NTNU – Trondheim
Norwegian University of
Science and Technology

Integration of RTK GPS and IMU for Accurate UAV Positioning

Integrasjon av RTK GPS og IMU for nøyaktig
UAV-posisjonering

Bjørn Amstrup Spockeli

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:	Bjørn Spockeli
Department:	Engineering Cybernetics
Thesis title (Norwegian):	Integrasjon av RTK GPS og IMU for nøyaktig UAV-posisjonering
Thesis title (English):	Integration of RTK GPS and IMU for Accurate UAV Positioning

Thesis Description: The purpose of the thesis is to improve the accuracy and reliability of RTK GPS by implementing an observer for velocity, position and attitude of the X8 fixed-wing UAV using RTK GPS and IMU measurements.

The following items should be considered:

1. Define the scope of the thesis and clarify what your contributions are.
2. Develop a navigational payload for the X8 fixed-wing UAV
3. Implement an observer for velocity, position and attitude based on RTK GPS and IMU data.
4. Investigate if the RTK GPS position data must be compensated for time delays and modify the observer to compensate for time delays if necessary.
5. Test the observer in Hardware-in-the-Loop(HIL) simulations
6. Experimental testing of the observer using the payload mounted in the X8 fixed-wing UAV
7. Conclude your results.

Start date:	2015-01-05
Due date:	2015-06-18

Thesis performed at:	Department of Engineering Cybernetics, NTNU
Supervisor:	Professor Thor I. Fossen, Dept. of Eng. Cybernetics, NTNU
Co-Supervisor:	Dr Nadia Sokolova, Dept. of Eng. Cybernetics, NTNU
	Jakob Mahler Hansen, Dept. of Eng. Cybernetics, NTNU

Abstract

Real-Time Kinematic GPS (RTK GPS) can provide centimeter accuracy and may be used for high accuracy positioning of Unmanned Aerial Vehicles (UAVs). In order to compute its position a Global Navigation Satellite System (GNSS) receiver processes measurements of signals from GNSS satellites. The processing results in a delay before the position measurement is available to the UAV. If the UAV is highly dynamic the delayed measurement will describe a past position, resulting in reduced accuracy.

This thesis presents the hardware and software architecture of an integrated Real-Time Kinematic GPS/Inertial Navigation System (RTK GPS/INS) for UAV applications that compensates for the time-delay. The RTK GPS system is constructed using single-frequency GPS receivers and the open-source Real-Time Kinematic Library (RTKLIB). The RTK GPS/INS navigation system is implemented in the DUNE(DUNE:Unified Navigation Environment)-framework running on the embedded payload computer of the UAV. The integration of RTK GPS and ineartial measurements is performed by a nonlinear observer estimating the delayed states and recovering the current states using inertial measurements. Hardware-in-the-Loop simulations and experimental flights with the x8 Skywalker UAV are used to test the navigation system.

Keywords: Unmanned Aerial Vehicle, Real-Time Kinematic GPS, Inertial Navigation, Nonlinear Observer

Sammendrag

(*Norwegian Translation of the Abstract*)

Real-Time Kinematic GPS kan gi posisjonsmålinger med centimeternøyaktighet og kan brukes til svært nøyaktig posisjonering av ubemannede luftfartøy, også referert til som droner. For at en GPS-mottaker skal kunne beregne sin egen posisjon må den prosesserer satellitmålinger som fører til en forsinkelse fra måletidspunktet til posisjonsmålingen er tilgjengelig for dronen. Dersom dronen beveger seg i høy hastighet, vil dette føre til at posisjonsmålingen er utdatert når den ankommer dronen og fører til redusert nøyaktighet.

Denne masteroppgaven presenterer hardware- og softwarearkitekturen til et integrert Real-Kinematic GPS/Treghetsnavigasjonssystem for ubemannede luftfartøy som kompenserer for denne tidsforsinkelsen. RTK GPS-systemet er konstruert ved å bruke to GPS-mottakere og Real-Time Kinematic Library (RTKLIB), et åpent kildekode bibliotek for sanntidsposisjonering. Navigasjonssystemet er implementert i DUNE: Unified Navigational Environment (DUNE)-rammeverket som kjøres på en datamaskinen i nyttelasten til dronen. Integrasjonen av posisjonsmålinger og treghetsmålinger utføres av en ulineær estimator som estimerer de forsinkede tilstandene til dronen og predikerer de nåværende tilstandene ved hjelp treghetsmålinger. Hardware-in-the-Loop simuleringer og eksperimentelle flyvninger er utført for å teste navigasjonssystemet.

Nøkkelord: Ubemannet luftfartøy, Real-Time Kinematic GPS, Treghetsnavigasjon, Ulineær estimator

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree MSc. at the Norwegian University of Science and Technology. The initiative of this thesis came from the Autonomous UAV project at the Center for Autonomous Marine Operations and Systems (AMOS).

I would like to thank my supervisor Professor Thor I. Fossen for his guidance and feedback. I would also like to thank my co-supervisors; Nadia Sokolova for valuable feedback on RTK matters and Jakob Mahler Hansen for excellent comments and productive discussions.

A huge thanks goes to João Fortuna and Kristian Klausen for their invaluable help at the Unmanned Aerial Vehicles Laboratory (UAV-Lab). Furthermore I want to thank the UAV-pilot Lars Semb for excellent flying in during experimental testing and the technical staff at the electronics and mechanical workshops at the Department of Engineering Cybernetics for all their help.

Lastly I want to thank my family and girlfriend for their never-ending support.

Trondheim, June 18, 2015

Bjørn Amstrup Spockeli

Contents

List of Figures

List of Tables

List of Abbreviations

C/A Coarse/Acquisition.

DUNE DUNE: Unified Navigational Environment.

ECEF Earth-Centered Earth Fixed.

ECI Earth-Centered Inertial.

EKF Extended Kalman Filter.

GLUED GNU/Linux Uniform Environment Distribution.

GNSS Global Navigation Satellite System.

GPS Global Positioning System.

HIL Hardware-in-the-Loop.

IMC Inter-Module Communication.

IMU Inertial Measurement Unit.

INS Inertial Navigation System.

IRL Integer Least-Squares.

LAMBDA Least-Squares Ambiguity Decorrelation Adjustment.

MLAMBDA Modified Least-Squares Ambiguity Decorrelation Adjustment.

NED North-East-Down.

NTNU Norwegian University of Science and Technology.

PRN Pseudorandom Noise.

RTK GPS Real-Time Kinematic GPS.

RTK GPS/INS Real-Time Kinematic GPS/Inertial Navigation System.

RTKLIB Real-Time Kinematic Library.

SIL Software-in-the-Loop.

TCP Transmission Control Protocol.

TOA Time of Arrival.

TOV Time of Validity.

UAV Unmanned Aerial Vehicle.

UAV-Lab Unmanned Aerial Vehicles Laboratory.

Chapter 1

Introduction

In this chapter the background and motivation for this thesis will be presented, followed by a brief introduction into previous work performed on the topic. The main objectives and contributions of the thesis will be introduced before finally presenting the organization of chapters and appendices of the thesis.

1.1 Background and Motivation

Single-frequency Real-Time Kinematic GPS can deliver centimeter accuracy by utilizing carrier-phase measurements in addition to code-phase measurements. Real-Time Kinematic(RTK) positioning relies on determining the number of wavelengths between the satellite and receiver, also referred to as integer ambiguity resolution. Unfortunately, the ambiguity resolution process is not very robust and signal noise, multipath and high receiver dynamics may cause the process to fail.

Furthermore, all GNSS receivers suffer from computational delays as the position cannot be computed instantaneously. This delay may be in the magnitude of several hundred milliseconds and may rightfully be neglected in low velocity applications, like ships. However, in high-dynamic applications like Unmanned Aerial Vehicles(UAVs) the delayed measurement will degrade the navigation solution since the measurement describes the past position of the UAV. This results in the centimeter accuracy being lost due to the errors introduced by the time-delay.

Integrating RTK GPS with an Inertial Navigation System(INS) will increase reliability and fault-tolerance. Further advantages are that the high-accuracy position measurements are available at a higher sampling rate and if the carrier-phase ambiguity resolution algorithm fails, the navigation system is able to coast through the outage. However, integrated GPS/INS navigations systems are often realized through computationally intensive Extended Kalman Filters(EKFs). Real-Time

Kinematic positioning involves the use of an ambiguity resolution algorithm which also is computational intensive. Implementing both on a small embedded computer will impose a significant computational load that can limit the usability of the system.

This thesis focus on the development of an integrated RTK GPS/INS navigation system for UAV applications that compensates for the computational time-delay. In order to reduce the computational load on the embedded computer the integration will be implemented using a nonlinear observer as opposed to an Extended Kalman Filter.

1.2 Previous Work

Carrier phase GPS positioning has been demonstrated to achieve centimeter accuracy (Hofmann-Wellenhof et al., 2008) and the ambiguity resolution problem has been studied extensively (Blewitt, 1989; Hatch, 1990; Euler and Landau, 1992; Teunissen, 1994). The application of RTK GPS for precision farming, machine control and aerial imagery is discussed in (Stempfhuber and Buchholz, 2011) and (Rieke et al., 2011). The development and testing of a system for autonomous precision recovery of fixed-wing UAVs using low-cost single-frequency RTK GPS is described in (Skulstad et al., 2015)

The topic of GNSS/INS integration has been studied for decades and is typically implemented using an EKF (Maybeck, 1982; Phillips and Schmidt, 2008; Farrell, 2008). The integration of RTK GPS/INS using EKFs has been explored by (Farrel et al., 2000; Yunchun and Farrell, 2003; Scherzinger, 2000) for automotive applications. The effects of time synchronization errors on an EKF-based loosely-coupled GNSS/INS have been studied by (Skog and Händel, 2011) and (Solomon et al., 2011) proposes a method for on-line estimating of GNSS latency in a real-time loosely-coupled GNSS/INS EKF-based navigation systems.

Nonlinear navigation observers for loosely integrated GPS/INS have been subject of increased research during the recent years as they have been shown to have significantly stronger stability properties than EKFs (Vik and Fossen, 2001; Hua, 2010; Grip et al., 2015). (Vik and Fossen, 2001) expanded the attitude observer of (Salcudean, 1991) to include gyro bias estimation used it in a nonlinear GNSS/INS integration observer with the assumption that an external attitude measurement is available. (Hua, 2010) used the vector-based attitude observer of (T.Hamel and Mahoney, 2006) and (Mahoney et al., 2008) to construct two algorithms for attitude and velocity estimation based on GNSS velocity, magnetometer and inertial measurements.

Based on the attitude observer of (Batista et al., 2011a,b), (Grip et al., 2012) designed an observer for estimating position, velocity, attitude and gyro bias using inertial, magnetometer and satellite-based position and velocity measurements. The authors improve the design in (Grip et al., 2011b) by representing the attitude in unit quaternions based on the attitude observer of (Mahoney et al., 2008).

In the field of time-delayed nonlinear observers (Khosravian et al., 2014, 2015) developed a nonlinear attitude estimation method consisting of a delayed observer coupled with a dynamic predictor. In a similar manner (Hansen et al., 2015) proposes a nonlinear observer structure for estimating position, velocity and attitude in a GNSS/INS system with delayed GNSS-measurements. The inertial and magnetometer measurements are delayed in time to coincide with the GNSS-measurements and a fast simulator is employed to recover the current states.

1.3 Problem Statement

The main objective of this thesis is the development of an integrated RTK GPS/INS navigation system for UAV applications that compensates for the time-delay introduced by receiver processing and achieve sub-meter accuracy. The following points will be considered in this thesis

- design and construction of a navigational payload for the x8 Skywalker
- determining the magnitude of the time-delay using the EVK-6T GPS receiver and the Real-Time Kinematic Library(RTKLIB).
- implementing the nonlinear observer structure from (Hansen et al., 2015) in DUNE to compensate for the time-delay
- testing the implemented observer in HIL-simulations with simulated time-delayed GPS measurements
- experimental testing of the implemented observer using the developed payload

1.4 Contributions

The main contributions of this thesis is

- Hardware and software architecture of an integrated RTK GPS/INS navigation system, see Ch. 6.
- Determination of RTK GPS time delay using EVK-6T and RTKLIB, see Sec. 8.1.
- Integration of RTKLIB in DUNE, see Sec. 6.1.1.
- On-line implementation of the time-delayed nonlinear observer from (Hansen et al., 2015) in C++, see Sec. 6.1.3.
- Hardware-in-the-Loop (HIL)-testing of the implemented observer, see Sec. 7.2
- Experimental testing of the implemented observer, see Sec. 8.3

1.5 Organization of the Thesis

This thesis is organized in ten chapters and four appendices. The chapters are indexed from 1 - 10 and the appendices are indexed from A-D. A short description of the chapters and appendices is given below.

Chapters

- Chapter Two presents the coordinate frames and the rotation matrices used to transform coordinates between the frames.
- Chapter Three contains an introduction to Real-Time Kinematic GPS.
- Chapter Four describes the system and observer equations used in this thesis.
- Chapter Five contains a short description of the software and hardware components of the navigational payload.
- Chapter Six describes the implementation of the observer equations in software and construction of the payload.
- Chapter Seven contains a description of the simulator architecture used for Software-in-the-Loop (SIL) and Hardware-in-the-Loop (HIL) simulations and presents the results obtained from the HIL simulations.
- Chapter Eight describes the experiments that have been conducted and presents the results.
- Chapter Nine discusses the results presented in Chapter Seven and Eight.
- Chapter Ten draws conclusions based on the results presented in Chapter Seven and Eight and the discussion from Chapter Nine before giving recommendations for future work.

Appendices

- Appendix A describes the configuration of the EVK-6T GPS receivers.
- Appendix B describes the configuration of the Real-Time Kinematic Library.
- Appendix C contains the build rules necessary to compile the Real-Time Kinematic Library for Glued.
- Appendix D contains the schematics, board layouts and bill of materials for the custom circuits used in this thesis.

Source Code

- The source code developed in this thesis can be found on the UAV-Lab git server http://uavlab.itk.ntnu.no:88/ under the branch *feature/rtkimu*

Chapter 2

Mathematical Background

This chapter presents the definition of the reference frames used for satellite and inertial navigation as well as the transformation matrices used to transform coordinates between the different frames. This chapter is based on (Fossen, 2011).

2.1 Reference Frames

This section describes the four reference frames used for satellite and inertial navigation.

2.1.1 ECI

The Earth-Centered Inertial (ECI) frame, denoted with subscript i, is defined as an inertial frame for terrestrial navigation, i.e a non-accelerating reference frame in which Newtons Laws of motion apply. The origin of the ECI-frame is the center of the earth with the x-axis pointing towards the vernal equinox, the z-axis along the rotational axis of the earth and the y-axis completing the right handed orthogonal coordinate system. The coordinate system is illustrated in Fig. 2.1.

2.1.2 ECEF

The Earth-Centered Earth Fixed (ECEF) reference frame, denoted by the e subscript, is defined with origin in the center of the earth. The x-axis points to the intersection of the prime meridian(Greenwich Meridian) and Equator, the z-axis points along the rotational axis of the Earth and the y-axis completes the right-handed orthogonal coordinate system.

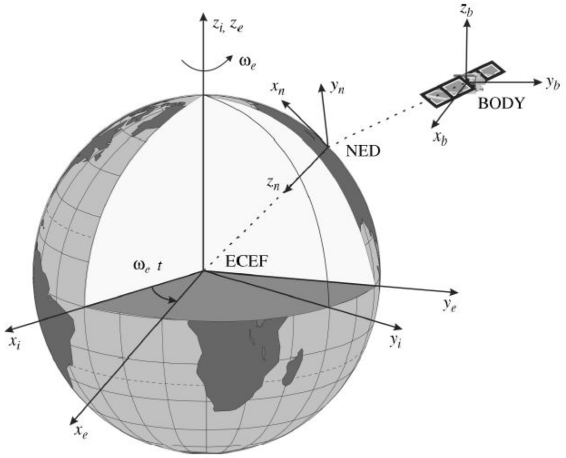


Figure 2.1: The ECEF-frame rotates with respect to the inertial ECI-frame with the angular velocity ω_e. The NED-frame is defined with the z-axis normal to WGS-84 and its position is defined relative to the ECEF-frame. The position and orientation of the Body-frame is described relative to the NED-frame. Figure from (Fossen, 2011)

The ECEF-frame rotates relative to the ECI-frame with an angular velocity $\omega_e = 7.2921 \times 10^{-5}\ rad/s$, illustrated in Fig. 2.1. Coordinates in the ECEF-frame may be given either in Cartesian or ellipsoidal coordinates, i.e latitude, longitude and height. Global Navigational Satellite Systems like GPS calculates position in cartesian coordinates, but most receivers output latitude, longitude and height. The vectorial representation of both coordinate forms can be seen below

$$\boldsymbol{p}^e_{b/e} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 \qquad\qquad \boldsymbol{\Theta}_{en} = \begin{bmatrix} l \\ \mu \end{bmatrix} \in \mathbb{S}^2$$

2.1.3 BODY

The BODY-fixed reference frame , denoted by the b subscript, is a moving reference frame with origin fixed to the vehicle, either the center of gravity or the . The axes of the body frame are usually chosen to coincide with the principle axes of inertia, i.e x-axis is the longitudinal axis(pointing in the forward direction), y-axis is the transversal axis(pointing to the right) and the z-axis is the normal axis(pointing upwards). The position and orientation of the vehicle is given relative to either the NED or the ECEF frame. The linear and angular velocities of the vehicle are given in the body frame, thus inertial measurements are given relative to the ECI-frame. The coordinate system is illustrated in Fig. 2.1.

2.1.4 NED

The North-East-Down (NED) reference frame is defined relative to the Earth reference ellipsoid (WGS84). The x-axis points toward true North , the y-axis points towards East and the z-axis points downwards normal to the ellipsoid.The coordinate system is illustrated in Fig. 2.1. The position vector and the Euler angle vector are shown below

$$\boldsymbol{p}^n_{b/n} = \begin{bmatrix} N \\ E \\ D \end{bmatrix} \in \mathbb{R}^3 \qquad \boldsymbol{\Theta}_{nb} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \in \mathbb{S}^3$$

2.2 Coordinate Transformations

This section describes the unit quaternions representation and the transformation matrices between some of the reference frames from Sec. 2.1.

2.2.1 Unit Quaternions

The singularities of the Euler angle representation may be avoided by using unit quaternions. A unit quaternion is defined as a real complex number with one real part η and three imaginary parts given by the vector $\boldsymbol{\varepsilon} = [\varepsilon_1\ \varepsilon_2\ \varepsilon_3]^\top$. Unit quaternions are usually expressed in the following form

$$\boldsymbol{q} = [\eta\ \varepsilon_1\ \varepsilon_2\ \varepsilon_3]^\top \tag{2.1}$$

Using quaternions instead of Euler angles, the linear velocity vector of the body-frame with respect to the NED-frame can be expressed as

$$\dot{\boldsymbol{p}}^n_{b/n} = \boldsymbol{R}^e_n(\boldsymbol{q})\boldsymbol{v}^b_{b/n} \tag{2.2}$$

where $\boldsymbol{R}^n_b(q)$ is the rotation matrix from the NED-frame to the body-frame given by

$$\boldsymbol{R}^n_b(q) = \begin{bmatrix} 1-2(\varepsilon_2^2+\varepsilon_3^2) & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\eta) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\eta) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\eta) & 1-2(\varepsilon_1^2+\varepsilon_3^2) & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\eta) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\eta) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\eta) & 1-2(\varepsilon_1^2+\varepsilon_2^2) \end{bmatrix} \tag{2.3}$$

As for the Euler angle representation, $\boldsymbol{R}_b^n(q) \in SO(3)$ which implies that $\boldsymbol{R}_b^n(q)^{-1} = \boldsymbol{R}_b^n(q)^\top$. The angular velocity transformation is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{T}_q(\boldsymbol{q})\boldsymbol{\omega}_{b/n}^b \tag{2.4}$$

where the transformation matrix $\boldsymbol{T}_{\boldsymbol{q}}(\boldsymbol{q})$ is given by

$$\boldsymbol{T}_{\boldsymbol{q}}(\boldsymbol{q}) = \frac{1}{2}\begin{bmatrix} -\varepsilon_1 & -\varepsilon_2 & -\varepsilon_3 \\ \eta & -\varepsilon_3 & -\varepsilon_2 \\ \varepsilon_3 & \eta & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & \eta \end{bmatrix} \tag{2.5}$$

2.2.2 Transformations between BODY and NED

The linear velocity of the origin of the body-frame with respect to the NED-frame can be expressed as

$$\boldsymbol{v}_{b/n}^n = \boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})\boldsymbol{v}_{b/n}^b \tag{2.6}$$

where $\boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})$ is the rotation matrix from the Body-frame to the NED-frame with $\boldsymbol{\Theta}_{nb}$, the Euler angles between NED and BODY as an argument. $\boldsymbol{v}_{b/n}^b$ is the body-fixed linear velocity expressed in the body-frame.

The rotation matrix is defined as the product of three principal rotations, i.e single axis rotations.The principal rotation matrices, $\boldsymbol{R}_{z,\psi}, \boldsymbol{R}_{y,\theta}, \boldsymbol{R}_{x,\phi}$ given by (2.7) describe the rotations about the z, y and x axes respectively.

$$\boldsymbol{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \quad \boldsymbol{R}_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad \boldsymbol{R}_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

The order in which the principal rotations are performed is not arbitrary, but follow the zyx convention from NED to body, with the Euler Angles ϕ, θ and ψ. Thus $\boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})$ is defined as

$$\boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb}) = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi} \tag{2.8}$$

Expanding (2.8) yields

$$\boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2.9}$$

The relationship between the Euler rate vector $\dot{\mathbf{\Theta}}_{nb}$ and the body-fixed angular velocity vector $\boldsymbol{\omega}^b_{b/n}$ is given by

$$\dot{\mathbf{\Theta}}_{nb} = \boldsymbol{T}_\Theta(\mathbf{\Theta}_{nb})\boldsymbol{\omega}^b_{b/n} \tag{2.10}$$

where $\boldsymbol{T}_\Theta(\mathbf{\Theta}_{nb})$ is a transformation matrix given by

$$\boldsymbol{T}_\Theta(\mathbf{\Theta}_{nb}) = \begin{bmatrix} 1 & sin(\phi)tan(\theta) & cos(\phi)\tan(\theta) \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi)/cos(\theta) & cos(\phi)/cos(\theta) \end{bmatrix} \tag{2.11}$$

It should be noted that is singular for pitch angles $\theta = \pm 90^\circ$ and that $\boldsymbol{T}_\Theta(\mathbf{\Theta}_{nb}) \notin SO(3)$, thus $\boldsymbol{T}^\top_\Theta(\mathbf{\Theta}_{nb}) \neq \boldsymbol{T}^{-1}_\Theta(\mathbf{\Theta}_{nb})$.

2.2.3 Transformations between ECEF and NED

The linear velocity of the body-frame given in the inertial frame may be expressed as

$$\dot{\boldsymbol{p}}^e = \boldsymbol{R}^e_n(\mathbf{\Theta}_{en})\dot{\boldsymbol{p}}^n \tag{2.12}$$

where $\mathbf{\Theta}_{en} = [l, \mu]^\top$ and $\boldsymbol{R}^e_n(\mathbf{\Theta}_{en})$ is the rotation matrix between the ECEF-frame and the NED-frame.

Similarly to the rotation matrix between the Body-frame and the NED-frame, $\boldsymbol{R}^e_n(\mathbf{\Theta}_{en})$ may be decomposed into principal rotations. Using the zyx-convention, the rotation from ECEF to NED achieved by performing two principal rotations: (1) rotate l about the z-axis and (2) rotate $-\mu - \frac{\pi}{2}$ about the y-axis. The procedure is illustrated in Figure 2.2. Thus the rotation matrix between the ECEF-frame and the NED-frame is given by

$$\begin{aligned} \boldsymbol{R}^e_n(\mathbf{\Theta}_{en}) &= \boldsymbol{R}_{z,l}\boldsymbol{R}_{y,-\mu-\frac{\pi}{2}} & (2.13) \\ &= \begin{bmatrix} cos(l) & -sin(l) & 0 \\ sin(l) & cos(l) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(-\mu - \frac{\pi}{2}) & 0 & sin(-\mu - \frac{\pi}{2}) \\ 0 & 1 & 0 \\ -sin(-\mu - \frac{\pi}{2}) & 0 & cos(-\mu - \frac{\pi}{2}) \end{bmatrix} & (2.14) \end{aligned}$$

Multiplying the two principal rotation matrices and the use of the trigonometric identities $cos(-\mu - \frac{\pi}{2}) = -sin(\mu)$ and $sin(-\mu - \frac{\pi}{2}) = -cos(\mu)$ yields

$$\boldsymbol{R}^e_n(\mathbf{\Theta}_{en}) = \begin{bmatrix} -cos(l)sin(\mu) & -sin(l) & -cos(l)cos(\mu) \\ -sin(l)sin(\mu) & cos(l) & -sin(l)cos(\mu) \\ cos(\mu) & 0 & -sin(\mu) \end{bmatrix} \tag{2.15}$$

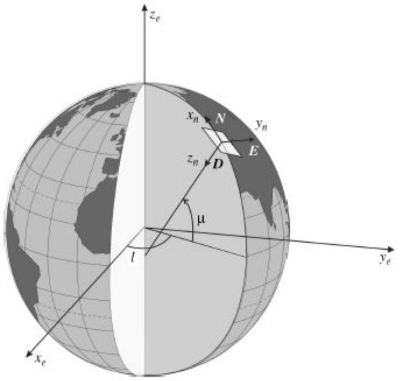


Figure 2.2: The relationship between the ECEF-frame and the NED reference frame using ellipsoidal coordinates. Figure from (Fossen, 2011)

2.2.4 Unit Quaternions to Euler Angles

The relationship between the Euler angles and unit quaternions is found by setting

$$\boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb}) = \boldsymbol{R}_b^n(q) \tag{2.16}$$

Expanding the matrices and denote the elements of $\boldsymbol{R}_b^n(q)$ by R_{ij}, $i, j \in \{1, .., 3\}$

$$\begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{2.17}$$

This results in a system with nine equations and three unknowns and solving for the Euler angles yields

$$\phi = atan2(R_{32}, R_{33}) \tag{2.18}$$

$$\theta = -sin^{-1}(R_{31}) = -tan^{-1}\left(\frac{R_{31}}{\sqrt{1 - R_{31}^2}}\right), \quad \theta \neq \pm 90^\circ \tag{2.19}$$

$$\psi = atan2(R_{21}, R_{11}) \tag{2.20}$$

where atan2(x,y) is the four-quadrant arctangent of the real parts of the elements of x and y satisfying $-\pi \leq atan2(y, x) \leq \pi$. Precautions should be taken against computational errors in the vicinity of the singularities.

