear variant known as the Extended Kalman Filter (EKF) are introduced below. Furthermore, on-line outlier detection and rejection methods, and a Fuzzy Logic Adaptive System (FLAS) to remedy the divergence issues in EKF will be presented.

2.1.1 The Linear Kalman filter

The KF estimates the states in a linear state space model by,

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_{u,k}\mathbf{u}_k + \mathbf{G}_{v,k} v_k \tag{2.1}$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{D}_k \mathbf{u}_k + e_k \tag{2.2}$$

where k denotes the measurement index, $\mathbf{F}_k$ is the transition matrix, $\mathbf{x}_k$ is the state vector, $\mathbf{G}_{u,k}$ is the model input matrix, $\mathbf{u}_k$ is the input, $\mathbf{G}_{v,k}$ is the uncertainty matrix, v_k is the process uncertainties, $\mathbf{y}_k$ is the output vector, $\mathbf{H}_k$ is the measurement matrix, $\mathbf{D}_k$ is the measurement input matrix and e is the measurement uncertainties. The initial state and covariance is given as,

$$E[\mathbf{x}_0] = \hat{\mathbf{x}}_{1|0} \tag{2.3}$$
$$\text{Cov}(\mathbf{x}_0) = \mathbf{P}_{1|0} \tag{2.4}$$
$$\text{Cov}(v_k) = \mathbf{Q}_k \tag{2.5}$$
$$\text{Cov}(e_k) = \mathbf{R}_k \tag{2.6}$$

where the double time index $k|m$ means *at time k given measurements up to time m*, $\mathbf{x}_0$ is the initial state vector, $\hat{\mathbf{x}}_{1|0}$ is the estimated state vector, $\mathbf{P}_k$ is the initial error covariance matrix, $\mathbf{Q}_k$ and $\mathbf{R}_k$ are the design matrices corresponding to model and measurement uncertainties, respectively (Verhaegen and Verdult 2007). According to (Gustafsson 2010), the initial state and the initial covariance may, in the absence of observations, be expressed as

$$\hat{\mathbf{x}}_{k|0} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|0} \tag{2.7}$$
$$\mathbf{P}_{k|0} = \mathbf{F}_k \mathbf{P}_{k-1|0} \mathbf{F}_k^T + \mathbf{G}_{v,k} \mathbf{Q}_k \mathbf{G}_{v,k} k^T \tag{2.8}$$

The Kalman Filter finds the best linear filter, utilizing the observation y_k as input,

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{M}_k \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k \mathbf{y}_k \tag{2.9}$$
$$\mathbf{P}_{k+1|k} = \mathbf{M}_k \mathbf{P}_{k|k-1} \mathbf{M}_k^T + \mathbf{L}_k \mathbf{R}_k \mathbf{L}_k^T \tag{2.10}$$

where $\mathbf{M}_k$ and $\mathbf{L}_k$ are optimized such that the estimates are unbiased and has minimum variance. Furthermore, the KF is the optimal linear (or nonlinear) state estimator and is asymptotically stable if the following assumptions are fulfilled (Kalman 1960):

- The process noise is white and Gaussian
- The measurement noise is white and Gaussian
- The initial state is Gaussian
- The system is linear
- The system is observable

For in-depth knowledge about the optimization procedure in KF, the reader is recommended to consult (Kalman 1960), (Verhaegen and Verdult 2007) or (Gustafsson 2010).

The linear discrete KF algorithm is summarized in (Gustafsson 2010) with the algorithm shown in Table 2.1,

where the best initialization is obtained if $\hat{\mathbf{x}}_{\mathbf{1|0}} = E(\mathbf{x}_0)$ and $\mathbf{P}_{1|0} = Cov(\mathbf{x}_0)$. Moreover, the design matrices are $\mathbf{Q}_k = \mathbf{Q}_k^T > 0$ and $\mathbf{R}_k = \mathbf{R}_k^T > 0$.

Table 2.1: Discrete Kalman Filter Algorithm.

Discrete Kalman Filter Algorithm	
1. Measurement update.	
The innovation covariance:	$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k\|k-1} \mathbf{H}_k^T + \mathbf{R}_k$
The Kalman gain:	$\mathbf{K}_k = \mathbf{P}_{k\|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$
State estimate update:	$\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k\|k-1} - \mathbf{D}_k \mathbf{u}_k)$
Error covariance update:	$\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k\|k-1}$
2. Step Update	
State estimation propagation	$\hat{\mathbf{x}}_{k+1\|k} = \mathbf{F}_k \hat{\mathbf{x}}_{k\|k} + \mathbf{G}_{u,k} \mathbf{u}_k$
Error covariance propagation	$\mathbf{P}_{k+1\|k} = \mathbf{F}_k \mathbf{P}_{k\|k} \mathbf{F}_k^T + \mathbf{G}_{v,k} \mathbf{Q}_k \mathbf{G}_{v,k}^T$

2.1.2 Extended Kalman Filter

While the KF estimates the states in a linear state space model, the EKF estimates the states in the non-linear case. In the Extended Kalman Filter, the non-linearities are expanded around the best current estimates. Similar to the linear KF, the recursion is split into a measurement and a time update, while the nonlinearities are dealt using Taylor expansion. Hence, the EKF does not have the optimality properties of linear KF. In addition, there is a risk that the filter can diverge if the point of linearization deviate too far from the true state. However, it is used extensively and proven to work exceptionally. Given a nonlinear function $z = g(x)$, the Taylor expansion around the best estimate $\hat{x}$ is (Walpole et al. 2011):

$$z = g(x) = g(\hat{x}) + g'(\hat{x})(x - \hat{x}) + O(2) \tag{2.11}$$

where g' denotes the *Jacobian*, while the *Hessian* is neglected since the EKF is based on the first two terms (Gustafsson 2010), and $O(2)$ denotes the rest terms. Note that this calculation works well as long as the rest terms are small. In other words, it is assumed low degree of nonlinearity and low signal-to-noise ratio. Hence, the estimation error is considered small, eliminating the divergence problem.

The EKF algorithm used in this thesis is based on the Discrete-time Algebraic Ricatti Equation (DARE) for the state covariance given in (Gustafsson 2010). The algorithm is shown in the table Table 2.2. However, it is important to note that there exist several EKF algorithms. For instance, in (Fossen 2011) the Jordan form of the error covariance matrix is used. Hence

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{h}'(\hat{\mathbf{x}}_{k|k-1})) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{h}'(\hat{\mathbf{x}}_{k|k-1}))^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \tag{2.12}$$

where $\mathbf{h}'$ denotes the Jacobian matrix of the measurement matrix $\mathbf{H}$, and where (2.12) yields a positive semi-definite matrix and thus avoids yet another computationally expensive matrix inversion. In practice however, the symmetry cannot be used and the computational power is not a problem today. In addition, the (Gustafsson 2010) algorithm works noticeably better in cases where $\mathbf{S}_k$ is close to singular. The practical advantages/disadvantages of the algorithm will be discussed more in Section 4.4.2.

Table 2.2: Discrete-Time Extended Kalman Filter (EKF) algorithm.

Discrete-Time extended Kalman Filter	
1. Measurement update.	
The innovation covariance:	$\mathbf{S}_k = \mathbf{h}'(\hat{\mathbf{x}}_{k\|k-1})\mathbf{P}_{k\|k-1}(\mathbf{h}'(\hat{\mathbf{x}}_{k\|k-1}))^T + \mathbf{R}_k$
The Kalman gain:	$\mathbf{K}_k = \mathbf{P}_{k\|k-1}(\mathbf{h}'(\hat{\mathbf{x}}_{k\|k-1}))^T\mathbf{S}_k^{-1}$
State estimate update:	$\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k\|k-1}))$
Error covariance update:	$\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{P}_{k\|k-1}(\mathbf{h}'(\hat{\mathbf{x}}_{k\|k-1}))^T\mathbf{S}_k^{-1}\mathbf{h}'(\hat{\mathbf{x}}_{k\|k-1})\mathbf{P}_{k\|k-1}$
2. Step Update	
State estimation propagation	$\hat{\mathbf{x}}_{k+1\|k} = \mathbf{f}(\hat{\mathbf{x}}_{\mathbf{k}\|\mathbf{k}})$
Error covariance propagation	$\mathbf{P}_{k+1\|k} = \mathbf{Q}_k + \mathbf{f}'(\hat{\mathbf{x}}_{\mathbf{k}\|\mathbf{k}})\mathbf{P}_{k\|k}(\mathbf{f}'(\hat{\mathbf{x}}_{\mathbf{k}\|\mathbf{k}}))^T$

2.1.3 Outlier Detection and Rejection

There are several ways to detect outliers in a measurement set. The detection and rejection of outliers are important since it will boost the performance of the KF. When an outlier is detected, it should be discarded from the measurement update. Note that the rejection of a measurement equals to setting the measurement uncertainty to very high (or infinity) for the corresponding measurement at the corresponding sample. Also note that very high R in return equals to very small K for the corresponding measurement at the corresponding sample. Therefore, to reduce the computational complexity, the corresponding column in K is set to zero when an outlier is detected.

The most simple and intuitive way to detect an outlier is to use the white and Gaussian property of the measurement noise. It is then natural to base the implementation on normalized residual. However, this requires that all the measurements are inliers and Gaussian distributed in the linear case. Since the implementation will be based on the nonlinear case, the central limit theorem can be used to approximate a Gaussian distribution (Gustafsson 2010). Consequently, the standard hypothesis test can be used to reject the null hypothesis of outliers. For an iterated Kalman filter, we reject a measurement as an outlier if:

$$T(y_k^i) > \chi^2_{\alpha,n_y^i} \tag{2.13}$$

where k is the sample number, i is the corresponding measurement, χ^2 is the chi-square distribution with corresponding confidence α and n_y^i degrees of freedom. The iterated test statistic is based on:

$$T(y_k^i) = (y_k^i - H_k^i\hat{x}_{k|k-1})^T(H_k^i P_{k|k-1}(H_k^i)^T + R_k^i)^{-1}(y_k^i - H_k^i\hat{x}_{k|k-1}) \sim \chi^2_{n_y^i} \tag{2.14}$$

where y_k^i represents the measurement from sensor i at iteration k. However, the quality of this method depends on correct information about the measurement noise. Divergence will occur if $\mathbf{R}_k$ is erroneous, since the outlier detection algorithm will start to reject too many measurements.

Assuming white and Gaussian measurement noise, several other outlier detection methods can be used. An even simpler method than the above mentioned, is based on checking the signal quality of the measurement before feeding it to the Kalman Filter. To do so, Grubbs

test (Grubbs 1950) or Z-scores can be used. The Z-score of a normal distribution is defined as (Walpole et al. 2011),

$$Z_i = \frac{Y_i - \hat{Y}}{s} \tag{2.15}$$

where $\hat{Y}$ denotes the sample mean and s denotes the sample standard deviation. However, such methods are not optimal for on-line outlier detection in dynamic targets. The method is based on mean value distances which may not be a good way of determining outliers for high speed operations.

Additionally, assuming high measurement frequency and slow movement of the tracked object, a median filter can be used to determine whether or not the measured data is an outlier. By comparing the measured data with the median of the last 3 samples for example, one can determine whether or not the measured data is outlier. One could also compare the mean value of the last 3 measurement with the current measurement. However, in the case of an outlier, this method will flag the some of the next measurements as outliers as well, since the outlier will affect the mean. On the other hand, a simple solution to that problem is to use the last three *approved* measurements.

The standard hypothesis test for outlier detection is implemented and discussed in Section 4.3.6.

2.1.4 Divergence Prevention - Weighted Extended Kalman Filter

The KF is an optimal estimation based on an internal model of the system dynamics and characterized by recursive evaluation (Schlee, Standish, and Toda 1967). Consequently it is dependent on known plant dynamics and zero mean white noise (Sasiadek and Wang 1999). If the model dynamics and the actual dynamics do not agree, divergence problems occur.

There is two kind of divergence (Gelb et al. 2001),

- ***True divergence*** - estimated error covariance goes to infinity.
- ***Apparent divergence*** - estimated error covariance is bounded, but larger than the predicted error covariance.

Note that the divergence due to modelling error is critical in Kalman Filter since the filter will continuously try to fit a wrong process. The computed estimation error will become very small, which is unrealistic, and the filter will start disregarding the measurements. Consequently, the filter will start to diverge.

The divergence problem can be eliminated by either estimating unmodeled states or make sure that all states are sufficiently excited by process noise. The former adds complexity to the filter and cannot actually guarantee that the instability is in fact due to the estimated unmodeled states (Gelb et al. 2001). The latter however, is easy to implement and makes sure that the system is driven by white noise and prevents rejection of new measurements. In practice, this means that the tuning matrices $\mathbf{R}$ and $\mathbf{Q}$ needs to be weighted continuously (Sasiadek and Wang 1999). Consequently, a weighting parameter α can be added such

that,

$$\mathbf{R}_k = \mathbf{R}_k \alpha \tag{2.16}$$

$$\mathbf{Q}_k = \mathbf{Q}_k \alpha \tag{2.17}$$

and a Fuzzy Logic Adaptive Controller (FLAC) can be used to continually adjust the noise strength in the filter's internal model and tune the filter. The theory of fuzzy logic is presented in in Section 2.2, a Fuzzy Logic Adaptive System (FLAS) implementation using weighted EKF is presented in Section 2.1.5 and the performance of the FLAC is discussed in Chapter 4.

2.1.5 Fuzzy Logic Adaptive System

In summary, there are two critical assumptions about process and measurement noise in EKF to work optimally:

- They are zero-mean white noise.
- The covariance Q and R are known.

Consequently, the residual of a perfectly tuned model should be a zero-mean white noise process. Remember that the residual is the term $(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}))$ in state estimation update in Table 2.2. If the residual is not zero-mean white noise, the filter is not performing optimally (Sasiadek and Wang 1999).

In practise however, the model is not perfectly tuned and the process noise is not always zero-mean. In those cases the residuals can be used to adapt the filter. Note that the residuals are in fact the difference between the actual measurements and the best measurement predictions based on the filters internal model. Thus, not zero-mean residual will actually lead to divergence of the filter estimations.

Furthermore, the covariance of the residuals are related to R and given by

$$\mathbf{P}_{res} = \mathbf{H}_k \mathbf{P}_{k-1} \mathbf{H_k}^T + \mathbf{R} \tag{2.18}$$

Generally when the covariance is becoming large and the residual is moving away from zero, the EKF is becoming unstable (Sasiadek and Wang 1999). Therefore, it would be natural to use the residual and covariance of the residuals as inputs to FLAS to decide the degree of divergence. In (Sasiadek and Wang 1999) the authors present a nine rule system such as:

1. ***If*** *the covariance of residuals is large and the mean values are zero* ***Then*** α *is large.*
2. ***If*** *the covariance of residuals is zero and the mean values are large* ***Then*** α *is zero.*

and so on. In practise however, analysis of the error covariance can be tricky. Nonetheless, a good sign of instability or model-measurement mismatch is a rapid change in the error covariance. Therefore, for simplicity, we can reduce our FLAC to these 3 rules:

1. ***If*** *the mean values of the residuals are small* ***Then*** α *is small.*
2. ***If*** *the mean values of the residuals are medium* ***Then*** α *is medium.*

3. ***If*** *the mean values of the residuals are large* ***Then*** α *is large.*

and a final check in cascade if the error covariance has changed radically since last time. If so, we excite the system even more.

Using absolute values of the residuals, the tentative mean value membership functions are shown in Figure 2.1. Keep in mind that this is a tuning method, and one may tune the parameters to optimize the results. This method is implemented and discussed in detail in Section 4.3.7

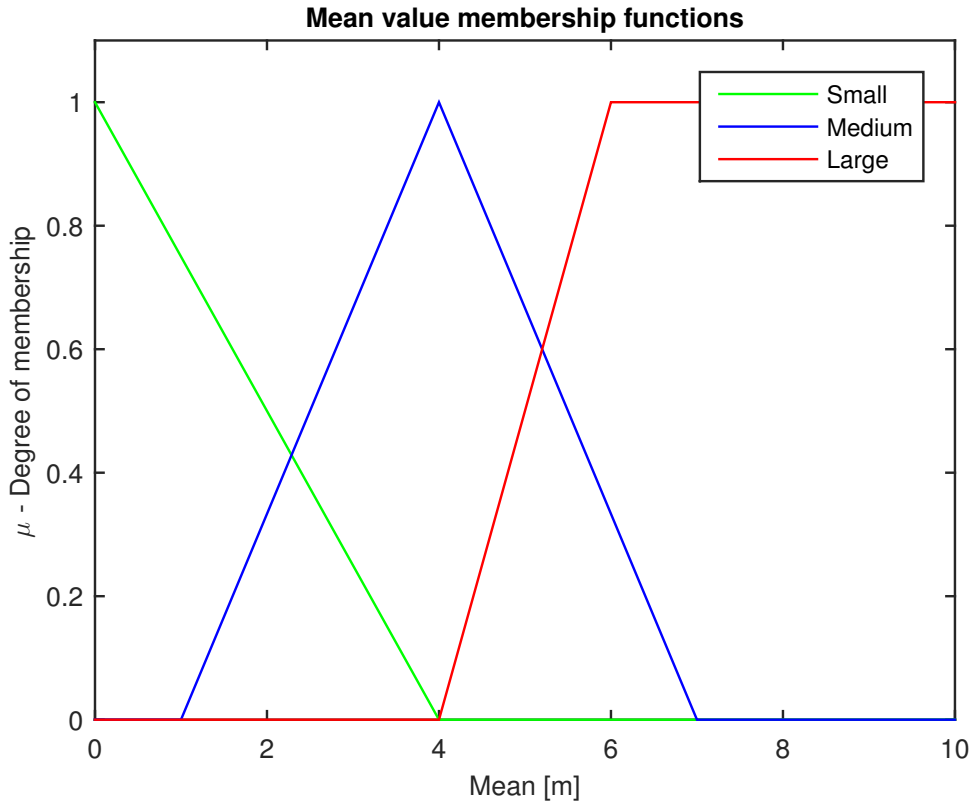


Figure 2.1: Mean value membership functions.

2.2 Fuzzy Logic

Fuzzy Logic Adaptive System (FLAS) discussed in Section 2.1.5 is based on fuzzy logic, a system introduced by Lotfi A. Zadeh in the 1960s. Fuzzy logic is simply a superset of Boolean logic that has been expanded to handle *grades* of truth, or *membership* ranging between zero and one. Using the notions of inclusion, union, intersection, complement, relation, convexity etc. a discrete logic can be expressed as fuzzy logic (Zadeh 1965). In this section a brief introduction to main ideas of basic fuzzy logic will be given.

2.2.1 Linguistic Variables, Membership Functions and Fuzzy Graphs

The term *fuzzy* refers to concepts where the solution cannot be expressed as "true" or "false", but rather *partially* true or false. In other words it can be see as formalization of approximate reasoning, introducing human concepts into machine learning. The most central concept of fuzzy applications is the linguistic variable (Zadeh 1975b).

For instance, Zadeh describes *age* as a linguistic variable, whose linguistic values are *young, middle-aged* and *old* (Zadeh 1973). Consider now *young* as our cognitive category. We need to assign each person in the world with fuzzy subset *young*. Therefore a membership function based on the person's age is constructed.

$$young(x) = \begin{cases} 1.0 & \text{if age}(x) < 20 \\ 1.0 - \frac{\text{age}(x)-20}{20} & \text{if } 20 \leq \text{age}(x) \leq 40 \\ 0 & \text{if age}(x) > 40 \end{cases}$$

Figure 2.2 shows the difference between the linguistic and numerical values of *young*. Note that rather than saying young or old, we can now interpret it as degree of truth, e.g. if James is 24 years old, we can say "James is young" = 0.8. Hence a more vivid explanations or rules can be created.

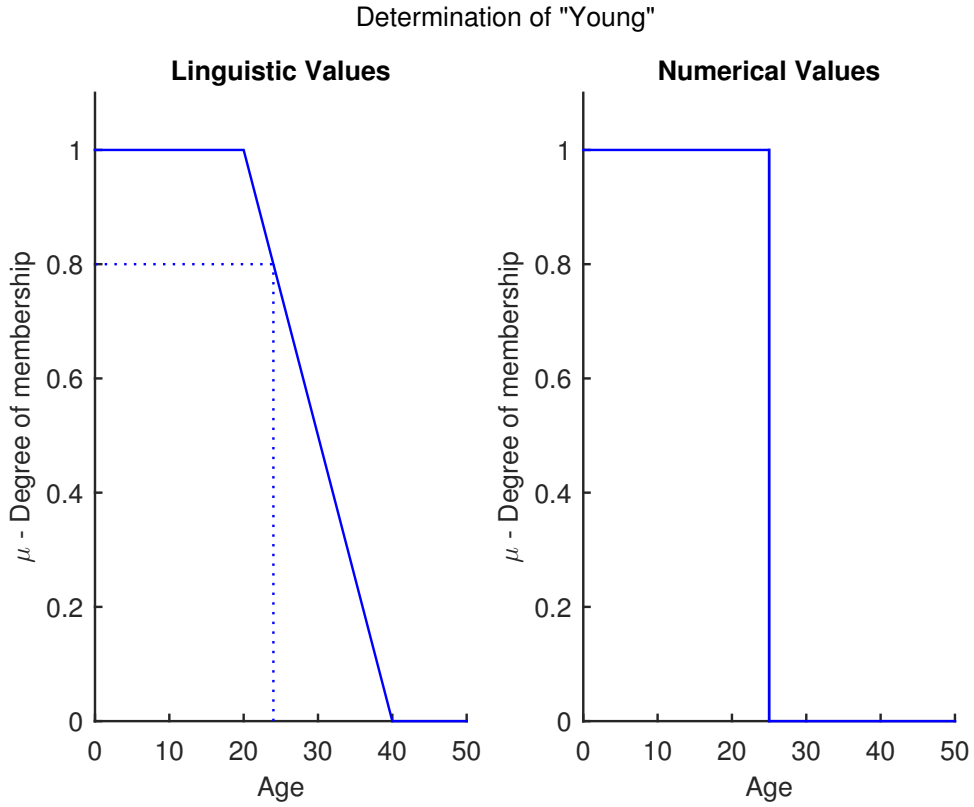


Figure 2.2: Difference between linguistic and numerical values of *young* in fuzzy logic. 24 Years is 80% young in Linguistic values, while numerical values would only say young or not.

2.2.2 Fuzzy Logic Rules

The concept of linguistic variables are just a point of departure for the concept of fuzzy logic. It simply results in a data compression using membership functions, fuzzy graphs and rules (Zadeh 1994). After all, the same procedure could in practise be described using a lot of *if-then* rules. However, it would not be practical.

In addition, more complex logical systems can be constructed using fuzzy logic. Imagine now, in addition to age, we also want to know the tallness of a person. How do we interpret the degree of truth, or the membership function of a person being young (statement **X**) *and* tall (statement **Y**), or young *or* tall, or young *not* tall?

Zadeh himself wanted that the process of *fuzzification* to be regarded as a methodology to generalize *any* specific discrete case to continuous (fuzzy) form, rather than regarding it as a single theory. Consequently, a lot of extension principles have been introduced over time, such as *fuzzy calculus* or *fuzzy differential equations* etc. (Zadeh 1975a).

However, the standard definitions of fuzzy logic will be used in this thesis. That is

$$\text{truth (not } \mathbf{X}) = 1.0 - \text{truth}(\mathbf{X}) \tag{2.19}$$

$$\text{truth}(\mathbf{X} \text{ AND } \mathbf{Y}) = \min(\text{truth}(\mathbf{X}), \text{truth}(\mathbf{Y})) \tag{2.20}$$

$$\text{truth}(\mathbf{X} \text{ OR } \mathbf{Y}) = \max(\text{truth}(\mathbf{X}), \text{truth}(\mathbf{Y})) \tag{2.21}$$

For more in-depth description the reader is recommended to consult (Zadeh 1994).

2.3 Reference Model for Trajectory Tracking

Many guidance systems are set to track a smooth signal of desired position, speed and/or acceleration. Therefore, a smooth time-varying trajectory $\mathbf{r}_d(t)$ is useful in many applications. Especially since the speed and acceleration can simply be obtained from time-differentiation of $\mathbf{r}_d(t)$ or $\dot{\mathbf{r}}_d(t)$. The subscript d denotes desired value. This section will discuss how such reference models can be designed.

In the literature, smooth reference models are usually generated based on physically motivated models (Fossen 2011). In this thesis, a *mass-damper-system* is used to generate state trajectories. However, it is important to be aware that the there might be physical limitations on speed and acceleration of the target vehicle. Therefore, speed and acceleration might need to be saturated before feeding it as a control input. Thus, there is a compromise between the performance and accuracy of the tracking. However, the problem is basically a tuning problem.

(Fossen 2011) utilizes the Low Pass (LP) filter structure when generating the state trajectories for a craft based on a mass-damper-system,

$$h_{lp} = \frac{\omega_{n_i}^2}{s^2 + 2\zeta_i \omega_{n_i} s + \omega_{n_i}^2} \tag{2.22}$$

where ζ_i are the relative damping ratios and ω_{n_i} is the natural frequencies. However, the same techniques can be used in aerial vehicles since the systems DoF are often the same. That being said, only translational dynamics will be regarded in this thesis. Thus, the MIMO mass-damper-system can be expressed as,

$$\mathbf{M}_d\ddot{\mathbf{p}}_d + \mathbf{D}_d\dot{\mathbf{p}}_d + \mathbf{G}_d\mathbf{p}_d = \mathbf{G}_d\mathbf{r} \tag{2.23}$$

where $\mathbf{M}_d$, $\mathbf{D}_d$ and $\mathbf{G}_d$ are positive design matrices and $\mathbf{p}_d = [x_N y_E z_D]^T$ is the desired NED position. Thus, the system can be expressed in Linear Time Invariant (LTI) state space form (Klausen, Fossen, and Johansen 2014),

$$\dot{\boldsymbol{\xi}}_d = \mathbf{A}_d\boldsymbol{\xi}_d + \mathbf{B}_d\mathbf{r} \tag{2.24}$$

where $\boldsymbol{\xi}_d := [\mathbf{p}_d^T, \dot{\mathbf{p}}_d^T]^T$ is the desired state vector, and $\mathbf{r}$ is a bounded reference vector. The state and input matrices are recognized as (Fossen 2011),

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}_d^{-1}\mathbf{G}_d & -\mathbf{M}_d^{-1}\mathbf{D}_d \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}_d^{-1}\mathbf{G}_d \end{bmatrix} \tag{2.25}$$

2.3.1 Velocity Reference Model - 2nd-Order Model

A velocity reference model can be constructed using Equation (2.22). Note that the Equation is in 2nd-order, which gives smooth desired velocity and acceleration (Fossen 2011). Thus, the 2nd-order system in vectorial form,

$$\ddot{\boldsymbol{\nu}}_d + 2\boldsymbol{\Delta}\boldsymbol{\Omega}\dot{\boldsymbol{\nu}} + \boldsymbol{\Omega}^2\boldsymbol{\nu}_d = \boldsymbol{\Omega}^2\mathbf{r} \tag{2.26}$$

where $\boldsymbol{\nu}_d$ is the desired velocity $\boldsymbol{\nu}_d = [u, v, w,]^T$, $\dot{\boldsymbol{\nu}}_d$ is the desired acceleration, $\ddot{\boldsymbol{\nu}}_d$ is the jerk, $\boldsymbol{\Delta} = diag\{\zeta_1, \zeta_2, ..., \zeta_n\}$, and $\boldsymbol{\Omega} = diag\{\omega_{n_1}, \omega_{n_2}, ..., \omega_{n_n}\}$. Thus, in LTI form as in Equation (2.24) with,

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\boldsymbol{\Omega}^2 & -2\boldsymbol{\Delta}\boldsymbol{\omega} \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Omega}^2 \end{bmatrix} \tag{2.27}$$

It is important to note that a step response in $\mathbf{r}$ will give a step response in desired acceleration $\ddot{\mathbf{v}}_d$. However, the speed and position will be Low Pass (LP) filtered, and thus smooth. Figure 2.3 shows step response of the 2nd-order model with $\omega = 0.7$ and $\zeta = 0.9$. Note the difference between the responses when there is a saturation on the velocity. The response is slower, but the steady state velocity is (Fossen 2011)

$$\lim_{t\to\infty} \mathbf{v}_d(t) = \mathbf{r}$$

However, the system can be tuned more accurately and the acceleration can be smoothed by utilizing a 3rd order reference model.

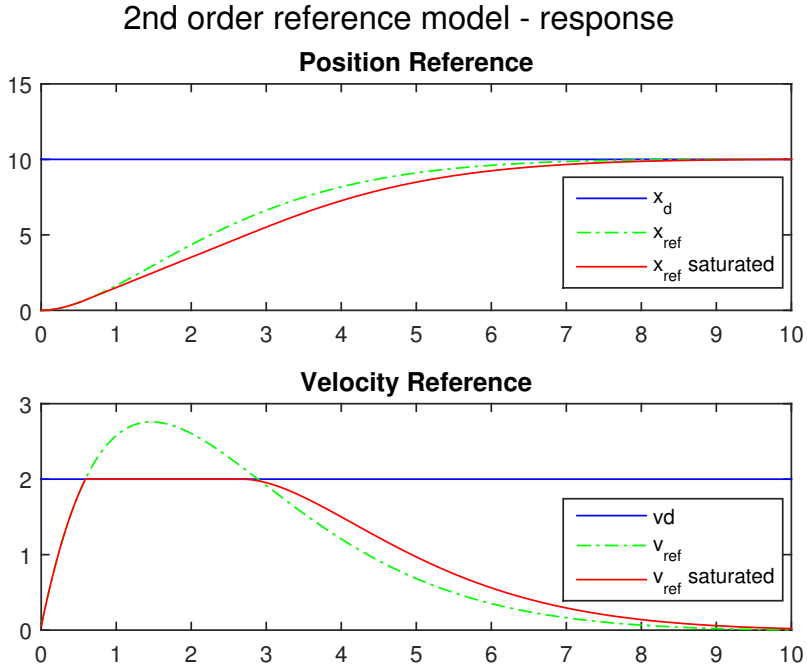


Figure 2.3: Response of 2nd-order reference model. The position and velocity is smooth. However, if there is a saturation on the speed, the response is slower.

2.3.2 3rd Order Reference Model

A 3rd order reference model will filter the steps in $\mathbf{r}$, thus smooth out the acceleration. Thus a first order LP filter can be cascaded with a mass-damper-spring system,

$$h_{lp} = \frac{\omega_{n_i}^2}{(1+T_i s)(s^2 + 2\zeta_i \omega_{n_i} s + \omega_{n_i}^2)} \tag{2.28}$$

where the first order LP filter with the time constant $T_i = 1/\omega_{n_i} > 0$ is added (Fossen 2011). Hence, the state space form for the reference model based on Equation (2.24),

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{\Omega}^3 & -(2\mathbf{\Delta}+\mathbf{I})\mathbf{\Omega}^2 & -(2\mathbf{\Delta}+\mathbf{I})\mathbf{\Omega} \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{\Omega}^3 \end{bmatrix} \tag{2.29}$$

where $\mathbf{\Delta} = diag\{\zeta_1, \zeta_2, ..., \zeta_n\}$, and $\mathbf{\Omega} = diag\{\omega_{n_1}, \omega_{n_2}, ..., \omega_{n_n}\}$.

2.4 Zero-Vibration Command Generation

The goal of zero-vibration command generation is to understand how a command can be generated such that it moves a system without vibration. The concept can simply be summed up by Figure 2.4. The impulse response of A_1 at t_0 can be cancelled out by an identical impulse, A_2, given at the right time t_1.

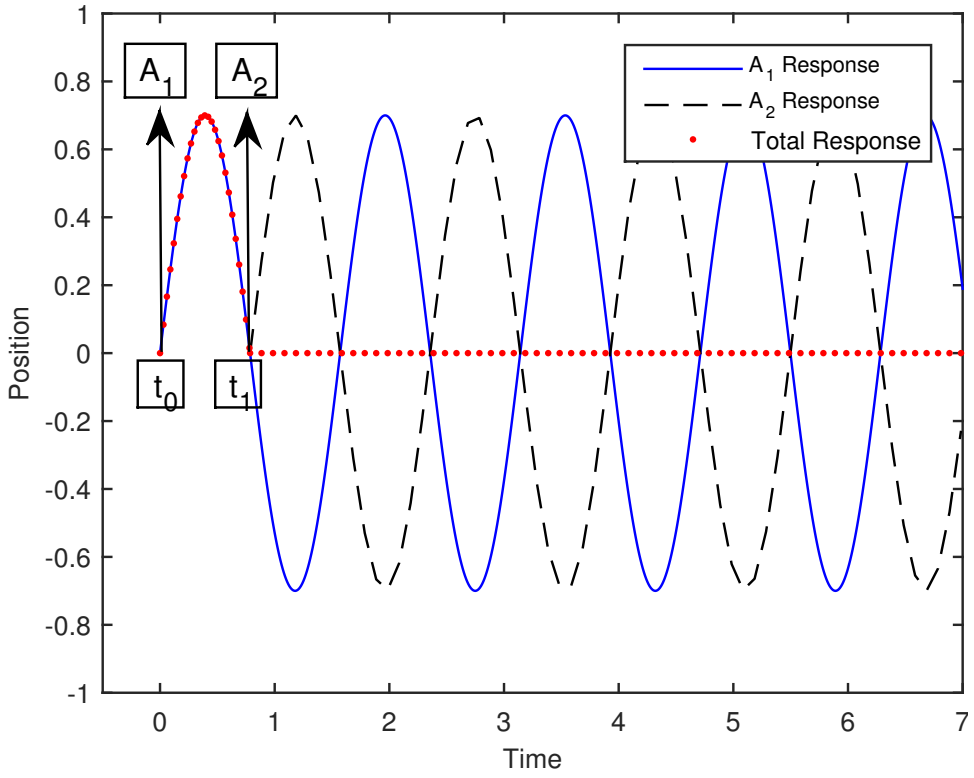


Figure 2.4: Concept of zero-vibration command generation. Signals A_1 and A_2 cancels each other, so the total response of the system is zero.

However, in reality, there is a physical damping on the load response, which makes the problem more complex. Consequently, the main objective is to derive the amplitudes (A_1, A_2) and the time locations (t_0, t_1) of the two impulses.

Assuming reasonable estimates of the systems natural frequency ω, and the damping ratio ζ, (Singh and Singhose 2002) describes the residual vibration that results from a sequence of impulses as:

$$V(\omega,\zeta) = e^{-\zeta\omega t_n}\sqrt{C(\omega,\zeta)^2 + S(\omega,\zeta)^2} \tag{2.30}$$

where,

$$C(\omega,\zeta) = \sum_{i=1}^{n} A_i e^{\zeta\omega t_i}\cos(\omega_d t_i) \tag{2.31}$$

$$S(\omega,\zeta) = \sum_{i=1}^{n} A_i e^{\zeta\omega t_i}\sin(\omega_d t_i) \tag{2.32}$$

where A_i are the amplitudes, t_i are the time location of the impulses, n is the number of impulses and $\omega_d = \omega\sqrt{1-\zeta^2}$. Furthermore, ω and ζ are defined as,

$$\omega = \sqrt{\frac{g}{\ell}}, \tag{2.33}$$

$$\zeta = \frac{p_d}{2\omega m_L} \tag{2.34}$$

where g is the gravity, ℓ is the rope length between the multirotor and the load, m_L is the load weight and p_d is damping factor due to external forces (mostly wind). In (Bisgaard 2007), Equation (2.30) is defined as a non-dimensional residual vibration term,

$$V = \frac{\text{Vibration amplitude caused by impulse sequence}}{\text{Vibration amplitude caused by unit impulse}} \tag{2.35}$$

Hence, the amplitudes and time location that lead to zero residual vibration can be found by setting Equation 2.30 to zero. Nevertheless, the following restrictions are necessary to avoid zero-valued or infinitely-valued impulses (Singh and Singhose 2002),

$$\sum A_i = 1 \tag{2.36}$$

$$A_i > 0, \quad i = 1, 2, ..., n \tag{2.37}$$

Consequently, in the case of 2 impulses, the four unknown parameters A_1, A_2, t_0 and t_1 can be found by solving (2.30), while also satisfying Equation (2.36) and (2.37). Both (Bisgaard 2007) and (Singh and Singhose 2002) presents the solution for the case of two impulses, hereafter called Zero Vibration Shaper (ZV), resulting in

$$A_1 = \frac{1}{1+K}, \quad A_2 = \frac{K}{1+K} \tag{2.38}$$

$$t_0 = 0, \quad t_1 = \frac{T_d}{2} \tag{2.39}$$

where T_d is the damped period of vibrations and,

$$K = exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right) \tag{2.40}$$

.

2.5 Convolution Sum In Discrete-Time Domain

The mathematical definition of the discrete time domain convolution is presented in (Proakis and Manolakis 1996),

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{2.41}$$

where $x[n]$ is the input signal, $h[n]$ is the impulse response, and $*$ denotes convolution. Note that the input term $x[n]$ is multiplied with the time-shifted terms of $h[n]$ and added up. However, to fully understand the idea of convolution, impulse response and impulse decompositions needs to be explained. Here, a brief summary will be given, assuming that the reader has general knowledge about digital signal processing.

Impulse Function Decomposition

A signal is often decomposed as a weighted sum of basis signals in electronics. Similar to Fourier series, where sum of sines and cosines are used to represent any periodic signal. In general, a signal is written as a sum of scaled and shifted delta functions (Proakis and Manolakis 1996),

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k] \tag{2.42}$$

where $\delta[n] = 1$ only at $n = 0$.

Impulse Response

Impulse response is the system output resulting from an impulse function as input. It is often denoted as $h[n]$,

$$h[n] \longrightarrow \boxed{System} \longrightarrow \delta[n]$$

For time invariant systems, time shifted impulse functions results time-shifted function,

$$h[n-k] \longrightarrow \boxed{System} \longrightarrow \delta[n-k]$$

For linear systems, a scaled input results in scaled response,

$$c \cdot h[n] \longrightarrow \boxed{System} \longrightarrow c \cdot \delta[n]$$

The Convolution Sum

Thus, combining the above mentioned properties, the convolution sum can be constructed,

$$\sum_{k=-\infty}^{\infty} c_k \cdot h[n-k] \longrightarrow \boxed{System} \longrightarrow \sum_{k=-\infty}^{\infty} c_k \cdot \delta[n-k] \tag{2.43}$$

and it is clear that if the input signal is $x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k]$, the output signal becomes $y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$.

In other words, the signal is decomposed into a set of impulses and the output signal is computed by adding the shifted and scaled impulse response. Note that it is sufficient to know about the impulse response, $h[n]$, to actually calculate how the system reacts for any input.

2.6 Delayed Feedback Controller Design

The swing reducing controller explained in this section is mainly based on (Bisgaard 2007). The idea is to absorb vibrations in an oscillating system by intentionally using a delayed feedback. A delayed feedback controller is in general not optimal, as it often leads to instability (Bisgaard 2007) . However, a typical feedback controller for the suspended load problem, discussed in Section 5.1, will amplify the induced swings. Therefore, the properties of a delayed feedback controller can be used to counteract the swing load motion simply by following the loads motion. The concept is explained in Figure 5.2.

Controller Design

Consider the second order linear system,

$$\ddot{x}(t) + 2\omega_n \zeta \dot{x}(t) + \omega_n^2 x(t) = u(t), \tag{2.44}$$

where ω_n is the natural frequency, ζ is the damping, x is the state and u is the input. Moreover, a proportional feedback of the delayed state can be added, thus

$$\ddot{x}(t) + 2\omega_n \zeta \dot{x}(t) + \omega_n^2 x(t) = G_d x(t - \tau_d), \tag{2.45}$$

where τ_d is the time delay and G_d is a design parameter. Taking it into Laplace domain results in,

$$x(s^2 + 2\omega_n \zeta s + \omega_n^2) = G_d e^{-\tau_d s} \tag{2.46}$$

Consequently, the design parameters (G_d, τ_d) can be derived such that the damping the system is maximized. More or less advanced solutions exist in the literature (Masoud and Nayfeh 2003). However, (Bisgaard 2007) suggests a simple approach, using the 2nd-order pade approximant,

$$e^{-\tau_d s} = \frac{1 - \tau_d/2s + \tau_d^2/12s^2}{1 + \tau_d/2s + \tau_d^2/12s^2} \tag{2.47}$$

Thus, the linear system theory can be applied to find appropriate design parameters. The state space form of the feedback controller can be modeled (Bisgaard 2007),

$$\mathbf{A}_c = \begin{bmatrix} \frac{-6}{\tau_d} & \frac{-12}{\tau_d^2} \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} G_d \\ 0 \end{bmatrix}, \quad \mathbf{C}_c = \begin{bmatrix} \frac{-12}{\tau_d} & 0 \end{bmatrix}, \quad \mathbf{D}_c = 1 \tag{2.48}$$

Delayed Feedback System

The goal of the controller derived in (2.48) is to dampen the pendulous modes of the suspended load within the limits of the multirotor performance. (Bisgaard 2007) introduces a way to find the controller parameters G_d and τ_d that achieves maximum damping while maintaining the multirotor dynamics.

