NTNU

Innovation and Creativity

Sensor Integration for Nonlinear Navigation System in Underwater Vehicles

Henrik Taule Foss
Einar Tøsdal Meland

Master of Science in Engineering Cybernetics

Submission date:	May 2007
Supervisor:	Kristin Ytterstad Pettersen, ITK
Co-supervisor:	Erik Kyrkjebø, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

RovNav is Poseidon Innovation's program for visualization of subsea operations. This program shows static objects like manifolds and pipelines in addition to dynamic objects like ROVs and AUVs. It is important that the position of these vessels are accurate in order to have agreement between the visualization of the position of these vessels and their actual position.
The challenge is to determine the position based on several sensors with different properties with regard to noise, accuracy and update frequenzy. This problem may be solved developing a sensor integrator that processes data from the different sensors and provides a stable estimate of the position for the RovNav program.

1. Modify the nonlinear observer presented in B. Vik's PhD thesis: Nonlinear Design and Analysis of Integrated GPS and Inertial Navigation Systems, 2000:4-W, to include HiPAP bias estimates.

2. Implement this modified nonlinear observer and run extensive robustness tests

3. Implement an unscented Kalman filter and run extensive robustness tests

4. Implement filter with time indexing to take into account that measurements will be available some time after being valid. (This issue is most relevant for the HiPAP)

5. Based on the theoretical properties and the simulation results above, do a comparison between the performance of the modified nonlinear filter, the unscented Kalman filter and the extended Kalman filter (developed in the pre-project)

6. Based on 5, make a recommendation for the preferred filter for RovNav, and perform an experimental verification of this filter.

Assignment given: 08. January 2007
Supervisor: Kristin Ytterstad Pettersen, ITK

"*Developing a good Kalman filter model is part art and part science.*"

Robert Grover Brown and Patrick Y.C. Hwang

Abstract

This thesis deals with three methods for integrating measurements from different sensors for an underwater vehicle. The sensors that were used are inertial measurement unit (IMU), Doppler velocity log (DVL), Hydro- acoustic position reference system (HPR) and tilt and heading measurements. The external measurements (DVL, HPR and attitude) are used to aid the inertial navigation system (INS) which uses the measurements from the IMU to calculate position, velocity and attitude.

The different methods presented are extended Kalman filter (EKF), unscented Kalman filter (UKF) and a nonlinear observer. The two Kalman filters were implemented as indirect filters, while the nonlinear observer was implemented as a direct filter. The main difference between the EKF and UKF is that UKF does not make any linarizations such that it captures the covariance of the system more accurate than EKF.

To compare the different approaches a navigation system was implemented using Matlab and simulations were carried out to test accuracy and robustness. The nonlinear observer has the most accurate position estimate when all measurements are available. It performed slightly better than UKF which again was more accurate than EKF. A greater difference was seen between UKF and EKF when the noise characteristics in the filters were wrong. For velocity and attitude all estimates were unbiased, but the nonlinear observer produced estimates with far more noise than what the Kalman filters did.

All filters handled losing the HPR well. The nonlinear observer did not manage to limit the error in the case of DVL loss as opposed to both Kalman filter which have limited error. They performed with the same grade of degradation of the estimates during the loss. When the measurement returned both Kalman filters immediately regained accuracy but the nonlinear observer did not manage to recover.

When losing the IMU measurements both Kalman filters had problems estimating changes in the attitude which again led to error in the position estimate. The position error is however much larger in EKF than UKF. The nonlinear observer has a structure with a separate attitude observer and therefore had a much better attitude estimate during the loss.

From these results UKF is considered the best choice for implementation in a real system. It performs accurate estimates during noisy conditions, and suffers only from limited degradation when measurements are lost, both external and inertial.

Foreword

This report is the result of our master's thesis at The Department of Engineering Cybernetics at The University of Science and Technology (NTNU), spring of 2006.

We would like to extend our thanks to our teaching supervisors: PostDoc Alexey Pavlov, PhD Erik Kyrkjebø and research fellow Even Børhaug. We also want to thank Jan Olav Hallset and Poseidon Innovation for providing us with the problem to be addressed, and for giving us advice and help along the way.

Trondheim, May 29, 2007

Henrik Taule Foss Einar Tøsdal Meland

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this thesis is to design, implement and test the performance of different methods for sensor integration for an ROV/AUV system. Sensor integration is a well known issue in the field of navigation and seeks to combine the best features of each senor to give the best estimate possible. The sensors to be integrated are inertial measurement unit(IMU), Doppler velocity log(DVL), tilt sensor, fluxgate compass, pressure sensor and a hydro-acoustic position reference system(HPR) supported by a Global positioning system(GPS). The system should produce accurate and robust estimates of position, velocity and attitude.

The system designed is to be used in Poseidon's tool for visualization of sub sea operations, RovNav. This program shows static objects like manifolds and pipelines in addition to dynamic objects like ROVs and AUVs. In order to have agreement between the visualization and the actual system state, it is important that the estimated position is accurate.

The problem is that the IMU has biases on it's measurements. This causes the Inertial Navigation System(INS) to have increasing error in time, hence these errors needs to be compensated for in order to achieve correct estimates. This is solved by using supporting measurements like GPS and DVL. These have lower update frequencies and higher noise level than the IMU, but does not drift in time and can be used to limit the error in the INS.

1.1 Design Plan

When designing the different filters a systematic approach is chosen. First the system equations are derived. They are then implemented in Matlab. Then simulators of the different measurements will be designed and implemented. The next step is to test the simulators to see if all measurements are simulated correctly, both with and without generated noise. To test if the structure of the integrator is correct it will first be tested with clean data, all noise characteristics set to zero. When this is done and the observer produces perfect "estimates" of all states, noisy measurements will be introduced.

If the filter in this case produces satisfactory results, the next step is to test the robustness of the filter with respect to different disturbances such as uncertainty in measurement noise characteristics and measurements losses and delay. If the filter performs satisfactory during these conditions it will be tested with data from real measurement sensors. This is done by gathering data from a real IMU and GPS mounted in a car. These data will then be used as input to the filter instead of data from the simulators. At last a performance evaluation of the real data test will decide if the filter is suited for implementation in RovNav.

1.2 Outline of thesis

The first chapters of this thesis are based on the work done in Foss & Meland (2006). They are included here to give complete background information of the work.

Chapter 2 gives an introduction to the different reference frames used. It also gives an introduction to most of mathematics needed to read this thesis.

Chapter 3 shows the derivation of the navigation equations used in the INS.

Chapter 4 gives error models for the navigation equations and the measurements.

Chapters 5 and 6 deals with the different integration techniques used. Here details about extended Kalman filter, unscented Kalman filter and the derivation of a nonlinear observer for sensor integration will be presented.

Chapter 7 deals with filter initialization.

Chapters 8 and 9 presents the criteria for evaluating the integrators. Here the reader will also find most simulation results and the evaluation discussion. Results from a real data test is found in chapter 9.

Chapter 10 gives conclusions with recommendations for Poseidon and suggested future work.

Appendix A gives a list og abbreviations used in the report.

Appendix B presents some tables and figures with simulation results.

Appendix C gives data sheets of the sensors used in this thesis.

Appendix D gives an overview of the attached simulation files.

Part I

System modelling

Chapter 2

Reference frames and mathematics

2.1 Reference frames

When dealing with navigation we have to define positions and velocities of one system relative to another. For global navigation it is natural to give position and velocity with respect to the earth. When making measurements with different sensors these are made in different reference frames, therefore we have to define the different reference frames used in this project. These are given in Fossen (2002) as

- Earth-centered inertial (ECI) frame. The ECI frame has it's center in the center of the earth with it's z-axis pointing towards the north pole, and it's x-axis pointing towards some fixed point i space. This frame is considered to be inertial, i.e. a non accelerating reference frame in which Newton's laws apply.
- Earth-centered Earth-fixed (ECEF) frame. The ECEF has, as ECI, it's center in the earth's center and it's z-axis pointing towards the north pole. Unlike ECI the ECEF is earth-fixed which means that it rotates with the earth relative to the ECI.
- North-East-Down (NED) frame. The NED-frame is defined as the tangent plane on the earth's surface at the location of the vessel. The x-axis pointing north, the y-axis pointing east and the z-axis pointing down.
- BODY-fixed frame. The body-fixed frame is fixed to the vessel with the x-, y- and z-axis most often chosen to coincide with the principal axes of inertia(x-axis pointing forward, y-axis pointing to starboard and z-axis pointing down).

Figure 2.1 shows how the reference frames are related to each other. To represent the different positions and velocities in the different frames we use the following superscripts for the different reference frames:

- i for ECI

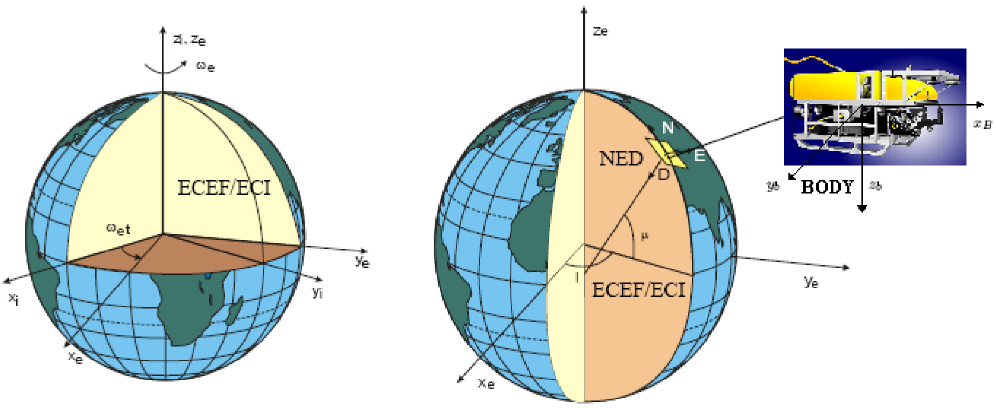

Figure 2.1: ECEF rotates with angular rate w_e w.r.t. ECI. NED is located on the earth's tangent plane, and the BODY-frame is fixed to the vessel

- e for ECEF
- n for NED
- b for BODY

2.2 Kinematics

This section will give a brief introduction to concepts of rigid body kinematics used in this report. Starting with the concept of rotational matrices to describe a rotation of one reference frame with respect to another. Further also the angle-axis parameterization and the concept of unit quaternions will be presented. A large part of this chapter is based on the work presented by Egeland & Gravdahl (2002) and Sciavicco & Sicilano (2000).

2.2.1 Rotation matrix

Given two coordinate frames a and b. A vector $\overrightarrow{v}$ decomposed in coordinate frame a is then written $\mathbf{v}^a$. That a vector is decomposed in coordinate frame a means that it is expressed by the unity vectors of that coordinate frame:

$$\mathbf{v}^a = \begin{pmatrix} v_1^a \\ v_2^a \\ v_3^a \end{pmatrix} \tag{2.1}$$

and when it is decomposed in frame b

$$\mathbf{v}^b = \begin{pmatrix} v_1^b \\ v_2^b \\ v_3^b \end{pmatrix} \tag{2.2}$$

What we want next is to find a relationship between the vectors $\mathbf{v}^a$ and $\mathbf{v}^b$. This relationship can be expressed as

$$\mathbf{v}^a = \mathbf{R}_b^a \cdot \mathbf{v}^b \tag{2.3}$$

where $\mathbf{R}_b^a$ is called the rotation matrix from b to a. This means that if you pre multiply the vector $\mathbf{v}^b$ by $\mathbf{R}_b^a$ the resulting vector is the vector $\mathbf{v}$ decomposed in $a \Rightarrow \mathbf{v}^a$. The rotation matrix is orthogonal and satisfies

$$\mathbf{R}_b^a = (\mathbf{R}_a^b)^{-1} = (\mathbf{R}_a^b)^T \tag{2.4}$$

If we introduce a third coordinate frame c the rotation from c to a can be performed as first a rotation from c to b, and then from b to a:

$$\mathbf{R}_c^a = \mathbf{R}_b^a \mathbf{R}_c^b \tag{2.5}$$

The elements of the rotation matrix for a simple rotation in three dimensions are:

$$\mathbf{R}_a^b = \begin{bmatrix} \overrightarrow{x_b} \cdot \overrightarrow{x_a} & \overrightarrow{x_b} \cdot \overrightarrow{y_a} & \overrightarrow{x_b} \cdot \overrightarrow{z_a} \\ \overrightarrow{y_b} \cdot \overrightarrow{x_a} & \overrightarrow{y_b} \cdot \overrightarrow{y_a} & \overrightarrow{y_b} \cdot \overrightarrow{z_a} \\ \overrightarrow{z_b} \cdot \overrightarrow{x_a} & \overrightarrow{z_b} \cdot \overrightarrow{y_a} & \overrightarrow{z_b} \cdot \overrightarrow{z_a} \end{bmatrix} \tag{2.6}$$

If we perform a rotation ψ about the z-axis as in figure 2.2 the rotation matrix of equation 2.6 becomes

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

Equivalent the rotations about the x- and y-axes become

$$\begin{aligned} \mathbf{R}_x(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\ \mathbf{R}_y(\theta) &= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \end{aligned} \tag{2.8}$$

where the angles ϕ, θ, ψ are called the Euler angles.

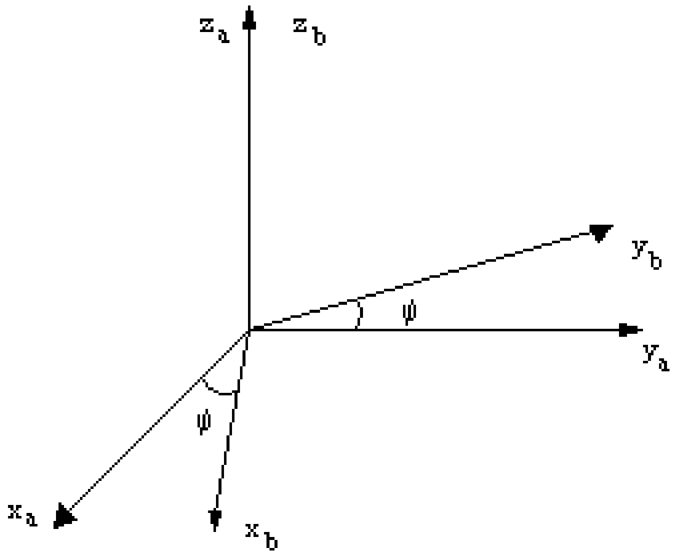

Figure 2.2: Rotation of an angle ψ about the z-axis

2.2.2 Angle-axis parameters

Since the rotation matrix $\mathbf{R}_a^b$ is orthogonal it can be shown that one of the eigenvalues is equal to one (Egeland & Gravdahl 2002), and the corresponding unit eigenvector $\mathbf{k}$ satisfies

$$\mathbf{R}_a^b \cdot \mathbf{k} = \mathbf{k} \tag{2.9}$$

Let $\mathbf{k}^a$ be the coordinate vector of $\overrightarrow{k}$ in reference frame a, then

$$\mathbf{k}^a = \mathbf{R}_b^a \cdot \mathbf{k}^b \Rightarrow \mathbf{k}^a = \mathbf{k}^b = \mathbf{k} \tag{2.10}$$

This shows that the vector $\mathbf{k}$ has the same coordinates in both a and b coordinate frames (Egeland & Gravdahl 2002), hence we can describe the rotation from a to b as a simple rotation θ about $\mathbf{k}$. The parameters $(\theta, \mathbf{k})$ are referred to as the angle-axis parameters and the rotation matrix can now be written as

$$\mathbf{R}_b^a = \mathbf{R}_{k,\theta} = \cos\theta\mathbf{I} + \mathbf{S}(\mathbf{k})\sin\theta + \mathbf{k}\mathbf{k}^T(1-\cos\theta) \tag{2.11}$$

where $\mathbf{S}(\mathbf{k})$ is the skew-symmetric matrix of $\mathbf{k}$ and is defined as the cross product operator.

$$\mathbf{S}(\mathbf{k}) = \mathbf{k}^\times = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \tag{2.12}$$

2.2.3 Quaternions

A weakness with the Euler-angle representation of the rotation matrix is that for $\theta = \pm\frac{\pi}{2}$ the matrix $\mathbf{R}_b^a$ becomes singular (Fossen 2002), we also encounter a singularity problem with the angle-axis representation for $\theta = 0°$ and $\theta = \pi$ (Sciavicco & Sicilano 2000). To deal with this problem we have to use an alternative representation of the rotation called quaternions which is defined by the angle-axis parameters as

$$\begin{aligned} \eta &= \cos\frac{\theta}{2} \\ \epsilon &= \mathbf{k}\sin\frac{\theta}{2} \end{aligned} \tag{2.13}$$

η and ϵ are called the Euler parameters and are constrained by the condition

$$\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1 \tag{2.14}$$

These Euler parameters may be put in a vector called the unit quaternion vector:

$$\mathbf{q} = \begin{bmatrix} \eta \\ \epsilon \end{bmatrix} \tag{2.15}$$

The rotation matrix given by the corresponding quaternion is (Egeland & Gravdahl 2002)

$$\mathbf{R}_e(\eta, \epsilon) = \mathbf{I} + 2\eta\mathbf{S}(\epsilon) + 2\mathbf{S}^2(\epsilon) \tag{2.16}$$

The quaternion extracted from $\mathbf{R}^T = \mathbf{R}^{-1}$ is the inverse quaternion defined as

$$\mathbf{q}^{-1} = \begin{bmatrix} \eta \\ -\epsilon \end{bmatrix} \tag{2.17}$$

The product of two quaternions is defined as (Egeland & Gravdahl 2002)

$$\mathbf{q} = \mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} \eta_1 \cdot \eta_2 - \epsilon_1^T \epsilon_2 \\ \eta_1 \cdot \epsilon_2 + \eta_2 \cdot \epsilon_1 + \mathbf{S}(\epsilon_1) \cdot \epsilon_2 \end{bmatrix} \tag{2.18}$$

The quaternion product of an inverse quaternion and another quaternion (Egeland & Gravdahl 2002)

$$\tilde{\mathbf{q}} = \mathbf{q}_1^{-1} \otimes \mathbf{q}_2 \tag{2.19}$$

is defined as the error quaternion.

2.2.4 Useful relations

- The given vector cross product $\mathbf{s} = \mathbf{p} \times \mathbf{v}$ can be expressed in terms of coordinate frames a and b as

$$\begin{aligned} \mathbf{s}^a &= \mathbf{S}(\mathbf{p}^a)\mathbf{v}^a \\ \mathbf{s}^b &= \mathbf{S}(\mathbf{p}^b)\mathbf{v}^b \end{aligned} \tag{2.20}$$

which can be written as

$$\mathbf{s}^a = \mathbf{R}_b^a \mathbf{S}(\mathbf{p}^b)\mathbf{R}_a^b \mathbf{v}^a \tag{2.21}$$

from where it can be seen (from 2.20) that

$$\begin{aligned} \mathbf{S}(\mathbf{p}^a) &= \mathbf{R}_b^a \mathbf{S}(\mathbf{p}^b)\mathbf{R}_a^b \\ \mathbf{S}(\mathbf{p}_b) &= \mathbf{R}_a^b \mathbf{S}(\mathbf{p}^a)\mathbf{R}_b^a \end{aligned} \tag{2.22}$$

This relationship is know as the similarity transform.

- An expression for $\dot{\mathbf{R}}_a^b$ will be derived here. As stated in section 2.2.1 the rotation matrix is orthogonal and satisfies

$$\mathbf{R}_b^a (\mathbf{R}_b^a)^T = \mathbf{I} \tag{2.23}$$

The time derivative of this product is

$$\frac{d}{dt}\left[\mathbf{R}_b^a (\mathbf{R}_b^a)^T\right] = \dot{\mathbf{R}}_b^a (\mathbf{R}_b^a)^T + \mathbf{R}_b^a (\dot{\mathbf{R}}_b^a)^T = 0 \tag{2.24}$$

According to Egeland & Gravdahl (2002) this shows that $\dot{\mathbf{R}}_b^a (\mathbf{R}_b^a)^T$ is a skew symmetric matrix defined as

$$\mathbf{S}(\mathbf{w}_{ab}^a) = \dot{\mathbf{R}}_b^a (\mathbf{R}_b^a)^T \tag{2.25}$$

where $\mathbf{w}_{ab}^a$ is said to be the angular velocity of frame b relative to a. From this relation it can be seen that the kinematic differential equation of the rotation matrix is

$$\begin{aligned} \dot{\mathbf{R}}_b^a &= \mathbf{S}(\mathbf{w}_{ab}^a)\mathbf{R}_b^a \\ \dot{\mathbf{R}}_b^a &= \mathbf{R}_b^a \mathbf{S}(\mathbf{w}_{ab}^b) \end{aligned} \tag{2.26}$$

- The rotation matrix is dependent on the angles or quaternion between the two reference frames. This orientation is subject to errors such that the rotation matrix also will become erroneous. According to Egeland & Gravdahl (2002) the rotation matrix can be written

$$\hat{\mathbf{R}} = \mathbf{R}\tilde{\mathbf{R}} \tag{2.27}$$

where $\hat{\mathbf{R}}$ is the "measured" matrix and $\tilde{\mathbf{R}}$ is the error in the rotation matrix. When the rotation matrix is a function of quaternions this can be written as:

$$\hat{\mathbf{R}} = \quad \mathbf{R}\tilde{\mathbf{R}} \quad = \quad \mathbf{R}(\mathbf{I} + 2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + \mathbf{S}^2(\tilde{\epsilon})) \tag{2.28}$$

$$= \quad \mathbf{R} + \delta\mathbf{R} \tag{2.29}$$

where $\delta\mathbf{R} = \mathbf{R}(2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + \mathbf{S}^2(\tilde{\epsilon}))$

- Egeland & Gravdahl (2002) has shown that the time derivative of the deviation quaternion is:

$$\dot{\tilde{\mathbf{q}}} = \mathbf{q}^{-1} \otimes \dot{\hat{\mathbf{q}}} + \dot{\mathbf{q}}^{-1} \otimes \hat{\mathbf{q}} \tag{2.30}$$

$$\dot{\mathbf{q}}^{-1} = -\mathbf{q}^{-1} \otimes \dot{\mathbf{q}} \otimes \mathbf{q}^{-1} \tag{2.31}$$

2.2.5 Cholesky Decomposition

Given a symmetric positive definite matrix A. The Cholesky decomposition produces a matrix L such that $A = LL^T$ or $A = L^T L$ (Nocedal & Wright 1999) where L is either a lower or upper diagonal matrix. L is regarded the square root of A.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \ddots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ 0 & l_{22} & \cdots & l_{2n} \\ \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & l_{nn} \end{bmatrix}^T \tag{2.32}$$

The elements of L can be found by (Nocedal & Wright 1999)

$$l_{ii} = \sqrt{a_{ii} - \textstyle\sum_{k=1}^{i-1} l_{ik}^2} \tag{2.33}$$

$$l_{ji} = \tfrac{a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik}}{l_{ii}} \tag{2.34}$$

for $i = 1, \cdots, n$ and $j = i + 1, \cdots, n$.

2.2.6 Lie Derivative

Given the nonlinear system

$$\dot{x} = f(x) + g(x)u \tag{2.35}$$

$$y = h(x) \tag{2.36}$$

Taking the derivative of h along the trajectories of the system $\dot{x} = f(x)$ is known as the Lie derivative (Khalil 2002). The notation used for the Lie derivative is

$$L_f^0 h(x) = h(x) \tag{2.37}$$

$$L_f^2 h(x) = L_f L_f h(x) = \frac{\partial(L_f h)}{\partial x} f(x) \tag{2.38}$$

$$L_f^k h(x) = L_f L_f^{k-1} h(x) = \frac{\partial(L_f^{k-1} h)}{\partial x} f(x) \tag{2.39}$$

2.2.7 Vector notation

The following table (table 2.1) gives the vector notation that will be used in this report.

Vector	Description
$\mathbf{p}_{ab}$	Vector pointing from origin in a to origin in b.
$\mathbf{v}_{ab}^{b}$	The linear velocity of origin in b w.r.t. origin in a decomposed in b (seen from b's perspective).
$\mathbf{a}_{ab}^{b}$	The linear acceleration of origin in b w.r.t. origin in a decomposed in b (seen from b's perspective).
$\mathbf{q}_{ab}$	Orientation of a relative to b. Attitude represented by quaternions $\mathbf{q} = \begin{bmatrix}\eta & \epsilon_1 & \epsilon_2 & \epsilon_3\end{bmatrix}^T$
ω_{ab}^{b}	The angular rate of b w.r.t. to a decomposed in b (seen from b's perspective).
r_{ab}	The distance from origin in a to origin in b. $r_{en} = \lvert\mathbf{p}_{en}\rvert$
$\mathbf{R}_{a}^{b}$	The rotation matrix which rotates a vector from a to b

Table 2.1: The labeling of vectors and rotation matrices

Chapter 3

Navigation equations

3.1 Inertial navigation system

The inertial navigation system (INS) is used to calculate the wanted states based on the inertial measurement unit (IMU) output. For further details on the IMU see section 4.1. When the IMU/INS system is strapped to the body of the ROV/AUV the measurements that are supplied from the IMU are $\mathbf{f}^b_{IMU}$ and ω^b_{IMU}. These describe the specific force and the angular velocity of the body frame in reference to an inertial frame. The goal of the INS is to use these vectors to calculate three different states:

- The attitude of the body in reference to the NED-frame, denoted $\mathbf{q}_{nb}$
- The velocity of the body in reference to the earth given in the NED-frame, denoted $\mathbf{v}^n_{eb}$
- The position given in longitude, latitude and height, denoted $\mathbf{p}^e$

This section will describe the derivation of the equations used to calculate these states. The choice of using Euler parameters (quaternion) to describe the attitude is done to ensure a non-singular representation of the attitude. This is important for an ROV that has the possibility of moving in all directions and attitudes. It is further more assumed that the velocity can be described in the NED frame. The NED frame is subject to singularity close to the north and south poles. By using the NED frame the system is limited to working in a certain distance from the poles to ensure non-singularity.

The derivation of these equations are described in Farrell & Barth (1999).

3.1.1 Attitude

The goal of this section is to derive a kinematic equation for the attitude quaternion $\mathbf{q}_{nb} = \begin{bmatrix} \eta & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^T$. In Egeland & Gravdahl (2002) it is shown that the kinematic differential equation for the quaternion vector is:

$$\dot{\mathbf{q}}_{nb} = \frac{1}{2}\mathbf{q}_{nb} \otimes \begin{bmatrix} o \\ \omega_{nb}^b \end{bmatrix} \tag{3.1}$$

The angular velocity between to frames of reference can be described as the difference between the angular velocity of both frames in reference to a third initial frame. Hence:

$$\omega_{nb}^b = \omega_{ib}^b - \omega_{in}^b \tag{3.2}$$

From Egeland & Gravdahl (2002) a rotation represented by a unit quaternion product is

$$\mathbf{q}_{nb} \otimes \begin{bmatrix} 0 \\ \omega_{in}^b \end{bmatrix} \otimes \mathbf{q}_{nb}^{-1} = \begin{bmatrix} 0 \\ \mathbf{R}_b^n \omega_{in}^b \end{bmatrix} = \begin{bmatrix} o \\ \omega_{in}^n \end{bmatrix} \tag{3.3}$$

Combining the equations 3.1-3.3 gives:

$$\dot{\mathbf{q}}_{nb} = \frac{1}{2}\mathbf{q}_{nb} \otimes \begin{bmatrix} o \\ \omega_{ib}^b \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 0 \\ \omega_{in}^n \end{bmatrix} \otimes \mathbf{q}_{nb} \tag{3.4}$$

In subsequent chapters it will be used that $\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n$ where w_{ie}^n is the earth's angular velocity.

3.1.2 Velocity

This section describes the derivation of the kinematic equation of the velocity of the vessel in reference to the earth given in the NED-frame. The relationship between the vessel velocity in the ECEF frame and its velocity in the NED-frame is described in Farrell & Barth (1999) as:

$$\mathbf{v}_{eb}^n = \mathbf{R}_e^n \dot{\mathbf{p}}_{eb}^e \tag{3.5}$$

Where $\mathbf{v}_{eb}^n$ is the velocity in the NED-frame and $\dot{\mathbf{p}}_{eb}^e$ is the time derivative of the position in ECEF coordinates. The derivative of 3.5 yields by using equation 2.26:

$$\dot{\mathbf{v}}_{eb}^n = \mathbf{R}_e^n \mathbf{S}(\omega_{ne}^e)\dot{\mathbf{p}}_{eb}^e + \mathbf{R}_e^n \ddot{\mathbf{p}}_{eb}^e \tag{3.6}$$

By using 3.5 and changing the direction of the angular velocity, the rotational matrix from BODY to NED cancel and the expression can be written:

$$\dot{\mathbf{v}}^n_{eb} = -\mathbf{S}(\omega^n_{en})\mathbf{v}^n_{eb} + \mathbf{R}^n_e \ddot{\mathbf{p}}^e_{eb} \tag{3.7}$$

To find an expression for $\ddot{\mathbf{p}}^e_{eb}$ Idsø (1999) showed that this equality holds:

$$\ddot{\mathbf{p}}^e_{eb} = \mathbf{R}^e_i \ddot{\mathbf{p}}^i_{eb} - 2\mathbf{S}(\omega^e_{ie})\dot{\mathbf{p}}^e_{eb} - \mathbf{S}(\omega^e_{ie})\mathbf{S}(\omega^e_{ie})\mathbf{p}^e_{eb} \tag{3.8}$$

Using this and some further steps, it can be showed that $\dot{\mathbf{v}}^n_{eb}$ can be written as:

$$\dot{\mathbf{v}}^n_{eb} = \mathbf{R}^n_b \mathbf{f}^b_{ib} + \bar{\mathbf{g}}^n_{eb} - (2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{ie}))\mathbf{v}^n_{eb} \tag{3.9}$$

The term $\bar{\mathbf{g}}^n_{eb}$ is referred to as plum bob gravity and is the sum of the gravity and the centripetal acceleration. Some mathematical steps where skipped in this derivation. For the complete derivation of the equations the reader is referred to Farrell & Barth (1999).

3.1.3 Position

To describe the geodetic position it has been showed in Farrell & Barth (1999) that one can use the $\mathbf{v}^n = [v_N, v_E, v_D]^T$ in the relation:

$$\begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} = \begin{bmatrix} r_\lambda + h & 0 & 0 \\ 0 & (r_\phi + h)cos(\lambda) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{\lambda} \\ \dot{\phi} \\ \dot{h} \end{bmatrix} \tag{3.10}$$

Where $\begin{bmatrix} \lambda & \phi & h \end{bmatrix}^T$ is the position in longitude, latitude and height. By inversion:

$$\begin{bmatrix} \dot{\lambda} \\ \dot{\phi} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{r_\lambda + h} & 0 & 0 \\ 0 & \frac{1}{(r_\phi + h)cos(\lambda)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} \tag{3.11}$$

By not assuming the earth to be circular the r_λ and r_ϕ is not equal, and can be described as:

$$r_\lambda = \frac{a(1 - e^2)}{[1 - e^2 sin^2(\lambda)]^{1.5}} \tag{3.12}$$

$$r_\phi = \frac{a}{[1 - e^2 sin^2(\lambda)]^{0.5}} \tag{3.13}$$

Where a is the equatorial radius of the earth and e is the eccentricity of the earth.

Chapter 4

Error modelling

This chapter is based on work done by Gade (1997) and Farrell & Barth (1999).

The relationship between the measured variable $\hat{x}$ and the real variable x is often described by a linear function. To describe this function you need to know the scale factor (k_1) and the offset (k_2) . The offset is the measurement when the real variable is equal to zero. The scale factor is the slope of the measured variable in relation to the real variable. The measurement can then be written in the following form:

$$\hat{x} = k_1 x + k_2 \tag{4.1}$$

If this function is known and accurate the variable is easily found by inverting the function. However, all sensors are subject to a certain error caused by time, temperature, past variable values[1] etc. Included in the error is also a contribution from the stochastic variation in the output of any system, called noise.

Figure 4.1 illustrates the difference between an actual relationship between x and $\hat{x}$ and the linear estimate.

To describe the error in the measurement it is also possible to write the equation 4.1 as the real value x plus an error δx:

$$\hat{x} = x + \delta x \tag{4.2}$$

Farrell & Barth (1999) shows that equation 4.2 can be expressed as following:

$$\hat{x} = \delta x_{sf} * x + \delta x_{bias} + \xi \tag{4.3}$$

The offset is often represented as a slowly varying bias, δx_{bias}. The scale factor error is represented by δx_{sf} and ξ is the white measurement noise. Not all the measurements have the same properties and are therefore subject to different errors. A summery of the instrument properties is given in table 4.1.

[1] When the measurement is dependent of its past values it is known as hysteresis

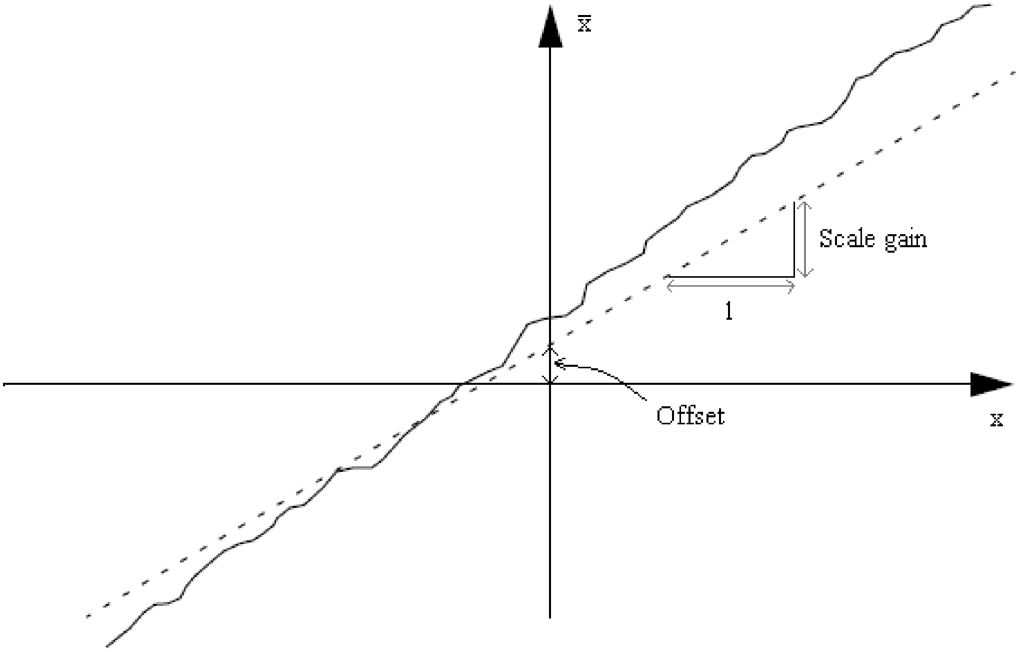

Figure 4.1: Linear measurement function

Sensor	Colored part	White-noise part
HiPAP + DGPS	X	X
IMU	X	X
DVL		X
Tilt		X
Compass		X

Table 4.1: Noise properties of measurement instruments

4.1 Inertial measurement unit

An inertial measurement unit is a device for measuring accelerations and angular velocities of an object. The IMU consists of three gyroscopes and three accelerometers.