Chapter 3

Real-Time Kinematic GPS

This chapter contains a brief description of the principles behind Real-Time Kinematic GPS and is based on (Hofmann-Wellenhof et al., 2008). It is assumed that the reader is familiar with the Global Positioning System(GPS).

3.1 Code and Phase Pseudoranges

In principle, the observables used in satellite navigation are ranges which are deduced from measured time or phase differences based on a comparison between received signals and receiver-generated signals. As there are two clocks involved, one in the satellite and one in the receiver, the ranges contain clock errors and are therefore denoted pseudoranges. Pseudoranges can also be obtained from carrier phase measurements. The equation for code pseudorange is given as

$$R = c\Delta t + c\Delta\delta = \varrho + c\Delta\delta \tag{3.1}$$

where ϱ is the distance between receiver and satellite at transmission time epoch t and the receiver at reception time epoch , c the speed of light and $\Delta\delta$ combined receiver and satellite clock error.. The equation for phase pseudorange is given as

$$\lambda\Phi = \varrho + c\Delta\delta + \lambda N \tag{3.2}$$

where $\lambda\Phi$ is the carrier phase derived pseudorange expressed in meters,ϱ is the distance between receiver and satellite at transmission time epoch t and the receiver at reception time epoch , λ the wavelength of the carrier wave, c the speed of light, $\Delta\delta$ clock error and N is the integer ambiguity

3.2 Bias and Noise

The error sources of a GNSS-receiver may be classified into three groups; satellite-related errors, propagation-medium-related errors and receiver-related errors, shown in Table 3.1. The following subsections describe the individual errors listed in Tab. 3.1

Table 3.1: Range biases

Source	**Effect**
Satellite	Clock bias Orbital Errors
Signal Propagation	Ionospheric refraction Tropospheric refraction
Receiver	Antenna phase center variation Clock bias Multipath

3.2.1 Clock Errors

Both the satellite and receiver clocks drift and introduces an error when calculating the transmission time.

$$\begin{aligned} r &= c(T_u - T_s) = c\Delta t \\ \rho &= c\left[(T_u + t_u) - (T_s + t_s)\right] \\ &= r + c(t_u - t_s) \end{aligned} \tag{3.3}$$

where r denotes the actual range and ρ the pseudorange. T_s and T_u denotes the time of transmission from the satellite and the time of arrival at the receiver, respectively. Lastly t_s and t_u denotes the satellite and receiver clock offset, respectively.

The satellite clocks are given corrections every 24 hours, keeping the error within one nanosecond, equivalent to a 30 cm range error. The receiver clock is not given corrections, but as the receiver clock offset is common to all pseudorange measurements, it may be eliminated by adding an extra measurement and solving for t_u.

3.2.2 Ephemeris Errors

Ephemeris describes the position of the satellites as a function of time. Predictions of satellite positions are included in the GPS signal as a part of the data message modulated onto the carrier wave. Errors in these predictions are denoted ephemeris errors and contribute to the overall inaccuracy of the ranging measurement.

3.2.3 Tropospheric and Ionospheric Delays

As GPS signals propagate from the satellite to the receiver, they are delayed due to refraction in the atmosphere, specifically the ionosphere and the troposphere. Refraction is the change of direction of an electromagnetic wave due to changes in its transmission medium. The wave deviates from the straight path between the satellite and receiver, and thus travels a longer path influencing the ranging measurement.

Several models have been developed in order to predict and compensate for atmospheric refraction. The Klobuchar model approximates the entire vertical ionospheric refraction and is able to remove on average 50 percent of the delay. The tropospheric delay is predicted through the Hopfield Model.

3.2.4 Multipath

Reflective surfaces in the vicinity of the receiver may cause signals to take a longer indirect path to the receiver, thus influencing the perceived transmission time. Several approaches have been developed in order to mitigate multipath errors. Pre-reception migitation involves the use of ground planes, choke ring and directional antennas to prevent multipath signals from being picked up by the reciever. Other methods use signal and data processing to reject multipath measurements.

3.3 The Phase Ambiguity

To account for the most significant error terms, Eq. 3.2 can be expanded as follows:

$$\lambda\Phi = \varrho + c\Delta\delta + \lambda N - \Delta^{Iono} \tag{3.4}$$

In order to determine the range using carrier wave measurements, the number of carrier cycles between the satellite and receiver must be determined. The PRN sequences within the code observable were created so that they are easy to align. This is not the case for the carrier waves that are uniform sinusoidal signals, seen in Figure 3.1, making one cycle indistinguishable from the other. Thus, the initial number of complete wavelengths N is unknown and denoted the integer ambiguity. Determining this number is referred to as ambiguity resolution.

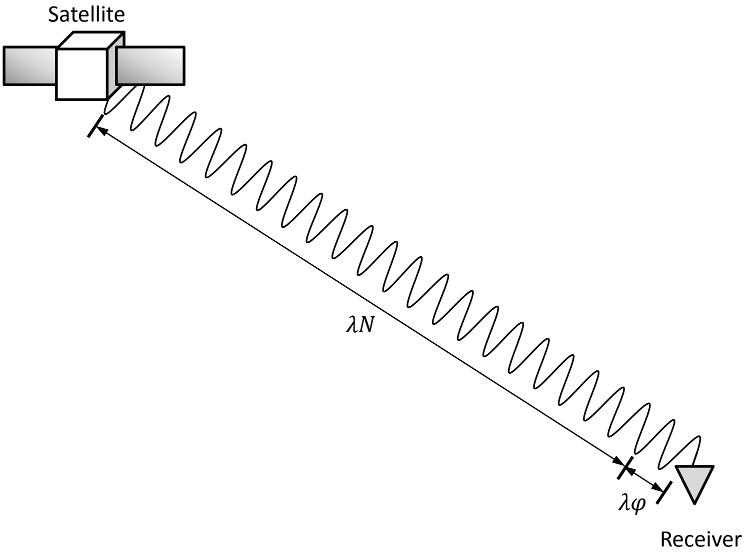


Figure 3.1: The ambiguity problem: The fractional phase ϕ , i.e. the phase shift from the satellite to the receiver can be measured, but not the number of complete cycles

3.4 Ambiguity Resolution

In order to fully exploit the high accuracy of the carrier phase observable, the ambiguities must be resolved or fixed to their correct integer value. Determining the ambiguities is referred to as ambiguity resolution.

In order to utilize the integer nature of the ambiguity, the systematic errors in the observation equation must be removed in order to isolate the ambiguity. The systematic errors, i.e clock errors, atmospheric refraction and orbital errors, are removed by differencing the phase measurements. Single differences are created by subtracting the measurement of receiver k from receiver j, see Fig. 3.2. Under the assumption that orbital and atmospheric errors are spatially correlated these error terms may be neglected. The observation equation of single differenced phase measurements is then given as

$$\lambda \Phi^j_{AB}(t) = \varrho^j_{AB}(t) + c\Delta\delta^j_{AB} + \lambda N^j_{AB} + \varepsilon^j_{AB} \tag{3.5}$$

where $\varrho^j_{AB}(t)$ is the difference in true range from satellite j to receiver A and B, $c\Delta\delta^j_{AB}$ is the receiver clock offset difference , N^j_{AB} is the ambiguity difference and ε^j_{AB} is a collective error term containing random noise and multipath errors.

Double differences are made by subtracting single differences from satellite j and k, resulting in the observation equation of double-differenced phase measurements is

$$\lambda\Phi^{jk}_{AB}(t) = \varrho^{jk}_{AB}(t) + \lambda N^{jk}_{AB} + \varepsilon^{jk}_{AB} \tag{3.6}$$

where the receiver clock errors are eliminated, leaving only the collective error term ε^{jk}_{AB}, containing random noise and multipath errors.

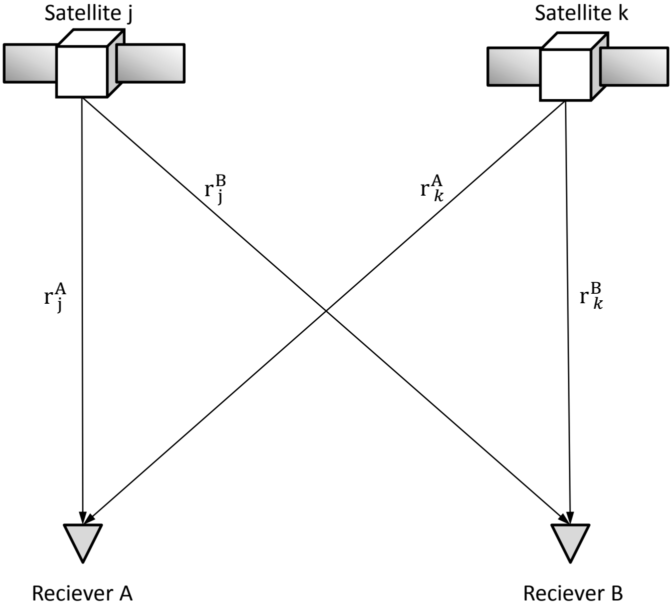


Figure 3.2: Differencing of GNSS measurements from two receivers where r^A_j,r^A_k,r^B_j and r^B_k are single measurements

Ambiguity resolution involves three major steps. First, the set of potential integer ambiguity combinations, denoted the search space, used by the resolution algorithm must be determined. In kinematic applications the search space is realized from the code range solution. It is apparent that a large search space will affect efficiency of the resolution process, however the search space should be conservatively selected as it must contain the true antenna position.

The second step is identifying the correct combination of integer ambiguities. A common approach is selecting the integer combination that minimizes the sum of squared residuals in the sense of least squares adjustment, i.e. the solution that best fits the data should be the correct result. The result of the least-square adjustment is often referred to as float solutions as the ambiguities are estimated as floating-point numbers. The float solution is then transformed to a integer solutions candidates by choosing the two closest integers as integer solutions. However, none of the integer combinations may be significantly better than the other and no selection can be made. A way of verifying the correctness of the selected combinations is needed, leading to the third step.

The third step is validation of the selected integer combinations. It has been shown that only the best and second-best integer candidate solutions must be validated (Teunissen and Verhagen, 2004), which led to the ratio test. The ratio is formed by the square norm of the second-best ambiguity residual vector and the squared norm of the best ambiguity residual vector. The ratio is compared against a threshold, often referred to as the critical value. The critical value is often set to 3, a threshold which has been determined empirically and is employed by many RTK softwares(Hofmann-Wellenhof et al., 2008). Thus the integer ambiguity is fixed to the integer candidate with the highest ratio above the critical value. If none of the candidates have a ratio above the critical value the ambiguity remains unsolved for that measurement epoch.

The by far most popular and successful resolution algorithm is the Least-Squares Ambiguity Decorrelation Adjustment (LAMBDA) method, delivering the optimum success rate of all admissible integer estimators. (Teunissen et al., 1997; Hofmann-Wellenhof et al., 2008)

3.5 Cycle Slip

Real-Time Kinematic positioning requires the accurate tracking of the number of carrier cycles between the receiver and the satellite. This is not always possible and sudden jumps, referred to as cycle slips, are inevitable. Cycle slips are defined as a sudden jump in the carrier-phase measurement due to loss of carrier lock in the phase-lock loop of a GNSS receiver. A schematic representation can be seen in Fig. 3.3, where the phase of the carrier wave is measured and plotted over time. A fairly smooth curve should be obtained, represented by the dashed line. However if a cycle slip occurs an instantaneous jump in the measured phase between by one or multiple number of carrier cycles, represented by the solid line.

There are multiple sources to cycle slips, (Hofmann-Wellenhof et al., 2008) identifies sources for cycle slips from the most probable to the least

1. Obstruction of the satellite signal due to trees, buildings, mountains.
2. Low Signal-to-Noise(S/N)-ratio due to bad ionospheric conditions, high receiver dynamics, or low satellite elevation
3. Incorrect signal processing by receiver software or malfunctioning satellite oscillators.

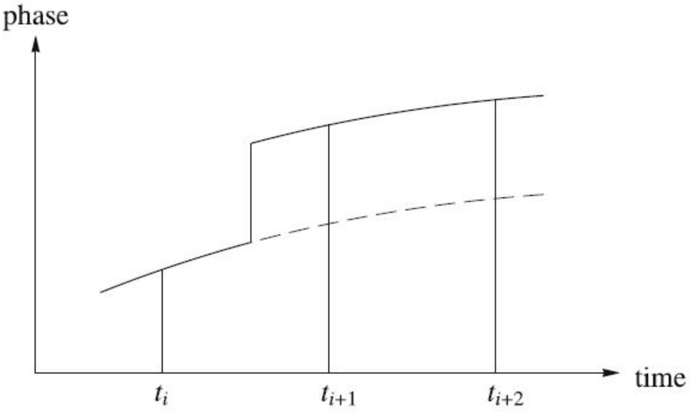


Figure 3.3: Schematic representation of a cycle slip. Figure from (Hofmann-Wellenhof et al., 2008)

Unless cycle slips are dealt with a reduction in accuracy ensues. Once a cycle slip has been detected there are two ways of handling it, either repairing the cycle slip or resetting the integer ambiguity and restarting the process of resolving the ambiguity. The latter is more computationally intensive and time-consuming, but is more secure as cycle slips maybe incorrectly repaired.

3.6 Accuracy

The precision of a pseudorange derived from code measurements is usually about one percent of the chip length, i.e length of the Pseudorandom Noise (PRN) sequence. The C/A-code is modulated onto the L1 carrier wave at 1.023 MHz and the P-code is modulated onto the L1 and L2 carrier wave at 10.23 MHz. The C/A-code has a chip length of 293.05m, yielding a precision of 2.9305m. The P-code has a chip length of 29.32m, yielding an achievable precision of 0.293m.

The precision of pseudoranges derived from phase measurements, assuming that the phase of the electromagnetic wave can be accurately measured to within 0.01 of a cycle results in an achievable precision of 1.9 mm. It should be noted that millimeter accuracy is only achievable with the receiver in a static configuration, i.e. the receiver is not moving and with a baseline not exceeding 20 km. The accuracy in a kinematic configuration is at the centimeter level if the integer ambiguity has been correctly resolved and at the decimeter level if it is unresolved, i.e float ambiguity.

3.7 Determination of RTKGPS Time-Delay

All GNSS receivers suffer from a non-constant processing delay, i.e the time it takes to calculate the position of the receiver (Solomon et al., 2011). For low-dynamic applications the delay may be neglected as it will have a small influence on the precision. However for dynamic high-accuracy applications a delay of only a tenth of a second will degrade the accuracy of the position measurements as the receiver may have moved considerably since the time of measurement.

The time-delay is identified using the approach described in (Hansen et al., 2015), using two signals from the GNSS receiver: pulse-per-second(PPS) and serial data. Pulse-per-second or PPS is a 1 Hz timepulse synchronized with GNSS-time, i.e the rising edge of the pulse will closely coincide with GNSS-time incrementing with a second. Since the rising edge of the PPS signal closely coincides with the measurement of the GNSS wave-signal it can be used to signify the Time of Validity (TOV) of the data. Ideally the data from the measurement would be available for use at TOV, however first the receiver must convert the wave-signal to its internal format and estimate position and velocity. Finally the data must be transmitted before it is available for use. The time at which the entire data signal is received is denoted Time of Arrival (TOA). Thus time elapsed from the rising edge of the time-pulse (TOV) to the last falling edge of the serial data line (TOA) is the time-delay τ of the GNSS-receiver, see Fig. 3.4.

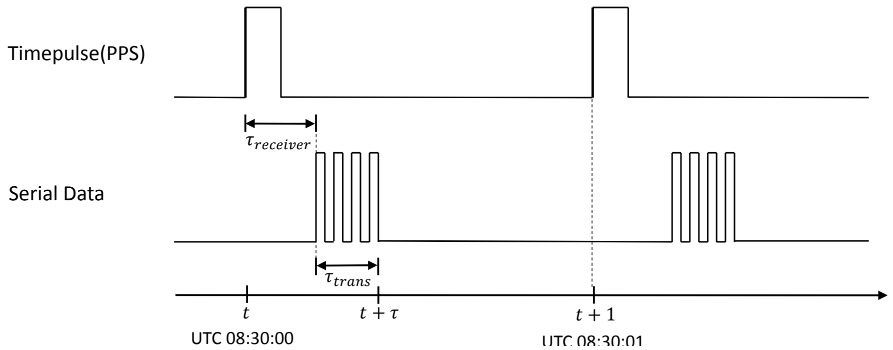


Figure 3.4: The timing of the pulse-per-second(PPS) signal and the serial data line of a GNSS-receiver

Fig. 3.4 illustrates the case where the update rate of GNSS-receiver is 1 Hz. If the update rate of the GNSS-receiver is higher than the PPS frequency, multiple data transmissions will occur within two consecutive timepulses. In this case the first delay will be determined as described above and the delays of the consecutive data transmissions are estimated as the time between the TOA of the data, shown in Fig. 3.5

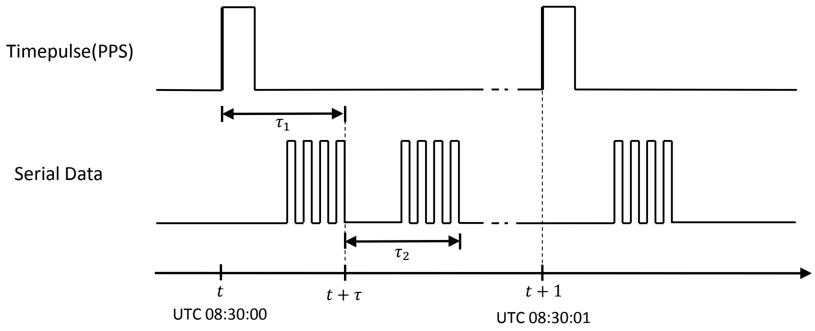


Figure 3.5: The timing of the pulse-per-second(PPS) signal and the serial data line of a GNSS-receiver with a update rate higher than 1 Hz

A Real-Time Kinematic GNSS-system requires additional processing of the data from the GNSS-receiver and data from a second base receiver through a radio-link. Commercial high-precision RTK-receivers perform the additional processing internally, thus approach described above can be used to determine the delay. However, a RTK GPS system may also be built using consumer-grade GNSS-receivers together with RTK GPS software running on an external computer. This introduces additional processing and transmission delays, see Fig. 3.6.

At $t+\tau_{\text{receiver}}$, the receiver has computed the pseudoranges and carrier-phase measurements for the given epoch and transmitted the data to the external computer running the RTK-software. The raw measurement data, i.e. pseudoranges and carrier-phase measurements are processed by the RTK-software. It should be noted that both delays consists of a processing delay followed by a transmission delay.

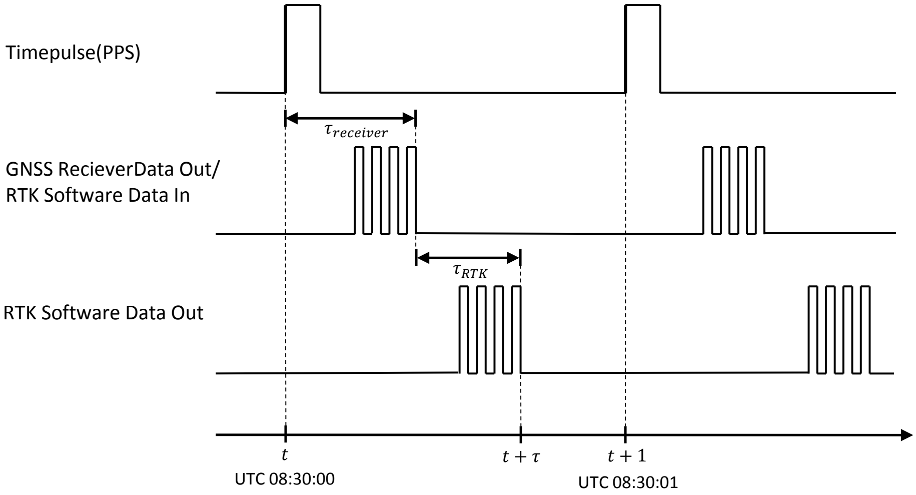


Figure 3.6: The addition of RTK software introduces an additional time-delay, τ_{RTK}

Chapter 4

Observers

This chapter presents the observers used in this thesis. First an introduction to the general observer methodology is given, followed by a description of the time-delayed attitude, position and velocity observers. The final section covers the discretization of the continuous-time observer equations.

4.1 General Observer Methodology

This section describes the general observer methodology given in (Grip et al., 2011c). Given the structure shown in Fig. 4.1, an observer for the first subsystem Σ_1 is to be implement. This observer is however dependent on the output z, which is not available and can therefore not be implemented. Instead an estimate of the output $\hat{z}$ is used rather than z. This estimate is obtained by extending the state space of the second subsystem Σ_2 to include the output of the first subsystem and construct an observer for this system.

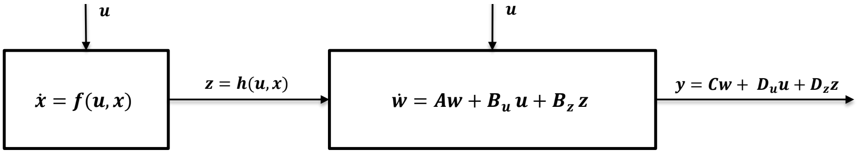


Figure 4.1: Nonlinear and linear systems in cascade

Consider the nonlinear system Σ_1 and the linear system Σ_2, shown in Fig. 4.1 and given by the following equations

$$\Sigma_1 : \begin{cases} \dot{x} = f(u,x) \\ z = h(u,x) \end{cases} \tag{4.1}$$

$$\Sigma_2 : \begin{cases} \dot{w} = Aw + B_u u + B_z z \\ y = Cw + D_u u + D_z z \end{cases} \tag{4.2}$$

It is assumed that an observer for the nonlinear system Σ_1 is on the form

$$\dot{\hat{x}} = f(u,\hat{x}) + g(u,\hat{x},z) \tag{4.3}$$

with the error dynamics

$$\dot{\tilde{x}} = e(t,\tilde{x}) \tag{4.4}$$

where $\tilde{x} := x - \hat{x}$ and $e(t,\tilde{x}) := f(u(t),x(t)) - f(u(t),x(t)-\tilde{x}(t)) - g(u(t),x(t)-\tilde{x}(t),z(t))$. It is further assumed that there exists a continuously differential function $V : \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}_{\geq 0}$ and positive constants $\alpha_1,...,\alpha_4$ such that for all $(t,\tilde{x}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_x}$

$$\alpha_1\|\tilde{x}\|^2 \leq V(t,\tilde{x}) \leq \alpha_2\|\tilde{x}\|^2 \tag{4.5}$$

$$\frac{\partial V}{\partial t}(t,\tilde{x}) + \frac{\partial V}{\partial \tilde{x}}(t,\tilde{x})e(t,\tilde{x}) \leq -\alpha_3\|\tilde{x}\|^2 \tag{4.6}$$

$$\|\frac{\partial V}{\partial \tilde{x}}(t,\tilde{x})\| \leq \alpha_4\|\tilde{x}\|^2 \tag{4.7}$$

As z eventually will be replaced by the estimate $\hat{z}$ a measure of the sensitivity of $g(u,\hat{x},z)$ to changes in z. It is therefore assumed that there exists an $L_1 > 0$ such that for all $(t,\tilde{x},\hat{z}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_x} \times \mathbb{R}^{p_z}$ the following inequality holds

$$\|g(u(t),\hat{x},z(t)) - g(u(t),\hat{x},\hat{z})\| \leq L_1\|z(t) - \hat{z}\| \tag{4.8}$$

Lastly the function

$$d(u,\dot{u},x) = \frac{\partial h}{\partial x}(u,x)\dot{u} + \frac{\partial h}{\partial x}f(u,x) \tag{4.9}$$

is defined and it is assumed that there exists an $L_2 > 0$ such that for all $(t,\hat{x}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_x}$,

$$\|d(u(t),\dot{u}(t),x(t)) - d(u(t),\dot{u}(t),\hat{x}\| \leq L_2\|x(t) - \hat{x}\| \tag{4.10}$$

It should be noted that Eq. 4.9 corresponds to the time derivative of the output $z = h(u,x)$.

The linear system Σ_2 is extended to include the output of the nonlinear system Σ_1 as an additional state. The state vector of the extended system is then given by $w_e = \begin{bmatrix} w^\top & z^\top \end{bmatrix}^\top$

$$\dot{w}_e = \mathcal{A}w_e + \mathcal{B}_u u + \mathcal{B}_d d(u, \dot{u}, x) \tag{4.11}$$

$$y = \mathcal{C}w_e + D_u u \tag{4.12}$$

where

$$\mathcal{A} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix},\ \mathcal{B}_u = \begin{bmatrix} B_u \\ 0 \end{bmatrix},\ \mathcal{B}_d = \begin{bmatrix} 0 \\ I \end{bmatrix},\ \mathcal{C} = \begin{bmatrix} C & D_z \end{bmatrix},$$

The observer for the extended system Eq. 4.11 is found by first defining the observers $\hat{w}$ and $\hat{z}$

$$\dot{\hat{w}} = A\hat{w} + B_u u + B_z\hat{z} + K_w(y - C\hat{w} - D_u u - D_z\hat{z}) \tag{4.13}$$

$$\dot{\xi} = \frac{\partial h}{\partial x}(u, \hat{x})g(u, \hat{x}, \hat{z}) + K_z(y - C\hat{w} - D_u u - D_z\hat{z}) \tag{4.14}$$

$$\hat{z} = h(u, \hat{x}) + \xi \tag{4.15}$$

where $\hat{w}$ and $\hat{z}$ are estimates of w and z. K_w and K_z are observer gains. Then the existing observer for the nonlinear system is altered to use the estimate $\hat{z}$ instead of z , i.e.

$$\dot{\hat{x}} = f(u, \hat{x}) + g(u, \hat{x}, \hat{z}) \tag{4.16}$$

Differentiating $\hat{z}$ and defining $\hat{w}_e = \begin{bmatrix} \hat{w}^\top & \hat{z}^\top \end{bmatrix}^\top$

$$\dot{\hat{w}}_e = \mathcal{A}\hat{w}_e + \mathcal{B}_u u + \mathcal{B}_d d(u, \dot{u}, x) + \mathcal{K}(y - \mathcal{C}\hat{w}_e - D_u u) \tag{4.17}$$

where $\mathcal{K} = \begin{bmatrix} K_w^\top & K_z^\top \end{bmatrix}^\top$.

