NTNU

Norwegian University of
Science and Technology

Tightly Integrated Doppler Velocity Log Aided Inertial Navigational System

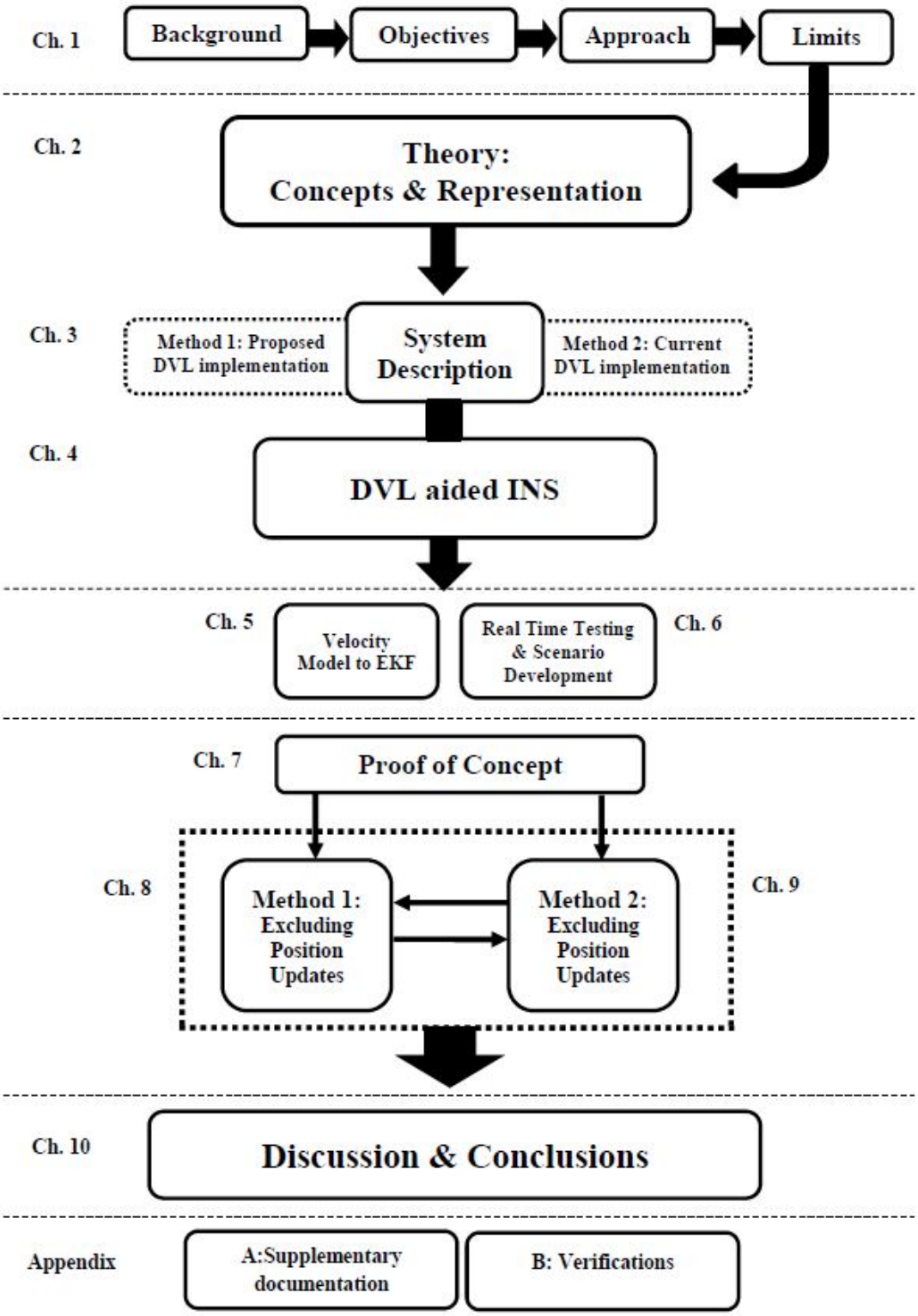
Anders Feyling

Master of Science in Industrial Cybernetics
Submission date: January 2018
Supervisor: Edmund Førland Brekke, ITK
Co-supervisor: Are Baardsgaard Willumsen, Kongsberg Maritime AS

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Thesis Structure

The figure on the succeeding page is meant to provide an illustrative overview of the forthcoming sections. **Chapter 1** begins with a general introduction presenting the area of interest. This introduction is subsequently funneled down into three research objectives. **Chapter 2**, **3**, and **4** presents the foundation for this thesis's estimation and mathematical work, and delineation of key concepts that has been introduced in **Chapter 1**. Moreover, it provides necessary elaborations on DVL and how DVL is utilized in Method 1 and Method 2. Chapter 5 and 6 sets up the footing for the forthcoming proof of concept and analysis by addressing the velocity model to Extended Kalman Filter and moreover elaborating on real time testing and the scenario development. **Chapter 7**, **8**, and **9** is to be viewed as the analysis chapters, whereas **Chapter 10** finally gathers the obtained knowledge and discuss upon the stated objectives of the thesis.



The approach behind this thesis is by definition quantitative and simulation based as its output is built upon the processing of data in quantitative form obtained as real measurements. The

obtained dataset has been provided by Kongsberg Maritime and consists of actual mission output gathered from sensory measurements. The results has been calculated by utilizing NavLab, which is a flexible and generic aided inertial navigation software developed as a research tool by the Navigation Group at FFI[1] and Kongsberg Maritime.

[1] Norwegian Defence Research Establishment

Preface

This thesis is the academic end product of the course *TTK*4920 *Engineering Cybernetics, Master's Thesis*. This course is mandatory for students in Industrial Cybernetics. The course was carried out the fall of 2017, The assignment was presented by control system engineer Are B. Willumsen at Kongsberg Maritime AS. I received consultation, real test data, and a 6 months long license for the post processing tool NavLab. The reader of this thesis is expected to familiar with terminology concerning estimation and statistics.

Trondheim, 10-01-2017

Anders Feyling

Acknowledgement

I would like to acknowledge my supervisors, Are B, Willumsen and Edmund Førland Brekke for their crucial guidance, encouragement and support throughout the process of writing this master thesis. Additionally, I wish to greatly thank the employees of Kongsberg Maritime, who have been very open and forthcoming throughout the entire process. I would also express the deepest and most sincere gratitude to my family for their unconditional support and patience throughout my years of school attendance.

Abstract

Background: Underwater navigation is an important aid for many industries, including oil and gas exploration, marine and subsea operations. These industries are highly dependent on a detailed map of the seabed, which can be supplied by AUVs, in order to conduct underwater operations safely and reliably. One commonly used velocity sensor on AUVs are called Doppler Velocity Log (DVL). Such sensors utilize the Doppler shift in sonic pulses to calculate vehicle velocities for system navigation. The main focus in this thesis has been on examining the quality of a tightly integrated DVL (referred to as Method 1) aided INS and to compare the quality of this method with an cartesian DVL (referred to as Method 2). Testing the performance of the two methods on real sensor data.

Method: Analyses and simulations were carried out by the use of the generic aided inertial navigation software developed by Kongsberg Maritime, NavLab. NavLab is implemented in Matlab, and is used for performing navigation calculations for navigational purposes. Particular attention was given to the velocity measurements from the DVL device, as the velocity measurements bounds the velocity error of the navigation system.

Findings: The proposed implementation of the tightly integrated DVL was proven to be a feasible method, as the Extended Kalman filter (EKF) was able to estimate velocities in transducer beams with an approximate mean errors of 0.02%. However, the EKF was not tuned for Method 1, meaning that the internal Kalman filter dynamics for Method 1 presented in this thesis are not sufficiently accounted for, and lead to that the navigational error did not decrease when using Method 1, relative to Method 2. This leads to a lack of a firm conclusion between the two methods. However, the work presented in this thesis forms a solid foundation for further research within the field of velocity and position estimation for AUVs in Kongsberg Maritime and the important issue of Extended Kalman filter tuning in Method 1 has been illuminated.

Sammendrag

Bakgrunn Navigasjon under vann er et viktig verktøy for mange industrier, inkluderer olje og gass-leting, marin virksomhet og subsea virksomhet. Disse industriene er avhengige av detaljerte kart av havbunnen, levert av AUV-er (Autonom Undervanns Fartøy), for å kunne drive virksomhet under vann på en trygg og pålitelig måte. En av de vanligste sensorene som brukes på AUV-er er såkalte Doppler Hastighets Logger (DVL). Slike sensorer tar i bruk Doppler-effekten i lydbølger for å kalkulerer hastigheten til fartøy, som brukes til systemnavigasjon. Kongsberg Maritime er interessert i å videreutvikle DVL integreringen på HAIN1000. Hovedfokuset i denne oppgaven derfor å undersøke kvaliteten til en tett integrert DVL (referert til som Metode 1) som blir bistått av en INS, samt sammenligne denne metoden med en kartesisk DVL (referert til som Metode 2). Testingen av ytelsen til de to metodene ble gjort med ekte sensor data.

Metode Analysene og simuleringene ble gjort i navigasjonsprogrammet, NavLab, som er utviklet av Kongsberg Maritime og FFI. NavLab er videre implementert i Matlab og brukes for å kjøre navigasjonskalkulasjoner for navigering. Det har vært ekstra fokus på hastighetsmålingene til DVL-enheten. Dette er fordi hastighetsmålingene avgrenser hastighetsfeilene i navigasjonssystemet.

Resultater Den foreslåtte implementeringen av tett integrert DVL har vist seg å være en gjennomførbar metode siden Extened Kalman filter (EKF) estimerte hastigheter til transduser stråler med en omtrentlig middelverdi av feilverdier på 0.2%. EFK-en var derimot ikke innstilt til Metode 1. Dette betyr at den interne dynamikken i Kalman filter-et for Metode 1 ikke er nok tilrettelagt for, og fører derfor til at navigasjonsfeilene ikke minker når man tar i bruk Metode 1 sammenlignet med Metode 2. Dette fører til en mangel på en håndfast sammenligning mellom de to metodene. Arbeidet som er presentert i denne oppgaven skaper derimot et solid fundament for videre forskning på hastighets- og posisjonsestimering for AVU-er i Kongsberg Maritime. Den viktige mangelen med filter-innstilling av Metode 1 har også blitt belyst.

Problem Formulation

The Doppler Velocity Log is crucial for accurate position estimates for underwater vehicles. A DVL integration method that effectively counters error growth is a vital tool to fully utilize the DVL measurements. The assignment will investigate whether a tightly integrated DVL is able to handle errors in a better manner than the currently used cartesian method, by testing on real data.

Abbreviations

Abbreviation	Explanation
ADCP	**A**coustic **D**oppler **C**urrent **P**rofiles
AUV	**A**utonomous **U**nderwater **V**ehicle
CKF	**C**ontinious **K**alman **F**ilter
DGPS	**D**ifferential **G**lobal **P**ositioning **S**ystem
EKF	**E**xtended **K**alman **F**ilter
ESKF	**E**rror **S**tate **K**alman **F**ilter
GNSS	**G**lobal **N**avigation **S**atelite **S**ystem
HiPAP	**H**igh **P**recision **A**coustic **P**ositioning
IMU	**I**nertial **M**easurement **U**nit
INS	**I**ntegrated **N**avigational **S**ystem
ISA	**I**nertial **S**ensor **A**ssembly
KF	**K**alman **F**ilter
LKF	**L**inearized **K**alman **F**ilter
RTS	**R**auch-**T**ung-**S**triebel Algorithm
TOF	**T**ime **O**f **F**light
UKF	**U**nscented **K**alman **F**ilter
USBL	**U**ltra **S**hort **B**ase **L**ine
UTP	**U**nderwater **T**ransponder **P**ositioning

Contents

Chapter 1

Introduction

The field of underwater navigation has in the past century been an important aid for many industries, including oil and gas exploration, marine-, subsea-, and military operations. These industries are highly dependent on a detailed map of the seabed in order to conduct underwater operations safely and reliably. *Global Navigaton Satellite System* (GNSS) signals are unable to reach submerged vehicles, therefore the navigational system cannot simply rely on GNSS signals to estimate position. However, there are methods that allow for accurate navigation underwater. Data gathered for seabed mapping is only relative to the AUV, meaning the data needs to be put into a larger frame of reference. It is therefore important that the position of the AUV is accurately estimated, such that the seabed map is accurate. A Doppler Velocity Log aided Inertial Navigation System (DVL Aided INS) contributes with navigation underwater, but it is important that the errors in the system estimates are low to ensure accurate estimations. Much work has been undertaken in investigating error contributions of DVL Aided INS. Jalving et al. (2004) has proven the effect of lawn-mower patterns to reduce error growth contributed by unobservable scale factors along straight trajectories. Jalving et al. (2004) has also investigated the importance of sound speed accuracy as the velocity scale factor is proportional to the speed of sound at the transducer head.

Hegrenæs and Berglund (2009) has reported the effectiveness, environmental estimation, and robustness of DVL Water Track aided INS, proving the capability of navigation in the mid-water zone where the seabed is out of range for Bottom Track. Accurate estimation of the sea current was also proven, increasing the accuracy of methods that rely on accurate sea current estimation, such as lever arm compensation.

Willumsen and Hegrenaes (2009) has demonstrated the value of data in underwater navigation, yielding improvement in quality and robustness of estimates. The smoothed value is in a practical sense the closest we can get to the true value, which is a valuable parameter when assessing navigational data.

An *Integrated Navigational System* (INS) coupled with a velocity sensor, a pressure sensor for depth measurement, various position sensors[1], and a compass for orientation update, are used to compute position, depth, orientation, and velocity. One commonly used velocity sensor used on Autonomous Underwater Vehicle (AUV) are called *Doppler Velocity Log* (DVL). To further build on the error analysis groundwork already laid down, I will examine the performance of a tightly integrated DVL INS, tested on real measurement data. The presented result verify that a tightly integrated method for DVL integration is feasible, performing very similar to the currently used implementation. The aim of this thesis aim to explore if a tight integration structure of the DVL will perform better than the current implementation method.
The objectives of this thesis are summarized as follows

1. Examine the feasibility of a tightly integrated DVL approach.
2. Compare the performance of the current DVL implementation with the tightly integrated implementation.
3. Evaluate if a tightly integration requires further testing

The velocity measurements bounds the velocity error of the navigation system, which in turn leads to only linear increase in position standard deviation. As opposed to inhibiting bounded acceleration error, leading to linear error in velocity and quadratic increase in positional error. If position was only derived for acceleration, the error in position would be squared, and a lot more difficult to compensate for. This is the reason why it is interesting to further explore the implementation of DVL aided INS. The more accurate the DVL measurements can be incorporated in the Kalman filter, the better the navigation system will perform.

1. Examine the feasibility of a tightly integrated DVL approach.
2. Compare the performance of the current DVL implementation with the tightly integrated implementation.
3. Evaluate if a tightly integration requires further testing

[1] *Differential Global Positioning System* (DGPS), *High Precision Acoustic Positioniung* (HiPAP), *Ultra Short Base Line* (USBL)

Chapter 2

Theory: Concepts and Representation

Some of the basic terminology mentioned in chapter 1 when dealing with estimation, statistics, and filtering will be presented in the following chapter. The navigational tool Navlab and its applicability will be introduced. This is followed by an introductory explanation of the post-processing schemed called smoothing.

2.1 Inertial Navigation System

An *Inertial Navigation System* (INS) according to Vik (2014) consists of an Inertial Measurement Unit (IMU) software that computes position, velocity, and attitude from the measurements. The IMU incorporates an Inertial Sensor Assembly (ISA), hardware to interface with the ISA, and software performing down-sampling, temperature- and vibration calibration. The ISA typically consists of three gyros and three accelerometers positioned in an orthogonal manner. This allows for the acceleration and angular velocity to be measured in three dimensions.

INS systems are categorized into two classes: gimbal and strapdown. The system type used on HUGIN, depicted in figure 2.1, an Autonomous Underwater Vehicle (AUV) produced by Kongsberg Martitume, falls within the strapdown group. A strapdown system is rigedly attached to the AUV. These solutions maintain attitude by calculating rotation matrices. This solution has less drift compared to a strapdown system, because the gimbal platform is exposed to more dynamic benign movements, allowing for sensors with less dynamic range to be used. This also

Figure 2.1: The HUGIN Autonomous Underwater Vehicle onboard H.U. Sverdrup II. Picture credited to Hegrenæs and Berglund (2009)

means that high order instrument errors are less important than when using a strapdown solution Vik (2014). While the strapdown solution requires more computational power than the gimbal system, its not a large problem as the cost of computational power is decreasing rapidly Vik (2014). Relative to the gimbal system, the required computational power is greater in strapdown systems, but this is not a problem since the cost of computation is so low in today's world. The advantages comes at the price: the gimbal system is heavier, takes up more space, is more expensive, and is generally not suited for use on AUVs. The system employed on HUGIN is therefore a strapdown solution.