Assuming a plant **P** and the delayed feedback controller **C** shown in (2.48), the positive closed loop system shown in Figure 2.5 can be described as:

$$\mathbf{H} = \frac{\mathbf{P}}{1 - \mathbf{PC}} \tag{2.49}$$

Consider now that the plant, which is the combined multirotor suspended load system, can be described by the linear state space system,

$$\mathbf{P} = \begin{cases} \dot{\mathbf{x}}_p = \mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\mathbf{u}_p \\ \mathbf{y}_p = \mathbf{C}_p\mathbf{x} + \mathbf{D}_p\mathbf{u}_p \end{cases}$$

where $\mathbf{x}_p$ is the state vector, $\mathbf{u}_p$ is the input vector, and $\mathbf{y}_p$ is the output vector. Thus, the combined dynamics of the system described in (2.49), can be derived in linear state space form(Bisgaard 2007),

$$\mathbf{H} = \begin{cases} \dot{\mathbf{x}}_h = \mathbf{A}_h\mathbf{x} + \mathbf{B}_h\mathbf{u}_h \\ \mathbf{y}_h = \mathbf{C}_h\mathbf{x}_h + \mathbf{D}_h\mathbf{u}_h \end{cases}$$

with

$$\mathbf{A}_h = \begin{bmatrix} \mathbf{A}_p & \mathbf{B}_p\mathbf{C}_c \\ \mathbf{B}_c\mathbf{C}_c & \mathbf{A}_c \end{bmatrix}, \quad \mathbf{x}_h = \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_c \end{bmatrix} \tag{2.50}$$

Consequently, the eigenvalues of the linear system of the combined dynamics are given by

$$(\mathbf{A}_h - \lambda\mathbf{I})\mathbf{x} = 0 \tag{2.51}$$

where $\boldsymbol{\lambda}$ is the vector of eigenvalues with corresponding damping vector $\boldsymbol{\zeta}$ in (2.46). Note that this system is

- Asymptotically stable if $\mathrm{Re}(\lambda_i) < 0$
- Marginally stable if $\mathrm{Re}(\lambda_i) \leq 0$ and there is no multiple eigenvalues with $\mathrm{Re}(\lambda_i) = 0$
- Unstable if $\mathrm{Re}(\lambda_i) > 0$

for all eigenvalues λ_i in $\mathbf{A}_h$ (Balchen, Andresen, and Foss 2003). Moreover, the controllability of the system is evident from the fact that the multirotor controller in the x,y and z direction is independent.

Thus the optimal controller design is found by solving the following optimization problem (Bisgaard 2007):

$$\underset{(G_d,\tau_d)}{\operatorname{argmax}} \quad \min_i \zeta_i \tag{2.52}$$

where the damping is defined as

$$\zeta_i = \frac{Re(\lambda_i)}{\omega_{ni}} \tag{2.53}$$

and where

$$\omega_{ni} = |\lambda_i| \tag{2.54}$$

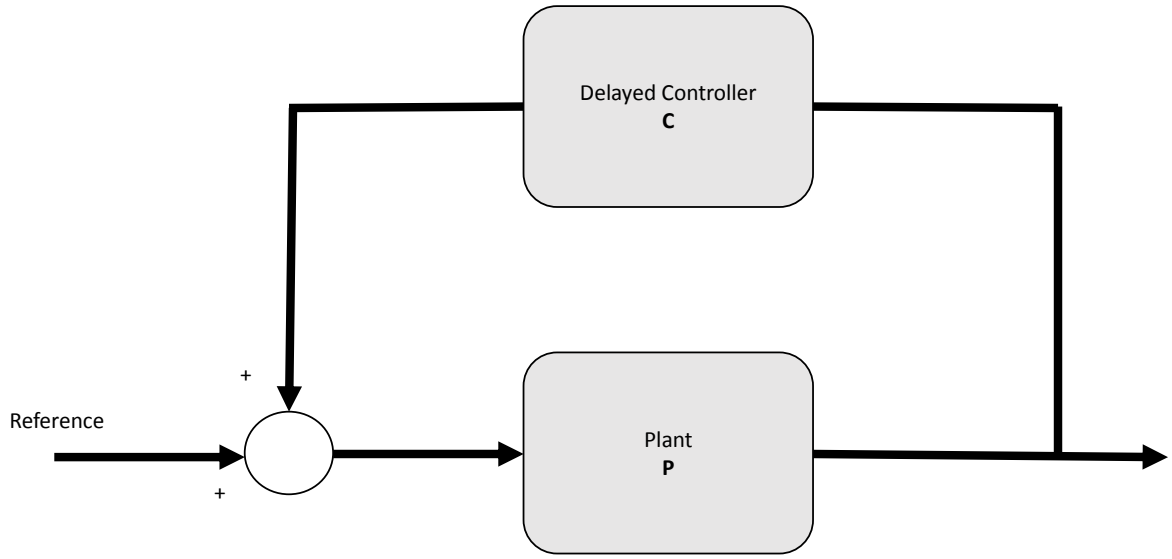


Figure 2.5: Delayed feedback controller design.

with the stability properties mentioned abov in mind.

Hence, delayed feedback controller design with maximum damping is achieved by solving the optimization problem in Equation 2.52. Note that this optimization problem is not discussed in this thesis and values presented in (Bisgaard 2007) is used. The delayed feedback controller used in this thesis is explained in detail in Section 5.5 The reader is recommended to consult (Nocedal and Wright 2006) for a more in-depth description of the solution of the constrained optimization problem in Equation 2.52.

Chapter 3

Introduction to the Multirotor Platform

This chapter presents the multirotor platform and its interaction with the navigation subsystems, completing the full multirotor system and its applications used in this thesis. The development of the multirotor platform in this Chapter was done in collaboration with fellow master student Jon-Håkon Bøe Roli.

A system overview is given, before presenting the different hardware components and explaining their applications. The chapter ends with an introduction to the necessary software tools. Further details regarding the main components can be found in Appendix A.

3.1 System Overview

The complete system can be divided into three segments:

- The multirotor with payload
- Mission control with the Ground Control Station (GCS) and pilot
- A high precision navigation subsystem

This thesis has investigated the use of a low-cost local navigation subsystem that works indoor and is based on radio signals using Atmel kit as shown in Figure 3.1. The Atmel kit and the development of this subsystem is discussed in extensive detail in Chapter 4. As seen in the figure it needs a minimum of three Atmel kits as *Initiators*, in addition to the one on the multirotor, to solve the problem of *trilateration*.

Moreover, two additional high precision navigation subsystems has been implemented to evaluate the quality of the Atmel system. The implementation has been performed in collaboration with fellow master students Jon-Håkon Bøe Roli and Andreas Lindahl Flåten. One of the systems is the *Piksi*, a low-cost, high-performance GPS receiver with RTK functionality for centimeter-level relative positioning accuracy and the other is based on OptiTrack motion technology using a state-of-the-art tracking system with sub-mm and sub degree accuracy. Both of the systems are briefly introduced in A.6 and A.7 respectively.

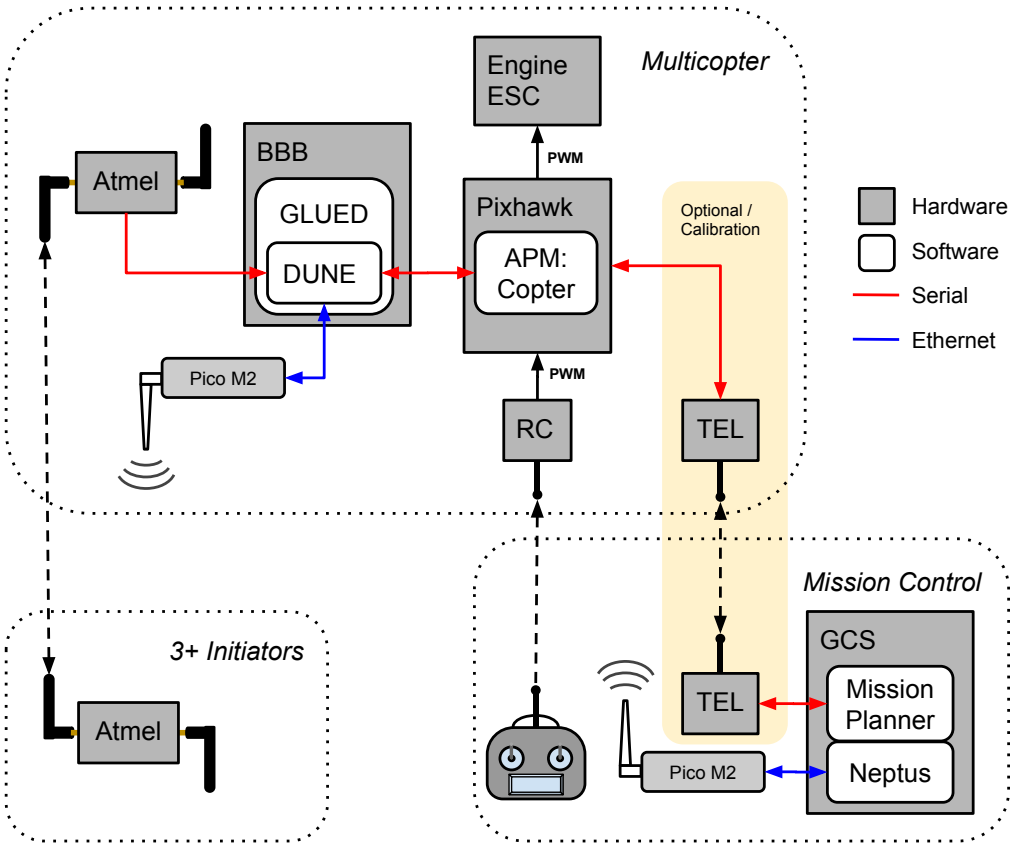


Figure 3.1: Overview of the complete system with the Atmel kit for precision navigation. A solid line illustrates a wired connection, while a dashed line illustrates a wireless connection. Arrowheads indicate the direction of information flow.

3.2 System Components

3.2.1 Multirotor

A hexacopter, a multirotor with six arms and rotors, is considered to be a good compromise between size, cost and weight. The ArduCopter 3DR Hexa B, shown in Figure 3.2, has been used with great success in previous projects at NTNU (Steen 2014; Andersen 2014), and was used as the multirotor platform in this project. A 4000 mAh 3 cell lithium polymer (LiPo) main battery is strapped to the bottom with velcro and used to power its motors.

3.2.2 Autopilot

The autopilot is responsible for controlling the speed of the 6 motors of the hexacopter through Pulse-Width Modulation (PWM) signals to their Electronic Speed Controller (ESC). Pixhawk, shown in Figure 3.3, was chosen as autopilot and it runs the APM:Copter software, described in Section 3.4, to achieve the necessary low-level control.

Pixhawk also contains the necessary sensors for basic navigation: An Inertial Measurement

Figure 3.2: The ArduCopter 3DR Hexa B. *Image by courtesy of 3drobotics.com.*

Unit (IMU) to determine the rotation and acceleration, a barometer for relative altitude measurements and an external GPS/compass module (not shown in Figure 3.1) to determine the position and direction.

The Pixhawk is powered by the same battery as the hexacopter and is also capable of monitoring its status.

3.2.3 Radio Controller

A Radio Controller (RC) is the typical tool used by a pilot for manual control of a UAV. The Spektrum DX7s shown in Figure 3.4 was used throughout this thesis. It sends a PWM signal to a receiver on the hexacopter, giving commands to Pixhawk. The autopilot then does the actual control of the hexacopter, as it is impossible for a human operator to stabilize it by commanding the speed of the individual motors.

3.2.4 Payload Computer

BeagleBone Black (BBB) shown in Figure 3.5a was chosen for the payload computer. Its use in a multirotor payload was advised by (Steen 2014) and (Klausen 2013) used it with great success on small autonomous robots.

The BBB is the brain of the hexacopter, connecting all the important components together. It provides guidance for the hexacopter through the Pixhawk and interfacing the on-board Atmel kit to receive distance measurements and calculate local positioning. It runs tasks in DUNE to achieve this, on a minimal Linux distribution called GLUED (see Section 3.3).

A BBB is also used to interface the high precision positioning system - *Piksi* at the base station. The high precision positioning system is used to determine the accuracy of the Atmel

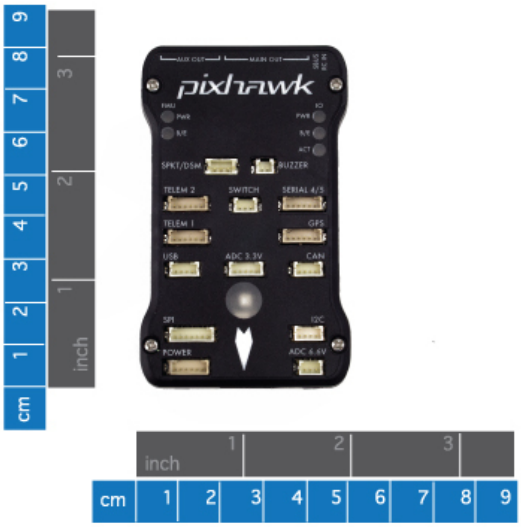


Figure 3.3: The 3DR Pixhawk. *Image by courtesy of 3drobotics.com.*

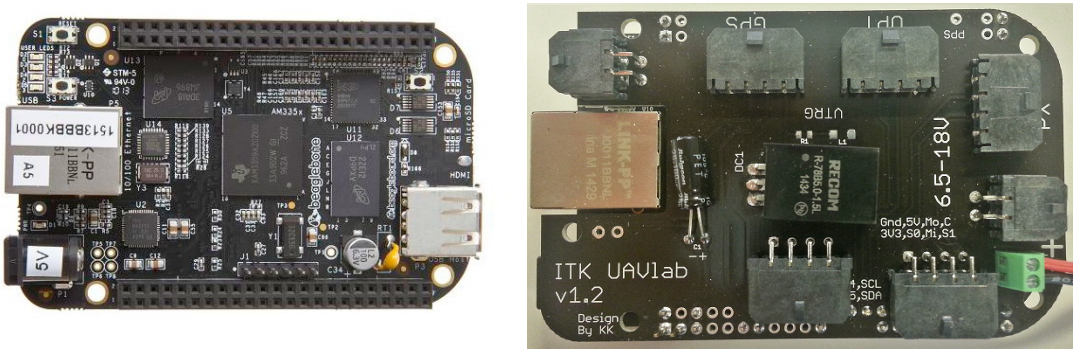
Figure 3.4: The Spektrum DX7s with its receiver. *Image by courtesy of hobbyfly.com.*

kit positioning system, which is discussed in Section 4.6. More about the integrated Piksi system can be found in (Røli 2015).

To supply the BBB with power and simplify connections to other components, a special add-on board was made, supplied with an 800 mAh 3 cell lithium polymer (LiPo) payload battery. The add-on board can be seen in Figure 3.5b attached to a BBB.

3.2.5 Ground Control Station

The Ground Control Station (GCS) in our system runs a software called Neptus to interact with DUNE on the payload computer and perform command and control. Neptus is also the tool responsible for data collection and analysis. A short introduction is given in section 3.3.3.



(a) The BBB. *Image by courtesy of beagleboard.org.*

(b) The custom made cape that is attached on-board BBB.

Figure 3.5: The payload computer, BeagleBone Black.

The yellow shaded link in Figure 3.1 is an optional telemetry link from Pixhawk, used to interface another control software called Mission Planner (3D Robotics 2015d) on the GCS. A short description of Mission Planner is given in section 3.4.1, but the only intended use in the system is as a simple tool to calibrate Pixhawk and configure settings in APM:Copter.

3.2.6 Communication

The PicoStation M2 HP shown in Figure 3.6 is used to provide wireless network communication between the different units running DUNE or Neptus.

Figure 3.6: The PicoStation M2 HP, with and without its plastic housing.

The secondary calibration link between Mission Planner and the Pixhawk is a set of two simple 433 Mhz radios from 3D Robotics (3D Robotics 2015a).

3.3 Introduction to DUNE

DUNE: Uniform Navigational Environment (DUNE) is an open-source software solution used on board of the system. It is a unified navigational environment for vehicle on-board system, developed by the department of electrical and computer engineering at university of Porto, Portugal. DUNE is a part of the project at the Laboratório de Sistemas e Tecnologias Subaquáticas (LSTS), which is an interdisciplinary research laboratory (LSTS 2015c). Their goal is to develop a platform that is able to control vehicles in network. They have used the system to control heterogeneous autonomous vehicles in both single and multi-vehicle operations with success (Calado et al. 2008). In addition, the system is architecture (Intel x86 or compatible, Sun SPARC, ARM, PowerPC and MIPS) and operating system independent (Linux, Solaris, Apple Mac OS X, FreeBSD, NetBSD, OpenBSD, eCos, RTEM, Microsoft Windows 2000 or above and QNX Neutrino), making it highly portable. Therefore, it is found to be suitable for our intended use.

DUNE can interact with all the sensors on-board, including payload and actuators. It also communicates with the navigation system, and can be responsible for the system control, maneuvering, plan execution and vehicle supervision (LSTS 2015a). The communication is based on *IMC: Inter Module Communication* (Calado et al. 2008).

In this thesis, DUNE is used for navigation, using mainly the Atmel kit, but also the Piksi and the Vicon camera system for positioning, while the Pixhawk is responsible for the control system (autopiloting). Note that DUNE is communicating with the Pixhawk. The measurements from Atmel, used in positioning is calculated in BBB. The result is fed to Pixhawk for control purposes. In return the sensor information in Pixhawk is fed to BBB, via DUNE to achieve better positioning and consequently better navigation. The integration of the navigation systems will be discussed in detail in Chapter 4.

3.3.1 Inter Module Communication (IMC)

Inter Module Communication (IMC) protocol is the messaging protocol developed by LSTS to build interconnected systems of vehicles, sensors and human operators that exchange real-time information about their environment and objectives (Calado et al. 2008). The protocol enables all vehicles to listen and/or send necessary information to each other. It is also possible to extend the protocol by adding new messages.

3.3.2 GLUED

GLUED is the operative system installed on the BeagleBone Black and where DUNE is executed. The operative system is developed by LSTS for easy cross compiling with the target system and configuration properties (LSTS 2015b). It is a lightweight Linux distribution capable of running on embedded systems.

3.3.3 Neptus

Neptus is the command and control software used to command and monitor of unmanned vehicle operations developed by the LSTS, to solve the problem of operator-vehicle interaction, including mission planning and analysis (LSTS 2015d). The software is used with great success by the department of electrical and computer engineering at university of Porto, Portugal (LSTS 2015d). The software has three main features

- Planning
- Execution
- Review and Analysis

By adding vehicle profiles of each vehicle, the software can be used in planning operations by simulating and validating before execution. During the execution, the software can visualize the real time data from vehicles. It supports simulations, vehicle monitoring and control of multiple vehicles. After the execution, it is possible to review and analyze the result, as the software logs the operation data. The data can be used to evaluate and adjust future executions. Communication in Neptus is also based on IMC.

3.4 Introduction to APM:Copter

APM:Copter is an open source platform for multirotors and helicopters, offering a wide range of autopilot solutions. The solutions include both remote and autonomous flight control. The latter includes WP navigation, mission planning and telemetry on a ground station for monitoring (3D Robotics 2015b). The modularity and the capabilities of the software is extensive. The interested reader is referred to (3D Robotics 2015b). There are several flight modes available in APM:Copter, out of which three is used in this thesis:

Stabilize
: Pilot input to the RC is interpreted as desired angles in roll and pitch and average speed of the motors. Without input the copter will automatically level itself. This is the mode allowing the most aggressive maneuvering from the pilot, while still aiding with stability.

Loiter
: The copter tries to maintain a consistent location and orientation. RC input from the pilot is interpreted as the adjustment of said location and orientation, resulting in simpler and smoother control.

Guided
: The copter is capable to guide itself dynamically to a given target location. The target location is given by the GCS using either Mission Planner or Neptus. The signals are sent over telemetry link and the pilot is able to take over manual control if necessary using RC.

3.4.1 Mission Planner

Mission Planner is the ground control station of the APM:Copter. It can be connected to the vehicle using a telemetry radio and facilitate a command and control center for the vehicle. The Mission Planner enables us to use click-and-click way-point entries, using Google maps or other local or global maps. Pre-made operation commands can be saved and executed easily. On-line information about the vehicle is also shown, including the output from the Pixhawk's serial terminal. Moreover, it offers interface with a PC flight simulator and includes an easy and intuitive GUI to tune parameters and optimize the autopilot's behaviour.. However, it was mostly used for configuration of the APM settings for out airframe and troubleshooting.

Chapter 4

Indoor Navigation Based on Atmel REB233SMAD-EK Kit

An indoor navigation system should be able to track local position of any given target inside buildings. Therefore, commercial GNSS are not useful for indoor target tracking. However, a portable indoor navigation system can also be used as an alternative or backup for GNSS in for example marine, arctic or urban operations where the GPS signals can be unstable or limited.

Today, both indoor and outdoor localization problems are solved separately. The outdoor positioning systems are mainly based on satellite technology, while the indoor positioning systems are based on radio waves, camera technology, magnetic fields, acoustic signals, or other sensory information. However, even though the cost of GNSS has dropped radically in the last decade, the indoor navigation systems remain expensive, inconvenient to transport and are operational in limited areas. Consequently, an alternative solution, based on low cost and lightweight sensors that works in both indoor and outdoor environments, is highly desirable.

In this thesis, the Atmel REB233SMAD-EK kit, hereafter called *Atmel kit*, is investigated for this purpose. The radio transmitters measure distance between two nodes using radio signals. With several transmitters, principle of trilateration and the Extended Kalman Filter (EKF), a local position estimator is developed.

This chapter will introduce the position estimator. First, the Atmel kit with its interface software and necessary changes made for DUNE integration will be presented. Then the range measurement quality and optimal setup for both indoor and outdoor operations will be introduced. Furthermore, an optimal estimator design will be explained and practical implementation considerations will be discussed. The estimator is first tested in MATLAB with simulated values. The results from the test will be presented and used in the real world experiments. Lastly, the results from both indoor and outdoor experiments will be discussed.

4.1 Measurement Interface Software

The Atmel kit consists of two separable hardware parts, a transceiver part with two antennas and a micro controller part that is programmable[1]. In addition, a basic ranging application for range measurements between two or more nodes is included. This section will introduce the Atmel ranging application and changes made in the software for DUNE integration.

4.1.1 Atmel Ranging Application

A simplified model of the range setup is shown in Figure 4.1. For the experiments, 8 Atmel kits are used, one as coordinator, six as initiators, hereafter called anchors, and one as reflector.

The coordinator is connected to a terminal computer. It initiates the setup configuration and requests range measurement to the anchor(s). The anchor(s) conduct range measurement with the reflector using given configuration by the coordinator. The measurement result is transferred back to the coordinator. Several configuration options are available, such as antenna Tx power, basic filtering, frequency selection etc. The transferred data contains the measured range in cm and a Distance Quality Factor (DQF). Detailed description about the Atmel kit hardware, measurement output, configurable settings and the roles of coordinator, initiator and reflector can be found in Appendix A.5.

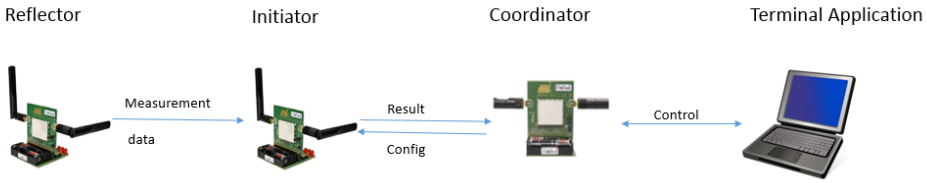


Figure 4.1: Simplified remote ranging setup. *Image by courtesy of atmel.com*

The ranging application that is included is easy to configure and operate. However, there are several drawbacks. First, the output data does not include any timestamp of the measured data. So the measurement times are unknown. Secondly, the coordinator waits for all the connected anchors to finish measuring before reporting back to the terminal. The measurements are conducted sequentially and with 6 anchors, a complete set of range measurements takes about 1.2 seconds. Assuming a velocity of 5 m/s, the distance traveled between the first and last measurements will be more than 5 m. Consequently, the estimated position in a dynamic system will deviate radically. Moreover, the coordinator and the terminal application are actually not necessary since the initiator will be connected to the target computer, where the calculation will be carried out. Therefore, the terminal application and the coordinator should be bypassed. The communication topology of the multirotor system is shown in Figure 3.1.

[1]The REB233SMAD-EK kit consists Atmel 2.4GHz transceivers AT86RF233 combined with Atmel ATxmega256A3 microcontrollers

Table 4.1: New binary output arrangement for Atmel kit for DUNE implementation.

Field	Bytes	Content
Preamble	1	0x55
Message type	2	Distance
Sender	2	Atmel kit
Length(n)	1	0x11
Payload	17	Distance (cm) - 4 bytes
		DQF (%) - 1 byte
		Reflector address - 4 bytes
		Anchor address - 4 bytes
		Timestamp (ms) - 4 bytes
CRC	2	

Therefore, some changes to the ranging application are necessary and will be discussed in the following section.

4.1.2 DUNE Integration

Necessary changes to the Atmel interface can be summed up by the following list:

- Increase the output frequency by changing the Atmel software to an easy to use and fast binary protocol.
- Add local timestamp with measurement time to the output data.
- Increase the output frequency by sending data as soon as they arrive instead of waiting for every anchor to finish measuring.
- Send measurement setup and range requests through DUNE directly to the Reflector, by-passing the Coordinator.
- Eliminate the need of a separate terminal application and a Coordinator.
- Eliminate the need of a separate battery on the Reflector. It should be powered from the BBB.

The development of the Atmel software was completed in cooperation with PhD candidate Krzysztof Cisek at the Department of Engineering Cybernetics, NTNU, Norway.

The output protocol was changed to the Swift Nav Binary Protocol (SBP). SBP is the native binary protocol used by the Piksi to transmit solutions, observations, status and debugging information (SwiftNAV 2015c). However, the SBP was primarily used as a fast and simple protocol to send the binary representation of C structs across serial links, with error detection capability and Cyclic Redundancy Check (CRC). The protocol was used since a portable C implementation was already provided as a part of libswiftnav, a platform independent library of GNSS related functions and algorithms for use by software-defined receivers and other software (SwiftNAV 2015a). Moreover, the interface of the Piksi was already implemented in DUNE, which made the Atmel interface implementation easy and of a protocol that worked well with DUNE. The resulting binary output is shown in Table 4.1.

In addition, a local timestamp was added as it can be seen in Table 4.1. Moreover the coordinator was bypassed by initiating the range requests to the reflector through DUNE. Hence the need of a terminal application was eliminated. Furthermore, the Reflector was programmed to send received data as soon as they arrived, and the Reflector was powered directly from the BBB with 3.3V.

These changes increased the measurement output frequency radically. While the Atmel ranging application had a frequency of about 1-1.5Hz, the new integrated Atmel kit could deliver frequencies up to 20-25Hz in perfect conditions.

4.2 Range Measurement Experiments

In this section, number of experiments conducted with the Atmel kit will be . The main goals of the experiments are to find an optimal measurement setup that minimizes signal deviation and improves accuracy. A preliminary test, where the relationship between the signal quality and antenna configurations will be discussed. Furthermore, maximum measurement range of the antennas will be determined. Moreover, the optimal measurement setup indoor and discussion about noise reduction will be presented. In addition, an interference test with the telemetry on-board multirotor will be introduced. Lastly, a discussion about the results will be presented.

The following experiments are based on the setup shown in Figure 4.1 with coordinator, terminal application, anchors and reflector.

4.2.1 Preliminary Experiments - Outdoor

The preliminary test was conducted outside in an open park near NTNU, Trondheim. Purpose of the test was to determine the quality of the measurement signals and the maximum measurement range of the Atmel kit. Moreover, their sensitivity to different scenarios, such as different reflector-initiator positions, antenna configurations, Tx power during ranging, static measurements and measurements during motion was investigated.

All of the nodes were placed in a triangle position as shown in Figure 4.2. Some of the distances were measured beforehand using a handhold laser measurement tool to have an initial idea of the distances. That being said, the measurements was not carried out thoroughly and a deviation up to 30 cm should be expected. The distances between the nodes can be seen in Figure 4.2. The anchors 1, 2 and 3 was 1.5 m above the ground while the rest was 0.3 m above the ground.

We emphasize that the main purpose of the preliminary test was to determine the deviation in signal quality related to the different antenna configurations and not the accuracy. Both 'open arm' - each antenna 45° upwards and 'one up' - one antenna straight, setup was tested. The measurement results are shown in Table 4.2. Note that the signal deviation is reduced remarkably when the 'open arm' configuration is used. Similar observations was made when the reflector position was changed. The deviation is believed to be a consequence of the 'one up' setup is more prone to multipath signals from the ground. Also note

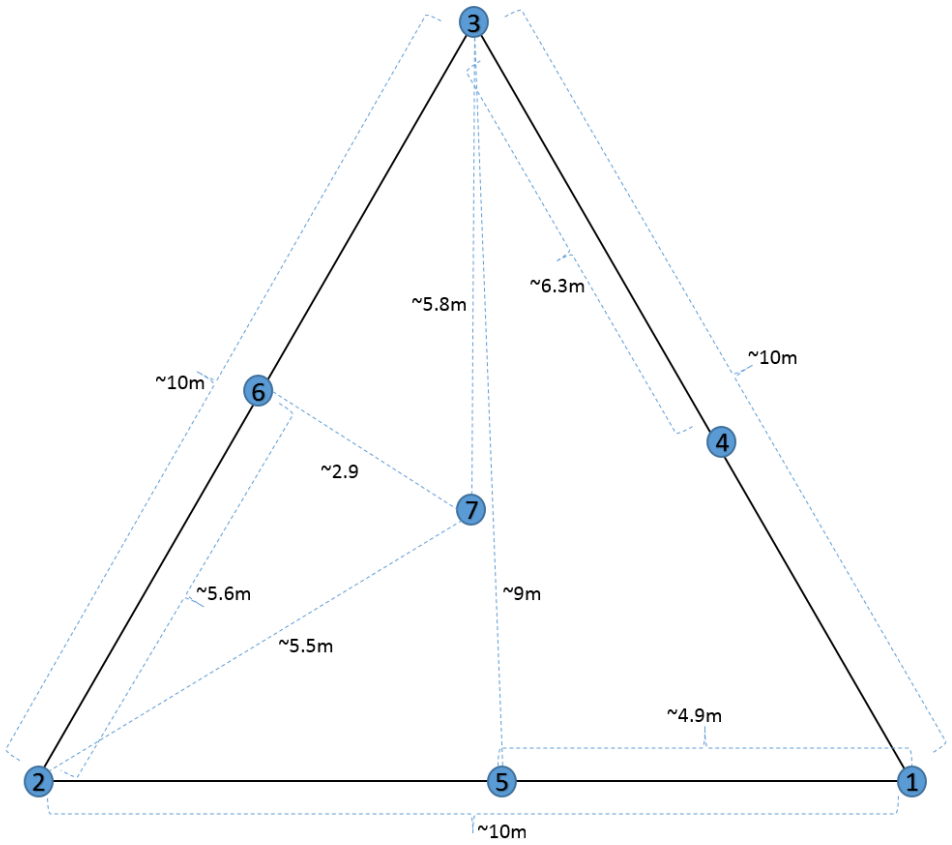


Figure 4.2: Measurement setup of the preliminary test.

that the measured mean values are higher than the actual values. However, this was expected since multipath will increase the travel time, and consequently the mean value of the range measurements.

Table 4.2: Measurement results from preliminary test. The measured distance mean and the standard deviation with two antenna configurations are shown.

From anchor 1 to:	Open Arm - mean	One up - mean	Open Arm -STD	One Up - STD
2	9.5568	8.8839	0.2599	0.7175
3	10.4783	10.3914	0.1140	0.2545
4	4.6083	4.5842	0.0649	0.3087
5	4.9182	5.1065	0.0666	0.0821
6	9.2304	8.9250	0.0752	0.6221
7	6.4272	6.1221	0.0874	0.6919

Maximum Signal Range

The maximum signal range was found by moving the reflector away from the anchors until the signal was lost. The reflector was held 2 meters above the ground and line of sight was kept during the measurements. The result from the range measurements is shown in Figure 4.3. Note that the maximum range achieved was about 70 meters. This is directly related to the frequency step of the signal that was set to 2 MHz. Which gives about 150 meters

in wavelength, and resulting in a theoretical maximum range of 75 meters. It is possible to reduce the frequency step down to 0.5 MHz, hence achieving a maximum range of 300 meters. This was later confirmed empirically at Breivika Airfield using 4dBm Tx power and is discussed in Section 4.6.1.

The same experiment was conducted with reduced Tx power during ranging. This parameter is of special interest since it is expected that the signal noise will be reduced indoors when the Tx power is reduced. The Tx power was reduced from 4dBm to -17dBm, resulting in 20 meters reduction of maximum range. In addition the amount of signal loss was increased. However, the noise was reduced.

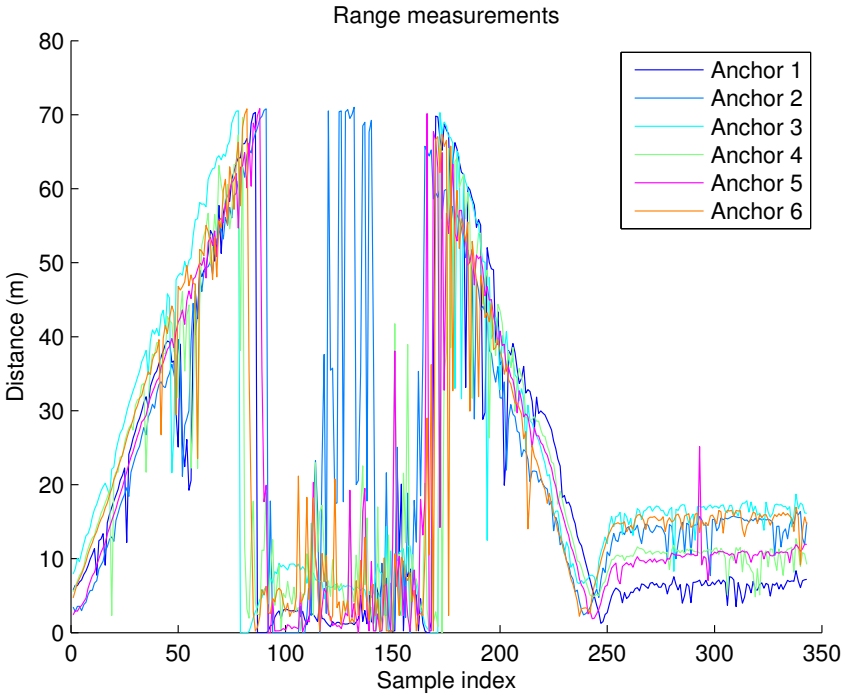


Figure 4.3: Measurement result of maximum range experiment. The reflector was moved away from the anchors and returned back when the signal was lost.

Conclusion

It is clear that the range measurement quality is higher when we use 'open arm' antenna position. The reduction in noise is significant. Moreover, the antenna power affects the noise in the measurements. Higher Tx power produce more noise while lower power reduce the signal quality. Consequently, there is a trade-off between noise and signal quality. Furthermore, the maximum range of the Atmel kit is satisfactory for indoor applications and is not seen as a limitation.

4.2.2 Measurement Performance - Indoor

The indoor range measurements were conducted in *Taklabben* at the Faculty of Information Technology, Mathematics and Electrical Engineering, NTNU. The anchors were placed in a measured and known position near the roof using a costum-made mount that was 3D-printed. One of the anchors with its mount is shown in Figure 4.4.

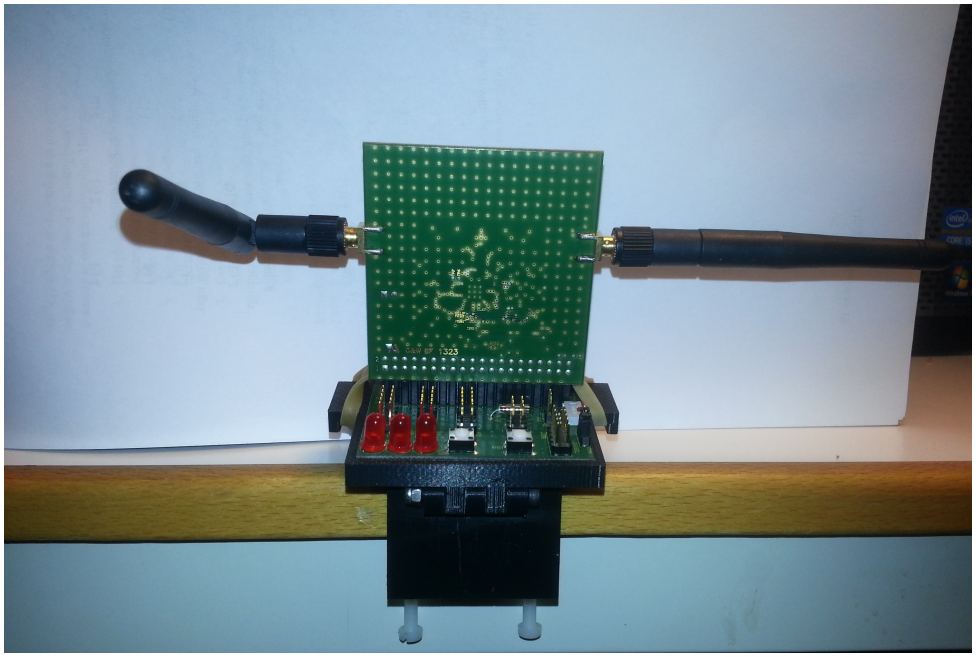


Figure 4.4: Image of Atmel transceiver with the custom-made mount for anchors.

The measurement set-up is shown in Figure 4.5. Distances between the anchors were measured carefully with a tape measure. However, a measurement deviation of 5-10 cm should be expected. Furthermore, the reflector was placed on a stool at center of the room. The NED distance between the reflector and origo was also measured.

Several experiments were conducted to find the optimal setup indoors. Optimal in this case refers to reduced noise and reduced effect of multipath. First, the optimal antenna configuration (antenna position and Tx power) was found, followed by the determination of accuracy.

Determining Optimal Indoor Antenna Configuration

Similar to the preliminary outdoor test, the optimal antenna position was determined by experiments. All the anchors were set to measure their distance to the reflector, both with the 'open arm' and 'one up' configuration. The 'one up' configuration is equal to the position shown in Figure 4.4. The test was conducted with -17 dBm, 0 dBm and with 4 dBm Tx power. Resulting values of the standard deviation after 100 samples of each experiment is shown in Table 4.3.