The error dynamics of the observer $\dot{\hat{w}}_e$ is defined from Eq. 4.11 and 4.17, with $\tilde{w} := w_e - \hat{w}_e$,

$$\dot{\tilde{w}}_e = (\mathcal{A} - \mathcal{K}\mathcal{C})\tilde{w}_e + \mathcal{B}_d\tilde{d}(t, \tilde{x}) \tag{4.18}$$

where $\tilde{d}(t, \tilde{x}) := d(u(t), \dot{u}(t), x(t)) - d(u, \dot{u}, x(t) - \tilde{x})$

The error dynamics of the observer for Σ_1 is defined by

$$\dot{\tilde{x}} = e(t, \tilde{x}) + g(u, \hat{x}, z) - g(u, \hat{x}, \hat{z}) \tag{4.19}$$

$$H(s) = (sI - \mathcal{A} + \mathcal{K}\mathcal{C})^{-1}\mathcal{B}_d \tag{4.20}$$

From the stability proof given in (Grip et al., 2011c) it follows that if K is chosen such that $\mathcal{A} + \mathcal{K}\mathcal{C}$ is Hurwitz and $\|H(s)\|_\infty < \gamma := 4\alpha_3/(4L_2^2 + \alpha_4^2 L_1^2)$ and there exists a $\gamma^* > 0$ such that $\gamma > \gamma^*$ for which $\mathcal{K}$ can be chosen such that $\mathcal{A} + \mathcal{K}\mathcal{C}$ is Hurwitz, then the error dynamics $\dot{\tilde{w}}$ and $\dot{\tilde{x}}$ are globally exponentially stable.

4.2 Nonlinear Time-Delayed Observer

This section presents the nonlinear time-delayed position, velocity and attitude observers used in this thesis.

4.2.1 System Equations

The system equations describing the time-delayed navigation vector and gyro bias are given by

$$\dot{p}^n(t-\tau) = v^n(t-\tau) \tag{4.21}$$

$$\dot{v}^n(t-\tau) = g^n(p^n(t-\tau)) + R_b^n(q_b^n(t-\tau))f^b(t-\tau) \tag{4.22}$$

$$\dot{q}_b^n(t-\tau) = \frac{1}{2}q_b^n(t-\tau) \otimes \bar{\omega}_{ib}^b(t-\tau) \tag{4.23}$$

$$\dot{b}^n(t-\tau) = 0 \tag{4.24}$$

where τ is a time delay, $p^n(t-\tau)$ is the delayed position expressed in the NED-frame, $v^n(t-\tau)$ is the delayed velocity expressed in the NED-frame, $g^n(p^n)$ is the position dependent plumb bob gravity, $q_b^n(t-\tau)$ is the quaternion representation of the attitude between the NED and Body-frame. The operator $\otimes$ denotes the Hamiltonian quaternion product, $\bar{x}$ refers to a quaternion with real part equal to zero and imaginary part x, i.e $\bar{x} = [0 \;\; x]^\top$. The structure of the time-delayed system is divided into two subsystem; the first system containing the attitude and bias equations and the second containing the translational motion equations, as seen in Fig. 4.2.

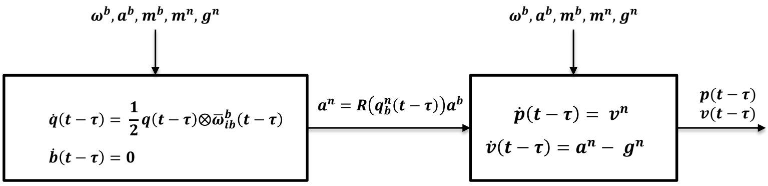

$\omega^b, a^b, m^b, m^n, g^n$
$\omega^b, a^b, m^b, m^n, g^n$
$\dot{q}(t-\tau) = \frac{1}{2}q(t-\tau)\otimes\bar{\omega}^b_{ib}(t-\tau)$
$\dot{b}(t-\tau) = 0$
$a^n = R(q^n_b(t-\tau))a^b$
$\dot{p}(t-\tau) = v^n$
$\dot{v}(t-\tau) = a^n - g^n$
$p(t-\tau)$
$v(t-\tau)$

Figure 4.2: Structure of the time-delayed system

The following sensor configuration and assumptions are prerequisites for the observers.

- Position measurements by an onboard GNSS-receiver experiencing a time-delay $p^e(t-\tau)$ in the ECEF-frame.
- Specific force of the vehicle, measured by an onboard IMU, f^b
- Angular velocity of the vehicle experienceing a constant bias b^b measured by an onboard IMU, $\omega^b_{ib} = \omega^b_{ib} + b^b$
- Magnetic field measurement of the Earths magnetic field observed from the vehicle, measured by an onboard magnetometer, m^b

4.2.2 Attitude Observer

The attitude observer is based on (Mahoney et al., 2008; T.Hamel and Mahoney, 2006) ,where the attitude is determined by comparing vectors measured in the BODY-frame with reference vectors in the reference frame. The quaternion-based attitude observer (Grip et al., 2011b) is given by

$$\dot{\hat{q}}^n_b(t-\tau) = \frac{1}{2}q^n_b(t-\tau)\otimes\left(\bar{\omega}^b_{ib}(t-\tau) - \bar{\hat{b}}^b(t-\tau) + \bar{\hat{\sigma}}(t-\tau)\right) \tag{4.25}$$

$$\dot{\hat{b}}^b(t-\tau) = \mathrm{Proj}(\hat{b}^b(t-\tau), -k_I\hat{\sigma}(t-\tau)) \tag{4.26}$$

The Proj$(\cdot,\cdot)$ term refers to parameter projection function given by

$$\mathrm{Proj}(\hat{b}^b, -k_I\hat{\sigma}(t-\tau)) = \begin{cases} I - \frac{\widehat{b^b}^\top\hat{b}^b}{\widehat{b^b}\widehat{b^b}^\top} & \text{if } \widehat{b^b}^\top\hat{b}^b > \|M_b\| \;\wedge\; \widehat{b^b}^\top\dot{b} > 0 \\ -k_I\hat{\sigma}(t-\tau) & \text{otherwise} \end{cases} \tag{4.27}$$

The projection function ensures that the gyro bias $\hat{b}^b$ is limited to a compact set defined by a sphere with radius $\|M_b\|$, i.e $\hat{b}^b > \|M_b\|$, where M_b is the upper bound for the gyro bias. The operator $\|.\|$ denotes the Euclidian norm for vectors.

The injection term $\hat{\sigma}$ is given by

$$\begin{aligned}\hat{\sigma}(t-\tau) &= k_1 v_1^b(t-\tau) \times R_b^n(\hat{q}_b^n(t-\tau))^\top v_1^n(t-\tau) \quad (4.28)\\ &+ k_2 v_2^b(t-\tau) \times R_b^n(\hat{q}_b^n(t-\tau))^\top v_2^n(t-\tau)) \quad (4.29)\end{aligned}$$

The reference vectors v_1^b, v_2^b ,v_1^n and v_2^n are given by

$$v_1^b = \frac{f^b}{\|f^b\|}, \quad v_2^b = \frac{m^b}{\|m^b\|} \times v_1^b \quad (4.30)$$

$$v_1^n = \frac{\hat{f}^n}{\|\hat{f}^n\|}, \quad v_2^n = \frac{m^n}{\|m^n\|} \times v_1^n \quad (4.31)$$

where f^b is the specific force in the BODY-frame, m^b is the magnetic field vector decomposed in the BODY-frame, m^n is the magnetic field vector decomposed in the NED-frame and $\hat{f}^b$ is an estimate of the specific force decomposed in the NED-frame.

The observer guarantees almost-global asymptotic stability under the assumption that either the reference vectors are constant or the reference vectors are time-varying, but the gyro measurements are bias-free. In (Grip et al., 2011a) the stability analysis is extended to semi-global exponential stability for time-varying reference vectors.

4.2.3 Position and Velocity Observer

In (Grip et al., 2011b) the design methodology from Sec. 4.1 is used to design a nonlinear GNSS/INS integration observer. This observer has in turn been modified in (Hansen et al., 2015) to account for delayed GNSS-measurements. The time-delayed observer equations are given by

$$\begin{aligned}\dot{\hat{p}}^n(t-\tau) =& \hat{v}^n(t-\tau) + \theta K_{pp}(R_n^e(\Theta_{en})^\top p^e(t-\tau) - p^n(t-\tau)) \quad (4.32)\\ \dot{\hat{v}}^n(t-\tau) =& \hat{f}^n(t-\tau) + g^n(p^n(t-\tau)) \quad (4.33)\\ &+ \theta^2 K_{vp}(R_n^e(\Theta_{en})^\top p^e(t-\tau) - p^n(t-\tau)) \quad (4.34)\\ \dot{\xi}(t-\tau) =& - R_b^n(\hat{q}_b^n(t-\tau))S(\hat{\sigma}(t-\tau))f^b(t-\tau) \quad (4.35)\\ &+ \theta^3 K_{\xi p}(R_n^e(\Theta_{en})^\top p^e(t-\tau) - p^n(t-\tau)) \quad (4.36)\\ \hat{f}^n(t-\tau) =& R_b^n(\hat{q}_b^n(t-\tau))f^b(t-\tau) + \xi(t-\tau) \quad (4.37)\end{aligned}$$

where constant parameter $\theta \geq 1$ is a tuning parameter. The $S(\cdot)$ operator denotes a skew-symmetric matrix. The gain-matrices K_{pp}, K_{vp} and $K_{\xi p}$ must be chosen such that $\mathcal{A} - \mathcal{C}\mathcal{K}$ is Hurwitz, where

$$\mathcal{A} = \begin{bmatrix} 0 & I_{3x3} & 0 \\ 0 & 0 & I_{3x3} \\ 0 & 0 & 0 \end{bmatrix}, \; \mathcal{C} = \begin{bmatrix} I_{3x3} & 0 & 0 \end{bmatrix}, \; \mathcal{K} = \begin{bmatrix} K_{pp} \\ K_{vp} \\ K_{\xi p} \end{bmatrix} \tag{4.38}$$

in order to ensure semi-global exponential stability. The only difference between the observer given in (Grip et al., 2011b) and (Hansen et al., 2015) is time-shifting the inputs. Therefore the stability proof of the time-delayed observer follows directly from (Grip et al., 2011b) as shifting the observer in time does not affect the stability of the system.

4.2.4 Observer Structure

The complete observer structure is shown in Fig. 4.3. The Attitude estimator is depicted as a single block to reduce the complexity of the figure. The transformation $f^n(t-\tau) = R_b^n(\hat{q}_b^n(t-\tau))f^b(t-\tau)$ is for the sake of simplicity included in attitude estimation block. Furthermore the single connection between the attitude observer and the fast simulator represents three separate inputs and not a vector.

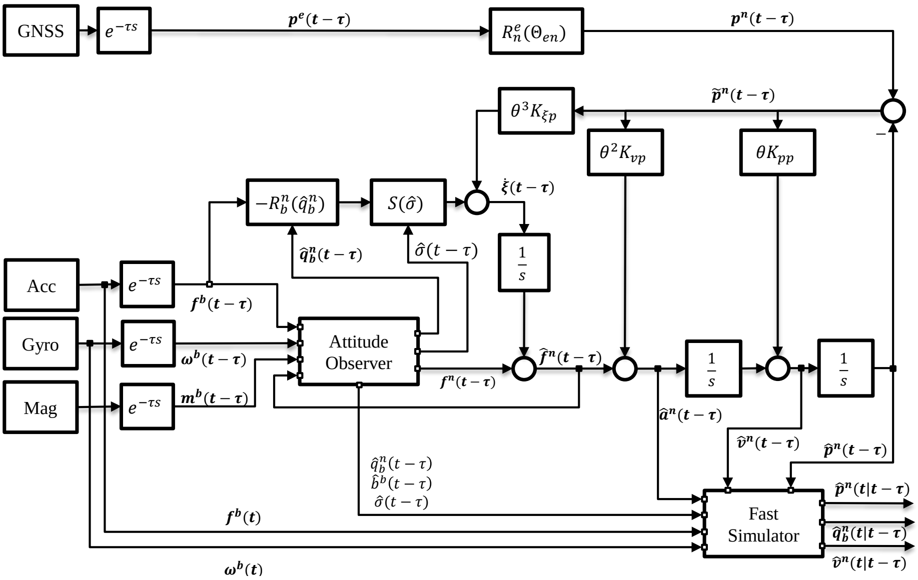


Figure 4.3: The nonlinear observer structure

4.2.5 Fast Simulator

The nonlinear estimator only estimates the delayed velocity and position , i.e. $\hat{v}^n(t-\tau)$ and $\hat{p}^n(t-\tau)$. A faster than real-time simulator is needed to calculate the current position and velocity estimates based on the buffered data (Hansen et al., 2015). Position and velocity in the NED-frame may be represented as a double integrator system

$$\dot{x} = Ax + Bu \quad A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{4.39}$$

where $x = \begin{bmatrix} p^n & v^n \end{bmatrix}^\top$ and $u = \begin{bmatrix} 0 & a^n \end{bmatrix}^\top$. The solution to the double integrator system is given by

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-s)}Bu(s)ds \tag{4.40}$$

Based on this (Hansen et al., 2015) proposes a fast simulator where $\hat{p}^n(t|t-\tau)$ and $\hat{v}^n(t|t-\tau)$, i.e the position and velocity at time t given that the position and velocity at time $t-\tau$ is known, is given by

$$\hat{v}^n(t|t-\tau) = \hat{v}^n(t-\tau) + \int_{t-\tau}^{t} R_b^n(\hat{q}_b^n(r))a^b(r)dr \tag{4.41}$$

$$\hat{p}^n(t|t-\tau) = \hat{p}^n(t-\tau) + \tau\hat{v}^n(t-\tau) + \int_{t-\tau}^{t}\int_{t-\tau}^{s} R_b^n(\hat{q}_b^n(r))a^b(r)drds \tag{4.42}$$

where a^b is the specific force given in the BODY-frame. The attitude from $t-\tau$ to t is not available since the attitude observer only has access to the delayed measurements. Consequently the fast simulator is expanded to estimate the attitude faster than real-time based on the data in the storage buffers using Eq. 4.25 and 4.26 . However it is assumed that the injection term is slow-varying, i.e. $\sigma(t-\tau) = \sigma(t)$. It is also assumed that the gravitational field does not change considerably during the interval $t-\tau$ to t , i.e. $g^n(t-\tau) = g^n(t)$.

4.3 Discretization of Observer Equations

The nonlinear observer is given in continuous-time and must be discretized in order to be realized in a physical system. The Corrector-Predictor approach described in (Fossen, 2011) is an effective to way to implement systems with multiple sensors that have different measurement rates. Given a nonlinear system in the form

$$\dot{x} = f(x,u) \tag{4.43}$$

the continuous-time observer for the nonlinear system may be expressed as

$$\dot{\hat{x}} = f(\hat{x}, u) + \gamma(y, \hat{y}) \tag{4.44}$$

where $\hat{x}$ is the observer state vector, u is a vector of inputs $f(\hat{x}, u)$ is a nonlinear function describing the relationship between $\hat{x}$ and u and the linear injection term

$$\gamma(y, \hat{y}) = K(y - \hat{y}) \tag{4.45}$$

The discrete-time corrector predictor formulation for the nonlinear observer model Eq. 4.44 in terms of Euler integration is given by

$$\hat{x}(k) = \bar{x}(k) + K_d \left[y(k) - \bar{y}(k) \right] \tag{4.46}$$

$$\bar{x}(k+1) = \bar{x}(k) + hf(\hat{x}(k), u(k)) \tag{4.47}$$

with the gain $K_d = hK$, where h is the sampling time and K is the injection term gain. At each time a measurement $y(k)$ is available, the corrector step updates the state $\bar{x}(k)$ to $\hat{x}(k)$ based on the measurement. When no measurement $y(k)$ is available the predictor step predicts the state based on the nonlinear system model Eq. 4.43 until a new measurement is available, also referred to as dead-reckoning

$$\bar{x}(k+1) = \bar{x}(k) + hf(\bar{x}(k), u(k)) \tag{4.48}$$

Using Eq. 4.46 and 4.47 the discrete-time corrector predictor formulation for the nonlinear observer, Eq. 4.32 - 4.35 results in the following corrector equations

$$\hat{p}^n(k) = \bar{p}^n(k) + h_{gps}\theta K_{pp}(R_n^e(\Theta_{en})^\top p^e(k) - \bar{p}^n(k)) \tag{4.49}$$

$$\hat{v}^n(k) = \bar{v}^n(k) + h_{gps}\theta^2 K_{vp}(R_n^e(\Theta_{en})^\top p^e(k) - \bar{p}^n(k)) \tag{4.50}$$

$$\hat{\xi}(k) = \bar{\xi}(k) + h_{gps}\theta^3 K_{\xi p}(R_n^e(\Theta_{en})^\top p^e(k) - \bar{p}^n(k)) \tag{4.51}$$

where h_{gps} is the sampling rate of the GPS receiver. The predictor equations are given by

$$\bar{p}^n(k+1) = \bar{p}^n(k) + h_{imu}\bar{v}^n(k) \tag{4.52}$$

$$\bar{v}^n(k+1) = \bar{v}^n(k) + h_{imu}(f^b(k) - g^n(\bar{p}^n(k))) \tag{4.53}$$

$$\bar{\xi}(k+1) = \bar{\xi}(k) - h_{imu}R_b^n(\hat{q}_b^n(k))S(\hat{\sigma}(k))f^b(k) \tag{4.54}$$

where h_{imu} is the sampling rate of the inertial measurement unit (IMU). The Corrector-predictor approach is limited to differential equations with linear injection terms, thus the attitude observer equations are implemented using forward Euler integration, i.e.

$$\hat{q}_b^n(k+1) = \hat{q}_b^n(k) + h_{imu}\frac{1}{2}\hat{q}_b^n(k) \otimes \left(\bar{\omega}_{ib}^b(k) - \bar{\hat{b}}^b(k) + \bar{\hat{\sigma}}(k)\right) \tag{4.55}$$

$$\hat{b}^b(k+1) = \hat{b}^b(k) + h_{imu}\mathrm{Proj}(\hat{b}^b(k), -k_I\hat{\sigma}(k)) \tag{4.56}$$

Chapter 5

Navigation System Components

This chapter provides a brief description of all the software and hardware components used by the navigation system.

5.1 Software

The implementation of the online observer requires a run-time environment for sensor interaction and RTK software. This will be provided by the LSTS Toolchain and the Real-Time Kinematic Library (RTKLIB) respectively.

5.1.1 LSTS Toolchain

This is an open-source control architecture and software toolchain developed by the Underwater Systems and Technology Laboratory (LSTS), which is an interdisciplinary research laboratory at the Faculty of Engineering, University of Porto (Pinto et al., 2012, 2013). The toolchain supports networked aerial and ocean vehicle systems, and is composed of the embedded operating system GLUED, on-board software DUNE, shore-side control software Neptus and a communication protocol which is shared by all components, IMC.

5.1.1.1 Glued

GLUED is a minimal Linux distribution targeted at embedded systems developed by LSTS. It is platform independent, lightweight and easy configurable through a single configuration file. that is created for each specific system. The configuration file is created for each target system and specifies the system architecture, network settings and which software packages that will be a part of the target distribution. Since GLUED is cross-compile ready, the operating system is compiled on a more powerful desktop computer rather than the embedded system itself.

5.1.1.2 DUNE

DUNE: Unified Navigational Environment (DUNE) is an open-source on-board software solution for unmanned vehicles, providing an operating-system and architecture independent platform abstraction layer. It is a runtime environment used to write generic embedded software in C++, and is responsible not only for every interaction with sensors, payload and actuators, but also for communication, navigation, control, maneuvering, plan execution and vehicle supervision. It is designed to contain multiple relatively small independent tasks run separate threads of execution. They are all connected to a bus where they can publish and subscribe to messages for communication in between them, using the commands dispatch and consume, respectively. This leads to a high degree of modularity, where new tasks, like new sensors, actuators or controllers, can be easily added, and old tasks can be enabled and disabled freely. Messages passed to and from the bus are specified by the communication protocol IMC.

5.1.1.3 IMC

Inter-Module Communication (IMC) is a communication protocol developed by LSTS to be used in their toolchain. It defines a common message set understood by all the vehicles, sensors and consoles. DUNE task running on the individual vehicles may subscribe to the messages sent on the message bus, an example is given in Fig. 5.1. IMC already contains many standard message types, but users may easily add new and customized message types.

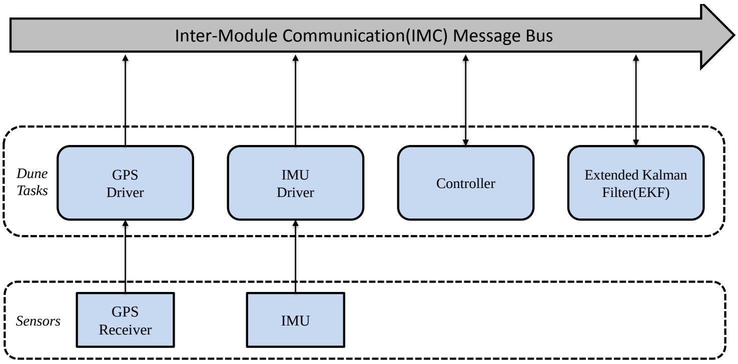

Inter-Module Communication(IMC) Message Bus
Dune Tasks
GPS Driver
IMU Driver
Controller
Extended Kalman Filter(EKF)
Sensors
GPS Receiver
IMU

Figure 5.1: Task Interaction and Message sending using the IMC Message Bus. Sensor, controller and estimator tasks communicate over the IMC message Bus.

5.1.1.4 Neptus

Neptus is an open-source command and control software for a single or fleets of unmanned vehicles and different types of sensors. Operators using Neptus can observe real-time data of networked vehicles, review previous missions and plan and simulate future missions. In this thesis Neptus is only used to extract the data logged by DUNE and to monitor the navigation system in-flight.

5.1.2 RTKLIB

Real-Time Kinematic Library (RTKLIB) is an open source program package for standard and precise positioning with several GNSS, among them GPS, Galileo, GLONASS and Bei- Dou(Takasu and Yasuda, 2009). RTKLIB supports several positioning modes for real-time and postprocessing applications. The RTKLIB program package consists of several GUI APs (graphical user interface application programs) for both Windows and CUI (command-line user interfaces) APs for UNIX, where two of the latter will be described in this section.

5.1.2.1 str2str

str2str is a stream splitter, taking one input stream and splitting it up into multiple output streams. Its function is to relay raw pseudorange data from one receiver to the real-time positioning AP rtkrcv, seen in Figure 5.2. For UAV applications, str2str is configured as a TCP server, taking the serial stream from the base receiver and making it accessible through a predefined port for subscribing rovers.

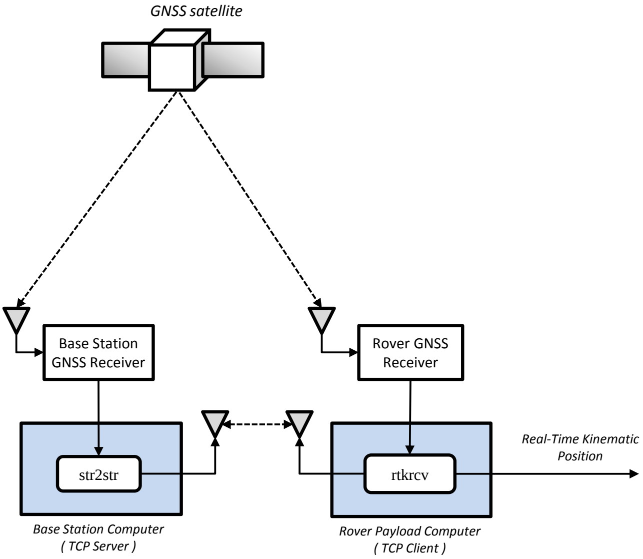

GNSS satellite
Base Station GNSS Receiver
Rover GNSS Receiver
str2str
rtkrcv
Real-Time Kinematic Position
Base Station Computer (TCP Server)
Rover Payload Computer (TCP Client)

Figure 5.2: The interaction between the RTKLIB command-line user interfaces str2str and rtkrcv

5.1.2.2 rtkrcv

rtkrcv is RTKLIB's real-time positioning AP and supports kinematic and moving-baseline positioning modes that may be used for high-precision positioning of UAVs, with fixed and moving base-station respectively. The AP is configured to obtain raw observation data from the base and rover GPS receivers through TCP/IP and serial connections, respectively. The input streams contains raw pseudorange and phase measurements and are double differenced to eliminate common errors. These double differenced phase and pseudorange measurements form the measurement vector of an Extended Kalman Filter (EKF) that estimates the position and integer ambiguities. The estimated rover position from the EKF is referred to as a float solution, i.e. a solution with unresolved ambiguity. The estimated carrierphase ambiguities are formulated as an Integer Least-Squares (IRL) problem and is solved using the LAMBDA and MLAMBDA methods (Takasu and Yasuda, 2009)

5.2 Hardware

This section describes the main hardware components of the payload. The additional circuits and Printed Circuit Boards(PCBs) used to interface the components are described in Ch. 6

5.2.1 UAV

The UAV used in this thesis is the X8 Skywalker from Skywalker Technologies. seen in Fig. 5.3. It is based on a flying-wing design, giving it excellent glide performance as well as fast low power cruise speed. The airframe is constructed from molded expanded polyolefin (EPO) foam, making the frame very robust. It has a large space within the fuselage making it ideal for experimental payloads.