Accelerometer

The accelerometer is a device that measures the specific force on an object in one dimension. By using three accelerometers mounted orthogonal to each other, it is possible to calculate the acceleration in three dimensions. There are numerous implementations of accelerometers, which all have different characteristics.

By using equation 4.3 to describe the error in the accelerometer, Farrell & Barth (1999) presents the following equation for each of the three accelerometers:

$$\hat{\mathbf{f}}^b_{ib,i} = \mathbf{f}^b_{ib,i} + \delta\mathbf{f}^b_{ib,i} * \mathbf{f}^b_{ib,i} + \delta\mathbf{f}^b_{ib,bias,i} + \xi_{f,i} \tag{4.4}$$

Where $\hat{\mathbf{f}}^b_{ib,i}$ represents the measurement and $\mathbf{f}^b_{ib,i}$ is the real value. The index i represents the 3 mounting axes (x', y', z'). The mounting axes will never be concurrent with the real axes in the body frame, and are therefore marked.

$$\delta\mathbf{f}^b_{ib,i} = \begin{bmatrix} \delta f^b_{ib,scalefactor,x} & \delta\phi_{xy} & \delta\phi_{xz} \\ \delta\phi_{yx} & \delta f^b_{ib,scalefactor,y} & \delta\phi_{yz} \\ \delta\phi_{zx} & \delta\phi_{zy} & \delta f^b_{ib,scalefactor,z} \end{bmatrix} \tag{4.5}$$

$\delta\mathbf{f}^b_{ib,scalefactor,x}$ is the error coefficient caused by the error in scale factor, while the of diagonal elements $\delta\phi_{ij}$ is the error caused by the fact that the mounting axes are not exactly concurrent with the frame axes. For this purpose the goal is to estimate the order of magnitude of the error. To do this the error is described as a stochastic variable. Errors that change over time can be modeled as stochastic processes. By representing the errors in this way it is possible to describe the error by statistical variables, such as standard deviation and variance.

Gyro A gyro measures the angular velocity of a object about a given axis. Gyros can be implemented in several different ways, each using a physical effect caused by rotation. The most known is probably the mechanical gyro that uses conservation of spin to find the rotational acceleration. Other implementations uses conservation of orientation (liquid ball gyro), Coriolis effect (tuning fork gyro) and Sagnac effect (fiberoptic gyro) to measure the rotation (Lawrence 1993).

As for acceleration the error in the gyro can be described with:

$$\hat{\omega}^b_{ib} = \omega^b_{ib} + \delta\omega^b_{ib} \tag{4.6}$$

Where the error term $\delta\omega^b_{ib}$ is:

$$\begin{aligned} \delta\omega^b_{ib} = &\begin{bmatrix} \Delta\omega^b_{ib,scalefactor,x} & 0 & 0 \\ 0 & \Delta\omega^b_{ib,scalefactor,y} & 0 \\ 0 & 0 & \Delta\omega^b_{ib,scalefactor,z} \end{bmatrix} \omega^b_{ib} \\ &+ \Delta\omega^b_{ib,bias} + \Delta\omega^b_{ib,skew} + \xi_\omega \end{aligned} \tag{4.7}$$

The error in different directions are independent of each other and must therefor be represented as vectors. This gives:

$$\Delta\omega^b_{ib,bias} = \begin{bmatrix} \Delta\omega^b_{ib,bias,x} \\ \Delta\omega^b_{ib,bias,y} \\ \Delta\omega^b_{ib,bias,z} \end{bmatrix} \quad and \quad \xi_\omega = \begin{bmatrix} \xi_{\omega,x} \\ \xi_{\omega,y} \\ \xi_{\omega,z} \end{bmatrix} \tag{4.8}$$

IMU errors

The error in both gyro and accelerometer measurements can be modeled as slowly varying bias with added white noise. The bias can be modeled as a $1st$ order Markov process:

$$\dot{\mathbf{b}}_{f_{IMU}} = -T_{f_{IMU}}^{-1}\mathbf{b}_{f_{IMU}} + \gamma_{f_{IMU}} \tag{4.9}$$

$$\dot{\beta}_{\omega_{IMU}} = -T_{\omega_{IMU}}^{-1}\beta_{\omega_{IMU}} + \gamma_{\omega_{IMU}} \tag{4.10}$$

where $\gamma_{_{IMU}}$ is the white noise that drives the bias. The expression for the acceleration measurement can then be written as

$$\mathbf{f}_{IMU}^{b} = \mathbf{f}_{ib}^{b} + \mathbf{b}_{f_{IMU}} + \xi_f \tag{4.11}$$

and the expression for the angular velocity as

$$\omega_{IMU}^{b} = \omega_{ib}^{b} + \beta_{\omega_{IMU}} + \xi_\omega \tag{4.12}$$

where ξ_f and ξ_ω are the measurement white noise.

4.2 Inertial navigation system

This section will describe the error in navigation states based on the navigational equations given in chapter 3. The error development is based on work done by Farrell & Barth (1999) and Idsø (1999).

$$\dot{\mathbf{v}}_{eb}^{n} = \mathbf{R}_b^n\mathbf{f}_{ib}^{b} + \bar{\mathbf{g}}_{eb}^{n} - (2\mathbf{S}(\omega_{ie}^{n}) + \mathbf{S}(\omega_{en}^{n}))\mathbf{v}_{eb}^{n} \tag{4.13}$$

$$\dot{\mathbf{q}}_{nb} = \frac{1}{2}\mathbf{q}_{nb} \otimes \begin{bmatrix} 0 \\ \omega_{ib}^{b} \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 0 \\ \omega_{ie}^{n} + \omega_{en}^{n} \end{bmatrix} \otimes \mathbf{q}_{nb} \tag{4.14}$$

$$\begin{bmatrix} \dot{\lambda} \\ \dot{\phi} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{v_N}{r_\lambda + h} \\ \frac{v_E}{(r_\phi + h)\cos\lambda} \\ -v_D \end{bmatrix} \tag{4.15}$$

4.2.1 Velocity

From the IMU there is a measurement of the specific force, $\mathbf{f}_{ib}^{b}$ which inserted in the velocity equation from 4.13 gives:

$$\dot{\hat{\mathbf{v}}}_{eb}^{n} = \hat{\mathbf{R}}_b^n\mathbf{f}_{IMU}^{b} + \hat{\bar{\mathbf{g}}}_{eb}^{n} - (2\mathbf{S}(\hat{\omega}_{ie}^{n}) + \mathbf{S}(\hat{\omega}_{en}^{n}))\hat{\mathbf{v}}_{eb}^{n} \tag{4.16}$$

In order to find an expression in the velocity error $\delta\dot{\mathbf{v}}^n_{eb}$, the real velocity, $\dot{\mathbf{v}}^n_{eb}$, is subtracted from the INS velocity, $\dot{\hat{\mathbf{v}}}^n_{eb}$.

$$\begin{aligned}\delta\dot{\mathbf{v}}^n_{eb} &= \dot{\hat{\mathbf{v}}}^n_{eb} - \dot{\mathbf{v}}^n_{eb}\\ \delta\dot{\mathbf{v}}^n_{eb} &= \hat{\mathbf{R}}^n_b\mathbf{f}^b_{IMU} + \hat{\bar{\mathbf{g}}}^n_{eb} - (2\mathbf{S}(\hat{\omega}^n_{ie}) + \mathbf{S}(\hat{\omega}^n_{en}))\hat{\mathbf{v}}^n_{eb}\\ &\quad - \mathbf{R}^n_b\mathbf{f}^b_{ib} - \bar{\mathbf{g}}^n_{eb} + (2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en}))\mathbf{v}^n_{eb}\end{aligned} \tag{4.17}$$

By introducing expressions:

$$\begin{aligned}\delta\mathbf{g}^n_{eb} &= \hat{\bar{\mathbf{g}}}^n_{eb} - \bar{\mathbf{g}}^n_{eb}\\ \mathbf{v}^n_{eb} &= \hat{\mathbf{v}}^n_{eb} - \delta\mathbf{v}^n_{eb}\end{aligned} \tag{4.18}$$

Equation 4.17 can then be written as:

$$\begin{aligned}\delta\dot{\mathbf{v}}^n_{eb} = \delta\mathbf{g}^n_{eb} &+ \hat{\mathbf{R}}^n_b\mathbf{f}^b_{IMU} - \mathbf{R}^n_b\mathbf{f}^b_{ib} - (2\mathbf{S}(\hat{\omega}^n_{ie}) + \mathbf{S}(\hat{\omega}^n_{en}))\hat{\mathbf{v}}^n_{eb}\\ &+ (2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en}))\hat{\mathbf{v}}^n_{eb} - (2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en}))\delta\mathbf{v}^n_{eb}\end{aligned} \tag{4.19}$$

The $\hat{\mathbf{v}}^n_{eb}$ can be rewritten using the following expressions:

$$\delta\omega_{ie} \quad = \hat{\omega}_{ie} - \omega_{ie} \tag{4.20}$$

$$\delta\omega_{en} \quad = \hat{\omega}_{en} - \omega_{en} \tag{4.21}$$

Using 4.20 and 4.21 in 4.19 yields:

$$\begin{aligned}\delta\dot{\mathbf{v}}^n_{eb} = \delta\mathbf{g}^n_{eb} &+ \hat{\mathbf{R}}^n_b\mathbf{f}^b_{IMU} - \mathbf{R}^n_b\mathbf{f}^b_{ib} - (2\mathbf{S}(\delta\omega^n_{ie}) + \mathbf{S}(\delta\omega^n_{en}))\hat{\mathbf{v}}^n_{eb}\\ &- (2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en}))\delta\mathbf{v}^n_{eb}\end{aligned} \tag{4.22}$$

Further more it can be used from equation 2.29 and 4.11 that:

$$\mathbf{R}^n_b \quad = \hat{\mathbf{R}}^n_b - \delta\mathbf{R}^n_b \tag{4.23}$$

$$\mathbf{f}^b_{ib} \quad = \mathbf{f}^b_{IMU} - \delta\mathbf{f}^b_{ib} \tag{4.24}$$

which gives

$$\begin{aligned}\hat{\mathbf{R}}^n_b\mathbf{f}^b_{IMU} - \mathbf{R}^n_b\mathbf{f}^b_{ib} &= \hat{\mathbf{R}}^n_b\mathbf{f}^b_{IMU} - (\hat{\mathbf{R}}^n_b - \delta\mathbf{R}^n_b)(\mathbf{f}^b_{IMU} - \delta\mathbf{f}^b_{ib})\\ &= \delta\mathbf{R}^n_b\mathbf{f}^b_{IMU} + \mathbf{R}^n_b\delta\mathbf{f}^b_{ib}\end{aligned} \tag{4.25}$$

Using 4.25 in 4.22:

$$\begin{aligned}\delta\dot{\mathbf{v}}^n_{eb} &= \delta\mathbf{g}^n_{eb} + \delta\mathbf{R}^n_b\mathbf{f}^b_{IMU} + \mathbf{R}^n_b\delta\mathbf{f}^b_{ib} - (2\mathbf{S}(\delta\omega^n_{ie}) + \mathbf{S}(\delta\omega^n_{en}))\hat{\mathbf{v}}^n_{eb} \\ &\quad - (2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en}))\delta\mathbf{v}^n_{eb}\end{aligned} \tag{4.26}$$

By assuming that the error in the rotation matrix $\hat{\mathbf{R}}^n_{b,INS}$ is only in the body frame, the rotation deviation can be written as

$$\delta\mathbf{R}^n_b = \mathbf{R}^n_b(2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + 2\mathbf{S}^2(\tilde{\epsilon}) \tag{4.27}$$

This inserted in 4.26 yields:

$$\begin{aligned}\delta\dot{\mathbf{v}}^n_{eb} &= \delta\mathbf{g}^n_{eb} + \mathbf{R}^n_b(2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + 2\mathbf{S}^2(\tilde{\epsilon}))\mathbf{f}^b_{IMU} + \mathbf{R}^n_b\delta\mathbf{f}^b_{ib} - (2\mathbf{S}(\delta\omega^n_{ie}) \\ &\quad + \mathbf{S}(\delta\omega^n_{en}))\hat{\mathbf{v}}^n_{eb} - (2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en}))\delta\mathbf{v}^n_{eb}\end{aligned} \tag{4.28}$$

4.2.2 Attitude

To ensure that the filter works for all attitudes, the attitude is represented by a quaternion. The quaternion is guaranteed to be continuous also around pitch angles (θ) close to $\pm 90°(\frac{\pi}{2})$, which is not the case for Euler angle representation.

The deviation in the attitude quaternion is in equation 2.19 defined as:

$$\tilde{\mathbf{q}} = \mathbf{q}^{-1} \otimes \hat{\mathbf{q}} \tag{4.29}$$

Where $\tilde{\mathbf{q}}$ represents the deviation between $\mathbf{q}$ and $\hat{\mathbf{q}}$. If the to quaternions coincide the deviation quaternion $\tilde{\mathbf{q}}$ is equal to the unit quaternion given as $\tilde{\mathbf{q}} = (1, (0))$. By using the definition of the time derivative of a quaternion from equations 2.31 and 2.30 we can write:

$$\dot{\tilde{\mathbf{q}}} = \mathbf{q}^{-1} \otimes \dot{\hat{\mathbf{q}}} - \mathbf{q}^{-1} \otimes \dot{\mathbf{q}} \otimes \mathbf{q}^{-1} \otimes \hat{\mathbf{q}} \tag{4.30}$$

Further more the kinematic differential equation for the quaternion $\hat{\mathbf{q}}$ can be described by:

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix} 0 \\ \omega^a_{IMU} - \hat{\omega}^a_{in}\end{bmatrix} \otimes \hat{\mathbf{q}} = \frac{1}{2}\hat{\mathbf{q}} \otimes \begin{bmatrix} 0 \\ \omega^b_{IMU} - \hat{\omega}^b_{in}\end{bmatrix} \tag{4.31}$$

Using this and $\hat{\mathbf{q}} = \mathbf{q} \otimes \tilde{\mathbf{q}}$, 4.30 can be written as:

$$\begin{aligned}\dot{\tilde{\mathbf{q}}} &= \frac{1}{2}\begin{bmatrix}0\\ \omega^b_{IMU} - \hat{\omega}^b_{in}\end{bmatrix} \otimes \mathbf{q}^{-1} \otimes \mathbf{q} \times \tilde{\mathbf{q}} - \frac{1}{2}\mathbf{q}^{-1} \otimes \begin{bmatrix}0\\ \omega^b_{nb}\end{bmatrix} \otimes \mathbf{q}^{-1} \otimes \mathbf{q} \otimes \tilde{\mathbf{q}}\\ &= \frac{1}{2}\begin{bmatrix}0\\ \omega^b_{ib} + \delta\omega^b_{ib} - \omega^b_{in} - \delta\omega^b_{in}\end{bmatrix} \otimes \tilde{\mathbf{q}} - \frac{1}{2}\begin{bmatrix}0\\ \omega^b_{ib} - \omega^b_{in}\end{bmatrix} \otimes \tilde{\mathbf{q}}\\ &= \frac{1}{2}\begin{bmatrix}0\\ \delta\omega^b_{ib} - \delta\omega^b_{in}\end{bmatrix} \otimes \tilde{\mathbf{q}}\end{aligned} \tag{4.32}$$

Using the definition of the quaternion product, 4.32 can be written as:

$$\dot{\tilde{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix}0 & -(\delta\omega^b_{ib} - \delta\omega^b_{in})^T\\ \delta\omega^b_{ib} - \delta\omega^b_{in} & \mathbf{S}(\delta\omega^b_{ib} - \delta\omega^b_{in})\end{bmatrix}\tilde{\mathbf{q}} \tag{4.33}$$

4.2.3 Position

Chapter 3 gave the description of how to model the geodetic position. Based on the relationship between geodetic position and navigation-frame velocity it is possible to describe the first order linear error dynamics. The geodetic position is given as:

$$\begin{bmatrix}\dot{\lambda}\\ \dot{\phi}\\ \dot{h}\end{bmatrix} = \begin{bmatrix}\frac{1}{r_\lambda + h} & 0 & 0\\ 0 & \frac{1}{(r_\phi + h)cos(\lambda)} & 0\\ 0 & 0 & -1\end{bmatrix}\begin{bmatrix}v_N\\ v_E\\ v_D\end{bmatrix} \tag{4.34}$$

The radii in the equation above is described by:

$$r_\lambda = \frac{a(1-e^2)}{[1 - e^2 sin^2(\lambda)]^{1.5}} \tag{4.35}$$

$$r_\phi = \frac{a}{[1 - e^2 sin^2(\lambda)]^{0.5}} \tag{4.36}$$

Where a is the equatorial radius of the earth and e is the eccentricity of the earth. By differentiation of equation 4.34 the position error can be determined as:

$$\begin{aligned}\begin{bmatrix}\delta\dot{\lambda}\\ \delta\dot{\phi}\\ \delta\dot{h}\end{bmatrix} &= \begin{bmatrix}0 & 0 & \frac{-v_N}{(r_\lambda + h)^2}\\ \frac{v_E sin(\lambda)}{(r_\phi + h)cos(\lambda)^2} & 0 & \frac{-v_E}{(r_\phi + h)^2 cos(\lambda)}\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\delta\lambda\\ \delta\phi\\ \delta h\end{bmatrix}\\ &+ \begin{bmatrix}\frac{1}{r_\lambda + h} & 0 & 0\\ 0 & \frac{1}{(r_\phi + h)cos(\lambda)} & 0\\ 0 & 0 & -1\end{bmatrix}\begin{bmatrix}\delta v_N\\ \delta v_E\\ \delta v_D\end{bmatrix}\end{aligned} \tag{4.37}$$

4.3 External measurements

In this section we will find a model of the errors we encounter in the measurements of position, velocity and attitude from external measurements. The position defined in 3.11 consists of the horizontal position λ and ϕ and by the depth h of the ROV/AUV. In equation 3.11 the position vector is defined as

$$\mathbf{p} = \begin{bmatrix}\lambda & \phi & h\end{bmatrix}^T \tag{4.38}$$

The velocity is defined in 3.9 as

$$\mathbf{v}^n = \begin{bmatrix}v_N & v_E & v_D\end{bmatrix}^T \tag{4.39}$$

and the attitude is defined in 3.1.1 as

$$\mathbf{q}_{nb} = \begin{bmatrix}\eta & \epsilon_1 & \epsilon_2 & \epsilon_3\end{bmatrix}^T \tag{4.40}$$

To get the complete position and velocity measurements we have to use many different measurement instruments, each with different characteristics and errors.

4.3.1 GPS

To get a reliable global position measurement most marine applications rely on GPS (Global Positioning System). GPS consists of three major segments: space, control and user (Farrell & Barth 1999). The space segments consists of the GPS satellites, the control segment is a system of tracking stations located around the world which keep track of the correction parameters for each satellite. The user segment is the antennas and receivers which process the satellite signals to get the position, velocity and the belonging time stamp. Although GPS is highly accurate it is not accurate enough for many applications due to errors and noise. The error may be divided into two groups: common-mode errors and non common-mode errors.

1. The common-mode errors are errors that are the same for every receiver in a given area such as atmospheric disturbances.

2. Non common-mode errors are errors that depend on the receiver, and is often larger in cheap receivers. E.g. measurement noise.

When looking at the magnitude of the errors it is clear that the common-mode errors deteriorates the signal the most (Farrell & Barth 1999). So if one could estimate these errors in a receiver at a known location and broadcast them to all the receivers in that local area, the accuracy of the GPS could be significantly improved. This problem has been solved and is the principle of DGPS, which is a version of GPS which is much more accurate. When most of the common-mode errors have been removed the largest contributor to DGPS error is known as multi path (Farrell & Barth 1999). Multi path is due to signal reflection from surfaces near the receiver that corrupts the satellite signal. This corruption leads to errors in calculating the time from a satellite signal was sent until it was received. This time is very important in the calculation of the position, hence multi path will result in position error. The magnitude of this error has a standard deviation of $\sigma_{DGPS} = 0.1 - 3m$ (Farrell & Barth 1999). Since the multi path is such a complex phenomenon we model the resulting noise as white noise $\xi_{DGPS} \sim \mathbf{N}(0, \mathbf{W}_{DGPS,k}\delta_{k,j})$ with standard deviation $\sigma_{W_{DGPS}}$. The matrix $\mathbf{W}$ is

$$\mathbf{W}_{DGPS} = \begin{bmatrix} \sigma^2_{W_{DGPS},\lambda} & 0 & 0 \\ 0 & \sigma^2_{W_{DGPS},\phi} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.41}$$

This gives the following model for the DGPS measurement:

$$\hat{\mathbf{p}}_{DGPS} = \mathbf{p} + \xi_{DGPS} \tag{4.42}$$

where

$$\xi_{DGPS} = \begin{bmatrix} \xi_{DGPS,\lambda} \\ \xi_{DGPS,\phi} \\ 0 \end{bmatrix} \tag{4.43}$$

4.3.2 HPR/ HiPAP

In order to get position measurements of a submerged vehicle DGPS may no longer be used. Instead we have to rely on a hydro acoustic positioning reference system (HPR). The HPR consists of a transmitter and a receiver. The transmitter is located in some known location and sends a signal towards the receiver located on the ROV/AUV. When the pulse hits the receiver it is activated and immediately sends a signal back to the transmitter. The transmitter side then calculates the position of the receiver relative to the transmitter. Depending on the application the transmitter can be located either at a ship at surface level, or in a network of transmitters at the sea-bed, often referred to as a LBL network. If the HPR is located on a ship which determines it's position by DGPS the resulting position measurement will include both DGPS error and HPR error:

$$\hat{\mathbf{p}}_{DGPS/HPR} = \mathbf{p} + \delta\mathbf{p}_{HiPAP} + \xi_{HiPAP} + \xi_{DGPS} \tag{4.44}$$

If a network of transmitters is chosen, a potentially higher accuracy may be achieved, in addition the white noise from DGPS is eliminated:

$$\hat{\mathbf{p}}_{HPR_{LBL}} = \mathbf{p} + \delta\mathbf{p}_{HPR_{LBL}} + \xi_{HPR_{LBL}} \tag{4.45}$$

Kongsberg Maritime has a HPR system called HiPAP (High Precision Acoustic Positioning), see figure 4.2. This system offers accurate positioning in a wide range

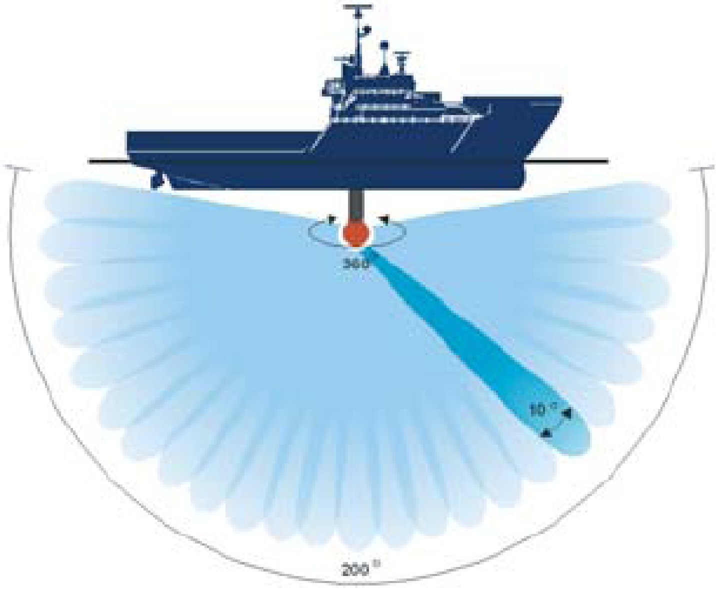

Figure 4.2: Kongsberg Maritime HiPAP illustration

excess 3000 meters (Jensen 2006).

The HPR error consists of three main parts:

1. Measurement noise
2. The velocity of sound is wrongly estimated such that the measured distance from the transmitter to the receiver is wrong.
3. Ray bending due to different temperatures and salt content at different depths.

Measurement noise may be modeled as white noise. While the other two will be slowly varying, and may be modeled as colored noise by a 1st order Markov process:

$$\dot{\delta \mathbf{p}}_{HiPAP} = -\frac{1}{\mathbf{T}_{\mathbf{HiPAP}}}\delta \mathbf{p}_{HiPAP} + \gamma_{HiPAP} \tag{4.46}$$

where

$$\frac{1}{\mathbf{T}_{\mathbf{HiPAP}}} = \begin{bmatrix} \frac{1}{T_{HiPAP,\lambda}} & 0 & 0 \\ 0 & \frac{1}{T_{HiPAP,\phi}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.47}$$

$T_{_{HiPAP,\lambda}} = T_{_{HiPAP,\phi}} = 60s$
The bias driving white noise is

$$\gamma_{_{HiPAP}} = \begin{bmatrix} \gamma_{_{HiPAP,\lambda}} \\ \gamma_{_{HiPAP,\phi}} \\ 0 \end{bmatrix} \tag{4.48}$$

is modeled as $\gamma_{_{HiPAP}} \sim \mathbf{N}(0, \mathbf{Q}_{HiPAP}(t)\delta(\tau))$. The standard deviation of $\gamma_{_{HiPAP}}$ is $\sigma_{Q_{HiPAP}}$. To find $\mathbf{Q}_{HiPAP}$ one need to determine if the elements of $\gamma_{_{HiPAP}}$ are correlated. A hydro acoustic positioning system bases it's positioning on measuring the vertical and horizontal angles and the distance to the receiver. It may be assumed that the distance measurement and the angle measurements are uncorrelated since the distance measurement is calculated by $d = v_{sound} \cdot t$ and the angles are measured by the phase shift of the returned signal. It may also be assumed that the noise in angles is uncorrelated since the measurement is made in two orthogonal planes(x and y), hence the noise in one plane will not affect the other. The matrix may be written as

$$\mathbf{Q} = \begin{bmatrix} \sigma^2_{Q_{HiPAP,\lambda}} & 0 & 0 \\ 0 & \sigma^2_{Q_{HiPAP,\phi}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.49}$$

To find the magnitude of $\sigma^2_{Q_{HiPAP}}$ we need information of the standard deviation of the noise. From the specifications it is given that the magnitude of the error in the angle measurements are $0.2°$ for HiPAP. This gives the following position error:

$$0.2° \cdot \frac{\pi}{180} \cdot \frac{r_{nb}}{r_{eb}} \approx 5.48 \cdot 10^{-10} rad/m \cdot r_{nb} \tag{4.50}$$

r_{nb} is the depth and r_{eb} is the distance from earth's center to the AUV, but is approximated by

$$r_{eb} \approx |r_{en}| = 6371 \cdot 10^3 m \tag{4.51}$$

From this we find the standard deviation of $\gamma_{_{HiPAP}}$ as

$$\sigma_{HiPAP}(t) = \sqrt{\frac{2(5.48 \cdot 10^{-10} r^2_{nb})}{T_{_{HiPAP}}}} = 1.00 \cdot 10^{-10} \cdot r_{nb} \tag{4.52}$$

We now have a complete model of the noise in HiPAP.

4.3.3 Depth

To complete the discussion on position measurement the noise in the depth measurement also has to be modeled. Depth is often measured by a pressure gauge located on the ROV/AUV. If the sensor is calibrated the error in a pressure gauge can be modeled as white noise $\xi_h \sim N(0, W_{h,k}\delta_{k,j})$ with standard deviation σ_h and $W_h = \sigma^2_{W_h}$. The error model for the pressure gauge is then

$$\hat{h} = r_{nb} + \xi_h \tag{4.53}$$

where r_{nb} is the real depth of the AUV.

4.3.4 Complete position model

All the position measurements can now be combined in one complete model where the position is the vector given 4.38:

$$\hat{\mathbf{p}}_{DGPS,HPR,h} = \mathbf{p} + \delta\mathbf{p}_{HiPAP} + \xi_{HiPAP} + \xi_{DGPS} + \xi_h \tag{4.54}$$

The white noise vector can then be combined into the resulting

$$\xi_{DGPS/HPR/h} = \begin{bmatrix} \xi_{DGPS,l} + \xi_{HPR,\lambda} \\ \xi_{DGPS,L} + \xi_{HPR,\phi} \\ \xi_h \end{bmatrix} \tag{4.55}$$

This gives the following model

$$\hat{\mathbf{p}}_{DGPS,HPR,h} = \mathbf{p} + \delta\mathbf{p}_{HiPAP} + \xi_{DGPS,HPR,h} \tag{4.56}$$

where $\delta\mathbf{p}_{HiPAP}$ is given in equation 4.47. Since ξ_{DGPS} and ξ_{HPR} are normally distributed the sum $\xi_{DGPS}+\xi_{HPR}$ will also be normally distributed, hence $\xi_{DGPS,HPR,h} \sim N(0, W_{DGPS,HPR,h,k}\delta_{k,j})$ where

$$\mathbf{W}_{DGPS,HPR,depth,k} = \begin{bmatrix} \sigma^2_{W_{DGPS,HiPAP,\lambda}} & 0 & 0 \\ 0 & \sigma^2_{W_{DGPS,HiPAP,\phi}} & 0 \\ 0 & 0 & \sigma^2_{W_{depth}} \end{bmatrix} \tag{4.57}$$

4.3.5 DVL

As a secondary velocity measurement the ROV/AUV use a Doppler velocity log(DVL). This is a sensor which sends beams of a known frequency towards the sea-bed. By measuring the frequency shift of the returned waves, a measurement of the ROV/AUVs velocity with respect to the sea bed is achieved. By sending out four beams in a Janus configuration a measurement in three dimensions is provided (Teledyne-Instruments 2005). The DVL works best when it's close to the sea bed and the error increases with increasing elevation with respect to the seabed. According to the ROV/AUVs operational elevation a proper frequency of the beams is chosen with increasing frequency with decreasing operational elevation.

By testing of the DVL it is found that the noise may be considered as pure white noise. A model of the DVL measurement will then become

$$\tilde{\mathbf{v}}^b_{eb,dvl,k} = \mathbf{v}_{eb,dvl,k} + \xi_{dvl,k} \tag{4.58}$$

Since the surge and sway velocities are computed separately we may assume that these noises are uncorrelated. The noise in heave will be slightly correlated with the velocities in surge and sway, but to simplify the model this is not accounted for. The resulting model for the white noise is $\xi_{dvl,k} \sim N(0, \mathbf{W}_{dvl,k}\delta_k)$ where

$$\mathbf{W}_{dvl,k} = \begin{bmatrix} \sigma^2_{\xi_{dvl,x,k}} & 0 & 0 \\ 0 & \sigma^2_{\xi_{dvl,y,k}} & 0 \\ 0 & 0 & \sigma^2_{\xi_{dvl,z,k}} \end{bmatrix} \tag{4.59}$$

According to specifications (Teledyne-Instruments 2005) the magnitude of the noise is

$$\begin{aligned} \sigma^2_{\xi_{dvl,x,k}} =& \sigma^2_{\xi_{dvl,y,k}} \approx 0.003m/s \\ \sigma^2_{\xi_{dvl,z,k}} \approx& 0.006m/s \end{aligned} \tag{4.60}$$

4.3.6 Attitude measurement

It can be seen from the specifications of the DVL (Teledyne-Instruments 2005) that it is also equipped with a tilt sensor. This gives the possibility of measuring the tilt in roll and pitch. These measurements are however restricted to a limited range. This tilt measurement can be combined with a compass to give a secondary measurement of the full attitude state.

A flux gate compass is an instrument to measure the intensity of the magnetic field along 3 axis. This is done by mounting three independent sensors along three orthogonal axis. The instrument has several error sources including sensor offset, scale factor min-match, sensor non-orthogonality, sensor tilt and compass deviation. All of them, except the last, are instrumentation errors.

Most of the disturbances that the compass is subject to can be removed by calibrating the compass prior use. But the compass will be influenced by a certain white noise, and the absolute error is in many systems limited around $2 - 3°$ (Olsond 2006).

From the data presented in the data sheet of the DVL, the measurements of the tilt and velocity are only valid for tilt angles between $\pm 15°$. It is therefore assumed that the DVL measurements are not available when the ROV/AUV is in so called helicopter mode[2].

The readings of tilt in 2-axes and a compass reading will describe the roll, pitch and yaw angels given in the body frame. By converting the rotation these angles describe

[2] The vehicle pitch angle is around ± 90 degrees

to a unit quaternion it is possible to compare this rotation to the INS estimate of the attitude. The urement can be written as

$$\hat{\mathbf{q}}_{nb_{dvl},k} = \begin{bmatrix} \hat{\epsilon}_{nb_{dvl},k} \\ \hat{\eta}_{nb_{dvl},k} \end{bmatrix} = \mathbf{q}_{nb,k} \otimes \xi_{q_{dvl},k} \tag{4.61}$$

This gives the following resulting model for the white noise $\xi_{\epsilon_{dvl},k} \sim N(0, \mathbf{W}_{\epsilon_{dvl},k}\delta_k)$ where

$$\mathbf{W}_{\epsilon_{dvl},k}\delta_k = \begin{bmatrix} \sigma^2_{\xi_{\epsilon_{1dvl},k}} & 0 & 0 \\ 0 & \sigma^2_{\xi_{\epsilon_{2dvl},k}} & 0 \\ 0 & 0 & \sigma^2_{\xi_{\epsilon_{3dvl},k}} \end{bmatrix} \tag{4.62}$$

Reasoning for not including $\eta_{nb_{dvl},k}$ can be found in chapter 6.1.