Raw data readings compiled from sensor measurement requires specific data handling routines in order to filter out the desired values. According to Gade (1997), the readings area function of true value, bias, and scale-factor error as well as other external factors, such as hysteresis and temperature. Depending on the precision requirement, the measurement function can be estimated with a straight line or a more complex polynomial, as described in figure 2.2. If a straight line is used, the parameter describing said line must be found. If a linear function is used, the parameter describing this line must be found. Assume the scalar variable a is to be measured. This can be parameterized as described in equation 2.1.

$$\tilde{a} = a + \delta a \tag{2.1}$$

where $\tilde{a}$ is the measured variable and δa represents the error differentiating the measured variable to the true variable. The error, δa is expressed as

$$\delta a = \underbrace{\Delta a_{scalefactor} x}_{\text{scale factor error}} + \underbrace{\Delta a_{bias} + \xi_a}_{bias} \tag{2.2}$$

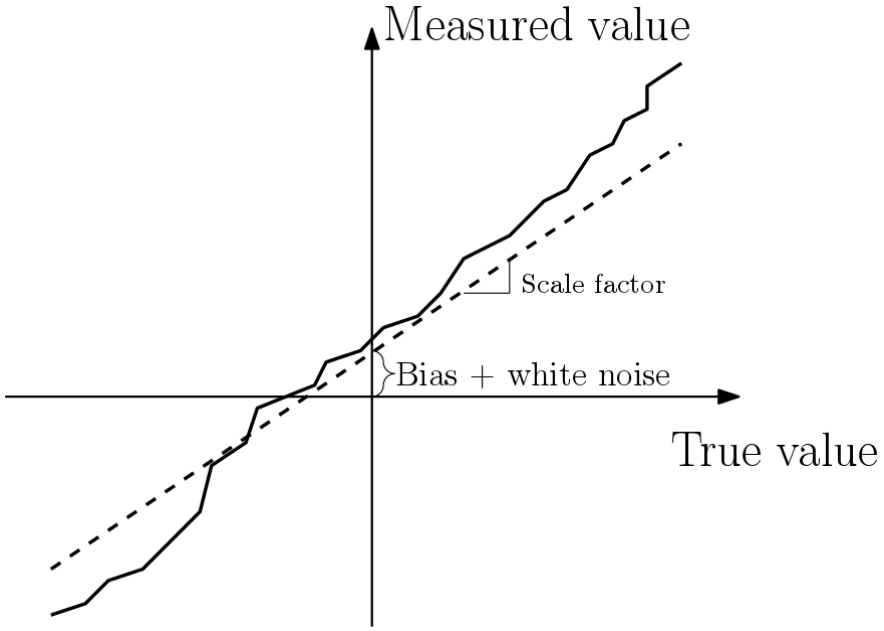


Figure 2.2: The relationship between the true and the measured value. The dashed line is the line approximation while the continuous line is the real relationship. The figure is adapted from Gade (1997).

Gade (1997) states that inertial sensors have a scale factor error that approximately so constant with respect to time. The bias is comprised of a slow varying bias plus white noise.

2.2 General Estimation

Generally, measurements contain a significant amount of noise, introducing uncertainties to the state measurements. Supplementing the measurement with a mathematical model of the dynamics, it is possible to estimate the state given the model is sufficiently accurate. A model that provides an estimate of an internal state is called a *state observer*. Both physical processes and mathematical models in this thesis will be presented on *state space* form; as a set of first order differential equations (see Chen (2013) for more information regarding state space formulations). The nomenclature of a state space equation can be seen in the equation set 2.3.

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u} + \gamma \tag{2.3a}$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \xi \tag{2.3b}$$

Equation 2.3a is the state equation which describes the dynamics of the states. **F** is the state matrix, representing the relationship of the states (x). **b** is the input matrix, representing the input (u) relative to the states and γ is the process noise. Equation 2.3b is the output equation

which describes the measurements. **H** represents the measurement matrix and describes the relationship between state and measurement and ξ is the measurement noise.

The system is linear if none of the terms in equations 2.3 contains non-linear terms. This property allows for many methods that rely on linearity to be utilized, such as the Kalman filter McGee Leonard and Schmidt Stanley (1985).

A representation of a state observer is seen in figure 2.3. The purpose is to eliminate the difference between $\mathbf{y}_k$ and $\hat{\mathbf{y}}_k$, which causes the values $\mathbf{x}_k$ and $\hat{\mathbf{x}}_k$ to converge.

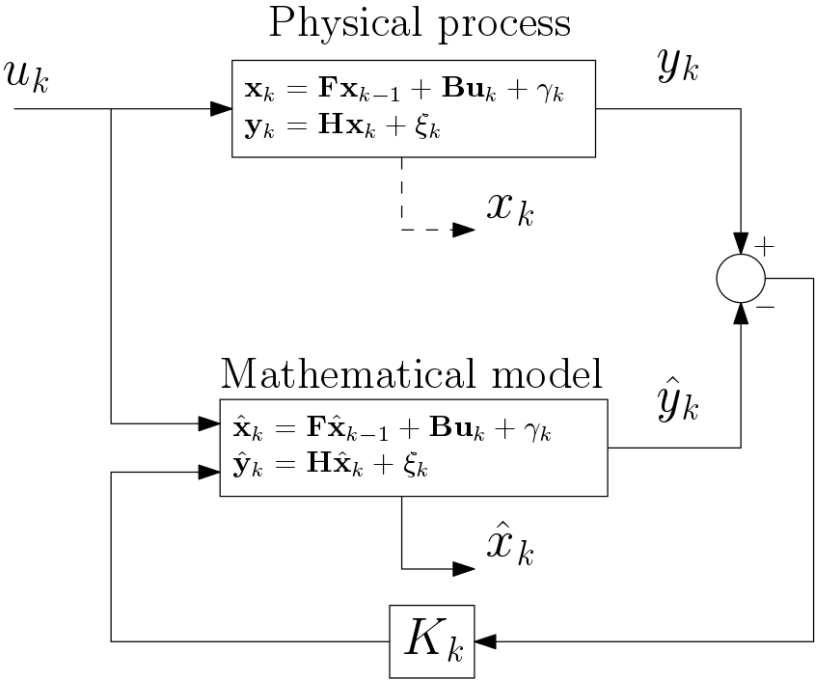


Figure 2.3: Convergence of y and $\hat{y}$ causes convergence of x and $\hat{x}$. The line from the Physical process is dashed, as this variable technically does not exist.

The system representation in equation 2.3 is required to be *observable* in order to use a KF as a state estimator. Observability according to Chen (2013) studies the possibility of estimating the states from the output. Another description cited in (Brown and Hwang, 2012) is that a system is said to be *observable* if for any unknown initial state $\mathbf{x}(0)$, there exists a finite $t_1 > 0$ such that the knowledge of the input $\mathbf{u}$ and the output $\mathbf{y}$ over $[0, t_1]$ suffices to determine uniquely the initial state $\mathbf{x}(0)$. Otherwise, the equation is said to be *unobservable* . In other words, if the system is observable, all system states can be either directly or indirectly measured. If the system in equation 2.3 is to be computed, it must first be discretized in time: the system changes values at predefined time steps. Discretization is necessary because computers only operate in discrete time steps. The time step throughout this thesis will be denoted with the subscript [k].

2.2.1 Estimator Bias

The bias of an estimator is the difference between the true value of the estimated parameter and the expected value of the estimator. An estimator is unbiased if:

$$E(\text{Estimate of X}) = E(X)$$

where the expected value of the estimate of a random variable X, is exactly equal to the expected value of the true value of X. In ideal cases, the estimator is unbiased. However realistic estimators will inhibit some bias. When an estimator is known to be biased, it is often possible to estimate and correct the bias through the use of a KF.

2.2.2 The Kalman Filter

The Kalman filter (KF) is an algorithm that computes parameters of interest from indirect and uncertain observations. The KFs inventor was the Hungarian-born American Rudolf Emil KálmánKalman (1960). The work was based on Norbert Wieners work on minimizing the mean-square error (Brown and Hwang, 2012). Kalman considered the noisy measurement to be a discrete sequence in time as opposed to a continuous-time signal. The term filter describes the process where the algorithm separates noise from useful information in signals

Introduction to Kalman Filtering

The Kalman filter (KF) is an *optimal recursive data processing algorithm*Brown and Hwang (2012). The algorithm uses information from the previous step to aid in obtaining information in the current step. A KF is used when it is necessary to estimate a system state when the state can only be indirectly measured It can also be used to combine multiple measurements from different sensors which are subjected to noise in order to improve accuracy. A Kalman filter has different functionality based on the objective. Functionalities include: estimate a system state when it cannot measure it directly or estimate a system state by combining measurements from different sensor subjected to noise. Common appliances for Kalman filtering are navigation, guidance, object tracking, signal-processing, computer vision, and control-theory. One of the first appliances of KF was on the Apollo project, where is was used for course estimation (McGee Leonard and Schmidt Stanley, 1985). There are some requirements for a KF to function correctly, concretized in the following list.

1. That both the process- and measurement noise is white and Gaussian.
2. The system is linear.
3. The initial state is Gaussian.

4. The system is observable.

2.2.3 Discrete Kalman filter

The implementation of discrete Kalman filtering is described in the following section. In order to implement a KF on a computer, which has discrete measurement in time, a discrete Kalman filter (DKF) can be implemented. The DKF computes discrete estimates in a series of iterative time steps. The system is concretized in the following mathematical discrete state-space model:

$$\mathbf{x}_{k+1} = \phi_k \mathbf{x}_k + \mathbf{u}_k + \Omega_k \gamma_k \tag{2.4a}$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \xi_k \tag{2.4b}$$

Where $\mathbf{x}_k$ is the process state vector at time t_k. Φ_k is a square matrix relating $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$. Δ_k is the input matrix. $\mathbf{u}_k$ is the input vector. γ_k is the process noise vector. $\mathbf{y}_k$ is the measurement at time t_k. H_x is the matrix containing the ideal connection between measurement and state vector at time t_k, and ξ_k is the measurement noise.

The covariance matrices Q_d and R_d relating the process noise w and measurement noise v are described in

$$E[\gamma_k \gamma_j^T] = \begin{cases} \mathbf{Q}_{d,k} & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases} \tag{2.5a}$$

$$E[\xi_k \xi_j^T] = \begin{cases} \mathbf{R}_{d,k} & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases} \tag{2.5b}$$

$$E[\gamma_k \xi_j^T] = \mathbf{0} \quad \text{for all } k \text{ and } j \tag{2.5c}$$

As equation set 2.5 shows, the process and measurement noise are both white and uncorrelated. A recursive algorithm (DKF) which combines the measurements and the system dynamics such that the *optimal state estimate* can be calculated. Optimal state estimate is defined in Gade (1997) as the state estimate which is closest to the expected value and has the lowest possible variance.

Assuming there exists an initial estimate $\bar{\mathbf{x}}_k$ of the process at time t_k, and that this estimate is based upon all prior knowledge of the process leading up to t_k, $\bar{\mathbf{x}}_k$ is denoted at the *a priori* state estimate. This is associated with the error covariance matrix $\bar{\mathbf{P}}_k$. The *a priori* estimates are manipulated such that the optimal estimates, the *a posteriori* estimates $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{P}}_k$ are produced. At this point, the measurement $\mathbf{y}_k$ is used to improve the a priori estimate. A function of the noisy measurement and the a priori estimate, together with the blending factor $\mathbf{K}_k$, yields the a posteriori estimate. The factor $\mathbf{K}_k$ is called the Kalman gain. The Kalman gain minimizes the mean-square estimation error.

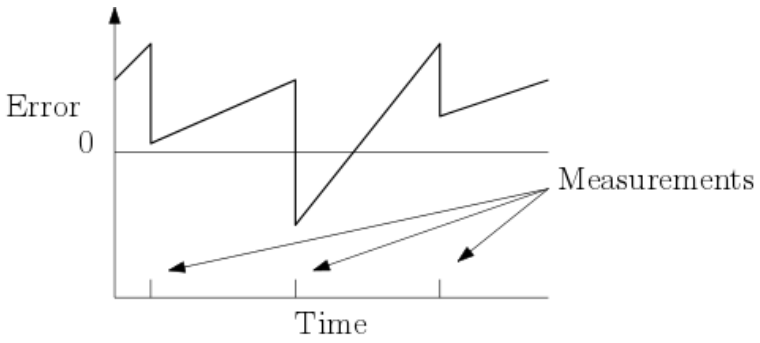


Figure 2.4: Standard deviation will always decrease at updates but actual error might increase on certain measurements, depending on if the measurements is wrong. The figure is an illustrative intepretation made by the author.

The gain is a relative weight matrix that determines how much weight will be placed on the measurements versus the current model estimations. The gain will vary depending on how recently the system received a measurement. Immediately after a measurement is received, the filter weighs the model estimate less, as the measured value is available, see figure 2.4. A special case is if the process noise is negligible. The result will then be a model estimate that does not change over time.

2.2.4 The Discrete Kalman Filter Algorithm

The DKF algorithm is explained in detail in the following steps. The procedure is based on the approach for the DKF algorithm in Brown and Hwang (2012). **Step 0 : Set initial parameters at time $k = 0$.** The initial values of the states $\bar{x}_0$ are gathered, and the initial error covariance matrix is computed by the equation

$$\bar{\mathbf{P}}_0 = E[(\mathbf{x}(0) - \hat{\mathbf{x}}_0)(\mathbf{x}(0) - \hat{\mathbf{x}}_0)^T] \tag{2.6}$$

It is seldom the case that state measurements are available at initialization in real-time applications. A solution is to either estimate these offline, or to set them initially high to account for errors. This initial step is only done once during run time.

Step 1 : Compute Kalman gain. The Kalman gain in obtained through

$$\mathbf{K}_k = \bar{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \bar{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_{d,k})^{-1} \tag{2.7}$$

Step 2 : Update state estimate with measurement $\mathbf{y}_k$. Using the available measurement, compute the a posteriori estimate. This value is then outputted by the filter, as the optimal estimate of $\mathbf{x_k}$.

$$\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \bar{\mathbf{x}}_k) \tag{2.8}$$

Step 3 : Compute error covariance matrix for updated estimate.

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\bar{\mathbf{P}}_k \tag{2.9}$$

A more robust version of the error covariance update equation is the *Joseph form* Gade (1997). The Joseph form increases the numerical stability of the covariance matrix, increasing the robustness of the matrix. However positive definitness is not guaranteed. The form was introduced in (Gade, 1997) to help against rounding errors in the computations. The Joseph form is described as follows:

$$\hat{\mathbf{P}}_k = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\bar{\mathbf{P}}_k[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]^T + \mathbf{K}_k\mathbf{R}_{d,k}\mathbf{K}_k^T \tag{2.10}$$

Step 4 : Project ahead Update the a priori estimates for the next iteration of the filter.

$$\bar{\mathbf{x}}_{k+1} = \Phi_k\hat{\mathbf{x}}_k \tag{2.11a}$$

$$\bar{\mathbf{P}}_{k+1} = \Phi_k\bar{\mathbf{P}}_k\Phi_k^T + \Omega_k\mathbf{Q}_{d,k}\Omega_k^T \tag{2.11b}$$

The algorithm for the Discrete Kalman filter is seen in figure 2.5.