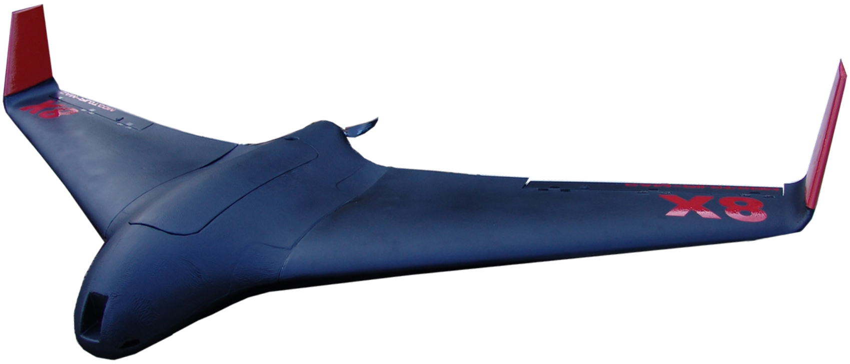


Figure 5.3: X8 Skywalker from Skywalker Technologies, Picture from www.campilot.tv

The wingspan of 2120mm allows for a maximum aircraft gross weight (AUW) of 3500g. Additional specifications can be found at (HobbyKing, 2015). The UAV has been fitted with the following components by the AMOS UAV-Lab at NTNU in order to be flyable:

TX	Spektrum DX7s
RX	Spektrum Remote Receiver SPM9645
Servo	Hitec HS-5085MG
Motor	Hacker A40-12S V2 14-Pole
ESC	Master Spin 66 Pro
Propeller	Aeronaut 13x8 folding prop.
Battery	2x Zippy Compact 4S 5000mAh 25C
Autopilot	3D Robotics Pixhawk w/3DR uBlox GPS with Compass kit and airspeed sensor.
Telemetry link	3D Robotics radio (433MHz) (VLOS operations)

5.2.2 Embedded Computer

The embedded computer in the UAV is the core of the navigation system and there are multiple requirements that must be satisfied. Firstly, the embedded computer has to be able to run the LSTS toolchain and RTK positioning algorithms in real-time. Secondly, weight and size are key factors in terms of flight endurance and payload capabilities, respectively. Furthermore, it must support the communication interfaces used by the peripheral components of the navigation system.

The BeagleBone Black seen in Figure 5.4 has been adopted by the UAV-Lab at NTNU as the preferred embedded computer for unmanned aerial vehicles. It is a small, low-cost, open source community-supported development platform built around the 1 GHz Sitara AM335x ARM Cortex-A8 processor from Texas Instruments and supports SPI,TTL Serial and I_2C. Further information and complete specifications can be found in (BeagleBoard, 2015)

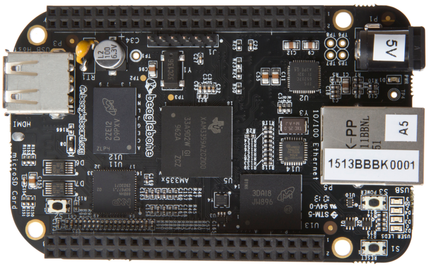


Figure 5.4: BeagleBone Black

5.2.3 GNSS Receiver

The uBlox EVK-6T, seen in Figure 5.5, is an evaluation kit based around the ublox LEA-6T which outputs raw measurement data, i.e. pseudorange and phase measurements. Strictly speaking, the EVK-6T from uBlox is not a RTK-GPS receiver, but has RTK capabilities when used in conjunction with RTKLIB (Takasu and Yasuda, 2009). This system was used by (Skulstad and Syversen, 2014) to successfully perform repeatable automatic landings. Complete specifications can be found in (uBlox AG, 2015a,b)

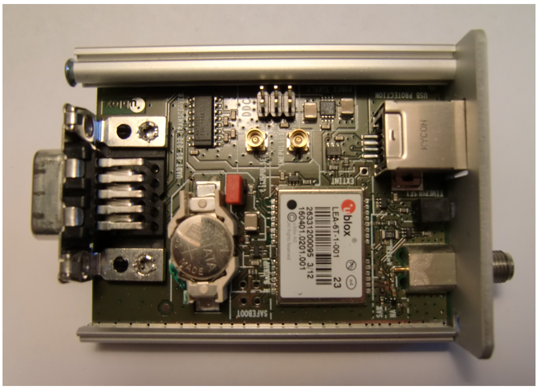


Figure 5.5: EVK-6T from uBlox AG with its protective cover removed

5.2.4 GNSS Antennas

The navigation system will use two GPS receivers, one in a static configuration and the other in a highly dynamic configuration. These two different applications impose distinct requirements on the individual antennas. It is essential that the rover antenna is small enough to be mounted on the UAV without affecting the aerodynamics. The antenna should be as light as possible in order to maximize flight time.

The M1227HTC-A-SMA from Maxtena, seen in Figure 5.6a, has been used by the Multicopter Group at the UAV-lab with good results. The antenna is small and compact and is extremely light weight, only 17 g. It is designed for L1/L2 GPS-GLONASS bands and may be used with out ground plane.Further information and the electrical specifications of the antenna can be found in (Maxtena, 2015)

The base station antenna does not have any restrictions on size or weight as it will be placed on the ground in proximity to the base station. However, placing the antenna close to the ground, buildings and other obstacles may lead to multipathing. Multipath rejection is therefore a key selection criteria when choosing the base station antenna. The base measurements should be as accurate as possible, imposing further requirements on interference handling, phase center stability and noise rejection.

The Novatel GPS-701-GG, seen in Figure 5.6b, was chosen as the base station antenna. It features reception in the L1 freguency band from both GPS and GLONASS stellites. More importantly the antenna provides good multipath rejection and has a highly stable phase center. Reception of both GPS and GLONASS signals allows for dual constellation applications, i.e. using both GPS and GLONASS satellite signals for positioning.

A stable phase center makes the antenna ideal for carrier-phase measurements and

(a) M1227HTC-A-SMA Helix antenna **(b)** Novatel GPS-701-GG

Figure 5.6: GNSS Antennas

baselines of any length. Furthermore it ensures that signal reception is unaffected by the rotation of the antenna or satellite elevation. Further information and the electrical specifications of the antenna can be found in (Novatel, 2015).

5.2.5 Inertial Measurement Unit

The ADIS16488CMLZ, seen in Fig. 5.7, is a complete inertial system that includes a triaxis gyroscope, a triaxis accelerometer, triaxis magnetometer, and pressure sensor. The factory calibration characterizes each sensor for sensitivity, bias, alignment, and linear acceleration. As a result, each sensor has its own dynamic compensation formulas that provide accurate sensor measurements.The SPI and register structure provide a simple interface for data collection and configuration control. Further information and the electrical specifications of the antenna can be found in (Analog Devices, 2015)

Figure 5.7: Analog Devices ADIS16488CMLZ

5.2.6 Radio Communication

A reliable radio link is essential for real-time kinematic GPS. The Rocket M5 from Ubiquiti Networks is a 5GHz Hi Power 2x2 MIMO AirMax TDMA Basestation, seen in Fig. 5.8. The Rocket M5 is capable of extremely long range performance of up to 50+ km and speeds exceeding 150 Mbps. Further information and the electrical specifications can be found in (Ubiquiti Networks, 2015)

Figure 5.8: Ubiquiti Networks M5 Rocket

Chapter 6

Implementation

This chapter describes the implementation of the observer structure described in Chap. 4 in C++ code as well as the hardware implementation of the payload.

6.1 Software Implementation

This sections describes the implementation of the nonlinear observer in the C++ programming language within DUNE. All the tasks are configurable through the vehicle specific .ini-file, which defines which tasks to run on the target system as well as the task parameters, e.g. serial ports, baudrates, controller gains. The following subsections describes the tasks developed in this thesis.

6.1.1 RTKGPS

In order to use the real-time kinematic positioning data within DUNE the sensor task RTKGPS was created. The position data is transmitted in a similar manner as a NMEA-sentence, but using whitespace as a delimiter instead of a comma. There are no start and end delimiters, but the sentence has a fixed length. The sentence order of is shown in Tab. 6.1.

Field 0 and 1 contains the UTC date and time respectively. RTKLIB may be configured to output the position estimate in ECEF-coordinates, LLH (Latitude, Longitude, Height) or ENU(East-North-Up). Depending on the option chosen, the position fields 2, 3 and 4 will change as indicated in Tab. 6.1. Field 5 contains the quality of the position data represented by an integer value. A value of 1 indicates that the ambiguity is fixed, a value of 2 indicates a float ambiguity and value of 5 indicate single receiver quality , i.e. no data received from the base receiver. Field

Table 6.1: RTKLIB Output Sentence

Field	**Description**
0	UTC year/month/day
1	UTC time
2	x-ecef(m), lat(deg) or e(m)
3	y-ecef(m), lon(deg) or n(m)
4	z-ecef(m), height(m) or u(m)
5	Quality
6	Number of Satelittes
7	sdx(m)
8	sdy(m)
9	sdz(m)
10	sdxy(m)
11	sdyz(m)
12	sdzx(m)
13	Age of Differential
14	Ambiguity Resolution Ratio

6 contains the number of satellites visible to the receiver, this includes satellites with low elevation angles. Fields 7 to 9 are the standard deviation of the x,y and z components respectively. Fields 10 to 13 are the square root of the value of the xy,yz and zx components of the estimated covariance matrix. Field 13 is the age of the differential correction , i.e. time difference between the observation data epochs of the rover receiver and the base station in seconds. Field 14 is the ambiguity resolution ratio, the certainty measure of ambiguity search algorithm.

The RTKGPS task spawns a child process that runs concurrently with the task itself. The child process handles the serial connection, i.e. maintains and continuously polls the serial connection. If a RTKLIB sentence is received the message IMC::DevDataText, containg the RTKLIB sentence, is dispatched to the IMC Message Bus. On initialization the parent task RTKGPS binds to the IMC::DevDataText message, i.e. it will consume all messages of that type sent on the IMC Bus. When a DevDataText message is consumed, the raw sentence is sent to the sentence parser which extracts the individual fields before sending the parsed data to the sentence interpreter. Finally the data is written into the correct fields of the IMC::RtkFix or the IMC::GpsFix message before dispatching it to the IMC Bus. The dataflow within the RTKGPS task is shown in Fig. 6.1.

The user configurable parameters for the RTKGPS Task are shown in Tab. 6.2. If the RTKGPS task does not receive any data from RTKLIB within the timeout period specified by the Input Timeout parameter, the task will restart itself. The Output Solution Format may be set to llh or ned, with the llh option set the RTKGPS task outputs the IMC::GpsFix message and ned option

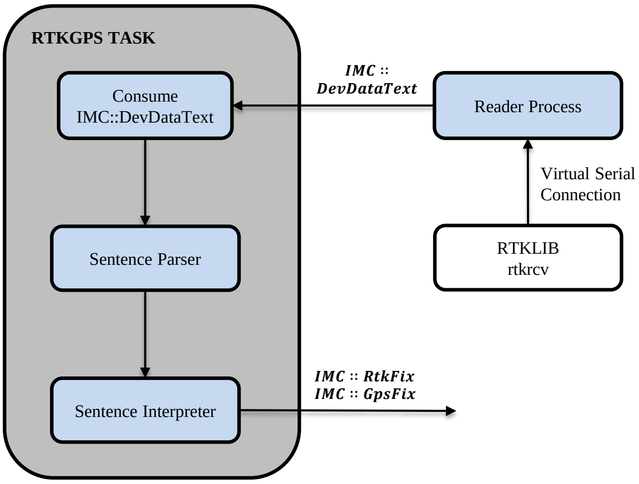


Figure 6.1: Data flow within the RTKGPS Task

Table 6.2: RTKGPS Task .ini-file Parameters

Parameter	**Description**
Serial Port - Device	Serial port device used to communicate with the sensor
Serial Port - Baud Rate	Serial port baud rate
Input Timeout	Number of seconds without input before task is restarted
Output Solution Format	Latitude,Longitude,Height: llh or NED-baseline: ned

6.1.2 LIMU

This task has not been written by the author, but it is briefly described as it is a key part of the software implementation. The LIMU (LSTS Ineartial Measurement Unit) Task is the driver task for the LIMU Board described in Sec. 6.2.3. The following IMC messages are dispatched by the task

- IMC::AngularVelocity
- IMC::Acceleration
- IMC::MagneticField
- IMC::Temperature

The user configurable parameters for the RTKGPS Task are shown in Tab. 6.3. The hard-iron offsets, i.e. the constant additive values to the output of each of the magnetometer axes, are specified by Hard-Iron Calibration option. The Raw Data option enables and disable the raw data output from the LIMU Board.

Table 6.3: LIMU Task .ini-file Parameters

Parameter	**Description**
Serial Port - Device	Serial port device used to communicate with the sensor
Power Channel - Name	Name of the power channel
Hard-Iron Calibration	Magnetometer Hard-Iron calibration Factors
Output Frequency	Output frequency of the LIMU Board
Raw Data	Set to true to enable raw data output

6.1.3 GpsImuNavigation

The nonlinear observer is implemented in the DUNE task GpsImuNavigation using the discretizied equtions from Sec. 4.3. The task binds to the following IMC messages

- IMC::Acceleration
- IMC::AngularVelocity
- IMC::MagneticField
- IMC::GpsFix

dispatched from the sensor task RTKGPS and LIMU. The interaction between the sensor tasks and the observer task is illustrated in Fig. 6.2. The tasks in DUNE run concurrently and the messages dispatched to the IMC Bus will be consumed in the order they were dispatched. The consume functions scale the sensor data to the appropriate units, e.g. Gauss to Tesla and milliG's to G's and sets flags indicating that new messages have arrived. The flags determine which equations should be used to update the observer estimates. The corrector equations should only update the estimates when a message from the RTKGPS is received and the predictor equations should only update the estimates when all the messages from the LIMU task for that given measurement epoch have been consumed.

Implementing the time delay the sensor measurements is done by using First-In-First-Out(FIFO) buffers, illustrated Fig. 6.2. The consume functions will append new sensor messages to the respective buffers simultaneously as the messages at the start of the buffer are used to update the variables used by the corrector-predictor equations, illustrated in 6.2.The size of the buffers is determined by the amount of messages that are received are equal to the sampling rate h times the time-delay τ. However only an integer number of measurements can be buffered and the size has to be rounded to the nearest integer value, thus being an approximation of the measured delay.

The earth's gravitational field is position dependent and should be taken in to account when implementing a navigation system. The quaternion-based attitude observer relies on reference vectors in order to compute the attitude. However these

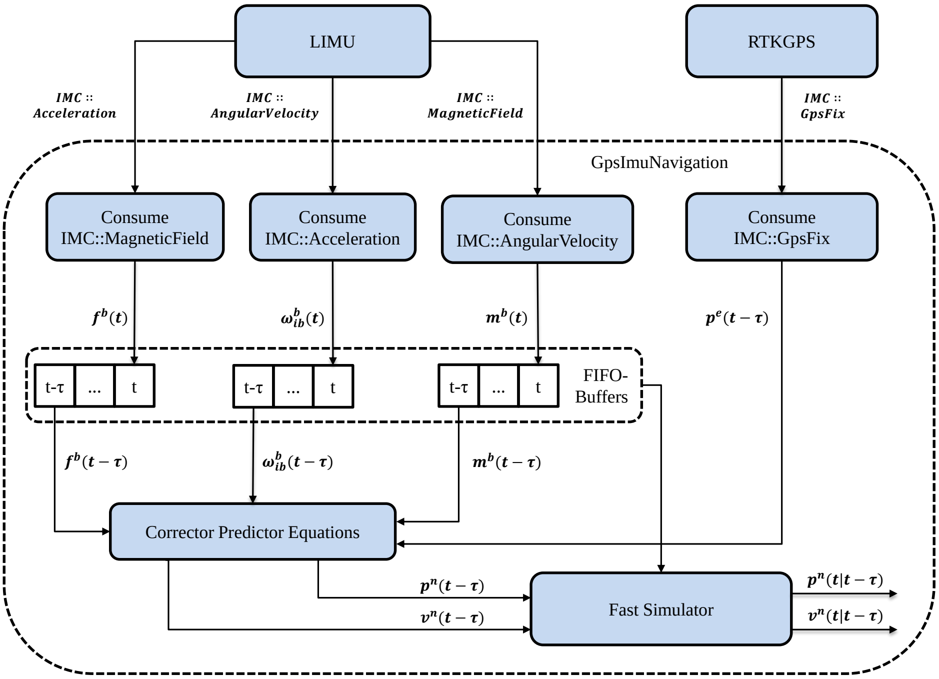


Figure 6.2: Interaction between the sensor tasks and the observer task, GpsImuNavigation

vectors are not measured and must therefore approximated through models that can be found in GeographicLib (Kerney, 2015), a set of C++ classes primarily used for geodetics.

GeographicLib's GravityModel class contains several models expand the gravitational potential of the earth as sum of spherical harmonics. The models also specify a reference ellipsoid, relative to which geoid heights and gravity disturbances are measured. Since the position of the UAV is given in the NED-frame, the **wgs84**, the WGS84 Reference Ellipsoid is the obvious choice . Then the gravitational field is approximated according to **egm84** , the Earth Gravity Model 1984.

The Magnetic Model Class contain several models that expand the internal magnetic potential of the earth as sum of spherical harmonics, but neglect magnetic fields due to the ionosphere, the magnetosphere, nearby magnetized materials and electric machinery. The model used is **igrf11** , the International Geomagnetic Reference Field (11th generation), which approximates the main magnetic field for the period 1900–2015. However for future use it is recommended to switch to **igrf12**, the 12th generation model, which approximates the main magnetic field for the period 1900–2020.

Table 6.4: GpsImuNavigation Task .ini-file Parameters

Parameter	**Description**
Simulation Mode	Set true for SIL/HIL simulations
GPS - Sample Rate	Sample rate of the GPS receiver
IMU - Sample Rate	Sample rate of the ineartial measurement unit
Time Delay	Time delay of the GNSS receiver
Initial Position - Latitude	Latitude of the origin of the local frame
Initial Position - Longitude	Longitude of the origin of the local frame
Initial Position - Height	Height of the origin of the local frame
Attitude Observer - Gain k1	Reference vector gain factor
Attitude Observer - Gain k2	Reference vector gain factor
Attitude Observer - Gain ki	Attitude observer gain
Attitude Observer - Mb	Upper bound on the Gyro Bias
PosVel Observer - Gain Kpp	Position Gain
PosVel Observer - Gain Kvp	Velocity Gain
PosVel Observer - Gain Kzp	Specific Force Estimate Gain
Path - Magnetic Model	Path to the Magnetic Model
Path - Gravity Model	Path to the Gravity Model

All the observer parameters may be changed through the .ini-file of the task without having to recompile the code. The parameters may also be changed in-flight through Neptus. The user .ini-file parameters for the observer task are listed in Tab. 6.4

6.2 Hardware Implementation

This section describes the system integration process , i.e how the hardware components from Ch. 5 are brought together to form the navigational payload.

6.2.1 Beaglebone Black Cape

The Beaglebone Black cape is an extension board for the Beaglebone black, designed at NTNU's UAV-Lab. The extension board board, seen in Fig 6.3, powers the Beaglebone, provides easy access to three of the serial ports of the Beaglebone Black and two power outputs connected to the input rail. The DC/DC converter steps the battery voltage down from 18 - 6.5v to 5v and can provide up to 1 A of current. This makes it possible to power peripheral devices in addition to the Beaglebone .

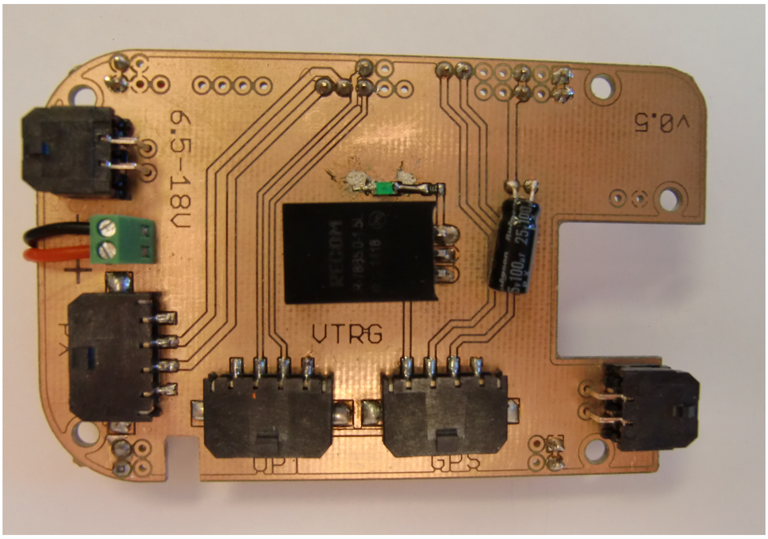


Figure 6.3: The BeagleBone Black Cape

6.2.2 RS232 Converter Circuit

The uBlox EVK-6T communicates either through the RS-232 or the USB interface, but the RS-232 interface should be used in order to access the timingpulse outputs. RS-232 is, however, not compatible with the 3.3v TTL of the BeagleBone Black and a level shifter circuit is needed to interface it with the EVK-6T. The schematics, board layout and bill of materials can be found in Appendix D.1

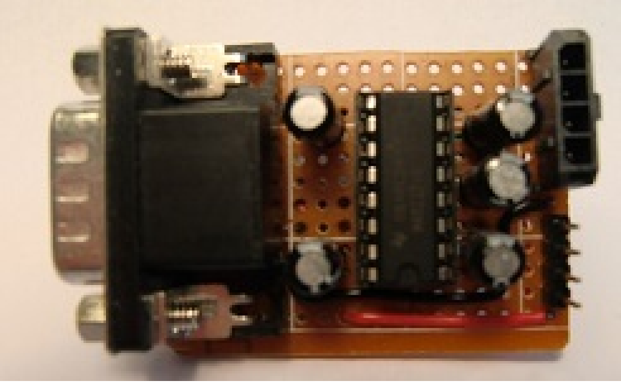
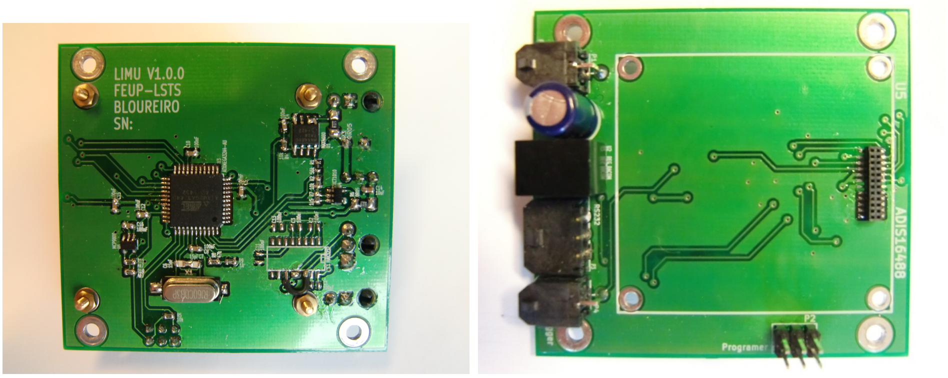


Figure 6.4: RS-232 Converter Circuit

6.2.3 LIMU Board

The ADIS16488 is usually mounted on a breakout board in order to access SPI(Serial Peripheral Interface). Furthermore an external power circuit is needed to power the IMU. This is not a viable solution due to the limited space within the x8 UAV and that the SPI ports of the Beaglebone are not accessible through the Beaglebone Cape. The LIMU(LSTS Ineartial Measurement Unit) Board provides a solution to both problems. In features a DC-DC stepdown converter and uses an Atmel ATXMEGA32A4-AU microcontroller, seen in Fig. 6.5, that takes care of the SPI communication and outputs the IMU data using 3.3v TTL serial. The 3.3V TTL serial output of the Atmel microcontroller is normally converted to RS-232, using a MAX323 chip, but this was removed in order to comply with the 3.3v TTL of the Beaglebone Black. The LIMU board also features an external trigger input for sensor synchronization, but this is not supported by the current firmware revision.



(a) LIMU Board Top Layer (b) LIMU Board Bottom Layer

Figure 6.5: The LSTS Ineartial Measurement Unit Board

6.2.4 Navigational Payload

A schematic of the payload is shown in Fig. 6.6. The entire payload is powered by the 15V, 10000 mAh UAV battery, however any input voltage between 6.5 - 18v may be applied. The LIMU board is connected one of the Beagelbone Black Cape power outputs, i.e. the same voltage level as the UAV battery. The ineartial measurement data from the ADIS16488CMLZ are transmitted to one of the serial ports of the Beaglebone Black. The Ethernet port of the BeagleBone Black is connected to a small network router, which in turn is connected to the Ubiquity M5 Radio. The Maxtena M1227HTC antenna is connected to the EVK-6T with a coaxial cable. The EVK-6T is usually powered through the USB-connector, but is here powered through the 6-pin header shown in Fig. 6.6. The RS-232 output is converted to 3.3v TTL by the converter circuit and transmitted to one of the serial ports of the Beaglebone Black.