Part II

Integrator design

Chapter 5

Sensor integration

5.1 Integration techniques

This section is based on work done in Farrell & Barth (1999) and Parkinson & Spilker (1996). As described in previous chapters, the GPS/HiPAP and the INS systems have different characteristics that complement each other. By taking advantage of the differences of the two systems, it is possible to get a navigation system that can utilize the desirable characteristics from both systems. The pros and cons for some of the measuring techniques are summarized in table 5.1.

If the system is set up with a stationary HiPAP in the area that the vehicle is operating, the system is not dependent on the GPS. This makes the problem with the possible jamming issues of the GPS irrelevant. The literature refers to two main approaches to filtering the GPS and INS signals.

5.1.1 Direct filtering

In the direct filtering approach for integrating GPS/HiPAP measurement and INS, the states in the filter is the absolute position, velocity and attitude. The implementation of the direct filter have a broad span in implementation complexity. The least complex being a switch that changes to GPS measurement when this is available, and a more complex implementation is using a Kalman filter.

Farrell & Barth (1999) points to three major drawbacks when using Kalman filter for the direct filtering approach:

- The Kalman filter covariance propagating equations would have to be iterated at the high rate of the inertial measurements. The covariance propagation equations are very computationally intensive. This would severely limit the rate at which the inertial measurements could be incorporated.

	GPS / HiPAP	INS
Pros	• Bounded error in position • No need for calibration on startup	• High update rate, bounded by computational approach and equipment • Integration of measurements reduce high frequency measurement noise
Cons	• GPS can be jammed • Low update rate • HiPAP gives delayed measurements • Can lose measurements in time periods	• Unbounded error in position and attitude • Integration amplify low frequency noise

Table 5.1: Pros and cons with INS and GPS/HIPAP

- The measurements driving the filter and the filter states have significant deterministic components that have to be represented by ad hoc models in the filter design.
- The filter must have high bandwidth, since it is estimating the total navigation state, which may change rapidly.

In section 5.3 a nonlinear observer for direct filtering is presented. This observer omits some of the drawbacks encountered when using a Kalman filter since there is no covariance calculations involved. The observer equations are not considerably more complex than the INS navigation equations. This means that the drawback of high bandwidth does not apply to this nonlinear observer.

5.1.2 Indirect filter

Instead of using the total navigation state as the state in the filter, it is possible to implement the filter indirectly. By using the error dynamics of the INS as a state space model and the error as the state. The measurement for the filter is then the difference between the INS and the GPS readings. This means that there will only be a measurement for the filter at the time steps when the GPS has measurements.

According to Farrell & Barth (1999) there are a few major advantages with the indirect approach compared with the direct, even if the setup is not as intuitive.

- The covariance update equations need be implemented at only the lower rate of the GPS update.

- Since the filter is designed based on an error model, all model parameters can be properly defined in a stochastic sense.

- The responsiveness of the navigation system is determined primarily by the update rate of the INS system and the bandwidth of the inertial sensors.

- The Kalman filter, since it is estimating slowly varying error quantities, can be a low-bandwidth system to attenuate any high frequency error on the GPS aiding signal.

In sections 5.2.1 and 5.2.3 the extended Kalman filter and the unscented Kalman filter for indirect integration will be presented.

There is also an other dimension to the level of integration which is important to mention. This is often called tight vs loose integration, and includes several levels of integration and complexity.

5.1.3 Loosely coupled

In the loosely coupled approach the states that are calculated from the measurements used for integration (Farrell & Barth 1999). This can be position, attitude, velocity etc. The advantage of this approach is the simplicity and flexibility of the system. By using the states from the navigation the filter computations will also be less computationally intensive. The simplicity of the approach is paid for by reduced system performance.

5.1.4 Tightly coupled

In the tightly coupled system integration of the raw accelerometer, gyro, pseudo range and delta range measurements are used in the integration filter. Tightly coupled systems is also considered as more accurate than loosely coupled implementations, but this come at the cost of simplicity. The requirements to computer throughput are also increased since both GPS and INS software must be implemented in the navigation system. In the tightly coupled approach, data from the integrated navigation solutions are fed back to the GPS receiver to aid the carrier tracking loops.

In this section the integration of measurements have mainly been referenced to GPS and INS, but it is possible to analyze the integration between other measurements in the same way. This being DVL, compass and depth sensors.

5.2 Kalman filters

5.2.1 Kalman filter

This chapter will present the first observer that will be tested. The Kalman filter is presented in Brown & Hwang (1997).

By combining what is known about the sensor errors and their characteristics a best possible estimate is computed by a Kalman filer. The filter requires a mathematical state-space model (Balchen et al. 2004) of the system information.

$$\begin{aligned}\mathbf{x}_{k+1} =& \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{\Delta}_k \mathbf{u}_k + \mathbf{\Omega}_k \gamma_k \\ \mathbf{y}_k =& \mathbf{D}_k \mathbf{x}_k + \xi_k\end{aligned} \tag{5.1}$$

Where γ_k and ξ_k is process and measurement noise. Both are white noise processes and uncorrelated. The literature shows how the Kalman algorithm finds a unbiased and minimum variance state estimate. The filter algorithm is given by:

The filter is first initialized at k=0.

$$\begin{aligned}\bar{\mathbf{x}}_0 =& \mathbf{E}(\mathbf{x}_0) \\ \bar{\mathbf{X}}_0 =& \mathbf{E}[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T]\end{aligned} \tag{5.2}$$

Kalman gain matrix:

$$\mathbf{K}_k = \bar{\mathbf{X}}_k \mathbf{D}_k^T (\mathbf{D}_k \bar{\mathbf{X}}_k \mathbf{D}_k^T + \mathbf{R}_k)^{-1} \tag{5.3}$$

Using measurements to update state estimate:

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \mathbf{D}_k \bar{\mathbf{x}}_k) \tag{5.4}$$

Covariance matrix update given in Joseph form:

$$\hat{\mathbf{X}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{D}_k) \bar{\mathbf{X}}_k (\mathbf{I} - \mathbf{K}_k \mathbf{D}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \tag{5.5}$$

Prediction of state estimate and covariance:

$$\begin{aligned}\bar{\mathbf{x}}_{k+1} &= \mathbf{\Phi}_k \hat{\mathbf{x}}_k + \mathbf{\Delta}_k \mathbf{u}_k \\ \bar{\mathbf{X}}_{k+1} &= \mathbf{\Phi}_k \hat{\mathbf{X}}_k \mathbf{\Phi}_k^T + \mathbf{\Omega}_k \mathbf{Q}_k \mathbf{\Omega}_k^T\end{aligned} \tag{5.6}$$

The matrices $\mathbf{\Phi}_k, \mathbf{\Omega}_k, \mathbf{\Delta}_k$ and $\mathbf{D}_k$ is given in the state space representation of the system.

Joseph form
The covariance matrix in equation 5.5 can be calculated in a simpler form, which is given as:

$$\hat{\mathbf{X}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{D}_k)\bar{\mathbf{X}}_k \tag{5.7}$$

This formula does not guarantee that $\hat{\mathbf{X}}_k$ is symmetric and positive semidefinite. For that reason the Joseph form of the covariance update has been established (Minkler & Minkler 1993). The Joseph form guarantees positive semi defint and symmetric $\hat{\mathbf{X}}_k$ also on implementations when the computer has a limited number of decimals.

5.2.2 Extended Kalman Filter

One way of dealing with a nonlinear system in a Kalman filter setting, is to do some kind of linearization. This can either be done about some nominal trajectory, or by updating the linearization every time step based on the measurements. The first method is called linearized Kalman filter the latter is called extended Kalman filter. The derivation of these filters are described in Brown & Hwang (1997).

The extended Kalman filter has a statistical advantage over the regular linearized filter, because it uses a trajectory that has been updated instead of using the "old trajectory". This is only valid in a statistical view. There might be a chance that the trajectory computed as the nominal trajectory is a better estimate for a given time step. Because of the interconnection between the linearized trajectory, the estimate and back to the trajectory, one might risk for the filter to diverge in special situations. The extended Kalman filter will in general not give an minimum variance estimate of the state vector since it is based on the local linearization around the estimated process states.

The equations for extended Kalman filter can be summarized as (Henriksen 1998):

Filtering

$$\begin{aligned}
\hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + \mathbf{K}_k[\mathbf{y}_k - \mathbf{g}_k(\bar{\mathbf{x}}_k)] && (5.8)\\
\hat{\mathbf{X}}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{D}_k)\bar{\mathbf{X}}_k(\mathbf{I} - \mathbf{K}_k\mathbf{D}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T && (5.9)
\end{aligned}$$

Where

$$\mathbf{D}_k = \frac{\delta\mathbf{g}_k}{\delta\mathbf{x}_k^T}(\bar{\mathbf{x}}_k) \qquad \mathbf{K}_k = \bar{\mathbf{X}}_k\mathbf{D}_k^T(\mathbf{D}_k\bar{\mathbf{X}}_k\mathbf{D}_k^T + \mathbf{R}_k)^{-1} \tag{5.10}$$

Prediction

$$\begin{aligned}
\bar{\mathbf{x}}_{k+1} &= \mathbf{f}_k(\hat{\mathbf{x}}_k, \mathbf{u}_k) && (5.11)\\
\bar{\mathbf{X}}_{k+1} &= \mathbf{\Phi}_k\hat{\mathbf{X}}_k\mathbf{\Phi}_k^T + \mathbf{\Omega}_k\mathbf{Q}_k\mathbf{\Omega}_k^T && (5.12)
\end{aligned}$$

And

$$\mathbf{\Phi}_k = \frac{\delta \mathbf{f}_k}{\delta \mathbf{x}_k^T}(\hat{\mathbf{x}}_k, \mathbf{u}_k) \tag{5.13}$$

From the filter equations it can be seen that the difference in extended versus regular linearized Kalman filter is in the calculation of the measurement matrix $\mathbf{D}_k$ and the system matrix $\mathbf{\Phi}_k$.

Since the extended Kalman filter may diverge if the linearization is performed at a bad state, there are two things that should be thought of when starting the filter. First the extended Kalman filter should always use the Joseph form of the covariance matrix update. Second it is important that the filter is started with a good estimate of the initial state. This can be done by using an accurate reference to start with, this way the error state in the filter can be set to zero.

5.2.3 Unscented Kalman Filter

A part of this paper is to test the performance of the unscented Kalman filter (UKF) for integration of IMU measurements and external measurements (DVL and HiPAP). The main motivation for using the UKF instead of the EKF is that the EKF has some flaws. According to Wan & v. d. Merwe (2000) these are due the linearization the EKF uses in covariance estimation, and that the predictions are merely the nonlinear function applied to the prior estimate (eqs. 5.34 and 5.35). This linearization can lead to large errors in the posterior state estimate and it's covariance. According to LaViola (2003) the calculation of the Jacobians, which are linear approximations of the nonlinear function, can be very complex and cause implementation difficulties. These flaws lead to sub-optimal performance and sometimes divergence of the filter.

The main idea behind the UKF is an alternative way to determine the covariance matrix for the reasons mentioned above. The UKF was first proposed by Julier & Uhlmann (1997) where it was shown that the UKF yields the same performance as the Kalman filter for linear systems. Later it will be shown that the expected performance of the UKF is superior to that of the EKF for nonlinear systems.

UKF is based on the unscented Transform. The idea behind this transformation is presented in Uhlmann (1994) where it was stated that *it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation.* The transform is best represented in an algorithmic manner

The unscented transform

1. Of the L-dimensional random variable $\mathbf{x}$ create $2L+1$ weighted sigma points given by

$$\chi^0 = \hat{\mathbf{x}}(k|k) \tag{5.14}$$

$$\chi^i = \hat{\mathbf{x}}(k|k) + (\sqrt{(L+\lambda)\mathbf{X}(k|k)})_i, i = 1...L \tag{5.15}$$

$$\chi^i = \hat{\mathbf{x}}(k|k) - (\sqrt{(L+\lambda)\mathbf{X}(k|k)})_{i-L}, i = L+1...2L \tag{5.16}$$

   where $(\sqrt{(L+\lambda)\mathbf{X}(k|k)})_i$ is the ith row or column of the matrix square root of $(L+\lambda)\mathbf{X}(k|k)$. These points are distributed so that the sample mean and sample covariance corresponds to the real mean and covariance, $\hat{\mathbf{x}}(k|k)$ and $\mathbf{X}(k|k)$. The weights associated with each point is given by

$$W_s^0 = \tfrac{\lambda}{L+\lambda} \tag{5.17}$$

$$W_c^0 = \tfrac{\lambda}{L+\lambda} + (1-\alpha^2+\beta) \tag{5.18}$$

$$W_s^i = W_c^i = \tfrac{1}{2(L+\lambda)}, i = 1...2L \tag{5.19}$$

$$\lambda = \alpha^2(L+\kappa) - L \tag{5.20}$$

   α is a scaling factor that determines the spread of the sigma points, β is a parameter used to incorporate prior knowledge of about the distribution of $\mathbf{x}$ and κ is a secondary scaling factor.

2. Create the transformed sigma points by propagating them trough the nonlinear function

$$\varkappa^i = \mathbf{f}(\chi^i), i = 0...2L \tag{5.21}$$

3. The mean of the transformed variable is given by the weighted average of the transformed sigma points

$$\bar{\mathbf{x}} = \sum_{i=0}^{2L} W_s^i \varkappa^i \tag{5.22}$$

4. The covariance of the transformed variable is given by the weighted outer product of the transformed sigma points

$$\bar{\mathbf{X}}_k = \sum_{i=0}^{2L} W_c^i \left[\varkappa^i - \bar{\mathbf{x}}_k\right] \left[\varkappa^i - \bar{\mathbf{x}}_k\right]^T \tag{5.23}$$

Properties of the unscented transform

The unscented transform selects the sigma points such that the statistics of the transformed variable are captured accurately up to the second order. This is due to the dimension of χ which is $(2L+1)$ that ensures that the central moment of $\mathbf{x}$ is captured correctly up the second order (Angrisani et al. 2005). Equation 5.22 may be written as

$$\bar{\mathbf{x}} = W_s^0 \mathbf{f}(\bar{\mathbf{x}}) + \sum_{i=1}^{L} W_s^i \mathbf{f}(\bar{\mathbf{x}} + \mathbf{s_i}) + \sum_{i=L+1}^{2L} W_s^i \mathbf{f}(\bar{\mathbf{x}} - \mathbf{s}_{i-L}) \tag{5.24}$$

where s_i is the i^{th} column of $\sqrt{(L+\lambda)\mathbf{X}}$.

Julier & Uhlmann (1997) describes the problem of developing a consistent, efficient and unbiased transformation of a nonlinear function $\mathbf{y} = \mathbf{f}(\mathbf{x})$. To describe the problem a Taylor series expansion of the equation is done about the mean of the random variable $\mathbf{x}$, noted $\overline{\mathbf{x}}$.

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= \mathbf{f}(\overline{\mathbf{x}} + \delta\mathbf{x}) \\ &= \mathbf{f}(\overline{\mathbf{x}}) + \mathbf{D}_{\delta\mathbf{x}}\mathbf{f} + \frac{\mathbf{D}^{\mathbf{2}}_{\delta\mathbf{x}}\mathbf{f}}{2!} + \frac{\mathbf{D}^{\mathbf{3}}_{\delta\mathbf{x}}\mathbf{f}}{3!} + \frac{\mathbf{D}^{\mathbf{4}}_{\delta\mathbf{x}}\mathbf{f}}{4!} + \cdots \end{aligned} \tag{5.25}$$

where the $\mathbf{D}_{\delta\mathbf{x}}\mathbf{f}$ operator evaluates the total differential of the function $\mathbf{f}(\cdot)$ when pertubated around a nominal value $\overline{\mathbf{x}}$ by $\delta\mathbf{x}$. $\delta\mathbf{x}$ is a zero mean Gaussian variable with covariance $\mathbf{X}$. Julier et al. (2000) shows that taking the mean of $\mathbf{y}$, noted $\overline{\mathbf{y}}$, yields:

$$\begin{aligned} \overline{\mathbf{y}} &= E[\mathbf{f}(\overline{\mathbf{x}} + \delta\mathbf{x})] \\ &= \mathbf{f}(\overline{\mathbf{x}}) + E\left[\frac{\mathbf{D}^{\mathbf{2}}_{\delta\mathbf{x}}\mathbf{f}}{2!} + \frac{\mathbf{D}^{\mathbf{4}}_{\delta\mathbf{x}}\mathbf{f}}{4!} + \frac{\mathbf{D}^{\mathbf{6}}_{\delta\mathbf{x}}\mathbf{f}}{6!} + \cdots\right] \end{aligned} \tag{5.26}$$

Where it has been exploited that $\delta\mathbf{x}$ is symmetric, and the odd terms evaluate to zero.[1] The m^{th} element of the Taylor series can be written as (Julier et al. 2000)

$$E\left[\frac{\mathbf{D}^{m}_{\delta x}\mathbf{f}}{m!}\right] = E\left[\frac{1}{m!}\left(\sum_{n=1}^{L}\delta x_n \frac{\partial}{\partial x_n}\right)^m \mathbf{f}(\mathbf{x})\Bigg|_{\mathbf{x}=\bar{\mathbf{x}}}\right] = \frac{1}{m!}\left(\sum_{n=1}^{L}\mu_n \frac{\partial}{\partial x_n}\right)^m \mathbf{f}(\bar{\mathbf{x}}) \tag{5.27}$$

where μ_n^m is the m^{th} order central moment of δx_n. From this 5.26 can be written as

$$\overline{\mathbf{y}} = \mathbf{f}(\overline{\mathbf{x}}) + \frac{1}{2!}\left(\sum_{n=1}^{L}\mu_n \frac{\partial}{\partial x_n}\right)^2 \mathbf{f}(\overline{\mathbf{x}}) + \frac{1}{4!}\left(\sum_{n=1}^{L}\mu_n \frac{\partial}{\partial x_n}\right)^4 \mathbf{f}(\overline{\mathbf{x}}) + \cdots \tag{5.28}$$

[1] In Maybeck (1979) it is shown that $E[\delta x^k] = 0$ is valid for any odd k if δx is a zero-mean Gaussian random variable

Using Taylor expansion on 5.24 yields (Angrisani et al. 2005)

$$\begin{aligned}\bar{\mathbf{x}} = W_s^0\mathbf{f}(\bar{\mathbf{x}}) + \sum_{i=1}^{L} W_s^i \left(\mathbf{f}(\bar{\mathbf{x}}) + \mathbf{s}_i\frac{\partial\mathbf{f}(\bar{\mathbf{x}})}{\partial x_i} + \frac{s_i^2}{2!}\frac{\partial^2\mathbf{f}(\bar{\mathbf{x}})}{\partial x_i^2} + \cdots\right) + \\ \sum_{i=L+1}^{2L} W_s^i \left(\mathbf{f}(\bar{\mathbf{x}}) - s_{i-L}\frac{\partial\mathbf{f}(\bar{\mathbf{x}})}{\partial x_i} - \frac{s_i^2}{2!}\frac{\partial^2\mathbf{f}(\bar{\mathbf{x}})}{\partial x_i^2} + \cdots\right)\end{aligned} \tag{5.29}$$

which by exploiting how the sigma points and corresponding weights are chosen(Angrisani et al. 2005), equation 5.29 can be written as

$$\bar{\mathbf{x}} = \mathbf{f}(\bar{\mathbf{x}}) + \frac{1}{2!}\sum_{i=1}^{L}\mu_i^2\frac{\partial^2\mathbf{f}(\bar{\mathbf{x}})}{\partial x_i^2} + \cdots + \frac{1}{k!}\sum_{i=1}^{L}\mu_i^k\frac{\partial^k\mathbf{f}(\bar{\mathbf{x}})}{\partial x_i^k} + \cdots \tag{5.30}$$

where k is all even numbers. By comparing 5.28 with 5.30 it can be seen that the unscented transform capture the properties of $\bar{\mathbf{x}}$ correctly up to the second order. One condition for this to be true is that the unscented transform need to calculate the correct value of the second order central moment μ^2. This is ensured by choosing the α parameter such that $L + \lambda = 3$ (for Gaussian distributions) (Julier et al. 2000). The higher order terms of μ^k is not calculated correct with this value of $L+\lambda$. These higer order terms are assumed to have less impact, and therefore the error in these terms are neglectable.

The covariance of $\mathbf{y}$ is $\mathbf{Y} = E[(\mathbf{y}-\bar{\mathbf{y}})(\mathbf{y}-\bar{\mathbf{y}})^T]$. Using 5.25 and 5.26 the covariance is:

$$\begin{aligned}\mathbf{Y} =& \Delta\mathbf{f}_x\mathbf{X}\Delta\mathbf{f}_x^T + E\left[\frac{\mathbf{D_{\delta x}f(D^3_{\delta x}f)^T}}{3!} + \frac{\mathbf{D^2_{\delta x}f(D^2_{\delta x}f)^T}}{2! * 2!} + \frac{\mathbf{D^3_{\delta x}f(D_{\delta x}f)^T}}{3!}\right] \\ &- E\left[\frac{\mathbf{D^2_{\delta x}f}}{2}\right]E\left[\frac{\mathbf{D^2_{\delta x}f}}{2}\right]^T + \cdots\end{aligned} \tag{5.31}$$

where $E[\mathbf{D^2_{\delta x}f}(\mathbf{D^2_{\delta x}f})^T] = \Delta\mathbf{f}_x\mathbf{X}\Delta\mathbf{f}_x^T$ is used. Julier & Uhlmann (1997) argues that

> since each term in the series is scaled by a progressively smaller and smaller term, the lowest order terms in the series are likely to have the greatest impact.

By applying the same analysis as above, Julier et al. (2000) shows that the unscented filter captures the properties of the covariance correct up to the second order.

Recall that in EKF the a posteriori estimate of the state $\hat{\mathbf{x}}_k$ and covariance matrix $\hat{\mathbf{X}}_k$ are found from

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}(\mathbf{y_k} - g(\bar{\mathbf{x}}_k)) \tag{5.32}$$

$$\hat{\mathbf{X}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{D}_k)\bar{\mathbf{X}}_k \tag{5.33}$$

And the a priori estimate of the state $\hat{x}_k$ and covariance matrix $\bar{X}_k$ from

$$\begin{aligned} \bar{\mathbf{x}}_{k+1} &= \mathbf{f}(\hat{\mathbf{x}}_k) && (5.34) \\ \bar{\mathbf{X}}_{k+1} &= \mathbf{\Phi}_k \hat{\mathbf{X}}_k \mathbf{\Phi}_k^T + \mathbf{\Omega}_k \mathbf{Q}_k \mathbf{\Omega}_k^T && (5.35) \end{aligned}$$

where

$$\begin{aligned} \mathbf{\Phi}_k &= \frac{\delta \mathbf{f}_k}{\delta \mathbf{x}_k^T}(\hat{\mathbf{x}}_k, \mathbf{u}_k) && (5.36) \\ \mathbf{D}_k &= \frac{\delta \mathbf{g}_k}{\delta \mathbf{x}_k^T}(\bar{\mathbf{x}}_k) && (5.37) \end{aligned}$$

where $\mathbf{f}_k$ is the nonlinear system and $\mathbf{g}_k$ is the nonlinear measurement function.

By comparing 5.31 with 5.35 and 5.26 with 5.28 it can be seen that the EKF approximations are accurate only if the second and higher order terms in the mean and fourth and higher order terms in the covariance are negligible(Julier et al. 2000). This linearization can in many practical situations inflict significant errors. Julier & Uhlmann (1997) shows that this is significant in problems where one must transform information between polar and Cartesian coordinate systems.

The uncertainty the linearization introduces can be compensated for by introducing what Julier & Uhlmann (1997) calls stabilizing noise. This means that the process noise matrix is increased, this in turn increase the transformed covariance. This is not a desirable solution since the estimate remains biased. There is no general guarantee that the transformed estimate remains consistent or efficient.

To illustrate the improved performance by the UKF compared to the EKF Wan & v. d. Merwe (2000) shows a simple example for 2-dimensional system given in figure 5.1.

UKF algorithm

As with the ordinary KF also the UKF utilize a prediction -correction manner. The algorithm for the UKF is as follows (LaViola 2003):

Prediction

- Calculate sigma points and weights as in eqs 5.14-5.20

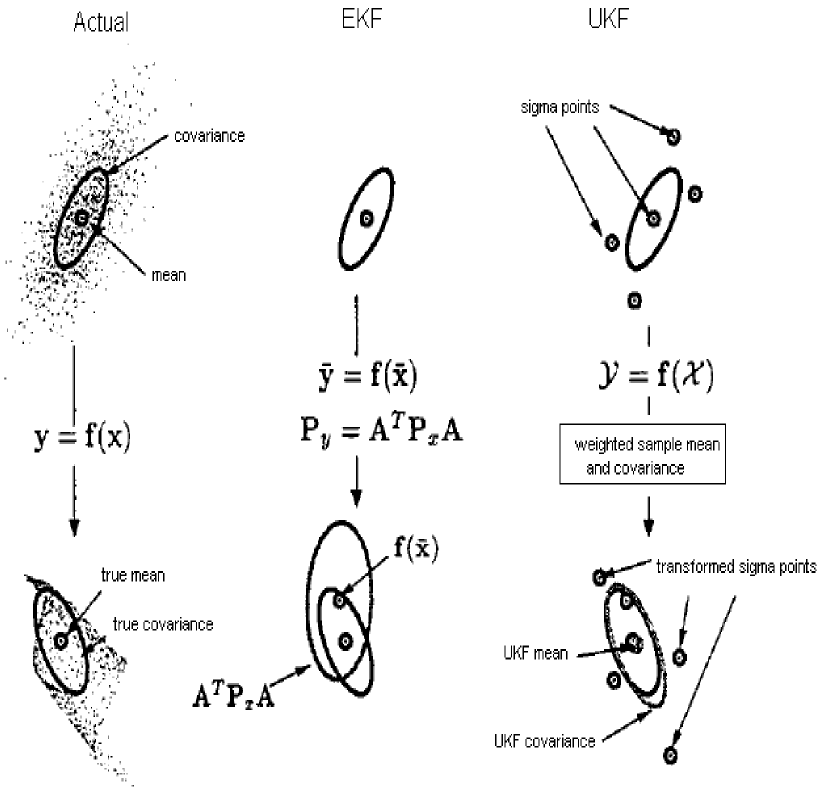

Figure 5.1: Example of the mean and covariance in EKF and UKF (Wan & v. d. Merwe 2000)

- Predict the state and covariance as

$$\begin{aligned}
\chi_k^i &= \mathbf{f}(\chi_{k-1}^i), i = 0...2L && (5.38)\\
\bar{\mathbf{x}}_k &= \sum_{i=0}^{2L} W_s^i \chi_{k-1}^i && (5.39)\\
\bar{\mathbf{X}}_k &= \sum_{i=0}^{2L} W_c^i \left[\chi_k^i - \bar{\mathbf{x}}_k\right]\left[\chi_k^i - \bar{\mathbf{x}}_k\right]^T + \mathbf{Q}_k && (5.40)
\end{aligned}$$

- Predict the measurement as

$$\begin{aligned}
\gamma_k^i &= \mathbf{g}(\chi_k^i), i = 0...2L && (5.41)\\
\bar{\mathbf{z}}_k &= \sum_{i=0}^{2L} W_s^i \gamma_k^i && (5.42)
\end{aligned}$$

Correction

- Make corrections to state and covariance estimate

$$\mathbf{X}^{-1}_{\bar{\mathbf{z}}_k\bar{\mathbf{z}}_k} = \sum_{i=0}^{2L} W_c^i \left[\gamma_k^i - \bar{\mathbf{z}}_k\right]\left[\gamma_k^i - \bar{\mathbf{z}}_k\right]^T + \mathbf{R}_k \tag{5.43}$$

$$\mathbf{X}_{\hat{\mathbf{x}}_k\bar{\mathbf{z}}_k} = \sum_{i=0}^{2L} W_c^i \left[\chi_k^i - \bar{\mathbf{x}}_k\right]\left[\gamma_k^i - \bar{\mathbf{z}}_k\right]^T \tag{5.44}$$

$$\mathbf{K}_k = \mathbf{X}_{\hat{\mathbf{x}}_k\bar{\mathbf{z}}_k}\mathbf{X}^{-1}_{\bar{\mathbf{z}}_k\bar{\mathbf{z}}_k} \tag{5.45}$$

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{z}_k - \bar{\mathbf{z}}_k) \tag{5.46}$$

$$\hat{\mathbf{X}}_k = \bar{\mathbf{X}}_k - \mathbf{K}_k\mathbf{X}_{\bar{\mathbf{z}}_k\bar{\mathbf{z}}_k}\mathbf{K}_k^T \tag{5.47}$$

$\mathbf{K}_k$ is the Kalman gain matrix. $\mathbf{R}_k$ and $\mathbf{Q}_k$ are respectively the measurement noise covariance matrix and the process noise covariance matrix. This algorithm only uses standard vector and matrix calculations. Assuming that $\hat{\mathbf{X}}_k$ is always real and positive definite the well known and numerically efficient Cholesky decomposition (section 2.2.5) can and should be used to find the matrix square root (Julier et al. 2000).

5.3 Nonlinear integration

As stated in section 5.1 creating reliable measurements from different sensors and measurements may be done either by estimating the states of the system directly(direct method) or by estimating the error development in the states (indirect method). Here a nonlinear observer for direct integration will be presented.

The traditional way of implementing an observer is to make a model of the dynamics of the real system, in this case an ROV/AUV, and use this model to estimate the states of the system. As may be seen from the equations of motion for an Autonomous Underwater Vehicle(AUV) given in Healey & Lienard (1993), an AUV is a highly nonlinear system with complex dynamics. I order to obtain a good mathematical model of the vessel an extensive series of tests is required to determine the hydrodynamic coefficients needed in the model. This is very time consuming and it is difficult to get accurate results. In addition it often introduces a lower level of accuracy because of model uncertainties (Gade & Jalving 1998). Another disadvantage of this method is that the model is vessel specific, which means that the model obtained for one vessel can not be used for a different type of vessel.

Instead of modeling the vessel dynamics an alternative is to model the dynamics of the INS, and estimate the states of the INS directly. These states are in every essence the same as the states of the vessel, i.e. position, velocity and attitude since the INS is exposed to the same motions as the real vessel. When modeling the INS dynamics the same configuration of measurement apparatus may be used in many different types of vessels and the characteristics of the INS will be the same. Therefore the same mathematical model may be re-used. From equations 3.4, 3.9 and 3.11 it can be seen that the dynamics of the INS is nonlinear. One option is to use a Kalman Filter (EKF or UKF), as presented in sections 5.2.2 and 5.2.3, or to make a nonlinear observer.

5.3.1 Nonlinear observer

Given the real system on the form

$$\begin{aligned} \dot{x} =& Ax + g(y,u) \\ y =& Cx \end{aligned} \tag{5.48}$$

an observer such as

$$\dot{\hat{x}} = A\hat{x} + g(y,u) + H(y - C\hat{x}) \tag{5.49}$$

may be designed. From this it may been seen that the error between the real and estimated states is

$$\dot{\tilde{x}} = (A - HC)\tilde{x} \tag{5.50}$$

It can be seen that the error dynamics are a pure linear system, hence designing the matrix H such that $A - HC$ is Hurwitz will result in asymptotic convergence of the error $\tilde{x}$ (Khalil 2002). However, in order for this to work we need to assume that $g(y,u)$ is perfectly known. If this is not the case the error will become

$$\dot{\tilde{x}} = (A - HC)\tilde{x} + g(y,u) - g_0(y,u) \tag{5.51}$$

where g_0 is a model of g. Because of the added error term $g - g_0$ it is not obvious that a Hurwitz $A - HC$ will guarantee convergence of the error. There exists many design techniques for designing nonlinear observers, one well known way to ensure convergence of the error dynamics is to design the observer such that a Lyapunov analysis (Khalil 2002) may be used to prove stability of the error dynamics.