It can easily be observed that the 'one-up' antenna position is less noisy compared to the 'open-arm' antenna position. This is the opposite of the observation outdoors. However, it

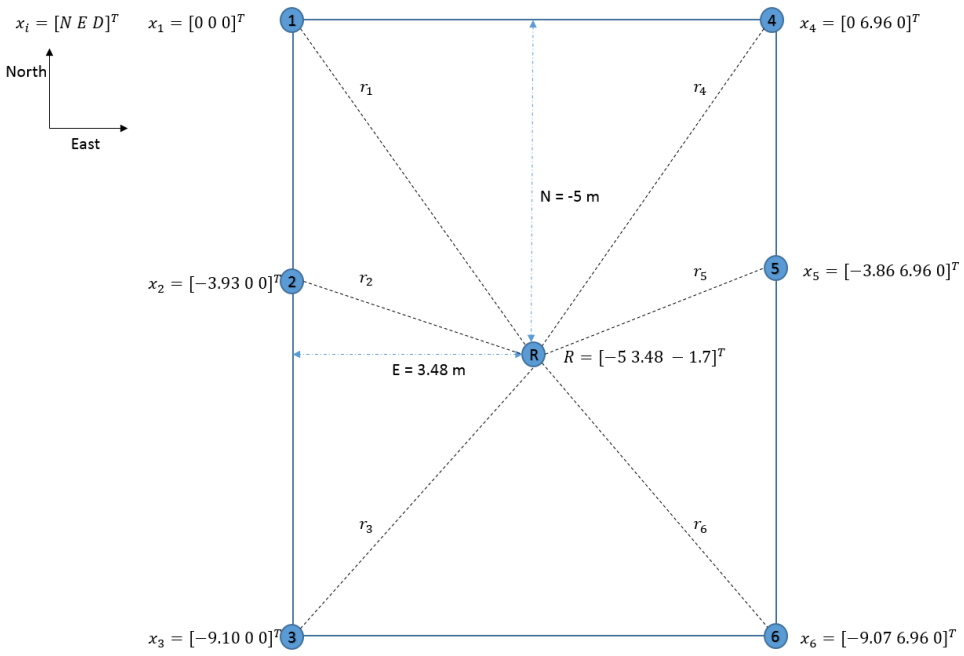


Figure 4.5: Indoor measurement setup.

Table 4.3: Standard deviations from range measurements indoor using different antenna configurations.

Measure to reflector	Open-arm			One-up		
Tx Power	-17 dBm	0 dBm	4 dBm	-17 dBm	0 dBm	4 dBm
Anchor 1	0.5246 m	0.5306 m	0.5870 m	0.2501 m	0.1868 m	0.2094 m
Anchor 2	0.1968 m	0.1937 m	0.1902 m	0.2623 m	0.0801 m	0.1474 m
Anchor 3	0.6197 m	0.6030 m	0.5497 m	0.6350 m	0.6168 m	0.6008 m
Anchor 4	0.8092 m	0.7278 m	0.6788 m	0.8493 m	0.6353 m	0.5882 m
Anchor 5	2.9703 m	2.8440 m	0.0801 m	0.1891 m	0.2415 m	0.0962 m
Anchor 6	1.5097 m	0.1401 m	0.5269 m	0.3208 m	0.2275 m	0.2398 m

can be explained by considering multipath. Since the 'open-arm' configuration points the antennas upwards, it is sensitive to multipath coming from above. For the outdoor case, this was not a problem. For indoor case however, the signals are bouncing on the walls, including the roof, making it prone to multipath from above. Since there are two antennas at each Atmel transceiver and the filtering in Atmel, explained in Appendix A.5, chooses the minimum distance measurement among the four measurements (2 antenna at anchor and 2 antenna at reflector, resulting in 4 measurements at each node measurement), the 'one-up' configuration is advantageous.

Furthermore, the signal deviation is reduced with higher Tx power on the antenna. However, according to (Atmel 2015b) lower antenna Tx power should reduce multipath. Nonetheless, it have been observed that the measurements are sensitive to their physical surroundings, which can explain the contradictory observations. In addition, it could be observed that the DQF is reduced by 20-30 % for the low Tx power. Therefore, 4 dBm, which is the maximum antenna Tx power, will be used hereafter.

The following section will investigate the accuracy of the indoor measurements.

Static Accuracy

The accuracy was determined by taking distances measurements between anchors since their relative distances are measured analogously. All the anchors normal to each other was measured with 1000 samples. Anchor 1 measured its distance to anchor 2, 3 and 4, while anchor 2 measured to anchor 1, 3 and 5, and so on. However, the signals were preprocessed before calculations. Due to signal loss and outliers, the mean value would be erroneous to used unprocessed for accuracy determination.

Consider the measurement between anchor 1 and anchor 2 shown in Figure 4.6 for instance. The outliers will affect the measured mean value radically. Therefore, the outliers are removed using Grubbs test with significance level of 0.05 (Grubbs 1950). The histogram in Figure 4.7 shows the difference between the unprocessed and processed data, with corresponding mean values of 3.9286 m and 3.9796 m respectively. Although the unprocessed measurement data seems to be closer to the real value (3.93 m), it is not representative. Due to signal loss, the measured mean value is reduced. Consequently, the unprocessed data is not reflecting the measurement accuracy correctly.

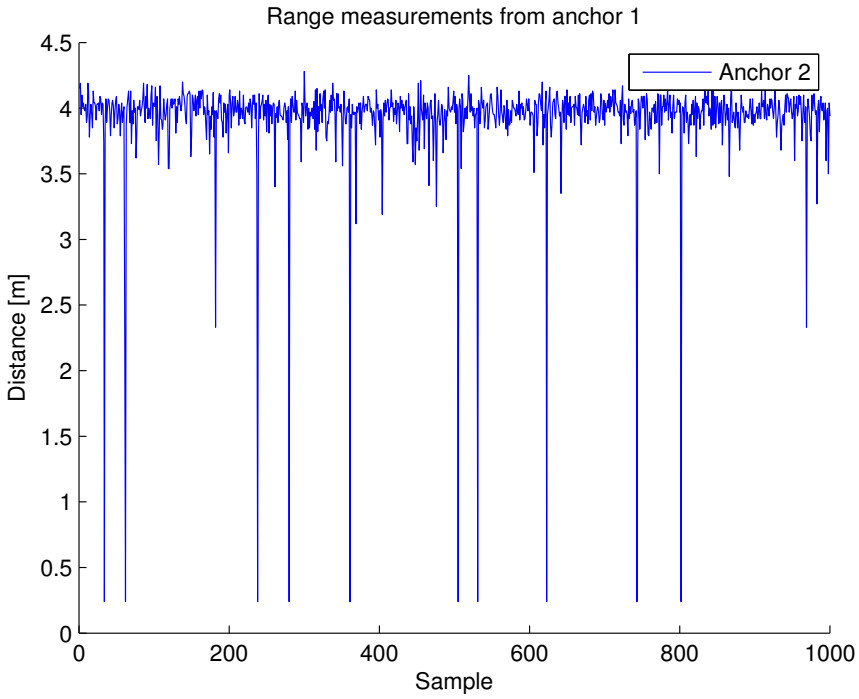


Figure 4.6: Distance measurement between anchor 1 and anchor 2 indoor at Taklabben with the setup shown in Figure 4.5.

It can be seen that the measured mean values differ from true values. The accuracy can deviate up to 1 m, excluding the most extreme deviations at anchor 5 and 4. The extreme deviation at anchor 5 can be explained by its position, where it had a reflective background. The deviations differ remarkably from anchor to anchor, which can be explained by the antenna

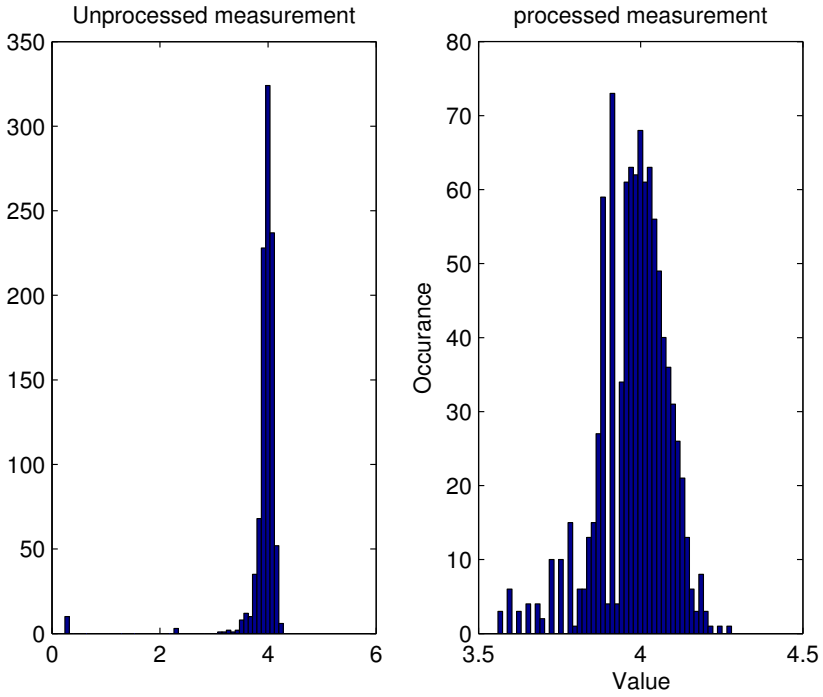


Figure 4.7: Histogram of measurement between anchor 1 and anchor 2. Unprocessed data on the left. Processed data on the right.

Table 4.4: Difference between true distance and measured mean values between anchor-to-anhcor measurements indoor. The true distances can be seen in Figure 4.5

From - to	Calculated error	From - to	Calculated error	From - to	Calculated error
1 - 2	-0.0496 m	3 - 1	-1.1913 m	5 - 2	0.4077 m
1 - 3	-1.3884 m	3 - 2	-0.3008 m	5 - 4	-8.6882 m
1 - 4	1.8180 m	3 - 6	0.3236 m	5 - 6	0.0810 m
2 - 1	0.1446 m	4 - 1	0.9819 m	6 - 3	0.1570 m
2 - 3	0.2990 m	4 -5	-4.3364 m	6 - 4	-0.2180 m
2 - 5	0.2154 m	4 - 6	-0.5428 m	6 - 5	0.4771 m

positions. The antenna configuration at each anchor was actually optimized to minimize deviation when measuring a target placed at the center of the room. Therefore, anchor to anchor measurements may not be optimal, proving the necessity of optimizing the anchor positions for each experiment.

4.2.3 Interference Test

The reflector and the PicoStation M2 HP are placed close to each other on the multirotor payload. Figure 4.8 shows the Picostation in white and Atmel reflector in green with black antennas. Both transceivers operate on 2.4 GHz band, which can be a source of measurement interference.

Figure 4.9 shows the percentage of signal loss at the reflector when the PicoStation was trans-

Figure 4.8: The multirotor system. Note that the Atmel reflector (green) and Picostation (white) are relatively close.

ferring data. The PicoStation was set to transfer over 2 MB/s during the test. Two Atmel transceivers were placed 5 m apart in a room. 1000 samples of range measurements were taken, with and without PicoStation near the Atmel transceivers. When the PicoStation was removed, the percentage of signal loss was about 0.3 %. When the PicoStation was set between the nodes, near the anchor, the percentage of signal loss was about 20%. 20% signal loss is very high and can cause safety issues during flight. However, when the PicoStation was removed from line of sight to the side of the anchor, the percentage of signal loss was reduced to 7.5%. This reduction is remarkable and the percentage is within acceptable limits.

However, another unexpected problem was unveiled during this test. The speed of the Atmel measurement outputs were radically slowed down by the signals from PicoStation M2 HP. The measurement frequency went from 20-25 Hz down to 1-2 Hz. This is found to be due to both using the same frequency band: namely 2.4 GHz. The problem was solved by reducing the signal power of the PicoStation from 23 dBm to -3 dBm and changing channels so the PicoStation used the upper frequency band of the 2.4 GHz, while the Atmel transceivers was set to use the lower frequency band. However, this solution brought up some more serious considerations. The signal power reduction reduced the range between the multirotor and the Ground Control Station (GCS) considerably. During the tests, the copter lost connection to GCS at about 30 m. This limitation is unacceptable for serious autonomous operations. Moreover, the data transfer between the GCS and the multirotor was also slowed down to critical levels.

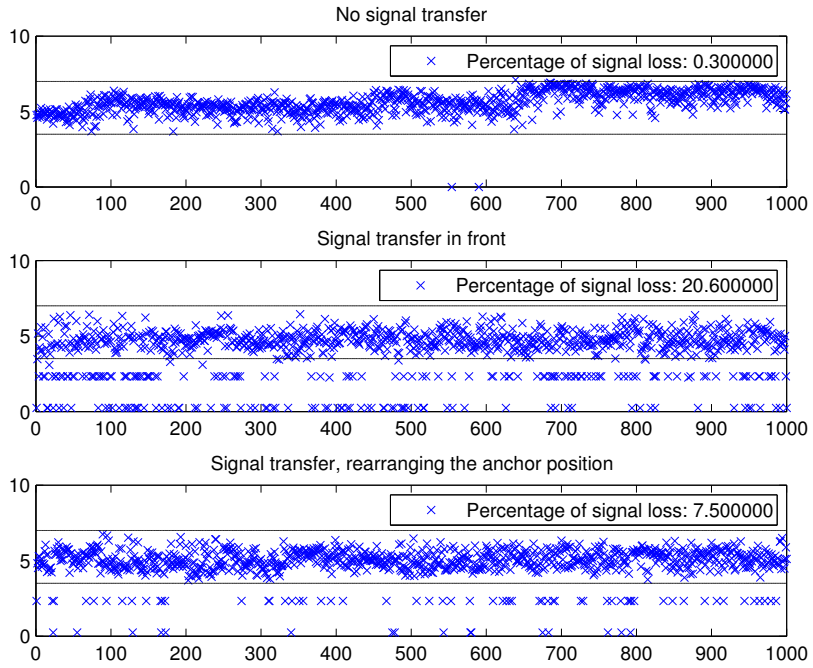


Figure 4.9: Interference test - influence of the hexacopter telemetry on Atmel kit range measurements.

4.2.4 Discussion

The Atmel kit seems to work satisfactory. They are easy to setup and configure. However, they are sensitive to their environments and multipath may cause problems indoor. It was therefore difficult to set an absolute measurement accuracy. Nonetheless, knowledge about the sensitivity can be used to determine the optimal working space of the equipment and set limits on the operation such that it can be carried out safely. The findings from the experiments are,

- Atmel kits are highly portable and easy to set up.
- Atmel kits are power efficient.
- Range measurement application by Atmel seems very sensitive to its surroundings.
- Measurements work satisfactory outdoors, with low signal deviation and low multipath generation.
- Multipath is a problem indoor, especially if the reflector is near the ground, roof or walls.
- The measurement frequency may be too low for fast dynamic targets.
- Atmel signals operate on the same frequency band (2.4GHz) as the payload telemetry, which can cause serious slowdown of the measurement frequency output.
- Reducing the payload telemetry signal power reduces the Atmel measurement fre-

quency slowdown. However, it also reduces the range and speed of the payload telemetry remarkably.

It is important to note that the measurements sensitivity to surroundings can be reduced utilizing the raw phase measurement between two radio transceivers for distance estimation. In (Pelka, Bollmeyer, and Hellbr 2014), the authors claim that the distance estimation accuracy is reduced to about 8 - 20 cm with similar Atmel radio transceivers. They use a mathematical model to estimate distances from phase measurements with multiple frequencies and provide a systematic analysis of possible sources of errors. Furthermore, (Dil and Havinga 2011) presents a novel radio interferometric positioning system, which can provide centimeter accuracy in the 2.4 GHz range. These methods should be investigated for future implementation to enhance the measurement quality, and consequently improve flight safety.

However, frequency reduction due to interference was unexpected. After consulting professor Fredrik Gustafsson, a professor at Department of Electrical Engineering, Linköping University, it was recommended to check out the company BeSpoon SAS. They offer several products for real time location systems. *UPosition IR-UWB Module* (BeSpoonASA 2015) is 13.4 mm * 13.4 mm * 2.55 mm in dimension, has Tx power up to -8 dBm and boast a range measurement precision down to 10 cm for ranges up to 880 m. And maybe most importantly, it operates on the frequency band from 3 to 5 GHz and costs 75% less than the Atmel kit. However, the product was still in development phase during critical part of the project and was not ready for experiments during this thesis.

4.3 Estimator Design - Target Tracking

The localization problem is solved in this section with the Extended Kalman Filter (EKF) algorithm, where the principle of trilateration and sensor fusion is utilized. First, a brief introduction to why the EKF was selected, followed by a brief introduction to the trilateration principle will be given. Moreover, the system state equations for the EKF, solving the positioning problem will be presented. Lastly, improvement of the standard EKF, utilizing sensor fusion techniques, outlier detection and rejection, and weighted covariance with Fuzzy Logic Adaptive Controller, will be introduced.

4.3.1 Filter Selection - Extended Kalman Filter (EKF)

The localization problem is solved utilizing measurement data from heterogeneous sensors. The measurement data are often fused utilizing the EKF method (St-pierre and Gingras 2004). The EKF is a nonlinear extension of the KF and is presented in Section 2.1.2. In addition, the Unscented Kalman Filter (UKF) is also a popular alternative. UKF propagates the so-called sigma points by function evaluations using unscented transformation, which differs from the EKF algorithm that is based on linearized models (Gustafsson 2010). It has been shown that UKF is superior to EKF for problems where the non-linearity is severe (Gustafsson and Hendeby 2012). However, in most practical problems where the non-linearity is

moderate, the difference in performance proves to be marginal (Cotugno et al. 2013). Furthermore, the computational cost is significantly greater than the EKF (St-pierre and Gingras 2004).

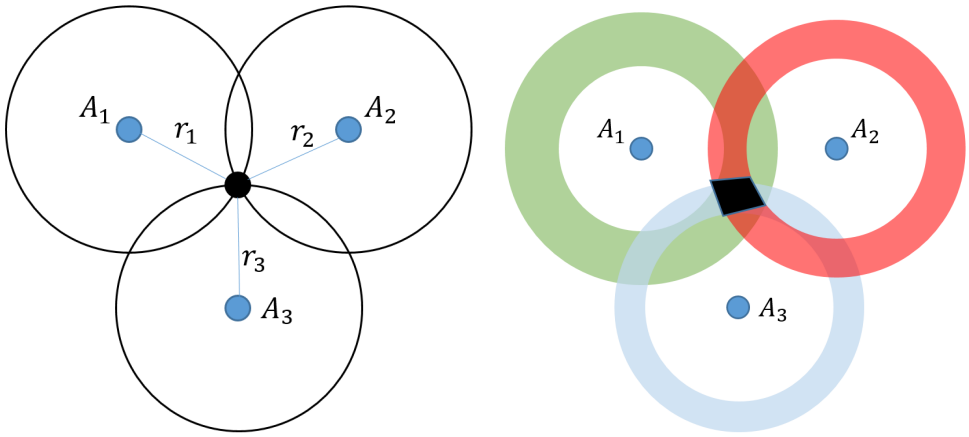
Also a common alternative for UKF is the Particle Filter (PF). The PF is a more generalized method and does not require the unmodeled noises to be Gaussian, as in KF, EKF and UKF (Gustafsson 2010). The posterior probabilities can be represented by weighted samples. (Lee et al. 2010) has shown that the PF outperforms UKF in tracking a target indoors based on radio-frequency identification on a ZigBee test-bed, especially if the environment is prone to multipath. However, if the assumption of a Gaussion noise model holds, the KF outperforms the PF in the linear case (Kalman 1960). Consequently, similar to the case in UKF, the benefit of the PF is often too small, compared to computational cost (György, Kelemen, and Dávid 2014).

The non-linearities in the localization problem based on trilateration are often moderate, therefore the Extended Kalman Filter is deemed sufficient as the solving algorithm.

4.3.2 Positioning Principle - Trilateration

Trilateration is an analytical localization method that uses distance measurements to locate the position of a target. Using reference points A_1, A_2 and A_3 with known positions and distances r_1, r_2 and r_3 to the target object, the object can be located at the intersection of the three circles. Figure 4.10a describes the simple case in 2 dimensions. It is easily transferred into 3D, simply by replacing the circles with spheres.

However, localization in reality is a bit more complicated. The measurements are often noisy and fluctuated. Figure 4.10b shows the principle in 2D. The location of the unknown target can be anywhere within the dark overlapped region rather than a single point of intersection. Thereby the uncertainty of the location is directly related to the accuracy of the range measurements.



(a) Trilateration **(b)** Trilateration with uncertainty

Figure 4.10: Principle of trilateration. Target position in black.

The localization problem can be solved by a simple algorithm. Assume that the coordinates

of the unknown node R are (N_R, E_R, D_R). The coordinates of the three anchor nodes A_1, A_2 and A_3 are (N_1, E_1, D_1), (N_2, E_2, D_2) and (N_3, E_3, D_3), and the distances between the target node R and the anchors are r_1, r_2 and r_3. The unknown position coordinates of R can then be found by solving the equations,

$$\begin{aligned}(N_1 - N_R)^2 + (E_1 - E_R)^2 + (D_1 - D_R)^2 &= r_1^2 \\ (N_2 - N_R)^2 + (E_2 - E_R)^2 + (D_2 - D_R)^2 &= r_2^2 \\ (N_3 - N_R)^2 + (E_3 - E_R)^2 + (D_3 - D_R)^2 &= r_3^2\end{aligned} \tag{4.1}$$

Although the equations seems very simple, the algorithm is easily affected by noise, interference and multipaths. Therefore, more than 3 anchors will be used in the EKF to reduce the deviation when solving the localization problem. In addition, the accuracy can be improved by adding other sensors, such as accelerometer and altimeter which will be discussed later.

4.3.3 Position Estimator Based on Extended Kalman Filter (EKF)

Kalman Filter (KF) utilizes prior knowledge of noise characteristics to filter out the measurement noise. However, the noise can not be modeled. It is assumed that the process and noise measurements are independent, white and Gaussian. For nonlinear processes, the EKF algorithm is used. The theory behind EKF is presented in Section 2.1.

There are different ways of modeling the trajectory of a moving object. The most simple one is to use the states x_N, x_E and x_D, referring to NED positions respectively, to obtain the Position model (P) using an EKF (Shareef and Zhu 2009). Velocity and acceleration states can be added to obtain Position-Velocity model (PV) and Position-Velocity-Acceleration model (PVA). The PVA model will be explained in detail in the following section, as it is an expansion of the PV model, which again is an expansion of the P model. However, all three models have been implemented and their performance is compared.

System State Equations

The state variables for the PVA model includes position, velocity and acceleration. Hence, the state variables can be represented by the state vector

$$\mathbf{X} = [x_N, x_E, x_D, v_N, v_E, v_D, a_N, a_E, a_D]^T \tag{4.2}$$

where x_N, x_E and x_D represents the NED position of the object respectively, v_N, v_E and v_D represents the NED velocity components respectively and a_N, a_E and a_D represents the NED acceleration components respectively.

For practical reasons, assume that the sampling period T between each measurement is very low. Then the motion of the tracked object can be regarded as uniformly accelerated in every direction between each sampling. Consequently, the kinematic equation of the relationship

between position, velocity and acceleration can be expressed using the Fundemental Theorem of Calculus. According to (Serway and Jewett 2013), with the assumption of constant acceleration, the relationship can be expressed as

$$x = x_0 + vT + \frac{1}{2}aT^2 \tag{4.3}$$

where x is the end position, x_0 is the start position, v is the velocity, a is the acceleration and T is the time difference between start and end. In practise however, assumption of constant acceleration may not hold, if the sampling frequency of the Atmel kit, discussed in Section 4.1.1, is not sufficiently low.

Note that (4.3) contains position, velocity and acceleration components. Hence, the PVA model will be based on (4.3), assuming constant acceleration. The PV model is based on (4.3) with $a = 0$, assuming constant velocity, and the P model is based on (4.3) with $a = 0$ and $v = 0$, assuming no motion.

The state-space model of the linear PVA model at time step k becomes,

$$\mathbf{X}_k = \mathbf{F}\mathbf{X}_{k-1} + \mathbf{W}_k \tag{4.4}$$

where $\mathbf{W}_k$ is the process noise given by:

$$\mathbf{W}_k = [0, 0, \omega_{a_N}, 0, 0, \omega_{a_E}, 0, 0, \omega_{a_D}]^T \tag{4.5}$$

ω_{a_N}, ω_{a_E} and ω_{a_D} is the zero mean Gaussian white noise $(0, \sigma^2_{a_N})$, $(0, \sigma^2_{a_E})$ and $(0, \sigma^2_{a_D})$, respectively. The state transition matrix $\mathbf{F}$ with the states $\mathbf{X} = [x_N, x_E, x_D, v_N, v_E, v_D, a_N, a_E, a_D]^T$ for the PVA model using (4.3) for north, east and down position becomes,

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 & \frac{T^2}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & T & 0 & 0 & \frac{T^2}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & T & 0 & 0 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

Note that there is no process input to this system. However, there are measurements inputs. The measured distances will be used to update the state of the object as given in the algorithm in Table 2.2. Furthermore, the process noise matrix Q accounts for the unmodeled factors of the system, i.e. the unmodeled factors are treated as random noise. In (4.6), the change in velocity is accounted for by acceleration, while the change in acceleration is not accounted for. Therefore, the effects of acceleration are treated as random noise (Shareef and Zhu 2009). However, keep in mind that the covariance matrix Q is in practice a design parameter and can be tuned freely to achieve optimal response from the system. During the

simulations and experiments, Q was set to be diagonal matrix with values found appropriate.

Also note that the PV and P model is achieved simply by using the states $\mathbf{X} = [x_N, x_E, x_D, v_N, v_E, v_D]^T$ and $\mathbf{X} = [x_N, x_E, x_D]^T$, respectively. Consequently, the dimensions of the state transition matrix $\mathbf{F}^{9x9}$ and the covariance matrix $\mathbf{Q}^{9x9}$ is reduced to $\mathbf{F}^{6x6}$ and $\mathbf{Q}^{6x6}$ for the PV model and $\mathbf{F}^{3x3}$ and $\mathbf{Q}^{3x3}$ for the P model. The n in $\mathbf{F}^{nxn}$ denotes the dimensions of the matrices.

System Measurement Equations

Assume that all distance measurements r are available at time step k. Using trilateration algorithm, discussed in Section 4.3.2 and including all the anchors available, the equations become,

$$\begin{aligned}
r_{1k} &= \sqrt{(x_{N1} - x_{Nk})^2 + (x_{E1} - x_{Ek})^2 + (x_{D1} - x_{Dk})^2} + \tilde{r}_{1k} \\
r_{2k} &= \sqrt{(x_{N2} - x_{Nk})^2 + (x_{E2} - x_{Ek})^2 + (x_{D2} - x_{Dk})^2} + \tilde{r}_{2k} \\
r_{3k} &= \sqrt{(x_{N3} - x_{Nk})^2 + (x_{E3} - x_{Ek})^2 + (x_{D3} - x_{Dk})^2} + \tilde{r}_{3k} \\
&\vdots \\
r_{nk} &= \sqrt{(x_{Nn} - x_{Nk})^2 + (x_{En} - x_{Ek})^2 + (x_{Dn} - x_{Dk})^2} + \tilde{r}_{nk}
\end{aligned} \tag{4.7}$$

where n is the number of anchors, x_{Ni}, x_{Ei}, x_{Di} are the known north, east and down positions of the anchors, respectively, x_{Nk}, x_{Ek} and x_{Dk} are the unknown north, east and down positions of the tracked object, and $\tilde{r_{ik}}$ is the measurement error. The system measurement equations can be expressed in vector form as:

$$\mathbf{h}_k = \mathbf{h}'\mathbf{X}_k + \tilde{\mathbf{r}}_{1k} \tag{4.8}$$

where $\mathbf{h}_k = [r_{1k} r_{2k} \dots r_{nk}]^T$ is the range measurements using current state output and $\mathbf{h}'$ is the measurement matrix that relates to the current state output. As explained in Section 2.1.2, the Jacobian is used since the output equations are nonlinear, hence:

$$\mathbf{h}' = \begin{bmatrix}
\frac{\partial r_1}{\partial x_N} & \frac{\partial r_1}{\partial x_E} & \frac{\partial r_1}{\partial x_D} & \frac{\partial r_1}{\partial v_N} & \frac{\partial r_1}{\partial v_E} & \frac{\partial r_1}{\partial v_D} & \frac{\partial r_1}{\partial a_N} & \frac{\partial r_1}{\partial a_E} & \frac{\partial r_1}{\partial a_D} \\
\vdots & & & & & & & & \\
\frac{\partial r_n}{\partial x_N} & \frac{\partial r_n}{\partial x_E} & \frac{\partial r_n}{\partial x_D} & \frac{\partial r_n}{\partial v_N} & \frac{\partial r_n}{\partial v_E} & \frac{\partial r_n}{\partial v_D} & \frac{\partial r_n}{\partial a_N} & \frac{\partial r_n}{\partial a_E} & \frac{\partial r_n}{\partial a_D}
\end{bmatrix} \tag{4.9}$$

where the partial derivatives are given by:

$$\begin{aligned}\frac{\partial r_i}{\partial x_N} &= \frac{x_{Ni} - x_N}{\sqrt{(x_{Ni} - x_N)^2 + (x_{Ei} - x_E)^2 + (x_{Di} - x_D)^2}} \\ \frac{\partial r_i}{\partial x_E} &= \frac{x_{Ei} - x_E}{\sqrt{(x_{Ni} - x_N)^2 + (x_{Ei} - x_E)^2 + (x_{Di} - x_D)^2}} \\ \frac{\partial r_i}{\partial x_D} &= \frac{x_{Di} - x_D}{\sqrt{(x_{Ni} - x_N)^2 + (x_{Ei} - x_E)^2 + (x_{Di} - x_D)^2}}\end{aligned} \tag{4.10}$$

where i denotes the anchor number. Note that the denominator in (4.10) can be zero and cause instability. However, the seemingly singular $\mathbf{h}'$ have both theoretical and practical solution. In theory, singular denominator also means singular numerator. Consequently, the measurement update converges to something meaningful. In practise, it is recommended to add at least a machine epsilon to the denominator as it always is a non-negative value. The partial derivatives of velocity and acceleration are zero, thus:

$$\mathbf{h}' = \begin{bmatrix} \frac{\partial r_1}{\partial x_N} & \frac{\partial r_1}{\partial x_E} & \frac{\partial r_1}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial r_n}{\partial x_N} & \frac{\partial r_n}{\partial x_E} & \frac{\partial r_n}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.11}$$

Furthermore, it is assumed that noise related to the distance measurements are independent. Consequently, the resulting noise values for each measurement will be along the diagonal of the measurement noise matrix $\mathbf{R}$:

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix} \tag{4.12}$$

where σ_i^2 represents the measurement variance of each anchor.

4.3.4 Sensor Fusion Approach to Filtering

Sensor fusion describes how different data sets or estimates can be combined to yield one single estimate (Gustafsson 2010). The purpose is to improve reliability, precision, robustness, dynamic capability, quality and output rate of the estimation. In fact, the idea is already utilized by using more than minimum required amount of range measuring sensors in our description in the system measurement equations. Remember that the trilateration principle requires only 3 range measurements to determine the targets position in 3D. Rest of the sensors are used to improve the precision and reliability.

In the next sections extra sensors used in this thesis and how they can be integrated into the system measurement equation will be described. Moreover, some of the implementation issues and techniques will be briefly discussed.

Altitude Sensor

An extra height sensor is considered to improve the position estimate in down position in the NED-frame. Due to high deviations in the range measurements, the anchors are required to be set far from each other to increase estimation precision. However, that is not always possible in indoor environments - the roof height is limited. Moreover, the anchors should actually be placed near the roof to reduce multipath. Consequently, the sensors are placed near the roof, often on a plane. Hence, two position solutions are available for each estimate due to trilateration principle. Thereby, the EKF might converge to the wrong solution in the down frame if the initial conditions are divergent.The problem is discussed and shown in simulation Section 4.4.

However, low cost ultrasonic range finder or an altimeter can be used to improve the estimation in the down reference. Hence, given that the initial down reference is the floor, the measurement matrix $\mathbf{H}$, and measurement noise matrix $\mathbf{R}_k$ becomes

$$\mathbf{h}' = \begin{bmatrix} \frac{\partial r_1}{\partial x_N} & \frac{\partial r_1}{\partial x_E} & \frac{\partial r_1}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial r_n}{\partial x_N} & \frac{\partial r_n}{\partial x_E} & \frac{\partial r_n}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial r_{height}}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.13}$$

where:

$$\frac{\partial r_{height}}{\partial x_D} = \frac{\partial x_D}{\partial x_D} = 1, \tag{4.14}$$

$$\mathbf{R}_k = \text{diag}[\sigma_{1k}^2, \sigma_{2k}^2, \ldots \sigma_{nk}^2, \sigma_{height}^2] \tag{4.15}$$

and where σ_{height} is the measurement variance of the height sensor. Note that if σ_{height} is smaller than σ_i, the estimator will trust the height sensor measurement more, which will increase the precision in down estimation.

Inertial Measurement Unit (IMU)

Remember that the multirotor autopilot is based on Pixhawk, which estimates the attitude of the system utilizing its own IMU. The IMU is mostly based on low cost sensors, which is characterized by high noise levels and time varying additive biases. (Mahony, Hamel, and Pflimlin 2008) shows how to obtain good attitude observer design using low-cost sensors. However, it is not optimal for slow moving objects and is time consuming. Therefore designing a separate attitude estimator is seen unnecessary in this thesis. The Pixhawks attitude estimation is satisfactory and therefore the attitude estimation is not discussed. However, the raw IMU accelerometer data together with the estimated attitude from Pixhawk can be used to improve the position estimate.

Measurements from a 3-axis accelerometer can be expressed as (Fossen 2011),

$$\mathbf{a}^b_{imu} = \mathbf{R}^{\mathbf{b}}_{\mathbf{n}}(\mathbf{\Theta_{nb}})(\dot{\mathbf{v}}^n_{m/n} + \mathbf{g}^n) + \mathbf{b}^b_{acc} + \boldsymbol{\omega}^b_{acc} \tag{4.16}$$

where $\mathbf{\Theta} = [\phi, \theta, \psi]^T$ is a vector of Euler angles and $\mathbf{R}^{\mathbf{b}}_{\mathbf{n}}(\boldsymbol{\theta}_{\mathbf{nb}})$ is the rotation matrix between NED and BODY frame shown in Equation (4.17), $\dot{\mathbf{v}}^n_{m/n}$ is the body fixed velocity located at $\mathbf{r}^b_m := \mathbf{r}^b_{m/b} = [x_m, y_m, z_m]^T$ with respect to the BODY-frame coordinate origin CO, $\mathbf{b}^b_{acc}$ is the accelerometer bias and $\boldsymbol{\omega}^b_{acc}$ is the additive zero-mean sensor measurement noise.

$$\mathbf{R}^{\mathbf{b}}_{\mathbf{n}}(\mathbf{\Theta_{nb}}) = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)c(\theta) & -s(\theta) \\ -s(\psi)c(\phi) + c(\psi)s(\theta)s(\phi) & c(\psi)c(\phi) + s(\phi)s(\theta)s(\psi) & c(\theta)s(\phi) \\ s(\psi)s(\phi) + c(\psi)c(\phi)s(\theta) & -c(\psi)s(\phi) + s(\theta)s(\psi)c(\phi) & c(\theta)c(\phi) \end{bmatrix} \tag{4.17}$$

Equation (4.16) can then be expressed as linear acceleration in the NED frame with,

$$\dot{\mathbf{v}}^n_{m/n} = \mathbf{R}^{\mathbf{b}}_{\mathbf{n}}(\mathbf{\Theta_{nb}})^T[\mathbf{a}^b_{imu} - \mathbf{b}^b_{acc} - \boldsymbol{\omega}^b_{acc}] - \mathbf{g}^n \tag{4.18}$$

Furthermore, the CO is located at the IMU. Hence the accelerometer measurements,

$$\dot{\mathbf{v}}^n_{m/n} = [r_{a_N}, r_{a_E}, r_{a_D}]^T \tag{4.19}$$

corresponds to measuring states a_N, a_E and a_D in system state Equation (4.2).

It is important to note that the linear acceleration is often expressed as (Hol 2011),

$$\dot{\mathbf{v}}^n_{m/n} = \mathbf{R}^{\mathbf{b}}_{\mathbf{n}}(\mathbf{\Theta_{nb}})^T[\mathbf{a}^b_{imu} - \mathbf{b}^b_{acc} - \boldsymbol{\omega}^b_{acc}] - \mathbf{g}^n - 2\boldsymbol{\omega}^{\mathbf{n}}_{\mathbf{ie}} \times \mathbf{v}^n_{m/n} \tag{4.20}$$

in the literature, where $2\boldsymbol{\omega}^{\mathbf{n}}_{\mathbf{ie}} \times \mathbf{v}^n_{m/n}$ is the Coriolis acceleration. However, a local NED reference system and low acceleration is used in this thesis. Consequently the Coriolis term can be neglected. Thus the EKF measurement equation can now be expressed as

$$\mathbf{h}' = \begin{bmatrix} \frac{\partial r_1}{\partial x_N} & \frac{\partial r_1}{\partial x_E} & \frac{\partial r_1}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial r_n}{\partial x_N} & \frac{\partial r_n}{\partial x_E} & \frac{\partial r_n}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial r_{height}}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial r_{a_N}}{\partial x_N} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial r_{a_E}}{\partial x_E} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial r_{a_D}}{\partial x_D} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.21}$$

where

$$\frac{\partial r_{a_N}}{\partial x_N} = \frac{\partial r_{a_E}}{\partial x_E} = \frac{\partial r_{a_D}}{\partial x_D} = 1\mathbf{R} = diag[\sigma^2_{1k}, \sigma^2_{2k}, \ldots \sigma^2_{nk}, \sigma^2_{height}, \sigma^2_{a_N}, \sigma^2_{a_E}, \sigma^2_{a_D}]\mathbf{h} = [r_{1k}, r_{2k}, \ldots r_{nk}, r_{height}, r_{a_N}, r_{a_E}, \tag{4.22}$$

,

and where σ_{a_N}, σ_{a_E} and σ_{a_D}, is the measurement variance of the accelerometer in NED respectively.

4.3.5 Sensor Integration - Practical Considerations

Integration of INS or IMU offers many advantages over stand-alone systems. New sensors can be added to the system in several ways. (Phillips and Schmidt 1996) discusses some of the GPS/INS integration methods, such as uncoupled, loose, tight and deep integration. The cases in this thesis are similar, but some implementation issues need to be addressed before implementation. For instance, the IMU utilized in the estimator is in fact inside the autopilot. The calculations are conducted using DUNE in the BBB, and sent back to the autopilot. Consequently, unnecessary delays are introduced into the system. Moreover, the frequency of accelerometer data from Pixhawk is very low, about 25 Hz. Therefore, some of the integration possibilities, their strengths and weaknesses will be briefly discussed here to better understand the system architecture with the available sensors.

Uncoupled Navigation Systems

The uncoupled architecture is the simplest, fastest and cheapest method of integration (Phillips and Schmidt 1996). It has high degree of redundancy and tolerance to system failure in the subsystems. In addition, new sensors can be easily added to the system. On the other hand, the IMU output will drift after some time. Thus the validity of the filter models together with the performance will decrease. The system design is shown in Figure 4.11

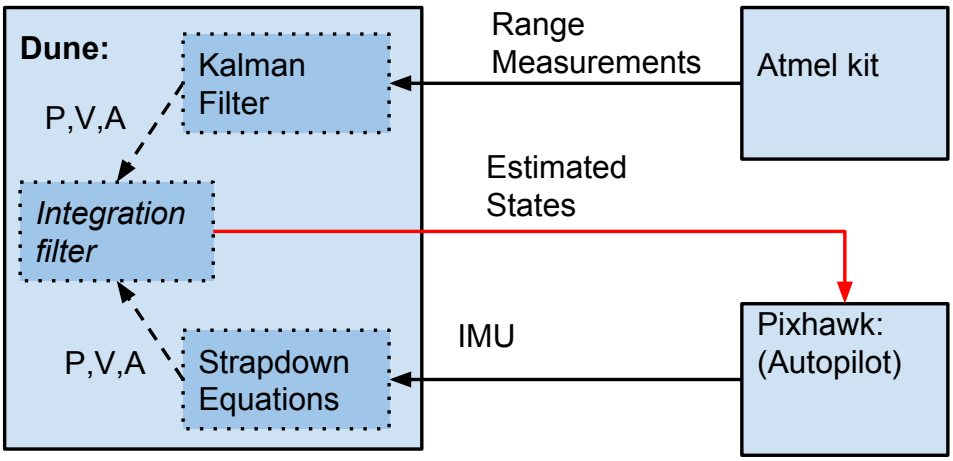


Figure 4.11: Uncoupled IMU/positioning system integration. The IMU from the Pixhawk and range measurements from Atmel is used separately to estimate Position (P), Velocity(V) and Acceleration(A), and then integrated before calculating an integrated estimation.