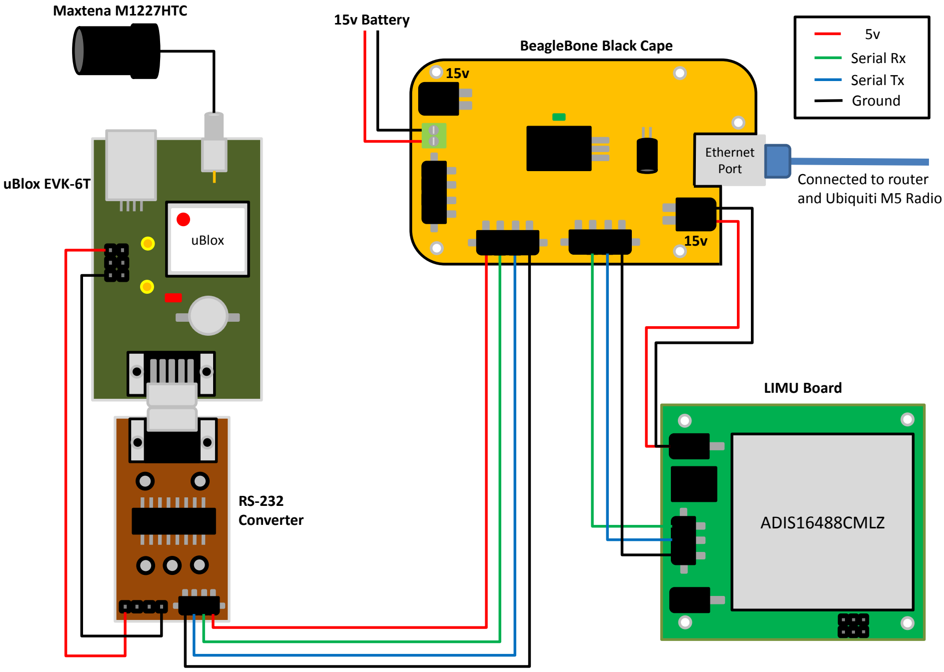


Figure 6.6: Payload Hardware Schematic

The payload mounted inside the fuselage of the x8 Skywalker is shown in Fig. 6.7, note that the wings are disassembled. The forward compartment contains the battery powering the payload and the engine. The middle compartment is reserved for experimental payloads. The ADIS16488 is mounted as close as possible to the center of mass of the UAV, approximately in the middle of the cross-bar dividing the middle and aft compartment. The aft compartment contains the Ubiquity M5 Radio, the Pixhawk Autopilot and a network router. The two dipole antennas mounted on both sides of the payload compartment are connected to the Ubiquity M5 Radio. The GNSS antennas are mounted to the lid of the payload compartment, with the Maxtena M1227HTC at the front and the 3DR uBlox GPS at the back. The small antenna close to the engine is the Spektrum Remote Receiver SPM9645 for radio telemetry.

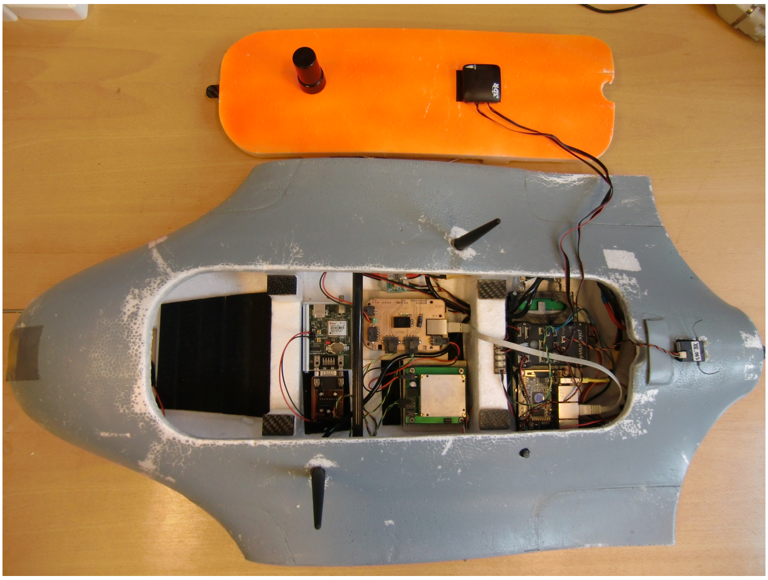


Figure 6.7: The navigational payload mounted in the fuselage of the X8 Skywalker

Chapter 7

Simulations

The platform and architecture independence of the LSTS-Toolchain makes it possible to run both Software-in-the-Loop (SIL) and Hardware-in-the-Loop (HIL) simulations to verify the implemented observer. SIL simulations have been used extensively by the author during the implementation of the observer. This chapter describes the simulator architecture for both SIL and HIL before presenting the results from HIL simulations only, since the SIL simulations performed are close to identical to those the HIL simulations.

7.1 Simulator Architecture

The Software-in-the-Loop setup can be seen in Fig 7.1 and uses the Ardupilot Software-in-the-Loop Architecture, shown in detail in Fig. 7.3. It is mainly used by developers for testing of Ardupilot software, but has been interface with DUNE by LSTS. The HIL architectures differs from the SIL architecture with DUNE being cross-compiled and run on the target system instead of the simulation computer, illustrated in Fig. 7.2.

The flight dynamics model used is JSBSim, an open source, multi-platform, object-oriented flight dynamics model (FDM) framework written in the C++ programming language (JSBSim, 2015). JSBSim is designed to support simulation modeling of any aerospace craft without the need for specific compiled and linked program code, instead relying specification written in an XML format, formally known as JSBSim-ML (JSBSim Markup Language), in which the aircraft mass and geometric properties are specified. The flight dynamics model in JSBSim continuously outputs flight dynamics data, i.e position, acceleration, angular velocity and attitude in the SITL_FDM struct over UDP.

Arduplane Desktop Executable uses the data to emulate sensors by adding noise

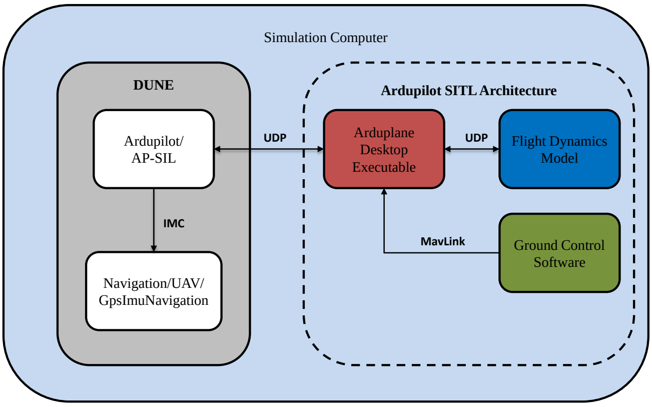


Figure 7.1: Software-in-the-Loop architecture

and bias to the data and dispatches Mavlink messages containing the emulated sensor data. The Mavlink messages are used by the Ardupilot EKF to estimate position, velocity and attitude. Based on this the Autopilot emulates PWM outputs to the servos and engine used by the flight dynamics model. The AP-SIL task in DUNE connects to one of the Serial-over-TCP ports of the Arduplane Desktop Executable and acts as an interface between Mavlink and IMC message protocols.

The simulator parameters like wind, bias, noise and sensor rates are configured through MavProxy, a Ground Control Software package for Mavlink based systems. However, it should be noted that the sensor rate from the Ardupilot Desktop Executeable is limited to 50 Hz.

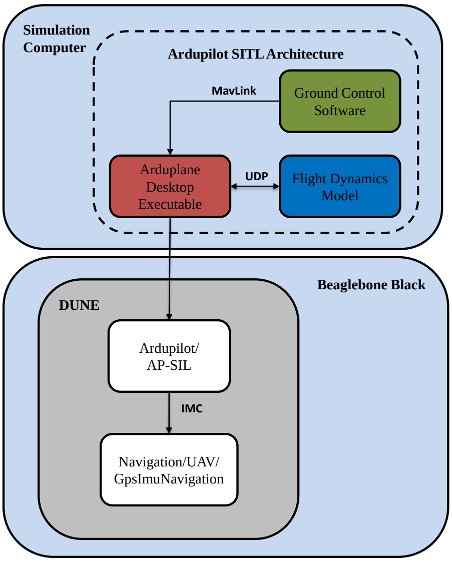


Figure 7.2: Hardware-in-the-Loop architecture

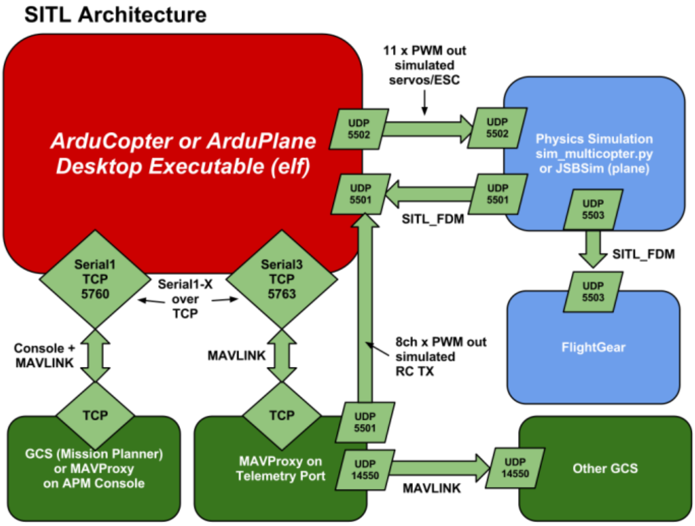

SITL Architecture
ArduCopter or ArduPlane Desktop Executable (elf)
11 x PWM out simulated servos/ESC
UDP 5502
UDP 5502
Physics Simulation sim_multicopter.py or JSBSim (plane)
UDP 5501
UDP 5501
SITL_FDM
UDP 5503
SITL_FDM
UDP 5503
FlightGear
Serial1 TCP 5760
Serial1-X over TCP
Serial3 TCP 5763
Console + MAVLINK
MAVLINK
8ch x PWM out simulated RC TX
TCP
GCS (Mission Planner) or MAVProxy on APM Console
TCP
MAVProxy on Telemetry Port
UDP 5501
UDP 14550
MAVLINK
UDP 14550
Other GCS

Figure 7.3: Ardupilot Software-in-the-Loop architecture

7.2 Hardware-in-the-Loop Simulations

The HIL architecture introduces a delay similar to that of GNSS receiver, i.e the actual position of the aircraft will lie ahead of the measured position, due to the conversion from the Mavlink Protocol to IMC in the AP-SIL Task. The delay makes it possible to test if the fast simulator is able to accurately predict the position and velocity of the UAV.

The following observer parameters were used during the simulation: $M_b = 0.5, k_1 = 0.05, k_2 = 0.05, \theta = 1, K_{pp} = 2.0I_{3x3}, K_{vp} = 1.0I_{3x3}$ and $K_{\xi p} = 0.5I_{3x3}$. The time-delay τ is assumed to be constant, i.e. $\tau = 0.240$. The emulated IMU and magnetometer measurements are provided at a rate of 50 Hz and GPS measurements at 5 Hz. The simulator parameters used are given in Tab. 7.1. Noise is added to accelerometer, gyroscope and magnetometer measurements. The bias of the accelerometer and the magnetic interference from the engine is set to zero.

Table 7.1: Simulator Parameters

Parameter	**Description**	**Value**	**Unit**
SIM_ACC_BIAS_X	Accelerometer x-axis Bias	0	
SIM_ACC_BIAS_Y	Accelerometer y-axis Bias	0	
SIM_ACC_BIAS_Z	Accelerometer z-axis Bias	0	
SIM_ACCEL_FAIL	Accelerometer Failure	0	
SIM_ACC_RND	Accelerometer Noise	2	
SIM_GYR_RND	Gyroscope Noise	5	
SIM_GPS_DISABLE	Disable GPS	0	
SIM_GPS_BYTELOSS	GPS Byteloss Percentage	0	
SIM_GPS_NUMSATS	Number of visible Satellites	10	
SIM_GPS_DRIFTALT	GPS Altitude Drift	0	
SIM_GPS_GLITCH	GPS Glitch Offsets	0	
SIM_GPS_HZ	GPS Update rate	5	
SIM_GPS_DELAY	GPS Delay in Samples	0	
SIM_MAG_RND	Magnetometer Noise	5	
SIM_MAG_MOT_X	Magnetometer X-axis Interference	0	
SIM_MAG_MOT_Y	Magnetometer Y-axis Interference	0	
SIM_MAG_MOT_Z	Magnetometer Z-axis Interference	0	
SIM_MAG_ERROR	Magnetometer Failure	0	
SIM_MAG_DELAY	Data Delay Magnetometer	0	
SIM_WIND_SPD	Wind Speed	0	
SIM_WIND_DIR	Wind Direction	180	
SIM_WIND_TURB	Wind Turbulance	0.2f	
SIM_WIND_DELAY	Data Delay Wind	0	

7.2.1 Results

This section presents the results from the HIL simulations using the tuning and simulator parameters listed above. The results are briefly commented in this section and are further discussed in Ch. 9.

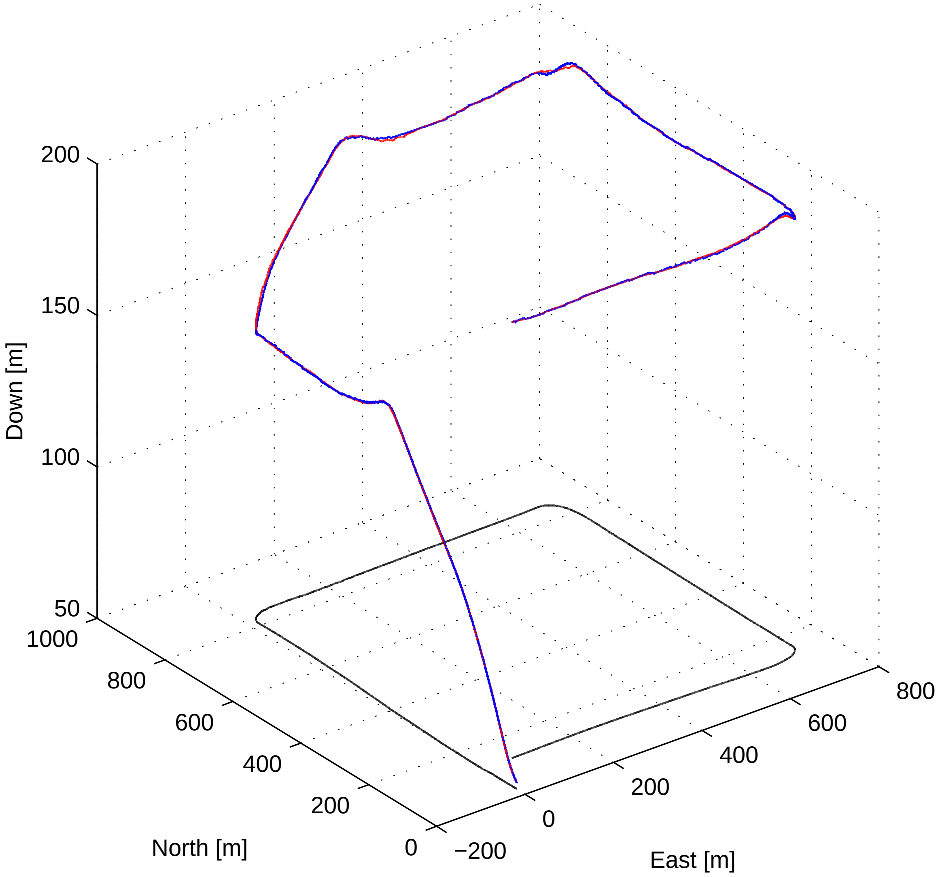


Figure 7.4: 3D-Plot of the flight path, predicted estimate $p(t|t-\tau)$ in (red, solid), true position $p(t)$ (blue, solid) and ground path (black, solid)

The 3-D flight path of the UAV is shown in Fig. 7.4. The UAV takes off from (0,0) flying Northward with a rapid ascent to 150 m before turning east and climbing further to a cruise altitude of 200m.

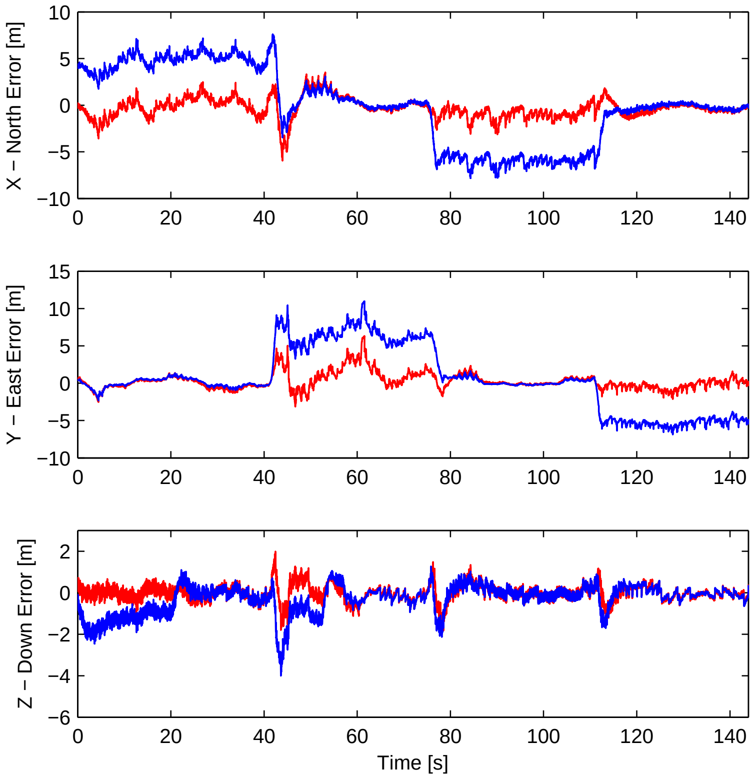


Figure 7.5: Position error in NED coordinates, predicted estimate $p(t|t-\tau)$ (red,solid) and delayed estimate (blue, solid)

Fig. 7.5 shows the position error in NED coordinates. The predicted estimate has a considerable smaller error than the delayed estimate and the position error is generally larger during take-off and ascension, interval from 0 to 75 seconds, than in level flight.

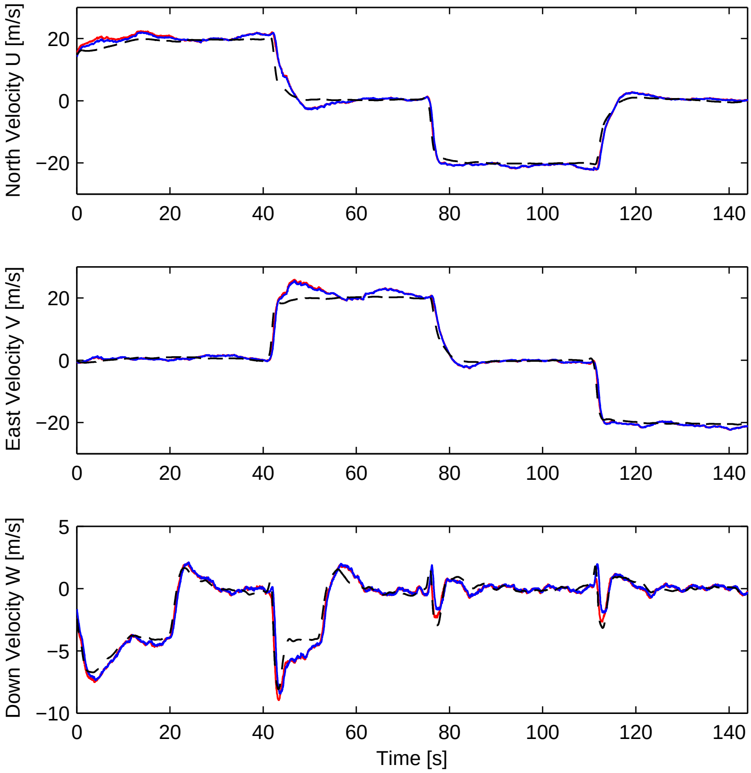


Figure 7.6: Velocity in NED coordinates, true velocity (black, dashed), predicted estimate (red, solid) and delayed estimate (blue, solid)

Fig. 7.6 shows the velocity in NED coordinates. The autopilot airspeed is set to $20m/s$ between the waypoints. The spikes in Down velocity coincide with the UAV climbing after turning, seen in Fig. 7.4.

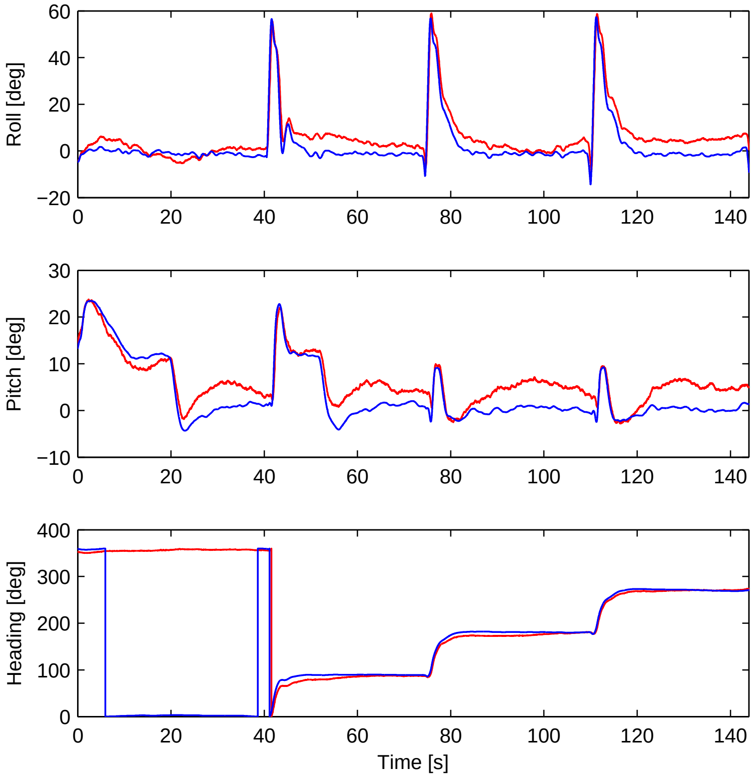


Figure 7.7: Attitude in Euler angles, true(blue) and estimated (red)

Fig. 7.7 shows the attitude presented in Euler angles. The large deviation in heading is due to the estimate wrapping from 360° to 0° considerably later than the reference.

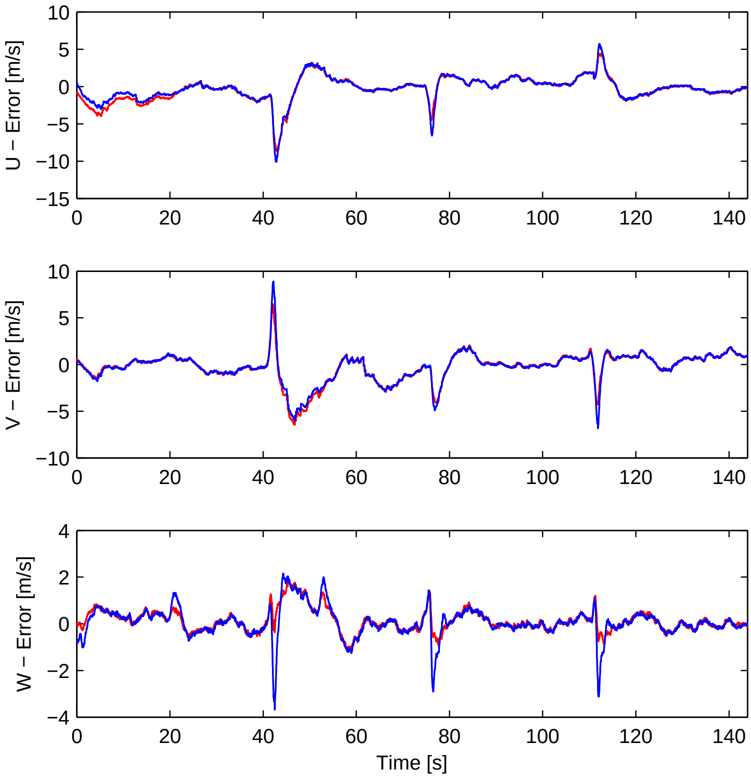


Figure 7.8: Velocity Error, predicted velocity (red) and delayed estimate(blue)

The velocity error relative to the true velocity of the UAV is shown in Fig. 7.8. The periodic spikes coincide with the UAV turingn sharply and the errors are mostly within the region of $\pm 2m/s$ between the turns.

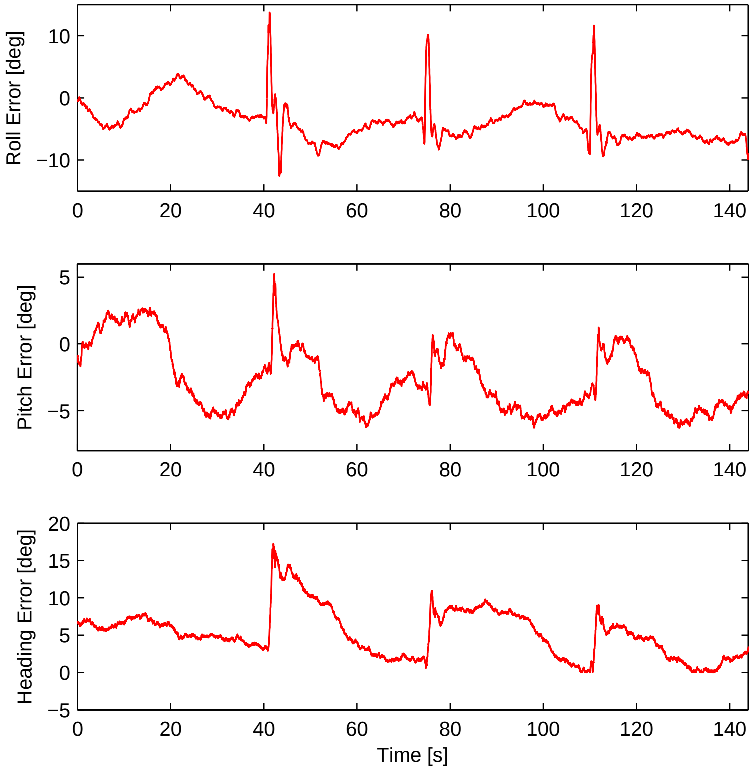


Figure 7.9: Attitude Error in Euler angles

Fig. 7.9 show the attitude error represented in Euler angles. The roll error are within region of ±10° and pitch errors are within the region of ±5° . The large error in heading is due to the estimate switching from 180 to -180

Chapter 8

Experimental

This chapter describes the setup of the experiments conducted in this thesis and presents the results.