5.3.2 Nonlinear observer for GPS and INS integration

Vik & Fossen (2001) suggested a nonlinear observer for direct GPS and INS integration. In Vik & Fossen (2001) globally exponentially stability(GES) of the error dynamics of the observer was proven by using Lyapunov theory. Given the strap down navigation equations from chapter 3 for attitude, velocity and position:

$$\begin{aligned}
\dot{\mathbf{q}}_{nb} &= \frac{1}{2}\mathbf{q}_{nb} \otimes \begin{bmatrix} o \\ \omega_{ib}^{b} \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 0 \\ \omega_{in}^{n} \end{bmatrix} \otimes \mathbf{q}_{nb} && (5.52)\\
&= \frac{1}{2}\begin{bmatrix} -\epsilon^{T} \\ \eta\mathbf{I} + S(\epsilon) \end{bmatrix}\omega_{ib}^{b} - \frac{1}{2}\begin{bmatrix} -\epsilon^{T} \\ \eta\mathbf{I} + S(\epsilon) \end{bmatrix}\omega_{in}^{n} && (5.53)\\
\dot{\mathbf{v}}_{eb}^{n} &= \mathbf{R}_{b}^{n}\mathbf{f}_{ib}^{b} + \bar{\mathbf{g}}_{eb}^{n} - (2S(\omega_{ie}^{n}) + S(\omega_{en}^{n}))\mathbf{v}_{eb}^{n} && (5.54)\\
\dot{\mathbf{p}}_{en} &= \mathbf{R}_{n}^{e}\mathbf{v}_{eb}^{n} && (5.55)
\end{aligned}$$

where $\mathbf{p}_{en}$ are the Cartesian ECEF coordinates and represent position. Longitude, latitude and height can easily be calculated from $\mathbf{p}_{en}$. The IMU error models are given in equations 4.9- 4.12 as

$$\begin{aligned}
\mathbf{f}_{IMU}^{b} &= \mathbf{f}_{ib}^{b} - \mathbf{b}_{f_{IMU}} + \xi_{f} && (5.56)\\
\dot{\mathbf{b}}_{f_{IMU}} &= -\mathbf{T}_{f_{IMU}}^{-1}\mathbf{b}_{f_{IMU}} + \gamma_{f_{IMU}} && (5.57)\\
\omega_{IMU}^{b} &= \omega_{ib}^{b} - \beta_{\omega_{IMU}} + \xi_{\omega} && (5.58)\\
\dot{\beta}_{\omega_{IMU}} &= -\mathbf{T}_{\omega_{IMU}}^{-1}\beta_{\omega_{IMU}} + \gamma_{\omega_{IMU}} && (5.59)
\end{aligned}$$

Vik (2000) proposes the following attitude observer:

$$\begin{aligned}
\dot{\hat{\mathbf{q}}} &= \frac{1}{2}\begin{bmatrix} -\hat{\epsilon}^{T} \\ \hat{\eta}\mathbf{I} + S(\hat{\epsilon}) \end{bmatrix}\left[\omega_{IMU} + \hat{\beta}_{\omega_{IMU}} + \mathbf{K}_{1}\tilde{\epsilon}sgn(\tilde{\eta})\right] && (5.60)\\
&\quad -\frac{1}{2}\begin{bmatrix} -\hat{\epsilon}^{T} \\ \hat{\eta}\mathbf{I} + S(\hat{\epsilon}) \end{bmatrix}\omega_{in}^{n}\\
\dot{\hat{\beta}}_{\omega_{IMU}} &= -\mathbf{T}_{\omega_{IMU}}^{-1}\hat{\beta}_{\omega_{IMU}} + \frac{1}{2}\mathbf{K}_{2}\tilde{\epsilon}sgn(\tilde{\eta}) && (5.61)
\end{aligned}$$

and the following velocity and position observer

$$\begin{aligned}
\dot{\hat{\mathbf{v}}}_{eb}^{n} &= \hat{\mathbf{R}}_{b}^{n}\left[\mathbf{f}_{imu}^{b} + \hat{\mathbf{b}}_{f_{IMU}}\right] + \hat{\mathbf{g}}_{eb}^{n} - \left[2S(\omega_{ie}^{n}) + S(\omega_{en}^{n})\right]\hat{\mathbf{v}}_{eb}^{n} && (5.62)\\
&\quad +\mathbf{K}_{3}\tilde{\mathbf{v}}_{eb}^{n} + (\hat{\mathbf{R}}_{e}^{n})^{T}\tilde{\mathbf{p}}_{en}\\
\dot{\hat{\mathbf{p}}}_{en} &= \mathbf{R}_{n}^{e}\hat{\mathbf{v}}_{eb}^{n} + \mathbf{K}_{4}\tilde{\mathbf{p}}_{en} && (5.63)\\
\dot{\hat{\mathbf{b}}}_{f_{IMU}} &= -\mathbf{T}_{f_{IMU}}^{-1}\hat{\mathbf{b}}_{f_{IMU}} + \mathbf{K}_{5}(\hat{\mathbf{R}}_{b}^{n})^{T}\tilde{\mathbf{v}}_{eb}^{n} && (5.64)
\end{aligned}$$

where $K1...K5$ are positive definite gain matrices. This is a simplified version not including IMU misalignment terms and GPS clock errors. Since we are dealing with a submerged vehicle GPS alone can not be used. It must be combined with a HiPAP (see section 4.3.2). For this reason a term to include the HiPAP bias error is needed. Since the magnitude of the HPR error is much larger than the GPS error (see chapter 4) this is the dominating error term, and we only include the estimate of HiPAP error. This is modeled as

$$\begin{aligned}
\mathbf{p}_{en_{HiPAP}} &= \mathbf{p}_{en} - \mathbf{b}_{p_{HiPAP}} + \xi_{HiPAP} && (5.65)\\
\dot{\mathbf{b}}_{p_{HiPAP}} &= -\mathbf{T}_{b_{HiPAP}}^{-1}\mathbf{b}_{p_{HiPAP}} + \gamma_{HiPAP} && (5.66)
\end{aligned}$$

The HiPAP bias enters the system through the HiPAP position measurement:

$$\mathbf{p}_{en_{HiPAP}} = \mathbf{p}_{en_{real}} + \mathbf{b}_{p_{HiPAP}} \tag{5.67}$$

The resulting velocity and position observer which includes a HiPAP bias error term is somewhat different from the one proposed by Vik & Fossen (2001). The new observer is designed by means of Lyapunov stability. The proposed observer is

$$\begin{aligned}
\dot{\hat{\mathbf{v}}}^n_{eb} &= \hat{\mathbf{R}}^n_b \left[\mathbf{f}^b_{imu} + \hat{\mathbf{b}}_{f_{IMU}}\right] + \bar{\mathbf{g}}^n_{eb} - \left[2S(\omega^n_{ie}) + S(\omega^n_{en})\right]\hat{\mathbf{v}}^n_{eb} \qquad (5.68)\\
&\quad + \mathbf{K_3}\tilde{\mathbf{v}}^n_{eb} + (\hat{\mathbf{R}}^n_e)^T(\tilde{\mathbf{p}}_{en_{HiPAP}} - \hat{\mathbf{b}}_{p_{HiPAP}})\\
\dot{\hat{\mathbf{p}}}_{en} &= \mathbf{R}^e_n(\hat{\mathbf{v}}^n_{eb} + \tilde{\mathbf{v}}^n_{eb}) + \mathbf{K_4}(\tilde{\mathbf{p}}_{en_{HiPAP}} - \hat{\mathbf{b}}_{p_{HiPAP}}) \qquad (5.69)\\
\dot{\hat{\mathbf{b}}}_{f_{IMU}} &= -\mathbf{T}^{-1}_{f_{IMU}}\hat{\mathbf{b}}_{f_{IMU}} + \mathbf{K}_5(\hat{\mathbf{R}}^n_b)^T\tilde{\mathbf{v}}^n_{eb} \qquad (5.70)\\
\dot{\hat{\mathbf{b}}}_{p_{HiPAP}} &= -\mathbf{T}^{-1}_{b_{HiPAP}}\hat{\mathbf{b}}_{p_{HiPAP}} - \mathbf{K}_6(\mathbf{R}^n_e\tilde{\mathbf{v}}^n_{eb} \qquad (5.71)\\
&\quad + \mathbf{K}^T_4(\tilde{\mathbf{p}}_{en_{HiPAP}} - \hat{\mathbf{b}}_{p_{HiPAP}}))
\end{aligned}$$

5.3.3 Stability Analysis

Theorem 4.2 in Khalil (2002) states:
Let $x = 0$ be an equilibrium point for $\dot{x} = f(x)$. Let $V : R^n \rightarrow R$ be a continously differentiable function such that

$$\begin{aligned}
V(0) = 0 \text{ and } V(x) > 0, \forall x \neq 0 \qquad (5.72)\\
||x|| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty \qquad (5.73)\\
\dot{V}(x) < 0, \forall x \neq 0 \qquad (5.74)
\end{aligned}$$

then $x = 0$ is globally asymptotically stable.

Attitude observer

The proposed attitude observer gives the following error dynamics for the attitude (Vik & Fossen 2001):

$$\begin{aligned}
\begin{bmatrix}\dot{\tilde{\eta}}\\ \dot{\tilde{\epsilon}}\end{bmatrix} &= \frac{1}{2}\begin{bmatrix}-\tilde{\epsilon}^T\\ \tilde{\eta}\mathbf{I} + \mathbf{S}(\tilde{\epsilon})\end{bmatrix}\left[\tilde{\beta}_{\omega_{IMU}} - \mathbf{K_1}\tilde{\epsilon}sgn(\tilde{\eta})\right] \qquad (5.75)\\
\dot{\tilde{\beta}}_{\omega_{IMU}} &= -\mathbf{T}^{-\mathbf{1}}_{\omega_{\mathbf{IMU}}}\tilde{\beta}_{\omega_{\mathbf{IMU}}} - \frac{1}{2}\mathbf{K_2}\tilde{\epsilon}sgn(\tilde{\eta}) \qquad (5.76)
\end{aligned}$$

To prove stability of eqs 5.76-5.75 Vik & Fossen (2001) proposes the following Lyapunov function candidate:

$$V_1 = \tilde{\beta}^T_{\omega_{IMU}}\mathbf{K_2}\tilde{\beta}_{\omega_{IMU}} + \begin{cases}(\tilde{\eta} - 1)^2 + \tilde{\epsilon}^T\epsilon, \tilde{\eta} \geq 0\\ (\tilde{\eta} + 1)^2 + \tilde{\epsilon}^T\tilde{\epsilon}, \tilde{\eta} < 0\end{cases} \tag{5.77}$$

Taking the derivative of $\mathbf{V}_1$ along the trajectories of the system yields

$$\begin{aligned} \dot{\mathbf{V}}_1 &= 2\tilde{\beta}^T_{\omega_{IMU}} \mathbf{K_2} \dot{\tilde{\beta}}_{\omega_{IMU}} + \begin{cases} -2\dot{\tilde{\eta}}, \tilde{\eta} \geq 0 \\ 2\dot{\tilde{\eta}}, \tilde{\eta} < 0 \end{cases} && (5.78) \\ &= -2\tilde{\beta}^{\mathbf{T}}_{\omega_{\mathbf{IMU}}} \mathbf{T}^{\mathbf{-1}}_{\omega_{\mathbf{IMU}}} \mathbf{K_2} \tilde{\beta}_{\omega_{\mathbf{IMU}}} - \tilde{\epsilon}^{\mathbf{T}} \mathbf{K_1} \tilde{\epsilon} \leq 0 && (5.79) \end{aligned}$$

It has been used that $\tilde{\eta}\dot{\tilde{\eta}} + \tilde{\epsilon}^T \dot{\tilde{\epsilon}} = 0$. With the constraint of unity quaternion $\tilde{\eta}^2 + \tilde{\epsilon}^T \tilde{\epsilon} = 1$, $\dot{V}_1$ becomes strictly negative (Vik 2000).

Position and velocity observer

The proposed observer gives the following error dynamics:

$$\begin{aligned} \dot{\tilde{\mathbf{v}}}^n_{eb} &= \hat{\mathbf{R}}^n_b \tilde{\mathbf{b}}_{\mathbf{f}_{IMU}} + (\mathbf{R}^n_b - \hat{\mathbf{R}}^n_b)\mathbf{f} - [2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en})]\, \tilde{\mathbf{v}}^n_{eb} && (5.80) \\ &\quad -\mathbf{K}_3 \tilde{\mathbf{v}}^n_{eb} - (\mathbf{R}^n_e)^T (\tilde{\mathbf{p}}_{en} + \tilde{\mathbf{b}}_{p_{HiPAP}}) \\ \dot{\tilde{\mathbf{p}}}_{en} &= -\mathbf{K}_4 (\tilde{\mathbf{p}}_{en} + \tilde{\mathbf{b}}_{p_{HiPAP}}) && (5.81) \\ \dot{\tilde{\mathbf{b}}}_{f_{IMU}} &= -\mathbf{T}^{-1}_{f_{IMU}} \tilde{\mathbf{b}}_{f_{IMU}} - \mathbf{K}_5 (\hat{\mathbf{R}}^n_b)^T \tilde{\mathbf{v}}^n_{eb} && (5.82) \\ \dot{\tilde{\mathbf{b}}}_{p_{HiPAP}} &= -\mathbf{T}^{-1}_{b_{HiPAP}} \tilde{\mathbf{b}}_{p_{HiPAP}} + \mathbf{K}_6 (R^n_e \tilde{\mathbf{v}}^n_{eb} + \mathbf{K}^T_4 (\tilde{\mathbf{p}}_{en} + \tilde{\mathbf{b}}_{p_{HiPAP}})) && (5.83) \end{aligned}$$

where $\mathbf{f} = \mathbf{f}_{IMU} + \mathbf{b}_{f_{IMU}}$.

From equation 2.29 we have

$$\tilde{\mathbf{R}}^n_b = \mathbf{I} + 2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + 2\mathbf{S}^2(\tilde{\epsilon}) \tag{5.84}$$

We can then rewrite $(\mathbf{R}^n_b - \hat{\mathbf{R}}^n_b)\mathbf{f}$ as

$$\begin{aligned} (\mathbf{R}^n_b - \hat{\mathbf{R}}^n_b)\mathbf{f} =& \hat{\mathbf{R}}^n_b (\tilde{\mathbf{R}}^n_b - \mathbf{I})\mathbf{f} \\ =& -2\hat{\mathbf{R}}^n_b \left[\tilde{\eta}\mathbf{I} + \mathbf{S}(\tilde{\epsilon})\right] \mathbf{S}(\mathbf{f})\tilde{\epsilon} \\ =& -\mathbf{E}(\hat{\mathbf{q}}, \tilde{q}, \mathbf{f})\tilde{\epsilon} \end{aligned} \tag{5.85}$$

To prove global asymptotic stability of the error dynamics (eqs 5.80-5.83) the following Lypaunov function is chosen:

$$\begin{aligned} \mathbf{V}_2 =& \frac{1}{2}(\tilde{\mathbf{v}}^n_{eb})^T \tilde{\mathbf{v}}^n_{eb} + \frac{1}{2}\tilde{\mathbf{p}}^T_{en}\tilde{\mathbf{p}}_{en} + \frac{1}{2}\tilde{\mathbf{b}}^T_{f_{IMU}} \mathbf{K}^{-1}_5 \tilde{\mathbf{b}}_{f_{IMU}} \\ &+ \frac{1}{2}\tilde{\mathbf{b}}^T_{p_{HiPAP}} \mathbf{K}^{-1}_6 \tilde{\mathbf{b}}_{p_{HiPAP}} > 0, \forall \mathbf{x} \neq 0 \end{aligned} \tag{5.86}$$

The derivative of $\mathbf{V}_2$ along the trajectories of the system is:

$$\begin{aligned}\dot{\mathbf{V}}_2 =&(\tilde{\mathbf{v}}^n_{eb})^T\dot{\tilde{\mathbf{v}}}^n_{eb} + \tilde{\mathbf{p}}^T_{en}\dot{\tilde{\mathbf{p}}}_{en} + \tilde{\mathbf{b}}^T_{f_{IMU}}\mathbf{K}_5^{-1}\dot{\tilde{\mathbf{b}}}_{f_{IMU}} + \tilde{\mathbf{b}}^T_{p_{HiPAP}}\mathbf{K}_6^{-1}\dot{\tilde{\mathbf{b}}}_{p_{HiPAP}}\\
=&(\tilde{\mathbf{v}}^n_{eb})^T\left[\hat{\mathbf{R}}^n_b\tilde{\mathbf{b}}_{f_{IMU}} - \left[2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en})\right]\tilde{\mathbf{v}}^n_{eb}\right]\\
&-(\tilde{\mathbf{v}}^n_{eb})^T\left[\mathbf{K}_3\tilde{\mathbf{v}}^n_{eb} + \mathbf{E}(\hat{\mathbf{q}},\tilde{\mathbf{q}},\mathbf{f})\tilde{\epsilon}\right] - (\tilde{\mathbf{v}}^n_{eb})^T(\mathbf{R}^e_n)^T\left[\tilde{\mathbf{p}}_{en} + \tilde{\mathbf{b}}_{p_{HiPAP}}\right]\\
&-\tilde{\mathbf{p}}^T_{en}\left[-\mathbf{K}_4(\tilde{\mathbf{p}}_{en} + \tilde{\mathbf{b}}_{p_{HiPAP}})\right] - \tilde{\mathbf{b}}^T_{f_{IMU}}\left[\mathbf{K}_5^{-1}\mathbf{T}^{-1}_{f_{IMU}}\tilde{\mathbf{b}}_{f_{IMU}} + (\hat{\mathbf{R}}^n_b)^T\tilde{\mathbf{v}}^n_{eb}\right]\\
&-\tilde{\mathbf{b}}^T_{p_{HiPAP}}\left[\mathbf{K}_6^{-1}\mathbf{T}^{-1}_{b_{HiPAP}}\tilde{\mathbf{b}}_{p_{HiPAP}} - \mathbf{R}^e_n\tilde{\mathbf{v}}^n_{eb} - \mathbf{K}_4^T(\tilde{\mathbf{p}}_{en} + \tilde{\mathbf{b}}_{p_{HiPAP}})\right]\\
=&-(\tilde{\mathbf{v}}^n_{eb})^T\mathbf{K}_3\tilde{\mathbf{v}}^n_{eb} - (\tilde{\mathbf{v}}^n_{eb})^T\mathbf{E}(\hat{\mathbf{q}},\tilde{\mathbf{q}},\mathbf{f})\tilde{\epsilon} - \tilde{\mathbf{p}}^T_{en}\mathbf{K}_6\tilde{\mathbf{p}}_{en} - \tilde{\mathbf{b}}^T_{f_{IMU}}\mathbf{K}_5^{-1}\mathbf{T}^{-1}_{f_{IMU}}\tilde{\mathbf{b}}_{f_{IMU}}\\
&-\tilde{\mathbf{b}}^T_{p_{HiPAP}}(\mathbf{K}_6^{-1}\mathbf{T}^{-1}_{b_{HiPAP}} - \mathbf{K}_4^T)\tilde{\mathbf{b}}_{p_{HiPAP}} - (\tilde{\mathbf{v}}^n_{eb})^T(\mathbf{R}^e_n)^T\tilde{\mathbf{p}}_{en}\\
=&-\beta(\mathbf{x}) < 0, \forall\mathbf{x} \neq 0\end{aligned} \tag{5.87}$$

where $\mathbf{x} = \begin{bmatrix}\tilde{\mathbf{v}}^n_{eb} & \tilde{\mathbf{p}}_{en} & \tilde{\mathbf{b}}_{f_{IMU}} & \tilde{\mathbf{b}}_{p_{HiPAP}} & \tilde{\epsilon}\end{bmatrix}^T$. To investigate the stability of the total system we use $\mathbf{V} = \mathbf{V}_1 + \mathbf{V}_2$ which is GAS if the following conditions are satisfied:

1. $$\begin{bmatrix}\tilde{\mathbf{v}}^n_{eb}\\ \tilde{\epsilon}\\ \tilde{\mathbf{p}}_{en}\end{bmatrix}^T \begin{bmatrix}\mathbf{K}_3 & \frac{1}{2}\mathbf{E}(\hat{\mathbf{q}},\tilde{\mathbf{q}},\mathbf{f}) & \frac{1}{2}(\mathbf{R}^e_n)^T\\ \frac{1}{2}\mathbf{E}^T(\hat{\mathbf{q}},\tilde{\mathbf{q}},\mathbf{f}) & \mathbf{K}_1 & 0\\ \frac{1}{2}\mathbf{R}^e_n & 0 & \mathbf{K}_6\end{bmatrix} \begin{bmatrix}\tilde{\mathbf{v}}^n_{eb}\\ \tilde{\epsilon}\\ \tilde{\mathbf{p}}_{en}\end{bmatrix} > 0 \tag{5.88}$$

   Which gives the following criteria:

   $$\mathbf{K}_3 > 0 \tag{5.89}$$

   $$\mathbf{K}_3\mathbf{K}_1 - \frac{1}{4}\mathbf{E}^T(\hat{\mathbf{q}},\tilde{\mathbf{q}},\mathbf{f})\mathbf{E}(\hat{\mathbf{q}},\tilde{\mathbf{q}},\mathbf{f}) > 0 \tag{5.90}$$

   $$\mathbf{K}_3\mathbf{K}_1\mathbf{K}_6 - \frac{1}{4}\mathbf{E}^T(\hat{\mathbf{q}},\tilde{\mathbf{q}},\mathbf{f})\mathbf{E}(\hat{\mathbf{q}},\tilde{\mathbf{q}},\mathbf{f})\mathbf{K}_6 - \frac{1}{4}\mathbf{R}^e_n\mathbf{K}_1(\mathbf{R}^e_n)^T > 0 \tag{5.91}$$

   Vik (2000) showed that since 5.90 must be valid for all $\tilde{\epsilon}$ it can be written as:

   $$\mathbf{K}_3\mathbf{K}_1 - \mathbf{S}^T(\mathbf{f}_{max})\mathbf{S}(\mathbf{f}_{max}) > 0 \tag{5.92}$$

   Using this 5.91 can writen as:

   $$(\mathbf{K}_3\mathbf{K}_1 - \mathbf{S}^T(\mathbf{f}_{max})\mathbf{S}(\mathbf{f}_{max}))\mathbf{K}_6 - \frac{1}{4}\mathbf{R}^e_n\mathbf{K}_1(\mathbf{R}^e_n)^T > 0 \tag{5.93}$$

   and is satisfied by choosing $\mathbf{K}_3$ positive definite and large so that the last term of 5.93 is dominated. 5.93 is also satisfied if $\mathbf{K}_6$ is positive definite and large, but $\mathbf{K}_6$ is also limited by the second condition in 5.94.

2. A second condition to ensure that the observer is GES is

   $$\mathbf{K}_6^{-1} > \mathbf{K}_4^T\mathbf{T}_{b_{HiPAP}} \tag{5.94}$$

Chapter 6

Kalman Filter design

The goal of the integrating observers is to provide an estimate of the total state (position, velocity and attitude) for a submerged vehicle (ROV or AUV). This is achieved by having a set of supporting instruments that measures the same states as the inertial navigation system (INS), this gives redundancy in the measurements which are combined to make one consistent estimate of the states.

In an ROV/AUV system there will be a position reference given by a HiPAP system with either reference to a ship with a DGPS, or to a fixed transponder network at the seabed. This position measurement is given at a slower data rate then the IMU measurement and is limited by the hydro acoustic link and the speed of sound in water.

The ROV/AUV will also have a Doppler velocity log, which gives the velocity of the vessel in reference to the seabed. The DVL can also give a measure of the velocity of the vessel in reference to the surrounding water. This would be useful if there existed an accurate estimate of the sea current velocity and heading surrounding the vessel, but it is assumed that such an estimate does not exist. Another feature of the DVL is that it contains a tilt sensor for measuring roll (ϕ) and pitch (θ), combined with a compass this gives a measurement of the full attitude state.

The depth is accurately measured with a pressure sensor on the vessel. By knowing the density of sea water and the atmospheric pressure at the surface, the depth is calculated.

The manufacturers of each instrument provide some specifications on how the error in the measurements behave. This allows us to model the errors in the different sensors as done in chapter 4. By combining the error models of the sensors with error equations of the INS (section 4.2) we can make an error state Kalman filter to estimate the error in the INS. This is done by using the error in the absolute states (position, velocity and attitude) as the state vector in the filter. The filter measurement will then be the difference in the measurements from the INS and the supporting instruments. This chapter will derive and present the model needed for implementing a Kalman filter for indirect integration. A schematic diagram of the filter structure is showed in figure 6.1.

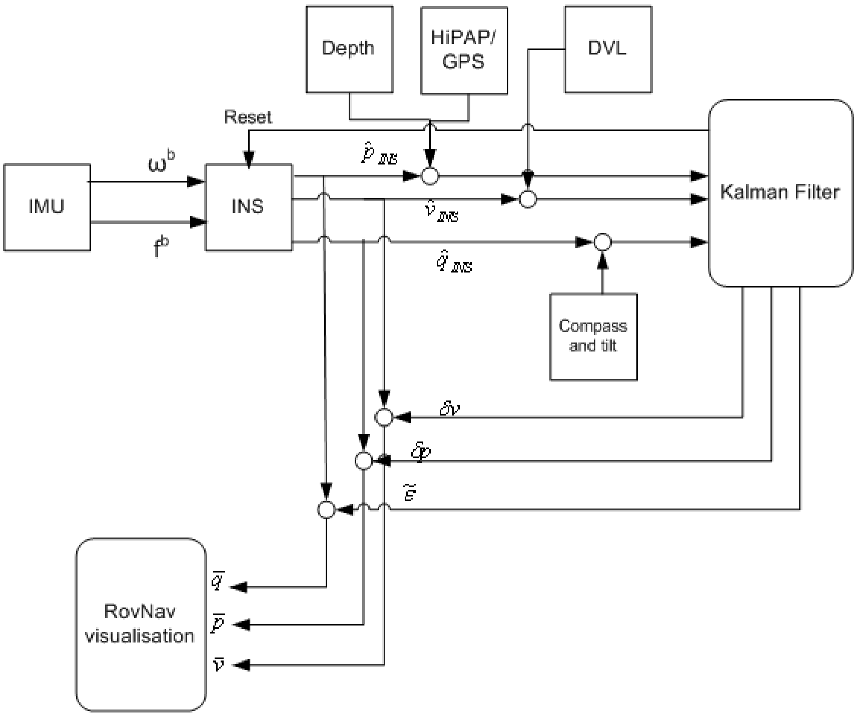

Figure 6.1: Schematic diagram of the filter structure

6.1 Simplification of error dynamics

In chapter 4 a model of the error dynamics was derived. Because of different noise ratios these equations can be significantly simplified. In this section a simplified model of the error dynamics for implementation in the Kalman filter will derived.

Velocity In equation 4.28 the error of the velocity was derived as

$$\begin{aligned}\delta\dot{\mathbf{v}}^n_{eb} = \delta\mathbf{g}^n_{eb} + \mathbf{R}_{\eta,\epsilon}(2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + 2\mathbf{S}^2(\tilde{\epsilon}))\mathbf{f}^b_{IMU} + \mathbf{R}^n_b\delta\mathbf{f}^b_{ib} - (2\mathbf{S}(\delta\omega^n_{ie})\\ + \mathbf{S}(\delta\omega^n_{en}))\hat{\mathbf{v}}^n_{eb} - (2\mathbf{S}(\omega^n_{ie}) + \mathbf{S}(\omega^n_{en}))\delta\mathbf{v}^n_{eb}\end{aligned} \tag{6.1}$$

δg^n_{eb} is the error in gravity. We can assume that the gravity at any given point can be estimated with high accuracy such that $\delta\mathbf{g}^n_{eb}$ can be neglected. It can be showed that the acceleration on the vessel due to coriolis forces can be upper bounded by $2\sqrt{6}\,\|\mathbf{v}\|\,(\frac{\|\mathbf{v}\|}{\mathbf{R}_e} + \omega_i e)$ (Farrell & Barth 1999). For a vessel traveling at 2 m/s this would be in the vicinity of $0.7mm/s^2$, the error of this acceleration will be even smaller and can be neglected. The angular velocity due to velocity in NED, ω_{en}, is derived from by dividing the velocity in NED by the earth's radius. This becomes very small ($10^{-7}\frac{rad}{s}$) and can therefore be neglected. $\delta\omega_{en}$ is even smaller and can also be removed from the expression. From this discussion we get the following for the error in velocity:

$$\delta\dot{\mathbf{v}}^n_{eb} = \mathbf{R}_{\eta,\epsilon}(2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + 2\mathbf{S}^2(\tilde{\epsilon}))\mathbf{f}^b_{IMU} + \mathbf{R}^n_b\delta\mathbf{f}^b_{ib} \tag{6.2}$$

Attitude In equation 5.75 the attitude error dynamics were derived as

$$\dot{\tilde{\mathbf{q}}} = \frac{1}{2}\begin{bmatrix} 0 & -(\delta\omega^b_{ib} - \delta\omega^b_{in})^T \\ \delta\omega^b_{ib} - \delta\omega^b_{in} & \mathbf{S}(\delta\omega^b_{ib} - \delta\omega^b_{in}) \end{bmatrix}\tilde{\mathbf{q}} \tag{6.3}$$

Since we are dealing with error dynamics which are assumed small we approximate $\tilde{\eta} \approx 1$. $\delta\omega_{in}$ is defined as the sum of $\delta\omega_{ie}$ and $\delta\omega_{en}$. Since we know the accurate value of ω_{ie} we neglect $\delta\omega_{ie}$, and for the same reason as for the velocity we also neglect $\delta\omega_{en}$. The resulting error equation for the attitude is

$$\dot{\tilde{\epsilon}} = \frac{1}{2}\delta\omega^b_{ib} + \frac{1}{2}\mathbf{S}(\delta\omega^b_{ib})\tilde{\epsilon} \tag{6.4}$$

Position The error dynamics for the position was given in equation 4.37 as

$$\begin{aligned} \begin{bmatrix} \delta\dot{\lambda} \\ \delta\dot{\phi} \\ \delta\dot{h} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & \frac{-v_n}{(r_\lambda+h)^2} \\ \frac{v_e sin(\lambda)}{(r_\phi+h)cos^2(\lambda)} & 0 & \frac{-v_e}{(r_\phi+h)^2 cos(\lambda)} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\lambda \\ \delta\phi \\ \delta h \end{bmatrix} \\ &+ \begin{bmatrix} \frac{1}{r_\lambda+h} & 0 & 0 \\ 0 & \frac{1}{(r_\phi+h)cos(\lambda)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta v_n \\ \delta v_e \\ \delta v_d \end{bmatrix} \end{aligned} \tag{6.5}$$

Since the earth radius is much larger than the depth of the vessel we may assume $r_\phi + h \approx r_\phi$ and $r_\lambda + h \approx r_\lambda$. To make the calculations more efficient we can also assume circular earth, i.e. $r_\phi = r_\lambda = r_e$, where $r_e = |r_{en}| = 6371 \cdot 10^3 m$ This results in the following position error dynamics:

$$\begin{aligned} \begin{bmatrix} \delta\dot{\lambda} \\ \delta\dot{\phi} \\ \delta\dot{h} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & \frac{-v_n}{r_e^2} \\ \frac{v_e sin(\lambda)}{r_e cos^2(\lambda)} & 0 & \frac{-v_e}{r_e^2 cos(\lambda)} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\lambda \\ \delta\phi \\ \delta h \end{bmatrix} \\ &+ \begin{bmatrix} \frac{1}{r_e} & 0 & 0 \\ 0 & \frac{1}{r_e cos(\lambda)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta v_n \\ \delta v_e \\ \delta v_d \end{bmatrix} \end{aligned} \tag{6.6}$$

For use in further use we write the model as $\delta\dot{\mathbf{p}} = \mathbf{\Psi}(\mathbf{p}, \mathbf{v})\delta\mathbf{p} + \mathbf{\Upsilon}(\mathbf{p})\delta\mathbf{v}$.

6.2 Error state space model

To write the system on a state space model, a state vector is introduced. This consists of the main errors in the system, and are used to compensate for the error in the INS measurements.

$$\mathbf{x} = \begin{bmatrix}\delta\mathbf{p}^T & \delta\mathbf{v}^T & \tilde{\epsilon}^T & \mathbf{b}^T & \beta^T & \mathbf{b}_{HiPAP}^T\end{bmatrix}^T \tag{6.7}$$

Each of the states in the state vector represents a three dimensional error vector.