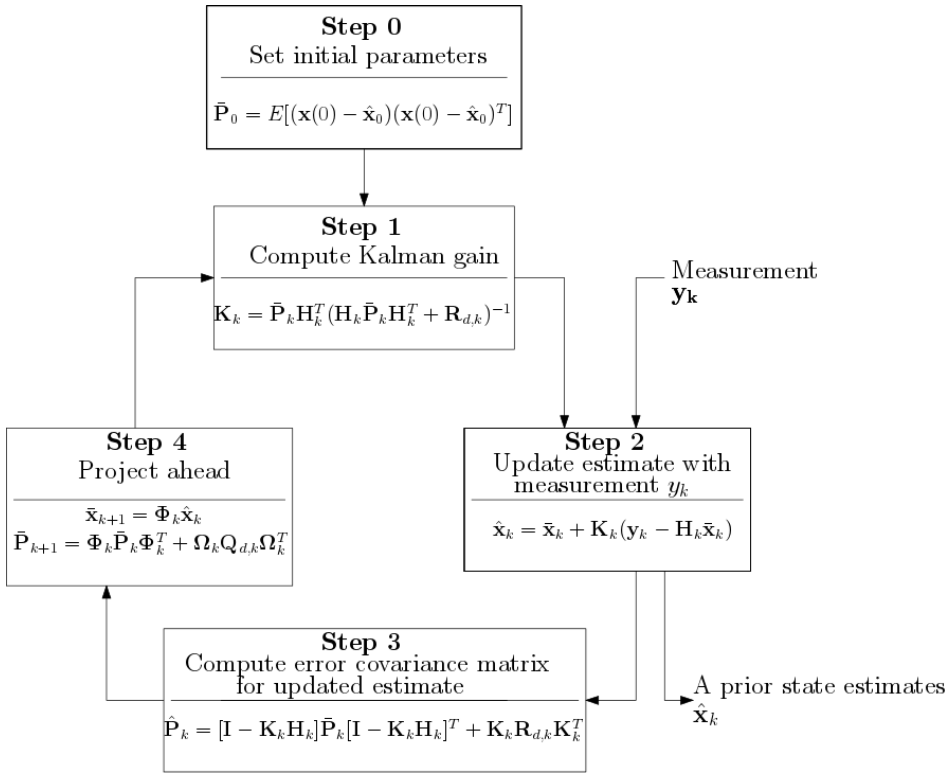


Figure 2.5: The iterative process of a Discrete Kalman filter. Adapted from Brown and Hwang (2012)

2.3 Nonlinear Filtering

Process dynamics are rarely linear, which means that the regular Kalman filter cannot be used as a state observer. As noted in section 2.2.2, one of the requirements for a Kalman filter to be implementable is that the system is linear. One solution may be by *linearizing* the system - linearization is the process of approximating a nonlinear equation by use of linear equations under certain conditions[1] Chen (2013). The two most prominent filters to handle nonlinear situations are the *Linearized Kalman filter* and the *Extended Kalman filter*. Brown and Hwang (2012) describes the LKF and EKF as the following: the LKF linearizes about some nominal trajectory in state space that does not depend on the measurement, whilst the EKF linearize about a trajectory which is continuously updated with the state estimates coming from the measurements.

2.3.1 Extended Kalman Filter

Extended Kalman filter can be described by assuming a nonlinear system that is defined by equation 2.12.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \gamma(t) \tag{2.12a}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), t) + \xi(t) \tag{2.12b}$$

The process noise and measurement noise are, in accordance with the assumptions in section 2.2.2, assumed Gaussian distributed, i.e. zero mean, with $\mathbf{Q}$ or $\mathbf{R}$ variance.

$$\gamma(t) \sim \mathcal{N}(0, \mathbf{Q}(t))$$

$$\xi(t) \sim \mathcal{N}(0, \mathbf{R}(t))$$

The linearization is of the form

$$\Delta\mathbf{x}^* = \mathbf{F}(\mathbf{x}^*, \mathbf{u}(t), t)\Delta\mathbf{x}(t) + \gamma(t) \tag{2.14a}$$

$$\Delta\mathbf{y}^* = \mathbf{H}(\mathbf{x}^*, t)\Delta\mathbf{x}(t) + \xi(t) \tag{2.14b}$$

where $\mathbf{F}$ and $\mathbf{H}$ are Jacobian matrices and $\mathbf{x}^*$ is an approximate trajectory.

[1] Continuous and analytic

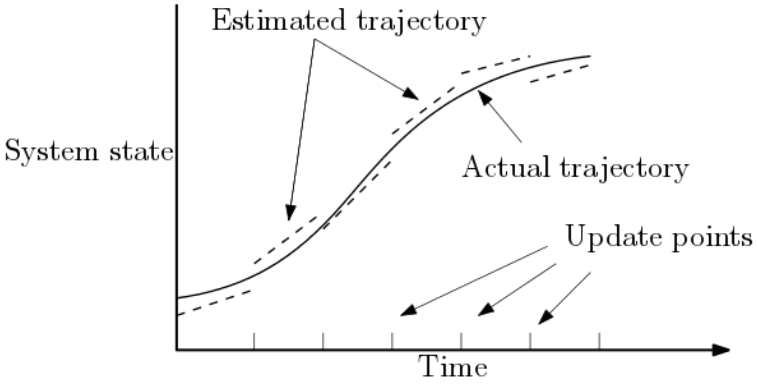


Figure 2.6: The trajectory is updated at each time step. The figure credited to Brown and Hwang (2012).

Figure 2.6 shows how a updated trajectory is computed based on the current system states at each time step.

2.4 Coordinate Frames

Figure 2.7 depict the four coordinate systems mentioned in this theses.

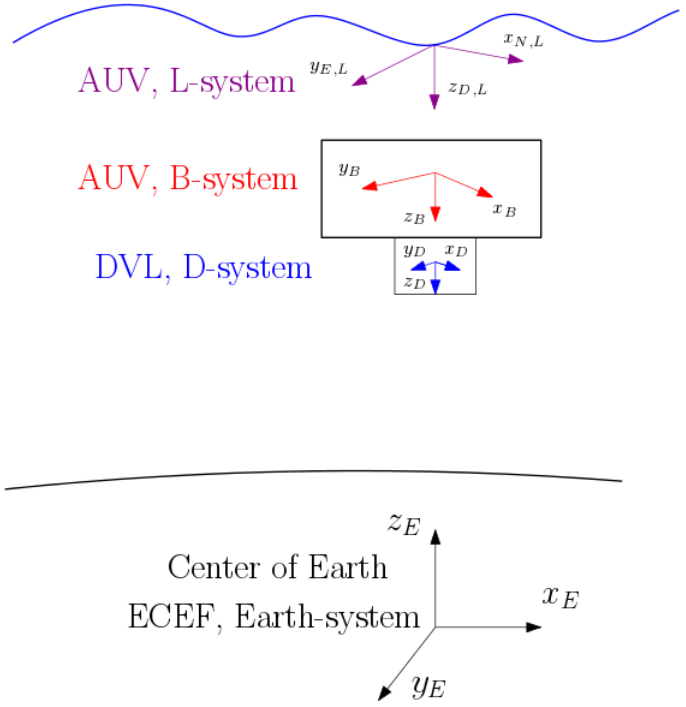


Figure 2.7: The x and y components of the L-system are perpendicular, as they represent North and East direction. The blue line at the top represents the water surface, while the black line at the bottom represents the bottom of the sea. The figure is an illustrative interpretation made by the author.

Table 2.1 contains basic information regarding the different coordinate frames shown in figure 2.7.

Name	Corresponds to	Definition
E	ECEF (Earth-Centered-Earth-Fixed)	Origin is placed in the geometric center of the Earth. The yz-plane coincides with the equatorial plane. z-axis points towards True north.
L (Local)	NED (North-East-Down)	z-axis directed downward, normal on the model of the Earth used. x- and y-axis directed towards north and each, respectfully. Origin is placed such that the vessels reference point have zero x- and y-value, and such that the surface of the Earth model have zero z-coordinate.
B	ABC(Body-fixed)	Coordinate system with origin in the vessels reference point. x-axis directed forward, y-axis directed towards starboard (to the right), and the x-axis is directed downwards.
D	ABC(Body-fixed)	Similar to B, but with origin in the DVLs reference point.

Table 2.1: Overview of the coordinate system used. The figure and explanations are credited to Gade (1997)

2.5 Filter Consistency

It is important to control if the estimates generated by the filter are in agreement with the measurements. Incorrect estimates can be generated due to a number of reasons, including sudden and significant changes in the measurements or for instance rapid changes in the vehicle attitude. Whatever the reason, it is crucial to have a check in place to validate the estimates. The basis for such a check can be to monitor that the expected measurements estimates follow the trend of the actual measurements.

2.5.1 Filter Consistency Check for Method 1

An adaptation of the NIS method is to list a histogram of the normalized squared innovation, and fit a Gaussian distribution to this histogram. This distribution will yield a standard deviation $\hat{\sigma}$ and mean $\hat{\mu}$ based on the normalized innovation histogram. The ideal values for the estimated standard deviation and mean value for an ideally functioning Kalman filter are

$$\begin{aligned} \hat{\sigma} &= 1 \\ \hat{\mu} &= 0 \end{aligned} \tag{2.15}$$

Figure 2.8 shows a sample histogram with a fitted estimated Gaussian (red line), and a Gaussian with the parameters in equation 2.15.

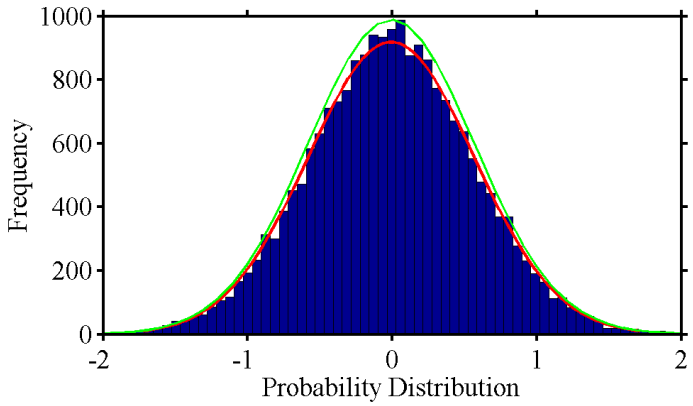


Figure 2.8: Example histogram of innovation squared normal distribution

The estimated Gaussian curve is found using the MATLAB function *histfit* Attaway (2013). The number of bins are equal to the square root the amount of innovation observations from the Kalman filter. The data inserted into the histogram function is categorized as a row vector containing the product of the square root of the NIS multiplied with the sign of the corresponding innovation. In other words, this can be expressed by the use of Matlab syntax, as

$$histfit(\sqrt{\text{NIS}} \odot sign(\mathscr{I}))$$

The *NIS* (see appendix A.6 for information regarding this parameter) vector containing the normalized innovation squared and $\mathscr{I}$ is a vector containing the innovations, i.e. difference between predicted and measured parameter. $\odot$ is element wise multiplication.

2.5.2 Filter Consistency Check for Method 2

The method described above only functions for scalar quantities. Therefore, the test must be modified to encompass innovations of dimension 3×1. This can be done by dividing element

j in the innovation vector $\mathscr{I}$ by the square root of the diagonal element $[j,j]$ of the covariance matrix.

$$histfit(\frac{\mathscr{I}(j)}{\sqrt{\mathscr{C}(j,j)}}) \tag{2.16}$$

where $\mathscr{C}$ is the innovation covariance matrix.

$$\mathscr{C} = H \cdot \hat{P} \cdot (H)^T + W \tag{2.17}$$

$\hat{P}$ is from equation 2.10, H is the measurement matrix, and W is the process noise. This method does not account for elements of the diagonal of $\mathscr{C}$.

2.6 Navigation Laboritory (NavLab)

NavLab serves as the platform upon which the experiments will take place. One of Navlab's core strengths is that the software can perform simulations, enabling testing and development of navigation data Gade (2005). Navlab also is able to post-process recorded data, such that the performance can be measured from real-life data. NavLab enable the user to choose to deactivate sensors (disabling measurements) before run-time, so that scenarios with faulty sensors can be explored. This functionality will be taken advantage of when comparing the two DVL integration methods in chapters 8 and 9.

2.7 Smoothing of Estimates

The method of smoothing estimates is defined by Willumsen and Hegrenaes (2009) as the mathematical equations that provide optimal estimates based on the entire measurement set. Estimates calculated in real-time (see 2.11a) are computed based on the measurements up until the current time step. Smoothed estimates are computed based on measurements from the past, and measurement from the future. It is therefore unavailable in real-time, and is only used in post-processing. However Hain Subsea provides semi-real-time smoothed estimates, meaning that smoothed estimates are available at a specified time lag.

The approach for computing smoothed estimates used in Willumsen and Hegrenaes (2009) is the RTS method (Rauch-Tung-Striebel Algorithm). The method can be described by doing a second Kalman filter sweep backwards in time. The last estimated parameter in real-time and the first computed smoothed estimate are equal, as they are computed under the same basis.

Add figure of smoothed estimates and add arrows.

Chapter 3

System Description

The aim of this chapter is to describe the current, and the proposed Doppler Velocity Log implementation to the navigation system.

3.1 Types of Integration Implementations

The various degrees of integration schemes can be organized into four main categories, according to Vik (2014). These includes *uncoupled-*, *loose-*, *tight-*, and *deep integration*. From uncoupled- to deep integration the performance and robustness against noise will increase, at the expense of increased complexity, possible lack of redundancy, and reduced flexibility Vik (2014). The increasingly complex integration architectures are dependent on how much mathematical manipulation the measurements endure before being inputted to the Kalman filter. The four aforementioned architectures are well explained in Vik (2014) but in the context of above surface navigation (where GPS signals are readily available). Therefore, not everything listen in Vik (2014) is applicable to underwater navigation. The INS structure that most closely resembles the goal structure of this thesis is tight integration. A tightly coupled system utilizes raw accelerometer and gyroscopic data as input to the filter. The raw data is not computed into estimates of position, velocity, and attitude to avoid error growth.

3.2 Method 1 - Proposed DVL Implementation

The solution proposed in this thesis is based around using the raw beam velocities as measurement input to the filter. This solution type resembles a tightly integrated system, as illustrated in figure 3.1. The gray blocks are sensors which does not impact the study in any meaningful way.

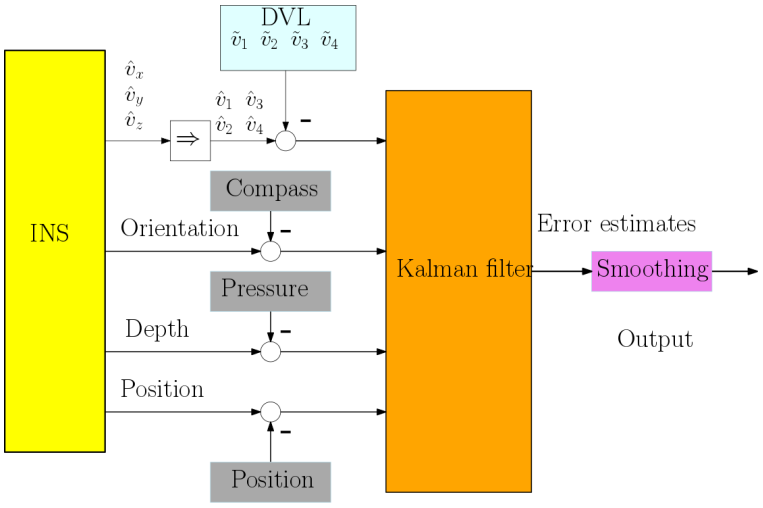


Figure 3.1: The HUGIN integrated INS structure for Method 1. The figure is inspired by a figure in Jalving et al. (2004).

For each prediction step will the Kalman filter run four update steps. The time difference between each measurement in a group i.e four beam measurements per step update is approximately zero, as each DVL transducer produces measurements independently.

3.3 Method 2 - Current DVL Implementation

The current DVL implementations translate the beam velocity into three orthogonal velocities (x,y,z) before inputted to the Kalman filter.

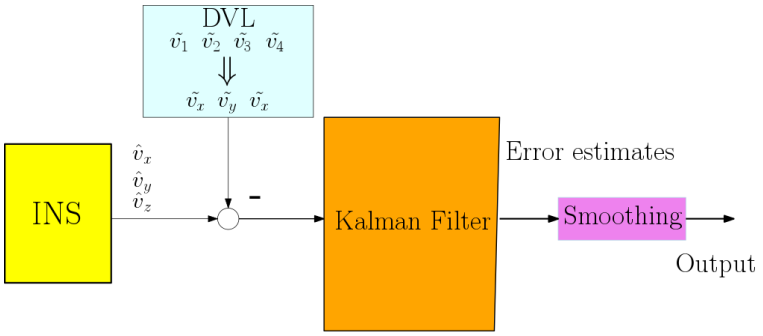


Figure 3.2: The HUGIN integrated INS structure for Method 2. The figure is inspired by a figure in Jalving et al. (2004).

Chapter 4

Doppler Velocity Log aided Inertial Navigational System

This chapter will explain how a Doppler Velocity Logger functions, how the DVL beam configuration is developed, and the reasoning behind the need for DVL calibration.