Loosely Coupled Integration

In the loosely coupled integration, IMU measurements can be used as feedback. The acceleration information from the IMU can be sent the positioning filter to improve the smoothing properties of the filter. In case of GPS positioning, this feedback can be used to decrease reacquisition time after GPS dropout, and the GPS data can be used to reset IMU to ensure freshly calibrated INS (Vik and Fossen 2000). However, in the case of local positioning based on range measurement and trilateration, these advantages cannot be utilized effectively. Figure 4.12 shows the system design.

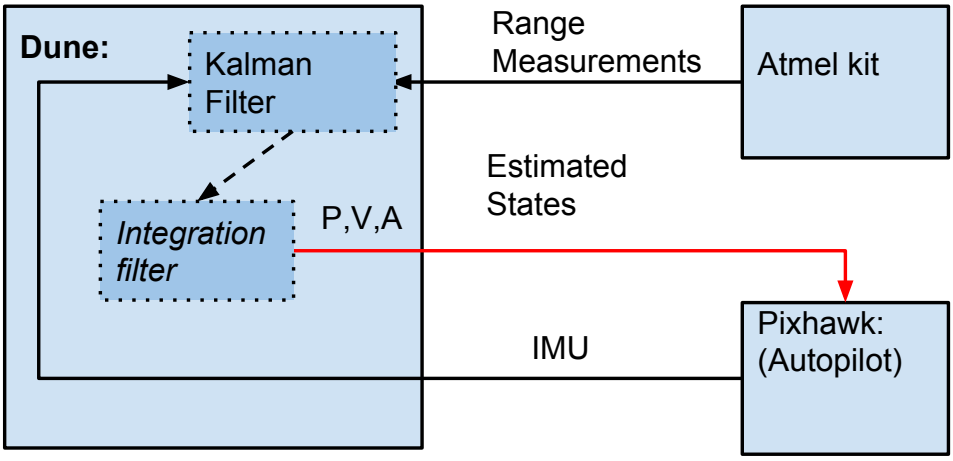


Figure 4.12: Loosely coupled IMU/positioning system integration. The range measurements from Atmel kit is used to estimate Position (P), Velocity(V) and Acceleration(A), while the acceleration measurement from the IMU is used as feedback to Acceleration estimation.

Tightly Coupled Integration

The tightly coupled integration is very similar to the loosely coupled integration. The difference is, instead of using the IMU measurements as feedback to smooth the estimates, they are used directly in the calculation of the estimates. In fact, all available raw measurement data are fused in the same filter to make a good estimate. The main advantage is that the filter can make an estimate even when the minimum required range measurements are not available. Moreover, the solution is more accurate in this method, given that the measurements are uncorrelated (Vik and Fossen 2000). On the other hand, the modularity is usually decreased, while the complexity is increased. However, in this thesis only the accelerometer data from the IMU will be used, hence the complexity is not seen as a problem. Therefore, this method has been implemented since it achieves the most accurate estimates. The system design is shown in Figure 4.13.

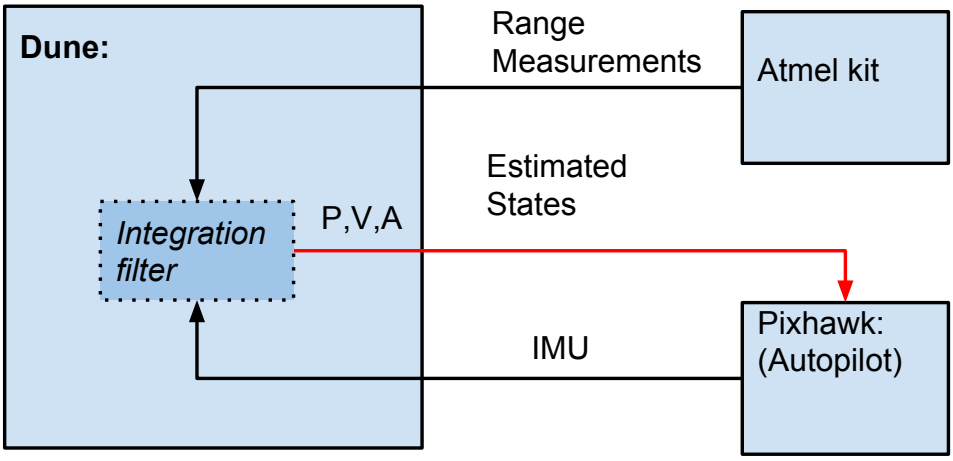


Figure 4.13: Tightly coupled IMU/positioning system integration. The range measurements from Atmel kit and the IMU data is used together to estimate Position (P), Velocity(V) and Acceleration(A).

Pixhawk Firmware Changes

Another method considered in the implementation of an indoor positioning system is by by-passing the DUNE platform. Since the IMU data is sent from the Pixhawk, the range measurements from the Atmel can be sent directly to The Pixhawk. Thus the delay between the sensor data and estimations are reduced. However, the Pixhawk firmware is regularly updated by the producer. Consequently, the code needs to be merged regularly, which is neither practical nor convenient. Moreover, the modularity is completely lost in a such solution. Therefore, this method is not utilized. The system design is shown in Figure 4.14.

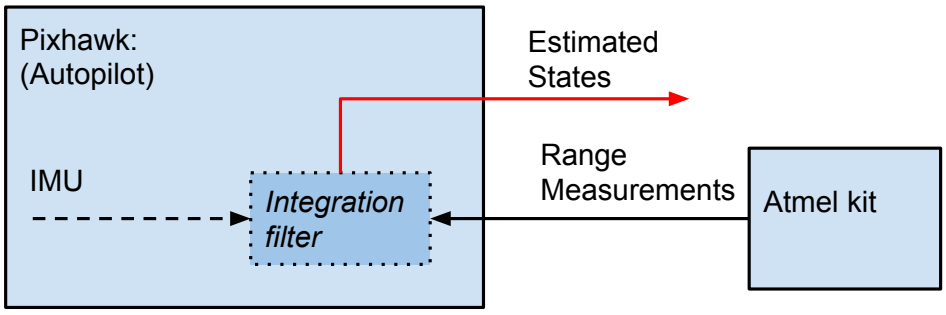


Figure 4.14: Tightly coupled IMU/positioning system integration in Pixhawk. By by-passing DUNE, the system can directly be designed in Pixhawk firmware. This design reduces the delay of the IMU, but is problematic since the Pixhawk firmware is updated regularly by the producer.

4.3.6 Outlier Detection and Rejection Method

This section is based on the outlier detection and rejection theory presented in Section 2.1.3. Here the implementation and values used in this thesis will be presented. The results will

be discussed in Section 4.4, 4.5 and 4.6. Both the standard hypothesis test inside the iterated EKF and the Grubbs test outside the EKF loop, as a signal quality check, will be presented.

Standard Hypothesis Test

Remember from Section 2.1.3 that the iterated test static is based on:

$$T(y_k^i) = (y_k^i - H_k^i \hat{x}_{k|k-1})^T (H_k^i P_{k|k-1} (H_k^i)^T + R_k^i)^{-1} (y_k^i - H_k^i \hat{x}_{k|k-1}) \sim \chi^2_{n_y^i}$$

where y_k^i represents the measurement from sensor i at iteration k. The confidence α is set to $\alpha = 0.05$, corresponding to $\chi^2 = 18.548$ with 6 degrees of freedom, which is the amount of measurement sensors used generally in this thesis. If $T(y_k^i) > \chi^2_{\alpha,n_y^i}$, the measurements are rejected as an outlier. The rejection is conducted by setting the corresponding R to infinity, or more conviniently by setting the corresponding column in the Kalman gain matrix K to zero.

Grubbs Test

The theory behind Grubbs test is discussed in 2.1.3. To improve the performance of the position estimation, the signal can be analyzed before feeding it to the EKF. Possible or obvious outliers can easily be flagged and removed from the measurement set before feeding the signal to the EKF algorithm. This method works well in the case of static target. In the dynamic case however, it is not as reliable as the standard hypothesis test mentioned above. This is mainly because it is based on the measurement mean, which can not be precisely calculated on a moving target. Especially if the measurement frequency is low. Therefore, the Grubbs test is only utilized in static cases.

4.3.7 Weighted Covariance - Fuzzy Logic Adaptive System (FLAS)

The Kalman Filter is an optimal estimator based on several assumptions. These assumptions are discussed in great detail in Section 2.1. If the assumptions are not fulfilled, the filter is not optimal and can start to diverge. The divergence problem is discussed in Section 2.1.4.

One of the assumptions for optimal estimation in KF is the assumption of zero mean white noise (Schlee, Standish, and Toda 1967). However, the initial experiments of the Atmel kit has shown that this assumption is not fulfilled due to outliers and multipath generation. Consequently, frequent outliers or/and multipaths can cause the KF to diverge. Moreover, the Atmel kit measurements are sensitive to its environments. Therefore, their deviation can fluctuate drastically. Consequently, setting a constant covariance matrix R in Table 2.2 can cause extensive calculation error.

Therefore, the covariance matrix R should be tuned adaptively to excite the system correctly. The fact that a perfectly tuned model should have zero-mean white noise process can be utilized. The weighting process is discussed in Section 2.1.4, but the implemented version will be presented here for readers convenience.

The covariance matrix R can be simply weighted as shown in Equation (2.16) or,

$$\mathbf{R}_k = \mathbf{R}_k \cdot \alpha \tag{4.23}$$

where α is a weighting parameter. The weighting parameter can be calculated using Fuzzy Logic Adaptive Controller (FLAC). Remember that the residual is the term $(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}))$ in state estimation update in Table 2.2. If the residual is not zero-mean white noise, the filter is not performing optimally (Sasiadek and Wang 1999). Hence using the absolute value of the residuals, a membership function can be designed. By analyzing the residual term, the input membership function shown in Figure 2.1 is found to be appropriate. The designed output membership function is shown in Figure 4.15. Note that the output membership constructed is actually a scaled version of the input membership functions, using the maximum degree of truth to decide the weighting parameter α. However, it is important to realize that this is just a tuning method, and can be tuned to optimize the system. The results from the FLAS are discussed in the Sections 4.4, 4.5 and 4.6.

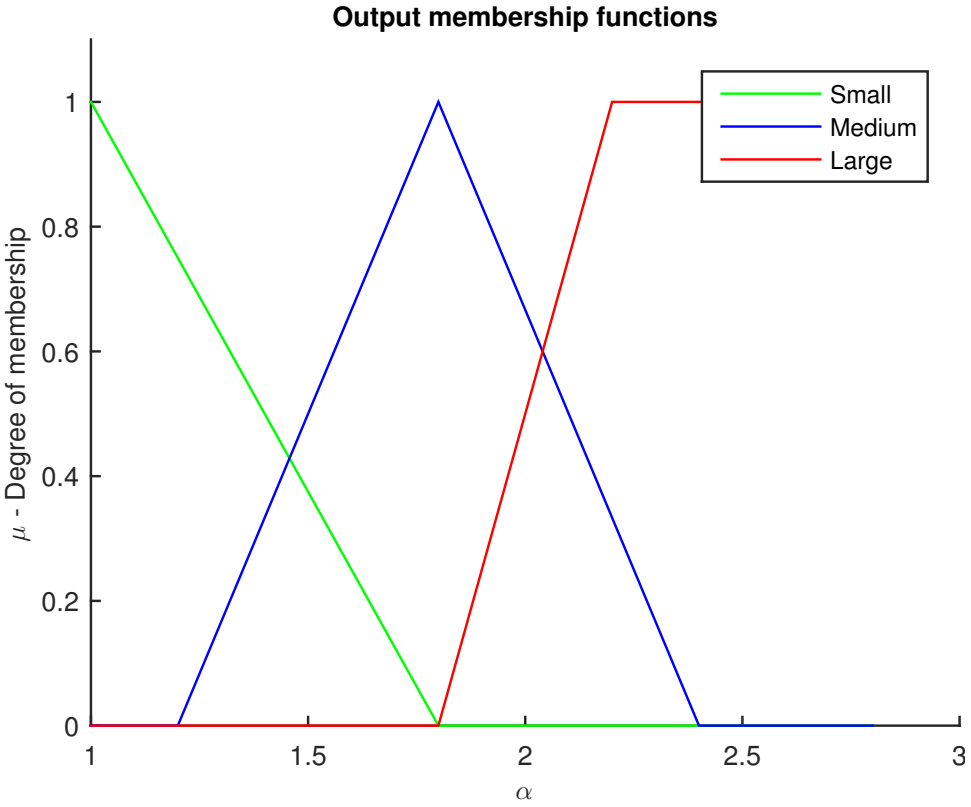


Figure 4.15: Output membership function used in EKF with FLAS. The covariance R is excited by multiplying it with α, which is.

4.4 Simulation of Positioning Estimator

The position estimation algorithms is studied in MATLAB simulator to begin with. The multirotor dynamics are simulated using the multirotor dynamics described in (Klausen, Fossen, and Johansen 2014), and wil not be discussed in this thesis. The interested reader is recommended to consult (Klausen, Fossen, and Johansen 2014).

4.4.1 Simulated Trajectory

The hexacopter was set to follow the following 5 Waypoint (WP)s in NED frame, starting from origo:

$$
\begin{aligned}
WP_1 &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \\
WP_2 &= \begin{bmatrix} 1 & 10 & -5 \end{bmatrix}^T \\
WP_3 &= \begin{bmatrix} 11 & 10 & 0 \end{bmatrix}^T \\
WP_4 &= \begin{bmatrix} 11 & 0 & -5 \end{bmatrix}^T \\
WP_5 &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T
\end{aligned}
$$

The tracked trajectory of the simulated multirotor is shown in Figure 4.16. The simulation time was set to 60 seconds, but the copter finished the trajectory in about 30 seconds. Note that the simulated anchors are also shown in Figure 4.16. Each anchor measures its distance to the multirotor during the simulation with 100Hz measurement frequency.

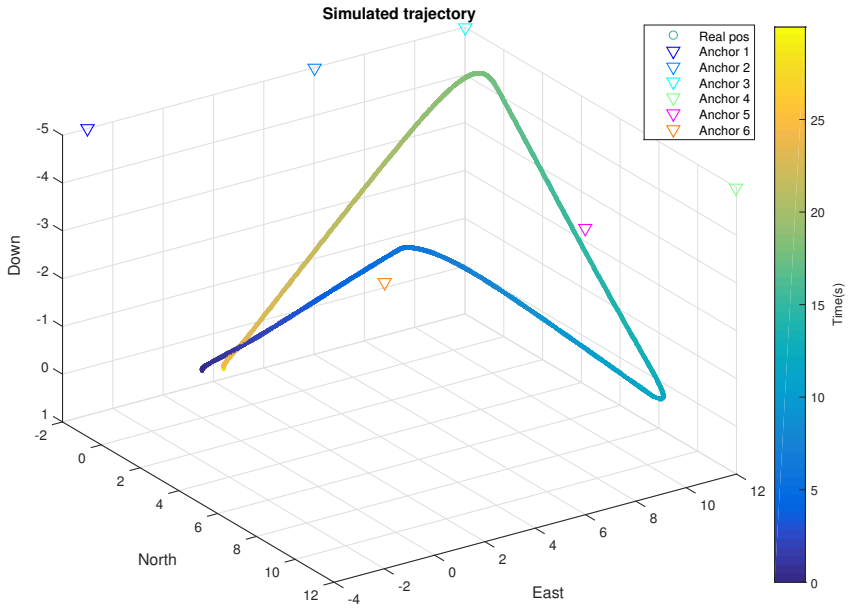


Figure 4.16: The tracked trajectory of the simulated multirotor. The simulated anchors are placed as shown in the figure and measure range to the copter with 100Hz measurement frequency.

4.4.2 Estimator Performance

In this section the results from the EKF algorithm discussed in Section 4.3 will be presented. Throughout the section, the same trajectory explained in Section 4.4.1 is used. The anchors are simulated with a measurement frequency of 100Hz. Each anchor measure their distance

to the target. The measurement standard deviation and percentage of outliers can be set in the developed simulator.

The following subsections will discuss different scenarios, EKF algorithms and system models. The result from this section will be used to determine the best implementation techniques for future in-flight experiments.

The Standard EKF Algorithm With Perfect Measurements

In the first simulation, no outliers and low zero mean standard deviation is used. The goal is to determine performance of the P and the PVA model.

The resulting range measurements and the error of the estimated position using EKF with the P model is shown in Figure 4.17 and Figure 4.18. The error between the real position and the estimated position has a standard deviation of about 20 cm.

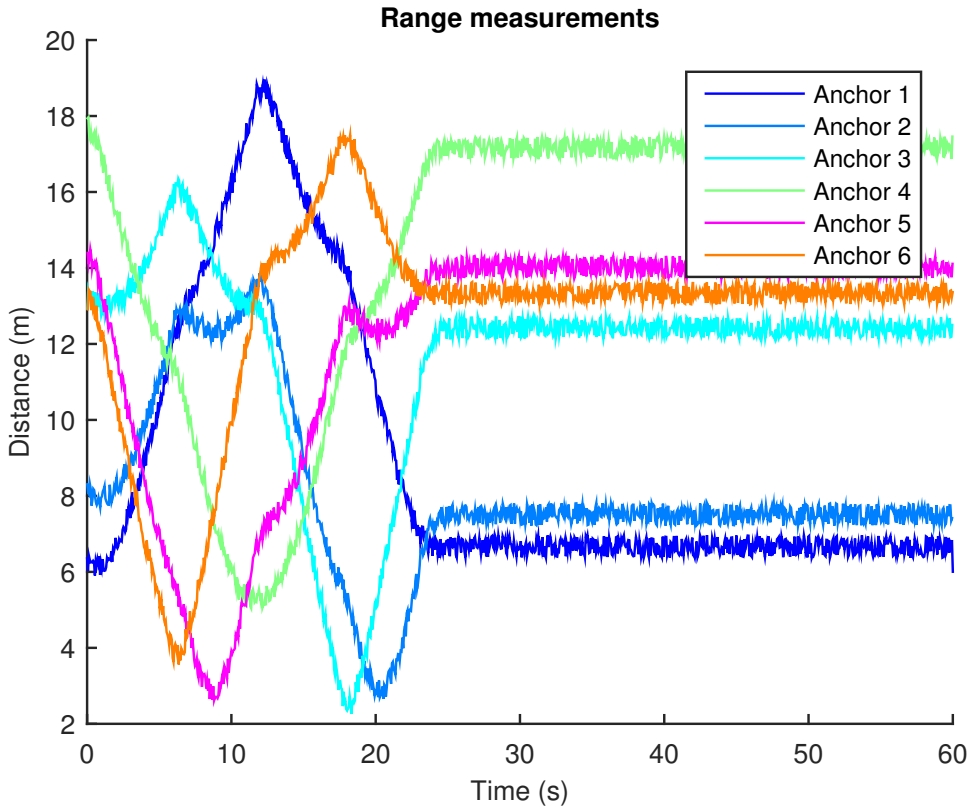


Figure 4.17: Simulated range measurements from each anchor. Note that there is no outliers and a standard deviation of 30 cm.

The same experiments are conducted using the PVA model and the error between the true position and the estimated position is shown in Figure 4.19. Note that standard deviation of the error in the North and East is now only 7 cm. It is a reduction of 35% compared to the P model solution.

However, the Down position estimation contains a gross error. The error is due to wrong estimation: since all the anchors are placed on the same height on a plane, the down estimation can have 2 solutions due to principle of trilateration. Consequently, if the target

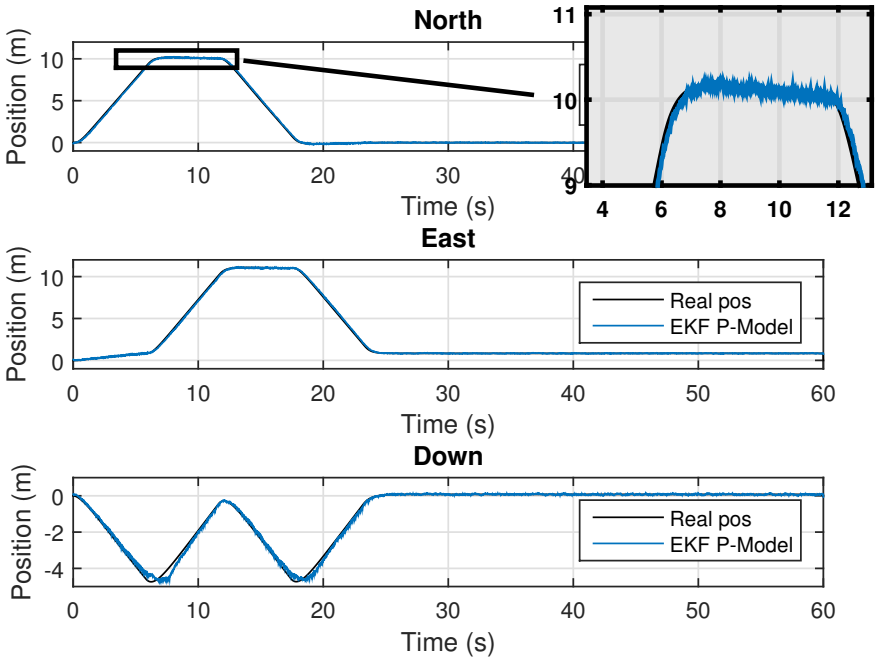


Figure 4.18: Estimated position using standard EKF and the P model. The error has a standard deviation of about 20 cm.

dynamics are fast and the measurement frequency is low, the model might not be able to follow the correct trajectory. The difference between the estimated and real velocity is shown in Figure 4.20.

The problem can be solved simply by setting the anchors in different heights, hence reducing the problem to have only one solution. While this is an easy fix in the simulator and in outdoors experiments, the problem remains for the indoor case. Since the measurement deviation can in reality be up to 80 cm, the height difference between the anchors should be at least 80 cm. However, the performance of the anchors are drastically reduced if they are not placed near the roof.

Other solutions are to increase the measurement frequency, or integrate accelerometer and altimeter data to correct the estimations, or excite the system more. Especially if there is a deviation between the model estimates and actual measurements. The latter will be discussed in this Section, while the others are discussed in Section 4.5.

Filter Monitoring - Measurements With Outliers

In this section the effect of outliers and erroneously covariance matrix R tuning will be discussed. It is important to emphasize that the covariance matrix R is not necessarily tuned wrong. The problem is rather that the sensor measurements are sensitive to their environments. Therefore, a constant covariance matrix R would be erroneous. The simulation was conducted using the P model with 20% outliers. Moreover, the covariance matrix R was set to 20% lower than the actual deviation. The results, with standard EKF, EKF with outlier re-

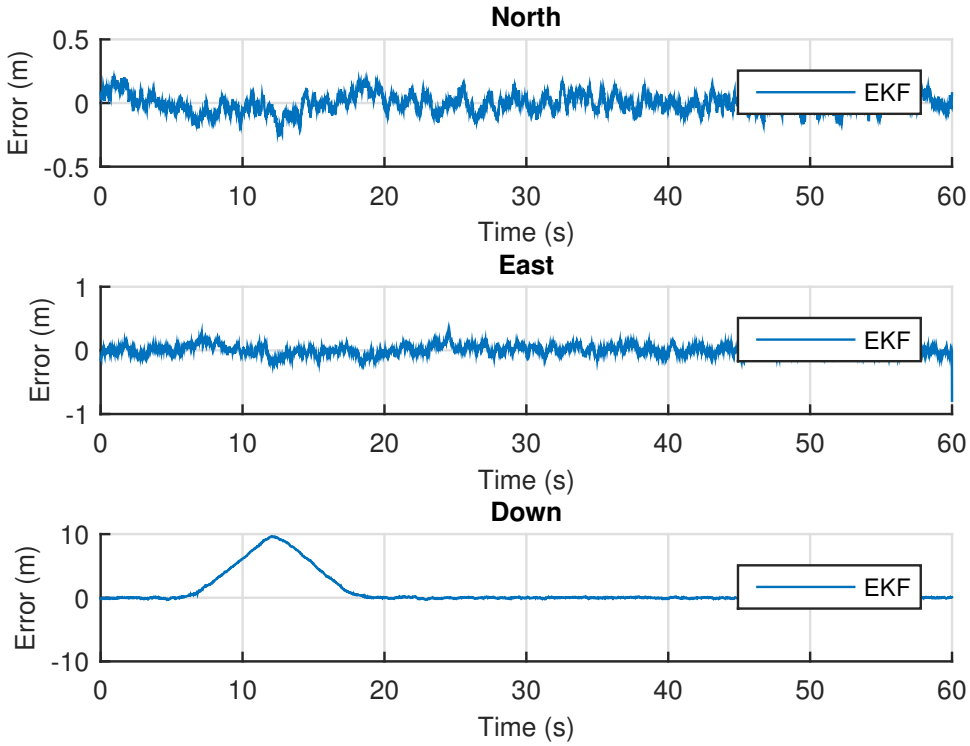


Figure 4.19: Error in estimated position using standard EKF and the P model. The error has a standard deviation of about 7 cm in the north and east position. In the Down position however, it is not able to track the position correctly.

jection and the EKF with FLAS is shown in Figure 4.21. The resulting range measurements are shown in Figure 4.22.

Figure 4.21 shows clearly that the EKF with outlier detection is closest to real value and smoothest. It also shows that the EKF with FLAS is better than standard EKF, but not as smooth as the EKF with outlier rejection. However, the problem of wrong solution or divergence is eliminated using the FLAS. Note how the standard EKF and EKF with outlier detection goes to wrong solution in the down estimation. However, the adaptive EKF based on FLAS is capable of detecting measurement and model mismatch and excite the system enough to not trust the measurements as much as the standard EKF does.

In addition, series of outliers may cause the outlier detection and rejection process to behave poorly. It will start rejecting too many measurements as outliers and may start to diverge. The problem is discussed in (Gustafsson 2010), where the author suggests to use a sliding window over a fixed time horizon to select just the last set of measurements. However, the problem related to environmental sensitivity, hence erroneous measurement covariance R, and multipath problems remain unsolved. Remember from Equation 2.14 that the outlier rejection is based on the measurement and the expected standard deviation. If the expected standard deviation, thereby R is erroneous, the outlier detection decision process will be based on wrong premises. Consequently, a better solution is to excite the covariance matrix R correctly. Therefore, the FLAC is suggested in this thesis to adaptively adjust R to improve the decision process and the estimation.

The result is shown in Figure 4.23. While the algorithm with outlier detection and rejec-

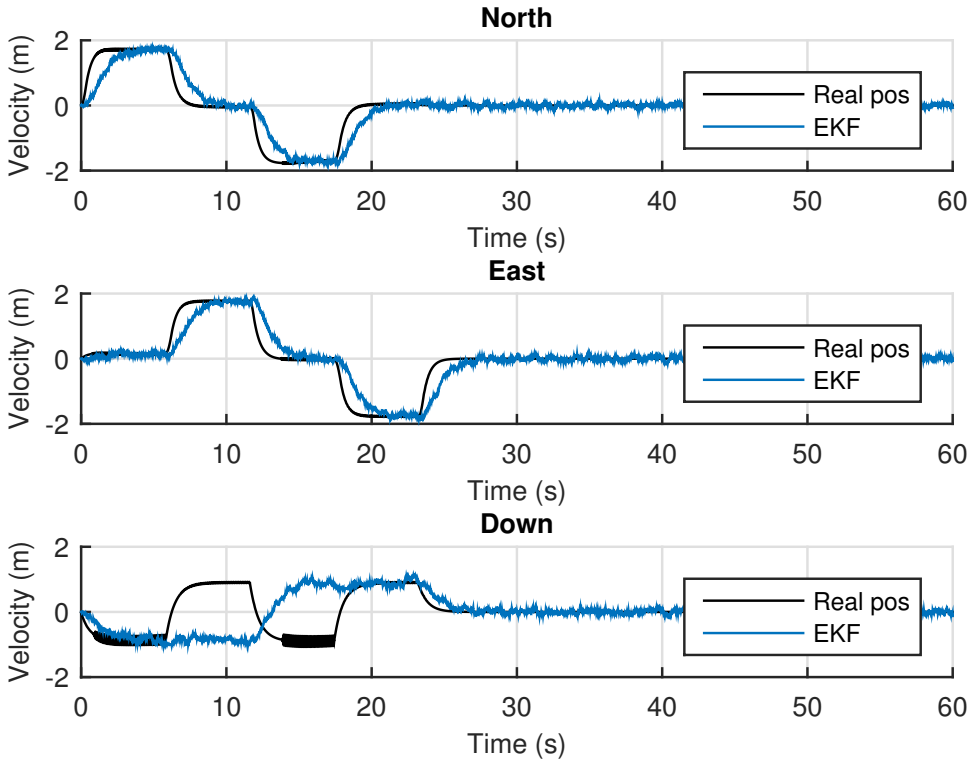


Figure 4.20: Difference between estimated and real velocity using standard EKF and the P model.

tion is converging to wrong solution in the down frame, the algorithm with both FLAS and outlier detection is capable of going to the correct solution. Consequently, it has the advantage of the outlier detection and rejection, smoothness and adaptively adjusting R using FLAC.

Extended Kalman Filter (EKF) - Different Covariances

It is important to note that the EKF filter used in this thesis is based on (Gustafsson 2010). However, the construction of the error covariance differ in the literature. (Fossen 2011) uses the Jordan form for instance,

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{h}'(\hat{\mathbf{x}}_{k|k-1}))\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k \mathbf{h}'(\hat{\mathbf{x}}_{k|k-1}))^T + \mathbf{K}_k \mathbf{R} \mathbf{K}_k^T \tag{4.24}$$

which is explained in Section 2.1.2. The error covariance in (Fossen 2011) is in the Jordan form with a nice quadratic form, which yields positive-definite matrix. However, the symmetry is not useful in this case and cannot be used in practice. Moreover, the form introduced in Table 2.1 results in better response in cases where the innovation covariance is close to zero (Gustafsson 2010). The difference between these two error covariances are shown in Figure 4.24.

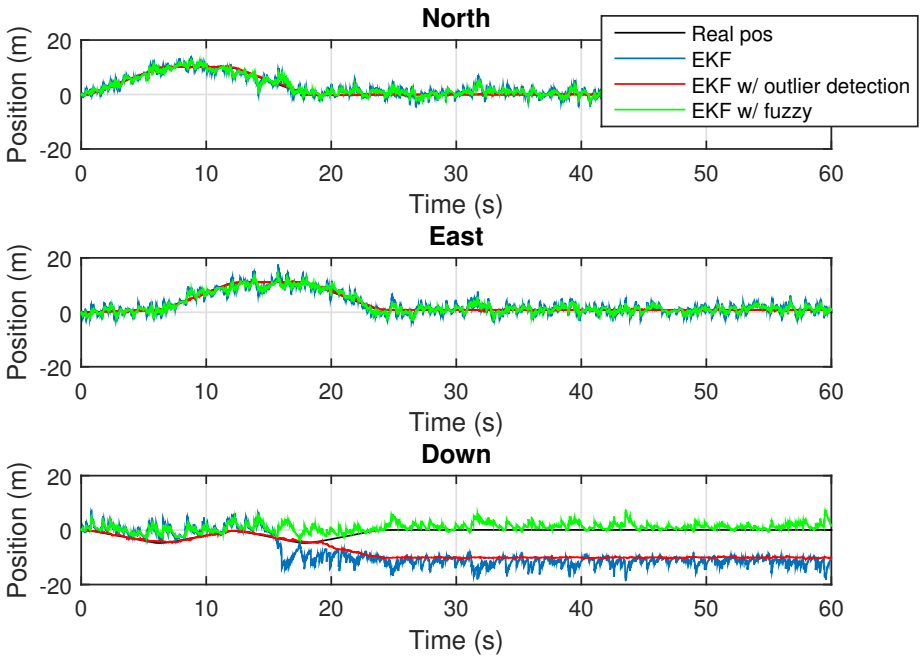


Figure 4.21: Position tracking with simulator using 20% outliers and using 20% lower σ (measurement deviation) than actual measurement deviation. We see that the EKF with FLAS is capable of exciting the system enough to follow correct solution while standard EKF and EKF with outlier detection fails.

4.4.3 Discussion

Findings from the simulator is very helpful for real-world experiments. It has been proven that the Extended Kalman Filter algorithm works satisfactory for position estimation. However, it is optimal only if it is tuned perfectly. Perfect tuning however, requires correct information about the measurements, which is not always possible in real world applications. Especially if the measurements are sensitive to their environments and their performance might decrease during the operation. In addition, the measurements may have outliers, which can result in gross errors or diverging estimations. Consequently an EKF algorithm that deals with both outliers and measurement sensitivity is necessary.

It has been shown that the outlier detection and rejection algorithm introduced in this thesis works properly. However, the decision process of outlier detection is also sensitive to correct information about the measurements. The decision process will behave poorly if the covariance matrix R is erroneous, or series of outliers occur after each other. Therefore, a monitoring algorithm is necessary. Fuzzy Logic Adaptive System is suggested in this thesis to adapt the covariance matrix R, hence improve the decision process of outlier detection. The results are promising, and the algorithm with both outlier detection and FLAS will be used further in this thesis.

Moreover, two system models have been used in the simulations: the Position model (P) and the Position-Velocity-Acceleration model (PVA) Model. Even though the PVA model is supposed to perform better (Shareef and Zhu 2009), it is based on the assumption that the

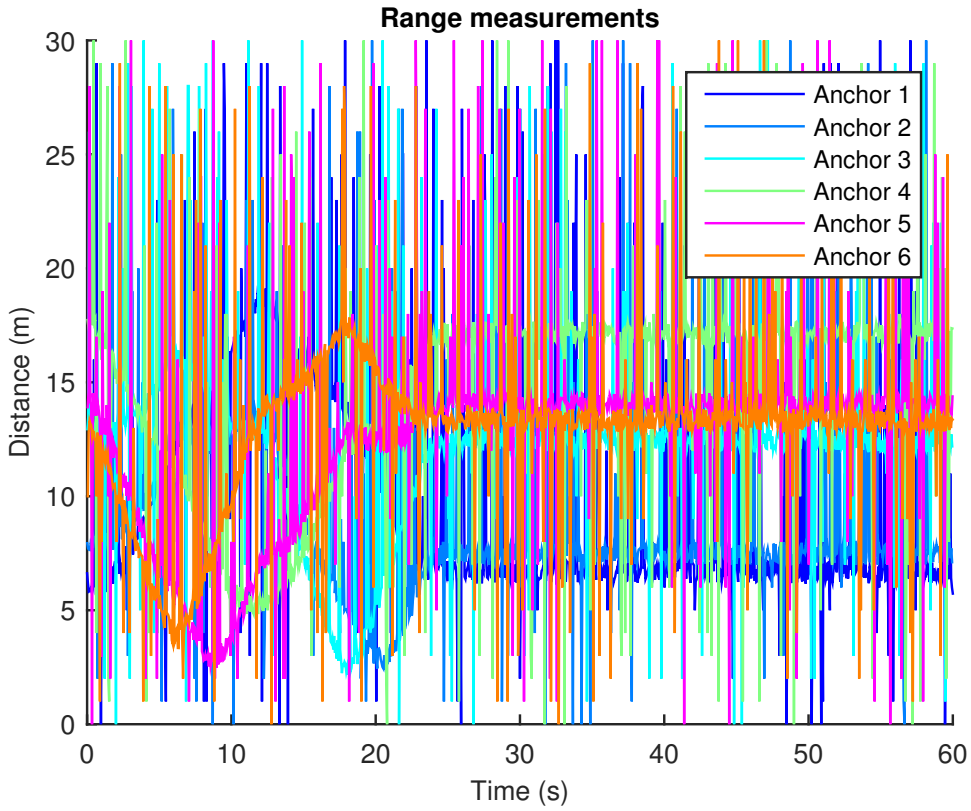


Figure 4.22: Range measurements with 20% outliers.

motion can be regarded as uniformly accelerated rectilinear motion between each sample (Li et al. 2011). In other words, it requires very high measurement frequency. Consequently, the performance is dependent of high frequency of the measurements.

It is worrisome that the PVA model performs poorly, in the sense of estimation performance, even in the simulations. In real-world application, the measurement frequency of the anchors are much lower. That being said, the simulated target had fast dynamics. However, the PVA model enables us to integrate the accelerometer data from the IMU, which can improve the estimation. The result of the integration will be discussed in the following sections.

4.5 Indoor experiments

In this section the indoor experiments will be presented. The indoor experiments are conducted in Taklabben and in Robotlabben at NTNU. While Taklabben has the advantage of being relatively big, the Robotlab is instrumented with OptiTrack motion cameras that is used as a sub-mm precision tracking system. Discussions will include problems regarding low measurement frequencies, integration of low cost IMU data and altimeter sensor.

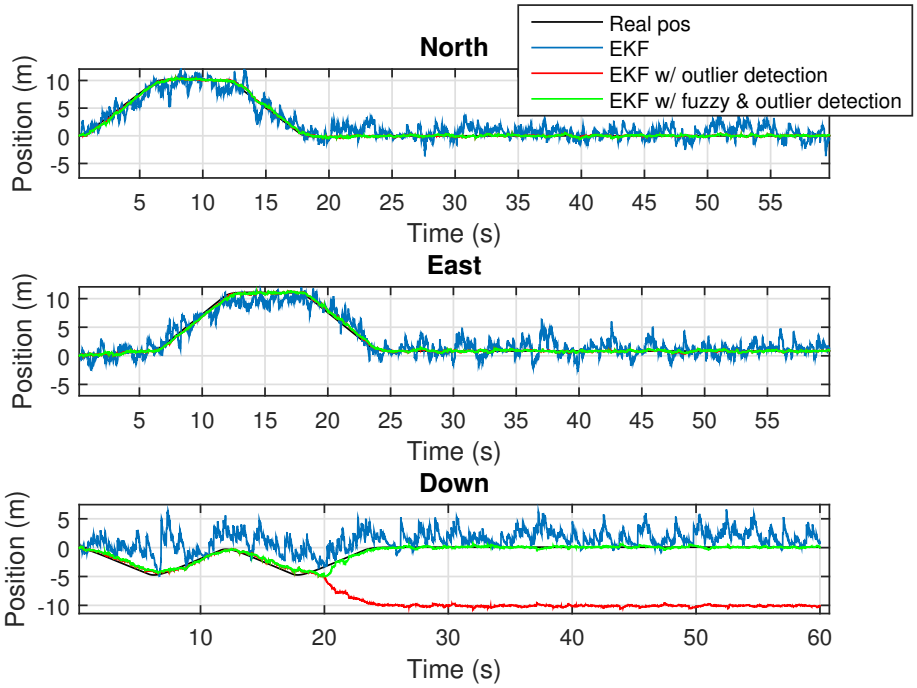


Figure 4.23: Range measurements with 20% outliers. The outlier detection behaves poorly if the R matrix is erroneous. However, using FLAS, the covariance matrix R is adjusted adaptively and improves the decision process of the outlier rejection. Hence the solution with fuzzy logic and outlier detection together works optimally.

4.5.1 Initial Experiments in Taklabben

Taklabben is used for initial experiments. Figure 4.25 shows the hexacopter flight at Taklabben. Six Atmel transceivers are used as anchors during the experiments. The goal of the experiment is to gain initial insight on which models and methods work well in indoor environments with dynamic target. Furthermore, the performance of low cost accelerometer sensor on a low-speed target is investigated. Note that there are no good ways to track the real position of the target at Taklabben. However, a known formation was flown to consider the models performance.

Figure 4.26 shows position estimation using the PVA model on a static target. Note that the outlier detection algorithm performs poorly; it diverges. This is found to be due to poor measurements and erroneous tuning parameters. Reason of the poor measurements are related to targets position. The hexacopter was placed on the ground, where the sensors are known to be prone to multipath. Moreover, the measurement frequency during this experiment was very low. Figure 4.27 shows the corresponding range measurements. The low frequency is later found to be due to interference in the 2.4 GHz band and discussed in Section 4.2.3.