The nonlinear state space model can be written in the following form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \Omega\xi(t) \tag{6.8}$$

By using the equations 6.2, 6.4, 6.6, 4.10, 4.9 and 4.46 the model can be written as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{\Psi}(\mathbf{p},\mathbf{v})\delta\mathbf{p} + \mathbf{\Upsilon}(\mathbf{p})\delta\mathbf{v} \\ \mathbf{R}_{\eta,\epsilon}(\mathbf{b} + \mathbf{2}(\mathbf{S}(\tilde{\epsilon})) + \mathbf{S^2}(\tilde{\epsilon}))\mathbf{f_{ib}^b} \\ \frac{\mathbf{1}}{\mathbf{2}}\beta + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{S}(\beta)\tilde{\epsilon} \\ -\mathbf{T_b^{-1}b} \\ -\mathbf{T}_\beta^{\mathbf{-1}}\beta \\ -\mathbf{T_{b_{HiPAP}}b_{HiPAP}} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\eta,\epsilon}\xi_\mathbf{v} \\ \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{I} - \mathbf{S}(\tilde{\epsilon}))\xi_\epsilon \\ \xi_\mathbf{b} \\ \xi_\beta \\ \xi_\mathbf{b_{HiPAP}} \end{bmatrix} \tag{6.9}$$

Where the T matrices are diagonal matrices with the Markov process time constants on the diagonal. This is the continuous representation of the nonlinear system. For use in a computer realization the model must be a discrete representation. This can be done in several different realizations. The forward Euler method is one option (Egeland & Gravdahl 2002). Forward Euler method with time step length h, is given as:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\mathbf{f}(\mathbf{y_n}, t_n) \tag{6.10}$$

The position measurement for the Kalman filter is

$$\begin{aligned} \mathbf{z}_1 =& \mathbf{p}_{INS} - \mathbf{p}_{GPS,HiPAP,h} \\ =& \mathbf{p} + \delta\mathbf{p}_{INS} - (\mathbf{p} + \delta\mathbf{p}_{HiPAP} + \xi_{GPS,HiPAP,h}) \\ =& \delta\mathbf{p}_{INS} - \delta\mathbf{p}_{HiPAP} - \xi_{GPS,HiPAP,h} \end{aligned} \tag{6.11}$$

and the velocity measurement is

$$\begin{aligned} \mathbf{z}_2 =& \mathbf{v}^n_{INS} - \mathbf{R}^n_b\mathbf{v}^b_{DVL} \\ =& \mathbf{v}^n + \delta\mathbf{v}^n_{INS} - (\mathbf{R}^n_b + \delta\mathbf{R}^n_b)(\mathbf{v}^b + \xi^b_{dvl,k}) \\ =& \delta\mathbf{v}^n_{INS} - \delta\mathbf{R}^n_b\mathbf{v}^b - \mathbf{R}^n_b\xi^b_{dvl,k} \end{aligned} \tag{6.12}$$

using equation 2.29, equation 6.12 can be written

$$\mathbf{z}_2 = \delta\mathbf{v}^n_{_{INS}} - \mathbf{R}^n_b\xi^b_{dvl,k} - \mathbf{R}^n_b\left[2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + 2\mathbf{S}^2(\tilde{\epsilon})\right]\mathbf{v}^b \tag{6.13}$$

The attitude error measurement from the DVL tilt and compass measurement, is written as $\mathbf{z}_3$.

$$\begin{aligned}\mathbf{z}_3 =& \mathbf{q}^{-1}_{dvl} \otimes \hat{\mathbf{q}}_{_{INS}} \\ =& (\mathbf{q}_{nb} \otimes \xi_{q_{dvl}})^{-1} \otimes \hat{\mathbf{q}}_{_{INS}}\end{aligned} \tag{6.14}$$

For simplicity this will be modeled as

$$\mathbf{z}_3 = \begin{bmatrix}\tilde{\eta}_{nb} - \xi_{\eta_{dvl}} \\ \tilde{\epsilon}_{nb} - \xi_{\epsilon_{dvl}}\end{bmatrix} \tag{6.15}$$

Since only the $\tilde{\epsilon}$ part of the quaternion is represented in the Kalman- filter the $\mathbf{z}_3$ sent to the filter is

$$\mathbf{z}_3 = \tilde{\epsilon}_{nb} - \xi_{\epsilon_{dvl}} \tag{6.16}$$

This yields

$$\begin{aligned}\mathbf{z}_k =& \begin{bmatrix}\delta\mathbf{p}_{_{INS}} - \delta\mathbf{p}_{_{HiPAP}} - \xi_{GPS,HiPAP,h} \\ \delta\mathbf{v}^n_{_{INS}} - \mathbf{R}^n_b\left[2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + 2\mathbf{S}^2(\tilde{\epsilon})\right]\mathbf{v}^b - \mathbf{R}^n_b\xi^b_{dvl,k} \\ \tilde{\epsilon}_{nb} - \xi_{\epsilon_{dvl}}\end{bmatrix} \\ =& \begin{bmatrix}\delta\mathbf{p}_{_{INS}} - \delta\mathbf{p}_{_{HiPAP}} \\ \delta\mathbf{v}^n_{_{INS}} - \mathbf{R}^n_b\left[2\tilde{\eta}\mathbf{S}(\tilde{\epsilon}) + 2\mathbf{S}^2(\tilde{\epsilon})\right]\mathbf{v}^b \\ \tilde{\epsilon}_{nb}\end{bmatrix} - \begin{bmatrix}\xi_{GPS,HiPAP,h} \\ \mathbf{R}^n_b\xi^b_{dvl,k} \\ \xi_{\epsilon_{dvl}}\end{bmatrix}\end{aligned} \tag{6.17}$$

In the Kalman filter we need the covariance matrix of the process noise. This is found by taking the covariance of the noise vector of equation 6.22.The covariance matrix is found to be

$$\begin{aligned}\mathbf{Q}_k =& \begin{bmatrix}\mathbf{R}_{\eta,\epsilon}\sigma^2_{\xi_f}\mathbf{R}^T_{\eta,\epsilon} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}(\mathbf{I} - \mathbf{S}(\tilde{\epsilon}))\sigma^2_{\xi_\omega}\frac{1}{2}(\mathbf{I} - \mathbf{S}(\tilde{\epsilon}))^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma^2_{\gamma_f}\cdot\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma^2_{\gamma_\omega}\cdot\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma^2_{\gamma_{HiPAP}}\cdot\mathbf{I}\end{bmatrix} \\ =& \begin{bmatrix}\sigma^2_{\xi_f}\cdot\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{4}(\mathbf{I} - \mathbf{S}^2(\tilde{\epsilon}))\sigma^2_{\xi_\omega} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma^2_{\gamma_f}\cdot\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma^2_{\gamma_\omega}\cdot\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \sigma^2_{\gamma_{HiPAP}}\cdot\mathbf{I}\end{bmatrix}\end{aligned} \tag{6.18}$$

The covariance matrix of the measurement noise is found by taking the covariance of the noise vector of equation 6.17. The covariance matrix is

$$\mathbf{R}_k = \begin{bmatrix} \sigma^2_{\xi_{DGPS,HiPAP,h}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma^2_{\xi_{DVL}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma^2_{\xi_{DVL}} \end{bmatrix} \tag{6.19}$$

6.3 Linearizations for use in EKF

The discretization requires that the derivatives of the system matrix is defined. The discrete representation of f_k is noted as Φ_k. For all the linear terms the differential only cause an element on the diagonal, where the state is differentiated with reference to the corresponding state variable, the nonlinear terms are therefore treated alone. Starting with the expression corresponding to $\dot{\tilde{\epsilon}}$:

$$\begin{aligned} \frac{\partial \mathbf{f_{7-9}}}{\partial \tilde{\epsilon}} &= \frac{\partial}{\partial \tilde{\epsilon}} \left(\frac{\mathbf{1}}{\mathbf{2}}\beta + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{S}(\beta)\tilde{\epsilon} \right) = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{S}(\beta) \\ \frac{\partial \mathbf{f_{7-9}}}{\partial \beta} &= \frac{\partial}{\partial \beta} \left(\frac{\mathbf{1}}{\mathbf{2}}\beta + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{S}(\beta)\tilde{\epsilon} \right) = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{I} + \frac{\mathbf{1}}{\mathbf{2}}\mathbf{S}(\tilde{\epsilon}) \end{aligned} \tag{6.20}$$

Further more, derivation of the expression corresponding to $\delta\dot{\mathbf{v}}$:

$$\begin{aligned} \frac{\partial \mathbf{f_{4-6}}}{\partial \mathbf{b}} &= \frac{\partial}{\partial \mathbf{b}} \mathbf{R}_{\eta,\epsilon}(\mathbf{b} + \mathbf{2}(\mathbf{S}(\tilde{\epsilon})) + \mathbf{S^2}(\tilde{\epsilon}))\mathbf{f^b_{ib}} = \mathbf{R}_{\eta,\epsilon} \\ \frac{\partial \mathbf{f_{4-6}}}{\partial \tilde{\epsilon}} &= \frac{\partial}{\partial \tilde{\epsilon}} \mathbf{R}_{\eta,\epsilon}(\mathbf{b} + \mathbf{2}(\mathbf{S}(\tilde{\epsilon})) + \mathbf{S^2}(\tilde{\epsilon}))\mathbf{f^b_{ib}} \\ &= \mathbf{2}\mathbf{R}_{\eta,\epsilon}(-\mathbf{S}(\mathbf{f^b_{ib}}) + \mathbf{S}(\mathbf{S}(\tilde{\epsilon})\mathbf{f^b_{ib}}) + \mathbf{S}(\tilde{\epsilon})\mathbf{S}(\mathbf{f^b_{ib}})) \end{aligned} \tag{6.21}$$

The derivation of $\mathbf{S^2}(\tilde{\epsilon})\mathbf{f^b_{ib}}$ with respect to $\tilde{\epsilon}$ is showed in Idsø (1999). Combining the equations derived over, result in a system matrix Φ as follows:

$$\begin{aligned} \mathbf{\Phi} = &\left[\begin{matrix} \mathbf{I} + h\hat{\mathbf{\Psi}}_k & h\hat{\mathbf{\Upsilon}}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & h2\hat{\mathbf{R}}^n_{b_k}(-\mathbf{S}(\hat{\mathbf{f}}_k) + \mathbf{S}(\mathbf{S}(\hat{\tilde{\epsilon}}_k)\hat{\mathbf{f}}_k) + \mathbf{S}(\hat{\tilde{\epsilon}}_k)\mathbf{S}(\hat{\mathbf{f}}_k)) \\ \mathbf{0} & \mathbf{0} & \mathbf{I} - \frac{h}{2}\mathbf{S}(\hat{\mathbf{B}}_{\omega_{IMU}}) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix} \right. \\ &\left. \begin{matrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ h\hat{\mathbf{R}}^b_n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{h}{2}\mathbf{I} - \frac{h}{2}\mathbf{S}(\tilde{\epsilon}_k) & \mathbf{0} \\ \mathbf{I} - h\mathbf{T}^{-1}_{f_{IMU}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - h\mathbf{T}^{-1}_{\omega_{IMU}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} - h\mathbf{T}^{-1}_{HiPAP} \end{matrix} \right] \end{aligned} \tag{6.22}$$

Taking the derivative of $\mathbf{z}_k$ w.r.t. $\mathbf{x}$, $\frac{\partial \mathbf{z}_k}{\partial \mathbf{x}_k^T}(\mathbf{x}_k)$ gives

$$\mathbf{D}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} & -2\hat{\mathbf{R}}_b^n\left[-\mathbf{S}(\hat{\mathbf{v}}_{eb}^b) + \mathbf{S}(\mathbf{S}(\tilde{\epsilon}_k)\tilde{\mathbf{v}}_{eb}^b) + \mathbf{S}(\tilde{\epsilon}_k)\mathbf{S}(\hat{\mathbf{v}}_{eb}^b)\right] & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{6.23}$$

6.4 Representation

The attitude in the system is represented by a quaternion. The three measurements of tilt in 2D and compass produce a set of Euler angles. These Euler angles are transformed to a quaternion by using the transformation presented in Diebel (2006). The transformation is called the Euler angle sequence (1,2,3), and is given by

$$\mathbf{q}_{123}(\psi, \theta, \phi) = \begin{bmatrix} \cos\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} \\ -\cos\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} \\ \cos\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} \\ \cos\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} - \sin\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} \end{bmatrix} \tag{6.24}$$

This transformation has a singularity in $\theta = \frac{\pi}{2} + n\pi$, for $n = 1, 2, 3, \ldots$. Since it is assumed that the tilt sensor is only available up to a limited tilt angle, the measured angel that inflicts the singularity will not be reached.

6.5 Filter feedback

If the filter is implemented without any form of feedback the error states can diverge towards infinity. This will also be the case for the INS estimates of position, velocity and attitude. Since the linearization in the extended Kalman filter only is valid for small errors, there need to be implemented some kind of reset of the INS navigation equations. The higher order elements in the Taylor series in the unscented Kalman filter is also neglected as described in section 5.2.3, and the reset is also necessary in this filter. The feedback reset is done by subtracting the Kalman filter error estimates from the INS states. This must be done with the updated states after a measurement. By subtracting the estimated error, the error states must also be reset to zero before the Kalman filter establishes the apriori estimate (prediction). The reset can be implemented at any time step, but it is not optimal to reset to often. If the filter is reset infrequently the error might grow large and the accuracy of the model is reduced. If the filter is reset too often the filter is not given a chance to find the a good estimate of the error. The filter need some steps of correction before it is reset.

6.6 Robustness

In control theory robustness is used as a term describing a systems sensitivity to uncertainty. In the Kalman filter approach the noise and bias error has to be modeled. The white measurement noise is modeled by its variance defined on the diagonal of the noise covariance matrices (R and Q). The biases, both on the IMU and HiPAP, are modeled as 1st order Markov processes. The model is defined by the variance of the white noise that drives the process and the process time constant.

The uncertainty of the process and measurement noise has different influence on the filter. If the measurement noise is modeled with a noise variance larger than the actual measurement noise, it can be seen from the Kalman filter equations (5.8 - 5.13) that this will cause the Kalman gain to be lower than optimal. The effect on the Kalman gain is opposite if the process noise is modeled with larger variance. This means that if the bias drift is considered to be large, more emphasis is put on the measurement.

The noise properties are seen as the only tuning factor in a linear Kalman filter. This is also valid for the extended Kalman filter. In addition to the covariance matrices the unscented Kalman filter also have three constants describing the spread of the sigma points (α, β and κ). One of the requirements for the Kalman filter to be a minimum variance filter, is thatR_k and Q_k are correct (Brown & Hwang 1997). This means that the actual implemented filter will not be optimal if these matrices are estimated wrongly. The tuning of the Kalman filter will be dependent on what the filter output will be used for. If the filter estimates are feedback to a controller, the estimate should be as accurate as possible. But in the RovNav application where the output are input to a visualization tool, one might wish very smooth estimates. This can be achieved by increasing the elements of the measurement noise variance matrix, but will be a trade off with respect to accuracy and filter response.

6.7 Observability

From control theory observability studies the possibility of estimating the internal states from the external output (Chen 1999). A linear discrete system is said to be observable if and only if the observability matrix O defined by

$$O = \begin{bmatrix} \mathbf{D} \\ \mathbf{D\Phi} \\ \vdots \\ \mathbf{D\Phi}^{n-1} \end{bmatrix} \tag{6.25}$$

has rank n. $\mathbf{D}$ and $\mathbf{\Phi}$ are the measurement and system matrices of the discrete linear system. If the rank of O is n the matrix may still be close to singular. A measure of how close a matrix is to being singular is the *condition number* defined as the ratio between the largest and smallest singular value of a matrix (Skogestad & Postlethwaite 2005)

$$\kappa(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)} \tag{6.26}$$

If A has a large condition number it is said to be ill conditioned. If this is the case for the observability matrix O then it is said that the system described by D and Φ has a low grade of observability.

For a nonlinear system it is not equally straight forward to investigate observability. Given the nonlinear system

$$\dot{x} = f(x, u) \tag{6.27}$$

$$z = h(x) \tag{6.28}$$

Taking the Lie derivatives (section 2.2.6) of h w.r.t. f yields

$$l(x, u) = \begin{bmatrix} L_f^0(h) \\ \vdots \\ L_f^{n-1}(h) \end{bmatrix} \tag{6.29}$$

Expanding this in a series about $x = x_0$ for $u = u_0$ gives

$$l(x, u_0) \approx l(x_0, u_0) + \frac{\partial l(x, u_0)}{\partial x}|_{x=x_0}\Delta x + h.o.t \tag{6.30}$$

then $dG = O = \frac{\partial l(x,u_0)}{\partial x}|_{x=x_0}$ must have rank n for the system to be locally observable in a vicinity around x_0 (Hedrick 2007). As for the linear case if O is ill conditioned the grade of local observability will be low.

The observability studies in this thesis are performed in Matlab. By investigating the observability properties of the navigation system (equations 3.4, 3.9 and 3.11) in this thesis it is found that O has full rank without attitude measurement. The condition number is however relatively high, which indicates that the observability is low. With a secondary attitude measurement the condition number is significantly reduced, hence observability is increased. For accuracy and robustness it is therefore recommended that a secondary attitude measurement is used.

6.8 Sensor latency

When fusing the measurements from the ROV with the current best prediction of the vehicle state, considerable care must be taken to incorporate the measurement in the optimal fashion. The inertial sensors has a high update rate and short time delay, where as the HiPAP is limited by the speed of sound in water. This delay is

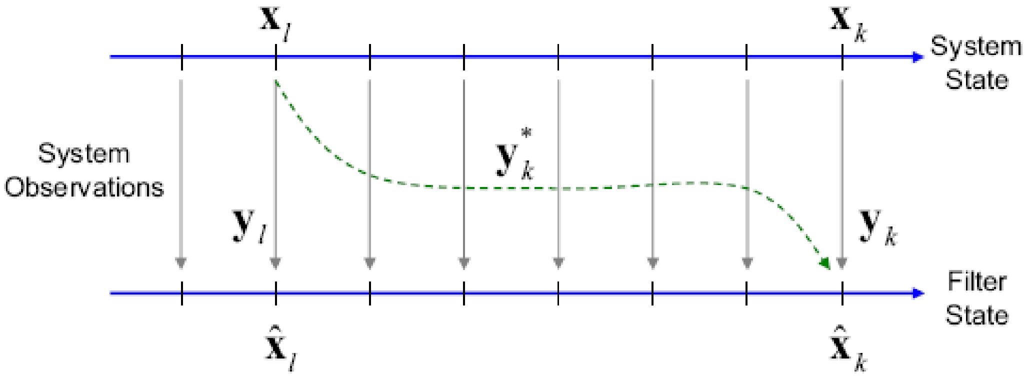

Figure 6.2: Measurement delayed, from Larsen et al. (1998)

dependent of the distance from the point of reference. Figure 6.2 illustrates the issue graphically. A system is given as:

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + v_k \\ y_k &= C_k x_k + n_k \end{aligned} \tag{6.31}$$

where n_k and v_k is measurement and process noise. A measurement that corresponds to the system state at $t = l$, arrives N samples later at $t = k$. This means that the filter receives some measurements valid for x_k and some valid at x_l at the same time. The delayed measurements y_k^* can be mathematically described as:

$$y_k^* = C_l x_l + n_k \tag{6.32}$$

There are several different ways of dealing with the sensor latency in a filter described in the literature (van der Merwe et al. 2004), (Larsen et al. 1998) and (Mandt et al. n.d.). The following will briefly describe some of the possible approaches.

The simplest way of dealing with the latency is to ignore it. This means that the filter assumes that the measurement is valid at the time it is available. For an error state filter this would only be valid if the error was much larger than the possible state change in the delay time period. If the ROV travels at $2m/s$ and the HiPAP measurement is 5 sec delayed this would introduce a 10 meter error, which is much larger than the possible INS drift in the same period. Because of the possible large error between the current state and the state at the time when the delayed measurement was valid, this approach is not used.

Another solution is to recalculate the complete trajectory for the delay period, when the lagged measurement is received. This approach requires all measurements and all filter estimates being saved in the delay period. The computational burden in this solution is significant, as well as the need for storage capacity. Dependent on the real time requirements of the estimates and the available CPU capacity, meeting the deadlines might not be possible.

Running the filter and navigation equation on a fixed delay is possible. This means that all measurements (including IMU) are stored until all data is avaliable. The filter delay can be set to the worst-case expected HiPAP latency. This would give optimal Kalman filter estimates, but all estimates would be delayed corresponding to the HiPAP delay. If the estimate is used in any real time application, for example in a control and guidance algorithm, this is not good solution.

Larsen et al. (1998) described a method that provides optimally fused estimates not only at the instance when the delayed measurements arrive, but also in the interim period between these updates. Larsen achieves this by running a second Kalman filter in parallel to the main filter. The second filter produces optimal estimates in the interim between measurements. The downside of having optimal estimates at all time, is that it will double the computational complexity.

Mandt et al. (n.d.) described an approach for the HUGIN project that meant running the inertial navigation and Kalman filter in real-time and make delayed measurements. This is based on the assumption that the error does not change significantly during the delay, and that the filter is designed as an error state filter. This solution requires the inertial data to be saved for the delay period. The filter measurements are made using inertial data with the correct time step together with the delayed secondary measurement (HiPAP, DVL, etc.). The measurements are evaluated using current covariance matrices and estimates in the Kalman filter. The feedback resets are done on the current inertial data. This solution is not CPU intensive and does not require memory to store matrices, this is why the HUGIN team implemented this solution on the HUGIN 3000 AUV.

Determine the delay

The sensor delay of an hydro acoustic positioning is highly dependent of the distance from the transponder to the measurement unit. In other words the distance from the underwater vehicle to its point of reference.[1] The speed of sound in water[2] limits both the rate at which the position is updated as well as the minimum delay possible to achieve. Some hydroacoustic positioning systems can estimate the delay, and therefore time stamp the measurement with the time the transponder replied.

[1] The point of reference is either a surface ship or receivers fixed to the seabed.

[2] The speed of sound in water is estimated to about $1500m/s$, but is highly dependent on both water temperature and salinity.

Reset with sensor latency

The challenge of reset of the INS equation in an error state Kalman filter for the HUGIN project is described in Mandt et al. (n.d.). If the filter is reset between the time a delayed measurement was valid and real time. An error that has already been estimated and reset is again measured and might be reset twice. This is solved by storing the changes in the error estimates and change the measurement equation to:

$$z_m = D_m(\overline{x}_m - \sum_{i=m}^{rt} \delta \hat{x}_i) - y_m \tag{6.33}$$

Stability

Time delay in measurement is a stability issue in observer design and in control theory in general. In this design it is assumed that the time delay is relatively small compared to the time constants of the error. This means that the error does not change considerably in the latency period.

The measurement will be most evident in the position update from the hydro acoustic positioning. If we assume that the velocity has a bounded error when the secondary velocity measurement is available, the position error growth rate is bounded. The theoretical proof of stability for the sensor integration techniques with measurement delays presented in this report, is out of the scope of this work.

Nonlinear Observer

The nonlinear observer does not distinguish between changes in the state caused by the actual movement of the vessel and changes made as error correction. This means that it is not possible to remove changes done in the error estimate over the delay period. For this reason the delayed measurement is incorporated as it is, with out any compensation for the time passed. The observers input is the error state, so only changes in the error over the delay period is of any concern to this solution. Since there is no form of reset, this simple solution is valid if the error or drift is slower than the delay period.

6.9 Convergence rates of KF

When initializing the filter there will always be a certain degree of error in the initial state. The rate of convergence towards the true state, is for the Kalman filter implementations dependent on the values of the covariance matrix. The magnitude of the initial covariance matrix elements should indicate the degree of uncertainty in the initial states.

The Kalman covariance matrix gives an indication on how certain the last filter estimate is. If the covariance has large elements there is large uncertainty in the estimate, and vice versa for small covariance elements. When the filter calculates (or is initiated with) a large covariance matrix the Kalman gain calculated will also be large. This will in turn weight the new measurements more relative to the last predicted state.

The main difference of the extended and unscented Kalman filter is how the covariance matrix is calculated. This will influence the behavior of the filter response to disturbances. This is significant if the two covariance matrixes do not converge towards the same values, and thereby give the filters different behavior.

Part III

Simulation & Discussion

Chapter 7

Initialization and Implementation

7.1 General Implementation

The mathematical filters that were presented in sections 5.2.2, 5.2.3 and 5.3.2 can be tested by implementing them in a mathematical simulator. By using this approach it is possible to investigate the properties(e.g. robustness and accuracy) of the filters.

The filters in this work have been implemented in Matlab. Both the filters and the simulation structure is based solely on Matlab basis functions.[1] This is done to make the transition to implementation in other languages easy. Matlab is equipped with several ready made toolboxes including a toolbox for implementing Kalman filter design. This toolbox has not been used in this work. By not using standard implementations the filters also allows for the user to look at the internal filter variables. The simulation structure consists of a trajectory simulator, signal generators and the filters. This setup is chosen to test the performance of the different filters without real measurement data, but with characteristics similar to that of real instruments.

The modules/functions are merged in a simulation loop, where the functions are called in sequence with different input parameters. By implementing the different modules as functions, the user has a good overview over which variables and parameters are used where. This is important since the simulation operates with several versions of the same parameter[2].

The goal of the simulation phase is to show that the filters presented in previous chapters are useful in estimating position, attitude and velocity for an ROV/AUV. This means that the filters' output should be a better estimate than any of the measuring instruments could produce alone. Since most of the measurements are received at a lower frequency the goal is also to produce a "more continuous" state estimate. Also the robustness of the algorithm will be tested both for errors in model param-

[1] Some of the functions used are from the Marine GNC Matlab toolbox (Fossen 2005). The code for these functions are included in the files described in appendix D

[2] For example the filter has a real position, two measured positions (INS and HiPAP) and a filtered position

eters and in loss of measurements. The fact that some signals may arrive some time after they are valid will also be tested. When the filters are tested we will be able to compare the different filters and determine which one has the best overall performance based on the criteria presented in chapter 8.

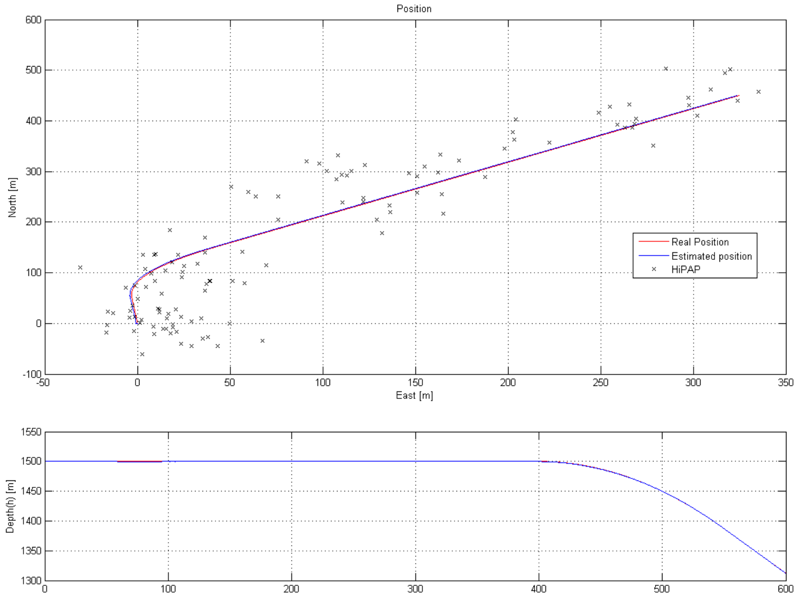

Figure 7.1: Reference trajectory (with HiPAP measurements and UKF estimate from simulation 2. See chapter 8.3.2).

To test the accuracy of the filters a reference trajectory was developed. The trajectory should test both position, velocity and attitude accuracy, and is therefore developed without taking into account the physical limitations of an ROV/AUV. It is assumed that the ROV/AUV can move in any direction with any rotation around its own axes. Further more the filters will be tested with different parameters in signal generation and filter model. This will point out sensitivity to model errors in the filters. Tests will also be run to see how well the filters work when the external measurements (HiPAP and DVL) are lost for a period of time, and how well they recover from a sensor outage. How the filters handles IMU drop out is also tested.

The simulations are run at 10Hz. It is assumed that the HiPAP is available every 5 second. This should be a conservative estimate, since the manufacturer estimates an update rate of 0.5 Hz when the ROV is at a depth of 1500 meter. The DVL is available at each time step, except when the tilt angles (roll and pitch) are greater than $\pm 15°$ where the DVL does not give readings (Teledyne-Instruments 2005).

All the instrument data used is based on data from the manufacturers specification for a selected instrument, see appendix C.

Latitude (λ)	$4°west$
Longitude (ϕ)	$56°north$
Depth	1500 m
North velocity	0 m/s
East velocity	0 m/s
Down velocity	0 m/s
Attitude, η	1
ϵ_1	0
ϵ_2	0
ϵ_3	0

Table 7.1: Initial states for all simulations

The trajectory is based on a series of input accelerations and angular velocities. From these inputs the real position, velocity and attitude are determined. Based on this information the measurements are created with the given noise and bias characteristics. All simulations start from the same position with the same initial states. The initial states for the system are given in table 7.1.

7.2 Filter Initialization

An issue that needs to be addressed is initialization of both INS and Kalman filters. The INS calculates the position, velocity and attitude from a given initial state by integrating the signals from the IMU . To ensure that the total estimate is as correct as possible as soon as possible, these initial values also needs to be correct. The initial INS state should be the best available state estimate at the given time. Measurements from the external measurements (HiPAP, DVL and tilt sensor) has satisfactory accuracy, and can be used for initialization. In addition to the INS the filter also has to be initialized. The Kalman filters need an initial state and an initial state covariance matrix ($\mathbf{X_0}$). The initial state is the assumed initial error in the INS. Since this error is the best estimate at the time, the assumed initial error will be zero. The initial state $\mathbf{x}_0$ covariance matrix ($\mathbf{X_0}$) should be chosen to reflect the uncertainty of the INS initialization. If the uncertainty is large the covariance should also be chosen large. This helps ensure a more rapid convergence of the filter, such that it achieves a satisfactory accuracy level within a few reset intervals (see chapter 6.5). If the initial INS error is very large (e.g. $1000m$ in position) the filter requires a longer period to converge to the correct value. In theory a limit initial error may exist that which exceeded leads to divergence of the filter, no such limit has been found.

In the case of the nonlinear observer the observer is integrated with the navigation equations, and therefore does not need a separate initialization. In this case there is no information in the observer about the initial error (no state covariance matrix), hence it is more important that the initial states are correct than is the case for the Kalman filters. In the case of constant observer gains, the recovery from the initial error may take considerably longer than for the Kalman filters.

7.3 Tuning of UKF

The UKF will be implemented as an indirect filter. The error state space model was derived in chapter 6.2, and the same model will be used when implementing the UKF. In EKF the only tuning parameters are the covariance matrices $\mathbf{R}_k, \mathbf{Q}_k$ and $\hat{\mathbf{X}}_0$. In UKF there are additional tuning parameters; α, β ans κ. α is a small positive value that determines the spread of the sigma points around $\bar{\mathbf{x}}$. From the conditions on the value of α from chapter 5.2.3, $\alpha = 0.4082$. β is used to incorporate prior knowledge of the distribution of $\mathbf{x}$. Wan & v. d. Merwe (2000) states that $\beta = 2$ is optimal if $\mathbf{x}$ Gaussian. κ is typically set equal to zero. The dimension of our system is 18, hence $L = 18$ and χ is a 18×37 matrix $(2L + 1)$.

7.4 Implementation and tuning of Nonlinear Observer

When implementing the nonlinear observer in Matlab it needs to be discretisized. For this purpose we used the forward Euler method from equation 6.10.

$$x_{k+1} = x_k + hf(x_k) \tag{7.1}$$

This gives the following discrete attitude observer:

$$\begin{aligned}
\hat{\mathbf{q}}_{,k+1} &= \hat{\mathbf{q}}_{,k} + h\left(\frac{1}{2}\begin{bmatrix} -\hat{\epsilon_{,k}}^T \\ \hat{\eta_{,k}}\mathbf{I} + \mathbf{S}(\hat{\epsilon}_{,k}) \end{bmatrix}\left[\omega_{_{IMU}}, k + \hat{\beta}_{\omega_{_{IMU}},k}\right.\right. && (7.2)\\
&\quad \left.\left. + \mathbf{K}_1\tilde{\epsilon_{,k}} sgn(\tilde{\eta_{,k}})\right] - \frac{1}{2}\begin{bmatrix} -\hat{\epsilon_{,k}}^T \\ \hat{\eta_{,k}}\mathbf{I} + \mathbf{S}(\hat{\epsilon_{,k}}) \end{bmatrix}\omega^n_{in,k}\right) \\
\hat{\beta}_{\omega_{_{IMU}},k+1} &= (\mathbf{I} - h\mathbf{T}^{-1}_{\omega_{_{IMU}}})\hat{\beta}_{\omega_{_{IMU}},k} + h\frac{1}{2}K_2\tilde{\epsilon_k} sgn(\tilde{\eta},k) && (7.3)
\end{aligned}$$

Since $\hat{q}$ must be of unit length, the quaternion is normalized each time step using

$$\hat{\mathbf{q}}_k = \frac{\hat{\mathbf{q}}_k}{||\hat{\mathbf{q}}_k||_2} = \frac{\hat{\mathbf{q}}_k}{\sqrt{\hat{\mathbf{q}}_k^T\hat{\mathbf{q}}_k}} \tag{7.4}$$

The discrete velocity/ position observer is

$$\begin{aligned}
\hat{\mathbf{v}}^n_{eb,k+1} &= \hat{\mathbf{v}}^n_{eb,k} + h\Big[\hat{\mathbf{R}}^n_{b,k}(\mathbf{f}^b_{imu,k} + \hat{\mathbf{b}}_{f_{IMU},k}) + \hat{\mathbf{g}}^n_{eb,k} - (2\mathbf{S}(\omega^n_{ie}) && (7.5)\\
&\quad +\mathbf{S}(\omega^n_{en,k}))\hat{\mathbf{v}}^n_{eb,k} + \mathbf{K}_3\tilde{\mathbf{v}}^n_{eb,k} + (\hat{\mathbf{R}}^n_{e,k})^T(\tilde{\mathbf{p}}_{en_{HiPAP}} - \hat{\mathbf{b}}_{p_{HiPAP},k})\Big] &&\\
\hat{\mathbf{p}}_{en,k+1} &= \hat{\mathbf{p}}_{en,k} + h\Big[\hat{\mathbf{R}}^e_{n,k}\hat{\mathbf{v}}^n_{eb,k} + \mathbf{K}_4(\tilde{\mathbf{p}}_{en_{HiPAP}} - \hat{\mathbf{b}}_{p_{HiPAP},k})\Big] && (7.6)\\
\hat{\mathbf{b}}_{f_{IMU},k} &= (\mathbf{I} - h\mathbf{T}^{-1}_{f_{IMU}})\hat{\mathbf{b}}_{f_{IMU},k} + h\mathbf{K}_5(\hat{\mathbf{R}}^n_{b,k})^T\tilde{\mathbf{v}}^n_{eb,k} && (7.7)\\
\hat{\mathbf{b}}_{p_{HiPAP},k+1} &= (\mathbf{I} - h\mathbf{T}^{-1}_{b_{HiPAP}})\hat{\mathbf{b}}_{p_{HiPAP},k} - h\big[\mathbf{K}_6\big\{\mathbf{R}^n_{e,k}\tilde{\mathbf{v}}^n_{eb,k} + && (7.8)\\
&\quad \mathbf{K}_4^T(\tilde{\mathbf{p}}_{en_{HiPAP}} - \hat{\mathbf{b}}_{p_{HiPAP},k})\big\}\big] &&
\end{aligned}$$

The gain matrices $\mathbf{K}_1 ... \mathbf{K}_6$ needs tuning. To help the tuning we made some simplifications to the system to make it linear and then implemented the observer as a linear Kalman filter (KF). This way a Kalman gain matrix $\mathbf{K}$ was calculated. Since the correction term in the KF is

$$\hat{\mathbf{x}_k} = \bar{\mathbf{x}_k} + \mathbf{K}(\mathbf{y} - \mathbf{D}\mathbf{x}) \tag{7.9}$$

it has a different structure than that of the nonlinear observer and the gains calculated by KF can not be used directly in the observer. Some of the sub matrices in $\mathbf{K}$ may however be used as an indication of the magnitude of some of the observer gain matrices $\mathbf{K}_1 ... \mathbf{K}_6$. The tuning must be performed such that the criteria from equations 5.88 and 5.94 are satisfied. To satisfy 5.88 $\mathbf{K}_1$ and $\mathbf{K}_3$ are chosen relatively large. $\mathbf{K}_4$ is chosen small since it is used in a velocity correction and $\mathbf{K}_6$ is chosen to satisfy 5.94. The stability proof for the observer is only valid for a continuous system, hence stability is not proven for the discrete observer(which is the one that will be implemented). Of this reason the time-steps of the discrete observer must be small. The Matlab- code for implementation can be found on attached CD-ROM. The structure of the nonlinear observer, being a direct filter, is more intuitive than that of the Kalman filters, but it demands a more time-consuming tuning process since there is no direct link between the noise characteristics and the observer gains.

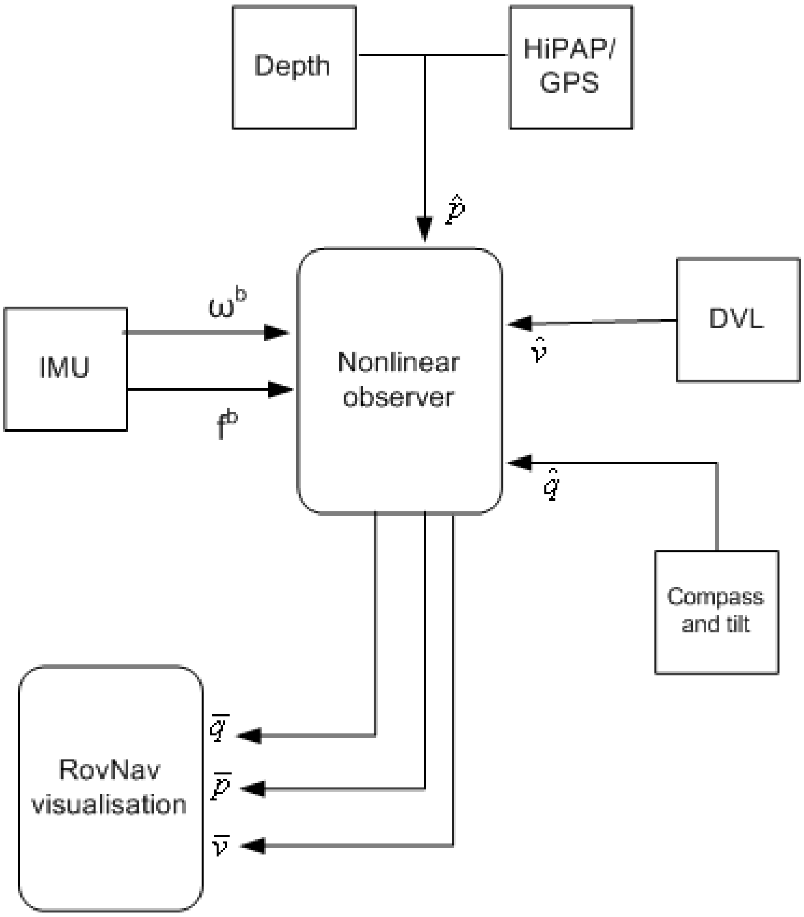

Figure 7.2: Schematic diagram of nonlinear observer structure

Chapter 8

Discussion of Filter Performance

8.1 Criteria for filter selection

To distinguish between the three different filters, a set of criteria for filter performance is selected. The purpose of the integrator algorithm is to use it in a real time application for visualizing the vehicle in operations. This put strict requirements on the robustness of the system. In this section several criteria of performance are described.