4.1 Doppler Velocity log

A Doppler velocity log (DVL) is a sensor used to measure the relative velocity of an underwater vehicle. Combined in real time current estimation, pressure readings, DVL water-track and DVL bottom-track are integrated in the nagivatin system to provide velocity aiding for the INS Hegrenæs and Berglund (2009). DVLs can be considered as a subgroup of Acoustic Doppler Current Profiles (ADCP), which measures the relative velocities at various points in the water flow Rudolph and Wilson (2012).

4.1.1 DVL configuration

The measurement principle used in a DVL is the Doppler effect, i.e the shift in frequency when moving towards or away from a relative target. The DVL data referenced in this thesis is collected by a Nortek DVL 500 NORTEK (2018). The four transducer beams are configured in a Janus configuration, such that the two beams in the heading direction are at an angle 45^o degrees away from the x-axis. The other two beams are mirrored, creating a symmetrical orientation. All beams are approximately 25^o degrees away from the z-axis. See figure 4.2 for a more descriptive illustration of the configuration.

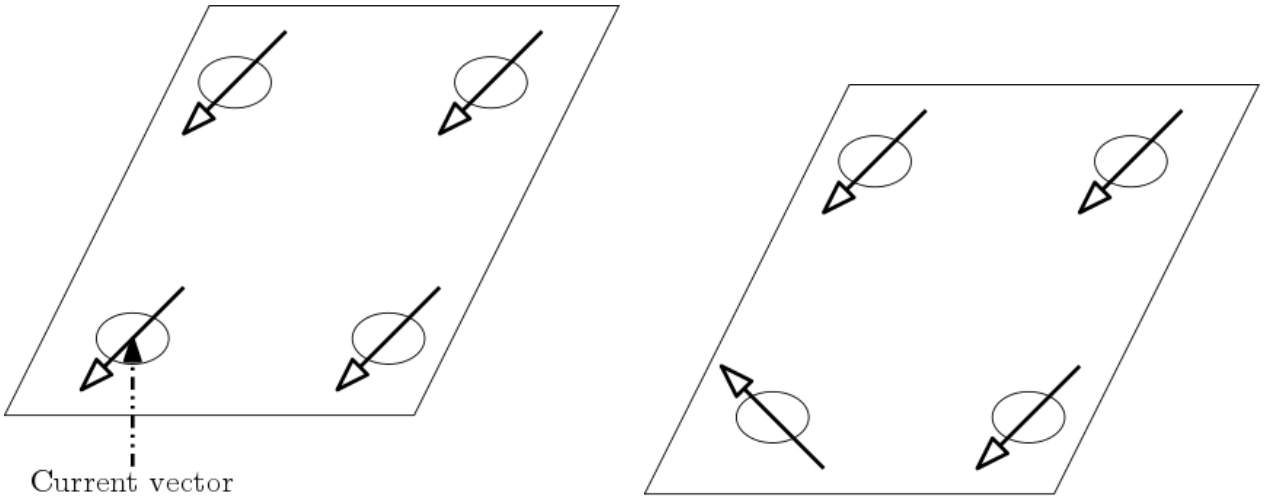
4.2 INS Aiding Sensors

DVL is not the only aiding sensor in a *inertial navigation system*. The Hugin 1000 Kongsberg Maritime (2017) utilizes a pressure sensor for depth measurements and an optional compass for attitude measurements. The compass is optional because some Hugin vessels use *gyrocompassing* instead of a compass. The navigation system must receive occasional updates based on the position accuracy requirements (Jalving et al., 2004). The preferred method of position updating is GPS surface fixing, a method where the AUV surfaces at regular intervals to receive position updates from GPS satellites. Depending on terrain restriction and mission parameters such as stealth and efficiency, the AUV might be confined to remain under water. For submerged position updates, the Hugin 1000 has some options. Underwater transponder positioning (UTP) utilizes transponders at the seabed to calculate position (Hegrenæs et al., 2009). The submersible can also utilize the bathymetric terrain if it is close to the seabed and the area has already been mapped. When operating in acoustic vicinity of a surface vessel e.g mothership, buey, or support ship, DGPS-USBL can be used to find position. A transceiver is mounted under the launch ship as well as a transponder mounted on the AUV. The distance between the vehicles can be found based on the time of flight (TOF). This can further be paired with the phase comparison of the arriving signal which enables for bearing to be deduced (Vickery, 1998).

4.3 DVL Measurement Equations

As the DVL uses multiple transducer beams with different attitude, trigonometry is used to calculate the velocity in all three directions. Readings from one beam is required per direction. This means that a DVL with four beams is overdetermined. Four components are used to calculate velocity in three directions. One pair of beams obtains one horizontal- and one vertical component, while the second pair obtains a perpendicular horizontal component and a second vertical component Indstruments (1996).

This yields the estimates of two horizontal velocities and two vertical velocities. As the beams samples measurements in different directions, the current velocity vector must be approximately equal in the horizontal plane to be able to apply trigonometry to find velocity. According to (Indstruments, 1996), horizontal homogeneity is often a reasonable assumption to consider. However, it is important the navigation system is informed when homogeneity is absent. Figure 4.1 portraits a horizontal view of two horizontal velocity cases: the left figure where horizontal velocity is present, and the right figure where one transducer beam has a noticeably different horizontal velocity. According to Indstruments (1996), situation 4.1b will on average have a larger error velocity than the situation illustrated in Figure 4.1a.



(a) All four beams have equal horizontal velocity. (b) One beam has a different horizontal velocity.

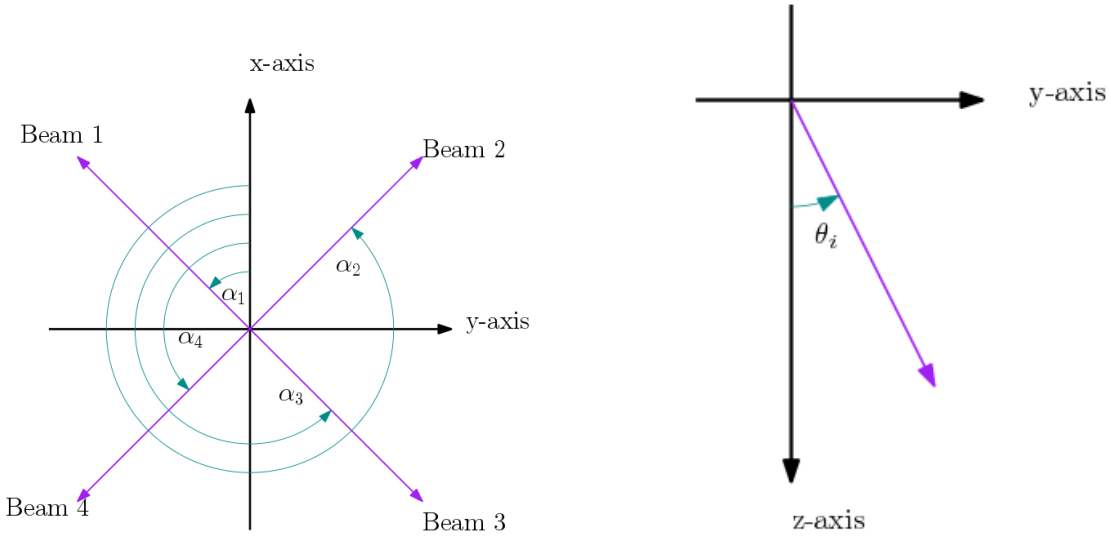
Figure 4.1: Figure credited to Indstruments (1996)

The error velocity is the difference between the two horizontal velocities. The parameter allows the system to evaluate whether the assumption concerning horizontal homogeneity is reasonable or not. This functionality provides integrity, as it allows the system to know when the velocity measurements are untrustworthy. In real life application, the error velocity will during ideal conditions fluctuate around zero, as the water will never be fully homogeneous.

Due to aforementioned reasons, the work in this thesis is thus made based on the assumption of horizontal homogenity is thus made by assuming that horizontal homogenity is present at all times.

The Scalar Unit Vector

Figure 4.2 is a display of a three dimensional vector being deconstructed along the x-,y-, and z-axis.



(a) The angle α_i displaying the beams' orientation with respect to the heading axis (x-axis) in a bottom-up perspective.

(b) The angle representing the beams' orientation with respect to the z-axis, is referred to as θ_i, where i is the beam labels.

Figure 4.2: DVL transducer beam configuration.

Using trigonometry, it is possible to create unit vectors along the transducer axis. Based on figure 4.2, the unit vector for beam i is:

$$\mathbf{u}_i = \begin{bmatrix} cos(\alpha_i)sin(\theta_i) \\ sin(\alpha_i)sin(\theta_i) \\ cos(\theta_i) \end{bmatrix} \tag{4.1}$$

The values initially set for the DVL configuration are listed in table 4.1.

Beam i	α_i	θ_i
Beam 1	45°	25°
Beam 2	315°	25°
Beam 3	225°	25°
Beam 4	135°	25°

Table 4.1: α and θ angles used in NAVLAB.

The values for i in 4.1 are stated in table 4.1. Note that the unit vector could just as easily been expressed as a matrix, by transposing $\mathbf{u}_i$ and stacking the the row vector on top of each other, as expressed in equation 4.2. However, this approach is not implemented for the calculations performed in MATLAB. The scalar variant described in Equation (4.1?) is easier to manipulate compared to the matrix expression in equation 4.2, and is thus the preferred approach in this work.

$$\mathbf{U} = \begin{bmatrix} cos(\alpha_1)sin(\theta_1) & sin(\alpha_1)sin(\theta_1) & cos(\theta_1) \\ cos(\alpha_2)sin(\theta_2) & sin(\alpha_2)sin(\theta_2) & cos(\theta_2) \\ cos(\alpha_3)sin(\theta_3) & sin(\alpha_3)sin(\theta_3) & cos(\theta_3) \\ cos(\alpha_4)sin(\theta_4) & sin(\alpha_4)sin(\theta_4) & cos(\theta_4) \end{bmatrix} \tag{4.2}$$

Expression 4.2 is not used for calculations in MATLAB, because the scalar variant is viewed by the author to be easier to manipulate.

Chapter 5

The Velocity Model to the Extended Error State Kalman Filter

The velocity input the Extended Error State Kalman Filter (EKF), is a sum of the measured and the estimated velocity. The Kalman filter states are therefore error parameters, and the resulting Kalman filter is called an *Error State Kalman filter*.

It is necessary to have at least two versions of the same parameter, obtained independently of each other, in order to produce the error states. The input to the EKF are therefore the difference between the parameters, e.g. $\hat{\mathbf{v}}^L_{EB} - \hat{\mathbf{v}}^L_{EB,DVL}$.

5.1 DVL Velocity Expression

The estimated velocity of the body (B) relative to earth (E) in the L-system (L) is the sum of the measured velocity of the body and the velocity of the DVL (D) in relationship with the body.

$$\hat{\mathbf{v}}^L_{EB,i} = (\mathbf{u}^B_i)^T \hat{\mathbf{R}}^L_B (\tilde{\mathbf{v}}^B_{EB} + \mathbf{v}^B_{BD}) + b \tag{5.1}$$

The velocity $\mathbf{v}^B_{BD}$ is approximately equal to zero as the DVL in rigidly attached to the AUV. Practically no motion exists between the two coordinate frames.

$$\hat{\mathbf{v}}^L_{EB,i} = (\mathbf{u}^B_i)^T \hat{\mathbf{R}}^L_B \tilde{\mathbf{v}}^B_{EB} + b \tag{5.2}$$

5.2 The Discrete Measurement Matrix $\mathbf{H}_k$

The objective of the following section is to identify the discrete measurement matrix $\mathbf{H}_k$ used in the ESKF. During operation the AUV will only receive measurements at discrete time steps. The measurement equation is therefore defined defined in a discrete manner, as in equation 5.3

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \xi_k \tag{5.3}$$

where $\mathbf{y}_k$ is the measurement, $\mathbf{x}_k$ is the states, and ξ_k is the process noise. To find the discrete measurement matrix $\mathbf{H}_k$ the input expression to the filter will be stated and linearized. The result will be an expression for $\delta\mathbf{y}_k$ The general error velocity measurement equation is equal to the measured velocity ($\tilde{\mathbf{y}}$) minus the estimated velocity ($\tilde{\mathbf{y}}$), expressed as

$$\delta\mathbf{y} = \tilde{\mathbf{y}} - \hat{\mathbf{y}} \tag{5.4}$$

The nonlinear states of the KF are

$$h_i(\mathbf{x}) = (\mathbf{u}_i^B)^T \mathbf{R}_L^B \mathbf{v}_{EB}^L + b \tag{5.5}$$

and the perturbed nonlinear states are

$$h_i(\mathbf{x}+\delta\mathbf{x}) = (\mathbf{u}_i^B)^T [(\mathbf{R}_L^B + \delta\mathbf{R}_L^B)(\mathbf{v}_{EB}^L + \delta\mathbf{v}_{EB}^L) + (b+\delta b)] \tag{5.6}$$

The linearized expression take the form as shown in equation 5.7, where ξ_{DVL} is the measurement noise.

$$\delta\mathbf{y} = h(\mathbf{x}) - h(\mathbf{x}+\delta\mathbf{x}) + \xi_{DVL} \tag{5.7}$$

By inserting the expressions from equation 5.5 and 5.6 into equation 5.7, it is possible to obtain a new representation of the system. This is expressed in equation 5.8 - 5.10 for each system state.

$$\delta\mathbf{y}_i = ((\mathbf{u}_i^B)^T \mathbf{R}_L^B \mathbf{v}_{EB}^L + b) - ((\mathbf{u}_i^B)^T (\mathbf{R}_L^B + \delta\mathbf{R}_L^B)(\mathbf{v}_{EB}^L + \delta\mathbf{v}_{EB}^L) + (b+\delta b)) + \xi_{DVL} \tag{5.8}$$

$$\delta\mathbf{y}_i = ((\mathbf{u}_i^B)^T \mathbf{R}_L^B \mathbf{v}_{EB}^L + b) - ((\mathbf{u}_i^B)^T (\mathbf{R}_L^B \mathbf{v}_{EB}^L + \mathbf{R}_L^B \delta\mathbf{v}_{EB}^L + \delta\mathbf{R}_L^B \mathbf{v}_{EB}^L + \delta\mathbf{R}_L^B \delta\mathbf{v}_{EB}^L) + b + \delta b) + \xi_{DVL} \tag{5.9}$$

$$\delta\mathbf{y}_i = -((\mathbf{u}_i^B)^T (\mathbf{R}_L^B \delta\mathbf{v}_{EB}^L + \delta\mathbf{R}_L^B \mathbf{v}_{EB}^L) - \delta b + \xi_{DVL} \tag{5.10}$$

(Gade, 1997) demonstrated that the expression given in equation 5.11 is valid. The parameter $\mathbf{e}_{BL}^B$ is a vector describing the difference in orientation between the B- and L-system, expressed in the B system. The skewness between the B- and L-system is denoted as $\delta\mathbf{R}_L^B$. The skewness can be interpreted as the error in orientation between the two systems.

$$\delta\mathbf{R}_L^B = \mathbf{S}(\mathbf{e}_{BL}^B)\mathbf{R}_L^B = \mathbf{R}_L^B \mathbf{S}(\mathbf{e}_{BL}^L) \tag{5.11}$$

Applying equation 5.11 to 5.12 yields a new expression containing the error of the estimated orientation between system B and L, $\hat{\mathbf{R}}_L^B$.

$$\delta\mathbf{y}_i = -((\mathbf{u}_i^B)^T (\mathbf{R}_L^B \delta\mathbf{v}_{EB}^L + \mathbf{R}_L^B \mathbf{S}(\mathbf{e}_{BL}^L)\mathbf{v}_{EB}^L) - \delta b + \xi_{DVL} \tag{5.12}$$

Linear theory states that a simple linear system $\mathbf{u}\times\mathbf{v}$ can be expressed as $-\mathbf{v}\times\mathbf{u}$. The same linear operation is performed on $S(\mathbf{e}^B_{BL})$ and $\mathbf{v}^L_{EB}$ to extract the error on vector form, see equation 5.13.

$$\delta\mathbf{y}_i = -(\mathbf{u}^B_i)^T(\mathbf{R}^B_L\delta\mathbf{v}^L_{EB} - \mathbf{R}^B_L\mathbf{S}(\mathbf{v}^L_{EB})\mathbf{e}^L_{BL}) - \delta b + \xi_{DVL} \tag{5.13}$$

Similarly, $\mathbf{e}^L_{BL}$ can be expressed as $-\mathbf{e}^L_{LB}$. Inserting $-\mathbf{e}^L_{LB}$ into equation 5.13 results in a new expression for the measurement equation for each transducer beam, see equation 5.14.

$$\delta\mathbf{y}_i = -(\mathbf{u}^B_i)^T(\mathbf{R}^B_L\delta\mathbf{v}^L_{EB} + \mathbf{R}^B_L\mathbf{S}(\mathbf{v}^L_{EB})\mathbf{e}^L_{LB}) - \delta b + \xi_{DVL} \tag{5.14}$$

Factoring out the system states from equation 5.14 results in a state-space representation shown in equation 5.3. Equation 5.15 yields the discrete measurements for each beam where the matrix H_k denotes the relationship between the system states. The measurement matrix, H_k, is then implemented in the Kalman filter. The subscript k is added to denote time step.

$$\delta\mathbf{y}_{k,i} = \underbrace{\begin{bmatrix} -(\mathbf{u}^B_{i,k})^T\mathbf{R}^B_{L,k}S(\mathbf{v}^L_{EB,k}) & -(\mathbf{u}^B_{i,k})^T\mathbf{R}^B_{L,k} & -1 \end{bmatrix}}_{H_k} \underbrace{\begin{bmatrix} \mathbf{e}^L_{LB,k} \\ \delta\mathbf{v}^L_{EB,k} \\ \delta b_k \end{bmatrix}}_{x_k} + \xi_{DVL,k} \tag{5.15}$$

5.3 Verification of H_k

A verification of the measurement matrix H_k is done in B.1. The test verified the matrix for use.