Figure 4.26 also shows that the divergence problem is solved utilizing FLAC to tune the covariance matrix R. Hence the covariance is not tuned erroneously. On the other hand, the problem does not occur using the P model or when the accelerometer is integrated which will be discussed later. The position estimated using the same measurement data, but with P

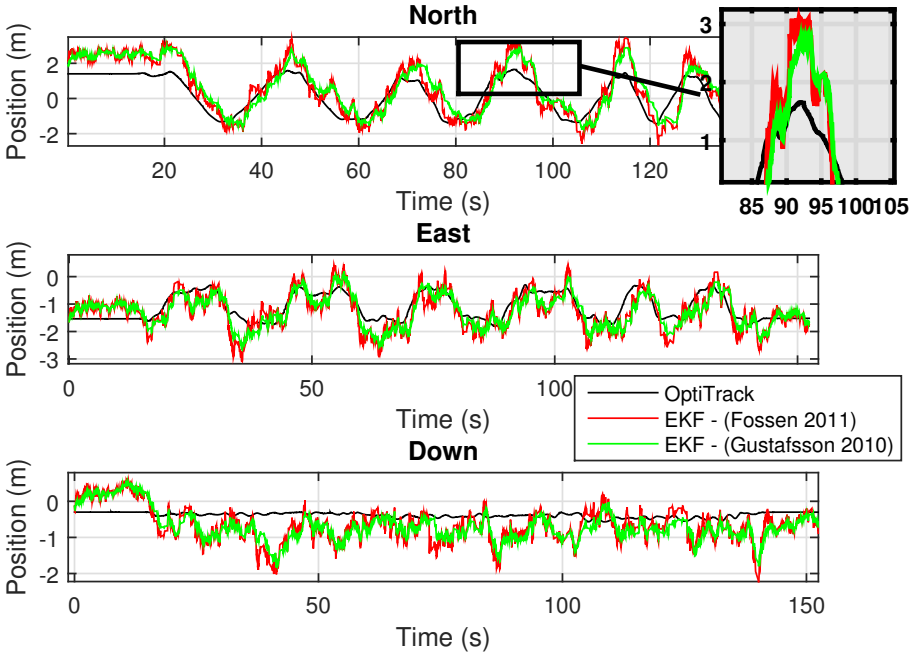


Figure 4.24: Different EKF filters exist in the literature. Plotted is the difference between (Gustafsson 2010) and (Fossen 2011), where the error covariance introduced are different.

model is shown in Figure 4.28. However, this model is too simple and is expected to perform poorly on dynamic targets. Therefore, the measurement frequency should be improved, either by boosting the performance of the Atmel kit or adding IMU measurements to the filter. Later the frequency of the Atmel kit measurements were improved by reducing the Tx power of the PicoStation. However, the reduction of Tx power resulted in limiting the operation range to about 30 meters. Therefore, the Tx reduction is not seen as a permanent solution. The IMU acceleration data integration into EKF will be discussed in the next section.

4.5.2 Integrating Accelerometer Data

Accelerometer data from the Pixhawks IMU is used to improve the EKF estimation. The integration technique is described in detail in Section 4.3.4. The main goal is to find out if the integration can increase the performance of the PVA model.

The hexacopter was piloted manually in a recognizable path; square formation inside the safety net. The safety net shown in Figure 4.25 is about 4mx4m, between 2-6m in local North frame and between 1.5m-5.5m in the local East frame. The resulting position estimation from the flight is shown in Figure 4.29. It is clear that EKF calculates reasonable position estimates. It is also clear that the solution does not diverge with the PVA model when the IMU data is integrated. However, the P model without IMU data still performs better. The PVA model has introduced new outliers as a result of integrating the accelerometer data.

Analysis of the accelerometer data reveals the reason; the accelerometer measurements are very poor. They are affected by copter vibrations during flight. Figure 4.31 shows the re-

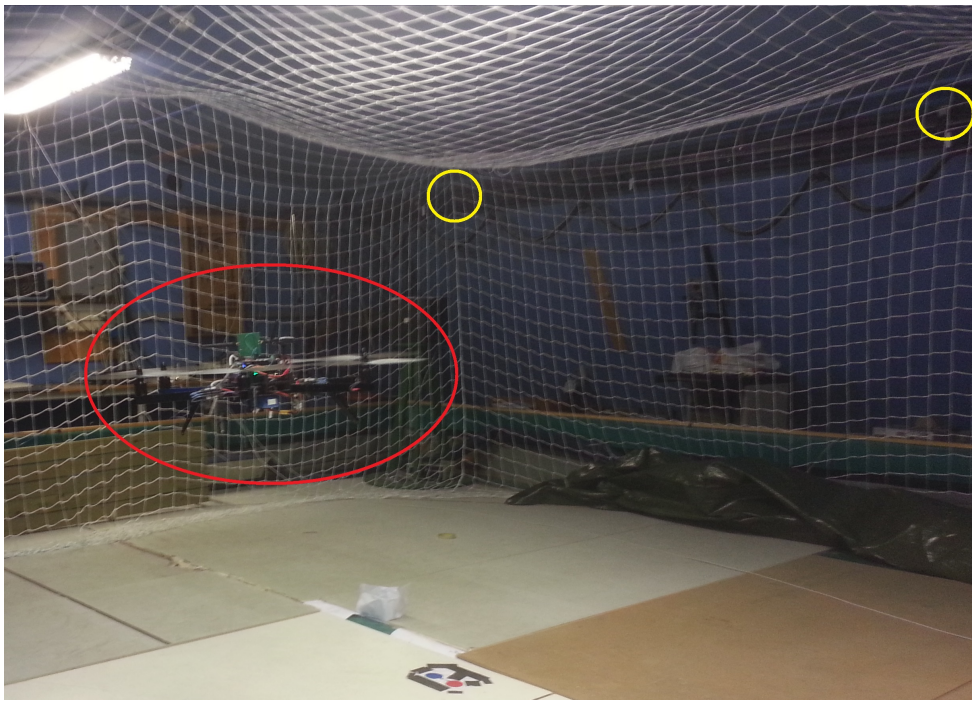


Figure 4.25: The hexacopter flight experiment indoor. The yellow circle shows the positions of two of the Anchors. Note that anchors are mounted near the roof. The net is set for safety during experiments.

sulting accelerometer data from the experiment. The IMU data seems like a perfect white noise. Therefore, the low cost accelerometer of the Pixhawk may not be suitable for indoor operations, especially if the operation speed is low. The result from the outdoor experiments where the operation speed is higher will be discussed in Section 4.6.

4.5.3 Performance Review Using Motion Tracking Technology

After the initial indoor experiments, more extensive analysis indoor is carried out. For this purpose, the OptiTrack motion tracking technology is used. The technology and the products used are explained in Appendix A.7. The system was integrated into the DUNE platform in cooperation with MSc student Andreas Flåten at Engineering Cybernetics, NTNU (2015). Six Atmel anchors were placed near the roof in the Robotlab at NTNU. The motion tracking system was able to track motion in a 2mx2mx2m space in the middle of the room, shown in Figure 4.32. Consequently the operation speed was very low and therefore, the accelerometer integration is not helpful and not be discussed in this section.

Figure 4.33 shows the result of the position estimation using P and PVA models with outlier detection and rejection algorithm with FLAS. It is clear that the P model have better performance than the PVA model. Especially in cases where the operation speed is low.

However, both the estimators can have deviations up to 2 meters. This can be explained by multipath since the target is near the floor during the operation. It has been shown that the Atmel measurements perform poorly near walls, roof and floors. In addition, the position of the anchors were measured by hand, giving another 10-20 cm error in the system. Therefore,

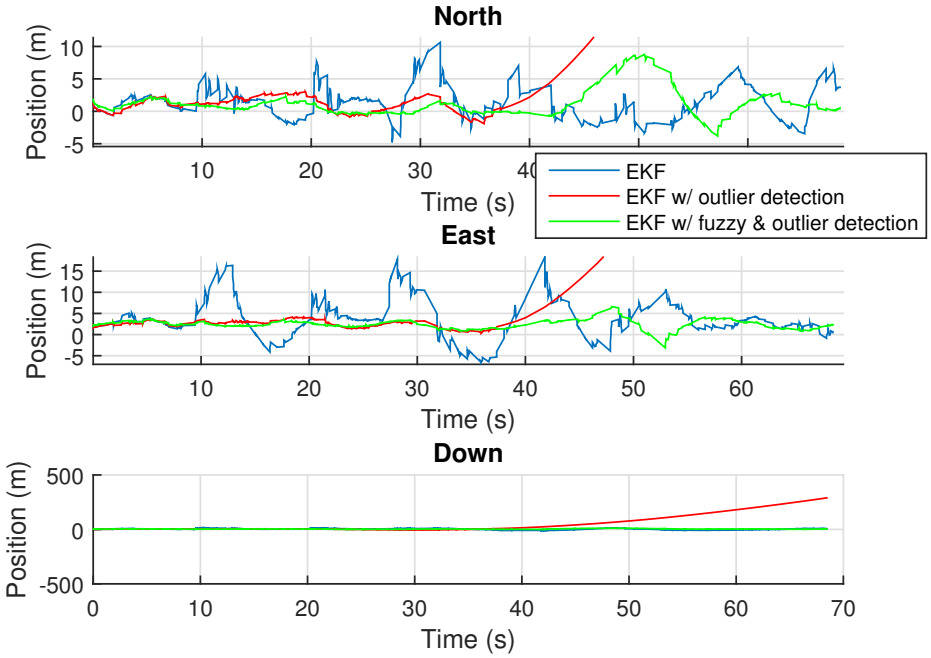


Figure 4.26: Indoor result using PVA model. The outlier rejection algorithms works poorly whitout FLAS due to poor measurements.

a tool to determine if a measurement setup is good should be developed. Such a tool is developed and explained in Section 4.5.4.

4.5.4 Signal Quality Analysis - Tool

Principle of trilateration is shown in Figure 4.10a. With measurement uncertainty, the result becomes as shown in Figure 4.10b. However, in real world cases, multipath and outliers may distort the measurements, and thus the position estimations. Therefore, a tool to determine signal quality is developed.

Before any experiment, a *"trilateration-test"* can be conducted by gathering range measurement data. The mean value of each anchor measurement, with corresponding standard deviations can be used to construct a *"trilateration-model"*. The result is shown in Figure 4.34 for such a test in Robotlab. Note that the static range measurement from the Atmel kits do not show promising results. However, this tool helps to cover which anchors cause the problem. Consequently, the worst anchors positions or antenna configurations can be changed. Thus, the result can be improved. Figure 4.35 shows the results after just changing some of the anchors antenna positions.

4.5.5 Integrating Altimeter Sensor

Initial experiments has shown that the estimation in the Down frame is often poor. This is mainly due to the anchors position. They are often placed near the roof and on a plane.

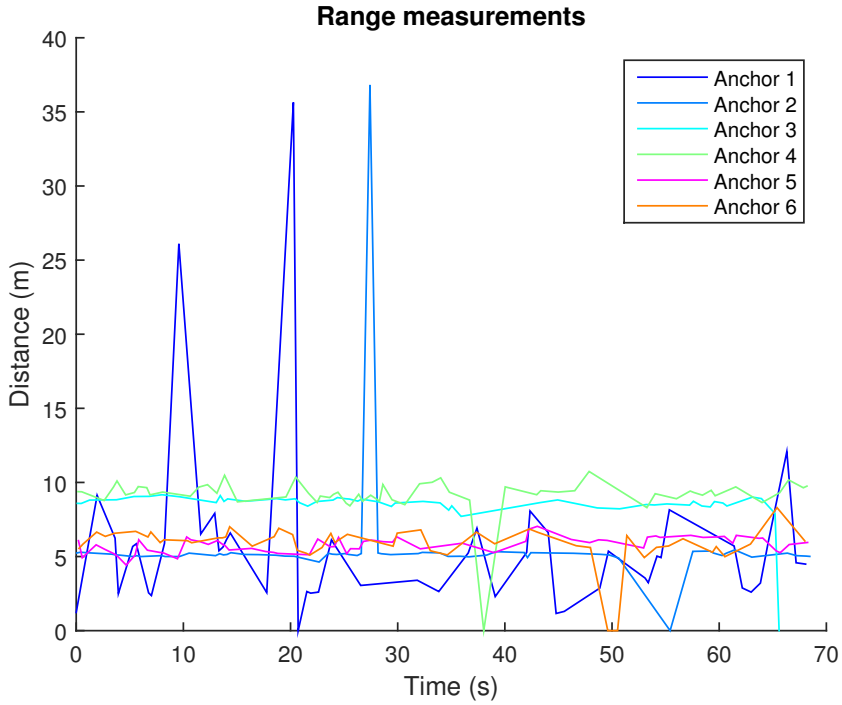


Figure 4.27: Resulting range measurements indoor - static target.

Consequently, the principle of trilateration gives in theory 2 solutions for each estimation. The problem can easily be avoided using an altimeter on the copter. However, in this experiment, we did not add a phisical height sensor. For practical reasons, the result of the OptiTrack solution in the down frame was fed as a height sensor to the EKF as shown in Section 4.3.4.

The result is shown in Figure 4.36. Measurement variance of the altimeter was set much smaller than the measurement variance of the Anchors: $\sigma^2_{height} << \sigma^2_{anchor}$. Consequently, the position estimation in down frame was improved radically. However, the rest of the system is not affected considerably by the improvement in the down frame, which is also shown in Figure 4.36.

4.5.6 Discussion

Experiments with the OptiTrack system has shown that the position estimation with Extended Kalman Filter works well. However, the degree of satisfaction is open for discussion. The estimation precision in worst cases can deviate up to 2 meters. Which is very high considering the operation speed during the experiments were below 1 m/s. Also, low measurement frequency of about 20-25 Hz, sets constraints on how much the operation speed can be increased and utilization of advanced mathematical models. However, the estimations are surprisingly good considering the quality of the measurements, which could be erroneous in even perfect conditions (see Figure 4.34). Moreover, the results seems to be one of the better estimations in the literature, using low-cost sensors based on 2.4 Ghz band utilizing PMU.

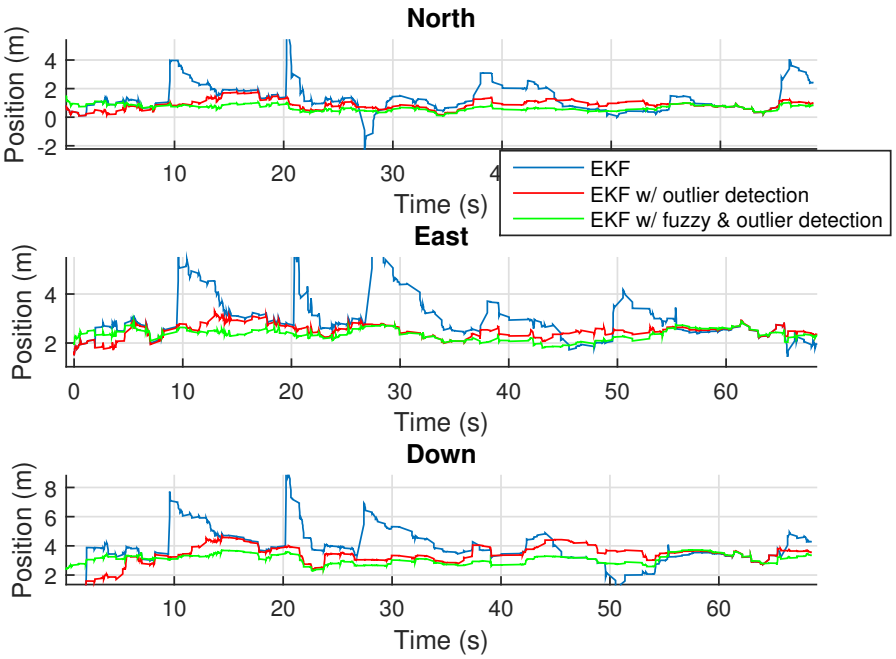


Figure 4.28: Indoor result using P model. The outlier rejection algorithms work better when the mathematical model is simpler. The gross errors due to poor measurement in standard EKF are rejected correctly.

4.6 Outdoor Experiments

Several in-flight tests using both manual and auto control were conducted at five different experiment excursions at Agdenes airfield. The purpose of the experiments was to find the maximum measurement range in-flight, and investigate if the Atmel kits performed better outdoor. For comparison, the Piksi - a low-cost, high-performance GPS receiver with RTK functionality for centimeter-level relative positioning accuracy, was used. The Piksi setup can be found in (Røli 2015). Figure 4.37 shows the operational setup at Agdenes airfield.

The anchors were placed on top of masts with different heights that were placed around at the airfield. Figure 4.38 shows the masts with attached anchors at the top. Note that the Atmel kits are very easy to setup as they are wireless and self-powered with AAA-batteries. The origin of the local NED frame was set at the Piksi base station, and their relative position in the North-East frame was found using the Testbed. The Testbed is just a test setup with BBB and Piksi, used mostly for debugging and mapping of Atmel anchors. The Down position was measured manually using a tape measure since the Piksi measurements in the Down frame are not reliable. More about the Piksi system, its measurement performance and accuracy, and the Testbed can be found in (Røli 2015)

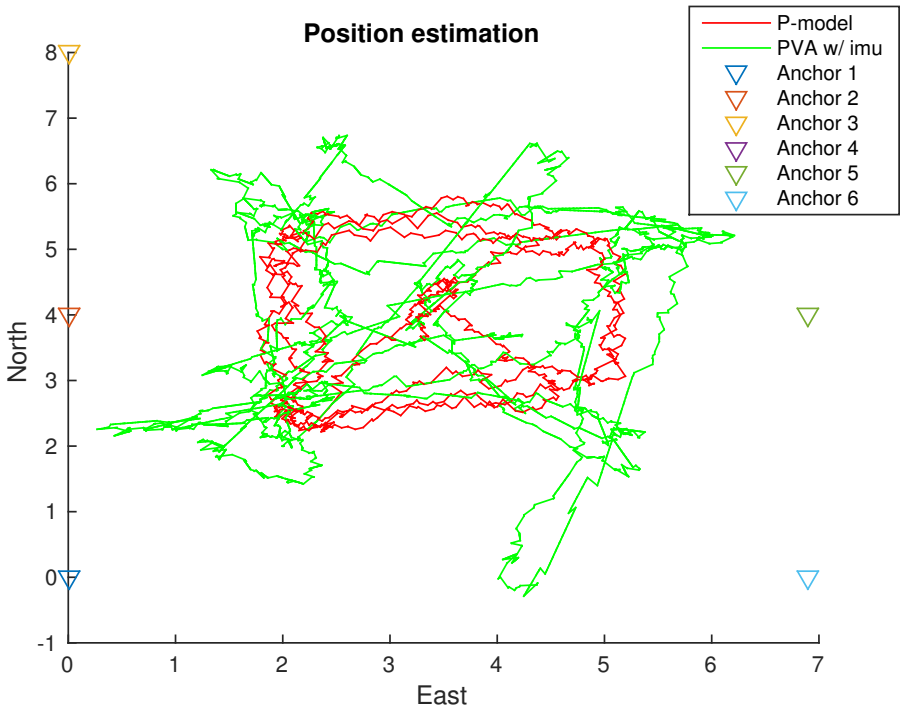


Figure 4.29: Indoor result comparison of P model and PVA with IMU in 2D.

4.6.1 Maximum Signal Range

Maximum range was determined by flying the multirotor away from the anchors. Line of sight was kept during the experiment. The maximum achieved range measurement was at about 200 meters. Remember from Section 4.2 that the theoretical maximum range should be around 300 meters. However, the signal quality was reduced immensely at 200 meters, which could be due to interference from the telemetry on-board the multirotor. Nonetheless, a range of 200 meters is more than satisfactory for the purpose of this thesis.

4.6.2 Signal Quality Analysis - Tool

The analysis tool discussed in Section 4.5.4 is also used for outdoor cases. The result of a static test using Piksi as reference is shown in Figure 4.39. The tool is very useful since the shortcomings of the system can be easily discovered. In Figure 4.39, we see that the anchor 3 is out of order and the anchor 5 has poor measurements. However, that is not a problem for our system since we have good redundancy. The rest of the anchors give impressive precision of their range measurements. Note that this result is much better than what was achieved indoor. Consequently, the multipath problem is more or less eliminated in outdoor experiments.

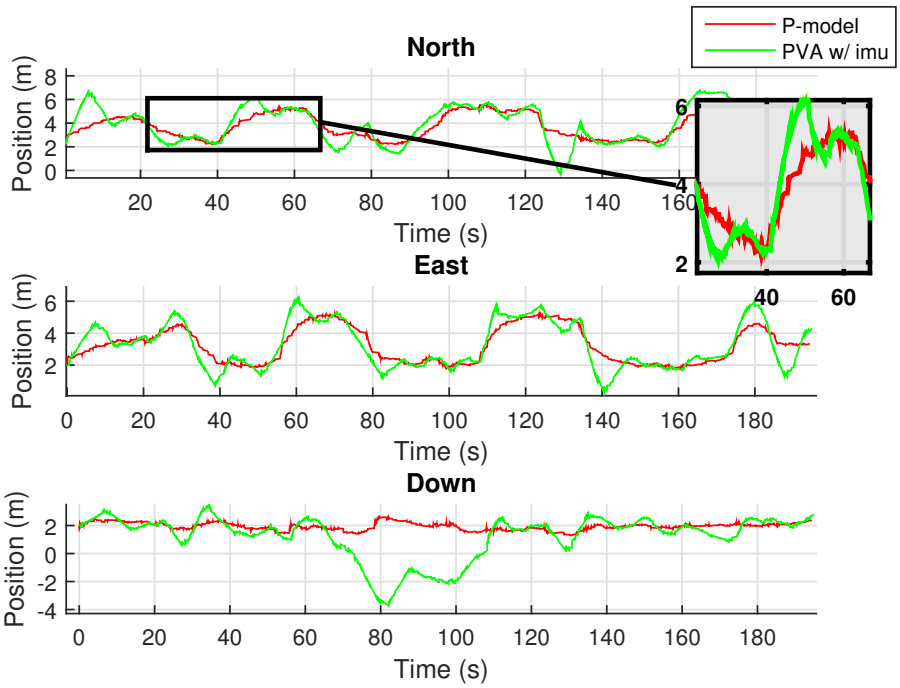


Figure 4.30: Positioning results using the PVA model with the accelerometer data integrated and the P model without IMU data.

4.6.3 Performance Review - High Speed Operations

The operation speed is usually higher on experiments conducted outdoor, while the measurement frequency remains the same. Therefore, the algorithms are re-reviewed to see how they perform on operations with relatively higher speeds. The result using the P model is shown in Figure 4.40. The performance is compared using the RTK satellite navigation technology - Piksi. Note that the outlier detection algorithm is able to reject outliers, however it starts to diverge after a while. This is due to longer periods with outliers. The residual term from the EKF is shown in Figure 4.40. Note that the residuals are highest after about 130 and 160 seconds, thereby most outlier occurrence. Also nota that this is where the EKF algorithm with outlier detection starts to diverge in the east and down frame respectively. However, utilizing the FLAS to adapt the covariance matrix R, the divergence problem is once again solved.

4.6.4 Integrating Accelerometer Data - Outdoor Operations

It has been shown that the IMU data from the Pixhawk is affected extensively from the multirotor vibrations in Section 4.5.2. The interference is so severe that the acceleration output seems like white noise. However, in outdoor operations, the speed is usually higher. Consequently, the IMU measurements contain more information. The result using accelerometer is shown in Figure 4.42. Note that the PVA model with integrated IMU data results in better estimates. This is expected, since the PVA model is better if the measurement frequency is increased. Moreover, the accelerometer data gives better result than in indoor operations

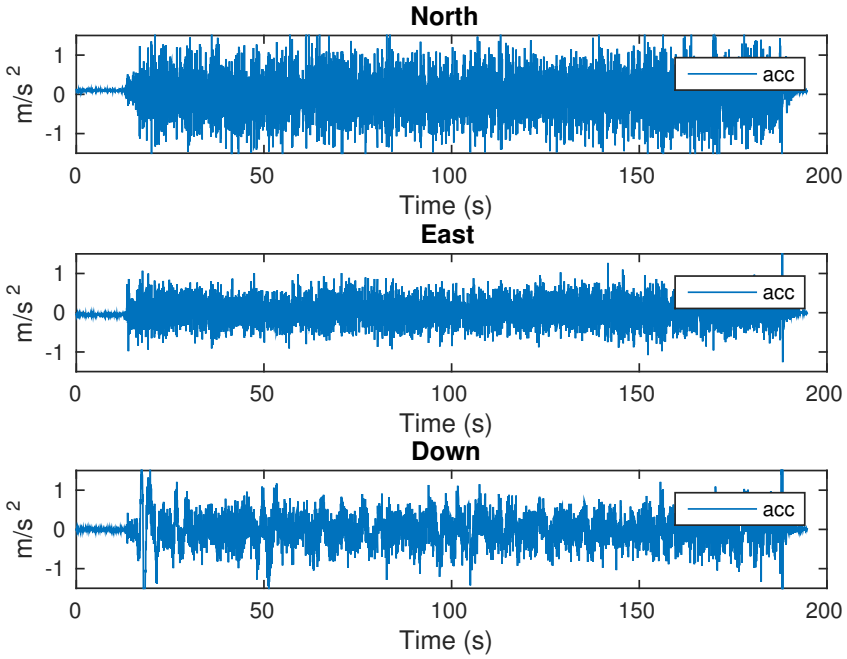


Figure 4.31: Accelerometer data from indoor experiment. The accelerometer measurements are affected more by the copter vibrations then the actual accelerations. Consequently, the operation speed indoor makes the low-cost IMU measurements more or less useless.

due to higher speed and g-forces. The accelerometer data is shown in Figure 4.43.

4.6.5 Discussion

Most of the problems regarding indoor experiments are eliminated outdoor. For instance, the multipath from the range measurements are eliminated. Moreover, the anchors are placed freely in different heights as well as the operation speed is increased. Hence the accelerometer data can be utilized.

However, the satisfaction of the estimation accuracy is still open for discussion. In worst cases, the estimation can deviate up to 1-2 meters. Consequently, operations that require high precision might be a problem. While 2 meters deviation can be acceptable outdoor, considering the accuracy offered by a traditional GPS is worse. However, low-cost navigation systems based on real-time kinematics, such as Piksi are showing promising results, even though they are not very reliable yet.

Figure 4.32: Indoor flight with the hexacopter in Robotlab. 6 anchors are used to track the position with the developed EKF while 16 OptiTrack cameras track the system.

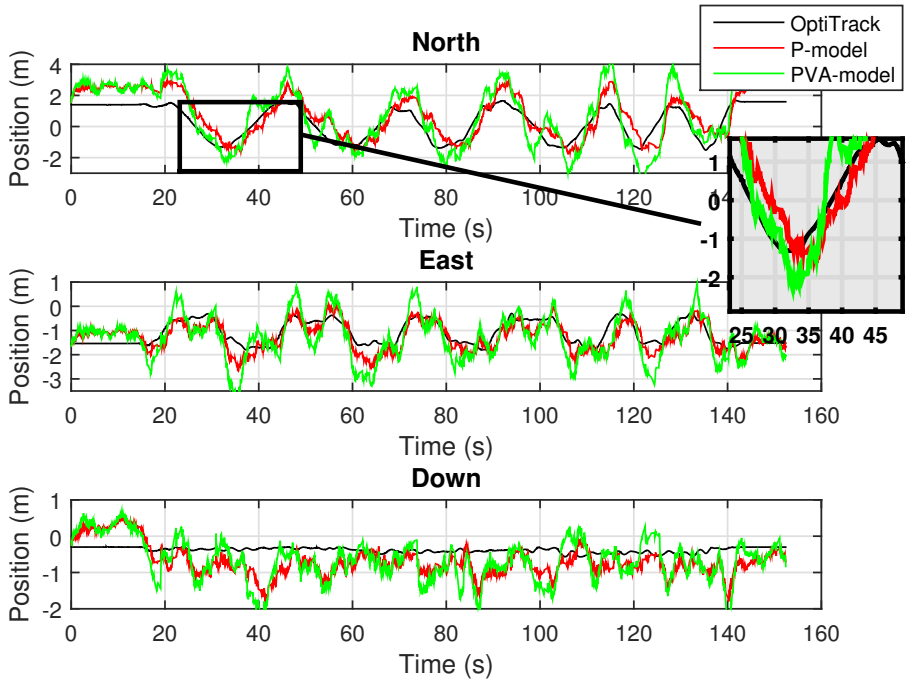


Figure 4.33: The EKF performance with P and PVA model indoor. Both models include outlier detection and rejection algorithm with FLAC to adjust the covariance matrix R. The OptiTrack motion tracking technology is used as a reference.

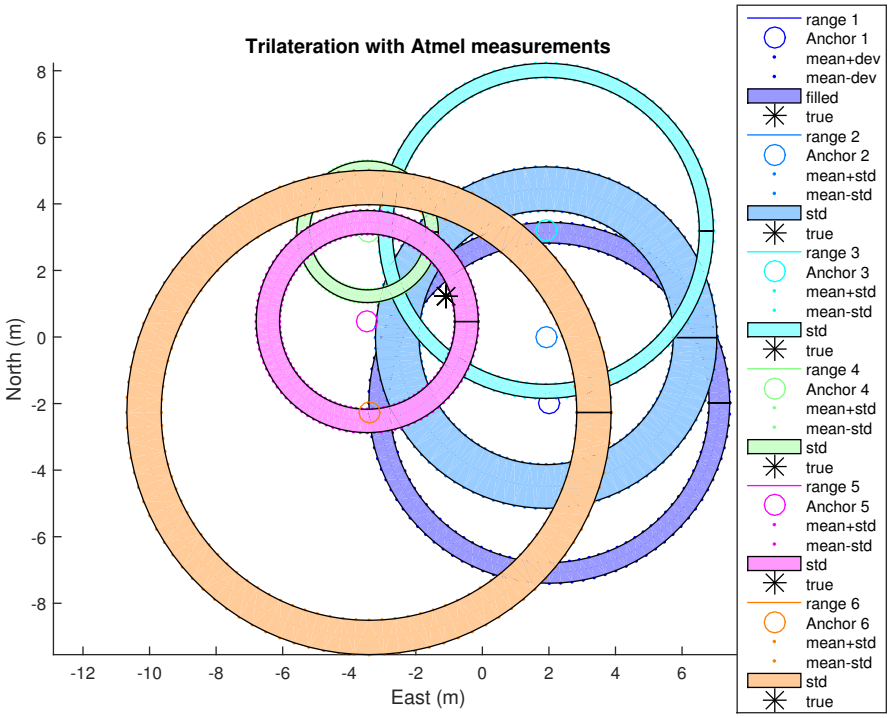


Figure 4.34: Principle of trilateration - indoor experiments. The projection of the Atmel range measurements in 2D with the standard deviation are shown. The real position shown with the star is the result from the OptiTrack system. It seems like the EKF have a bad starting point to estimate the position using the range measurements.

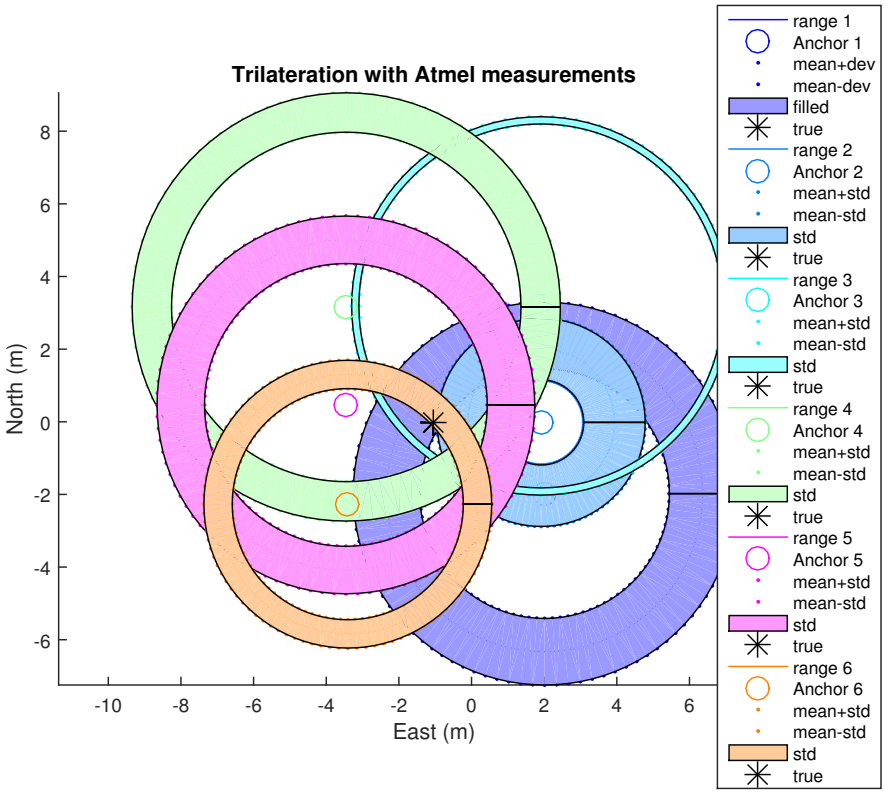


Figure 4.35: Principle of trilateration, improved - indoor experiments]. The projection of the Atmel range measurements in 2D with the standard deviation are shown. The real position shown with the star is the result from the OptiTrack system. The measurement setup can be easily improved using this technique.

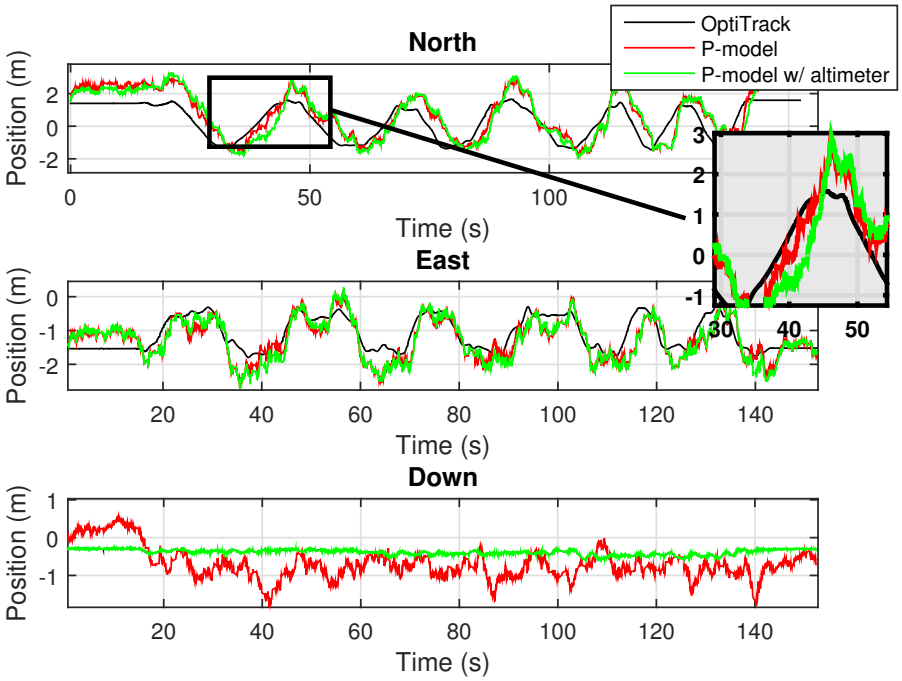


Figure 4.36: Result of positioning using P model with and without altimeter. The OptiTrack solution in Down frame is fed as an altimeter in the EKF. Altimeter improves the Down position estimation drastically, while the rest of the system is not affected much.

Figure 4.37: Outdoor experiment setup. Red circle shows 2 of 6 Atmel anchors placed around the field. Yellow circle shows one of the hexacopters. Green circle shows the telemetry between the ground control station and the testbed. Also the hexacopter has the same telemetry link. Black circle shows the Piksi base station and the Testbed Piksi antenna. The hexacopter has also the Piksi antenna.

Figure 4.38: Atmel anchors attached to masts during outdoor experiments. Picture is from Agdenes airfield.

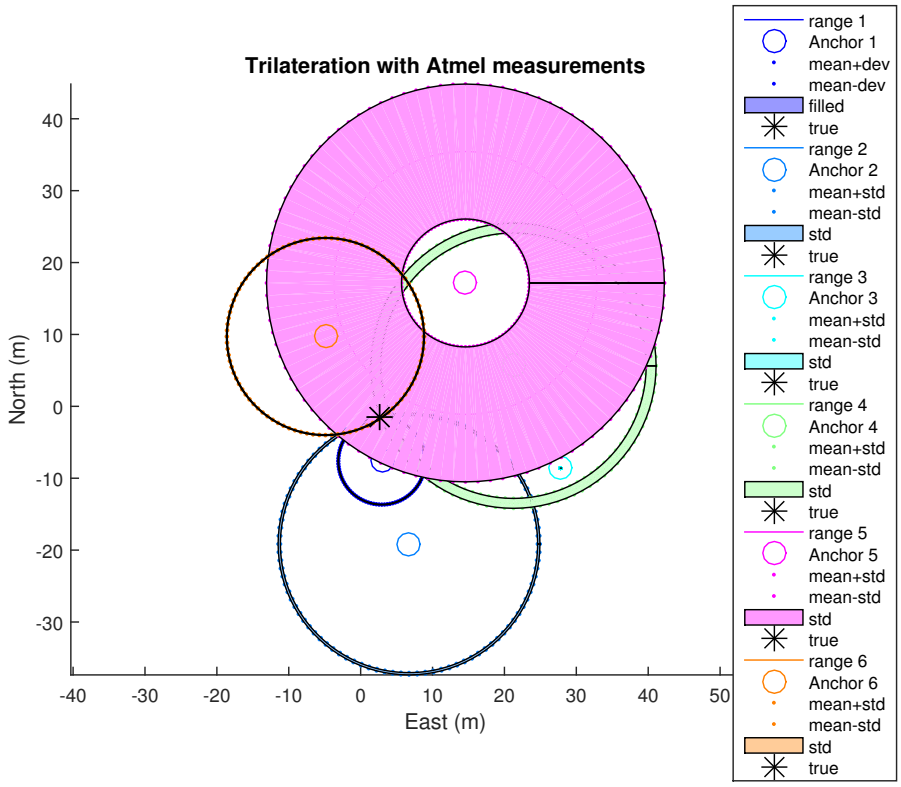


Figure 4.39: Principle of trilateration - outdoor experiment. The projection of the Atmel range measurements in 2D with the standard deviation is shown. The real position shown with the star is the result from the Piksi navigation system. Note that the anchor 3 is out of order and anchor 5 gives bad range measurements, probably due to bad positioning. However, the resulting anchors measure correctly and the result is much better than indoor case where there is multipath problem.

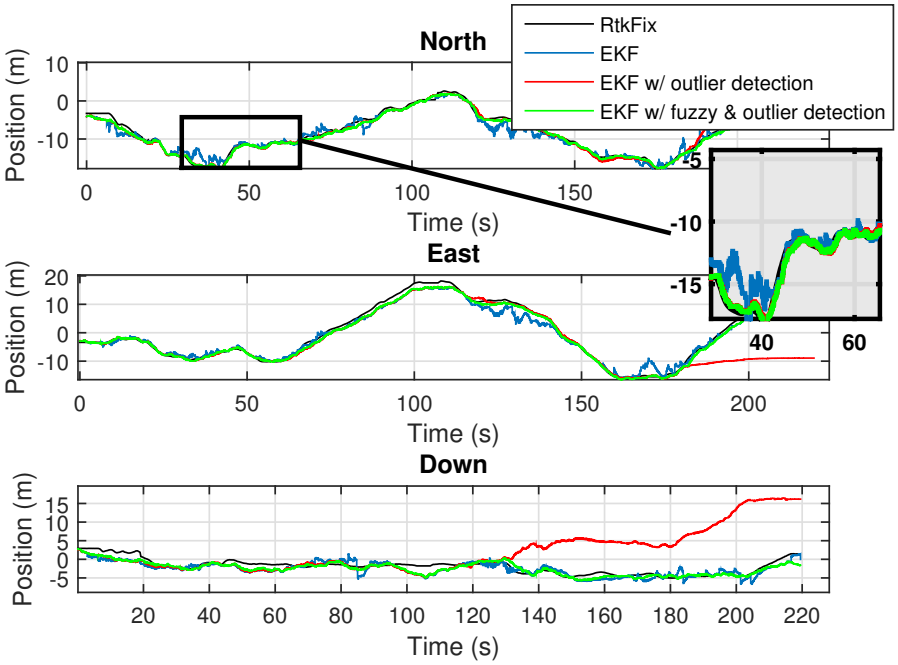


Figure 4.40: EKF algorithm performance - outdoor experiment. The EKF with outlier detection and FLAS rejects outliers, while avoiding divergence during periods with high outlier occurrence. Piksi - rtk GPS is used as reference.