Standard deviation

In Walpole et al. (2002) standard deviation (STD) is described as the root mean square (RMS) deviation of the values from their arithmetic mean. Standard deviation is the most common measure of statistical dispersion, measuring how widely spread the values in a data set are. If the data points are all close to the mean, then the standard deviation is close to zero.

The mathematical definition of the STD is given by:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{8.1}$$

where $\overline{x}$ is the mean value, and x_i is the value of the measured variable.

To classify the performance of the sensor integration filter, the standard deviation of the estimation error is calculated. This gives a measure of the dispersion of the estimation error of the filter. The STD is only valid if it is reasonable to assume that the expected value (mean) of the error is constant.

Mean error

The mean is the expected value of a random variable. The mean is the sum of all the observations divided by the number of observations. Mathematically described the arithmetic mean is:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{8.2}$$

The mean is used to make sure that the estimation error is not biased. If the mean of the error over the simulated time series is very different from zero, this is an indication that the filter is biased.

Max error

The maximum error in the estimated state over the simulated time series gives an indication of the filter performance. The magnitude of the maximum error must be analyzed with a little caution. A larger error at one point in time does not necessarily mean that the filter performance is worse. This is because the signal errors are random signals, and the ”worst case” error might cause the best filter to give a larger error for one time step. For this reason the magnitude of the error is divided into sections and the average error over one of these time sections is analyzed.

Run time

The different filter all have slightly different structures, with different calculation taking place. This means that they will require different computer power. Since the vehicle will run as an AUV without power supply part of the time it is desired that the filter algorithm is as power efficient as possible. How much computer power is needed for each of the three filters can be determined from how long it takes for a specific filter to run a given simulation. A filter that takes longer to run a simulation of a given time series will require more calculations and more computer power.

Graceful degradation

A property of the sensor integrator is also how well it performs on a sensor dropout. This property can be described as graceful degradation (or fail soft) (Burns & Wellings 2001). It is essential that the estimates are reliable even if some of the sensors fall out for a period of time. This implies that the estimated error should not grow faster than the actual error in the remaining measurements in the system, which practically means that if all measurements except the IMU are lost, the total system error should be limited by the INS drift.

8.2 Simulations

To evaluate the filters we ran five separate simulations to test robustness and accuracy.

Simulation 1

Simulation 1 is run with noise characteristics from the instrument specifications. It is assumed that the real noise has characteristics close to that specified by the manufacturer.

Simulation 2

In simulation 2 the filters' robustness to noise is tested. In this simulation it is assumed that the actual noise characteristics are different from those specified in the data sheet. This is done by changing (increasing) the magnitude of all generated white noise, including that which drives the colored noise processes. The noise characteristics in the filter remain the same as in specifications. All measurements are available throughout the simulation.

Simulation 3

The third simulation is performed to test robustness to dropout of external measurements, and how well the filters recover once the measurements are returned. First HiPAP was lost for 600 seconds before the filter recovered for 200 seconds. Then the DVL was lost for 600 seconds. The filter once again got a 200 second recovery before all external measurements were lost for 400 seconds followed by a recovery period.

Simulation 4

The fourth simulation is performed to test robustness to dropout of inertial measurements (IMU), and how well the filters recover once the measurements are returned. A 200 second IMU loss was tested.

Simulation 5

The last simulation test the performance when HiPAP and DVL measurements arrive some time after they are valid. The HiPAP was tested with a 5 second delay while the DVL had a 0.5 second delay.

8.3 Performance Evaluation

To decide which filter is best suited for implementation in RovNav the simulation results will be evaluatied using the criteria presented in section 8.1. In the plots of error estimates each state has a bar plot related to it which gives the average error over one hundredth of the simulation time.

8.3.1 Accuracy

The accuracy of the filters is determined by how close the estimated trajectory is to the real trajectory. From figure 8.1 it can be seen that the EKF estimates the error to a accuracy in north- east position to less than $40cm$. The depth estimate is very accurate since it is based on a pressure-depth sensor and not on the HiPAP depth measurement. The velocity accuracy in figure 8.4 is very good with errors less than $5mm/s$. The error seen in the plot results from some zero mean white noise that passes through the filter. The attitude estimate for the EKF is also accurate, see figure 8.6. In the attitude estimate the oscillations in the start of the simulations are caused by the initialization of the state covariance matrix. When the elements corresponding to the attitude error has converged to the right magnitude region, the estimate is smooth and the error is less than $0.1°$ for all three dimensions. It can also be seen that the error in yaw(ψ) is larger than that of roll(ϕ) and pitch(θ), this is because there is more noise in the heading fluxgate compass than in the tilt sensor which introduces more uncertainty in this estimate.

Figure 8.2 shows the position error when using the unscented Kalman filter. The position error for the UKF is considerably lower than for the EKF. The error in north- east position to less than $6cm$, and the depth error is less than $4cm$. The velocity estimate for the EKF and the UKF does not differ at all, see figure 8.4 and 8.5. Both the accuracy and noise levels are the same. The attitude estimates for the UKF also show the influence of the converging state covariance matrix as with the EKF. The errors for the UKF are of the same magnitude as for the EKF.

The position accuracy of the nonlinear observer can be seen in figure 8.3. The position error is on average about the same as the unscented Kalman filter for this simulation, but it should be noted that there are impulses in the error every 5 sec. These spikes are induced by the position measurements from the HiPAP, and can also be seen in the velocity estimate. These spikes are a result of the low frequency

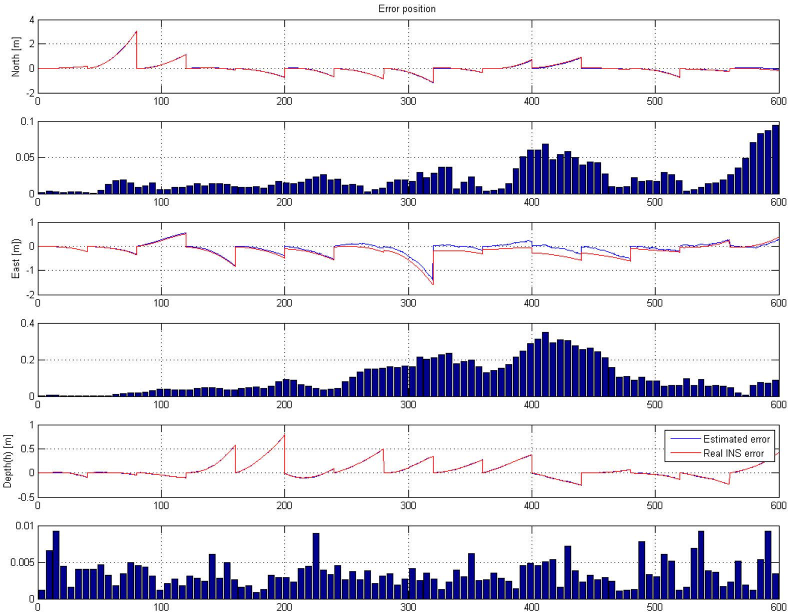

Figure 8.1: Position error with EKF in simulation 1

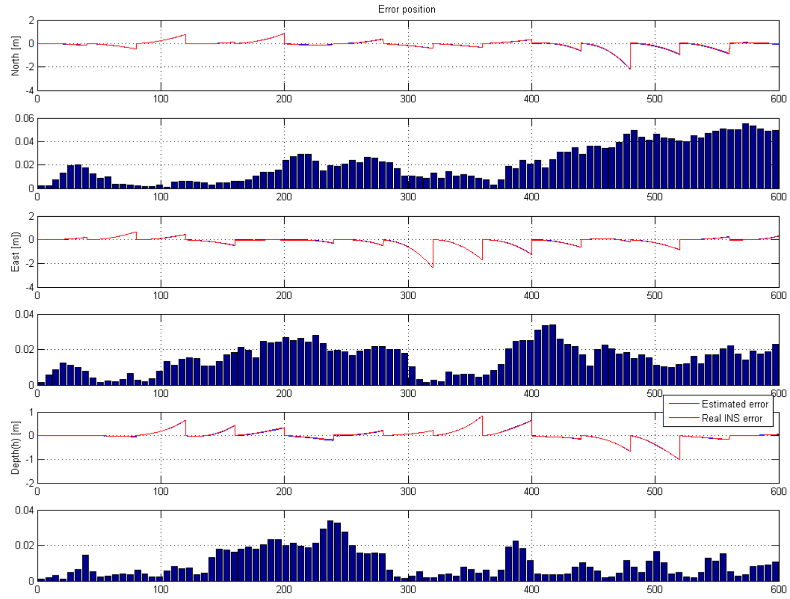

Figure 8.2: Position error with UKF in simulation 1

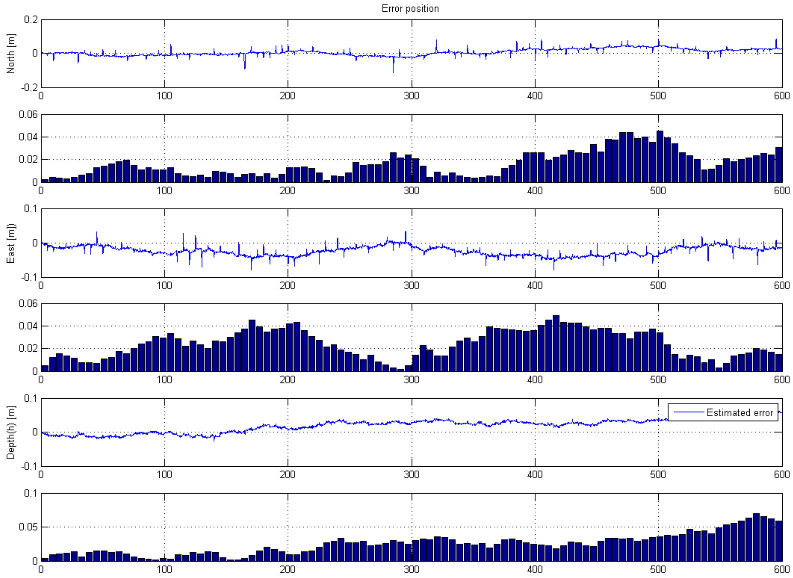

Figure 8.3: Position error with Nonlinear observer in simulation 1

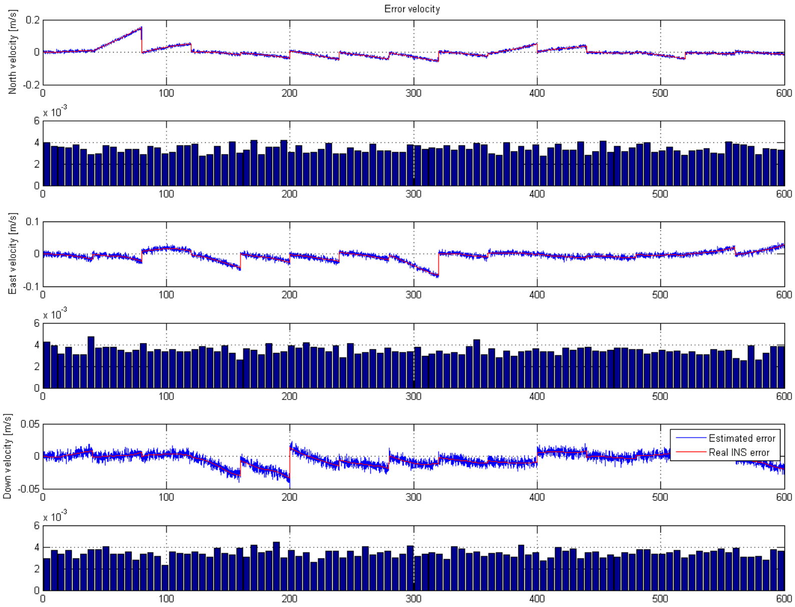

Figure 8.4: Velocity error with EKF in simulation 1

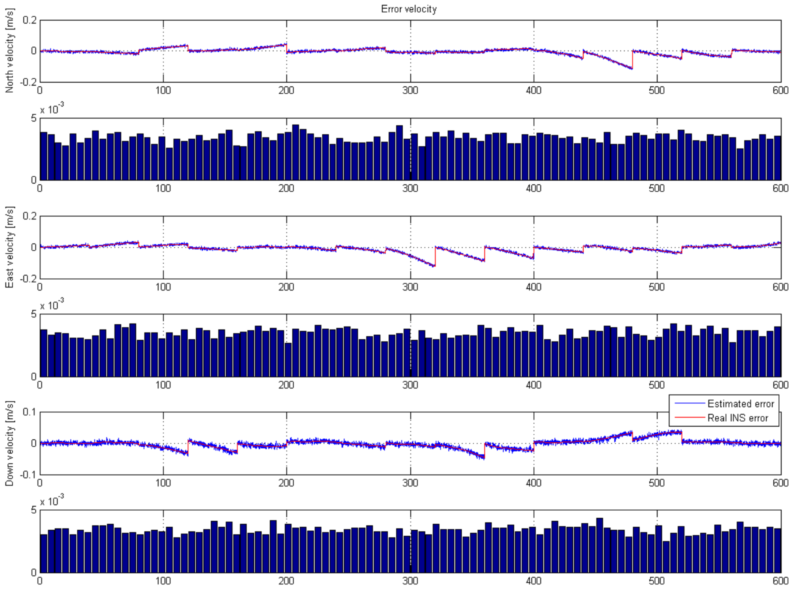

Figure 8.5: Velocity error with UKF in simulation 1

of the HiPAP measurement which leads to every time a position measurement is received the HiPAP bias has changed and an error is introduced in the position estimate. This error is feed back to velocity estimate (equation 5.68), and are seen as spikes since the velocity observer quickly reduces this error. From figure 8.8 it can be seen that the estimate of attitude is very noisy. This is caused by the fact that the nonlinear observer, as opposed to the Kalman filters, has no information of the noise of the measurements to remove measurement noise. The overall attitude and velocity error is limited and quite small, despite of the white noise in the estimate.

Simulation 1 shows that the position accuracy of the nonlinear observer and the UKF is slightly better than the performance experienced with the EKF. This can also be seen by looking at the position mean and maximum errors for simulation 1 (appendix B.1) where it can be seen that the nonlinear observer and UKF has about the same values while EKF has much larger values. The Kalman filters perform very much the same for velocity and attitude estimation. The nonlinear observer gives an attitude and velocity estimate with more noise than the KFs. This can be seen by investigating the standard deviation of the velocity and attitude errors (appendix B.1) which are much higher than for the KFs. For more details about the statistical properties of the filters for simulation 1, see appendix B.1.

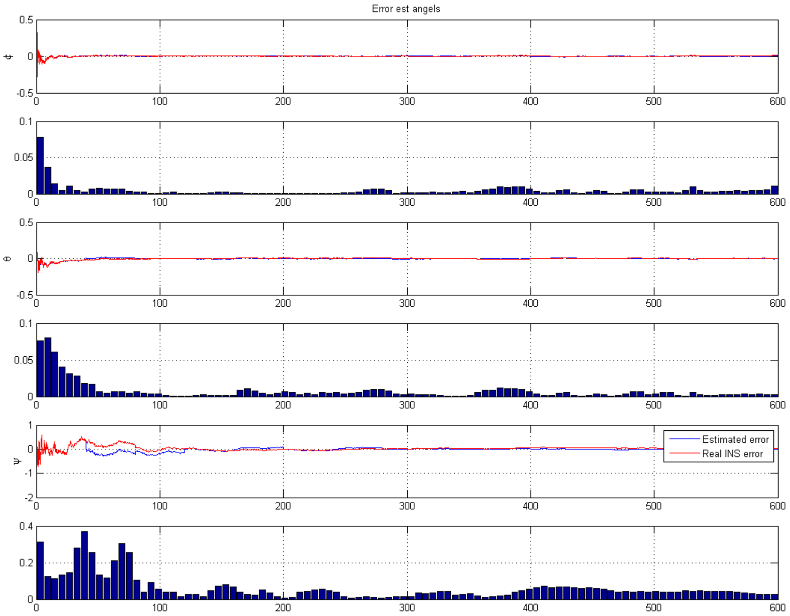

Figure 8.6: Attitude error with EKF in simulation 1

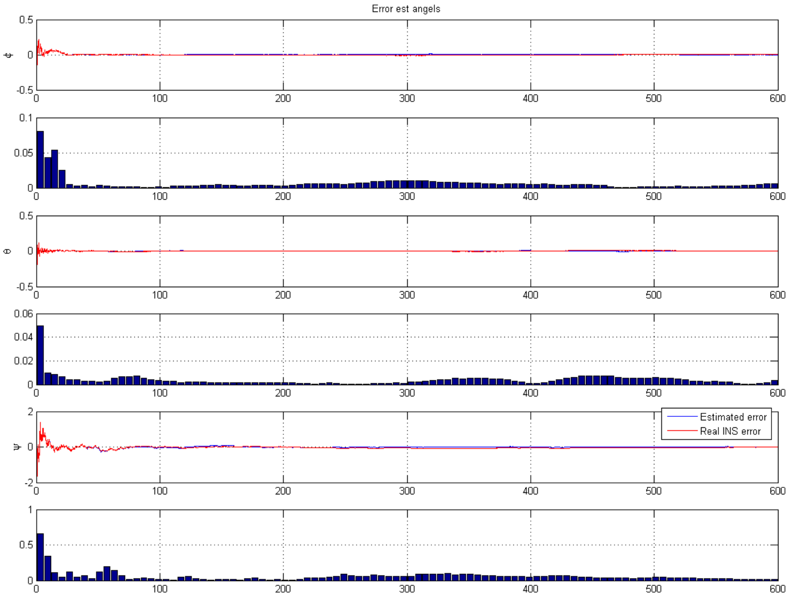

Figure 8.7: Attitude error with UKF in simulation 1

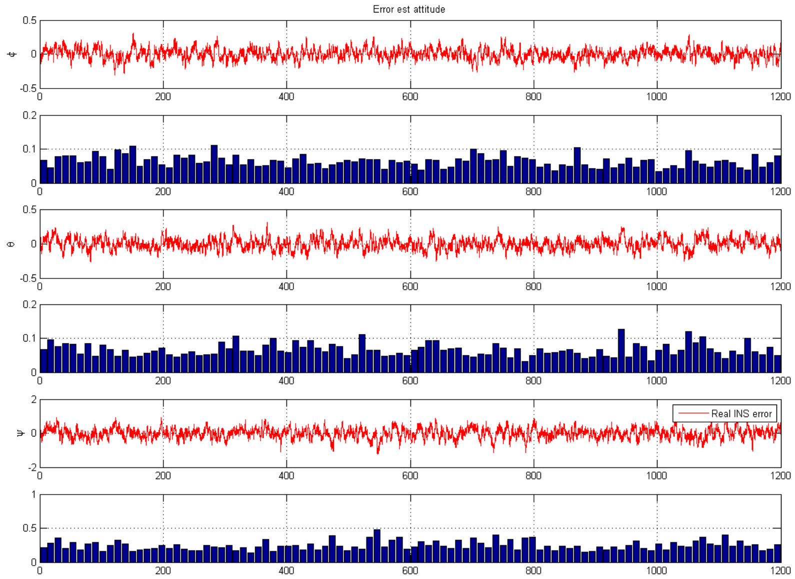

Figure 8.8: Attitude error with Nonlinear observer in simulation 1

8.3.2 Robustness

Robustness to noise

In simulation 2 the noise characteristics of the filters are different than that of the generated noise. The magnitude of the white noise driving the Markov processes modeling the IMU drift is increased, and the filter must track faster changes in the error. Figure 8.9 shows that the north-east position error is around $2m$ for the EKF. Even though the error in the INS grows faster, the filter still manage to track the error development, and the total error experienced by the user is still relatively small. The velocity error in figure 8.10 is larger than in simulation 1 to about a factor of ten. This is mostly caused by the increased white noise for the DVL passing trough the estimates without being filtered. In attitude it is once again seen oscillations due to the initial error in the state covariance matrix. These oscillations settles quickly, and the estimate of roll(ϕ) and pitch(θ) are accurate within $0.2°$. The yaw(ψ) estimate again has a larger error of the same reason mentioned for simulation 1.

The UKF has a slightly more accurate position estimate than the EKF in simulation 2. From figure 8.11 it can be seen that the north-east position error is limited to just over $1m$. This difference can be argued to the randomness of the noise, but running the simulations numerous times show that on average the accuracy is slightly better with the UKF. A reason why the UKF seemingly performs better during more noisy conditions is that the nonlinearities of the system become more dominant. In section

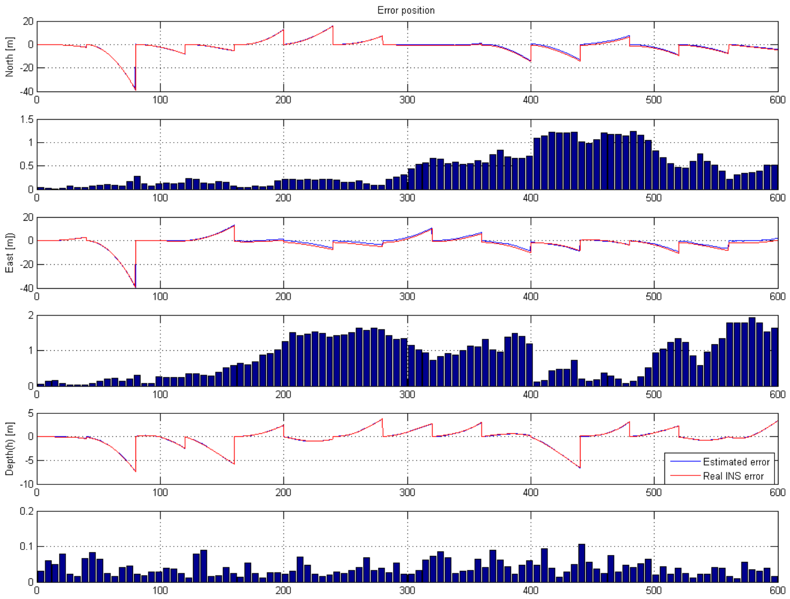

Figure 8.9: Position error with EKF in simulation 2

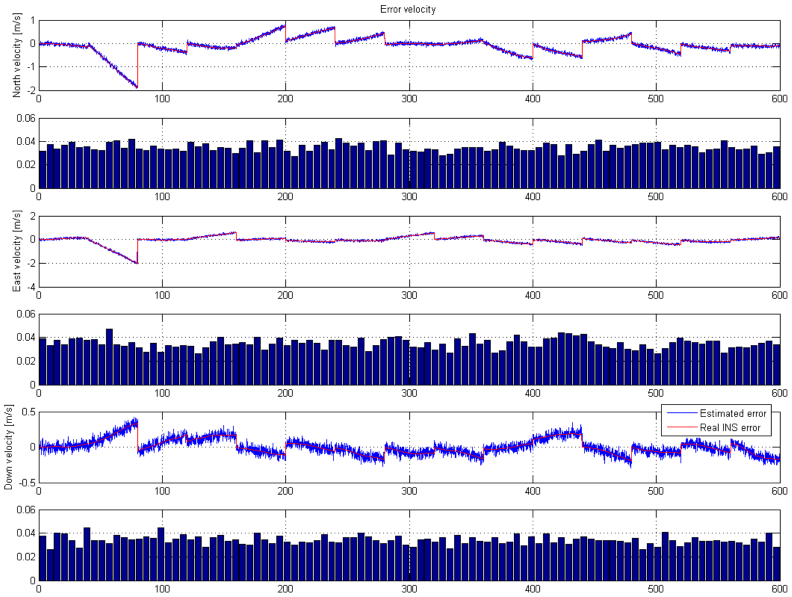

Figure 8.10: Velocity error with EKF in simulation 2

5.2.3 it is argued why the UKF is better suited for estimation of highly nonlinear systems. The accuracy of the velocity and attitude for the UKF in simulation 2 was, as in simulation 1, the same as with the EKF.

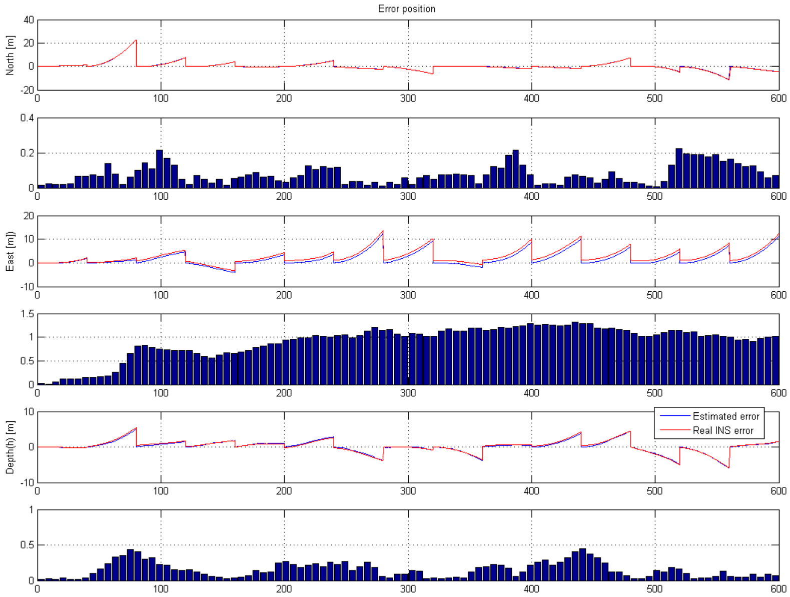

Figure 8.11: Position error with UKF in simulation 2

What should be noted in simulation 2 is that the magnitude of the white noise on all estimates is larger than in simulation 1. The reason for this is that the KFs believe that the magnitude of the measurement noises still are as specified in the data sheets, while they in fact are ten times larger. In section 6.6 it is argued that this gives a more noisy estimate because too much emphasis is put on the noisy measurement. This is especially apparent in the velocity measurement because the DVL measurement noise variance defined in the $\mathbf{R}$ matrix is relatively low(compared to the HiPAP). The Kalman-gain for the DVL measurement is therefore relatively higher than for the HiPAP measurement, which gives higher noise sensitivity. Although the noise is more apparent on the velocity estimate it is introduced in all estimates. The performance of the two KFs are equally good in the velocity estimate.

As in simulation 1 the estimates in simulation 2 are very noisy for the nonlinear observer. The error in position is increased by a factor of ten from simulation 1 (see figure 8.12). This is the same factor that all noise parameters where multiplied by in simulation 2. The spikes in the position estimate are larger in simulation 2. This is caused by more noise on the HiPAP measurements and that the HiPAP bias changes more between two consecutive measurements. It can be seen from figure 8.12 that the largest spike in north position is almost 1 meter. The largest position error, when not taking the spikes into account, is around $50cm$ in east position. The noise is also evident in attitude estimate in figure 8.13. The depth estimate of the nonlinear observer is worse than that of the KFs, this is caused by the fact that the nonlinear observer does not distinguish between the good depth measurement end the not so good HiPAP position measurement. The Kalman filter takes into account that the depth measurement is in fact very accurate. The larger error in depth can be reduced by tuning the observer gains. The nonlinear observer is still stable when the measurements are degraded, and all errors are bounded in the simulation.

Simulation 2 shows that the nonlinear observer achieves the best grade of position accuracy when the noise is increased. Slightly better than the UKF which again is better than the EKF even though the UKF has a higher mean in east direction (see appendix B.1). From the standard deviations it can be seen that in the velocity estimate the UKF and the EKF performs with about the same noise and level of accuracy. The nonlinear observers suffers from severely increased spikes in the velocity. The attitude estimate is best in the Kalman filters while much noise degrades the attitude in the nonlinear observer. For more details about the statistical properties of the filters for simulation 2, see appendix B.1.

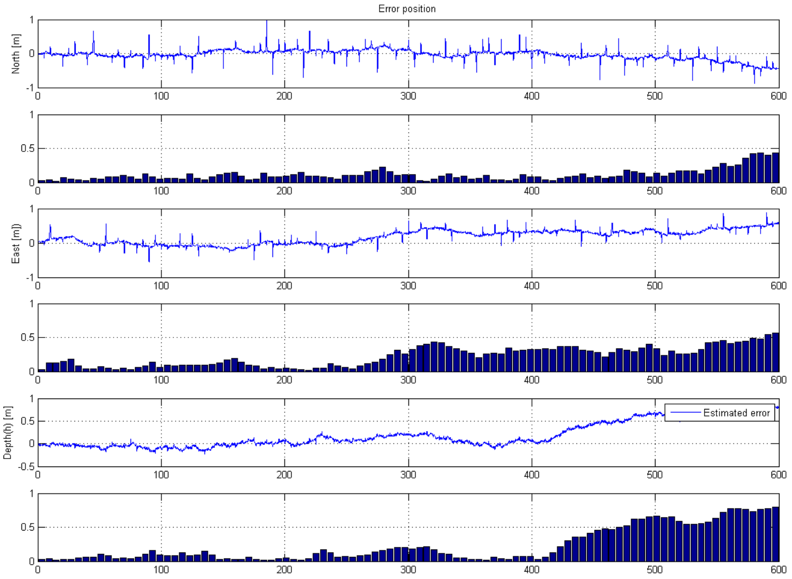

Figure 8.12: Position error with Nonlinear observer in simulation 2

Robustness to loss of external measurements

In simulation 3 robustness to loss of external measurements was tested. First the loss of HiPAP was tested (from $200 - 800s$). This resulted in very little influence on the estimates in any of the sensor integrators. Although there is no position correction to

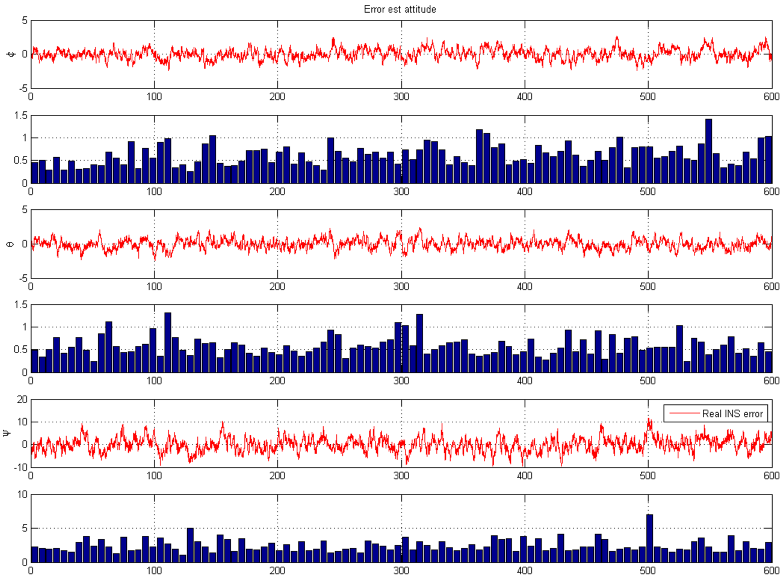

Figure 8.13: Attitude error with Nonlinear observer in simulation 2

limit the position error there seems to be very little drift in the position. This is due to the good velocity estimate which significantly reduces the position drift. If figures 8.14 and 8.15 are examined a random walk in the position can be seen. If a longer HiPAP loss was tested a larger position drift could be seen in all observers. The error growth in position is a result of the noise in the velocity estimate which is integrated to give position error. In the other states (velocity and attitude) no degradation of the estimates due to HiPAP loss is observed. For the nonlinear observer it is worth noticing that when the hydro acoustic position measurement is lost, the spikes are no longer evident in the position and velocity estimate. For the nonlinear observer the attitude is not influenced by the HiPAP loss since the attitude and position/ velocity observers are decoupled.

The next test was loss of DVL for $600s$(from $1000 - 1600s$). As can be seen from figures 8.16-8.21 this has much greater influence on the estimates than the HiPAP loss. Since there is no longer a secondary velocity measurement the velocity estimate must rely on the position for correction. Noise in the position estimate is magnified due to derivation, which makes the velocity estimate more noisy. It is also more difficult to estimate the accelerometer bias, such that more of $\mathbf{b}_{f_{IMU}}$ is integrated to give velocity and position error.