5.4 DVL Error Model

As described in the Introduction chapter of this thesis, the DVL sensor used to gather data in this work is a Nortek 500 DVL (NORTEK, 2018), which has a documented accuracy of 0.2% of speed. This high accuracy comes at the compliment of such a high frequency. Jalving et al. (2004) stated that the DVL accuracy is dependent on frequency, where higher frequency yield a better accuracy at the cost of a decreased DVL range.

According to (Gade, 1997), the error of $\hat{\mathbf{v}}^L_{EB}$ was found to be approximately white, w_{DVL}. (Gade, 1997) states that velocity noise along and across the trajectory can be assumed uncorrelated as they are calculated independently of each other. The velocity noise in the z-direction will in practice be correlated with noise in x- and y- direction. (Gade, 1997) does not emphasize the modeling of this error, as it only contribute with vertical position error with small pitch-angles. Large pitch angles are only explicitly used during accent and decent of the vehicle/sensor, i.e. when requirement for position accuracy is relatively low.

The measurement noise for the DVL is assumed Gaussian distributed:

$$\xi_{DVL} = \mathcal{N}(0, W_{DVL,k}\delta_{kj}) \tag{5.16}$$

where the measurement noise is set equal

$$W_{DVL,k} = \left[\sigma^2_{w_{DVL,k}}\right] \tag{5.17}$$

The measurement noise is unchanged $\sigma^2_{w_{DVL,i,k}} = 0.008$ m/s.

5.5 Erroneous Velocity Sources

The velocity updates inside the Kalman filter has to account for wrongful contributions to the velocity, for instance angular velocity of the AUV contributed in a small way to the velocity measured in the DVL. A method called *lever arm compensation* is required to mend this problem.

5.5.1 Lever Arm Compensation

As the DVL is a certain distance away from the IMU, the DVL will have an angular velocity about the IMU. The angular velocity yields a tangential velocity at the DVL that must be accounted for in the filter. Figure 5.1 is showing the resulting tangential velocity in question. The expression

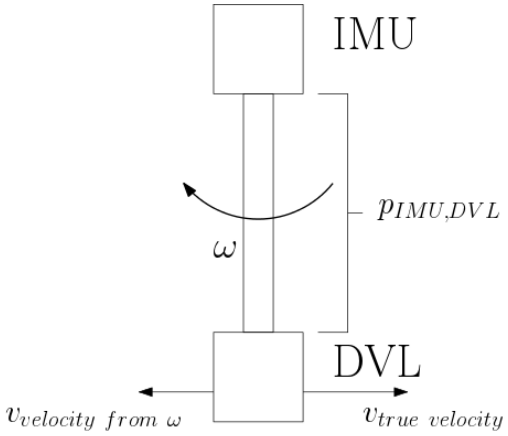


Figure 5.1: The velocity measured by the DVL is subjected to the angular velocity ω of the AUV. $p_{IMU,DVL}$ denotes the distance from the center of the IMU to center of the DVL

for the tangential velocity is summarized in equation 5.18, where $\hat{\mathbf{v}}_{EB}$ is the estimated velocity.

$$\mathbf{v}_{EB,true} = \tilde{\mathbf{v}}_{EB} - \omega_{\text{current velocity}} \cdot p_{BD} \tag{5.18}$$

Equation A.5 shows that a velocity vector in coordinate system B, translated to coordinate

system A is equal to the velocity vector in B transformed into A, plus the angular velocity times the distance, rotated into the A system.

In the context regarding velocity between the DVL and IMU, equation A.5 becomes

$$\mathbf{v}_{BD,i}^{B} = (\mathbf{u}_{i}^{B})^{T}\mathbf{v}_{BD}^{B} + (\mathbf{u}_{i}^{B})^{T}\mathbf{S}(\omega_{LB}^{B})\mathbf{p}_{BD}^{B} \tag{5.19}$$

The angular velocity of the body, with reference to the L-system ω_{LB} is a sum of the angular velocity of

- the body system, relative to inertial space
- the earth system, relative to inertial space
- the L-system, relative to the earth system

This is summarized in equation 5.20.

$$\omega_{LB,i} = \omega_{IB} - \omega_{IE} - \omega_{EL} \tag{5.20}$$

ω_{LB} is the angular velocity $\frac{deg}{s}$ the body (AUV) has relative to the L-system, ω_{IB} is the angular velocity measured by the gyros on-board the AUV, ω_{IE} is the rotational velocity of the Earth, and ω_{EL} is a function of the AUV position and velocity. Generally, the angular velocity of the AUV is significantly larger compared to both the rotation velocity of both the rotational velocity of the Earth and the angular velocity the L-system has in reference to Earth, combined. Therefore, it can be assumed that

$$\omega_{LB} \approx \omega_{IB} \tag{5.21}$$

Equation 5.19 can thus be expressed on the form

$$\mathbf{v}_{BD}^{L} = (\mathbf{u}_{i}^{B})^{T}\mathbf{S}(\omega_{IB}^{B})\mathbf{p}_{BD}^{B} \tag{5.22}$$

The angular velocity is estimated by measuring the change in angles measured by gyros on-board, and multiplying with the data rate of the IMU. The IMU system used has a data rate of 100 Hz.

5.5.2 DVL Position Accuracy

The horizontal drift in position is according to Jalving et al. (2004) determined by the error in the estimated north and east velocity. The main contributors of this error are the error in body-fixed velocity and the error in heading. The error in $v_{EB,x}$ and $v_{EB,y}$ are determined by the low-frequency error in the DVL. When position measurements are not available, the scale factor

error is not observable when traveling along straight trajectories. This is demonstrated chapter 8 and chapter 9.

According to Jalving et al. (2004), the INS will almost momentarily gain a higher velocity uncertainty compared to the DVI accuracy in situations without position aiding, regardless of the initial accuracy of the INS. This in turn favors the DVL NORTEK 500 described in chapter 4, empathizing velocity accuracy of 0.2% of traveled distance. By considering for instance a velocity of $2\frac{m}{s}$ along on axis, the along track error drift cumulates to 0.2% of the traveled distance, corresponding to 14.4 m/hour. It is suggested in Jalving et al. (2004) that the problem of unobservable scale factor can be partially mended by either performing a lawn-mower pattern in which all the turns in this pattern will allow the scale factor error to be observable, or by rotating 360' at regular intervals.(all the turns in such a pattern will allow the scale factor error to be observable) or by rotating 360^o at regular intervals. The latter method require accurate DVL time stamping: the requirement set by Jalving et al. (2004) is that the error produced by incorrect time stamps stemming from vehicle acceleration and angular acceleration should be less than the scale factor error. It is therefore of interest to examine how the position accuracy behaves in periods when turning and straight lines.

5.5.3 DVL Configuration Misalignment

The angle values in table 4.1 do not accounted for misalignments in the AUV, used to collect the data. If the DVL configuration in table 4.1 are implemented, the result will most likely be incorrect, as the angles have not been calibrated. The calibration is performed by running recorded navigation data with a DVl through a calibration process to calculate alignment and other fixed parameters of the DVL-IMU combination. A calibration was done at Kongsberg Maritime, with the result of

$$DVL_{correction} = \begin{bmatrix} roll_{correction} \\ pitch_{correction} \\ yaw_{correction} \end{bmatrix} = \begin{bmatrix} -0.2033° \\ -0.0341° \\ -0.5414° \end{bmatrix} \tag{5.23}$$

These Euler angles are then calculated into a rotation matrix, and used to rotate the values displayed in table 4.1, as seen in equation 5.24. A rotation matrix $\mathbf{R}_{New}^{Old}$ is designed based on the values in 5.23.

$$\mathbf{u}_B, corrected = \mathbf{R}_{New}^{Old} * \mathbf{u}_B \tag{5.24}$$

Chapter 6

Real-Time Testing and Scenario Development

This chapter explains how the testing scenarios are set up and which period of the mission is of special interest.

6.1 Setup

The mission from where the data was gathered had a duration of five and a half hours long in its entirety. But the only time period of interest is where the DVL is active, and acquiring measurements, see figure 6.1. As the figure shows, there are no significant large periods of time where the measurements from each beam are not continuous. There is a small period of time right after 4000 seconds where the position measurements were interrupted, but this event in unaffected by the results presented in this thesis, as the testing time in this thesis focuses on starts after the event.

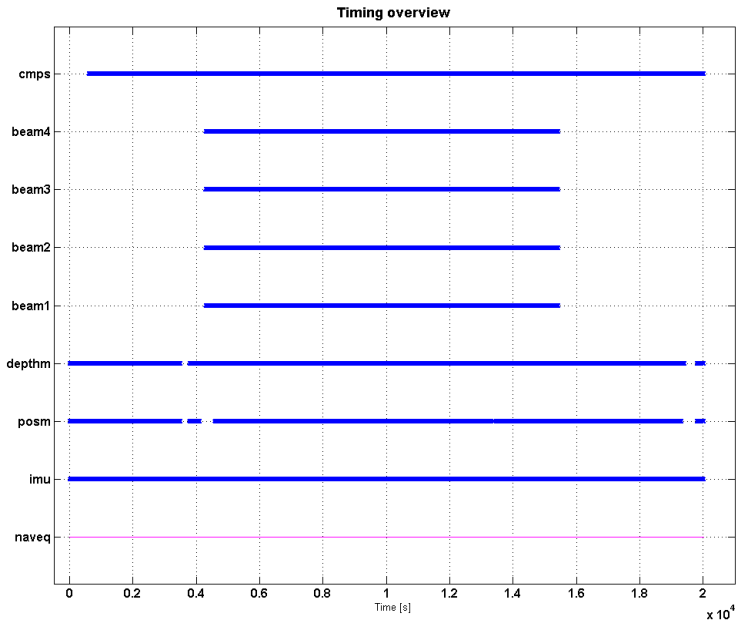


Figure 6.1: The gaps are time periods where no measurement was registered.

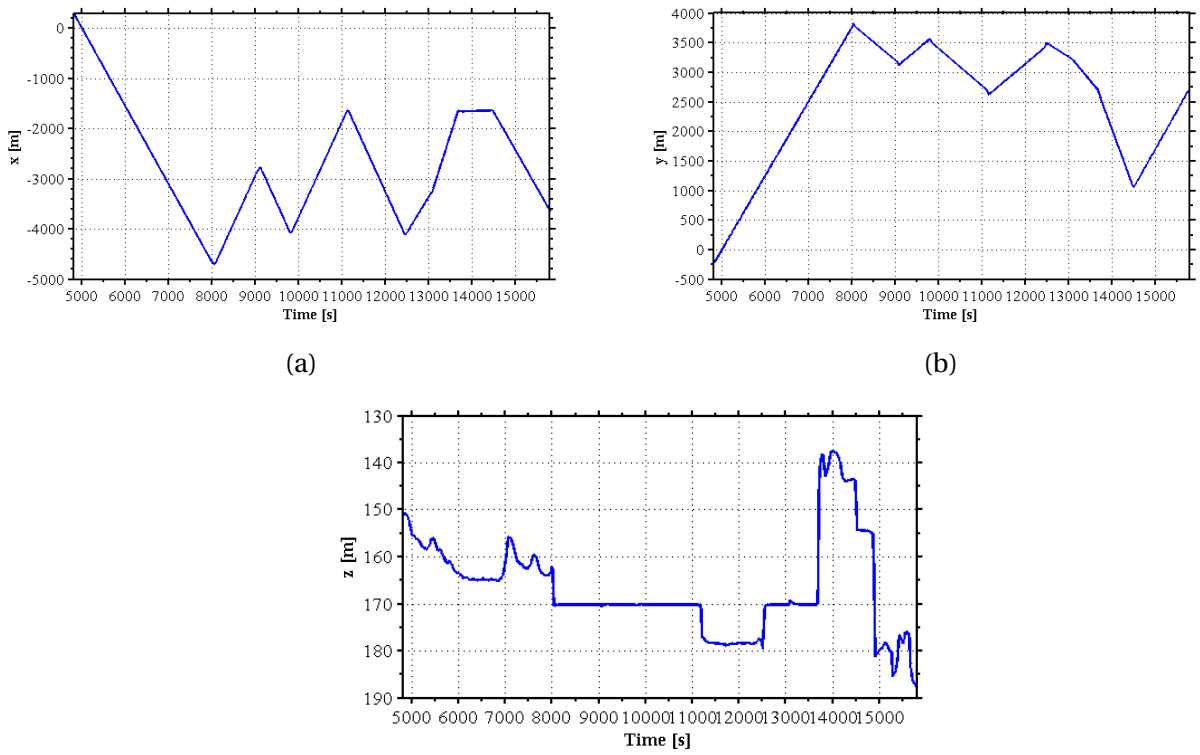
"The DVL only recorded measurements within the time range from approximately 4300 to 15800 seconds. The time span used in further evaluations will be from 4800 to 15800 seconds. The reason why the testing start some 500 seconds after DVL measurements are available is to allow the system time to settle. It is assumed no previous knowledge of the sensors before start, so the initial covariances are set high to account for the errors in the beginning.

Figure containing information regarding the beams in chapter chapter 7 and chapter 8 are positioned so the figure mirrors the position of the actual beam. See figure 4.2 for clarification.

6.2 Plots of Position

The measured position in relationship to coordinate system M is shown in figure 6.2. Coordinate system M is a two dimensional tangent coordinate system used for plotting the position of the AUV. The purpose of M is to map the RT position of the AUV. The need for this coordinate system is rooted in readability; the RT position is calculated in three dimensional coordinates (x,y,z), so M removes the z and transposes the remaining axis to a two dimensional plane. Depth is included to complete the positioning.

The position measurements depicted used for chapter 7 to 9 are depicted in figure 6.2.



(a) (b)

(c)

Figure 6.2: The route the AUV will travel.

6.3 Periods of Interest and Comparison Basis

According to Willumsen and Hegrenaes (2009) and Jalving et al. (2004), the trajectory that poses the largest challenge to estimate states, is along a straight line. It is of interest to examine how the across- and along track error is affected. At points where the AUV changes direction it is expected that these errors will be reduced, as the scale-factor error becomes observable during turning Jalving et al. (2004).

The main comparison basis between the current and new DVL implementation is to analyze how well they perform without position updates. The computations are done on real-time data, meaning that the true value of any reading is unknown. Therefore it is of interest to examine how well the navigation system handles periods where no position measurement is available.