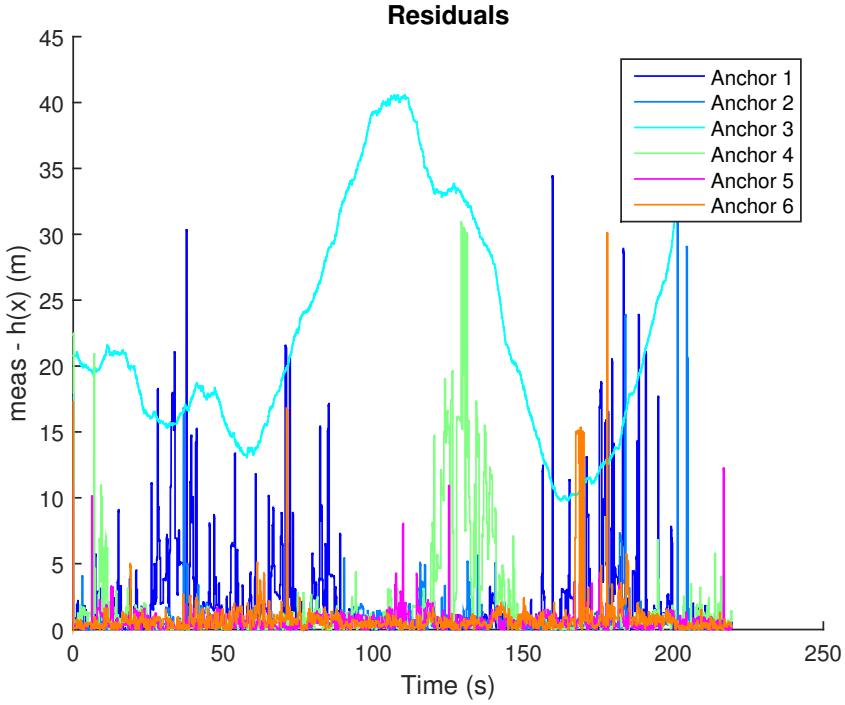


Figure 4.41: Plot of residual term in EKF filter. Note that the anchor 3 is out of order and do not measure distance. The outliers occur mostly after 130 and 160 seconds.

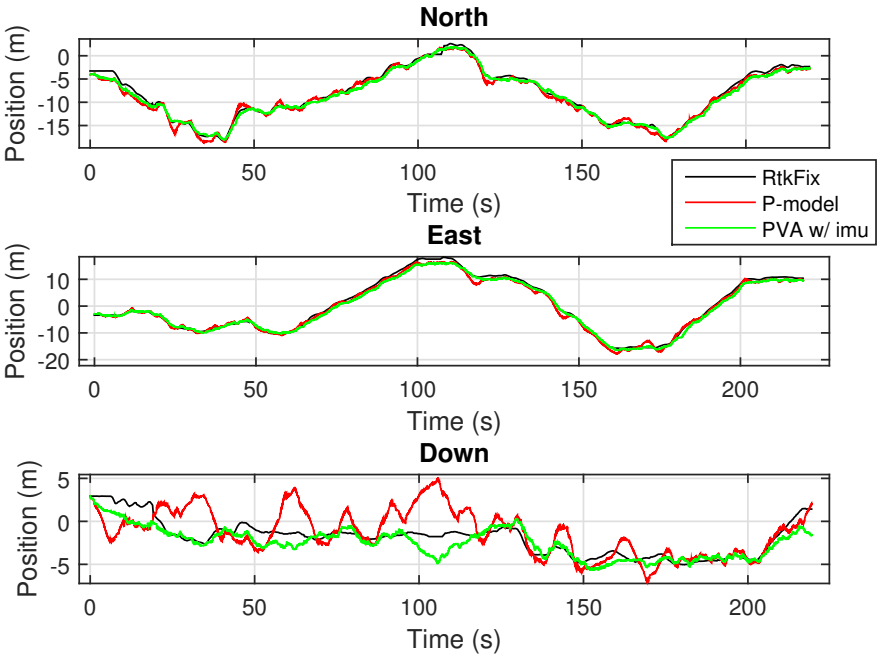


Figure 4.42: Performance comparison of P model and PVA model with IMU - outdoor experiment.

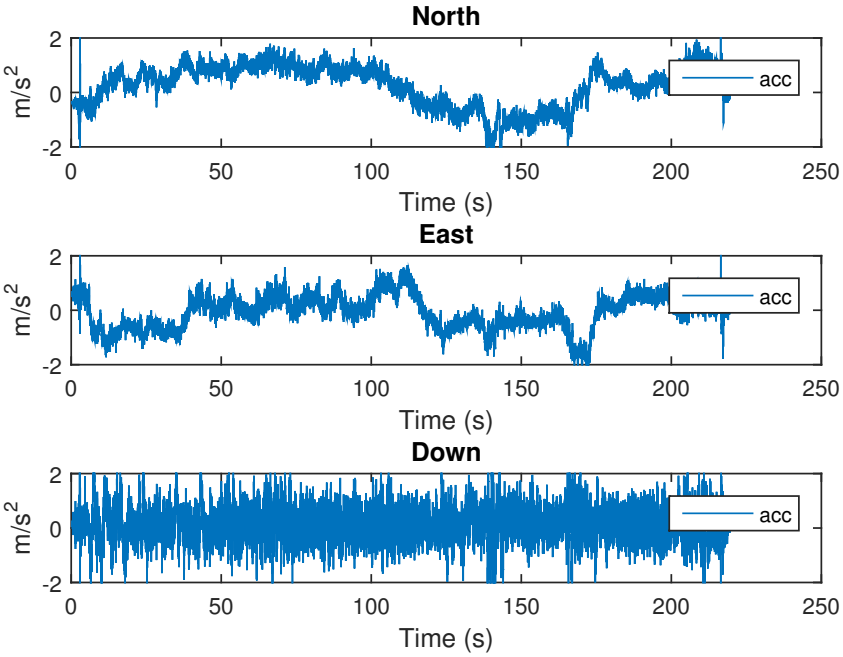


Figure 4.43: Raw accelerometer data during outdoor flight.

Chapter 5

Suspended Load Control

Autonomous vehicles can be used to carry, drop or pick up objects in hazardous environments without risking human life. In case of autonomous multirotors, this often implies an additional suspended load on the system. A suspended load often swings and introduces external forces. The external forces can make the system unstable or uncontrollable. In addition, uncontrolled swings on a suspended load is not desirable for drop and pick up operations. Therefore a swing reducing control scheme for autonomous multirotor systems with suspended load is crucial.

Suspended load control for autonomous vehicles is rare in the literature. Therefore, this chapter is mainly based on (Bisgaard 2007). However, most of the solutions are inspired from applications of overhead gantry cranes. Crane applications exhibits the pendulum like motion, but with simpler actuator dynamics. In cranes the well known *input shaping* technique is utilized. Input shaping filter is basically a feedforward control that shapes the reference command such that it avoids exciting the systems vibrational modes (Omar 2012). However, this method has no feedback to the system and does not deal with swings due to external forces. Therefore a second controller with feedback to the system is necessary to deal with unmodelled swings.

This chapter will present the suspended load problem, and introduce a control scheme that avoids swing excitation and damps induced swings. Results from both simulations and in-flight experiments will be presented.

5.1 The Suspended Load Problem

Human operators have difficulties damping swings on a suspended load connected to a helicopter (Hoh, Heffley, and Mitchell 2006). The swings on the load are often amplified by pilot induced oscillations. The concept is shown in Figure 5.1. The swing is amplified if the pilot counteracts the load swing by actuating in the opposite direction. However, if the pilot follows the motion of the suspended load as shown in Figure 5.2, the swing is damped. In other words, a negative feedback controller will amplify the load swing, while a positive feedback will damp it.

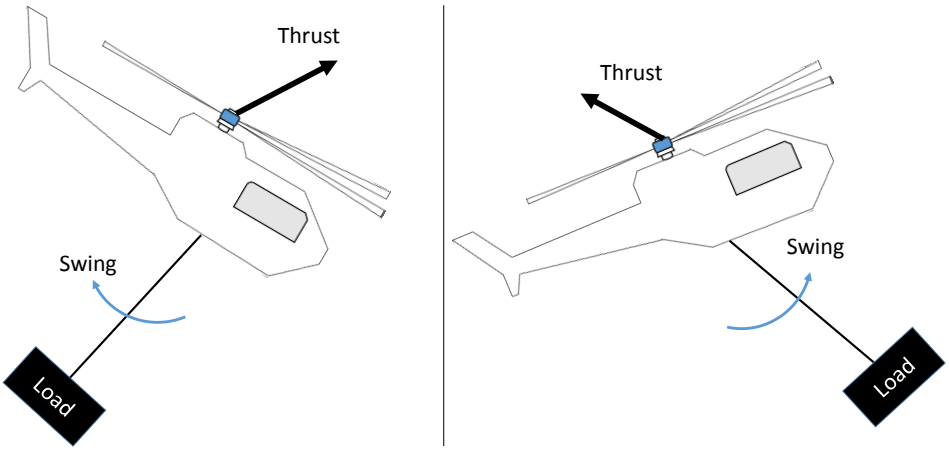


Figure 5.1: A pilot trying to counteract the swing load motion by actuating on opposite direction will amplify the load swing.

The main control problem of a multirotor system with suspended load is to stabilize the system while the actuators are used to avoid swing excitation on the load. Consequently, the problem is similar to controlling gantry cranes, but with added complexity since the multirotors are unstable by nature. It is important to realize that multirotors have good actuation on the yaw and z-axis, while the important longitudinal and lateral actuation for swing damping are poor. Therefore, the control system should be carefully considered.

5.2 Assumptions

This thesis will not consider the multirotor dynamics. Therefore, the load is assumed to be sufficiently small and light, such that the multirotor autopilot is unaffected by the load. Moreover, it is assumed that the multirotor is able to track a desired reference trajectory even with a suspended load. Consequently, the main problem is reduced to damp the load swing, reducing the complexity of the control design. Furthermore, the load is regarded as a point-mass connected by a rigid link at the centre of the gravity of the multirotor.

5.3 Control Scheme Design - Overview

The assumptions made in Section 5.2 was initially experimented with at Agdenes airfield and confirmed. The multirotor was flown manually to induce swings and set to loiter mode. Furthermore, a second operator induced swings by pulling the load manually as shown in Figure 5.3. The multirotor behaviour was not noticeably affected by the suspended load.

Therefore, the control design of the system can be separated into four independent blocks as shown in Figure 5.4. Note that the multirotor controller is regarded as a *black box* in this thesis and will not be explained in detail. The input shaping and the delayed feedback controller will be discussed in Section 5.4 and 5.5 respectively. The theory of the reference model is introduced in Section 2.3, while the practical implementation is presented in Section 5.7 and

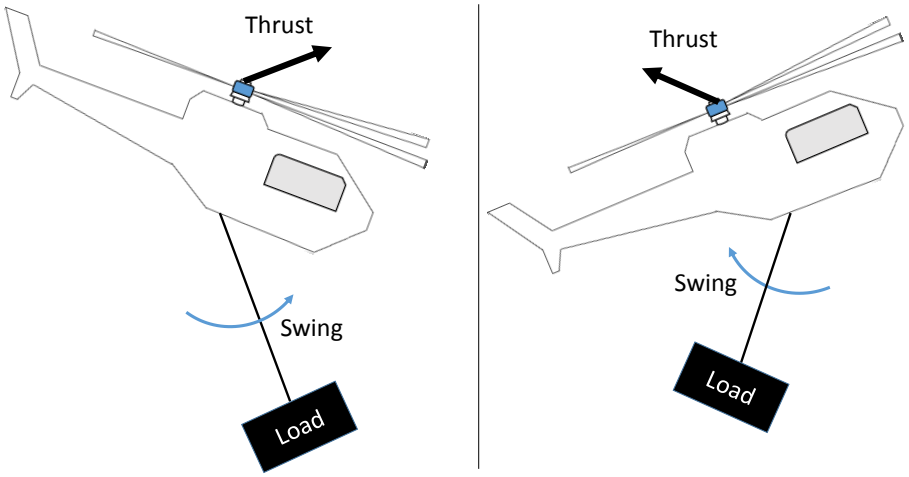


Figure 5.2: A pilot trying to counteract the swing load motion by following the swing load motion will damp the swing.

5.8.1. The suspended load model is modeled as a cart-pendulum system and used in the design of delayed feedback control. It is described in Section 5.5.1. All the design components will be summarized in Section 5.6.

5.4 Feedforward Swing Damping Control: Input Shaping

The concept of input shaping is based on a simple idea: *Vibrations caused by an impulse on a system can be cancelled by another impulse given at the right time.* The concept is explained in detail in Section 2.4 and demonstrated with Figure 2.4. The impulse response of A_1 at t_0 is cancelled by identical impulse given at t_1, which is exactly half a period later. However, this is only the case if there is no physical damping on the system.

Therefore, finding the parameters for A_1, A_2, t_0 and t_1 for a pendulum like motion system is more complex in reality. Section 2.4 describes the methodology and the parameters are given by the Equations (2.38), (2.39) and (2.40). The equations are shown below for readers convenience,

$$A_1 = \frac{1}{1+K}, \quad A_2 = \frac{K}{1+K} \tag{5.1}$$

$$t_0 = 0, \quad t_1 = \frac{T_d}{2} \tag{5.2}$$

where T_d is the damped period of vibrations and,

$$K = exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right) \tag{5.3}$$

.

See 2.4 for more detailed description of the parameters.

Figure 5.3: A picture from initial experiments with suspended load. The operator induces swings by pulling a rope that is attached to the load. The load weights 0.4 kg and attached 0.7m below the multirotor. The multirotor is on loiter mode during the experiments.

5.4.1 Command Shaping - Zero-Vibration Command Generation

The concept of oscillation free transition between hover to forward flight using input shaping is illustrated in Figure 5.5 for a suspended load system. Note that this is achieved simply by convolving the input shaper with the reference model. Figure 5.6 shows the position response of the multirotor system implemented as as described in 5.5.1. The multirotor is given a reference command with and without a shaped command, and the response of the suspended load is shown. It is clear that the system is able to avoid swing excitation with the shaped command.

Robustness of the *zero-vibration command generation* is dependent on exact knowledge about the natural frequency and damping of a system. (Singh and Singhose 2002) concludes that the method is capable of avoiding oscillation. However, in practise, identifying the exact values might not be possible. Figure 5.7 illustrates that there is still a significant amount of vibration left in the system if there is 10% error in the frequency. The frequency is defined in Equation (2.33).

The robustness of the input shaping method to modeling errors can be improved by adding another requirement: *Derivative of the residual vibration with respect to the natural fre-*

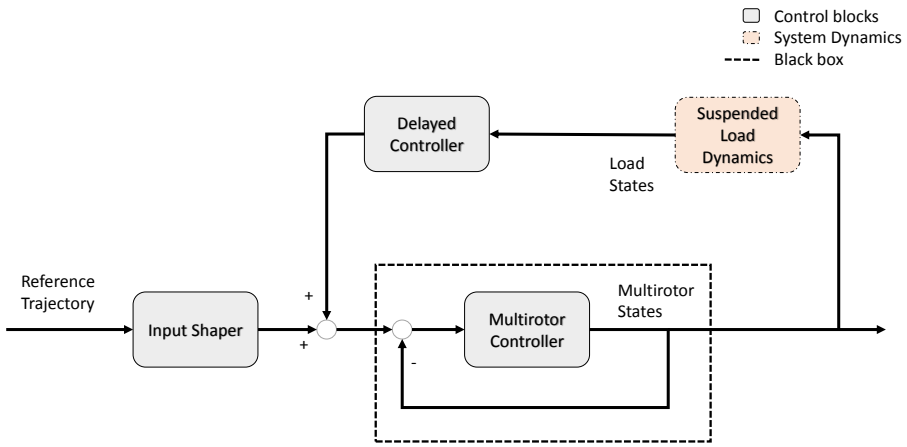


Figure 5.4: Suspended load control that damps swings. Each block is explained in the following sections. The input shaper in 5.4, the delayed feedback control in 5.5, suspended load model in 5.5.1, and reference model in 2.3, while the multirotor autopilot is regarded as a black box and will not be discussed.

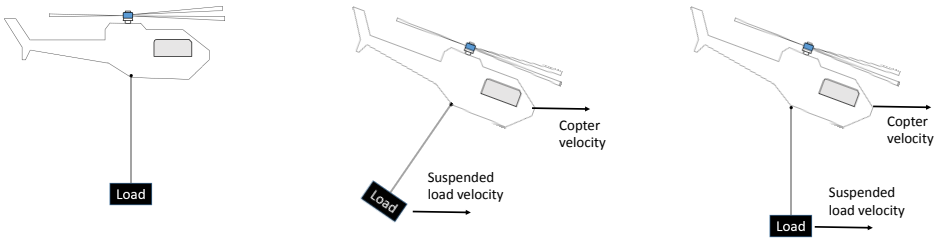


Figure 5.5: The concept of oscillation free transition from hover to forward flight on a suspended load system.

quency and damping is set to be zero (Bisgaard 2007). In other words, using Equation (2.26),

$$\frac{d}{d\omega}V(\omega,\xi)=0 \tag{5.4}$$

$$\frac{d}{d\xi}V(\omega,\xi)=0 \tag{5.5}$$

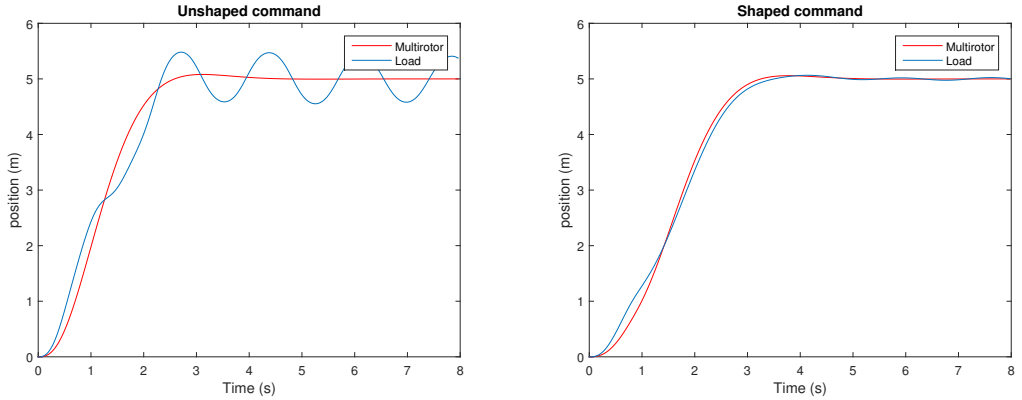
and thus, deriving Zero Vibration and Derivative Shaper (ZVD). (Singer 1989) shows that (5.4) and (5.5) can be reduced to,

$$\sum_{i=1}^{N} A_i t_i e^{\xi\omega t_i}\cos(\omega_d t_i)=0 \tag{5.6}$$

$$\sum_{i=1}^{N} A_i t_i e^{\xi\omega t_i}\sin(\omega_d t_i)=0 \tag{5.7}$$

where A_i are the amplitudes, t_i are the time location of the impulses, n is the number of impulses and $w_d=\sqrt{1-\xi^2}$.

Consequently, the complexity of the system increases, while the improvement is minimal. Moreover, the system dynamics are slowed down, resulting in extended command time (Bisgaard 2007). In addition, the solution will not be able to reduce swings due to external



(a) Unshaped command. **(b)** Shaped command.

Figure 5.6: Load response of shaped and unshaped command, showing an oscillation free transition from hover to forward flight using input shaping.

forces. Therefore an additional feedback controller, that deals with swings induced by external forces will be introduced in Section 5.5.

5.4.2 Real-Time Implementation

The input shaper is implemented by convolving the sequence of impulses with the desired command (Bisgaard 2007),

$$\bar{\mathbf{r}}_s = \mathbf{S} * \bar{\mathbf{r}} \tag{5.8}$$

where $\bar{\mathbf{r}}$ is the initial command, $\mathbf{S}$ is the input shaper, $*$ denotes convolution, and $\bar{\mathbf{r}}_s$ is the shaped command. In Section 2.3, the reference model was introduced as a smooth signal (also in Figure 5.4). However, in reality, the reference signal is discrete, hence discrete time domain convolution is necessary.

The mathematical definition of the discrete time domain convolution is (Proakis and Manolakis 1996),

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{5.9}$$

where $x[n]$ is the input signal, $h[n]$ is the impulse response, and $*$ denotes convolution.

Implementation of Convolution

Remember from Section 2.5 that it is actually sufficient to know about the impulse response, $h[n]$ in (5.9), to actually know about a systems reaction to any input. Also remember that the input signal in this case is the reference signal, shown in Figure 5.3. Moreover, the input shaper can be regarded as the impulse response. Thus, (5.8) can be expressed as (5.9), where

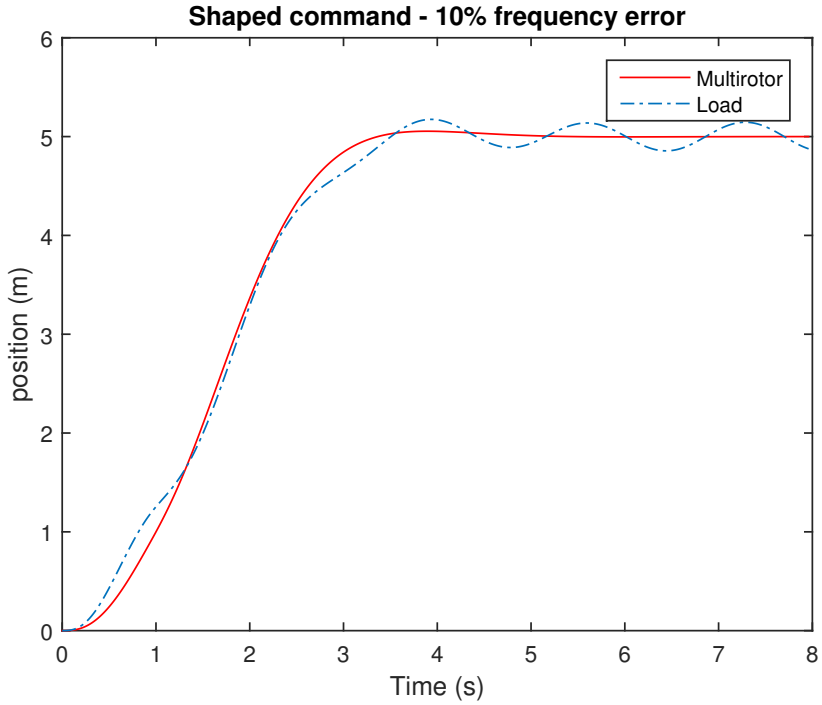


Figure 5.7: Load response with 10% frequency error - shaped command. There is still significant amount of oscillation left on the payload.

$\sum x[n] = \bar{\mathbf{r}}$, $y[n] = \bar{\mathbf{r}}_s$, $\sum h[n] = \mathbf{S}$ and where,

$$h[n] = \begin{cases} A_1 & \text{if } n = t_0 \\ A_2 & \text{if } n = t_1 \\ 0 & \text{otherwise} \end{cases}$$

Consequently, with $t_0 = 0$, the output sequence becomes,

$$\begin{aligned}
y[t_0 = 0] &= \sum_{k=-\infty}^{\infty} x[k] \cdot h[0-k] = x[0] \cdot h[0] = x[0] \cdot A_1 = x[n] \cdot A_1 \\
y[1] &= \sum_{k=-\infty}^{\infty} x[k] \cdot h[1-k] = x[0] \cdot h[1] + x[1] \cdot h[0] = x[1] \cdot A_1 = x[n] \cdot A_1 \\
y[2] &= \sum_{k=-\infty}^{\infty} x[k] \cdot h[2-k] = x[0] \cdot h[2] + x[1] \cdot h[1] + x[2] \cdot h[0] = x[n] \cdot A_1 \\
&\vdots \\
y[t_1] &= \sum_{k=-\infty}^{\infty} x[k] \cdot h[t_1-k] = x[n-t_1] \cdot A_2 + x[n] \cdot A_1 = x[t_0 = 0] \cdot A_2 + x[n] \cdot A_1 \\
&\vdots \\
y[n] &= = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] = x[N-t_1] \cdot A_2 + x[n] \cdot A_1
\end{aligned}$$

thus, the output from the convolution is in fact,

$$y[n] = \begin{cases} x[n] \cdot A_1 & \text{if } n < t_1 \\ x[n-t_1] \cdot A_2 + x[n] \cdot A_1 & n \geq t_1 \end{cases}$$

which is easily implemented given that A_1, A_2 and t_1 is known. This form is also more intuitive than (5.8).

The performance of this method is presented in Section 5.7 and 5.10.

5.5 Feedback Swing Damping Control: Delayed Control

The input shaping filter does not have feedback to the system. Thus, unable to cope with modelling errors, and externally induced oscillations. Therefore a delayed feedback controller is necessary. Note that a delayed feedback controller is in general not an optimal solution, as it can lead to performance deterioration or instability. However, (Masoud, Daqaq, and Nayfeh 2004) has shown that a delayed resonator, which is a similar case, can be used to damp swings on ship cranes. Furthermore, (Bisgaard 2007) utilizes this technique for a helicopter suspended load system with great success.

In contrast to (Bisgaard 2007), the delayed feedback controller introduced in this section, will be design to damp load oscillations and will not consider the effects of the load on the vehicle. Thus, the solution will be uncoupled from the vehicle autopilot.

5.5.1 Delayed Feedback Control for Susupended Load System

The multirotor autopilot is assumed to track a reference signal with a sufficiently small and light load. Therefore, the multirotor dynamics are neglected in the design of the delayed feedback control. Moreover, it is assumed that the multirotor is operating in the NED frame, and the motion in the down direction does not affect the load dynamics.

For simplicity, consider zero pitch, roll and yaw, thus making the body- and NED-frame equal. Consequently, the load dynamics can be regarded as a pendulum moving on a plane in the north-east direction. Therefore, the system dynamics from multirotor position to load angle is modeled as a second order linear system (Egeland and Gravdahl 2002),

$$\frac{\theta_L}{x_b} = \frac{-s^2/\ell}{s^2 + g/\ell} \quad and \quad \frac{\phi_L}{y_b} = \frac{-s^2/\ell}{s^2 + g/\ell} \tag{5.10}$$

where ℓ is the wire length, g is the gravity, ϕ_L is the load angle around the x_b axis, θ_L is the load angle around the y_b axis. The x_b and y_b axis are given in the body-frame as shown in Figure 5.8.

(Bisgaard 2007) presents a delayed feedback controller that damps swings on a cart-pendulum system,

$$x_r(t) = G_d \ell sin(\theta_L(t - \tau_d)) \tag{5.11}$$

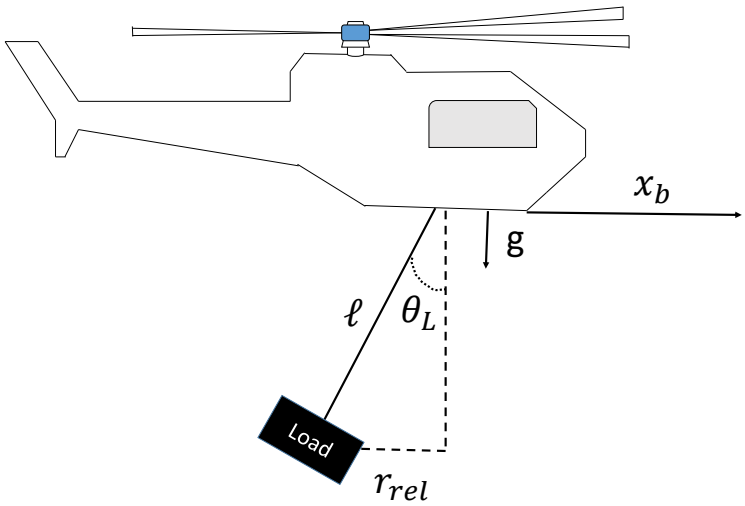


Figure 5.8: The copter-pendulum system shown along x-axis.

where τ_d is the controller delay, G_d is the design parameter, and ℓ is the rope length between copter and the load. The system described above is in fact the same as a cart-pendulum system that is restricted to 1 dimension(see Figure 5.8). Therefore, the same delayed feedback controller derived in (Bisgaard 2007) can be utilized, thus achieving the delayed feedback controller for a multirotor system in Nort-East plane,

$$x_r(t) = G_d \ell sin(\theta_L(t-\tau_d)) \tag{5.12}$$
$$y_r(t) = G_d \ell sin(\phi_L(t-\tau_d)) \tag{5.13}$$

Moreover, note that the term $\ell sin(\theta_L(t-\tau_d))$ and $\ell sin(\phi_L(t-\tau_d))$ in (5.12) and (5.13) are describing the relative position between the multirotor and the load in x_b and y_b axis. Thus, the feedback is given in the body frame. This relative position is shown in Figure 5.8 as r_{ref}.

However, the multirotor has in fact roll, pitch and yaw, which can not be neglected. Therefore, the system actually is described as shown in Figure 5.9.

Consequently, the relative position between the load and the copter in the NED-frame becomes,

$$\mathbf{P}_{ln} = \mathbf{R}_b^n \mathbf{R}_l^b \mathbf{L} \tag{5.14}$$

where

$$\mathbf{R_b^n(\Theta_{nb})} = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi)+c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi)+c(\psi)c(\phi)s(\theta) \\ s(\psi)c(\theta) & c(\psi)c(\phi)+s(\phi)s(\theta)s(\psi) & -c(\psi)s(\phi)+s(\theta)s(\psi)c(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \tag{5.15}$$

with $\mathbf{\Theta_{nb}} = [\phi, \theta, \psi]^T$ is the rotation of the body in relation to the NED frame,

$$\mathbf{R}_l^b = \mathbf{R}_x(\phi_L)\mathbf{R}_y(\theta_L) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi_L) & -s(\phi_L) \\ 0 & s(\phi_L) & c(\phi_L) \end{bmatrix} \begin{bmatrix} c(\theta_L) & 0 & s(\theta_L) \\ 0 & 1 & 0 \\ -s(\theta_L) & 0 & c(\theta_L) \end{bmatrix} \tag{5.16}$$

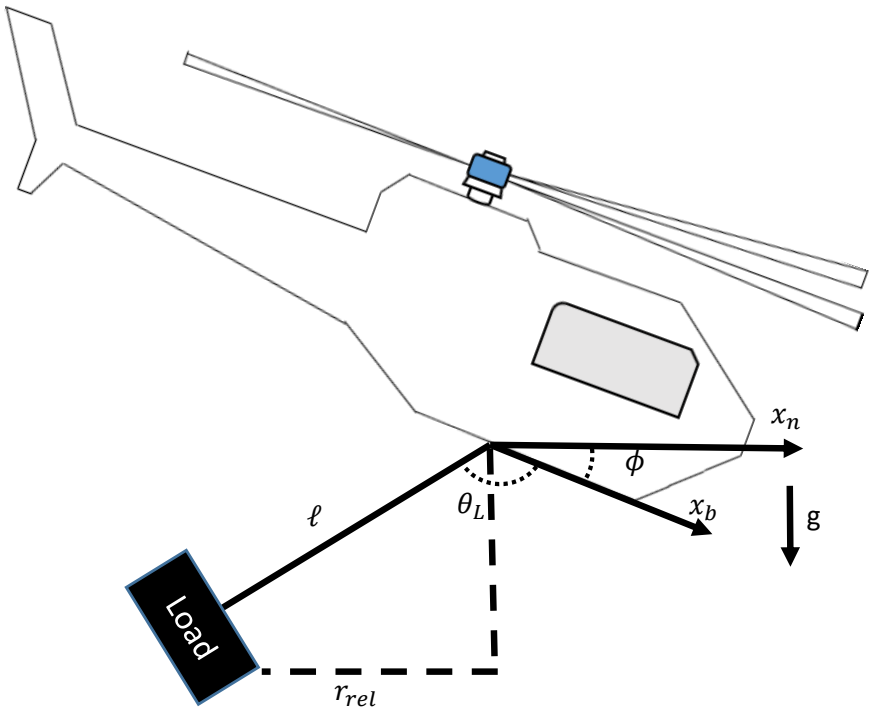


Figure 5.9: The copter-pendulum system in the NED frame.

is the loads rotation in relation to the body, and $\mathbf{L} = [0, 0, \ell]^T$. Remember that ℓ is the rope length, ϕ_L and θ_L is the loads rotation about x_b and y_b axis in the body frame respectively. Hence, the controller in (5.12) and (5.13) can be rewritten to,

$$\mathbf{x}_n(t) = G_d \mathbf{P}_{ln} \tag{5.17}$$

Note that the design parameter G_d is independent of the wire length, which means that the controller does not need to be redesigned when the rope length is changed. Another nice property is that the velocity reference can be easily achieved by derivation, thus

$$\dot{\mathbf{x}}_n(t) = G_d \dot{\mathbf{P}}_{ln} \tag{5.18}$$

Moreover, the controller delay is defined as normalized with respect to pendulum oscillation period T_n (Bisgaard 2007),

$$\tau_d = T_n \tau_n \tag{5.19}$$

where

$$T_n = 2\pi \sqrt{\frac{l}{g}} \tag{5.20}$$

Consequently, both control parameters G_d and τ_d, are designed such that they are normalized to the pendulum length. In other words, the controller parameters are independent of the wire length ℓ, and thereby automatically adaptive to wire length changes. Moreover, G_d and τ are calculated solving the optimization problem shown in Equation (2.52). However, this optimization is not discussed in this thesis and values presented in (Bisgaard 2007) is used.

5.6 Control Scheme Design - Summary

The control scheme design overview described in Section 5.3 will be summarized in this Section. All the components from the block box design in Figure 5.4 is explained in detail in the above sections. However, the system is extensive, and a brief summery will be given here for readers convenience. There will not be introduced new components in this section.

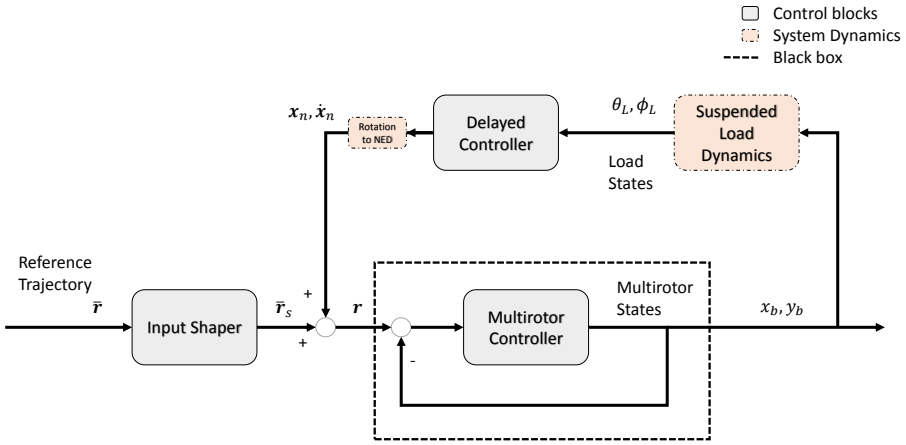


Figure 5.10: The suspended load control design. Figure shows each blocks interaction with each other.

The system is summarised in Figure 5.10, where $\bar{\mathbf{r}}$ is a reference signal, $\mathbf{S}$ is the input shaper, and $\bar{\mathbf{r}}_s$ is defined as,

$$\overline{\mathbf{r}_s} = \bar{\mathbf{r}} * \mathbf{S} = y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{5.21}$$

where,

$$h[n] = \begin{cases} A_1 & \text{if } n = t_0 \\ A_2 & \text{if } n = t_1 \\ 0 & \text{otherwise} \end{cases}$$

resulting in,

$$y[n] = \begin{cases} x[n] \cdot A_1 & \text{if } n < t_1 \\ x[n-t_1] \cdot A_2 + x[n] \cdot A_1 & n \geq t_1 \end{cases}$$

with,

$$A_1 = \frac{1}{1+K}, \quad A_2 = \frac{K}{1+K} \tag{5.22}$$

$$t_0 = 0, \quad t_1 = \frac{T_d}{2} \tag{5.23}$$

where T_d is the damped period of vibrations and,

$$K = exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right) \tag{5.24}$$

Moreover, the multirotor autopilot is regarded as a black box, returning the multirotors position. The pendulum dynamics from the multirotor position to pendulum are modeled as a second order linear system,

$$\frac{\theta_L}{x_b} = \frac{-s^2/\ell}{s^2 + g/\ell} \quad and \quad \frac{\phi_L}{y_b} = \frac{-s^2/\ell}{s^2 + g/\ell} \tag{5.25}$$

described in the body-frame. The output of the time-delayed feedback is the displacement of the load, which is be added to the longitudinal and lateral directions of the multirotor trajectory. The displacement is given in Equation (5.17) and (5.18)

5.7 Simulations - MATLAB

This section will introduce the simulation study conducted to determine the performance of the input shaping filter and delayed feedback controller. The control scheme used in the simulation study is shown in Figure 5.11.

This section will first introduce each of the blocks shown in Figure 5.11 with corresponding parameters utilized in the simulation study. The results from the simulation study will also be presented.

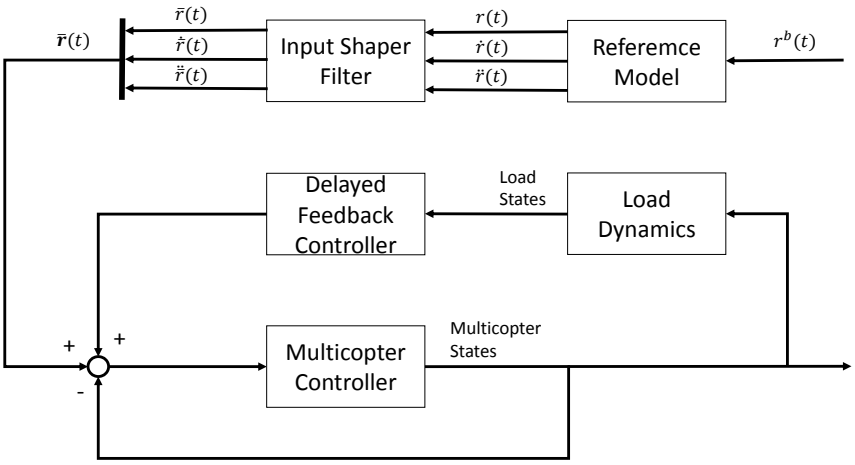


Figure 5.11: Control design used in the simulation study.

5.7.1 Simulation Setup and Parameters

Each block in the control design overview shown in Figure 5.11 will be presented in this section.

Reference Model

The reference model is generated using a 3rd order reference model based on a mass-damper system. The model is introduced in Section 2.3.2, but for readers convenience shown below,

$$\dot{\boldsymbol{\xi}}_d = \mathbf{A}_d \boldsymbol{\xi}_d + \mathbf{B}_d \mathbf{r} \tag{5.26}$$

with,

$$\mathbf{A}_d = \begin{bmatrix} 0 & \mathbf{1}_{3x3} & 0 \\ 0 & 0 & \mathbf{1}_{3x3} \\ -\omega_0^3\mathbf{1}_{3x3} & -(2\zeta+1)\omega_0^2\mathbf{1}_{3x3} & -(2\zeta+1)\omega_0\mathbf{1}_{3x3} \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \omega_0^3\mathbf{1}_{3x3} \end{bmatrix} \tag{5.27}$$

where $\omega_{n_i} = 2$ and $\zeta = \frac{1}{\sqrt{2}}$.

Input Shaper

The input shaping filter is implemented by convolving the reference model and the input shaper command as discussed in Section 5.4. Parameters used in the simulation study is shown in Table 5.1.

Table 5.1: Input shaper filter parameters used in the simulation study.