8.1 Experimental Time-Delay Determination

The experimental setup for determining the delay can be seen in Figure 8.1. The ublox EVK-6T receivers has been configured to output raw measurement data, pseudoranges and carrier-phase measurements, this procedure is described in Appendix A.

The RTKLIB is used to process the data from the two GPS receivers. The RTKLIB function str2str streams the raw measurements from the base receiver to the rover over a TCP connection. Raw measurement data from both uBlox EVK-6T receivers is processed by the RTKLIB function rtkrcv and used to determine the position of the rover by resolving the integer ambiguity. The RTK-GPS position solution is parsed and timestamped by the RTKGPS Task in Dune. The rising edge of the Pulse-Per-Second signal is detected and timestamped by the PPS Task. Both sensor tasks dispatch IMC messages to the IMC message bus and the Logging task subscribes to these messages and logs them for later analysis.

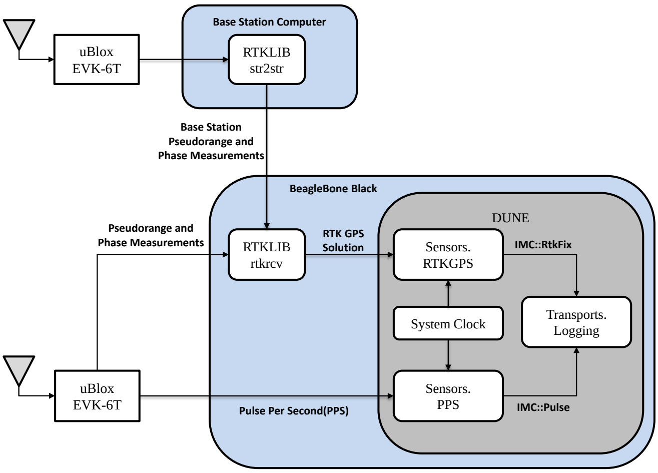


Figure 8.1: Experimental setup and signal flow for determining the time-delay of the RTK-system.

8.1.1 Results

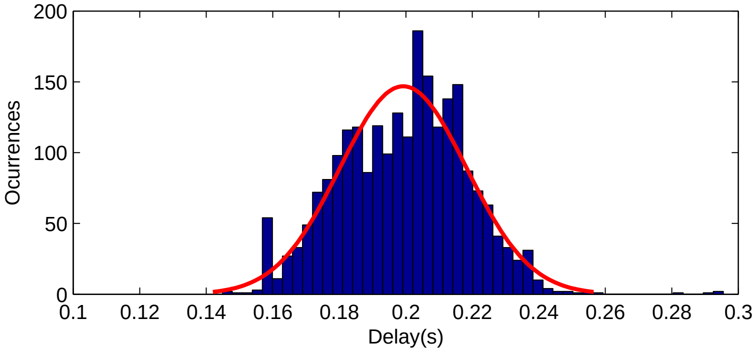


Figure 8.2: Histogram showing the distribution of the time-delay measurements over the course of 45 minutes with 1 Hz measurements using the u-Blox EVK-6T

Fig. 8.2 shows the distribution of the observed time-delays and their occurrences. A Gaussian Distribution has been fitted to the observed data and superimposed on the data. The distribution has a mean $\mu = 199.2\,ms$ and variance $\sigma^2 = 0.36379\,ms$.

8.2 In-Flight Testing of the RTK GPS System

The observer is dependent on regular updates from the RTK GPS system which only has been tested in a static configuration. Test flights were conducted at Agdenes Airfield on the 21th May in order to evaluate the performance of the standalone RTK GPS system in flight. The experimental setup for the flight is shown in Fig. 8.3.

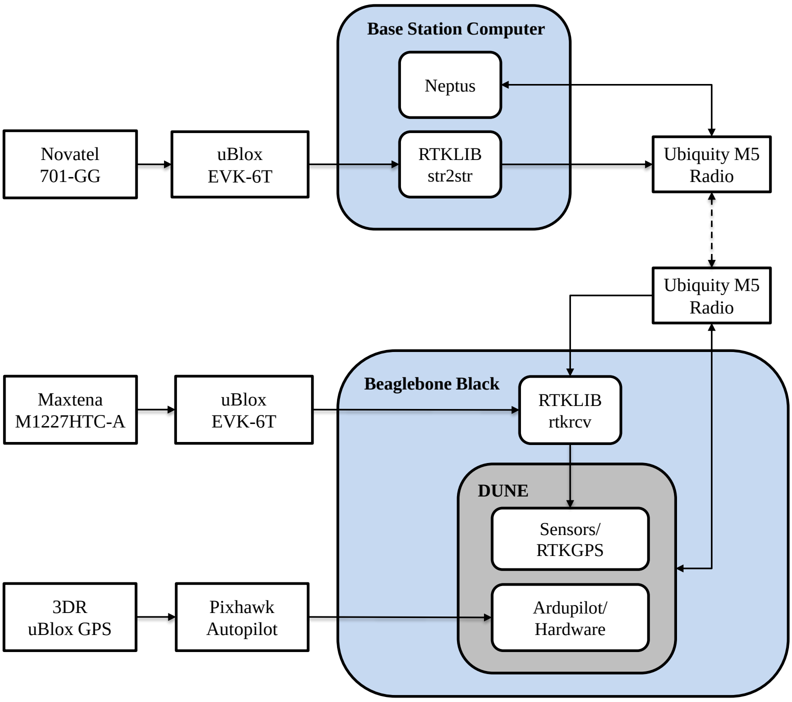


Figure 8.3: The experimental setup used for the in-flight testing of the RTK GPS system

The base station and the UAV use the Novatel 701-GG antenna and the Maxtena M1227HTC-A Helix antenna together with the uBlox EVK-6T GPS receiver respectively. The raw data from the base station is forwarded to the UAV by the RTKLIB AP str2str through a pair of Ubiquity M5 Radios. The raw data from both base station and UAV is processed the RTKLIB AP rtkrcv. The estimated position is handled by the sensor task RTKGPS with in Dune, i.e parsed and logged. The Ardupilot/Hardware task handles sensor and data input from the Pixhawk autopilot, used by the pilot for aileron and speed control in manual mode.

The Pixhawk runs the Ardupilot EKF, using rate gyroscopes, accelerometer, magnetometer, GPS, airspeed and barometric pressure measurements to estimate the position, velocity and angular orientation of the flight vehicle. The estimated position logged by the Ardupilot/Hardware Task is used as a reference for evaluating the RTK-GPS system. All messages sent on the IMC Message Bus within DUNE are broadcasted to the base station through the radio-link and shown to the operator in Neptus.

8.2.1 Results

The 3-D flight path is shown in Fig. 8.5 and the projection of the flight path in the North-East plane is shown in Fig. 8.4. The offset between the EKF estimate and the RTK-GPS position is due to the two systems being initialized with different origins relative to the local frame. An outage in the RTK GPS measurements can be seen shortly after take-off. Fig. 8.6 shows the RTK GPS measurements, where the measurements with fixed ambiguities are highlighted. The percentage of measurements with fixed ambiguities was 34.72 percent and the percentage of measurements with floating ambiguities was 65.28.

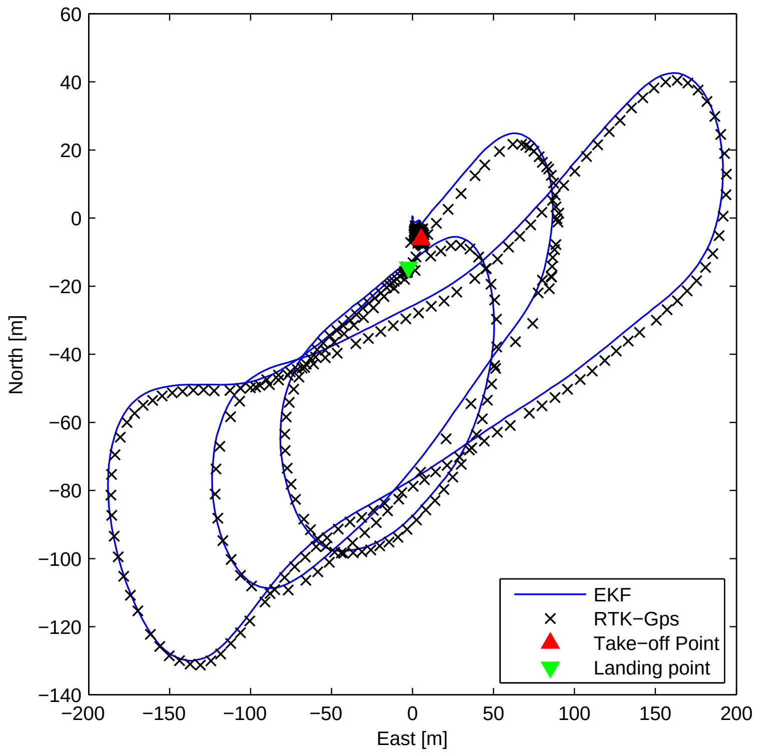


Figure 8.4: Projection of the flight path in the North-East plane

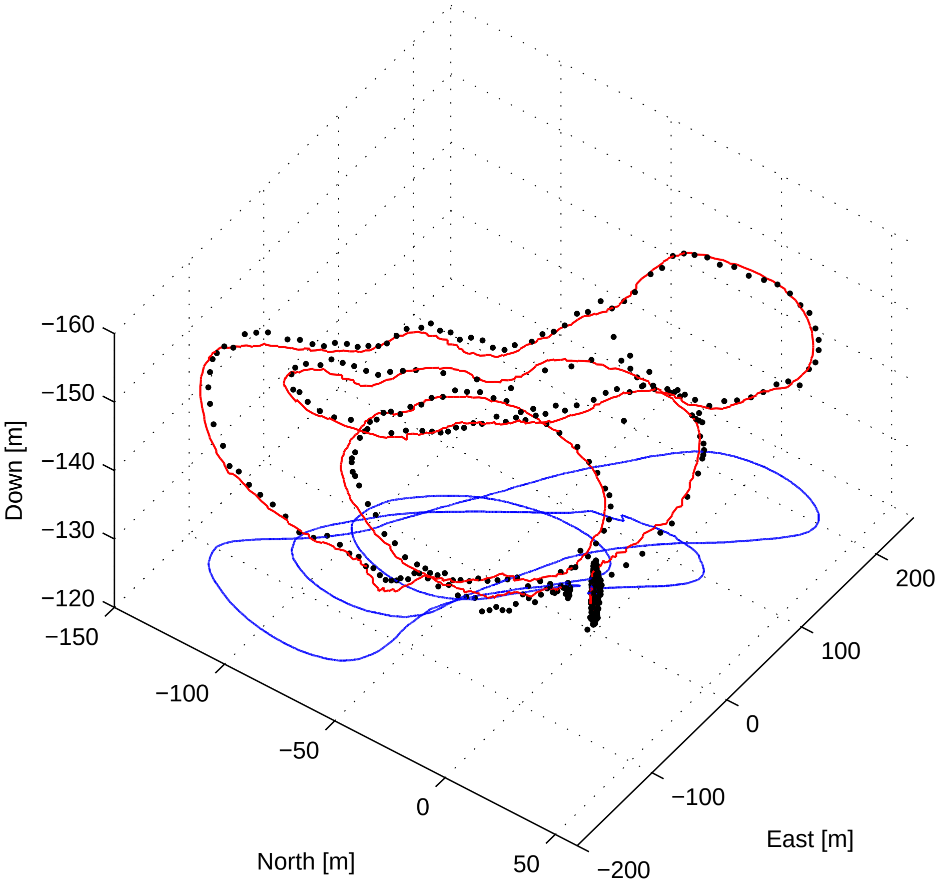


Figure 8.5: 3-D flight path, EKF position estimate (red, solid), RTK GPS measurements (black, dots) and ground path (blue, solid)

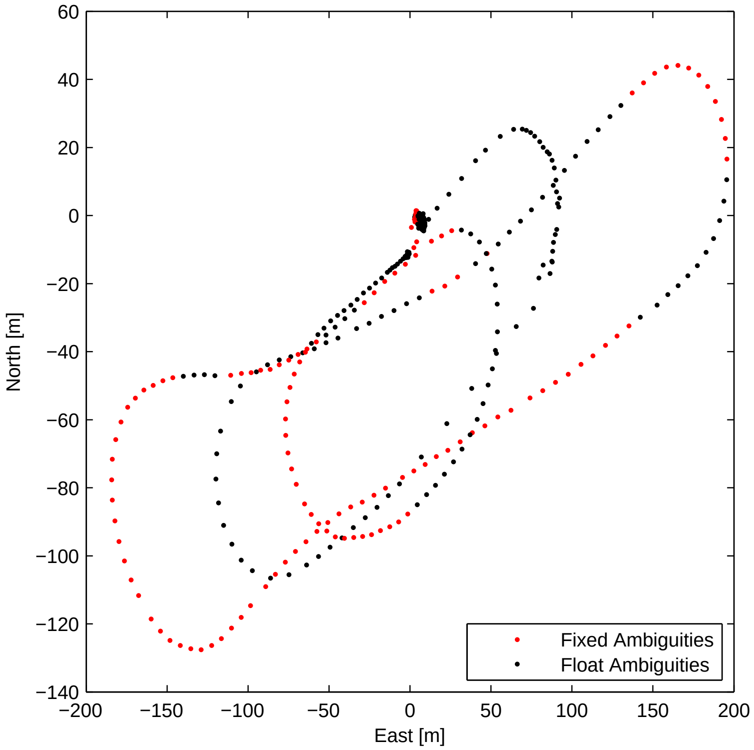


Figure 8.6: 2D projection of the flight path in the North-East plane highlighting the RTK GPS Measurements with fixed ambiguities

8.3 Experimental Testing of the Navigation System

Experimental flights were performed at Agdenes Airfield on the 28th of May 2015. The goal of the flights was to test the performance of the navigation system. The experimental setup is almost identical to the setup used to test the RTK GPS system, with the addition of the ADIS16488 IMU, the LIMU Task and the observer task GpsImuNavigation, shown in Fig. 8.7.

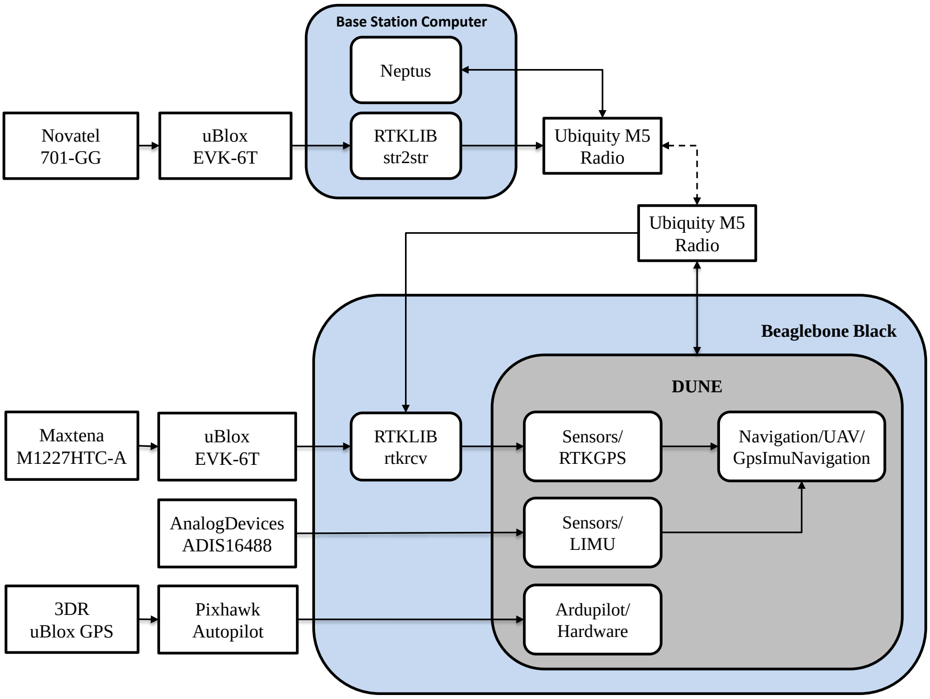


Figure 8.7: Experimental Setup

The following observer parameters were used during the simulation: $M_b = 0.5, k_1 = 0.05, k_2 = 0.05, \theta = 1, K_{pp} = 2.0I_{3x3}, K_{vp} = 1.0I_{3x3}$ and $K_{\xi p} = 0.5I_{3x3}$. The time-delay is assumed constant, i.e $\tau = 0.200$. The IMU and magnetometer measurements are provided at a rate of 50 Hz and GPS measurements at 5 Hz. The accelerometer bias and initial gyro bias is found by averaging measurements from the ADIS16488 while the UAV is static on the ground with known attitude. The UAV is flown manually by the pilot, launched using a catapult and belly landed on the grass runway.

8.3.1 Results

This section presents the results from one of the total of three flight that were performed at Agdenes Airfield. The results are briefly commented in this section and further discussed in Ch. 9.

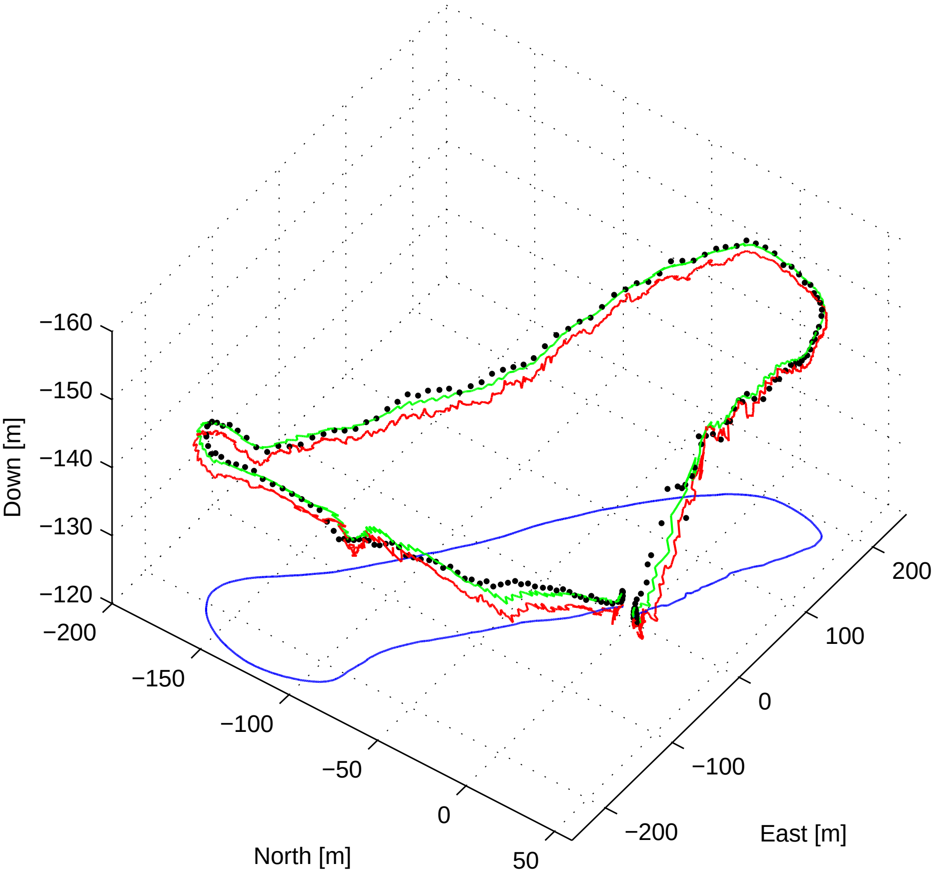


Figure 8.8: 3D plot of flight path, predicted position (red, solid), delayed estimate (green, solid), RTK GPS measurements (black, dots) and ground path (blue, solid)

The 3-D path of the UAV is shown in Fig. 8.8. The delayed estimate $p(t-\tau)$ is in agreement with the RTK GPS measurements, while the predicted position $p(t|t-\tau)$ is offset from the RTK GPS measurements on the vertical axis.

The projection of the flight path in the North-East plane is shown in Fig. 8.9, with UAV flying clockwise along the path. The predicted position $p(t|t-\tau)$ and the delayed estimates follows the RTK GPS measurements except during take-off and landing where the estimates diverge.

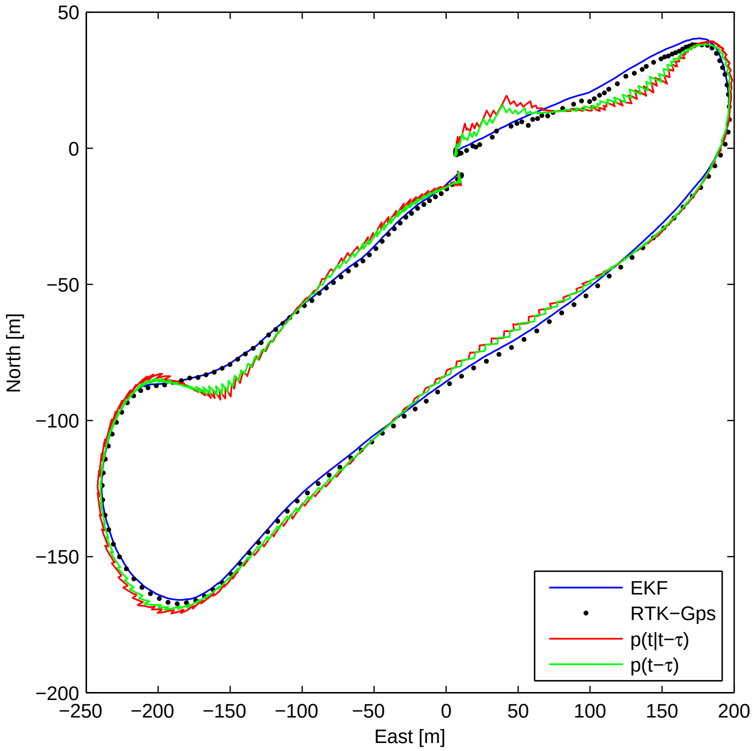


Figure 8.9: Projection of the flight path, estimates and RTK GPS measurements in the North-East plane

Fig. 8.10 shows the RTK GPS measurements during the flight. The measurements where the integer ambiguity has been resolved are highlighted in red and the measurements where the ambiguity is unresolved, denoted a float ambiguity, are shown in blue. The percentage of measurements with fixed ambiguties was 3.88 percent and the percentage of measurements with float ambiguities was 96.12 percent.

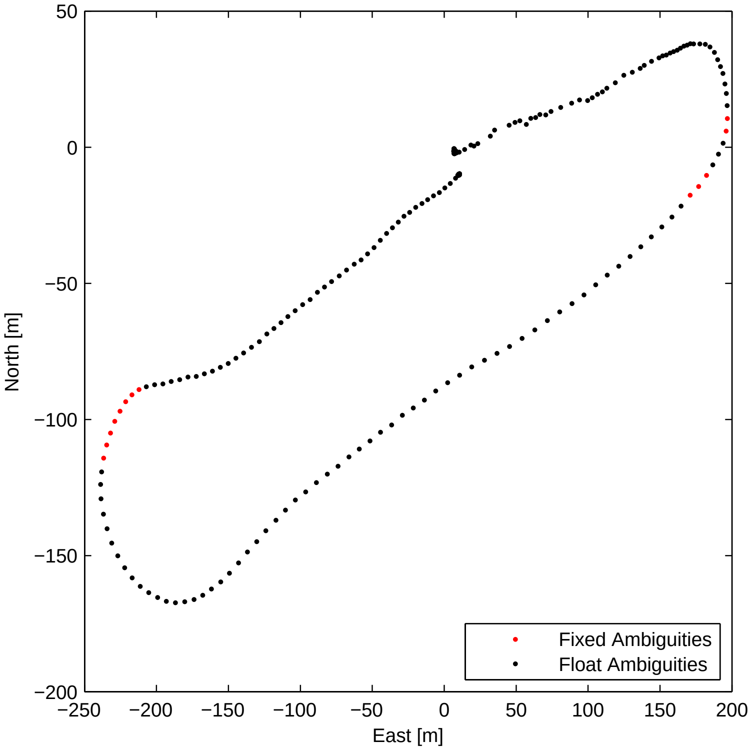


Figure 8.10: RTK GPS measurements, measurements with fixed ambiguities are highlighted in red

The position error in NED coordinates relative to the RTK GPS measurements is shown in Fig. 8.11. In order to compare the estimates and the RTK GPS measurements the measurements have been interpolated using linear interpolation. Take-off occurs around 15 seconds and approach for landing is initiated at 50-55 seconds. It should be noted that the predicted estimate has a greater error than the delayed estimate. Both errors are large during take-off and landing, but during the interval 25-55 seconds the north errors are within the region $\pm 6m$ and the east errors are within -5 to $15m$. The delayed down estimate is within the region $\pm 2m$ and while the predicted estimate is within the region 0 to $-4m$.