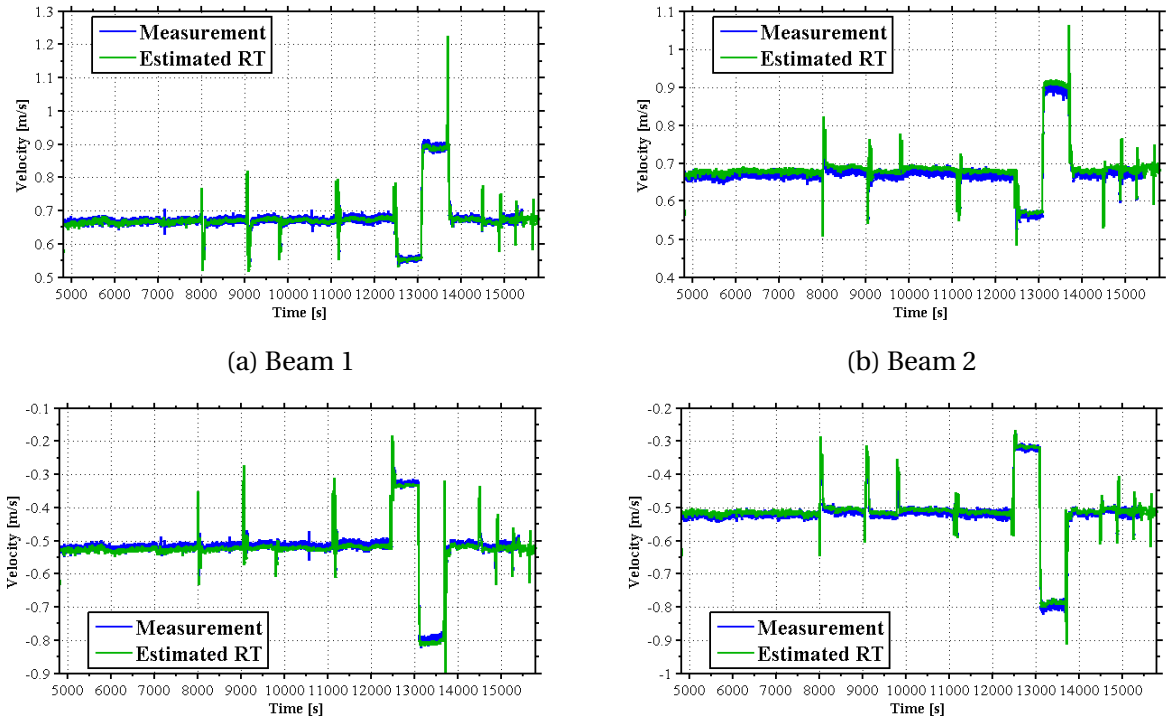
Chapter 7

Proof of Concept - Tightly integrated DVL

The aim of this chapter is to demonstrate that the methods of tightly integrated DVL is feasible. This means that the estimates are not overly erratic in behavior. The discussion on whether the tightly integrated solution is better or worse than the current solution will not be further discussed in chapter 7.

7.1 Velocity Measurements

The raw beam velocity measurements are displayed in figure 7.1. The obtained beam measurements are displayed in figure 7.1 and highlights the behavior over time for the four beams. The velocity in beam 3 and 4 is negative. This stems from the configuration of the DVL beam transducers seen in figure 4.2, where the beams are positioned/oriented in the opposite x-direction of the heading, resulting in negative measurements.



(a) Beam 1

(b) Beam 2

(c) Beam 4

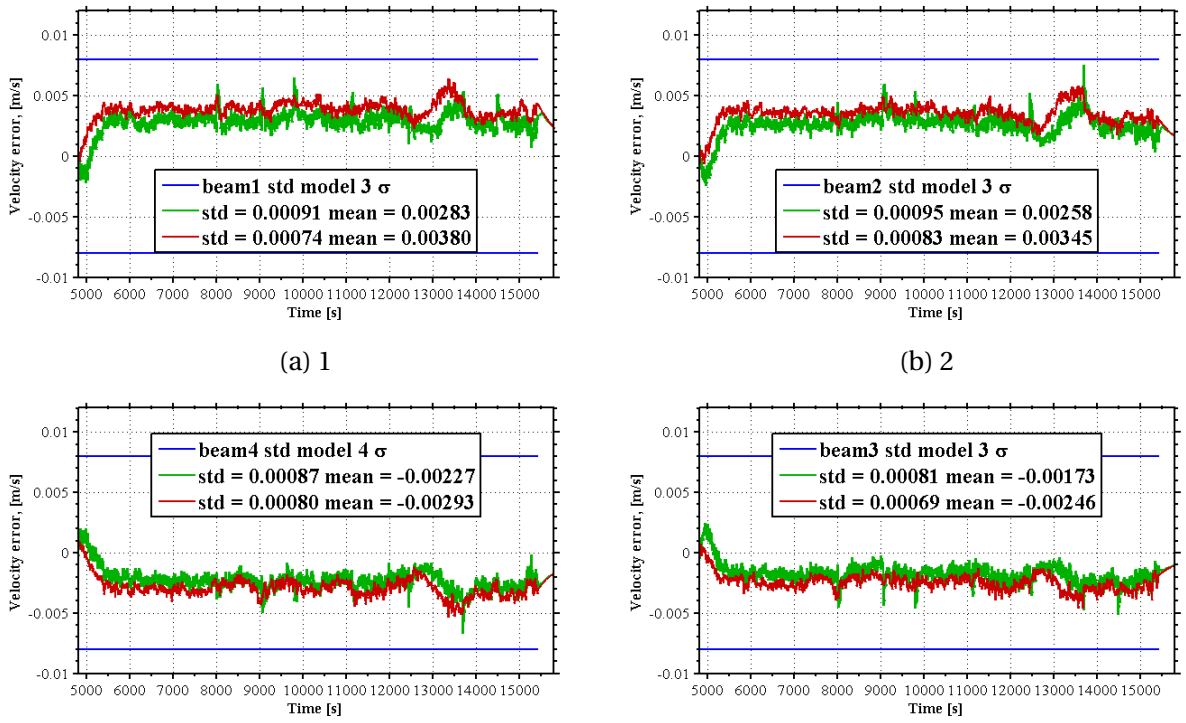
(d) Beam 3

Figure 7.1: Measured and estimated real-time velocity for beams 1 to 4

From figure 7.1 a relative large difference in velolcity in the interval from 12 000 to 14 000 seconds is observed. This change in velocity is most likely caused by ascension of the AUV. During ascension and descension, the DVL is unreliable in acquiering velocity data.

7.1.1 Error in real-time estimated velocity

The real-time velocity errors are either offset to the positive y-axis, or the negative y-axis, depending on the direction of the beam measurement.



(a) 1

(b) 2

(c) 4

(d) 3

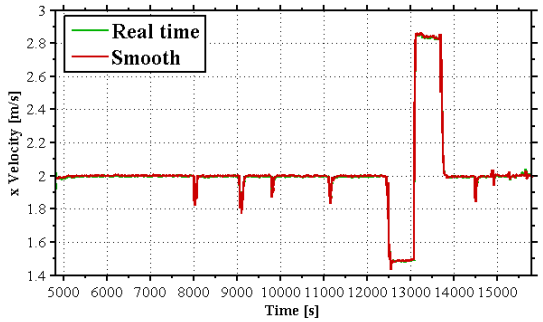
Figure 7.2: The error between the real time estimates and measurements, and the smoothed value of the estimate.

7.2 Estimated Autonomous Underwater Vehicle Velocities

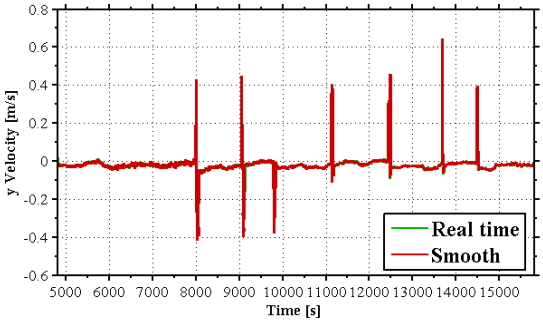
The velocity of the AUV in the B system for Method 1 is shown in figure 7.3. The velocity in all three directions inhibit relatively low oscillations, except at points where the AUV is turning.

7.2.1 Estimated AUV velocity, Method 1

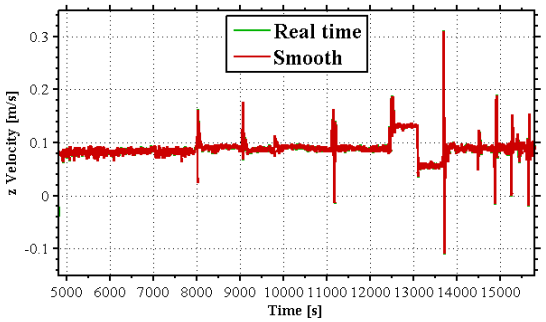
The AUV velocity based on the tightly integrated DVL implementation is shown in figure 7.3. The figure shows a stable x-velocity approximately equal to $2\frac{m}{s}$, except during ramp changes in the z-velocity.



(a) Estimated velocity in x direction



(b) Estimated velocity in y direction

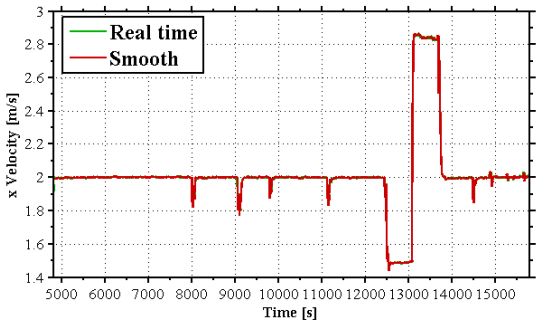


(c) Estimated velocity in z direction

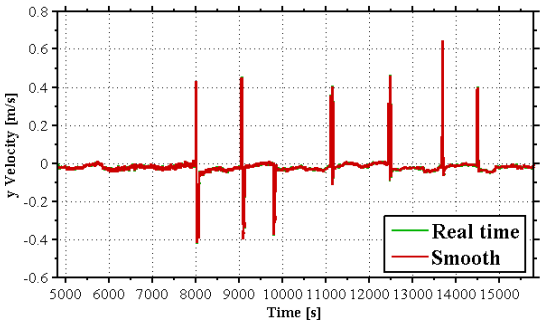
Figure 7.3: Real-time estimated AUV velocity for Method 1 with position updates.

7.2.2 Estimated AUV velocity, Method 2

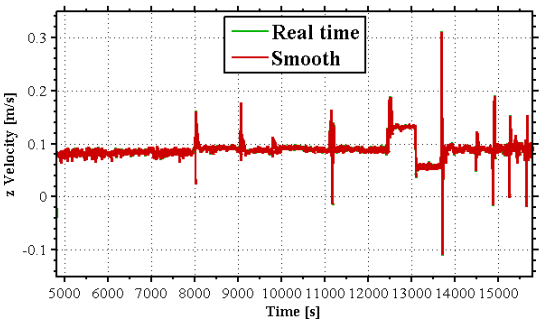
Following are the velocities of the currently implemented DVL system. This plot is included display that Method 1 yields results close to Method 2.



(a) Estimated velocity in x direction



(b) Estimated velocity in y direction



(c) Estimated velocity in z direction

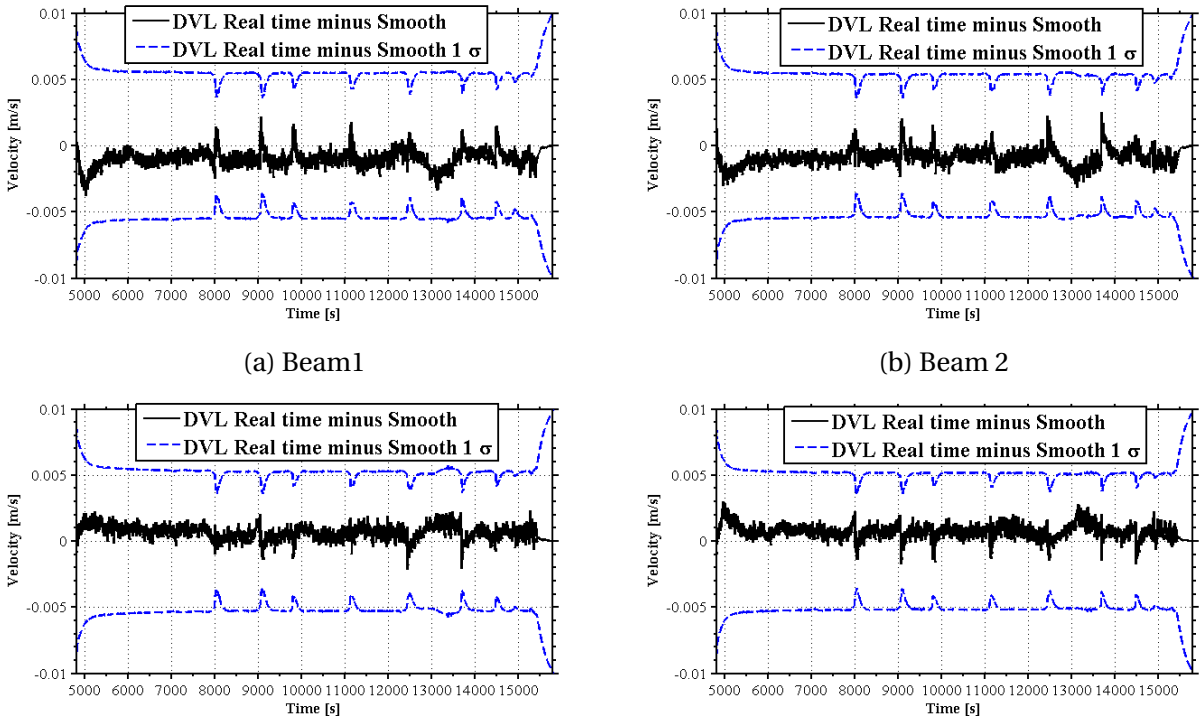
Figure 7.4: Real-time estimated AUV velocity for Method 2 with position updates.

7.3 Real Time Velocity minus Smooth Velocity

The real time estimates minus the smooth estimates yields a quantity close to the true error, as the smoothed value is the most accurate reading available. The σ bound in the real time estimated standard deviation.

7.3.1 Real time minus smooth velocity for Method 1

The real time estimated velocity minus smooth velocity for Method 1 is shown in figure 7.5.



(a) Beam1

(b) Beam 2

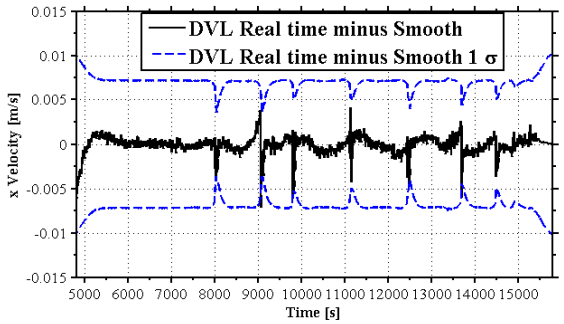
(c) Beam 4

(d) Beam 4

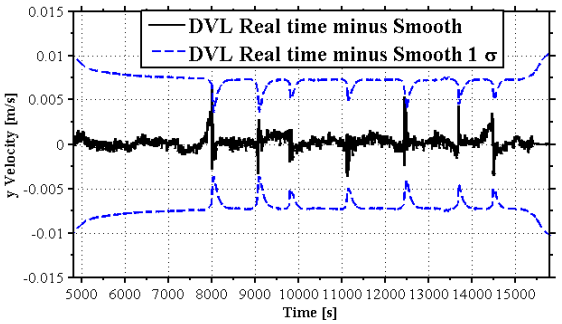
Figure 7.5: Real time velcoity minus smooth velocity for Method 1

7.3.2 Real time minus smooth velocity for Method 2

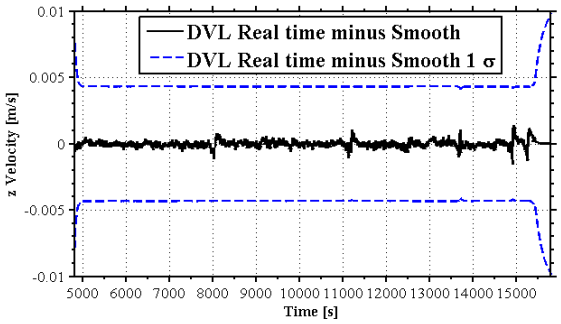
The real time velocity, minus the smooth velocity for Method 2 is shown in figure 7.6.