Parameter	Definition	Value
g	Gravity	9.81 m/s^2
ℓ	Rope length	0.7
P_d	Damping factor (Wind)	0.01
m_L	Load weight	2 kg
ω	$\sqrt{\frac{g}{\ell}}$	3.7436
ζ	$\frac{p_d}{2\omega m_L}$	$6.6781e^{-4}$
ω_d	$\omega\sqrt{1-\zeta^2}$	3.7436
K	$exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)$	0.9979
A_1	$\frac{1}{1+K}$	0.5005
A_2	$\frac{K}{1+K}$	0.4995
T_d	$\frac{2\pi}{\omega_d}$	1.6784
t_0	Time of initial impulse	0
t_1	$\frac{T_d}{2}$	0.8392 s

Multirotor Controller and Multirotor dynamics

The multirotor dynamics and controller used in the simulation study is based (Klausen, Fossen, and Johansen 2015). The multirotor controller will not be discussed in this thesis and will be regarded as a black box. It is assumed that the controller is able to track a reference signal with a suspended load on the system.

Load Dynamics

The suspended load dynamics are modelled as a pendulum, where it is considered as a point mass connected by a rigid link at the centre of gravity of the multirotor. Thus, the dynamical model is found by utilizing Kane's method (Egeland and Gravdahl 2002), and adding a linear damping force **D** (Klausen, Fossen, and Johansen 2015),

$$\dot{\boldsymbol{\eta}} = \boldsymbol{\nu} \tag{5.28}$$

$$\mathbf{M}^*(\boldsymbol{\eta})\dot{\boldsymbol{\nu}} + \mathbf{C}^*(\boldsymbol{\eta},\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{G}(\boldsymbol{\eta}) + \mathbf{D}\boldsymbol{\nu} = \boldsymbol{\tau} \tag{5.29}$$

where $\boldsymbol{\eta} = [\mathbf{P}, \psi_L, \theta_L]^T$, $\mathbf{M}^*(\boldsymbol{\eta})$ and $\mathbf{C}^*(\boldsymbol{\eta},\boldsymbol{\nu})$ is shown in (5.31) and (5.32) respectively, $\boldsymbol{\tau} := [\mathbf{F}, \mathbf{0}_{1x3}]^T$, $\boldsymbol{\nu} := [\mathbf{v}^T, \dot{\psi}_L^T, \dot{\theta}_L^T]$, $\mathbf{D} = diag\{0,0,0,P_d,P_d\}$ and

$$\mathbf{G}(\boldsymbol{\eta}) = \begin{bmatrix} 0 \\ 0 \\ -g(m_L + m_c) \\ Lgm_L c\theta_L s\psi_L \\ Lgm_L c\psi s\theta_L \end{bmatrix} \tag{5.30}$$

$$\mathbf{M}^*(\boldsymbol{\eta}) = \begin{bmatrix} m_L + m_c & 0 & 0 & 0 & \ell m_L c\theta_L \\ 0 & m_L + m_c & 0 & -\ell m_L c\psi_L s\theta_L & \ell m_L s\psi_L s\theta_L \\ 0 & 0 & -\ell m_L c\psi_L \theta_L & -\ell m_L c\theta_L s\psi_L & -\ell m_L c\psi_L s\theta_L \\ 0 & \ell m_L c\psi_L c\theta_L & \ell m_L s\psi_L c\theta_L & \ell^2 m_L c\theta_L^2 & 0 \\ \ell m_L c\theta_L & \ell m_L s\psi_L s\theta_L & -\ell m_L c\psi_L s\theta_L & 0 & \ell^2 m_L \end{bmatrix} \tag{5.31}$$

$$\mathbf{C}^*(\boldsymbol{\eta}, \boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -\ell\dot{\theta}_L m_L s\theta_L \\ 0 & 0 & 0 & \ell m_L(\dot{\psi}_L c\theta_L s\psi_L + \dot{\theta}_L c\psi_L s\theta_L) & \ell m_L(\dot{\psi}_L s\theta_L c\psi_L + \dot{\theta}_L s\psi_L c\theta_L) \\ 0 & 0 & 0 & -\ell m_L(\dot{\psi}_L c\theta_L c\psi_L - \dot{\theta}_L s\psi_L s\theta_L) & -\ell m_L(\dot{\theta}_L c\theta_L c\psi_L - \dot{\psi}_L s\psi_L s\theta_L) \\ 0 & 0 & 0 & -\frac{1}{2}\ell^2\dot{\theta}_L s(2\theta_L) & -\frac{1}{2}\ell^2\dot{\psi}_L s(2\theta_L) \\ 0 & 0 & 0 & \frac{1}{2}\ell^2\dot{\psi}_L s(2\theta_L) & 0 \end{bmatrix} \tag{5.32}$$

Note that $\mathbf{P}$ is the multirotors position in the NED frame, and ψ_L and θ_L is the load angle around the east and north axis respectively.

The parameters related to load dynamics are shown in Table 5.2

Table 5.2: Parameters related to load angle used in the simulation study.

Parameter	Definition	Value
g	Gravity	9.81 m/s^2
m_L	Load weight	0.4 kg
m_c	Copter weight	2.5 kg
ℓ	Rope length	0.7 m
P_d	Damping factor (Wind)	0.01

Delayed Feedback Controller

The delayed feedback controller is implemented using (5.17) and (5.18) with the values in Table 5.3.

5.7.2 Results

The case studied in the simulation study consists of tracking one waypoint reference as shown in Figure 5.12. The waypoint is given in NED-frame: $WP = [1, 1, -1]^T$, whereas the copters initial position is at origo.

Table 5.3: Delayed feedback parameters used in the simulation study.

Parameter	Definition	Value
G_d	Design parameter	0.325
τ_n	Design parameter	0.325
T_n	Pendulum oscillation period	1.6784 kg
τ_d	Time delay : $T_n\tau_n$	0.5455

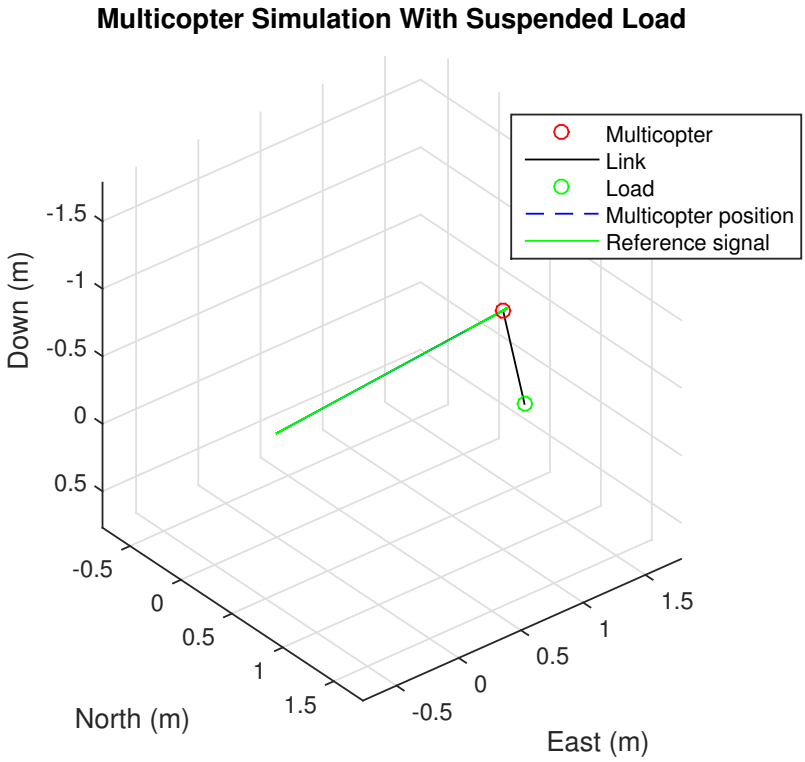


Figure 5.12: Simulated path of the multirotor in 3D. The multirotor goes from origo to WP = $[1,1,-1]^T$

The generated reference signal is shown in Figure 5.13. Note a 3rd-order reference model is used, thus smooth position, velocity and acceleration reference. It is assumed that the controller is able to track the given reference.

Resulting Load Angle Response

The resulting load angle response from the experiment is shown in Figure 5.14. The load angle responses of both θ_L and ψ_L, with and without the controllers are shown.

Figure 5.14 shows clearly that the input shaper is able to avoid swing excitation in the transition from hover to forward flight. However, it is not able to cope with the remaining oscillations due to modelling error which is more clearly shown in Figure 5.15. The delayed feedback controller however, is not able to avoid swing excitation as well as the input shaper, but is actively damping the remaining oscillations. Consequently, the two controllers can be

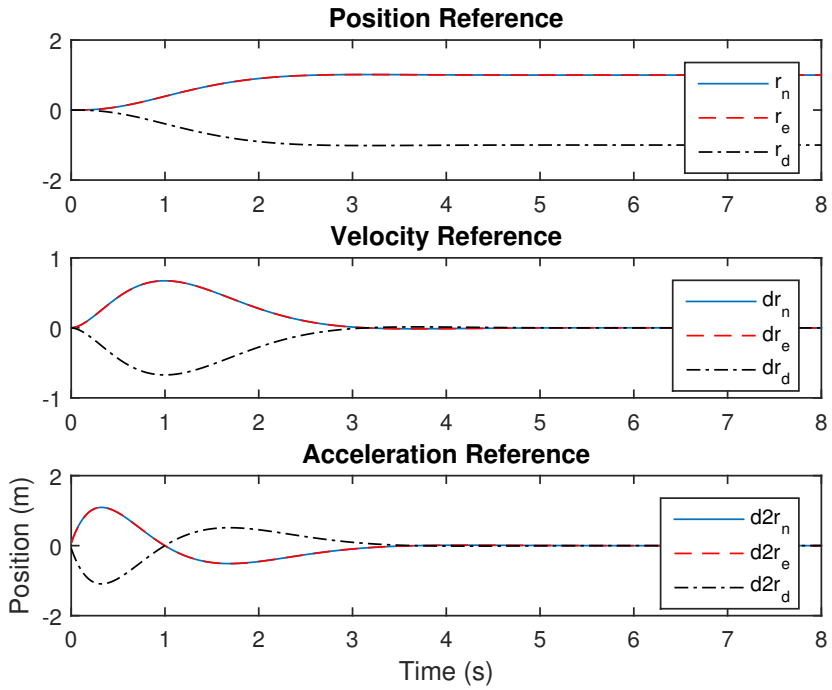


Figure 5.13: Reference signal generated to follow the path shown in Figure 5.12. The signals are used by the multirotor controller as shown in Figure 5.11.

combined to achieve maximum damping.

However, Figure 5.15 shows that the performance is not improved when both controllers are activated at the first transition. The input shaper have better performance without the delayed feedback. This can be explained by the design of the feedback controller. Remember that the delayed feedback controller basically tries to damp a predicted swing based on a mathematical model and an old measurement (thus the name delayed feedback). On the other hand, the input shaper reduces the initial swing. Consequently, the feedback controller is trying to counteract an expected swing that is in fact already damped by the input shaping filter. Thus the initial swing reduced by the input shaper is amplified by the feedback controller due to model mismatch.

The problem is easily solved by delaying the activation of the delayed feedback controller. Figure 5.15 shows the response when the delayed feedback is activated 3.5 seconds after the first move and when both controllers are active at the same time.

5.7.3 Discussion

The simulation study shows that both the input shaping filter and the delayed feedback controller works satisfactory. While the input shaper is able to avoid initial swing excitation, the feedback controller damps the remaining oscillations.

The performance is deteriorating when both controllers are active at the initial transition. It has been shown that the controllers are counteracting each other. However, the problem is

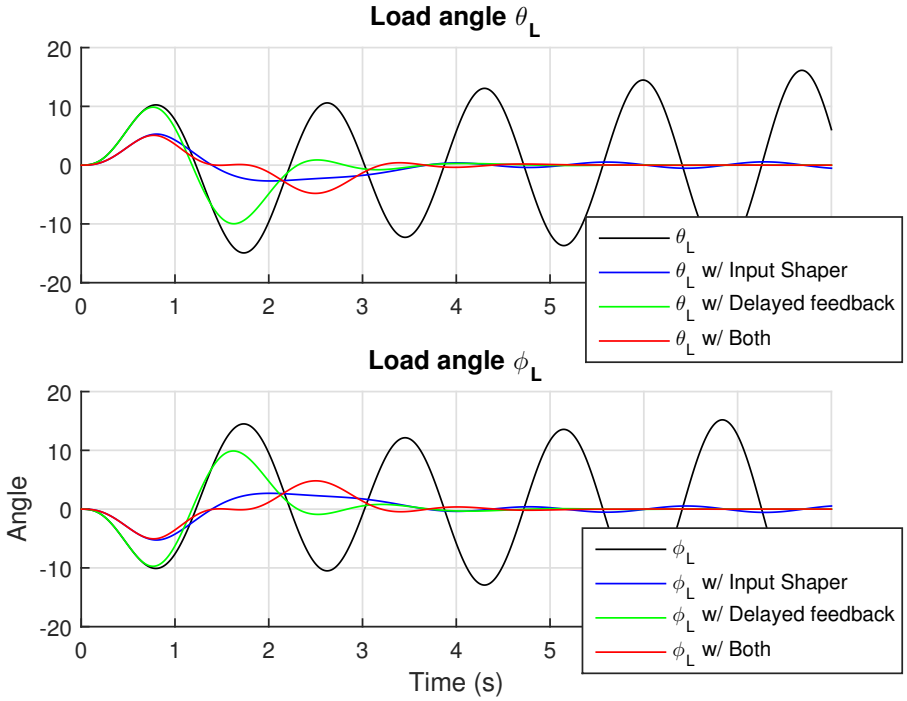


Figure 5.14: Resulting load angle responses given the reference signal shown in Figure 5.13. The input shaper avoids swing excitation, and the delayed feedback controller damps remaining oscillations. When both are activated, the initial swing damping performance deteriorates. However, all three scenarios shows rapid improvement compared to response without suspended load controller.

solved by delaying the activation of the delayed feedback controller. It should be activated after the initial swings are avoided by the input shaper.

5.8 Simulations - Software in the Loop (SITL)

The simulation study presented in the section above shows promising results. However, the simulation study is based on a different multirotor controller than the multirotor system used in this thesis. Therefore, a Software in the Loop (SITL) simulation, where the behaviour of the multirotor autopilot should be tested.

The SITL is delivered by Ardupilot. It runs the ArduCopter code, which is the autopilot of the multirotor system. It can be run without any hardware (*APM open source autopilot*), thus reducing complexity and problem sources. Consequently, the multirotor behaviour and the controllers can be implemented and tested in a safe environment.

It is important to note that the SITL simulations simulate multirotor behaviour and does not include the suspended load dynamics. Consequently, the performance of the suspended load controller can not be examined before the suspended load dynamics are added. MATLAB is used to simulate the load dynamics and is added in the control loop as shown in Figure 5.16.

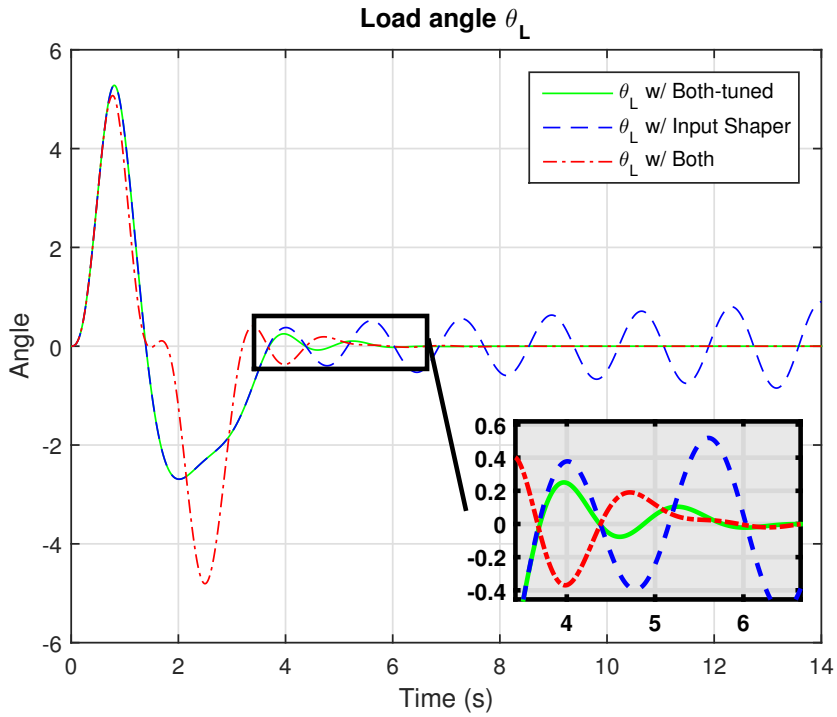


Figure 5.15: Resulting load angle responses given the reference signal shown in Figure 5.13. The performance deteriorates when both controllers are active. Problem is solved by activating the delayed feedback controller after 3.5 seconds.

This section will first introduce the controller setup where SITL, MATLAB and DUNE is used to setup the simulations. The result of the setup and simulations will be discussed.

5.8.1 Software in the Loop (SITL) - Simulation Setup

The controller scheme for the SITL study with added suspended load dynamics and controllers are shown in Figure 5.16. Note that the load dynamics are implemented in MATLAB and connected to the system using IMC communication protocol. The multirotor dynamics and autopilot behaviour is simulated in SITL platform, while the suspended load controllers are implemented in the DUNE platform.

The setup is very similar to the introduced simulation study in Section 5.7.1. However, the way reference signal is utilized by the autopilot is remarkably different. This section will discuss the reference model and the autopilot design. The rest of the setup is the same as introduced in Section 5.7.1.

Reference Model

The implemented reference model is based on the second order system introduced in Section 2.3.1 with $\omega = 0.1$ and $\xi = 0.9$. However, the multirotor autopilot accepts only velocity reference as input. Thus, the reference velocity is redesigned, such that it includes the reference position and obtains the same form as introduced in Section 2.3.1. However, it is essential to understand how the autopilot is utilizing the reference. Therefore, a brief explanation about the autopilot will first be introduced.

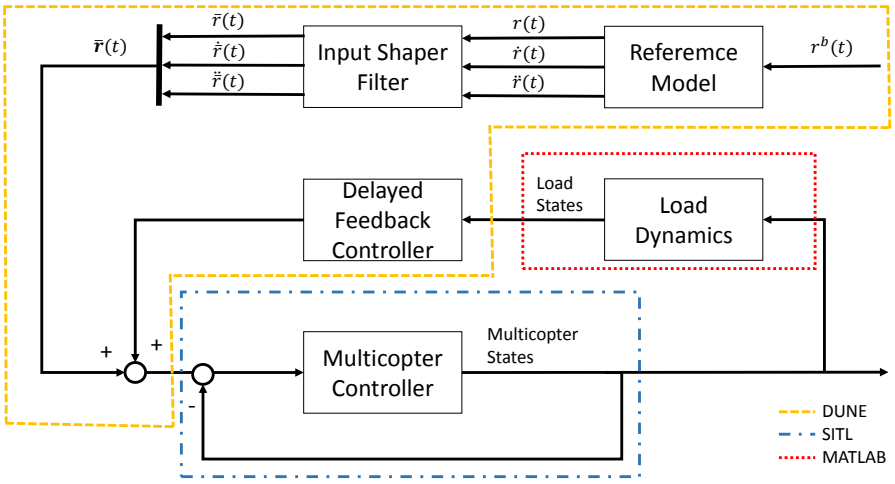


Figure 5.16: Control scheme for suspended load system used in the SITL study. The reference model, input shaping filter and the delayed feedback controller is implemented in DUNE. The load dynamics are implemented in MATLAB and the multirotor behaviour is simulated using SITL. The communication protocol between different parts are based on IMC.

The autopilot is based on double integrator (*APM open source autopilot*)

$$m\ddot{\mathbf{x}} = \tau \tag{5.33}$$

$$\tau = \mathbf{a}^d - k_p(\mathbf{v} - \mathbf{v}_{ref}) \tag{5.34}$$

where m is the weight, τ is the control input, $\mathbf{v}_{ref}$ is the velocity reference from the DUNE, and the acceleration $\mathbf{a}^d$ is calculated by numerical integration of the velocity. Consequently, the reference velocity can be derived such that,

$$\mathbf{v}^d_{ref} = \mathbf{v}^d - k_p(\mathbf{p} - \mathbf{p}^d) \tag{5.35}$$

where the superscript d denotes desired value, $\mathbf{p}$ and $\mathbf{v}$ is the position and velocity reference from the 2nd order reference model. k_p is just a tuning parameter. Thus the double integrator in (5.33) becomes,

$$\tau = \mathbf{a}^d - k_d(\mathbf{v} - \mathbf{v}^d) - k_p k_d(\mathbf{p} - \mathbf{p}^d) \tag{5.36}$$

Now, consider

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^d \tag{5.37}$$

thus (5.36) can be rewritten,

$$\ddot{\tilde{\mathbf{x}}} + k_d\dot{\tilde{\mathbf{x}}} + k_p\tilde{\mathbf{x}} = 0 \tag{5.38}$$

Note that this have the same form as the 2nd order reference model introduced in Section 2.3.1 and Equation (2.24). Consequently, the response should be equal as well, resulting in,

$$\lim_{t\to\infty} \tilde{\mathbf{x}}(t) = 0$$

asymptotically stable system.

That being said, some implementation issues regarding the experiment design needs to be addressed.

5.8.2 Software in the Loop (SITL) - Implementation Issues

Experiments using the control scheme shown in Figure 5.16 was somewhat more problematic than expected. For instance, the suspended load dynamics are simulated based on numerical integration. It is dependent on high frequency update. However, the frequency during the SITL was not sufficiently high.

The main limitation was due to low output rate from the autopilot. The Pixhawk output rate was about 25Hz. Moreover, the communication between MATLAB and DUNE based on IMC protocol has proven to be too slow (about 10Hz). Furthermore, the computational power of the computer was pushed to the limits; running SITL, DUNE, MATLAB and Neptus at the same time. Therefore, the controllers could not be extensively studied before actual experiments. However, the SITL simulations were ran using dummy values for the load angles. In that way, the controller behaviour could be tested to some extent.

In addition, the frequency of the reference model update was limited to 25Hz due to limitations in Pixhawk I/O rate. However, the Pixhawk inner loop runs in 400Hz. Consequently, the controller model introduced in Equation (5.36) actually runs in 2 different rates: 25 Hz for the position and velocity reference, and 400Hz for the acceleration part. The effect of this difference is not studied and assumed to be negligible.

Moreover, Equation (5.36) actually contains an integrator part which is neglected in the analysis. The integrator part does not change the stability properties of the 2nd order system. However it is not compensated or accounted for in the input shaping filter. Remember from Section 5.4 that 10% frequency error led to noticeable performance deterioration(Figure 5.7). Consequently, the effect of the integrator part on the total response is not accounted for.

In the next section, the system was ran on the payload computer to confirm that the code was not computationally expensive. The actual in-flight experiments will be introduced in Section 5.10.

5.9 Hardware in the Loop (HITL) Verification - Load Dynamics

The load dynamics could not be simulated in the SITL simulations due to low sampling frequency, and is discussed in Section 5.8.2. However, the suspended load controller and all the hardware is implemented and experimented with in this section. Mainly to verify that computations are not too computationally expensive for the payload computer.

Figure 5.17 shows the result. All the components except the multirotor motors are run by the payload computer. The data from DUNE ran in BBB is sent to Neptus using IMC protocol. Moreover, MATLAB was used to plot the multirotor orientation and the load in real time since Neptus do not have support for the load dynamics.

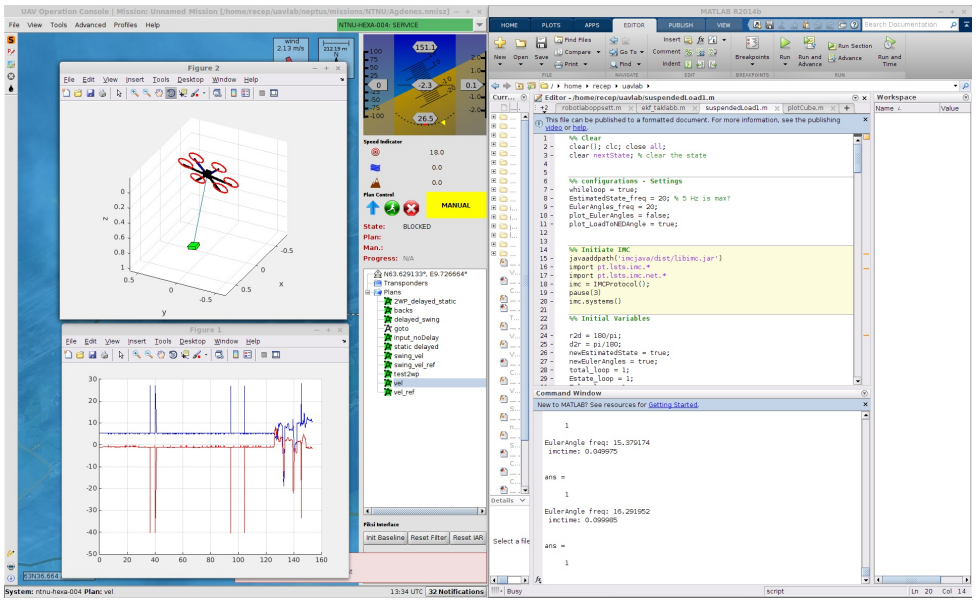


Figure 5.17: The figure shows DUNE, Neptus and MATLAB interaction. The sensor data from the payload is plotted and shown in Neptus. MATLAB is used to plot the load together with the copter in real time. The figure in the down left side shows the load angle in the NED-frame where θ_L is shown in blue and ψ_L is shown in red.

5.10 Outdoor Experiments - Verification

Several in flight experiments were conducted at five different occasions at Agdenes airfield. Note that the multirotor system at this stage complex and consists of many dependent hardware and software components. Figure 5.18 shows the multirotor with the custom made payload mount, the Atmel kit, Piksi system with the helix antennas, GPS antenna, the decoders that measure load angle, the Picostation and the 3DR antenna. Also note that the legs are expanded to make room for the sensors below.

Moreover, no similar experiments have been conducted at the department. Consequently, many unexpected operational obstacles had to be overcome and lots of new experience gained.

This section will discuss the main encountered problems before introducing the experiment results.

5.10.1 Encountered Problems - In-Flight Experiments

Several minor failures and crashes throughout this thesis have caused minor and major setbacks. Although the physical damage was non-critical, debugging was time consuming due to complexity of the system. The multirotor consists of many different, but dependent hardware and software components, that is developed by several students.

For instance, Figure 5.19 shows one of the early crashes. The crash was caused due to low battery power and failed fail-safe warning function. Even though the problem is not directly

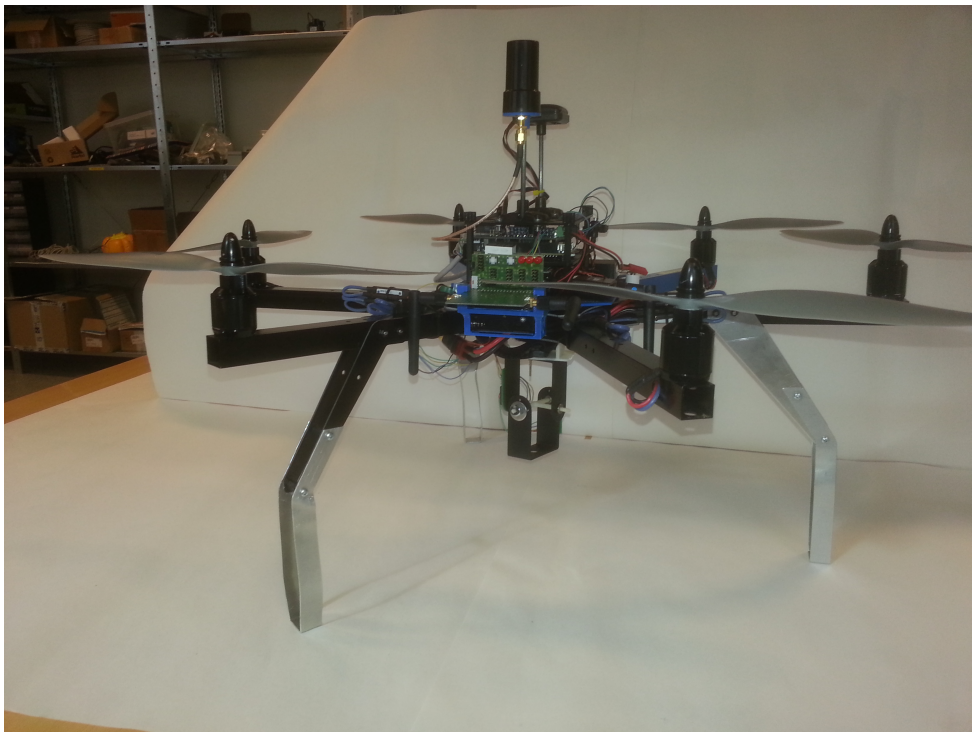


Figure 5.18: The hexacopter is used during the suspended load experiments. The suspended load is attached at the sensors below the multirotor system. The legs are elevated to protect the sensors during take-off and landing.

related to this thesis, operational errors also caused setbacks. Luckily all the hardware components except the plastic payload mount survived. The multirotor frame was also changed due to minor misalignment on the frame.

Therefore, many hours should be set aside for operational errors during an experiment in the future. Most of these problems could, and was fixed within hours. However, due to long travel time and distance between the labratory and the airfield, even small errors resulted in major setbacks and many hours of extra work.

Hardware Error - Loose Cable

One of the major setbacks was due to weird multirotor behaviour. The autopilot would sometimes stop sending information to DUNE, meaning that the data would not be logged. This error was often followed by a sudden movement that often led the multirotor to flip over and crash. There was no apparent reason for this kind of behaviour. The problem was triggered at random times and never found until a major crash. Analysis of the autopilot data showed that the multirotor would stop tracking the desired roll value, which resulted in sudden jump in desired roll value. The data is shown in Figure 5.20. Consequently, it was concluded that the autopilot was bugged since the behaviour did not occur on the other multirotor project at the department.

Therefore, the Pixhawk was replaced, but the problem remained. It was later found out that one of the rotor motor cables was loose. Consequently, it would sometimes loose connec-

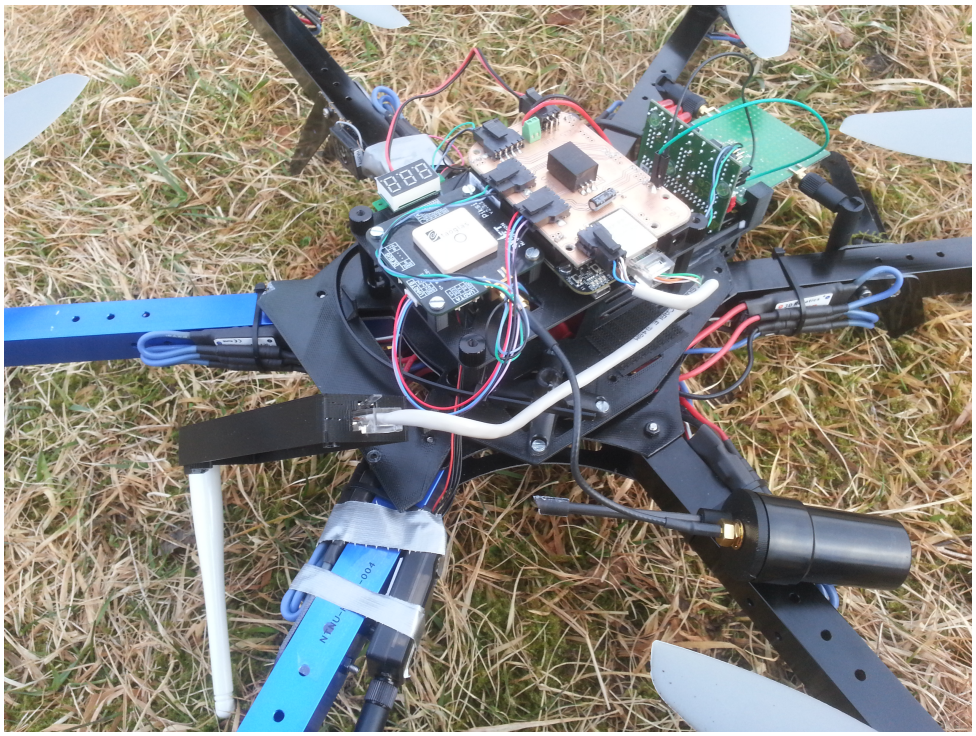


Figure 5.19: The Picture shows damages on the copter from one of the early experiments. The crash was caused by low battery power and failed fail-safe warning. The hardware survived, but the payload plate, including the masts for the GPS antennas got broken.

tion, which the autopilot unsuccessfully tried to counteract.

Software Error - Race Condition

Another major setback was due to problems regarding plan execution. The autopilot would not change its mode from *loiter* to *guided* when a command was sent. Or did so in a very delayed fashion. This problem was suspected to be due to ineffective code implementation. Although the *cpu* data showed no abnormalities, there was no memory information. However, the problem remained after the code was reimplemented in a more efficient way. It was later unveiled a small error in the configuration file, which triggered a race condition. Consequently, the problem occurred randomly and was difficult to locate.

5.10.2 Results - In-Flight Experiments

Regardless of setbacks discussed in Section 5.10.1, many in-flight experiments with the suspended load system was conducted. The experiments took place outside at Agdenes airfield. Unfortunately the weather conditions at Agdenes was often windy. Consequently, the environmental changes between the tests was uncontrollable and significant. Figure 5.21 shows how much the wind could affect the load and the multirotor when it was set to loiter. However, the multirotor was still stable and controllable.

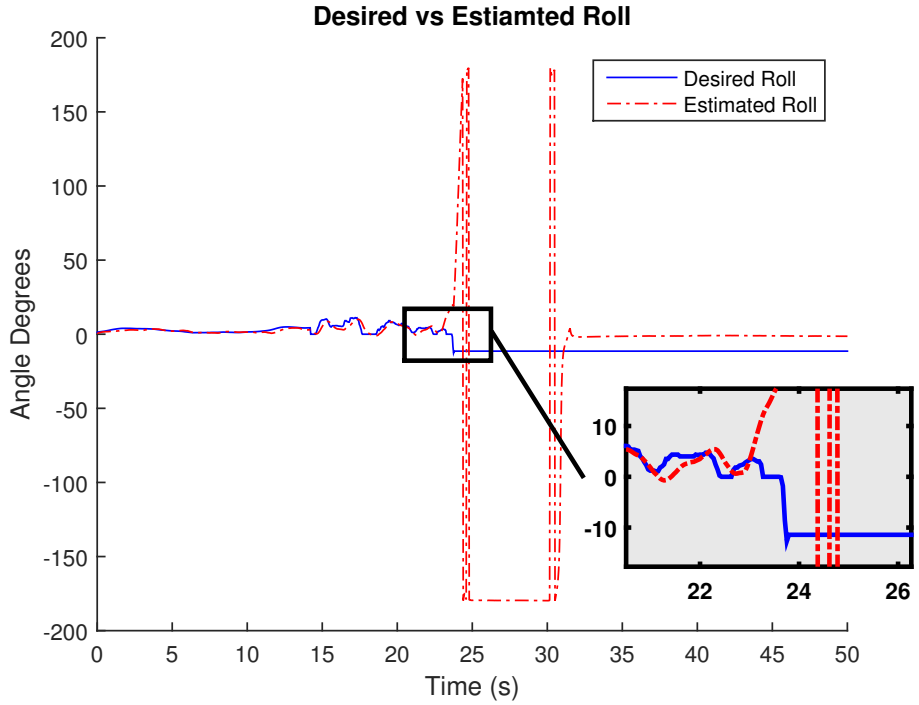


Figure 5.20: The unexpected change in desired roll led the multirotor to flip and crash. The log shows that the multirotor system is not able to track the desired roll. The problem is related to hardware error, where one of the rotor cables was loose. Consequently, the autopilot was trying to counteract by giving high desired roll value on the opposite direction.

Swing Reduction While Following A Path

Figure 5.22 shows the path of three different experiments with suspended load. A load that weights 0.4 kg was attached 0.7m below the multirotor system as shown in Figure 5.21. The multirotor system is started from loitering mode about the same point and set to track the same WP. The plan generates a reference signal to reach the WP which is fed to the autopilot as described in Section 5.8.1. Note that the multirotor system is having hard time following a straight path to the WP. The reason is combination of strong wind casts and low operational speed. The speed was kept low during experiments for safety reasons. Thus, the multirotor is not fully able counteract the effects of wind during an operation. The problem could be avoided if the experiments could be conducted indoors.

Figure 5.23 shows the load angle response from the experiments. The input shaping method seems to work properly and reduce initial swing excitation. However, the performance of the controllers are difficult to determine due to severe external influences. The wind changed the conditions between the experiments and during each experiment. Moreover, the effect of the wind generated from the rotors above the load was not accounted for. The wind from the rotors pushed the load to the middle, and damped swing amplification.

Consequently, the performance of the control design is inconclusive. Therefore, the focus was changed to loiter and reduce swings induced by the wind. Performance o the delayed feedback control is discussed in the following section.

Figure 5.21: The hexacopter in the air with suspended load. Strong wind casts could push the system and load away. However, the multirotor controller was able to stabilize the system.

Swing Reduction In Loiter Mode

The multirotor was set to loiter. The wind caused the load to swing, while the delayed feedback controller was activated and supposed to damp the induced swings. The corresponding load angle response are shown in Figure 5.24. At first glance, the delayed feedback controller seems to work satisfactory. However, the conditions between the experiments changed radically and could not be controlled. Thus the difference might be due to significant changes in the environment.

Figure 5.25 shows another experiment set with the same setup. However, the performance of the delayed feedback controller does not seem as significant as in Figure 5.24. Consequently, without an indoor experiment where the external forces can be controlled, another way of determining performance of the suspended load controller is necessary.

Swing Reduction - Performance Analysis

A set of measurements will be used to determine if the load swing is reduced with the delayed feedback controller. Table 5.4 shows the mean load angle value from six different experiments where 300 samples have been gathered at each experiment with and without delayed feedback controller. Although the results indicate that the delayed feedback controller have some effect, the foundation of this claim is thin.

The results can be disproved considering the autopilots ability to sufficiently track a desired reference command. Remember from Section 5.2 that it is assumed that the multirotor au-

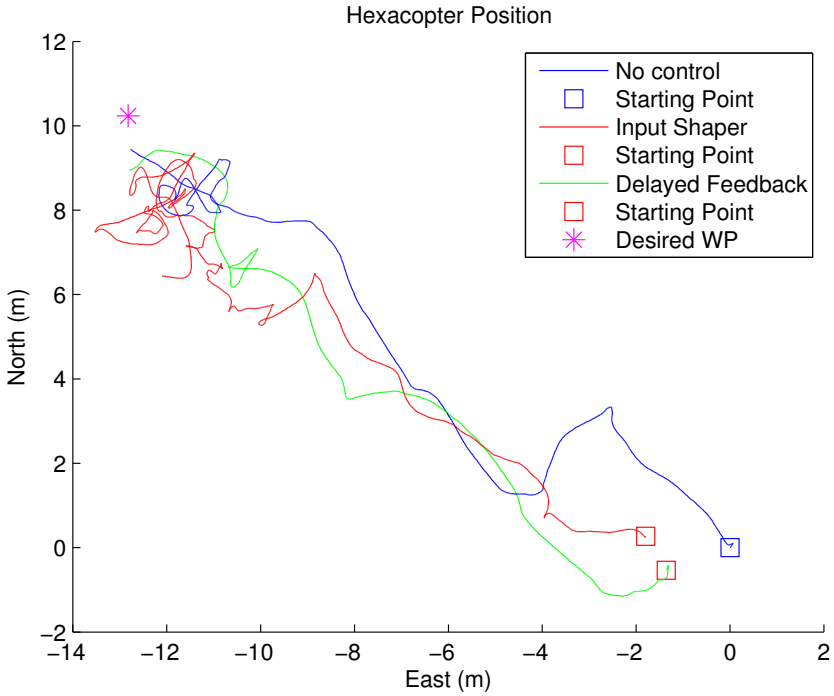


Figure 5.22: The executed and flown flight plan by the Hexacopter. The Hexacopter is started at the square marks and set to follow the WP shown. The figure shows the path flown using the velocity controller with and without the shaped reference model, and delayed feedback control.