For the two Kalman filters the position error however is bounded since the HiPAP is still available, which again results in bounded velocity error. To estimate the HiPAP bias correctly the filters are however dependent on the velocity measurement, since this is no longer available the position error may grow large. During

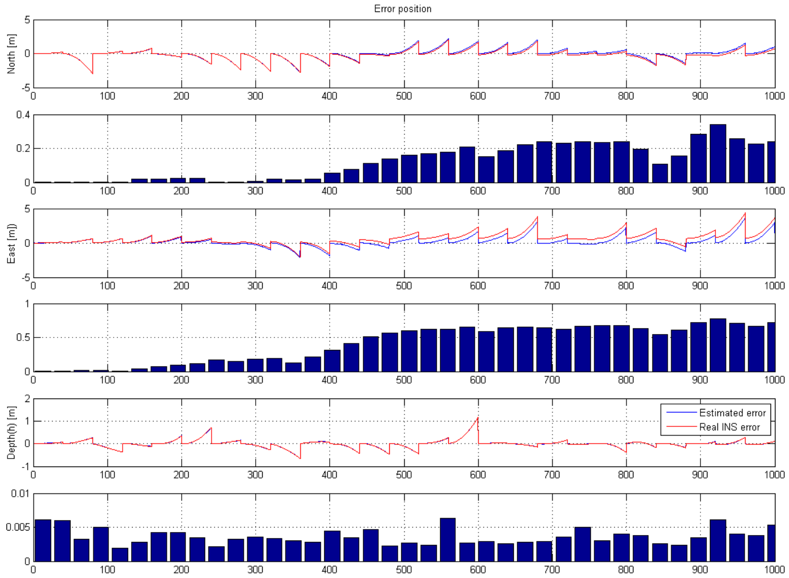

Figure 8.14: Position error with EKF in the first $1000s$ of simulation 3

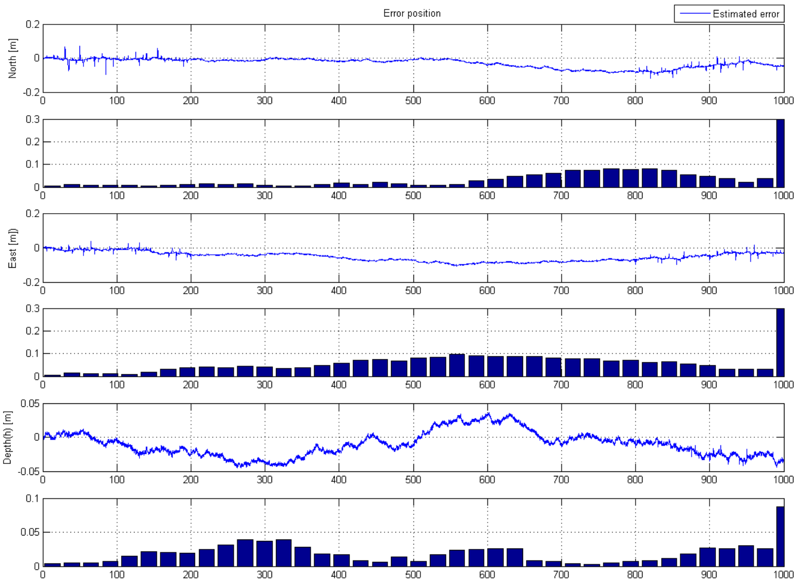

Figure 8.15: Position error with Nonlinear observer in the first $1000s$ of simulation 3

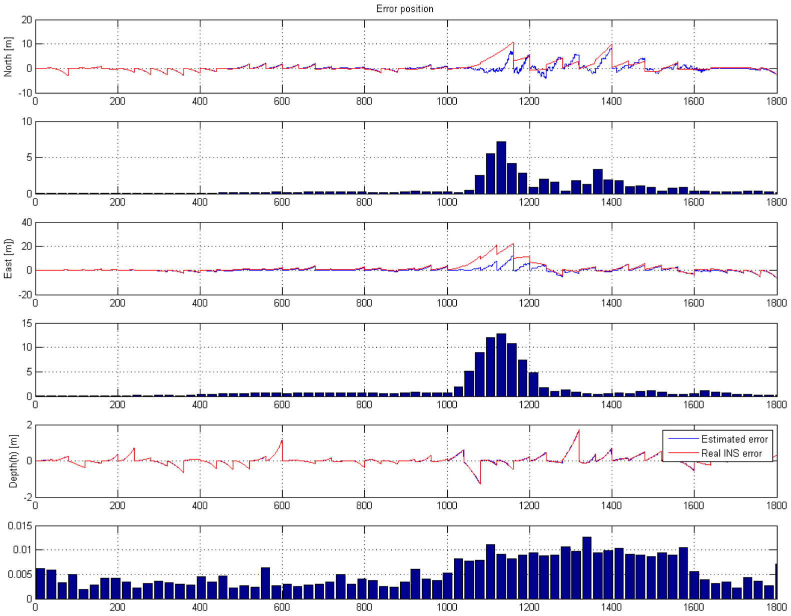

Figure 8.16: Position error with EKF in the first $1800s$ of simulation 3

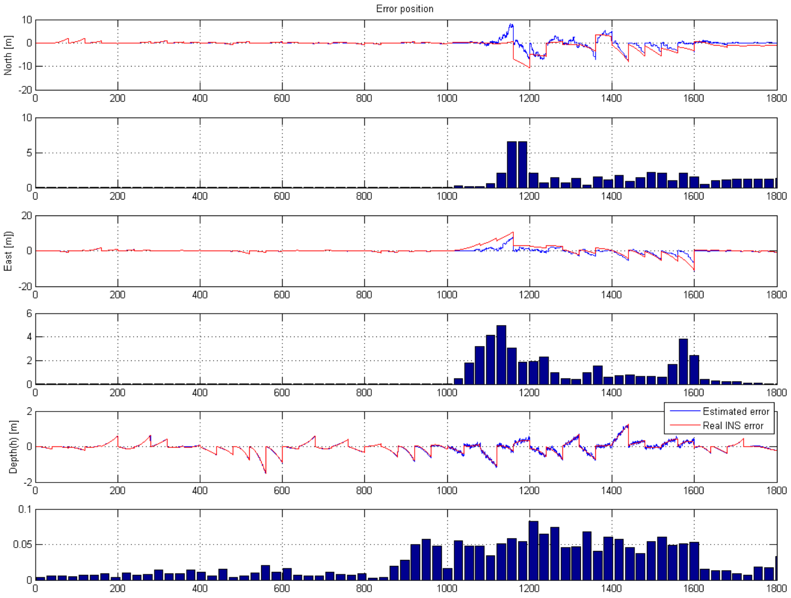

Figure 8.17: Position error with UKF in the first $1800s$ of simulation 3

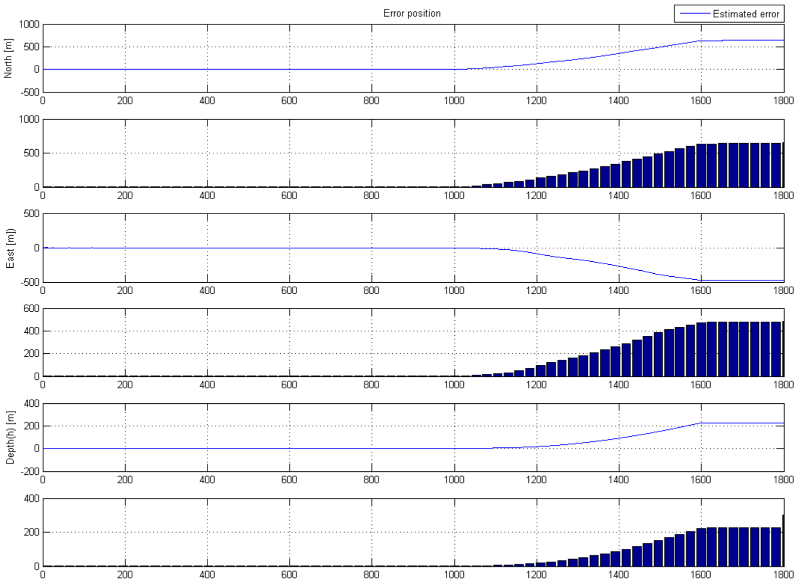

Figure 8.18: Position error with nonlinear observer in the first $1800s$ of simulation 3

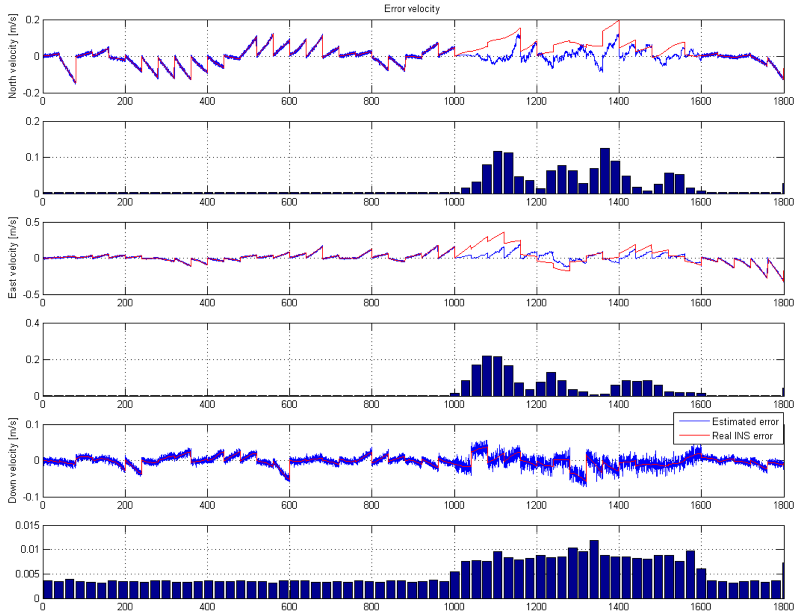

Figure 8.19: Velocity error with EKF in the first $1800s$ of simulation 3

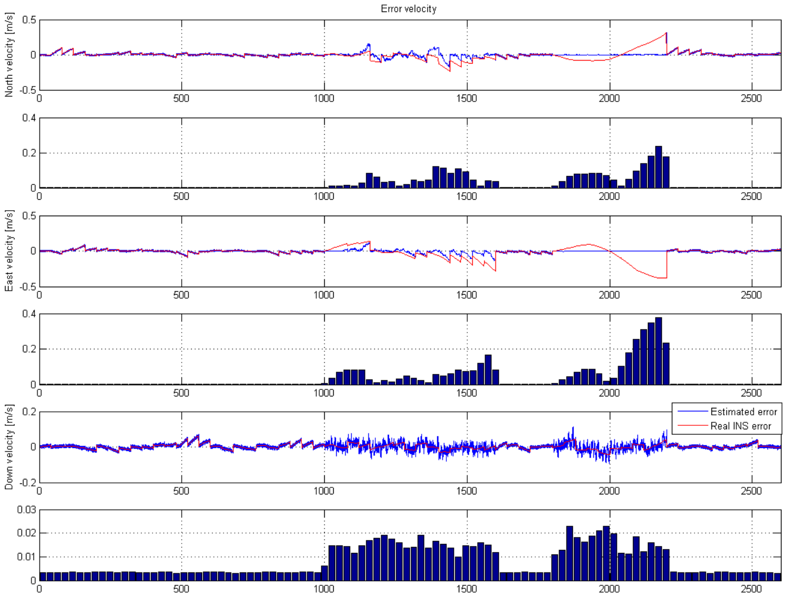

Figure 8.20: Velocity error with UKF in simulation 3

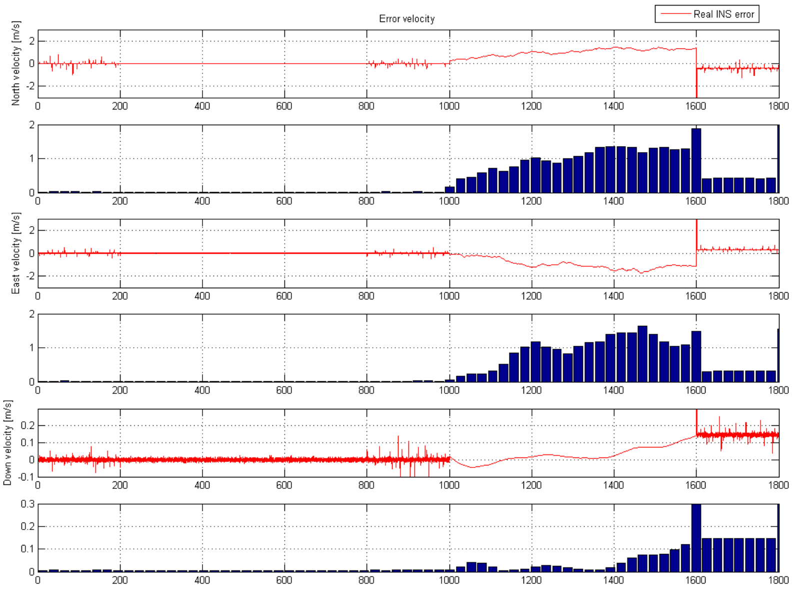

Figure 8.21: Velocity error with nonlinear observer in the first $1800s$ of simulation 3

normal operation the HiPAP measurements are used to make sure that large errors in the position estimate are corrected. The position estimate therefore rely mostly on the DVL measurement for small corrections in velocity to limit the position error growth. The reason why the DVL is so important to the position estimate is that it helps give a good estimate of the HiPAP bias. This fact leads to a trade-off between robustness and accuracy in the case of DVL loss for both Kalman filters. If the measurement covariance of the HiPAP is chosen large much emphasis is put on the DVL measurement and good accuracy is achieved when all measurements are available. If DVL is lost the filter's most important measurement is lost, and the performance is significantly degraded by an even greater position and velocity error. The error during the DVL loss would be smaller if the HiPAP was emphasized more. The attitude estimate is also significantly effected by the DVL dropout. From the navigation equations (in chapter 3) the connection between attitude and velocity can be seen. When there is an error in velocity it will be seen as an error in the rotation matrix $\mathbf{R}_b^n$. Changes will be made to correct this and uncertainty in the attitude estimate is introduced. When comparing the performance of the two Kalman filters it can be seen that the errors in the estimates due to DVL loss are about the same for the two.

The DVL loss has an even greater impact on the nonlinear observer. Figure 8.21 shows that the velocity estimate in the nonlinear observer is not corrected with only position measurements. The velocity error quickly induces a position error. The position error is not corrected by the HiPAP measurements, and the error grows until the DVL returns. During the $600s$ DVL drop out the position error grows to between $500 - 1000$ meters in all three directions. The nonlinear observer needs the velocity estimate to be correct in order to estimate the HiPAP bias, and when the DVL is lost the estimated HiPAP bias goes to zero. The HiPAP is not used for correcting the velocity during DVL loss, this is to ensure that the observer does not become unstable during the loss. It should also be noted that in contrast to the KF, the nonlinear observer does not recover fast when the DVL returns. This is the case for both position and velocity. The position error stabilizes when the DVL returns, but is not reduced towards zero. The reason for this is that the observer does not manage to distinguish between HiPAP bias and position error. All position error is regarded HiPAP bias such that the term $\tilde{\mathbf{p}}_{en_{HiPAP}} - \hat{\mathbf{b}}_{p_{HiPAP}}$ from equation 5.69 still evaluates to zero. This is clearly a great disadvantage of this observer. The attitude is not influenced by the DVL dropout. The nonlinear position and velocity observer does not handle a DVL dropout in this implementation.

The last part of simulation 3 is to test how the filters handle losing all other secondary measurement except the pressure based depth meter, and how well the estimates recovers when the measurements are returned. The nonlinear observer will be treated separately.

When all measurements are lost the estimates go to zero, hence the integrated states are the same as the uncorrected INS-states. Since there is no bias correction the errors begin to grow unbounded, except the depth and down velocity that still has the depth meter for correction.

When measurements are returned the estimates in the KFs recovers quickly (uses only a few time steps to recover). The reason why this recovery is so fast is that during the outage the estimates are zero. This gives a large difference between the measured and estimated states when the measurements are returned. The parameters of the state covariance matrix $\bar{X}$ become large, hence much weight is put on the measurements in the time steps following the measurement recovery and the large error is quickly reduced. This can also be seen directly by investigating the values of the Kalman gain matrix K when the measurements are returned.

When all measurements are lost the nonlinear velocity and position drifts uncontrollably. The position error grows exponentially and as for the case when the DVL was lost, the position correction is very slow when the measurements return of the same reason as for the DVL loss. This is however with the chosen gains. If the gains are increased when the error is large, the recovery could be significantly faster. Increasing the gains is a trade-off between accuracy and the recovery rate after large errors. A way to solve this problem could be to make an adaptive observer such that the gains are of the proper magnitude according to the magnitude of the measured error. This task is outside the scope of this report, and has not been solved. When the measurements return the velocity error makes a leap (this can be seen in figure B.1 in appendix B.2). The reason for this is the position error element in the velocity observer equation. Since the position error has grown large in the dropout period, the position error term gets a very large value when the HiPAP returns. This error is transferred to the velocity. The attitude observer drifts according to the error in the gyro measurement. When the compass and tilt sensors returns the attitude observer recovers quickly.

In simulation 3 the UKF and EKF had about the same performance with limited error for both HiPAP and DVL loss followed by a fast recovery in the estimates when measurements returned. The nonlinear observer only handled the HiPAP loss. When the DVL was lost both position and velocity grew very large and did not recover upon measurement return.

Robustness to IMU loss

In simulation 4 the filters were tested with respect to robustness to an IMU loss for 200 seconds (from $200 - 400s$). It was in this simulation the greatest difference in performance between the two KFs was seen. When looking at the position estimates in figures 8.22 and 8.23 it can be seen that the error in the EKF is about 10 times larger than in the UKF. For the other states (velocity and attitude) the two filters again has very similar performance. The problem in the EKF is that it estimates the position error as a HiPAP bias (same problem as for the nonlinear observer when experiencing DVL loss), which leads to an error in the resulting position estimate. The filter is then unable to separate between what is position error and what is HiPAP

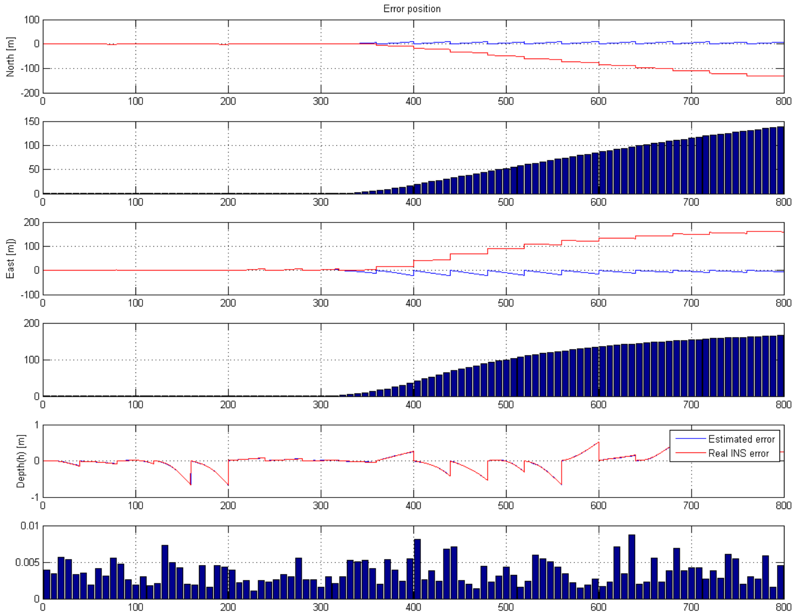

Figure 8.22: Position error with EKF in simulation 4

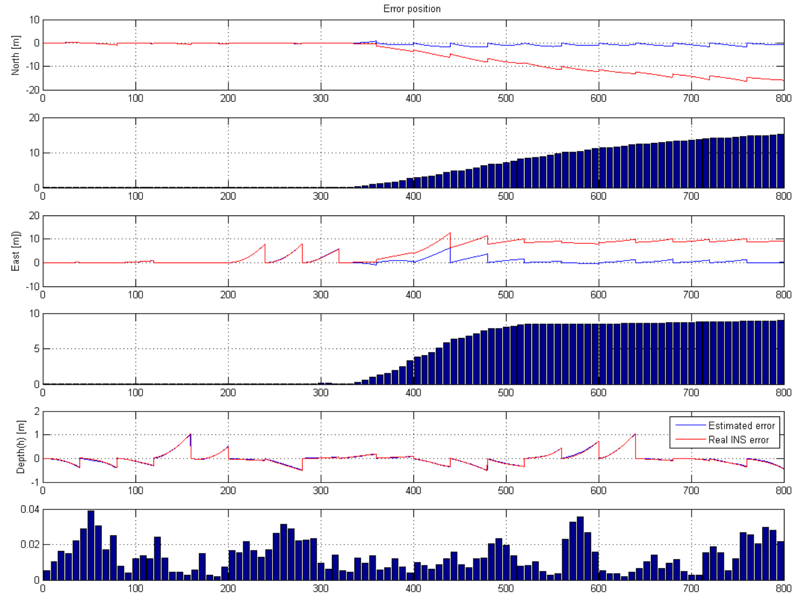

Figure 8.23: Position error with UKF in simulation 4

bias. This problem is not so evident in the UKF. The reason for this difference is found in the covariance calculation. In chapter 5.2.3 it is argued that the unscented transform more accurately captures the real covariance of the system, hence the Kalman gain matrix K from the UKF utilizes the available measurements in a better way.

What needs to be noticed in simulation 4 is that the filters do not recover right away when the IMU measurements are returned. This gives an indication of how much the external measurements are weighted when they are all available, since there are no inertial measurements. The greatest error is found in the attitude. From the measurement specifications (see appendix C) it can be seen that the measurement noise in yaw(ψ) is larger than that for roll and pitch, hence the yaw measurement is weighted less than the other two Euler angles. When the gyro measurement is lost the filter only has the noisy compass for heading correction. The filter does not however put much emphasis on this measurement due to the large noise and does not manage to track the changes in heading. From figure 8.24 and 8.25 it can be seen that when the heading is held constant for a longer period, the estimate gradually converges towards the correct value. The position error observed in simulation 4 is caused by the error in the yaw estimate. The position error of simulation 4 would be significantly less if the vehicle was not rotating in yaw during the IMU dropout. The position errors has to be corrected by the HiPAP measurement which is not weighted much, and is only received every 5th second. Since it takes long before the yaw error

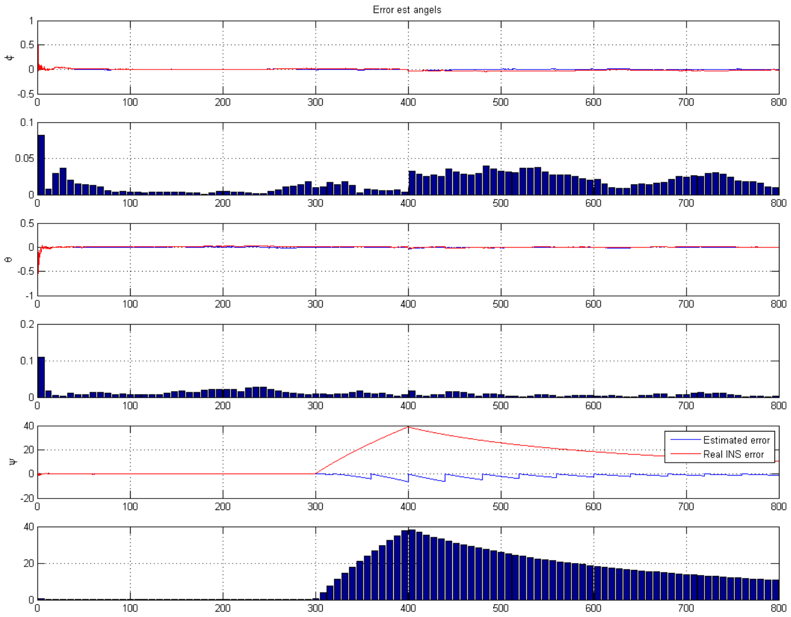

Figure 8.24: Attitude error with EKF in simulation 4

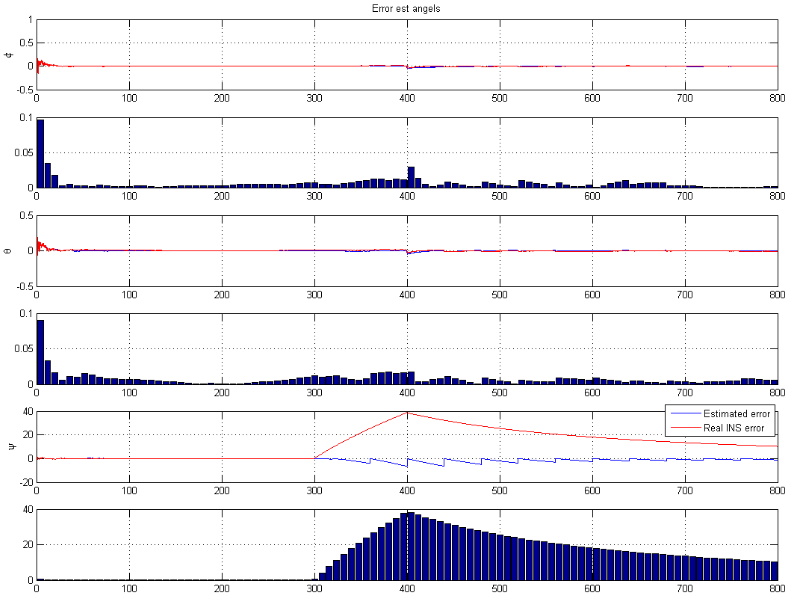

Figure 8.25: Attitude error with UKF in simulation 4

is corrected it also takes long before the position becomes correct. If the external attitude and position estimates were weighted more (had smaller elements in the $\mathbf{R}$ matrix) the filter would converge more rapidly, but this would lead to more noise on the attitude estimates. Since the velocity estimate is less influenced by IMU loss it can be argued that the DVL is the external measurement which is emphasized the most.

The results from the above discussion are in good accordance with the results from simulation 3 where it could be seen that the quality of the estimates were mostly influenced by a DVL loss.

The nonlinear observer performs very well during IMU dropout. The loss of a measurement seems to have little influence on the error, which is not increased during the 200 seconds dropout. The errors in simulation 4 are actually comparable to the errors of the nonlinear observer in simulation 1. The largest change in error is visible in the yaw angle, figure 8.26, where the error grows to about $1°$. This is a result of that the nonlinear observer utilizes a decoupled attitude observer, such that the secondary attitude measurement is much more emphasized in the nonlinear observer than it is in the Kalman filters.

Simulation 4 showed that the EKF was most influenced by the IMU drop out by great errors in position and attitude. UKF also had problems estimating the attitude but had a much smaller position error than the EKF. The nonlinear observer performed very good during this simulation with accuracy comparable to that of all

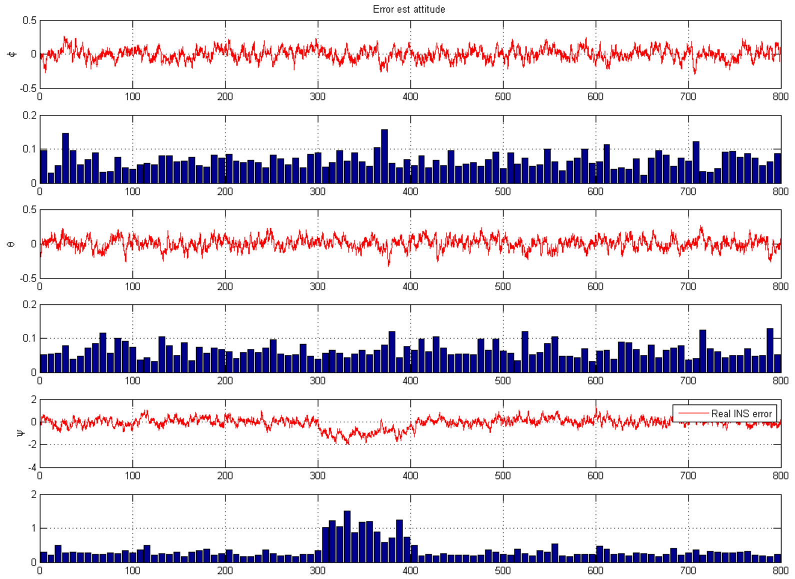

Figure 8.26: Attitude error with nonlinear observer in simulation 4

measurements available.

8.3.3 Sensor latency

The time indexing of measurements was implemented and tested on both Kalman filters. The filters where run with the same noise specifications as in simulation 1, with a $5s$ delay in position and $0.5s$ delay in velocity measurements. The statistics of the tests are presented in appendix B. The tests showed that the delayed measurements was able to limit the errors as in the previous simulations. The accuracy was slightly degraded, a fact that can be explained by the change in the INS error over the latency period. The mean of the error stayed low, which is an indication that the estimate is still only influenced by a small bias. In delaying the velocity measurement from the DVL, the filter performed satisfactory up to a certain limit of delay. The difference in impact of the delays is due to the fact that the velocity error dynamics are faster than the position error dynamics, and that the DVL measurement has a greater influence on the estimate of the reasons mentioned in section 8.3.2.

The implementation of the nonlinear observer in this work was not stable with a delay in velocity measurement. Delay in the position measurement had, as in the KFs, little effect on accuracy compared to simulation 1.

8.3.4 Runtime

For all simulations the nonlinear observer gain distinction in the runtime evaluation. The nonlinear observer performs about the same number of calculations as the INS navigation equations. In contrast the Kalman filters requires the navigation equations must be run in parallel with the filtering algorithms. This is also evident in the run time, which is greatly increased for both Kalman filter implementations. LaViola (2003) experienced a great difference in runtime between EKF and UKF when estimating quaternion motion. This is in accordance with our results, where the UKF used on average $9ms$ for each filter iteration while the EKF used $1ms$. [1]

8.3.5 Graceful degradation

The Kalman filters use a separate calculation of the inertial navigation equations and correct the error in these by the factor calculated in the filter. If there is no measurement input to the filter all the error states in the filter are estimated to zero. This means that the Kalman filter has no influence over the inertial navigation equations if there is no measurements. The INS error is then limited to the integral or double integral, for velocity and position respectively, of the IMU error. By examine the observer equations for the nonlinear observer in section 5.3 it is evident that when all measurements are lost, the resulting equations are equal to the regular INS equations from chapter 3.1. This means that the nonlinear observer should also have an error limited by the IMU error when all measurements are lost. The difference between the KFs and the nonlinear observer is seen when measurements are returned.

[1] Runtime is dependent on the computer hardware. These simulations where run on a Pentium D CPU $2.8Ghz$ with 2GB of RAM.

8.4 Pros and Cons

A summary of the pros and cons of the different filter methods are presented in table 8.1.

	EKF	UKF	Nonlinear observer
Pros	Well known method Robust Fast runtime Graceful degredation Easy to account for delays	Very robust Accuracy Easy to account for delays Graceful degredation No linearization	Intuitiv implementation Computational efficient Globally stable (continous) Accuracy No linearization
Cons	Not globally stable Not intuitive implementation Lower accuracy Calculation of Jacobian	Not intuitive implementation Not globally stable Computationally intensive	Sensitive to measurement loss Static gains Difficult to get good tuning Unstable with DVL delay

Table 8.1: Pros and cons for the different sensor integrator methods

Chapter 9

Experimental verification of filter

In order to test the sensor integrator on real measurement data a data series is collected. This is done with the use of an IMU and a GPS receiver installed in a car. The filters are designed with 3 supporting measurements (velocity, position and attitude), with the possibility of neglecting either one in the case of absent or significantly degraded measurements. The IMU used is a MTi IMU from Xsens Motion Technologies (Xsens-Technologies n.d.) This measures acceleration in 3 DOF, angular velocity in 3D as well as earth-magnetic field sensor in 3D. As a secondary position measurement a Garmin 16A GPS receiver was used.

With this setup the position from the GPS and attitude measurement from the earth-magnetic field sensor in Euler angles (no velocity measurement available) were used to support the acceleration and gyro measurements of the IMU to give position, velocity and attitude estimates. Since a GPS was used and not an HPR, there is no HPR bias to account for in the estimates. The goal of this test was to use the measurements to aid the IMU to give estimates with limited error.

The IMU used in these tests is not of navigation quality, as opposed to the data used in the simulations in chapter 8. The drift in both velocity and attitude is therefore considerably worse than what can be assumed in a IMU of navigation quality.

Based on the simulations done in chapter 8 the real data test is performed merely on the Kalman filter observers. This is because the nonlinear observer stability proof in section 5.3 is only valid with all measurements available. From the real data test there was no velocity measurement, which simulation 3 from section 8.3.2 showed is crucial for the nonlinear observer. The filters where not run in real time, but the data was gathered and post processed. The same data series was used for both Kalman filters.

9.1 Evaluation

Since these data series are based on real data, it is not possible to compare the estimates with real states or errors. The trajectories can only be compared with the GPS samples. Therefore the filters can only be evaluated trough comparing between the different filter solutions.

The IMU data was sampled by Matlab at $50Hz$ (real IMU frequency was $100Hz$). The GPS updates at $1Hz$, this means that there is only a GPS measurement at every $50th$ iteration of the INS.

9.2 Results

It is evident that the IMU is not of the grade that is useful for navigation. But even with this IMU the filter structure can estimate and compensate for the bias drift in the IMU. The problem with using an IMU of this quality is the robustness. As mentioned in chapter 8.3.2 the filter can not estimate the errors if the supporting measurements are lost (in this case the GPS). Hence the robustness of both position and velocity estimates in the case of GPS dropout is directly dependant of the IMU quality.

The trajectory tested was one where the IMU traveled almost the same trajectory two times (figure 9.1). As can be seen the estimates are far more noisy with this IMU. A GPS measurement is received once every second, and it can be seen clearly that the drift caused by the IMU bias is significant (up to 2 meters). When it comes to performance the two filters once again has similar performance for all state estimates. From figures 9.2 and 9.3 it is not possible to distinguish between the two filters for north and east velocity and attitude estimates. For the down velocity the UKF has more deflections than the EKF, these are also transmitted to the elevation estimate.

The main result from this experiment was to verify that the filters tested handles real data satisfactory. By this we mean that the errors were kept limited during very noisy conditions with only position and attitude measurement.