(a) Estimated real time velocity minus smooth velocity in x direction



(b) Estimated real time velocity minus smooth velocity in y direction

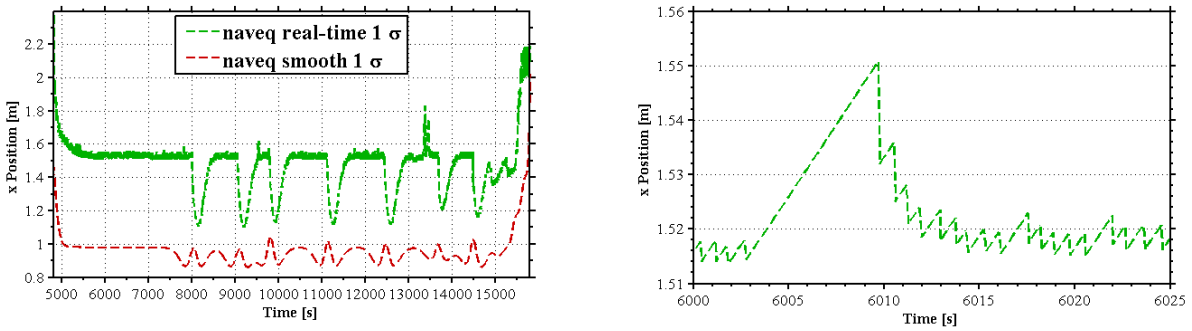


(c) Estimated real time velocity minus smooth velocity in z direction

Figure 7.6: Estimated real time velocity minus smooth velocity for Method 2 with position updates.

7.4 Error in Navigational Position

Only the navigational position error in the x-direction is shown because the error is essentially the same in the y-direction. The real time error starts initially approximately 20 meters (not shown in figure 7.7). Figure 7.7b is a zoomed in version of the left figure, and is included so that the ridges are properly displayed. When the ridge is directed downwards, the navigational system has just received a measurement update, resulting in a decreasing error. With time, the uncertainty of the estimated parameters increases, which causes the increasing error.



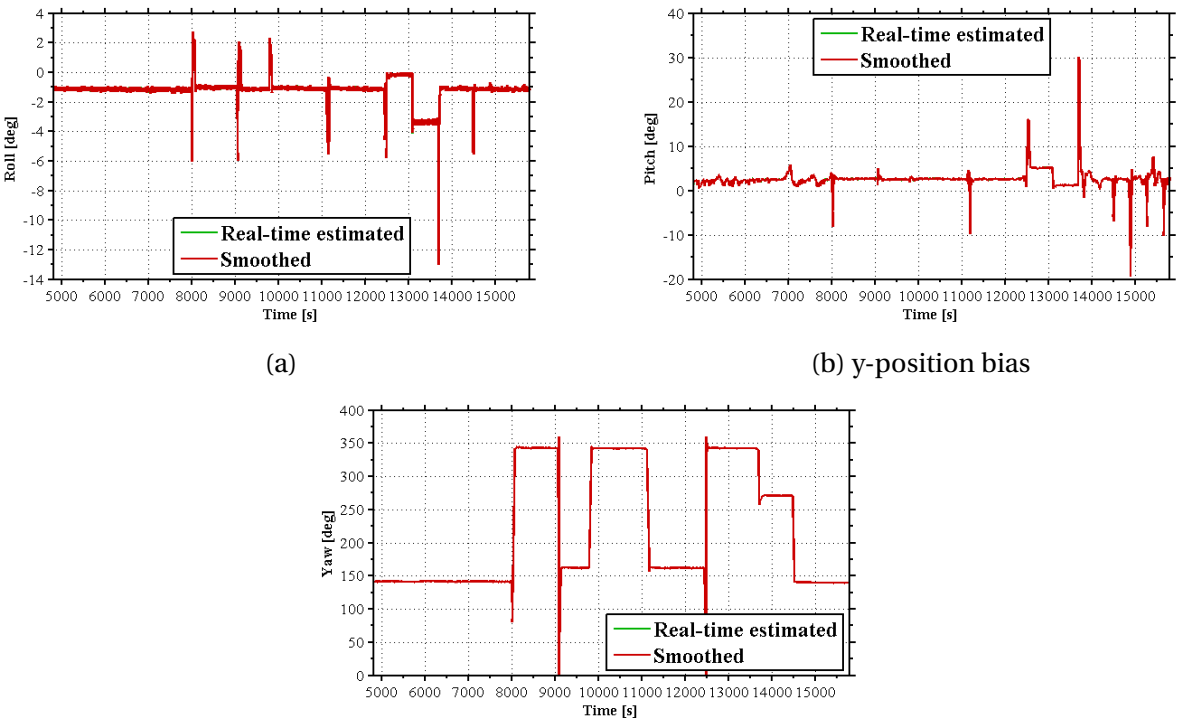
(a) x-position error

(b) A closer look on the ridges in figure 7.7a.

Figure 7.7: The standard deviation for the filters position estimate.

7.5 Attitude Angles

The orientation of the AUV during testing is estimated as shown in figure 7.8. The large yaw angles are caused by the AUV changing direction in the xy-plane, while the pitch spikes are correlated to the change in depth.



(a)

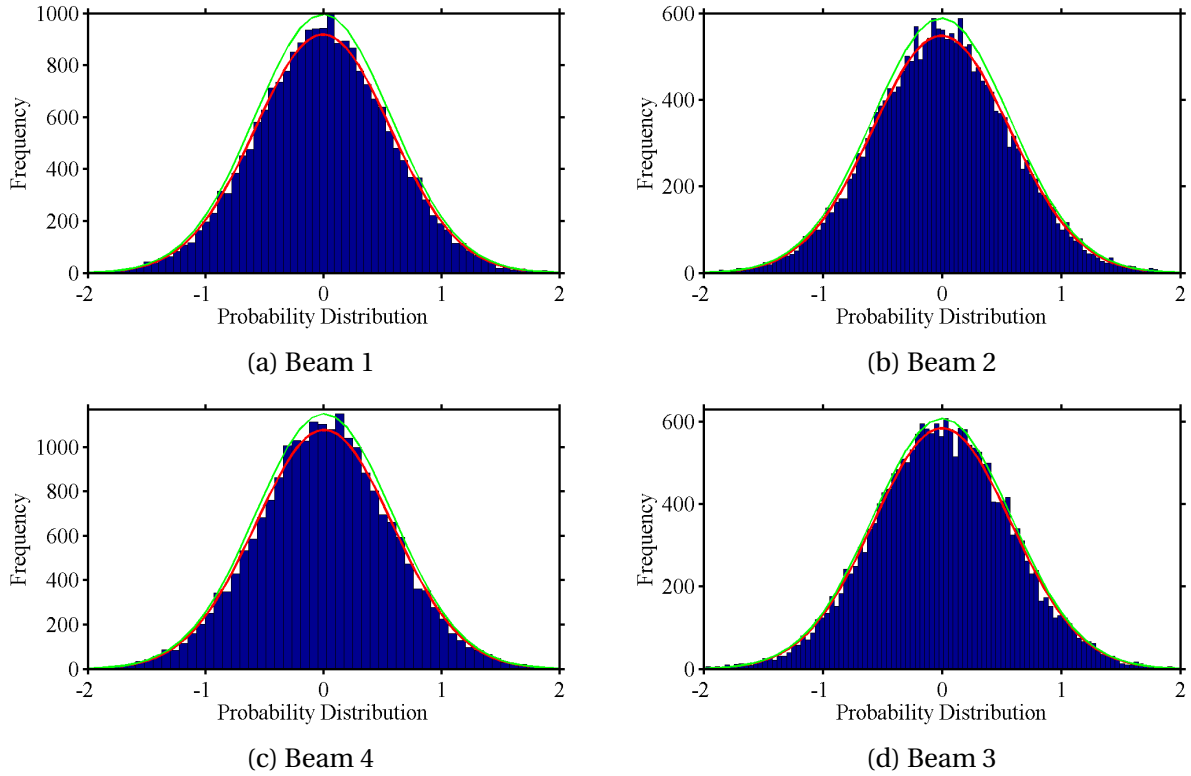
(b) y-position bias

(c) y-position bias

Figure 7.8: Real-time estimated and smoothed attitude angles. The green real time plot lies almost exactly under the red smoothed curve and is not visible in the figure.

7.6 Filter Consistency Test for Method 1

The filter consistency test mentioned in section 2.5 is done for Method 1, and its result is shown in figure 7.9.



(a) Beam 1

(b) Beam 2

(c) Beam 4

(d) Beam 3

Figure 7.9: Innovation squared for all four beams.

The characteristics of the fitted Gaussian are listed in table 7.1.

Beam$_i$	$\hat{\mu}_i$	$\hat{\sigma}_i$
1	-0.0072	0.5741
2	-0.0074	0.5746
3	0.000033	0.5792
4	0.0026	0.5800

Table 7.1: Means ($\hat{\mu}_i$) and standard deviations ($\hat{\sigma}_i$) of the fitted Gaussian distribution (red line in figure 7.9)

7.7 Performance of Concept

Based on the overall impression given by the results in this chapter, Method 1 is clearly a suitable candidate to replace Method 2. The Kalman filter does not struggle with estimating DVL

measurements, as seen in figure 7.1. The error between measured and estimated velocity is well within the boundary of 1σ ($0.008\frac{m}{s}$) throughout the mission, even when the AUV is changing direction. Method 1 estimates the attitude angles close to identically as Method 2.

In section 7.3 the real time velocity estimate are subtracted from the smoothed estimate. As the smoothed estimate are the closest measurement to the true value available, it can be subtracted from the estimate to find the most feasible error. The errors in figures 7.5 are all close to zero, except for some fluctuations. The error is relatively larger than the trend when the AUV is changing direction. In figure 7.6, the error is noticeably larger in the two upper plots. The lower plot, which depict the error in the downward direction is close to zero. The reason the error in 7.6c is so small may be credited to the face that depth is not crucial to find position, except at decent and ascent, as was mentioned in section 5.4 The characteristics of the depth sensor are unknown to the author, so no assumptions regarding depth sensor will be taken. In figures 7.6a and 7.6b the error is larger than in figure 7.5. At turning points the error approaches the estimated standard deviation, meaning that the estimates became more different from the measurements.

The navigational error is shown in figure 7.7. Along straight trajectories the navigational error lies at approximately 1.5 meters. The navigational system receives position updates at regular intervals, so the error does not grow beyond a certain limit, as was mentioned in section 5.5.2. When turning, the navigational error decreases with a factor of 75%, down to 1.75 meters. This error reduction stems from the scale factor error being partially observable during turns, also mentioned in 5.5.2.

The filter consistency check done in section 7.6 shows that the filter yields normalized squared innovations that closely resembles Gaussian probability distribution. The mean values μ_i from table 7.1 are all close to zero (as is ideal for a Gaussian). The size of the standard deviations are relatively close to each other. Such low standard deviations are indications that the added measurement noise ξ is too high. The measurement noise is currently set to $0.008\frac{m}{s}$.

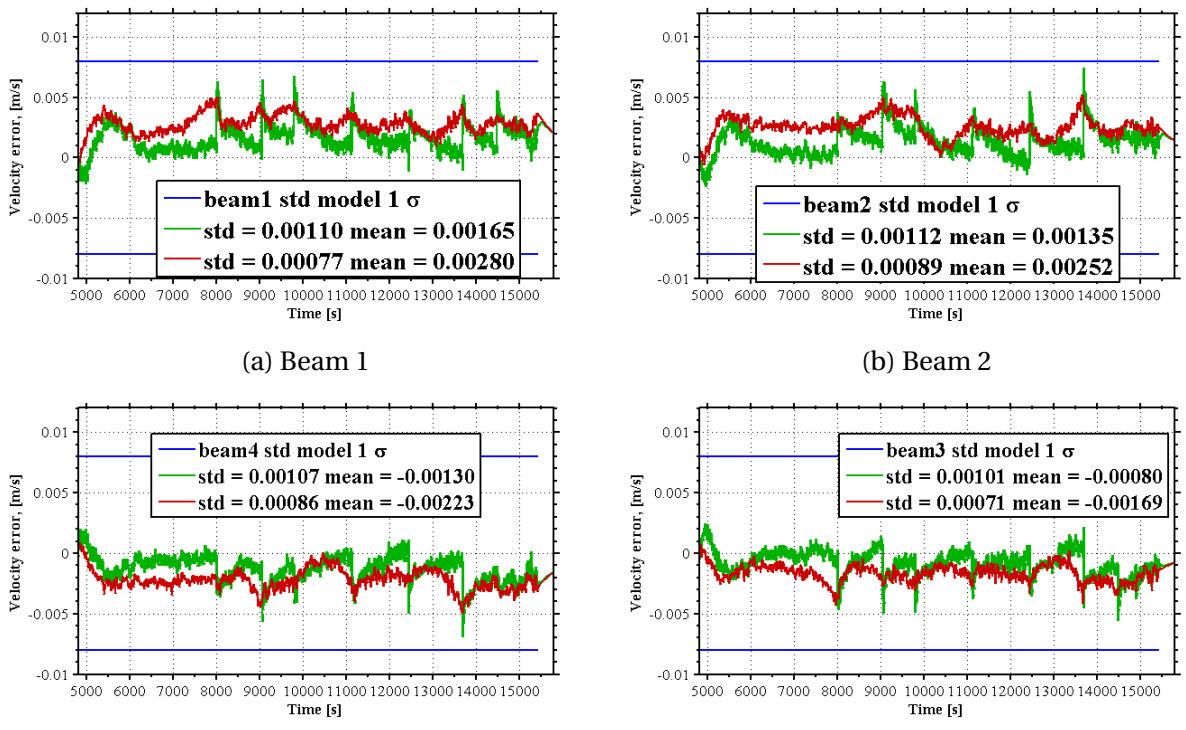
Chapter 8

Method 1: Excluding Position Updates

This chapter presents the results of a real-time testing of the tightly integrated DVL implementation without position updates from 5600 seconds to 15200 seconds.

8.1 Error in Real Time Estimated Velocity

The real error in real time estimated velocity is shown in figure 8.1.



(a) Beam 1

(b) Beam 2

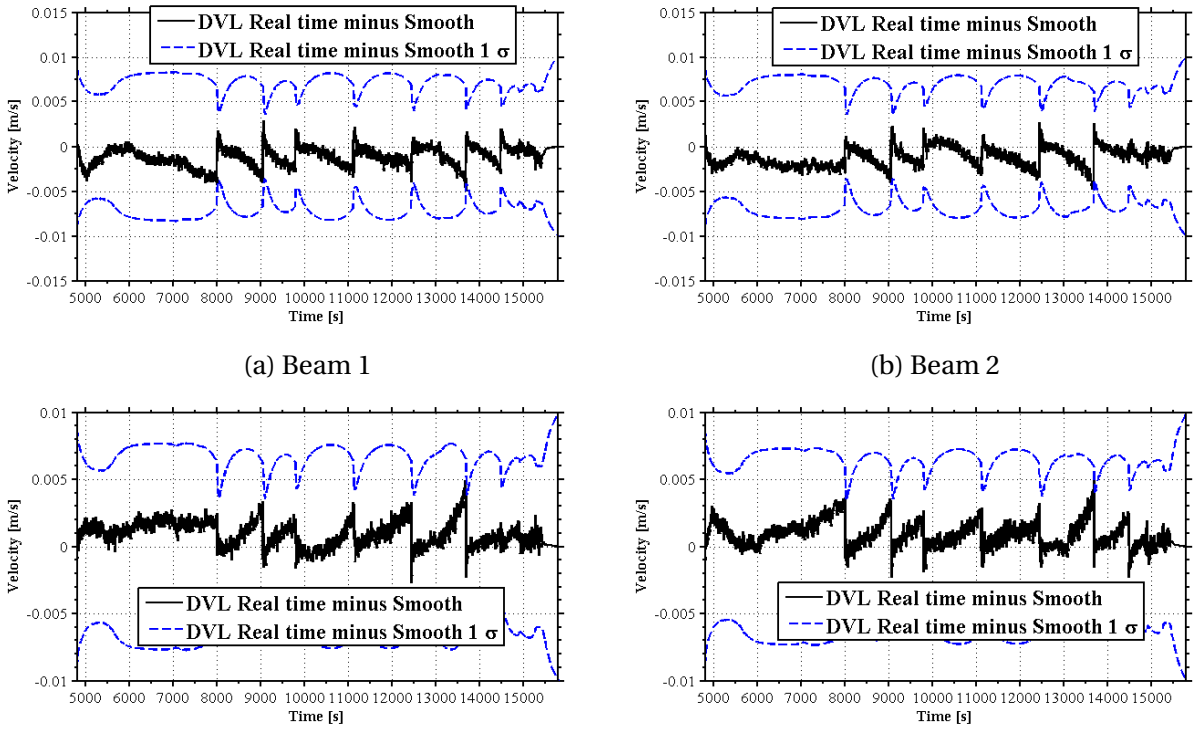
(c) Beam 4

(d) Beam 3

Figure 8.1: Real-time estimated and smooth velocity error

8.2 Real time Velocity minus Smooth Velocity

The real time velocity minus the smooth velocity is depicted in figure 8.2.