Table 5.4: Mean load angle from six different experiments where 300 samples have been gathered at each experiment with and without delayed feedback controller

	Mean Standard Deviation: ϕ_L	**Mean Standard Deviation: ϕ_L**
No Controller	7.9464	8.1548
Delayed Feedback	3.7989	6.5505

topilot is able to sufficiently track a given reference trajectory. However, Figure 5.26 shows that there is an extensive difference between the reference signal and the actual velocity of the multirotor system. The difference between the reference velocity (Equation (2.26)) and the desired velocity (Equation (5.35)) are small. Moreover, Figure 5.27 shows extensive difference between the reference position and the estimated position, which is the best estimation of the vehicle position.

Therefore, it can be concluded that the multirotor autopilot is not able to sufficiently track a given reference command. However, it should be added that this is true for low velocity commands. Moreover, the experiments were tested in a difficult environment, where the effects of wind was significant.

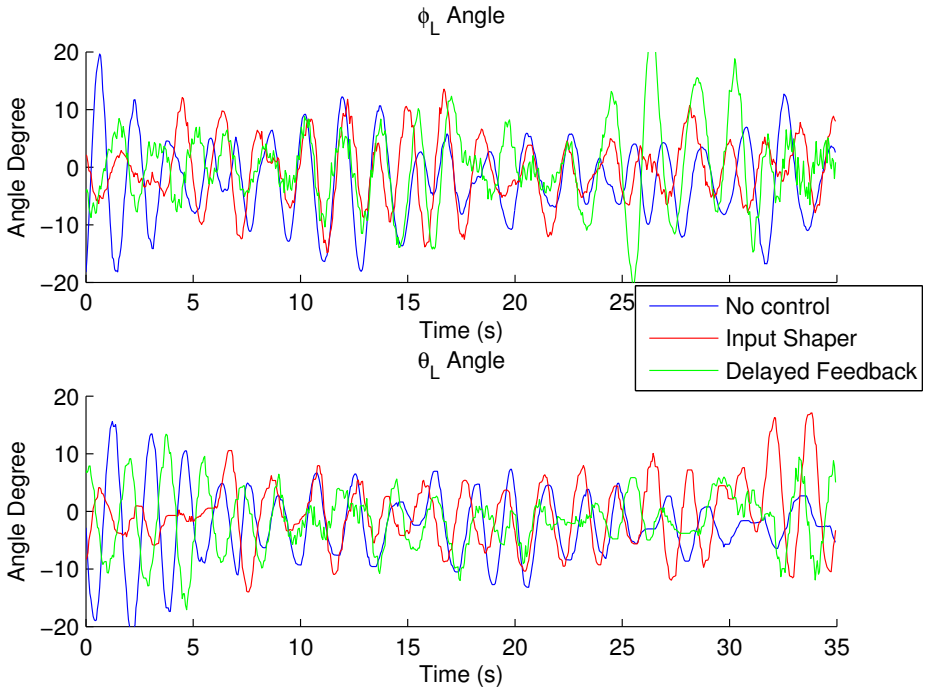


Figure 5.23: Resulting load angle from the operation described in Figure 5.22. Note that the input shaping in fact reduces the intial swing excitation. However, the performance of the delayed feedback controller is poor. More tuning is required.

5.11 Discussion

The simulation study shows promising results. However, in-flight experiments were inconclusive. Mainly since the autopilot was not able to track the given reference signal. That being said, the experiments were conducted at Agdenes airfield which was very windy during most of the flights. Moreover, the operational speed was kept low for safety reasons. Therefore, a more extensive testing in a controllable environment is necessary. In addition, the experiments were based on traditional GPS and not a high precision navigation system.

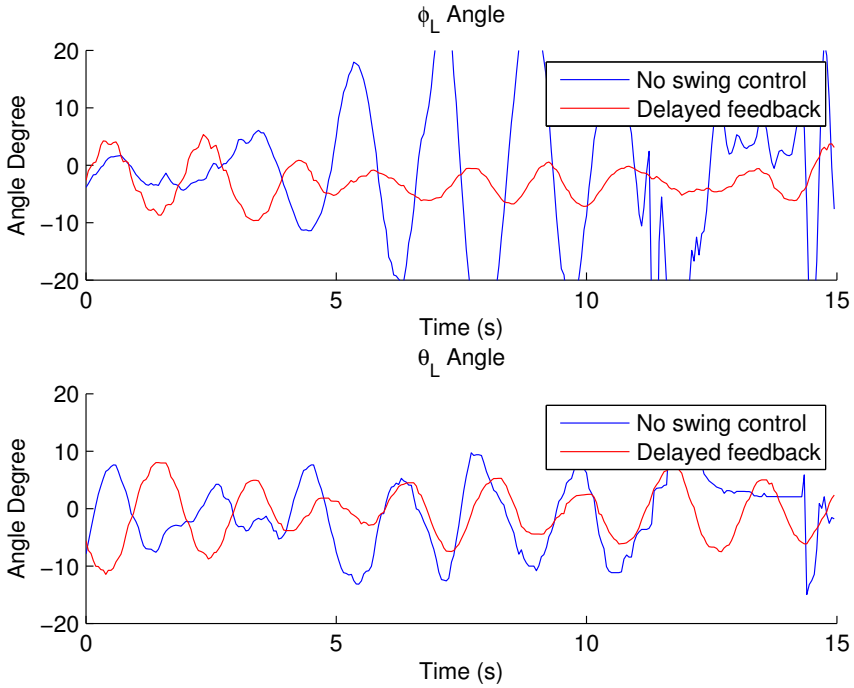


Figure 5.24: Load angle response from static flight with and without feedback controller. External changes affect the result if the experiment is compared to Figure 5.25 which is the same setup.

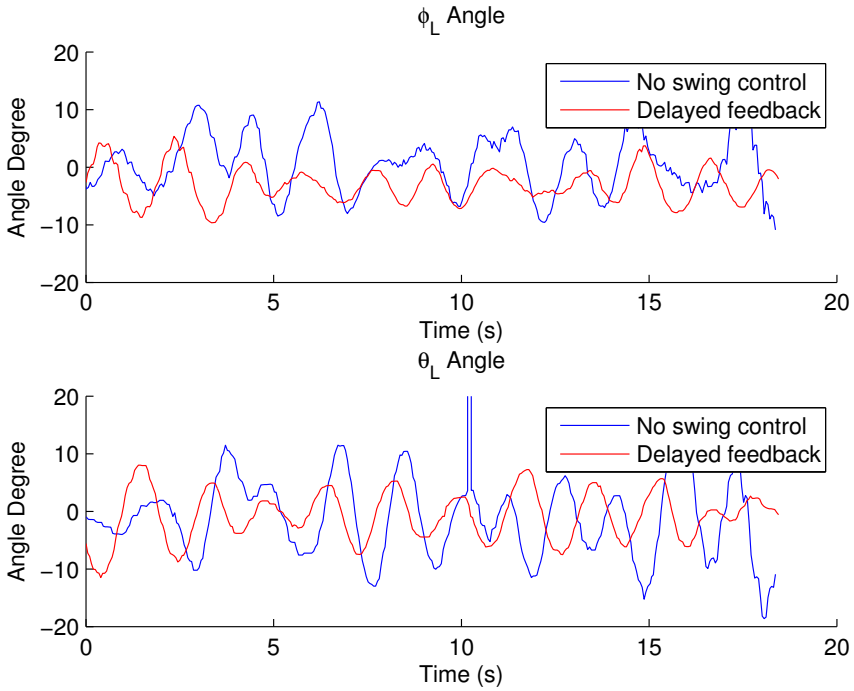


Figure 5.25: Load angle response from static flight with and without feedback controller. External changes affect the result if the experiment is compared to Figure 5.24 which is the same setup.

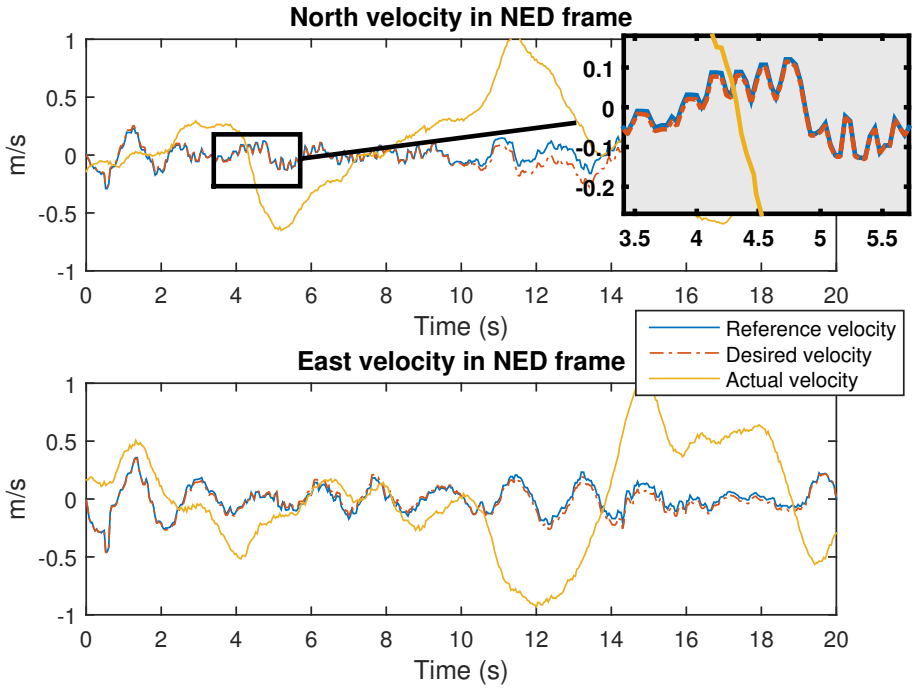


Figure 5.26: Figure shows the difference between the desired velocity fed to the autopilot (5.35), the calculated velocity reference (2.26) and the estimated velocity of the multirotor.

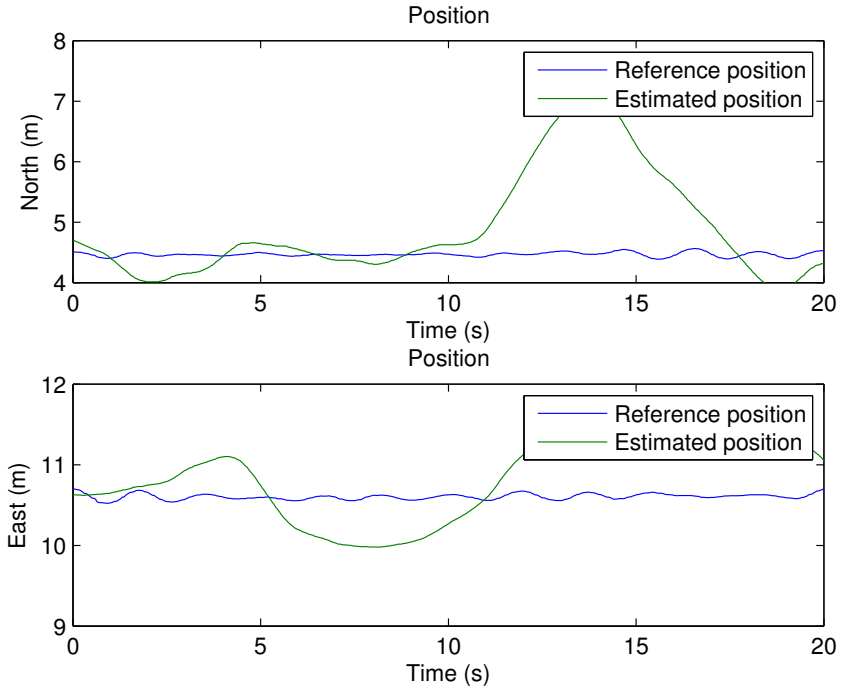


Figure 5.27: Figure shows the difference between the reference position and the estimated multirotor position. The autopilot tracks the position poorly.

Chapter 6

Conclusion and Closing Discussions

The goal of this thesis was to develop a high precision navigation system for indoor applications and a control system for multirotor suspended load systems. Three main objectives was investigated,

- A complete system overview of a multirotor platform with all the necessary components and software.
- A low-cost, light weight, easy to deploy and robust navigation system providing necessary precision for indoor operations and applications where traditional GPS systems may not be suitable.
- Suspended load control suitable for accurate pickup and placement operations, including swing free trajectory tracking.

An indoor navigation system was developed based on distance measurements from radio transceivers. The target tracking problem was solved based on principle of trilateration. The EKF was used as the solving algorithm with sensor fusion approach to filtering. Extensive experiments with the radio transceivers was conducted, such that the best setup and configurations was found. The problem of multipath was discussed in detail, and several proposals for improvements was given. Moreover, the measurement frequency of the radio tranceivers was boosted by over 20 times, by changing the output protocol.

The solution of the EKF algorithm was based on Position model (P), Position-Velocity model (PV) and Position-Velocity-Acceleration model (PVA) models, where altimeter and IMU data was integrated. The P model was smooth and gave best estimations for low-speed targets, while the PVA model had the most potential with good tuning, higher measurement frequency and integration of the available sensors on-board the multirotor system. Moreover, an outlier detection and rejection algorithm was implemented and improved the solution. The robustness of the system was improved by adaptive tuning of the measurement covariance based on fuzzy logic.

The algorithm was tested in MATLAB simulations and in-flight experiments, both indoor and outdoor. High precision positioning systems, such as RTK based navigation system for outdoor and OptiTrack motion capture positioning system for indoor experiments was used as reference. It was shown that the system has potential and can deliver submeter accuracy.

The suspended load problem was solved introducing a control scheme that was independent of the multirotor autopilot design. It was assumed that the load was sufficiently light and small, and that the autopilot was able to follow a given reference. Therefore the autopilot was treated as a black box most parts of this thesis. An easy to implement control design was introduced. The control design included both feedforward and feedback control. The so-called input shaping filter was utilized to generate zero-vibrational commands, while a delayed feedback control based on load angle measurements was used to damp oscillations. Also a tunable reference model for target tracking was introduced.

The suspended load control was tested in MATLAB - and SITL simulations, and HITL experiments. Moreover, several in-flight experiments was conducted. The results are presented and discussed. While simulation study shows promising results, the real world-application was inconclusive due to limitations on the autopilots ability to track a reference trajectory.

6.1 Future work

Several issues need to be addressed before the system can be classified as fully operational. First, the hardware used in the development of the high precision indoor positioning system is operating in the same frequency band as the GCS communication. This causes the measurement frequency to drop considerably, while the communication between multirotor and GCS becomes unstable. Therefore, one of the systems should be changed to a hardware that is based on another frequency band. *UPosition IR-UWB Module* by BeSpoon (BeSpoonASA 2015), is suggested as an alternative to the Atmel system utilized in this thesis. However, the Picostation M2 HP, which is the antenna used for GCS communication, may also be changed as it caused severe interference problems to the RTK based navigation system, discussed in (Røli 2015).

Moreover, even with the improved source code, the Atmel kit delivers a frequency of about 25 Hz. Consequently, the equipment may not be appropriate for high speed target tracking.

The multirotor autopilots (Pixhawk) I/O frequency also cause some limitations. It is able to give I/O rate up to 50 Hz. However, due to bugs in the software, 25Hz rate was used throughout this thesis. This rate is too low if the IMU data will be used in high speed operations, which is discussed in (Spockeli 2015). Moreover, the autopilot does not seem to be able to track a reference signal sufficiently enough in hoover and slow speed operations. However, the experiments was conducted based Pixhawk standard GPS setup and in windy environments. The autopilot should be tested with a high precision navigation system, and preferably indoor.

Appendix A

Hardware Specifications

A.1 ArduCopter 3DR Hexa B

The 3DR Hexa B is a 6 armed multicopter produced by 3D Robotics at headquarters in San Diego, California(3D Robotics 2015c). It consists of 6 aluminum arms, connected by two fiberglass frame boards. Its specifications are summarized in Table A.1

Table A.1: Technical specifications of the 3DR Hexa B.

Motors	6x AC2830-358 850kv
Propellers	6x plastic 11 x 47 (length x pitch)
Diameter	590 mm
Weight without payload	1550 g
Lift capacity	>1.0 kg

A.2 Pixhawk

Pixhawk is an autopilot system designed by the PX4 open-hardware project and manufactured by 3D Robotics (3D Robotics 2015e). In addition to extensive interface options, included multithreading capabilities, high processing power and Unix/Linux like programming environment, the system also includes several sensors. The list of the included interfaces, sensors and microprocessors are shown in table A.2.

A.3 BeagleBone Black

The BeagleBone Black is simply a low cost development platform for developers (BeagleBoard 2015). It can run several Unix based platforms, android, Cloud9 IDE etc. However, for the UAV purposes, GLUED, which is introduced in section 3.3.2, is used. The BeagleBone Black contains:

Table A.2: Pixhawk specifications

Microprocessor:
32-bit STM32F427 Cortex M4 core with FPU
168 MHz/256 KB RAM/2 MB Flash
32 bit STM32F103 failsafe co-processor
Sensors:
ST Micro L3GD20 3-axis 16-bit gyroscope
ST Micro LSM303D 3-axis 14-bit accelerometer / magnetometer
Invensense MPU 6000 3-axis accelerometer/gyroscope
MEAS MS5611 barometer
Interfaces:
5x UART (serial ports), one high-power capable, 2x with HW flow control
2x CAN
Spektrum DSM / DSM2 / DSM-X® Satellite compatible input up to DX8 (DX9 and above not supported)
Futaba S.BUS® compatible input and output
PPM sum signal
RSSI (PWM or voltage) input
I^2C®
SPI
3.3 and 6.6V ADC inputs
External micro USB port

- AM335x 1GHz ARM® Cortex-A8 processor
- 512MB DDR3 RAM
- 4GB 8-bit eMMC on-board flash storage
- 3D graphics accelerator
- NEON floating-point accelerator
- 2x PRU 32-bit microcontrollers

and has the following connectivity options:

- USB client for power & communications
- USB host
- Ethernet
- HDMI
- 2x 46 pin headers

A.4 PicoStation M2 HP

The PicoStation M2 HP is a compact, but powerful radio used as the main communication link in this project. With one unit configured as an access point and the other as clients, all the units can communicate seamlessly with each other through Ethernet (IP).

The full specifications for the PicoStation can be found in the datasheet (Ubiquiti-Networks 2015), while some are summarized in Table A.3.

Table A.3: Specifications for PicoStation M2 HP.

Networking Interface	10/100 Ethernet Port
Operating frequency	2412 - 2462 MHz
Antenna	External, 2 dBi Omni Antenna
Indoor range	Up to 200 m
Outdoor range	Up to 500 m
Power Method	Power over Ethernet
Weight	0.1 kg (with housing)

A.5 Atmel REB233SMAD-EK Kit

The REB233SMAD-EK consists of the Atmel 2.4GHz transceivers AT86RF233 combined with the Atmel ATxmega256A3 microcontrollers and is shown in figure A.1. The radio transceiver features hardware accelerators for IEEE 802.15.4 signal processing and more advanced features like AES encryption, high data rate modes and antenna diversity, where the main feature is error rate measurement. However, the most interesting part during this thesis was the range measurement feature, which was not supported in the SMAD-EK version. Therefore the firmware is downgraded to the RTB version (RTB v.1.1.7), where the main feature is range measurement. The essential applications for this project will be described in the following subsections.

A.5.1 Atmel AVR2150: RTB Evaluation Application

The RTB Evaluation Application is operated on the Atmel REB233SMAD-ek hardware platform. The main feature is to evaluate the AT86RF233 radio transceiver Phase difference Measurement Unit (PMU). In contrast to the SMAD-EK version, the RTB provides a platform to:

- Demonstrate AT86RF233 phase difference measurement technology
- Evaluate the AT86RF233 radio transceiver performance
- Test the enhanced radio transceiver feature set

The features makes it possible to evaluate the range measurements accurately by analysis of the extensive signal data. The performance of the system is discussed in chapter 4 . Addi-

Figure A.1: The REB233SMAD based hardware platform. *Image by courtesy of atmel.com*

tional performance information, more extensive description of the application and how to downgrade firmware can be found at (Atmel 2015c) and (Atmel 2015b).

A.5.2 Ranging Application

The sequence diagram of the ranging application is shown in Figure A.2. Assuming the transceivers are configured and initialized correctly, they will be set to a state where they will be waiting for range requests. The ranging request is sent from the Coordinator, which is connected to a base station or terminal application as shown in Figure 4.1. Note that the Coordinator is only initiating measurements and does not measure the distances to itself (remote ranging). The Coordinators ranging request is sent alongside with the ranging parameters and Reflector address to the Initiator. The following sequence is performed after the Coordinator has sent a request:

1. Initiator receives ranging request from Coordinator.
2. Initiator sends ranging command to Reflector.
3. Initiator sends a timing synchronization frame to Reflector.
4. Reflector sends a ranging measurement start frame back to Initiator.
5. Ranging measurement between Initiator and Reflector.
6. Initiator requests measurement data from Reflector.
7. Reflector sends measurement data to Initiator.
8. Initiator calculates distance.
9. Initiator sends distance data to Coordinator.

The transferred data to the Coordinator contains the Initiator and Reflector address, weighted distance in cm and wighted Distance Quality Factor (DQF) in %. It is also possible to obtain unweighted distance for each antenna separately, in total of 4 measurements, with the corresponding DQF. The amount of Initiators in the sequence can be configured, while the amount of Reflectors on is bound to one in the given application. However,some changes such as bypassing the coordinator, changing the message protocol to SBP and adding more reflectors which is directly connected to DUNE has been made. Theese changes are discussed in 4.

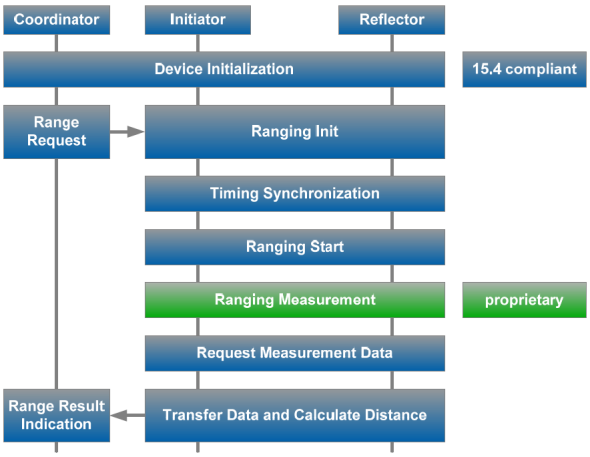


Figure A.2: Ranging application sequence diagram. *Figure by courtesy of atmel.com*

Ranging parameters

The application allows the user to make modification of system and ranging measurement parameters. These parameters with their default values are listed below:

- Channel: 26
- Own Short Address: 0x0000
- Initiator Short Address for Remote Ranging: 0x001
- Initiator Long Address for Remote Ranging: 0x00042FFFF175C7D
- Reflector Short Address: 0x0002
- Reflector Long Address: 0x000425FFFF175C9D
- PAN-Id: 0xCAFE
- Ranging Addressing Scheme: 0
- Coordinator Addressing mode: 2 - Short addresses
- Filtering length during continuous Ranging: 5
- Filtering method for continuous Ranging: 0 - Average if distance and DQF
- Default Antenna: 0

- Antenna Diversity: 1
- Provide all Measurement Results: 0 - The final result set only displayed
- Apply Minimum Threshold during weighted Distance Calculation: 1
- Frequency Start: 2403 - Ranging measurement start frequency in MHz [2324 ...2527]
- Frequency Step: 2 - 2MHz
- Frequency Stop: 2443 - Ranging measurement start frequency in MHz [2324 ...2527]
- Verbose level: 0
- Tx Power during Ranging: -17 - Tx power in dBm during ranging measurement cycle [-17...4]
- Provide Ranging Tx for next Ranging: own power is applied to peer node.

More detailed information about these parameters can be found in (Atmel 2015a). The default values are used if nothing else is mentioned during the project.

A.5.3 Antenna Configuration

The antenna configuration is of special interest, due to high sensitivity of the system to their configuration. Atmel recommends an angle of 90° between the two swivel antennas for the best result indoors. This recommendation is tested in Section 4.2.2. The recommended configuration can be achieved by either of the following descriptions:

- 'open arm' - each antenna 45° upwards or
- 'one up' - one antenna straight

A.6 Piksi

Piksi is a low-cost, high-performance GPS receiver with RTK functionality for centimeter-level relative positioning accuracy produced by Swift Navigation. It is capable of tracking 14 satellites and calculating PVT solutions at up to 50 Hz. While the current firmware (v 0.15) is only capable of utilizing the L1 GPS code, the hardware is also capable of receiving Galileo and GLONASS signals.

To calculate a baseline, Piksi needs to compare phase measurements from a minimum of 5 satellites with those from another receiver.

Below is a brief overview of some of Piksi's specifications. More details can be found in the Piksi datasheet (SwiftNAV 2015b).

A.6.1 System Architecture

The Piksi receiver architecture consists of three main components:

- An RF **front-end** that downconverts and digitizes the RF signal from the antenna.
- An **FPGA** running SwiftNAP which performs basic filtering and correlation operations on the digitized signal stream.
- A **microcontroller** which controls SwiftNAP and programs the correlation operations, collects the results, and processes them all the way to PVT solutions.

A.6.2 Connections

UART Two UARTs providing serial interface that can transmit NMEA-0183 messages or send/receive massages in the Swift Nav Binary Protocol (SBP).

USB A Micro-USB socket, configurable as a USB-Serial bridge to the microcontroller (functioning identically to the two dedicated UARTs) or as an interface to SwiftNAP. With a built-in bootloader, new firmware can easily be added.

External antenna An SMA input featuring a switchable 3.3 V bias can support an active antenna drawing a maximum of 57 mA.

In addition to this, Piksi has a micro JTAG header and a debug/expansion header.

A.7 OptiTrack

The OptiTrack is the worlds largest provider of optical motion capture technology (NaturalPoint 2015). They offer tracking applications for several purposes, such as film, games, sports training and biomechanics. It also provides object tracking in 6 DoF with sub-mm precision and with support for real-time and offline data analyzing.

The software used in this thesis is the Motive:tracker, which provides up to 120Hz with sub-mm precise positioning information in 6 DoF and orientation of the tracked object (OptiTrack 2015b). The software is used together with motion capture cameras that was set up and installed at the Department of Engineering Cybernetics, NTNU. During the thesis, 16 Flex 13 cameras was used (OptiTrack 2015a).

Bibliography

3D Robotics (2015a). *3DR-radio*. URL: http://copter.ardupilot.com/wiki/common-optional-hardware/common-telemetry-landingpage/common-3dr-radio-version-2/ (visited on 06/07/2015).

— (2015b). *ArduCopter*. URL: http://copter.ardupilot.com/wiki/introduction/ (visited on 06/07/2015).

— (2015c). *Arducopter 3DR Hexa B*. URL: https://pixhawk.org/platforms/multicopters/3dr_arducopter_hexa_b (visited on 06/07/2015).

— (2015d). *Mission Planner*. URL: http://planner.ardupilot.com/ (visited on 06/07/2015).

— (2015e). *Pixhawk Autopilot System*. URL: http://3drobotics.com/pixhawk-autopilot-system/ (visited on 06/07/2015).

Andersen, Håvard Lægreid (2014). "Path Planning for Search and Rescue mission using multicopters". MA thesis. Norwegian University of Science and Technology.

ArduPilot. *APM open source autopilot*. URL: http://dev.ardupilot.com/wiki/where-to-get-the-code/ (visited on 05/28/2015).

Atmel (2015a). *Atmel AVR2150: RTB Evaluation Application - User's Guide*. URL: http://www.atmel.com/Images/Atmel-8441-RTB-Evaluation-Application-Users-Guide_Application-Note_AVR2150.pdf (visited on 06/07/2015).

— (2015b). *Atmel AVR2151: RTB Evaluation Application - Quick Start Guide*. URL: http://www.atmel.com/Images/Atmel-8442-RTB-Evaluation-Application-Quick-Start-Guide_Application-Note_AVR2151.pdf (visited on 06/07/2015).

— (2015c). *REB233SMAD-EK*. URL: http://www.atmel.com/tools/reb233smad-ek.aspx (visited on 06/07/2015).

Balchen, Jens G., Trond Andresen, and Bjarne A. Foss (2003). *Reguleringsteknikk*. Norges teknisk-naturvitenskapelige universitet (NTNU). ISBN: 8247151472.

BeagleBoard (2015). *BeagleBone Black*. URL: http://beagleboard.org/black (visited on 06/07/2015).

BeSpoonASA (2015). *UPosition IR-UWB Module - Generic*. URL: http://spoonphone.com/en/home/12-uposition-ir-uwb-module-basic.html (visited on 06/07/2015).

Bisgaard, Morten (2007). *Modeling, Estimation, and Control of Helicopter Slung Load System*.

Biswas, Joydeep and Manuela Veloso (2012). "Depth camera based indoor mobile robot localization and navigation". In: pp. 1697–1702.

Calado, Pedro et al. (2008). "Implementation of a Control Architecture for Networked Vehicle Systems". In: *Proc. 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV)*.

Chae, Heesung et al. (2005). "StarLITE : a new artificial landmark for the navigation of mobile robots". In: *Proceedings of the 1st Japan-Korea Joint Symposium on Network Robot Systems*.

Cicolani, Luigi S. and Gerd Kanning (1992). "Equations of motion of slung-load systems, including multilift systems". In: *NASA Technical Memorandum 102246*.

Cotugno, Giuseppe et al. (2013). "Extended and Unscented Kalman Filters for mobile robot localization and environment reconstruction". In: *21st Mediterranean Conference on Control and Automation*, pp. 19–26. DOI: 10.1109/MED.2013.6608693. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6608693.

Dil, B. J. and P. J. M. Havinga (2011). "Stochastic Radio Interferometric Positioning in the 2.4 GHz Range, journal = Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems". In: pp. 108–120.

Egeland, O. and J.T. Gravdahl (2002). *Modeling and Simulation for Automatic Control*. Marine Cybernetics. ISBN: 9788292356012.

Engedy, István and Gábor Horváth (2009). "A Global , Camera-based Mobile Robot Localization". In: *International Symposium of Hungarian Researchers on Computational Intelligence and Informatics* 10, pp. 217–228.

Fossen, Thor I. (2011). *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, Ltd. ISBN: 9781119991496.

Gelb, Arthur et al. (2001). *Applied Optimal Estimation*. The M.I.T. Press. ISBN: 0262200279.

Grubbs, F. E. (1950). "Sample criteria for testing outlying observations". In: *The Annals of Mathematical Statistics* 21, pp. 27–58.

Gustafsson, Fredrik (2010). *Statistical Sensor Fusion*. Lantmateriverket, Gavle, Sweden, pp. 153–210. ISBN: 9789144077321.

Gustafsson, Fredrik and Gustaf Hendeby (2012). "Some Relations Between Extended and Unscented Kalman Filters". In: *Signal Processing, IEEE Transactions* 60, pp. 545–555.

György, Katalin, András Kelemen, and László Dávid (2014). "Unscented Kalman Filters and Particle Filter Methods for Nonlinear State Estimation". In: *Procedia Technology* 12, pp. 65–74. ISSN: 22120173. URL: http://linkinghub.elsevier.com/retrieve/pii/S2212017313006427.

Hoh, Roger H., Robert K. Heffley, and David G. Mitchell (2006). "Development of Handling Qualities Criteria for Rotorcraft with Externally Slung Loads". In: *American Helicopter Society Annual Forum* October.

Hol, Jeroen (2011). *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. URL: http://photon.isy.liu.se:81/publications/doc?id=2434.

Ivler, Christina M., Mark B. Tischler, and J. David Powell (2011). "Cable Angle Feedback Control Systems to Improve Handling Qualities for Helicopters with Slung Loads". In: *AIAA Guidance, Navigation and Control Conference*, pp. 1–27.

Kalman, Rudolf E. (1960). "A New Approach to Linear Filtering and Prediction Problems 1". In: *Transactions of the ASME – Journal of Basic Engineering* 82 (Series D), pp. 35–45.

Klausen, Kristian (2013). "Cooperative Behavioural Control for Omni-Wheeled Robots". MA thesis. Norwegian University of Science and Technology.

Klausen, Kristian, Thor I Fossen, and Tor Arne Johansen (2014). "Suspended Load Motion Control using Multicopters". In: *Mediterranean Conference of Control and Automation (MED)* 22.

— (2015). "Nonlinear Control of a Multirotor UAV with Suspended Load". In:

Lee, Kung-Chung et al. (2010). "A comparison between Unscented Kalman Filtering and particle filtering for RSSI-based tracking". In: *2010 7th Workshop on Positioning, Naviga-*

tion and Communication, pp. 157–163. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5650817.

Li, Xia et al. (2011). "Research of Localization and Tracking Algorithms Based on Wireless Sensor Network". In: *Journal of Information & Computational Science* 4.90304002, pp. 708–715.

LSTS (2015a). *DUNE: Unified Navigation Environment*. URL: http://lsts.fe.up.pt/toolchain/dune (visited on 06/07/2015).

— (2015b). *GLUED/Linux Uniform Environment Distribution*. URL: http://lsts.fe.up.pt/toolchain/glued (visited on 06/07/2015).

— (2015c). *LSTS*. URL: http://lsts.fe.up.pt/ (visited on 06/01/2015).

— (2015d). *Neptus Command and Control Software*. URL: http://lsts.fe.up.pt/toolchain/neptus (visited on 06/07/2015).

Mahony, Robert, Tarek Hamel, and Jean Michel Pflimlin (2008). "Nonlinear complementary filters on the special orthogonal group". In: *IEEE Transactions on Automatic Control* 53.5, pp. 1203–1218.

Martinez, Ramiro, Adrian Jimenez-Gonzalez, and Anibal Ollero (2011). *Sensor Fusion - Foundation and Applications*. CC BY-NC-SA 3.0 license., pp. 41–60. ISBN: 978-953-307-446-7.

Masoud, Ziyad N., Mohammed F. Daqaq, and Nader. A. Nayfeh (2004). "Pendulation Reduction on Small Ship-Mounted Telescopic Cranes". In: *Journal of Vibration and Control* 10.8, pp. 1167–1179.

Masoud, Ziyad N. and Ali H. Nayfeh (2003). "Sway Reduction on Container Cranes Using Delayed Feedback Controller". In: *Nonlinear Dynamics* 34.3-4, pp. 347–358.

NaturalPoint (2015). *NaturalPoint Inc.* URL: https://www.naturalpoint.com/ (visited on 05/01/2015).

Nocedal, Jorge and Stephen J. Wright (2006). *Numerical Optimization*, pp. 304–351. ISBN: 9780387303031.

Omar, Hanafy Mohammad (2012). "Anti-swing controller based on time-delayed feedback for helicopter slung load system near hover". In: *Journal of Aerospace Technology and Management* 4.3, pp. 297–305.

OptiTrack (2015a). *Motion Capture Systems*. URL: http://www.optitrack.com/products/flex-13/ (visited on 05/01/2015).

OptiTrack, Motive:Tracker (2015b). *Motion Capture & DoF Object Traking*. URL: http://www.optitrack.com/products/motive/tracker/ (visited on 05/01/2015).

Palunko, Ivana, Rafael Fierro, and Patricio Cruz (2012). "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach". In: *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2691–2697.

Pelka, Mathias, Christian Bollmeyer, and Horst Hellbr (2014). "Accurate Radio Distance Estimation by Phase Measurements with Multiple Frequencies". In: *The Fifth International Conference on Indoor Positioning and IndoorvNavigation 2014 (IPIN 2014)* October.

Phillips, Richard E. and George T. Schmidt (1996). "System Implications and Innovative Applications of Satellite Navigation". In: *Advisory Group for Aerospace Research and development: AGARD-LS-207*, pp. 9.1–9.18.

Potter, James, William Singhose, and Mark Costello (2011). "Reducing Swing of Model Helicopter Sling Load Using Input Shaping". In: *International Conference on Control and Automation (ICCA)* 9.

Proakis, John G. and Dimitris G. Manolakis (1996). *Digital Signal Processing (3rd Ed.): Principles, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. ISBN: 0133737624.

Røli, Jon-Håkon Bøe (2015). "Cooperative Control and RTK Navigation System for Multicopters". MA thesis. Norwegian University of Science and Technology.

Sasiadek, J.Z. and Q. Wang (1999). "Sensor fusion based on fuzzy Kalman filtering for autonomous robot vehicle". In: *Proceedings 1999 IEEE International Conference on Robotics and Automation* May, pp. 2970–2975.

Schlee, F. H., C. J. Standish, and N. F. Toda (1967). "Divergence in the Kalman filter." In: *AIAA Journal* 5.june, pp. 1114–1120.

Serway, Raymond and John Jewett (2013). *Physics for scientists and engineers*. Cengage Learning.

Shareef, Ali and Yifeng Zhu (2009). "Localization Using Extended Kalman Filters in Wireless Sensor Networks". In: *DigitalCommons@UMaine* April.

Singer, Neil C. (1989). "Residual Vibration Reduction in Computer Controlled Machines". In:

Singh, Tarunraj and William Singhose (2002). "Tutorial on Input Shaping / Time Delay Control of Maneuvering Flexible Structures". In: *American Control Conference*.

Spockeli, Bjørn (2015). "Integration of RTK GPS and IMU for Accurate UAV Positioning". MA thesis. Norwegian University of Science and Technology.

St-pierre, Mathieu and Denis Gingras (2004). "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system". In: *2004 IEEE Intelligent Vehicles Symposium*, pp. 831–835.

Steen, Thor Audun (2014). "Search and Rescue Mission Using Multicopters". MA thesis. Norwegian University of Science and Technology.

Stergiopoulos, John and Anthony Tzes (2007). "An adaptive input shaping technique for the suppression of payload swing in three-dimensional overhead cranes with hoisting mechanism". In: *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, pp. 565–568.

Svendsen, Jon Glenn Gjevestad (2001). *Real-Time Phase Ambiguity Resolution in Global Navigation Satellite Systems*.

SwiftNAV (2015a). *libswiftnav documentation*. URL: http://docs.swift-nav.com/libswiftnav/ (visited on 06/01/2015).

— (2015b). *Piksi Datasheet v2.3.1*. URL: http://docs.swift-nav.com/pdfs/piksi_datasheet_v2.3.1.pdf (visited on 06/07/2015).

— (2015c). *The Swift Nav Binary Protocol*. URL: http://docs.swiftnav.com/wiki/SwiftNav_Binary_Protocol (visited on 06/01/2015).

Takasu, Tomoji and Akio Yasuda (2004). "Cycle Slip Detection and Fixing by MEMS - IMU / GPS Integration for Mobile Environment RTK-GPS". In: *Tokyo University of Marine Science and Technology, Japan*.

Ubiquiti-Networks (2015). *PICOM2HP Datasheet*. URL: http://www.ubnt.com/downloads/datasheets/picostationm/picom2hp_DS.pdf (visited on 06/07/2015).

Verhaegen, Michel and Vincent Verdult (2007). *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press. ISBN: 9781107405028.

Vik, Bjørnar and Thor I. Fossen (2000). "A Nonlinear Observer for Integration of GPS and Inertial Navigation Systems". In: *Modeling, Identification and Control* 21.4, pp. 192–208. DOI: 10.4173/mic.2000.4.1.

Walpole, Ronald E. et al. (2011). *Probability and statistics for engineers and scientists*. Vol. 3. Pearson. ISBN: 0132047675.

Zadeh, L.A. (1965). "Fuzzy sets". In: *Information and Control* 3.8, pp. 338–353. ISSN: 1558-2531.

— (1973). "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes". In: *Systems, Man and Cybernetics, IEEE Transactions on* SMC-3.1, pp. 28–44.

— (1975a). *Fuzzy sets and their applications to cognitive and decision processes*. Academic Press Inc. ISBN: 9780127752600.

— (1975b). "The concept of a linguistic variable and its application to approximate reasoning—II". In: *Information Sciences* 8.4, pp. 301–357.

— (1994). "Fuzzy logic, neural networks, and soft computing". In: *Fuzzy systems* 37.3, pp. 77–84.

Zhou, Shuqiao, Haoran Feng, and Ruixi Yuan (2009). "Error Compensation for Cricket Indoor Location System". In: *2009 International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 390–395. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5372772.