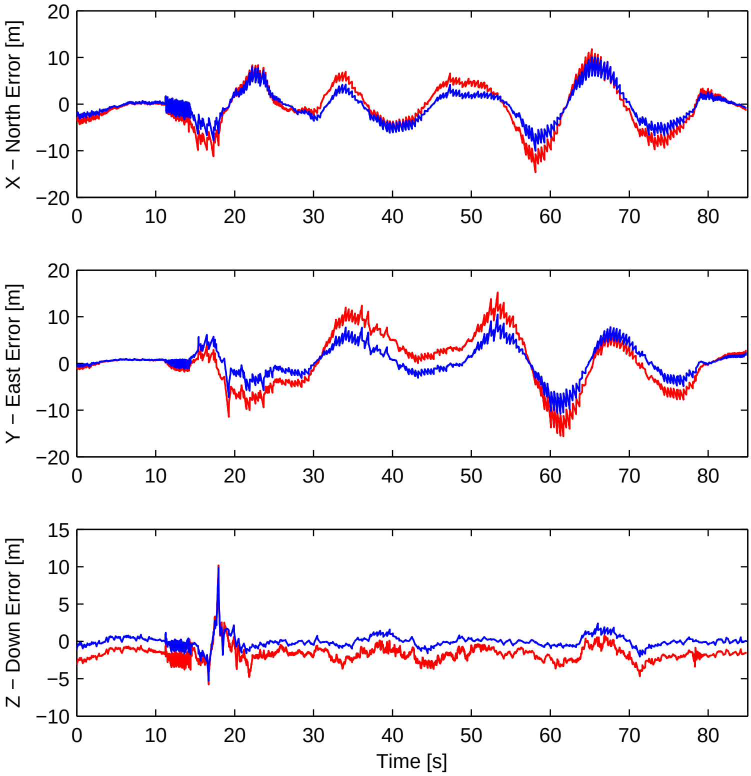


Figure 8.11: Position error in NED coordinates relative to the RTK GPS measurements, predicted estimate (red, solid) and delayed estimate (blue,solid)

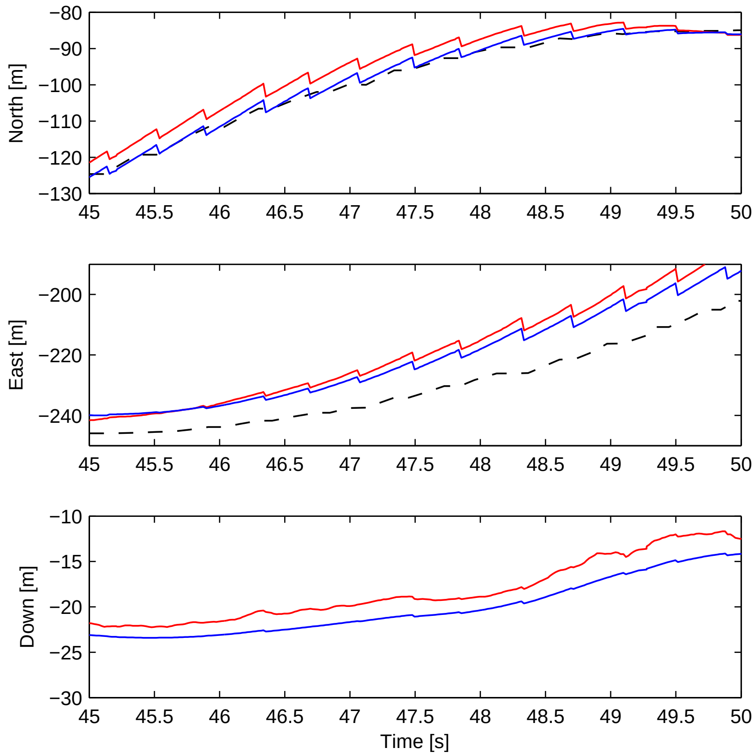


Figure 8.12: An enlarged section of the position in NED coordinates, predicted position (red, solid), delayed estimate (blue, solid) and EKF reference (black, dashed). The fast simulator predicts the current position based on the delayed estimates.

An enlarged section of the position in NED coordinates is shown in Fig. 8.12. The fast simulator predicts the current position, shown in red, based on the delayed estimate, shown in blue. The EKF reference is not present in z-axis plot as there is a systematic offset between the estimates and the EKF height estimate.

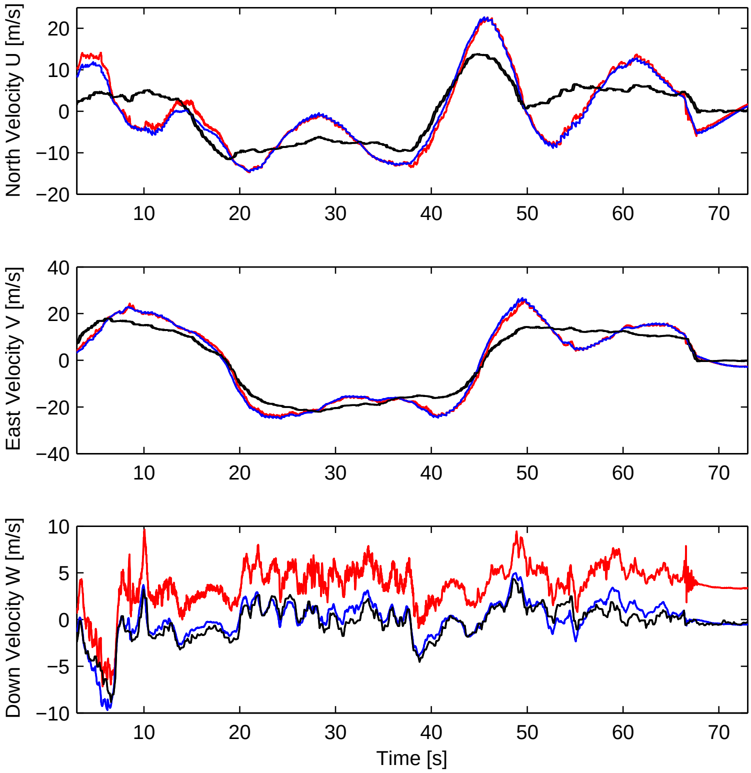


Figure 8.13: Velocity, EKF estimate (black solid), predicted velocity (red, solid) and delayed estimate (blue, solid)

The velocity during the flight is shown in Fig. 8.13. The north velocity estimate exhibits oscillatory behavior. The east velocity estimate is in agreement with the reference, but exhibits slow response. The delayed estimate of the down velocity is in agreement to the reference, while the predicted estimate appears to be offset from the delayed estimate.

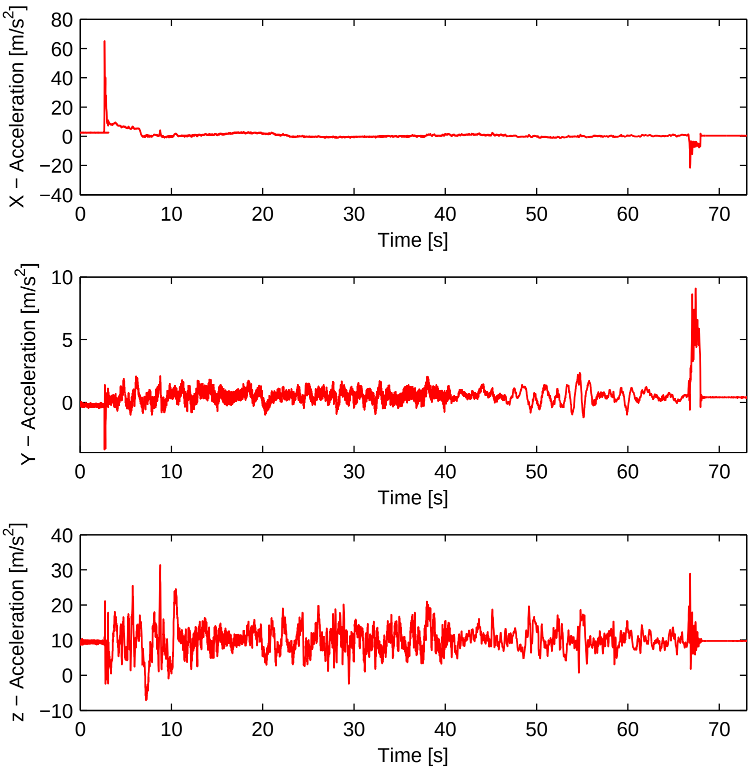


Figure 8.14: Accelerometer Measurements

Fig. 8.14 shows the accelerometer measurements during the flight. A large spike in the x-axis measurement is observed during take-off and a large spike in the y-axis measurements is observed during approach for landing.

The attitude represented by Euler angles is shown in Fig. 8.15. All estimates appear to be offset from reference estimates with intervals where the estimates are in agreement with the reference. The heading estimate appears to converge towards the reference midflight before diverging at the end of the flight.

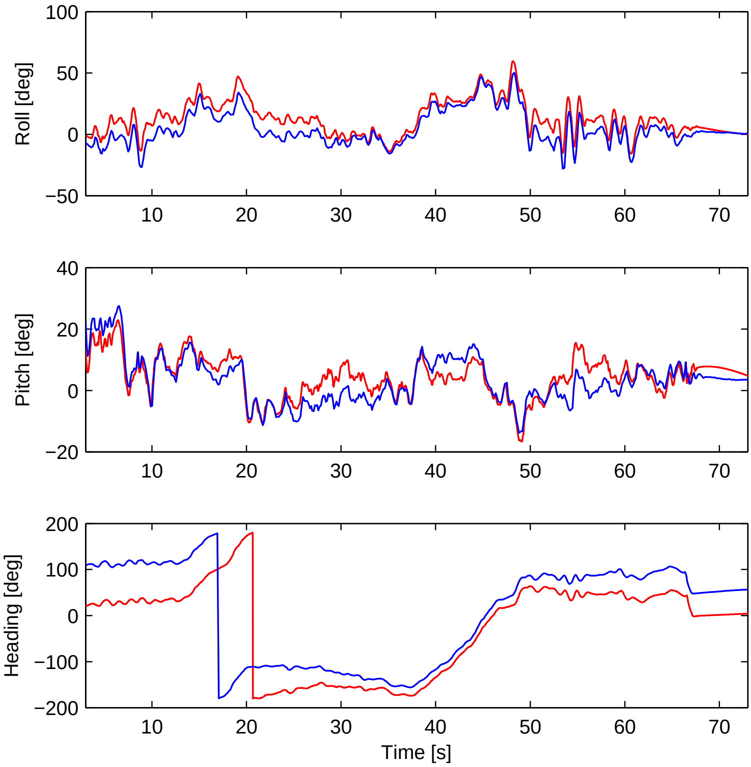


Figure 8.15: Attitude during flight represented in Euler Angles, EKF reference (blue, solid) and predicted estimate (red, solid)

8.4 Lab Testing of the INS

Due to the large errors in the results from the flights at Agdenes additional tests at the UAV-Lab were conducted to evaluate the performance of the inertial navigation system using the parameters from the experimental flights. The test setup is seen in Fig. 8.16 with the ADIS16488 mounted on the LIMU Board and powered by the Beaglebone via its extension board. The inertial data is logged by the payload computer and retrieved for analysis.

The objective of the tests was to

- determine drift in attitude and position estimates with the IMU static
- verify convergence of the gyro bias estimates
- evaluate the response of the attitude observer to principle rotations about the z and x-axis

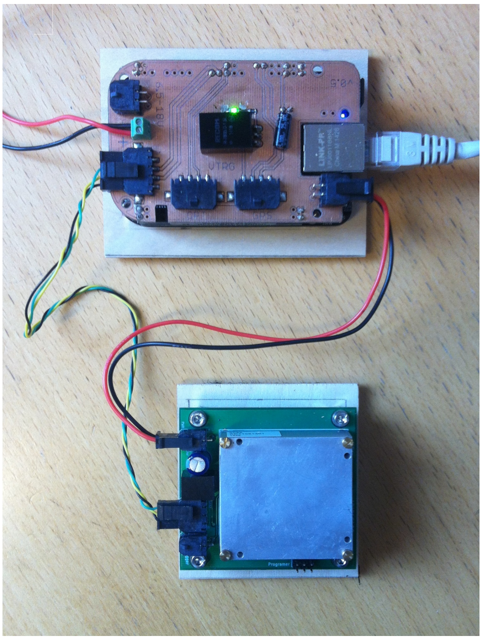


Figure 8.16: Experimental setup used for INS testing

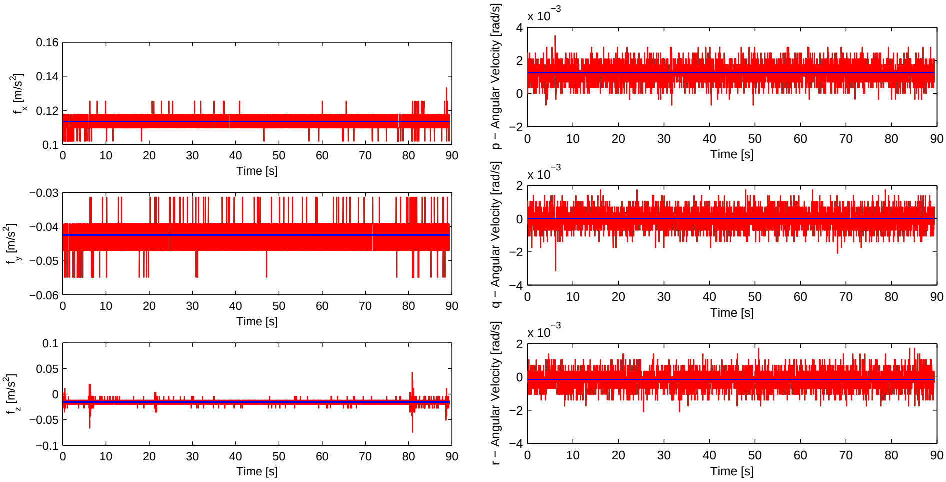
8.4.1 Results

The results are briefly commented in this section and further discussed in Ch. 9.

Measurements of the specific forces and angular velocities witht the IMU at rest are shown in Fig. 8.17a and Fig. 8.17b respectively. The constant bias is found by averaging the measurements and is shown in blue.

The drift in the position ,velocity and attitude estimates is shown in Fig. 8.18a, 8.19a and Fig. 8.18b respectively. The position estimates drift considerably and after 10 seconds the RMS errors are $6.8112m$, $3.6558m$ and $0.1871m$ for the North,East and Down position respectively. The roll angle drifts at about 1 deg/second and the pitch angle drifts at about 0.5 degrees/second. The heading angle oscillates, but appears to be converging towards 2 degrees. The estimated gyro bias is shown in Fig. 8.19b.

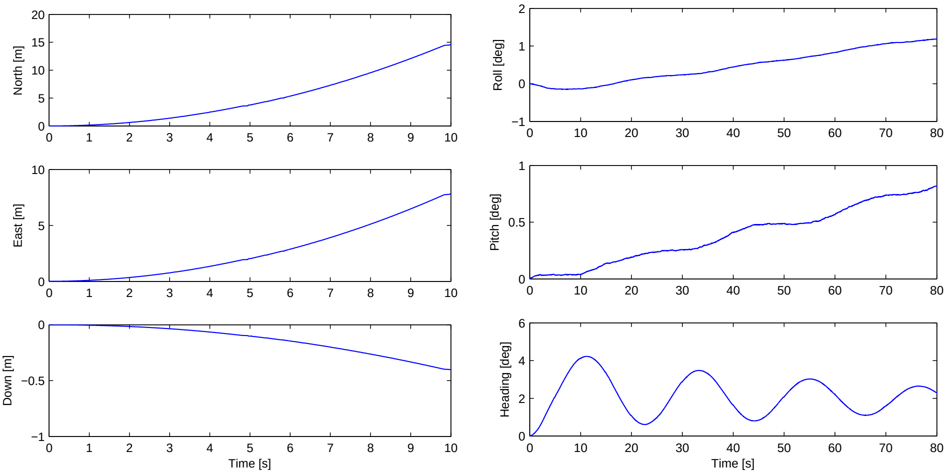
The response of the attitude observer with the IMU being turned 90 degrees about the z-axis with an initial heading angle of zero is shown in Fig. 8.20a. The heading angle is 90 degrees, but drifts of to 85 degrees. The response of the attitude observer with the IMU being turned 90 degrees about the x-axis with an initial roll angle of zero is shown in Fig. 8.20b. The roll estimate quickly converges to 90 degrees in approximately 5 seconds and drifts of to 94 degrees towards the end of the test.



(a) Specific force with the IMU static (red,solid) with gravity subtracted on the z-axis and mean acceleration(blue,solid)

(b) Measured angular velocity (red, solid) and mean angular velocity(blue, solid)

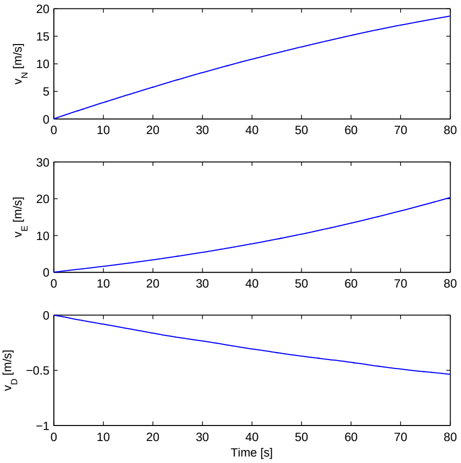
Figure 8.17: Bias Measurements



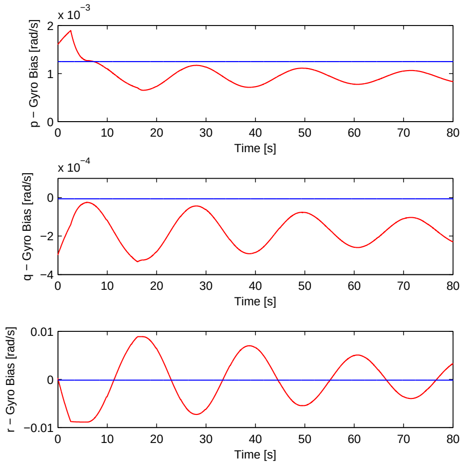
(a) Position drift with the IMU static

(b) Attitude drift with the IMU static

Figure 8.18: Estimate drift with the IMU at rest

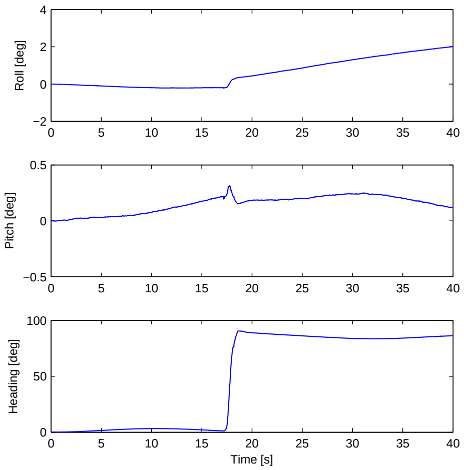


(a) Velocity Drift

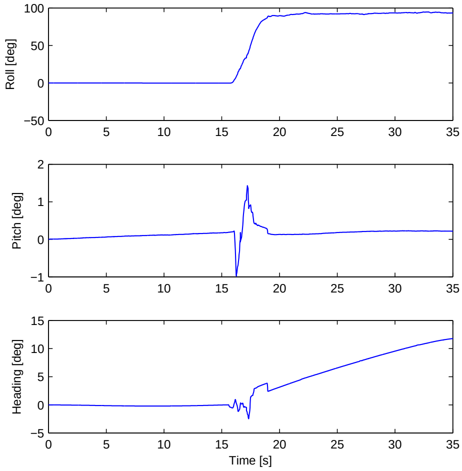


(b) Estimated gyro bias while the IMU is static, estimated bias (red, solid) and mean gyro bias(blue,solid)

Figure 8.19: Velocity drift and estimated gyro bias



(a) Response of the attitude observer - 90 degree turn about the z-axis



(b) Response of the attitude observer - 90 degree turn about the x-axis

Figure 8.20: Attitude observer response

Chapter 9

Discussion

This chapter discusses the results obtained from simulations and experiments from Ch. 7 and 8.

9.1 RTK GPS Time-Delay Determination

The results found in Sec. 8.1 is in agreement with the results found in (Hansen et al., 2015), where a time-delay with a Gaussian distribution mean value of $154.5ms$ was observed using a u-Blox EVK-6T. With the addition of additional processing of the data from the EVK-6T, i.e. RTKLIB, it was expected that the time-delay would increase. Thus the result of a measured time-delay with mean $199.2ms$ is very reasonable.

It should be noted that the observed time-delay will vary with receiver type and the internal Kalman filter parameters. The satellite constellation, i.e geometry, elevation angles and number of visible satellites may also affect the observed time-delay.

9.2 HIL Simulations

The HIL simulations show that the implemented observer is able to run on the embedded system. Furthermore the results show a considerable increase in accuracy when employing the fast simulator using delayed simulated GPS measurements. The predicted estimate has a considerably smaller error than the delayed estimate, seen in Fig. 7.5, verifying that the fast simulator accurately predicts the current position. However, it is assumed that the time-delay is constant, when it in fact is varying, leading to an error of 1-3 m in the predicted estimates.

The velocity estimates are for the most part in agreement with the reference velocities, seen in Fig. 7.6. Some deviations are seen during the climb after take-off, but the estimates quickly converge to the true velocity as the UAV reaches the set altitude.

The attitude estimates appears to be drifting in both roll and pitch in intervals with little change in attitude. This suggests that there is bias in the angular velocity measurements that is not taken into account. This was later confirmed by testing the INS, see Sec. 8.4 and 9.5. The heading angle converges to the true value between turns and is not experiencing the same drift as pitch and roll as the heading is derived from the magnetometer measurements and not gyro measurements.

There are several limitations with the simulator setup that may have affected the results of the simulations. Firstly the measurement rate of the emulated sensor measurements is limited to 50 Hz, due to an internal limitation in the Ardupilot Firmware. Secondly there are no variables governing the gyro bias in the simulations, see Tab. 7.1.

9.3 In-Flight Testing of the RTK GPS System

The results from the in-flight testing of the RTK GPS system, confirm that the RTK GPS system, comprising of the EVK-6T and RTKLIB has been successfully integrated with DUNE. The percentage of measurements with fixed ambiguities was 34.72 percent and the percentage of measurements with floating ambiguities was 65.28 during the flight. Similar results were obtained by (Skulstad et al., 2015) using the same system, i.e. two EVK-6T receivers and RTKLIB, but with different antennas.

9.4 Experimental Flights

In Fig. 8.11 there appears to be a constant offset in height between the delayed and predicted estimate. The offset occurred due to a sign error in the fast simulator implementation leading to gravity being added and not subtracted from the accelerometer measurements. This error propagated into the velocity estimate, seen in Fig. 8.13, which in turn leads to error in the position estimate, seen in Fig. 8.11 and 8.8.

The large errors in the position estimates during take-off and approach for landing,seen in Fig 8.9, were caused turbulence, clearly seen by looking at the accelerometer measurements in Fig 8.14. Disregarding the take-off and approach for landing there are still considerable position errors, seen in Fig 8.11, with magnitude up to 10 m. However, it should be noted that the predicted error will be larger than the delayed estimate since it is compared to the RTK GPS position containing the time-delay error. Examining Fig. 8.9 it is evident that the position error term should be weighted higher in the correction step of the observer.

The large initial error in the attitude estimates , seen in 8.15, was due to the attitude observer being wrongly initialized. This especially evident in the heading estimate, where the initial error is close to a 100 degrees. The initial offsets in roll and pith are 6 and 15 degrees respectively. The errors in roll and pith during the flight is due to the gyro bias not being correctly estimated, see Sec. 8.4 and Sec. 9.5.

The heading estimate appears to be lining up with the EKF reference, but diverge towards the end of the flight. This is most likely due to the magnetometer measurements containing the magnetic disturbances from the engine and other payload components. An additional error source could be the time-varying reference vector m^n used by the attitude observer. The vector is given by a model of the earths magnetic field, but may not be more accurate than a fixed reference vector based on averaging magnetometer measurements with the UAV at rest pointing towards true north.

The low percentage of fixed RTK GPS measurements, seen in Fig 8.10, is due to a change in the settings of RTKLIB, specifically lowering the number of iterations used per measurement by the ambiguity resolution algorithm. This led to more unresolved ambiguities, however, the accuracy float solutions are at the decimeter level and should not affect the estimates considerably.

9.5 Lab Testing of the INS

Examining Fig. 8.19b it is evident that the attitude observer does not estimate the gyro bias correctly. The estimated gyro bias converges very slowly and not towards the constant bias. This leads to drift in the attitude estimates, seen in Fig 8.18b, 8.20a and 8.20b. Examining Fig. 8.18a it is noted that the drift in position is considerable and outages in the RTK GPS measurements lasting longer than 2.5 seconds will lead to errors at the meter level.

Unfortunately, this was not discovered until after the experimental flights. In retrospect it is clear that the flight tests were rushed and additional systematic testing of gyro bias estimate convergence and drift in the INS estimates should have been performed prior to experimental testing.

Chapter 10

Conclusion and Future Work

This chapter presents the conclusions drawn from the results and suggests areas for future work in order to improve the navigation system.

10.1 Conclusion

The main objective of this thesis has been the development of an integrated RTK GPS/INS navigation system for UAV applications that compensates for the time-delay introduced by receiver processing and achieving sub-meter accuracy.

Much of the work with this thesis has been devoted to the design and construction of the payload for the x8 Skywalker, with the following objectives being reached:

- a fully functional sensor platform with the ADIS16488CMLZ IMU and the EVK-6T GPS receiver using the Beaglebone Black as payload computer.
- The RTK GPS system, comprising of the EVK-6T and RTKLIB has been successfully integrated with DUNE through the task RTKGPS which interprets the position data.
- the time-delay of the RTK GPS system has been determined to be $199.2ms$ through experimental testing.
- The time-delayed nonlinear observer structure of (Hansen et al., 2015) has been implemented in DUNE Task GpsImuNavigation

The navigation system has been tested through HIL simulations and experimental flights with x8 Skywalker using the developed payload. The HIL simulations show that observer compensates for large parts of the position error introduced by the time-delay, but the accuracy of the position estimate is between 1-3 meters.The results from the experimental flights show that the position estimates from the navigation system have errors of up to 10 meters. In both cases the performance goal of sub-meter accuracy is not achieved.

There are several reasons to why the desired sub-meter accuracy was not achieved. Firstly, the observer parameters used for the experimental flights were found through HIL simulations with optimal conditions. Thus the performance seen in the experimental flights can largely be attributed to poor tuning of the observer. It is clear that additional testing under real-life conditions is necessary to improve the performance of the navigation system. Secondly, the attitude observer was not sufficiently tested prior to the experimental flights resulting in sub-optimal performance. Lastly it has been assumed that the time-delay is constant,but it may vary significantly. Thus, the full potential of the navigation system has not been exploited.

Recommendations for future work that may improve the performance of the system is given in the following section.

10.2 Future Work

The payload and implementation of the nonlinear observer presented in this thesis can be developed further in a number of ways to improve the accuracy of the navigation system.

First and foremost rigorous testing of the INS should be performed to correct the errors that were revealed from the tests in Sec. 8.4. This should be followed by additional flight test to verify that the errors have been removed and to tune the observer under real-life conditions.