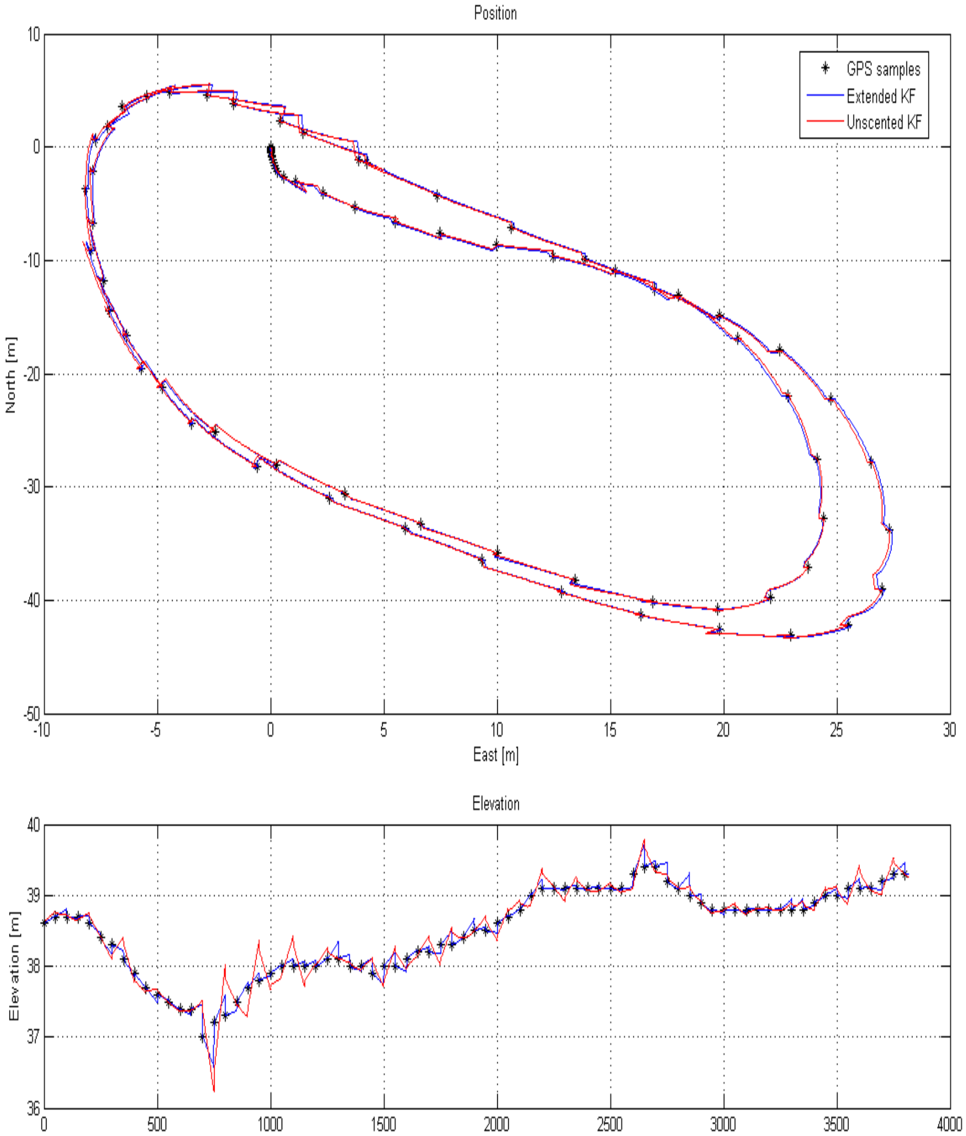

Figure 9.1: Position estimate from real data test

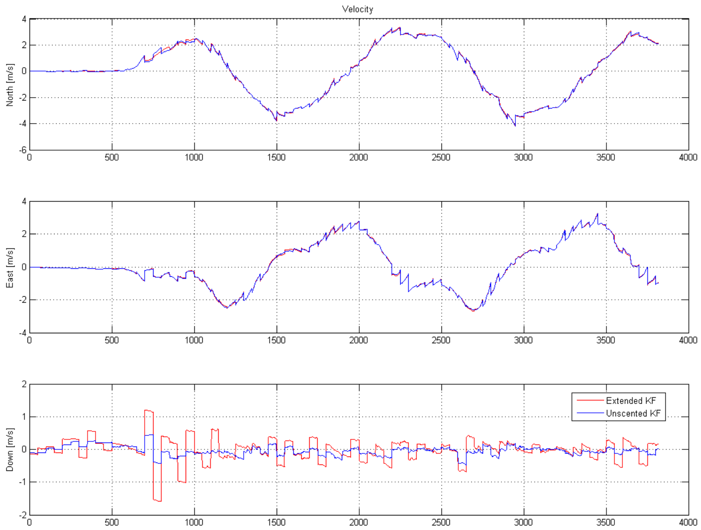

Figure 9.2: Velocity estimate from real data test

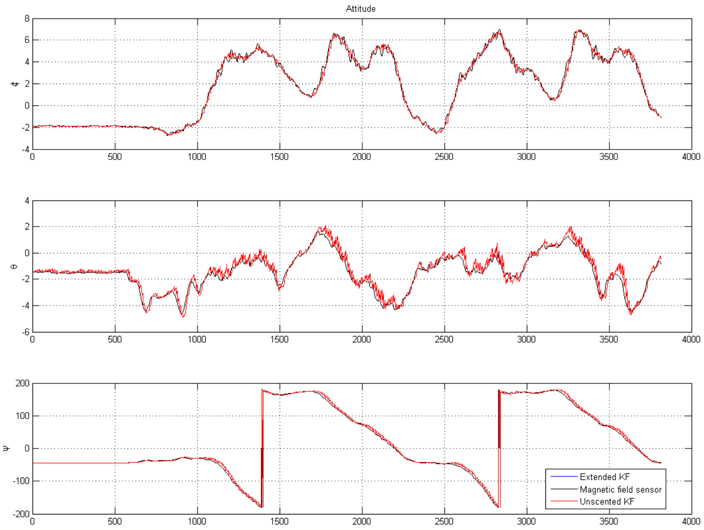

Figure 9.3: Attitude estimate from real data test

Chapter 10

Conclusions and suggested future work

10.1 Conclusion

In this work three different solutions has been presented to solve the integration between measurements for an underwater vehicle. The goal was to determine the position, velocity and attitude of the vehicle.

Simulations have shown that for the position estimate the nonlinear observer is slightly more accurate than that of the unscented Kalman filter both with correct and wrong estimates of noise characteristics. The extended Kalman filter provided the poorest degree of accuracy. The reason for this is that neither the UKF nor the nonlinear observer perform any kind of linearization as opposed to the EKF. This linearization lead to a loss of information about the dynamics of the system, hence the poorer estimate. For attitude and velocity all filters have good accuracy, but the nonlinear observer has no information about noise magnitude and can therefore not filter the white noise accurately such that more noise is visible on the estimates.

When it comes to loss of external measurements the two Kalman filters have proven more robust. All three observers handles HiPAP loss very well. In the case of DVL loss the Kalman filters suffers from degraded performance followed by fast recovery, the nonlinear observer does not manage to recover from the induced error because it estimates the position error as HiPAP bias. This is a great weakness of the nonlinear observer structure. The simulations performed showed that the robustness to sensor dropout (especially DVL) is greatly influenced by the quality of the IMU. For all the three observers tested, it is important that the IMU is of good quality if the risk of drop out of supporting measurements are considered high. Both Kalman filter solutions perform consistent estimates with only velocity and position measurements. But to increase observability and robustness it is recommended to also include a secondary attitude measurement.

The IMU is a very robust measurement unit. In the unlikely case of IMU dropout the nonlinear observer performed very well. Both Kalman filter had great problems with estimating the attitude. Because of the integrated structure of the Kalman filter the attitude measurement is not much emphasized in the Kalman filter. The nonlinear observer gains from having a decoupled attitude observer, this gives more emphasis on the attitude measurement, hence a better attitude estimate is achieved without IMU. The attitude error in the KFs also gives position error. This error is however ten times larger in EKF than in UKF.

It has been shown that the DVL is the most important of the secondary sensors since it has a high update frequency and relatively low measurement noise. It is also important to have a good velocity estimate since the attitude error, in the Kalman filters, is observable through the velocity. A good velocity estimate can not limit the position error, but decreases the position error drift rate considerably. A good velocity estimate is also essential for achieving a good estimate of the HiPAP bias.

The nonlinear observer is tuned trough the observer gains. This tuning needs to be changed depending on the measurement noise characteristics. Since the HiPAP bias error characteristics changes in different depths, the tuning also needs to change as the vessel descents. The observer gains are not directly proportional to the noise so this adaptive tuning is complicated. The Kalman filters are more easily tuned through the noise covariance matrices. The noise covariance elements can be modeled dependent on the depth, and therefore the tuning can change as a function of depth.

The Kalman filters was shown to work with a low grade IMU and no velocity measurements trough the use of real data series collected from an IMU and a GPS receiver. The real data test did not show any big difference in the UKF and EKF performance.

From these results the recommended method for sensor integration is the unscented Kalman filter. It has proven good accuracy and is also very robust to both measurement noise and loss of measurements.

10.2 Suggested future work

During the development of this work some issues that needs to be addressed further have arisen.

- Stability proof of EKF and UKF for sensor integration
- Develop conditions for stability of filter with delayed measurements.
- Designing a update law for the observer gains.
- Implementing the filters in the RovNav application.

Bibliography

Angrisani, L., D'Apuzzo, M. & Moriello, R. S. L. (2005), 'The unscented transform: a powerful tool for measurement uncertainty evaluation', *IEEE transactions on instrumentation and measurement* **55**, 737–743.

Balchen, J. G., Andresen, T. & Foss, B. (2004), *Reguleringsteknikk*, Institutt for teknisk kybernetikk.

Brown, R. G. & Hwang, P. Y. (1997), *Introduction to random signals and applied Kalman filtering*, John Wiley & Sons, Inc.

Burns, A. & Wellings, A. (2001), *Real- Time Systems and Programming Languages*, Pearson Education Limited.

Chen, C. (1999), *Linear System Theory and Design*, Oxford University Press.

Diebel, J. (2006), Representing attitude: Euler angles, unit quaternions, and rotation vectors, Technical report, Stanford University, California, U.S.A.

Egeland, O. & Gravdahl, J. T. (2002), *Modeling and simulation for automatic control*, Marine Cybernetics.

Farrell, J. & Barth, M. (1999), *The global positioning system and inertial navigation*, McGraw-Hill.

Foss, H. T. & Meland, E. T. (2006), Rovnav sensor integrator design. Projectwork autumn-06.

Fossen, T. I. (2002), *Marine Control Systems*, Marine Cybernetics.

Fossen, T. I. (2005), 'Marine gnc toolbox', Internet. http://www.cesos.ntnu.no/mss/MarineGNC.htm Available: 05-26-07.

Gade, K. (1997), Integrering av treghetsnavigasjon i en autonom undervannsfarkost, Master's thesis, The Norwegian University of Science and Technology (NTNU).

Gade, K. & Jalving, B. (1998), 'An aided navigation post processing filter for detailed seabed maping uuvs', *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology* pp. 19–25.

Healey, A. J. & Lienard, D. (1993), 'Multivariable sliding-mode control for autonomous diving and steering of unmanned underwater vehicles', *IEEE JOURNAL OF OCEANIC ENGINEERING* **18**, 327–339.

Hedrick, J. K. (2007), Control of nonlinear dynamic systems: Theory and applications. Lecture notes Control of Nonlinear Dynamic Systems, University of California, Berkeley.
Web: http://www.me.berkeley.edu/ME237/6_cont_obs.pdf
Prof. J. Karl Hedrick
5104 Etcheverry Hall
Email: khedrick@me.berkeley.edu
Available: 05.15.2007.

Henriksen, R. (1998), *Stokastiske systemer; analyse, estimering og regulering*, Institutt for teknisk kybernetikk, Norges teknisk-naturvitenskapelige univeristet.

Idsø, E. S. (1999), Biasestimering ved indirekte gps/tns integrasjon, Master's thesis, The Norwegian University of Science and Technology (NTNU).

Jalving, B. & Gade, K. (1998), 'Positioning accuracy for the hugin detailed seabed mapping uuv', *Oceans Conference Record (IEEE)* **1**, 108–112.

Jensen, E. (2006), Kongsberg hipap, Technical report, Kongsberg Maritime AS.

Julier, S. J. & Uhlmann, J. K. (1997), 'A new extension of the kalman filter to nonlinear systems', *Proceedings of SPIE - The International Society for Optical Engineering* **3068**, 182–193.

Julier, S., Uhlmann, J. & Durrant-Whyte, H. F. (2000), 'A new method for the nonlinear transformation of means and covariances in filters and estimators', *IEEE Transactions on Automatic Control* **45**.

Khalil, H. K. (2002), *Nonlinear Systems*, Prenctice Hall.

Larsen, T. D., Andersen, N. A. & Ravn, O. (1998), 'Incorporation of time delayed measurment in a discrete-time kalman filter', *Proceedings of the 37th IEEE conferance on Decision and Control* .

LaViola, J. J. (2003), 'A comparison of unscented and extended kalman filtering for quaternion motion', *Proceedings of the American Control Conference* **3**, 2435–2440.

Lawrence, A. (1993), *Modern Inertial Technology*, Springer Verlag.

Mandt, M., Gade, K. & Jalving, B. (n.d.), 'Integrating dgps-usbl position measurments with inertial navigation in the hugin 3000 auv'.

Maybeck, P. S. (1979), *Stachastic Models, Estimation and Control, Volume 1*, Academic Press.

Minkler, G. & Minkler, J. (1993), *Theory and Application of Kalman Filtering*, Magellan Book Company.

Nocedal, J. & Wright, S. J. (1999), *Numerical optimization*, Springer-Verlag New York Inc.

Olsond (2006), Fluxgate magnetometer fgm-301, Technical report, Watson Industries, Inc www.watson-gyro.com.

Parkinson, B. W. & Spilker, J. J. (1996), 'Global positioning system : theory and applications', *American Institute of Aeronautics and Astronautics* **II**.

Sciavicco, L. & Sicilano, B. (2000), *Modelling and Control of Robot Manipulators*, Springer.

Skogestad, S. & Postlethwaite, I. (2005), *Multivariable feedback control*, John Wiley & Sons, Ldt.

Teledyne-Instruments (2005), Workhorse navigator doppler velocity log (dvl), Technical report, Teledyne RD Instruments.

Uhlmann, J. (1994), 'Simultaneous map building and localization for real time applications', *Univ. Oxford, Tech. Rep.* .

van der Merwe, R., Wan, E. A. & Julier, S. I. (2004), 'Sigma-point kalman filters for nonlinear estimation and sensor-fusion - applications to integrated navigation -', *Proceedings of the AIAA Guidance Navigation & Control Conference* .

Vik, B. (2000), Nonlinear design and analysis of integrated GPS and inertial navigation systems, PhD thesis, The Norwegian University of Science and Technology (NTNU).

Vik, B. & Fossen, T. I. (2001), 'A nonlinear observer for gps and ins integration', *Proceedings of the IEEE Conference on Decision and Control* **3**, 2956–2961.

Walpole, Myers, Myers & Ye (2002), *Probability & Statistics for Engineers & Scientists*, Prentice-Hall, Inc.

Wan, E. A. & v. d. Merwe, R. (2000), 'The unscented kalman filter for nonlinear estimation', *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000* pp. 153–158.

Xsens-Technologies (n.d.), Miniature attitude and heading reference system, Technical report, www.xsens.com.

Part IV

Appendix

Appendix A

List of Abbreviations

Abbreviation	Explaination
INS	Inertial Navigation System
IMU	Inertial Measurement Unit
GPS	Global Positioning System
DGPS	Differntial Global Positioning System
HiPAP	High Precision Acoustic Positioning
KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
ECI	Earth-centered inertial frame
ECEF	Earth-centered Earth-fixed frame
NED	North-East-Down frame
ROV	Remotely Operated Vehicle
AUV	Autonomous Underwater Vehicle
DVL	Doppler Velocity Log
HPR	Hydro-acoustic Position Reference System
LBL	Long Base Line
GES	Globally Exponential Stable
RovNav	Poseidon's visualization tool for sub-sea operations

Appendix B

Results

B.1 Results

For simulation 1 and 2 the standard deviation, mean and max error was calculated. The values have been calculated for the error in position, velocity and attitude. The Max error value presented is the max avarage error over a time section which is one hundredth of the total simulation time. [1]

Simulation 1

	Velocity error (m/s)			Position error (m)			Attitude error (degrees)		
Standard deviation	0.0042	0.0043	0.0043	0.0292	0.0963	0.0044	0.0126	0.0158	0.0935
Mean error	0.0000	0.0001	0.0000	-0.0017	-0.0978	0.0002	-0.0001	0.0025	-0.0308
Max error	0.0041	0.0047	0.0044	0.0940	0.3475	0.0092	0.0778	0.0798	0.3713

Table B.1: Simulation 1 with EKF.

	Velocity error (m/s)			Position error (m)			Attitude error (degrees)		
Standard deviation	0.0043	0.0044	0.0043	0.0250	0.0104	0.0113	0.0138	0.0098	0.0979
Mean error	0.0000	-0.0001	-0.0001	0.0102	-0.0137	0.0057	0.0010	0.001	0.0348
Max error	0.0044	0.0042	0.0043	0.0554	0.0340	0.0338	0.0802	0.0495	0.6614

Table B.2: Simulation 1 with UKF.

	Velocity error (m/s)			Position error (m)			Attitude error (degrees)		
Standard deviation	0.0722	0.0497	0.0109	0.0193	0.0135	0.0213	0.0799	0.0766	0.3143
Mean error	-0.0009	0.0005	-0.0000	0.0069	-0.0241	0.0204	0.0024	0.0065	0.0152
Max error	0.0689	0.0570	0.0121	0.0450	0.0494	0.0694	0.1353	0.1250	0.4948

Table B.3: Simulation 1 with Nonlinear observer.

	Velocity error (m/s)			Position error (m)			Attitude error (degrees)		
Standard deviation	0.0437	0.0437	0.0428	0.4169	0.6963	0.0491	0.1443	0.2503	1.3571
Mean error	-0.0020	0.0008	-0.0003	-0.4024	-0.7096	-0.0029	0.0517	-0.0420	0.0684
Max error	0.0424	0.0469	0.0448	1.2399	1.9339	0.1059	0.5652	1.1165	7.2013

Table B.4: Simulation 2 with EKF.

	Velocity error (m/s)			Position error (m)			Attitude error (degrees)		
Standard deviation	0.0433	0.0438	0.0427	0.0824	0.3369	0.1815	0.1247	0.1720	1.2471
Mean error	-0.0000	0.0019	-0.0008	-0.0491	0.9110	0.0442	-0.0152	0.0107	-0.4706
Max error	0.0426	0.0466	0.0420	0.2226	1.3175	0.4483	0.7411	0.9766	7.9551

Table B.5: Simulation 2 with UKF.

Simulation 2

Sensor latency

These results are from simulations where sensor latency was modelled. The position measurment was delayed $5s$ and the velocity $0.5s$. The noise characteristics used was the same as for simulation 1.

B.2 Simulation figures

Here the simulation figures that were not presented in chapter 8.3 will be found.

	Velocity error (m/s)			Position error (m)			Attitude error (degrees)		
Standard deviation	0.7999	0.5112	0.1205	0.1522	0.2064	0.2817	0.7792	0.7317	3.0984
Mean error	0.0072	0.0131	-0.0012	-0.0359	0.1723	0.1898	0.0026	-0.0636	0.0185
Max error	0.8104	0.5114	0.1195	0.4382	0.5639	0.7990	1.4051	1.3032	6.9500

Table B.6: Simulation 2 with Nonlinear observer.

[1] The max hight of the bar plots presented in the simulation results section

	Velocity error (m/s)			Position error (m)			Attitude error (degrees)		
Standard deviation	0.0126	0.0113	0.0090	0.1425	0.0797	0.0067	0.0188	0.0179	0.0912
Mean error	0.0003	0.0000	-0.0001	-0.1018	-0.0569	-0.0002	-0.0019	-0.0102	0.0127
Max error	0.0292	0.0257	0.0164	0.2878	0.2975	0.0205	0.0832	0.0955	0.6120

Table B.7: Simulation with sensor latency in EKF.

	Velocity error (m/s)			Position error (m)			Attitude error (degrees)		
Standard deviation	0.0110	0.0108	0.0090	0.1227	0.0686	0.0189	0.0226	0.0194	0.1379
Mean error	-0.0001	0.0001	0.0000	0.2086	0.1166	-0.0039	0.0048	-0.0037	-0.0264
Max error	0.0242	0.0221	0.0147	0.1116	0.2748	0.0455	0.1439	0.1299	0.7537

Table B.8: Simulation with sensor latency in UKF.

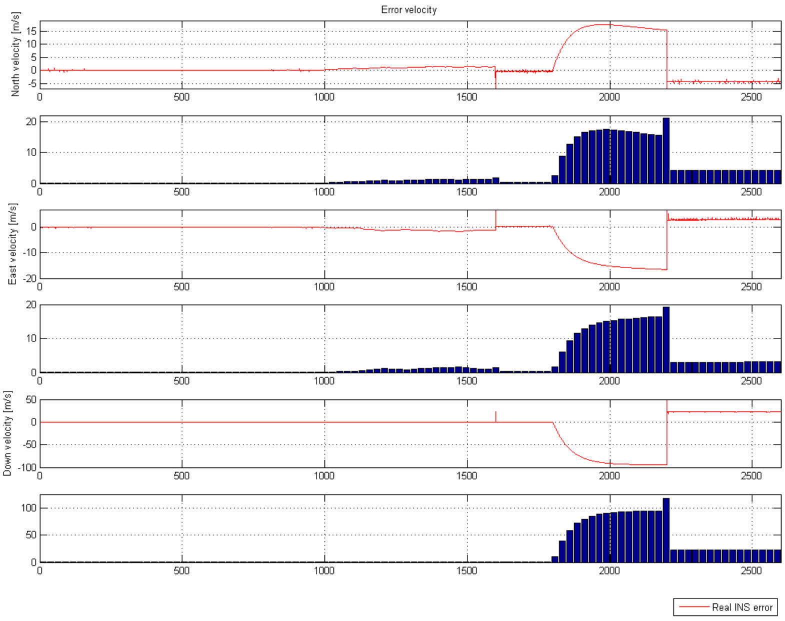

Figure B.1: Velocity error with nonlinear observer for simulation 3

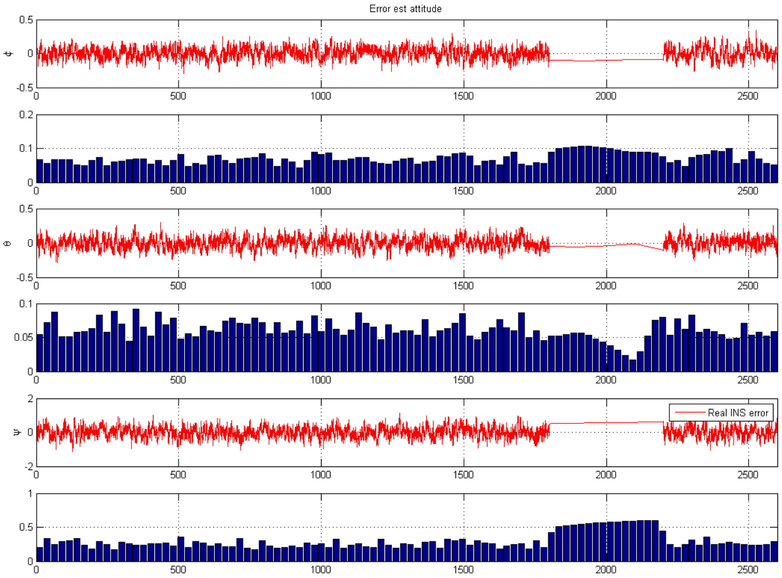

Figure B.2: Attitude error with nonlinear observer for simulation 3

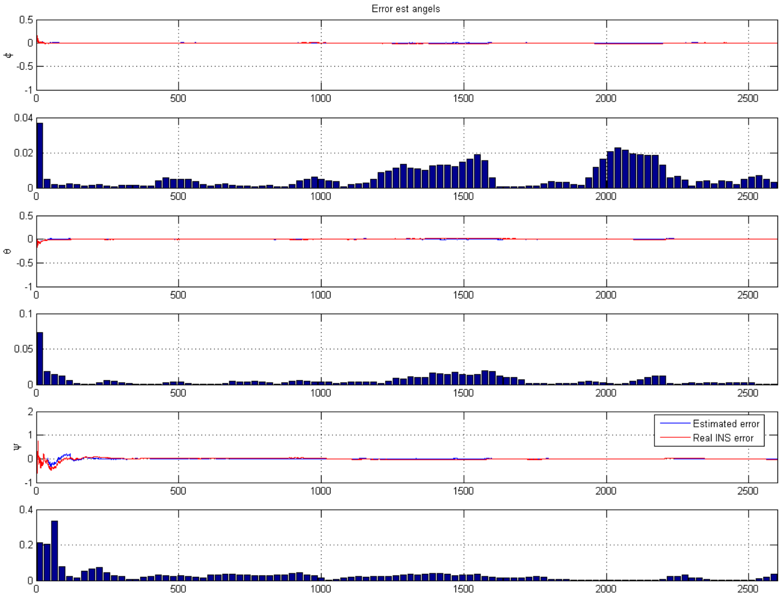

Figure B.3: Attitude error with UKF for simulation 3

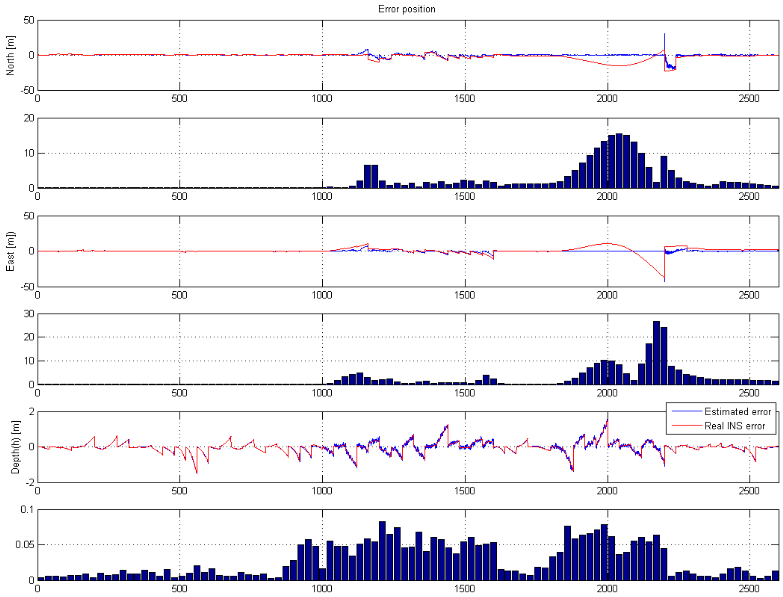

Figure B.4: Position error with UKF for simulation 3

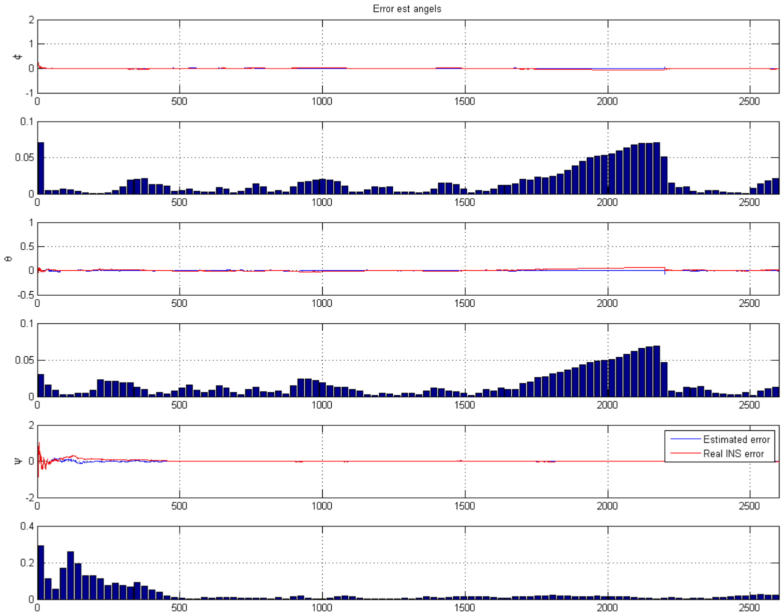

Figure B.5: Attitude error with EKF for simulation 3

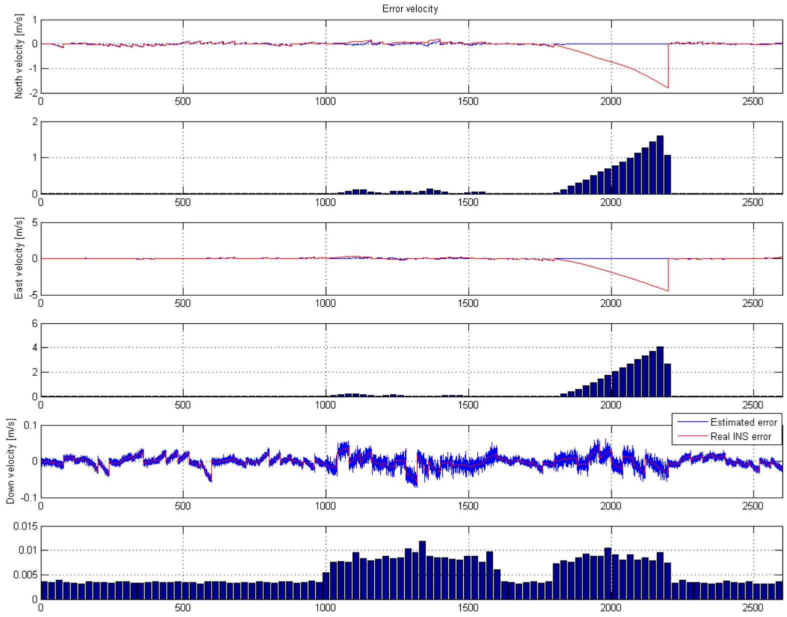

Figure B.6: Velocity error with EKF for simulation 3

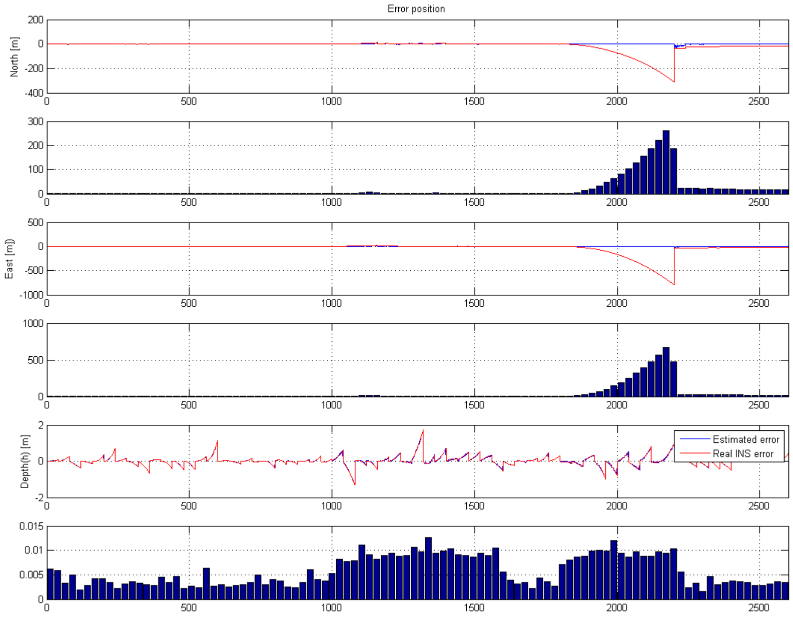

Figure B.7: Position error with EKF for simulation 3

Appendix C

Instrument data

In the simulation done in this work some instrument data is assumed. This information is gathered from the data sheets from the selected instrument. This appendix will summarize the instrument specifications.

C.1 Inertial measurement unit

The LN-200 fiber optic inertial measurement unit from Northrop Grumman Corporation was used in the simulations. This unit consists of 3 solid-state fiber optic gyros and 3 solid-state silicon accelerometers. In this work the IMU is assumed strapped down in the body frame. This meaning that the axes of the IMU is concurrent with the axes in the body frame. The statistical properties given in the data sheet is represented in table C.1.Where applicable, 1μ applies.

Performance specification for the LN-200			
Gyro		Accelerometer	
Bias Repeatability	$1^\circ/hr$ to$10^\circ/hr$	Bias Repeatability	$300\mu g - 3.0mg$
Scale factor Accuracy	100-500 ppm	Scale factor Accuracy	300-5000 ppm
Random walk	$0.07 - 0.15^\circ/\sqrt{h}$, PSD level	White noise	$50\mu\sqrt{Hz}$PSD
Bias variation	$0.35^\circ/h$, 100s correlation time	Bias variation	$50\mu g$, 60s correlation time

Table C.1: LN-200 IMU characteristic

Performance specification for the WHN 300	
Bottom Velocity	
STD at $1m/s$	$\pm0.3cm/s$
STD at $3m/s$	$\pm0.6cm/s$
Ping rate	$7Hz$max
Max tilt	$\pm15^\circ$
Tilt measurement accuracy	$\pm0.5^\circ$
Compass accuracy	$\pm2^\circ$@60° dip

Table C.2: WHN-300 DVL characteristic

C.2 Doppler velocity log

Teledyne's WHN-300 Doppler velocity log (DVL) is selected for the simulations. This section is based on both the data sheet and communication with Teledyne customer services. The DVL uses a 4-beam Janus array convex transducer with 30° beam angle. As well as bottom velocity the instrument can also give a tilt measurement as well as a compass reading. The water reference velocity that is also available in this instrument is not used in the setup in this work. Table C.2 shows the most important quantitative specifications on the instrument.

It has in this work been made some simplifications regarding the DVL. The instruments data rate will be reduced when the vessel is at a greater altitude above the seabed. The second characteristic that is not regarded in this work, is that the standard deviation increase when operating close to the bottom.

C.3 Hydro acoustic positioning

Kongsberg Maritime has a wide selection of High Precision Acoustic Positioning systems (HiPAP). The simulations in this work is based on the data for the HiPAP 500. For the simulations it is assumed that the system is based on one transducer mounted on a vessel. It is also possible to introduce redundancy and increase the accuracy by using a so called Long Base Line positioning principle.

The simulation data used is summarized in table C.3.

Operating range	$1 - 4000$ m
Angle Accuracy	$0dBS/N$: 0.30° $10dBS/N$: 0.18° $20dBS/N$: 0.12°
Range detection accuracy	$< 20cm$

Table C.3: HiPAP 500 characteristic

Pressure Range	$700m$
Accuracy	0.01% of full scale
Resolution	1×10^{-8}

Table C.4: Digiquartz 8DP700 characteristic

C.4 Depth sensor

To measure the depth beneath the ocean surface, all that is needed is a pressure sensor. The Norwegian Defense Research Establishments (FFI) Hugin AUV uses a Digiquartz pressure sensor from Paroscientific Inc. The instruments for the Hugin is described in (Jalving & Gade 1998). It has therefore been used data for this sensor also in this work. It is important to note that the depth measurement should be compensated for tidal water, or else this will be a great factor of error. One solution is to have a depth sensor on a known depth/point, and use this as reference.

Appendix D

CD-ROM contents

The enclosed CD-ROM contains files for running simulations of all filter solutions. It is organized as follows.

- *ExtendedKalman* contains all files necessary for running simulations of the extended Kalman filter.
- *UnscentedKalman* contains all files necessary for running simulations of the unscented Kalman filter.
- *NonlinObserver* contains all files necessary for running simulations of the nonlinear observer.
- *RealDataTest* contains two folders:
    - *extended* contains files for running the real data test for extended Kalman filter.
    - *unscented* contains files for running the real data test for unscented Kalman filter.