Correcting for the mean value of the time-delay considerably improves the position estimates, however, the best approach would be to correct for the individual time-delays. This could be done by either estimating the time-delays or measuring the actual delay real-time using the approach described in Sec. 3.7.

The firmware on the LIMU board limits the measurement rate from the ADIS16488 to 100 Hz. New firmware should be developed in collaboration with LSTS in order to achieve a higher measurement rate.

A better simulator setup should be found due to the emulated sensor rate limitation of the Ardupilot Software-in-the-Loop architecture. One solution is that the flight dynamics data from JSBsim could be sent directly to a simulator task in DUNE where the sensor data is emulated.

The current payload only uses GPS satellites, but other satellite constellations could easily be included since RTKLIB supports GLONASS GALILEO and BEIDOU. In order to utilize the satellites of other GNSS constellations the EVK-6T receiver would have to be replaced. This should significantly increase the number of visible satellites, which in turn should improve the conditions for the integer ambiguity search algorithm resulting in a higher number of fixed ambiguities.

Bibliography

Analog Devices (2015). Datasheet ADIS16488A . http://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16488A.pdf, Accessed: 07-06-2015.

Batista, P., Silvestre, C., and Oliveira, P. (2011a). GES attitude observers-Part I: Multiple general vector observations. In *18th IFAC World Congress*, pages 2985–2990.

Batista, P., Silvestre, C., and Oliveira, P. (2011b). GES attitude observers-part II: Single vector observations. In *IFAC World Congr*, pages 2991–2996.

BeagleBoard (2015). BeagleBone Black. http://beagleboard.org/black, Accessed: 07-06-2015.

Blewitt, G. (1989). Carrier Phase Ambiguity Resolution for the Global Positioning System Applied to Geodetic Baselines up to 2000 km. *Journal of Geophysical Research*, 94:10.187 –10.203.

Euler, H. J. and Landau, H. (1992). Fast GPS ambiguity resolution on-the-fly for real-time applications. In *Sixth International Geodetic Symposium on Satellite Positioning*.

Farrel, J. A., Givargis, T. D., and Barth, M. J. (2000). Real-Time Differential Carrier Phase GPS-Aided INS. *IEEE Transactions on Control Systems Technology*.

Farrell, J. (2008). *Aided navigation: GPS with high rate sensors*. McGraw-Hill.

Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley.

Grip, H., Fossen, T., Johansen, T., and Saberi, A. (2015). Globally exponentially stable attitude and gyro bias estimation with application to GNSS/INS integration. *Automatica*, 51.

Grip, H. F., Fossen, T. I., Johansen, T. A., and Saberi, A. (2011a). Attitude estimation based on time-varying reference vectors with biased gyro and vectro measurements. In *IFAC World Congr.*, pages 8497–8502.

Grip, H. F., Fossen, T. I., Johansen, T. A., and Saberi, A. (2011b). Nonlinear Observer for GNSS-Aided Inertial Navigation with Quaternion-Based Attitude Estimation. *American Control Conference (ACC).*

Grip, H. F., Fossen, T. I., Johansen, T. A., and Saberi, A. (2012). A Nonlinear Observer for Integration of GNSS and IMU Measurements with Gyro Bias Estimation . *American Control Conference.*

Grip, H. F., Johansen, T. A., and Saberi, A. (2011c). Observers for Cascaded Nonlinear and Linear Systems. *IEEE Conference on Decision and Control and European Control Conference(CDC-ECC).*

Hansen, J. M., Fossen, T. I., and Johansen, T. A. (2015). Nonlinear Observer for INS Aided by Time-Delayed GNSS Measurements - Implementation and UAV Experiments. In *International Conference on Unmanned Aircraft Systems (ICUAS'15).*

Hatch, R. (1990). Instantaneous Ambiguity Resolution. In *Kinematic Systems in Geodesy, Surveying and Remote Sensing*, pages 299 –308.

HobbyKing (2015). Skywalker X-8 . http://www.hobbyking.com/hobbyking/store/__27132__Skywalker_X_8_FPV_UAV_Flying_Wing_2120mm.html, Accessed: 07-06-2015.

Hofmann-Wellenhof, B., Lichtenegger, H., and Wasle, E. (2008). *GNSS - Global Navigation Satellite Systems - GPS, GLONASS, Galileo and more.* SpringerWienNewYork.

Hua, M. D. (2010). Attitude estimation for accelerated vehicles using GPS/INS measurements. *Control Engineering Practice 18.7*, pages 723–732.

JSBSim (2015). JSBSim Flight Dynamics Model - Open Source Flight Dynamics and Control Software Library in C++ . http://jsbsim.sourceforge.net/JSBSim/index.html, Accessed: 14-05-2015.

Kerney, C. (2015). Geographic library. http://sourceforge.net/projects/geographiclib/, Accessed: 14-05-2015.

Khosravian, A., Trumpf, J., Mahony, R., and Hamel, T. (2014). Velocity aided attitude estimation on SO (3) with sensor delay. *IEEE Conf. Decision and Control.*

Khosravian, A., Trumpf, J., Mahony, R., and Hamel, T. (2015). Recursive Attitude Estimation in the Presence of Multi-rate and Multi-delay Vector Measurements. In *American Control Conference.*

Mahoney, R., T.Hamel, and J.-M.Pflilin (2008). Nonlinear complementary filters on the Special Orthogonal Group. *IEEE Trans. Automat. Contr.*, 53(5):1203–1218.

Maxtena (2015). M1227HCT-A-SMA Datasheet . http://www.farnell.com/datasheets/1681376.pdf, Accessed: 07-06-2015.

Maybeck, P. S. (1982). *Stochastic models, estimation, and control.* Academic Press.

Novatel (2015). Datasheet GPS-701-GG . http://www.novatel.com/assets/Documents/Papers/GPS701_702GG.pdf, Accessed: 07-06-2015.

Phillips, R. E. and Schmidt, G. T. (2008). GPS/INS integration. *AGARD Lecture Series: System Implications and Innovative Applications of Satellite Navigation*, 207.

Pinto, J., Calado, P., Braga, J., Dias, P. S., Martins, R., Marques, E., and Sousa, J. B. (2012). Implementation of a control architecture for networked vehicle systems. *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV 2012).*

Pinto, J., Dias, P. S., Martins, R., Fortuna, J., Marques, E., and Sousa, J. B. (2013). The LSTS Toolchain for Networked Vehicle Systems. *MTS/IEEE Conf. OCEANS.*

Rieke, M., Foerster, T., Geipel, J., and Prinz, T. (2011). High-precision positioning and real-time data processing of UAV-systems. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science 38.*

Salcudean, S. (1991). A globally convergent angular velocity observer for rigid body motion. In *Automatic Control, IEEE Transactions*, pages 1493–1497.

Scherzinger, B. M. (2000). Precise robust positioning with inertial/GPS RTK. *Proceedings of ION-GPS.*

Skog, I. and Händel, P. (2011). Time synchronization errors in loosely coupled GPS-aided inertial navigation systems. *IEEE Transactions on Intelligent Transportation Systems.*

Skulstad, R. and Syversen, C. L. (2014). Low-cost instrumentation system for recovery of fixed-wing UAV in a net. Master's thesis, Norwegian University of Science and Technology.

Skulstad, R., Syversen, C. L., Merz, M., Sokolova, N., Fossen, T. I., and Johansen, T. A. (2015). Net Recovery of UAV with Single-Frequency RTK GPS. In *IEEE Aerospace Conference.*

Solomon, P. D., Wang, J., and Rizos, C. (2011). Latency Determination and Compensation in Real-Time GNSS/INS Integrated Navigation Systems. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 3822.*

Stempfhuber, W. and Buchholz, M. (2011). A precise, low-cost RTK GNSS system for UAV applications. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science 38.*

Takasu, T. and Yasuda, A. (2009). Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. *International Symposium on GPS/GNSS*, pages 4–6.

Teunissen, P. J. (1994). A new method for fast carrier phase ambiguity estimation. In *IEEE Position Location and Navigation Symposium*, pages 562–573.

Teunissen, P. J. and Verhagen, S. (2004). On the foundation of the popular ratio test for GNSS ambiguity resolution. In *ION GNSS*, pages 2529–2540.

Teunissen, P. J. G., Jonge, P. J., and Tiberius, C. C. J. M. (1997). Performance of the LAMBDA method for fast GPS ambiguity resolution. *Navigation 44.3*, pages 373–383.

T.Hamel and Mahoney, R. (2006). Attitude estimation on SO(3)-based on direct inertial measurements. In *IEEE Int. Conf. Robotics Automation*, pages 2170–2175.

Ubiquiti Networks (2015). Ubiquity Rocket M airMax Basestation . https://www.ubnt.com/airmax/rocketm/, Accessed: 07-06-2015.

uBlox AG (2015a). EVK-6 u-blox Evaluation Kits User Guide. https://www.u-blox.com/images/downloads/Product_Docs/EVK-6_UserGuide_%28GPS.G6-EK-10040%29.pdf, Accessed: 07-06-2015.

uBlox AG (2015b). LEA-6 DataSheet. http://www.u-blox.com/images/downloads/Product_Docs/LEA-6_DataSheet_%28GPS.G6-HW-09004%29.pdf, Accessed: 07-06-2015.

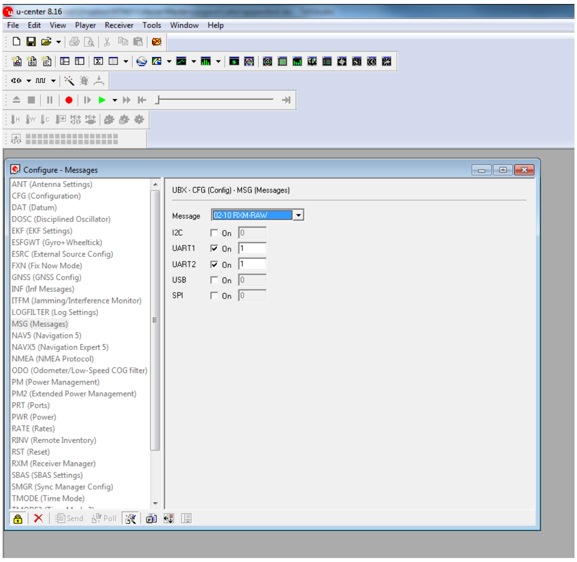
Vik, B. and Fossen, T. (2001). A nonlinear observer for GPS and INS integration. In *IEEE Conf. Decision and Control*, pages 2956–2691.

Yunchun, Y. and Farrell, J. (2003). Magnetometer and differential carrier phase GPS-aided INS for advanced vehicle control. *IEEE Transactions on Robotics and Automation*, 19.

Appendix A

EVK-6T Configuration

The EVK-6T receivers do not output raw data in their default configuration and must therefore be configured to enable this feature. This is done by connecting the receiver to uCenter, the configuration software tool for uBlox GNSS receivers. In order to output raw data, i.e. phase and pseudorange measurements, the messages RXM-RAW and RXM-SFRB were enabled, see Fig. A.1 and A.2. A complete description of the RXM-RAW message is foung on page 185 of (uBlox AG, 2015b).

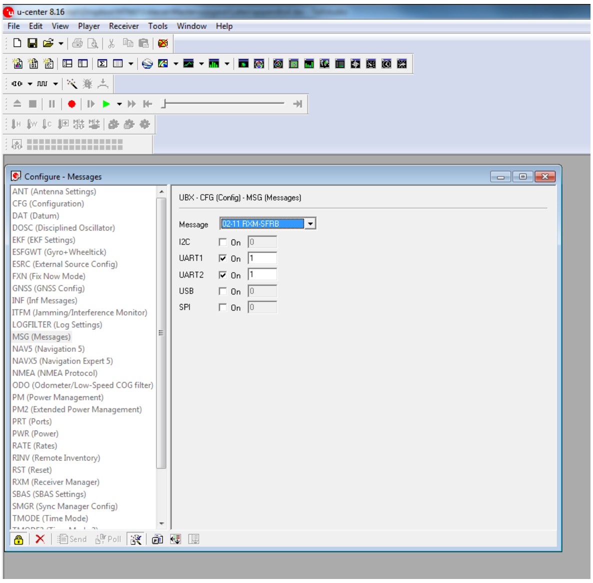


UBX - CFG (Config) - MSG (Messages)

Message 02-10 RXM-RAW

I2C	□ On	0
UART1	☑ On	1
UART2	☑ On	1
USB	□ On	0
SPI	□ On	0

Figure A.1: Screenshot from u-center showing how to enable the RXM-RAW message

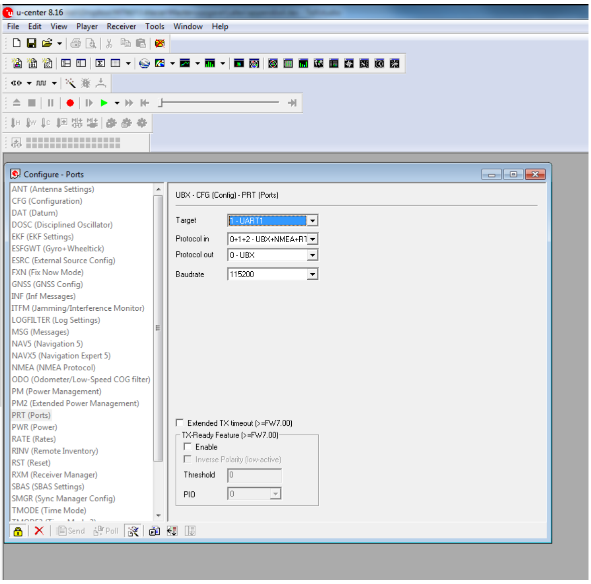


UBX - CFG (Config) - MSG (Messages)

Message 02-11 RXM-SFRB

I2C	□ On	0
UART1	☑ On	1
UART2	☑ On	1
USB	□ On	0
SPI	□ On	0

Figure A.2: Screenshot from u-center showing how to enable the RXM-SFRB message



UBX - CFG (Config) - PRT (Ports)

Target: 1 - UART1
Protocol in: 0+1+2 - UBX+NMEA+R1
Protocol out: 0 - UBX
Baudrate: 115200

□ Extended TX timeout (>=FW7.00)

TX-Ready Feature (>=FW7.00)
□ Enable
□ Inverse Polarity (low-active)
Threshold: 0
PIO: 0

Figure A.3: Screenshot from u-center showing how to select the output port and protocol

In addition to enabling raw measurement data output, the output port must be set to output the data using the UBX protocol, shown in Fig. A.3. Furthermore, the receivers should communicate with the base station and rover computers using the serial instead of USB interface, which is the default option. This was done by altering the CFG-PRT message. A complete description of the CFG-PRT message is found on page 129 of (uBlox AG, 2015a).

Appendix B

RTKLIB Configuration File

This appendix contains the RTKLIB configuration file used in this thesis, shown in Listing B.1

```
console-passwd      =admin
console-timetype    =gpst
console-soltype     =dms
console-solflag     =1
inpstr1-type        =serial
inpstr2-type        =tcpcli
inpstr3-type        =off
inpstr1-path        =uart/2:115200:8:n:1:off
inpstr2-path         =192.168.1.35:50021
inpstr3-path      =
inpstr1-format      =ubx
inpstr2-format      =ubx
inpstr3-format      =sp3
inpstr2-nmeareq     =off
inpstr2-nmealat     =0
inpstr2-nmealon     =0
outstr1-type        =serial
outstr2-type        =file
outstr1-path        =pts/2:115200:8:n:1:off
outstr2-path        =rtklib_output%Y%m%d%M.pos
outstr1-format      =llh
outstr2-format      =xyz
logstr1-type        =file
logstr2-type        =off
logstr3-type        =off
logstr1-path        =Agdenes%Y%m%d%h%M.log
logstr2-path        =ref_%Y%m%d%h%M.log
logstr3-path        =cor_%Y%m%d%h%M.log
misc-svrcycle        =50
misc-timeout         =30000
misc-reconnect       =30000
misc-nmeacycle       =5000
```

```
misc-buffsize      =32768
misc-navmsgsel     =all
misc-startcmd      =../rtkstart.sh
misc-stopcmd       =../rtkshut.sh
file-cmdfile1      =../data/ubx_raw_5hz.cmd
file-cmdfile2      =../data/ubx_raw_5hz.cmd
#file-cmdfile3      =
pos1-posmode       =movingbase
pos1-frequency     =l1
pos1-soltype       =forward
pos1-elmask        =10
pos1-snrmask       =off
pos1-dynamics      =off
pos1-tidecorr      =off
pos1-ionoopt       =brdc
pos1-tropopt       =saas
pos1-sateph        =brdc
pos1-exclsats      =
pos1-navsys        =1
pos2-armode        =fix-and-hold
pos2-gloarmode     =off
pos2-arthres       =3
pos2-arlockcnt     =0
pos2-arelmask      =0          # (deg)
pos2-aroutcnt      =5
pos2-arminfix      =5
pos2-slipthres     =0.05       # (m)
pos2-maxage        =30         # (s)
pos2-rejionno      =30         # (m)
pos2-niter         =5
pos2-baselen       =0        # (m)
pos2-basesig       =0          # (m)
out-solformat      =llh        # (0:llh,1:xyz,2:enu,3:nmea)
out-outhead        =on         # (0:off,1:on)
out-outopt         =on         # (0:off,1:on)
out-timesys        =gpst       # (0:gpst,1:utc,2:jst)
out-timeform       =hms        # (0:tow,1:hms)
out-timendec       =3
out-degform        =deg        # (0:deg,1:dms)
out-fieldsep       =
out-height         =ellipsoidal # (0:ellipsoidal,1:geodetic)
out-geoid          =internal   # (0:internal,1:egm96,2:egm08_2.5,3:egm08_1,4:gsi2000)
out-solstatic      =all        # (0:all,1:single)
out-nmeaintv1      =0          # (s)
out-nmeaintv2      =0          # (s)
out-outstat        =off        # (0:off,1:state,2:residual)
stats-errratio     =100
stats-errphase     =0.003      # (m)
stats-errphaseel   =0.003      # (m)
stats-errphasebl   =0          # (m/10km)
stats-errdoppler   =1          # (Hz)
stats-stdbias      =30         # (m)
stats-stdiono      =0.03       # (m)
stats-stdtrop      =0.3        # (m)
stats-prnaccelh    =1          # (m/s^2)
```

```
stats-prnaccelv     =0.1          # (m/s^2)
stats-prnbias       =0.0001       # (m)
stats-prniono       =0.001        # (m)
stats-prntrop       =0.0001       # (m)
stats-clkstab       =5e-12        # (s/s)
ant1-postype        =llh          # (0:llh,1:xyz,2:single,3:posfile,4:
    rinexhead,5:rtcm)
ant1-pos1           =0            # (deg|m)
ant1-pos2           =0            # (deg|m)
ant1-pos3           =0            # (m|m)
ant1-anttype        =
ant1-antdele        =0            # (m)
ant1-antdeln        =0            # (m)
ant1-antdelu        =0            # (m)
ant2-postype        =llh          # (0:llh,1:xyz,2:single,3:posfile,4:
    rinexhead,5:rtcm)
ant2-pos1           =0            # (deg|m)
ant2-pos2           =0            # (deg|m)
ant2-pos3           =0            # (m|m)
ant2-anttype        =
ant2-antdele        =0            # (m)
ant2-antdeln        =0            # (m)
ant2-antdelu        =0            # (m)
misc-timeinterp     =off          # (0:off,1:on)
misc-sbasatsel      =0            # (0:all)
file-satantfile     =
file-rcvantfile     =
file-staposfile     =
file-geoidfile      =
file-dcbfile        =
file-tempdir        =temp
file-geexefile      =
file-solstatfile    =
file-tracefile      =
```

Listing B.1: RTKLIB configuration file

Appendix C

Build Rules for RTKLIB in Glued

RTKLIB is not a native package within Glued and must therefore be built and compiled for the target system. The build rule for rtkrcv is shown in Listing C.1 and the build rule for Listing C.2. The build rules are placed in the rules folder within the Glued directory.

C.1 rtkrcv

```
version=\
(
  "2.4.2"
)

url=\
(
  #"http://www.rtklib.com/prog/rtklib_$version.zip"
)

md5=\
(
  #f39e76cf9eaa07e6974279a4a0065cab
)

build_dir="rtklib_$version"
build_always=false

build()
{
  cd app/rtkrcv/gcc
  $cmd_make \
  CC=$cmd_target_cc \
  CTARGET="$cfg_target_gcc_flags -DMINPRNQZS_S=0 -DMAXPRNQZS_S=0" \
  BINDIR="$cfg_dir_rootfs/bin"
}

target_install()
{
  cd app/rtkrcv/gcc
  $cmd_make \
  CC=$cmd_target_cc \
  CTARGET="$cfg_target_gcc_flags -DMINPRNQZS_S=0 -DMAXPRNQZS_S=0" \
  BINDIR="$cfg_dir_rootfs/bin" \
  install

  # copy configuration file
  cp -Rp "$pkg_dir/rtklib" "$cfg_dir_rootfs/opt/"
}
```

Listing C.1: Build Rule for rtkrcv

C.2 str2str

```
version=\
(
  "2.4.2"
)

url=\
(
  "http://www.rtklib.com/prog/rtklib_$version.zip"
)

md5=\
(
  f39e76cf9eaa07e6974279a4a0065cab
)

build_dir="rtklib_$version"
build_always=false

build()
{
  cd app/str2str/gcc
  $cmd_make \
  CC=$cmd_target_cc \
  CTARGET="$cfg_target_gcc_flags" \
  BINDIR="$cfg_dir_rootfs/bin"
}

target_install()
{
  cd app/str2str/gcc
  $cmd_make \
  CC=$cmd_target_cc \
  CTARGET="$cfg_target_gcc_flags " \
  BINDIR="$cfg_dir_rootfs/bin" \
  install
}
```

Listing C.2: Build Rule for str2str

Appendix D

Schematics and Board Layouts

D.1 RS232 to 3.3TTL Level Shifter

The schematic of the converter circuit can be seen in Fig. D.1. The bill of materials is listed in Tab. D.1.

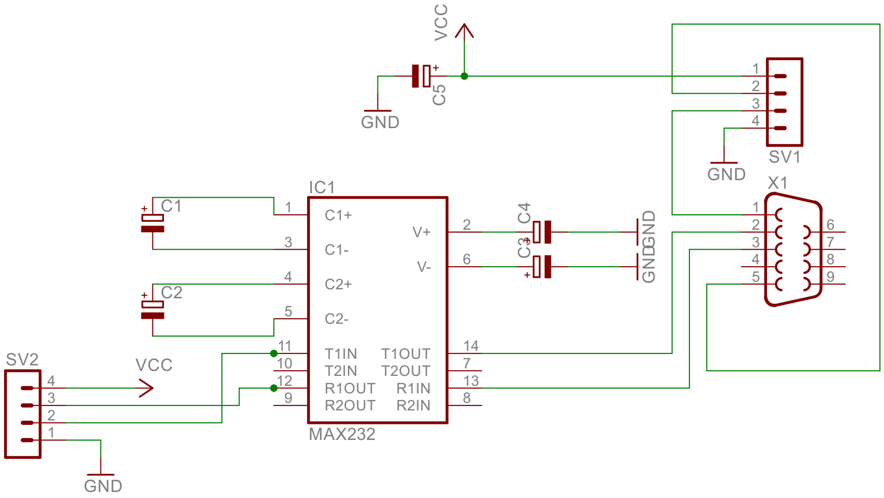


Figure D.1: Converter Schematic

Table D.1: Bill of Materials

Component	**Description**	**Value**
IC1	Max232	n/a
C1	Elecrolytic Capacitor	10 μF
C2	Elecrolytic Capacitor	10 μF
C3	Elecrolytic Capacitor	10 μF
C4	Elecrolytic Capacitor	10 μF
C5	Elecrolytic Capacitor	10 μF
X1	Female D-Sub 9 Connector	n/a
SV1	4-pin Male Header	n/a
SV2	4-pin Male Header	n/a

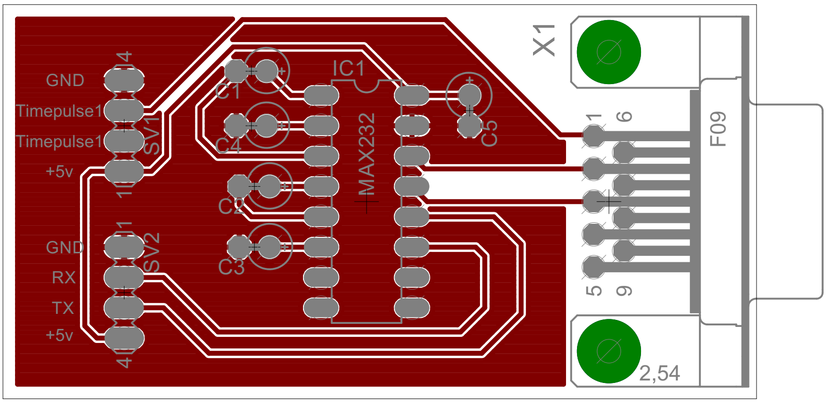


Figure D.2: Top Layer Layout

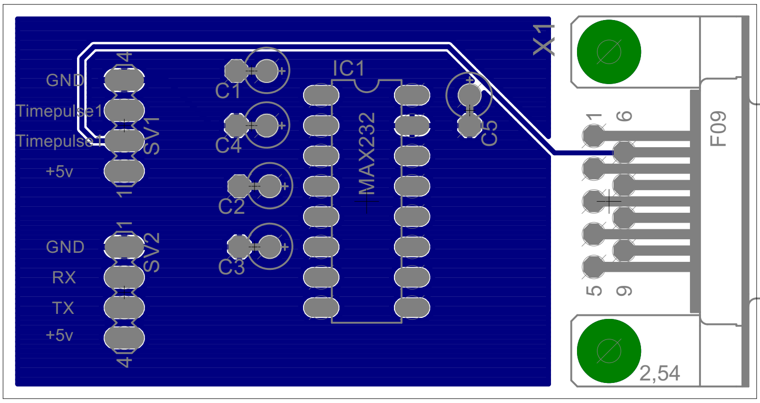


Figure D.3: Bottom Layer Layout