(a) Beam 1

(b) Beam 2

(c) Beam 4

(d) Beam 3

Figure 8.2: Real time velocity minus smooth velocity for Method 1 without position updates.

8.3 Error in navigational position

The navigational error of the navigational system is depicted in figure 8.3.

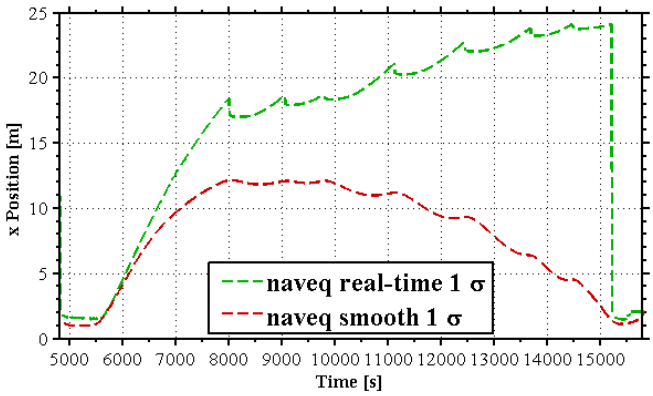
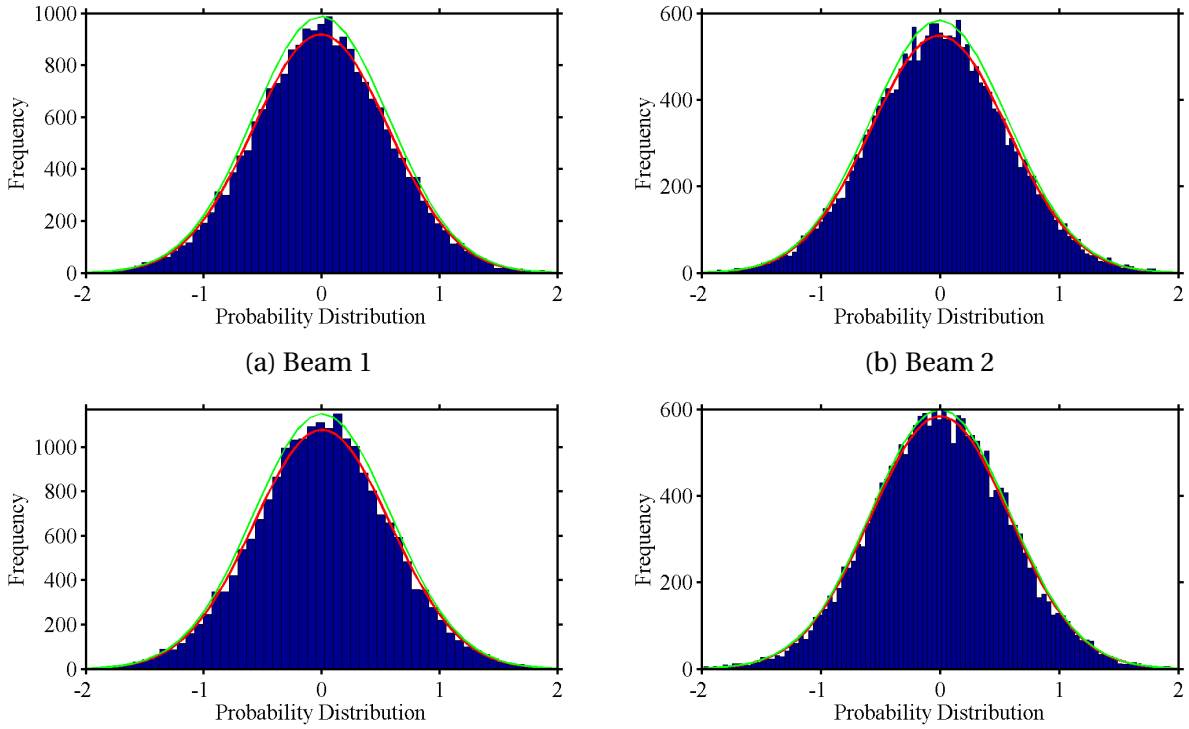


Figure 8.3: The standard deviation for the filters position estimate. The standard deviation in x and y are approximately the same (the difference is in the 10^3 range).

8.4 Filter Consistency Test

In accordance with 2.5.1, the histograms of the squared innovations are shown in figure 8.4 with corresponding Gaussian values in table 8.1.



(a) Beam 1

(b) Beam 2

(c) Beam 4

(d) Beam 3

Figure 8.4: NIS for all velocity beams for Method 1 without position updates

	$\hat{\mu}$	$\hat{\sigma}$
Beam 1	-0.0077	0.5740
Beam 2	-0.0076	0.5746
Beam 3	0.000473	0.5792
Beam 4	0.0027	0.5800

Table 8.1: Standard deviation and mean based off of the fitted Gaussian distribution from figure 8.4

8.5 Performance of Method 1

The difference between measured velocity and real time estimated velocity is plotted in figure 8.1. Comparing the means in 8.1 to figure 7.2, a slight decrease is registered. This suggests that the error is on average closer to zero without position updates than without positioning aiding. Note that the actual beam measurements do not change; only the estimates.

The real time estimated velocity minus the smoothed velocity in figure 8.2 has a different trend than in figure 7.5. As the AUV travels along straight lines the error move away from zero. This phenomenon is not unexpected, as position updates are disabled. The error canceling effect when the AUV changed direction is still present, as the error moves towards zero at times of turning. The error in Beam 1 and Beam 2 are generally negative, which means that the filter overestimates the velocity 5.4. Similarly, Beam 3 and Beam 4 underestimates the velocity during these time periods.

The navigational error shown in figure 8.3 increases steeply right after the position measurements are disabled. The relatively small increases in navigational error is due to the canceling effect of changing direction. At its peak the navigational error is approximately 24 meters.

The filter consistency check for Method 1 reports slight increases in the mean and standard deviations compared to 7.1, except for the standard deviation in beam 3 and 4[1]. The characteristics of the fitted Gaussians in table 8.1 implies that the normalized square innovations are Gaussian, but the small standard deviations suggests also that for Method 1 the measurement noise is too high.

[1] To the precision of 4 decimals.

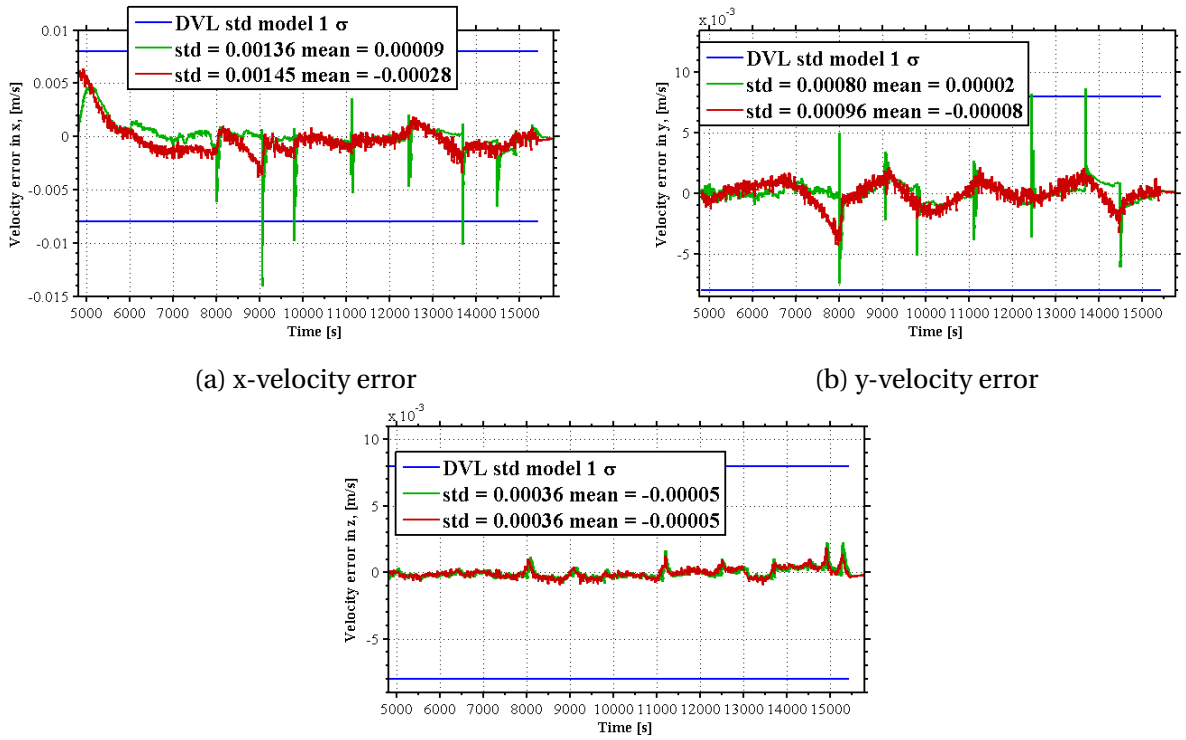
Chapter 9

Method 2: Excluding Position Updates

This chapter presents how the current DVL readings are implemented and which problems are present. Figures from the real-time test will be included to illustrate the problems.

9.1 Error in Real Time Estimated Velocity

The real error in real time estimated velocity is shown in figure 9.1.



(a) x-velocity error

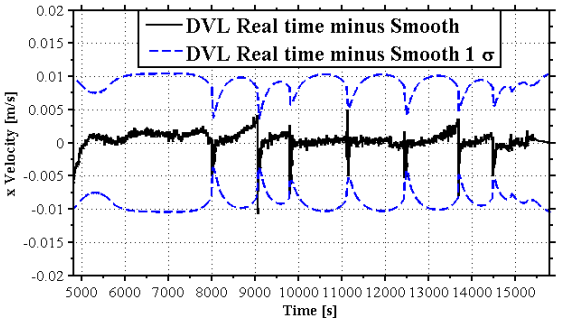
(b) y-velocity error

(c) z-velocity error

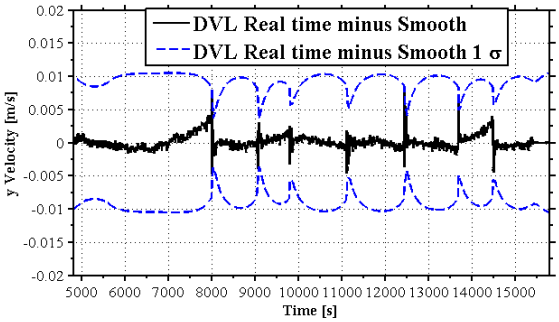
Figure 9.1: Real-time estimated and smooth velocity error

9.2 Real time Velocity minus Smooth Velocity

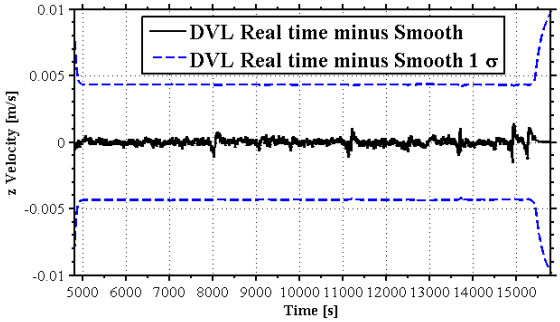
The real time velocity minus the smooth velocity is depicted in figure 9.2.



(a) Estimated real time velocity minus smooth velocity in x direction



(b) Estimated real time velocity minus smooth velocity in y direction



(c) Estimated real time velocity minus smooth velocity in z direction

Figure 9.2: Estimated real time velocity minus smooth velocity for Method 1 without position updates.

9.3 Estimated error in navigational position

The estimated navigational positioning for Method 2 without position updates is shown in figure 9.3

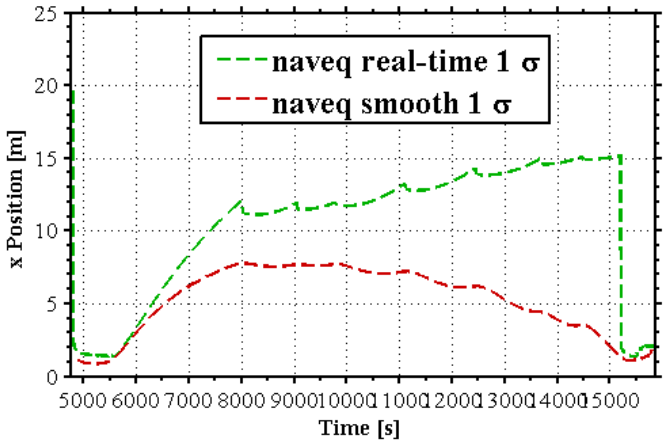
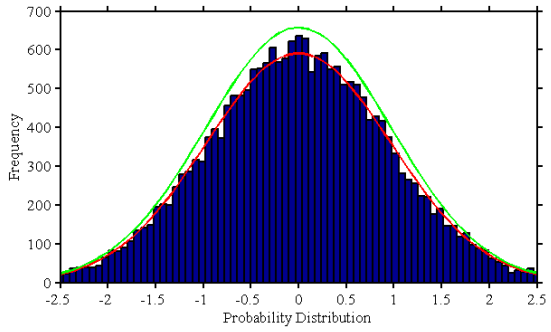


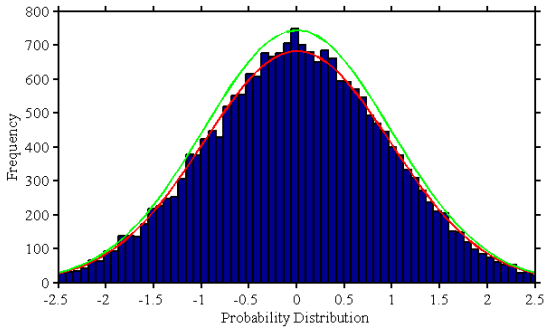
Figure 9.3: The standard deviation for the filters position estimate. The difference between the x and y position error is so small (in the 10^{-3} range), so only the x position error is listed.

9.4 Filter Consistency test

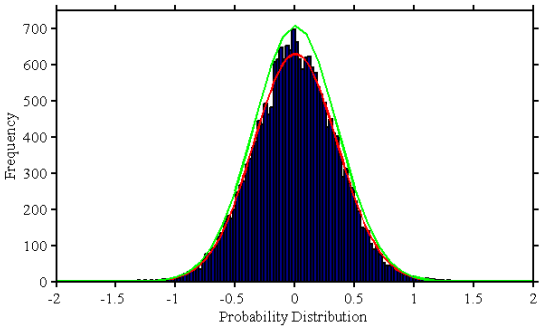
The squared innovation referenced in section 2.5.2 is performed on the data set.



(a) Direction x



(b) Direction y



(c) Direction x

Figure 9.4

The resulting values for the fitted Gaussians in table 9.1.

	$\hat{\mu}$	$\hat{\sigma}$
Innovation Stage X	-0.0023	0.9697
Innovation Stage Y	0.0023	0.9783
Innovation Stage Z	0.0056	0.3509

Table 9.1: Means ($\hat{\mu}$) and standard deviations ($\hat{\sigma}$) of the fitted Gaussian distribution (red line in figure 9.4

9.5 Performance of Method 2

The real time estimated velocity error is depicted in figure 9.1. Apart from the velocity error in the z-direction, the velocity errors are more erratic during vehicle turns than in 8.1. In addition to the larger fluctuation, the mean real-time velocity error for Method 2 is many times smaller than using Method 1. The smoothed estimate has a higher mean value in 9.1a and 9.1b on account of the large error spikes.

The difference between the real time velocity and the smoothed velocity is shown in figure 9.2. The error has higher extreme points than in 8.2, but remains closer to zero during straight trajectories.

The navigational error has the same tendencies as with 8.3, but the error increases slower during the absent of position updates. Reaching a maximum of 15 meters before the position updates are resumed.

The filter consistency check done in 9.4 shows that the innovations for stage Z and Stage Y are very near Gaussian. Indicated by the standard deviation in 9.1, the measurement noise added to the covariance is the right amount. Stage 3 has a low standard deviation.

Chapter 10

Conclusions, Discussion, and Recommendations for Further Work

10.1 Discussion

The work presented in this thesis focuses on specific challenges related to underwater navigation by the use of AUVs and how the Doppler Velocity Log connected to the AUV vessel perform for different integration methods. Analyses and simulations were carried out by the use of NavLab, which is a flexible and generic aided inertial navigation software application developed by Kongsberg Maritime. NavLab is implemented in Matlab, and is used for performing navigation calculations for navigational purposes. The pre- and post-processing of data measurements as well as analyses performed in this thesis are therefore performed by the use of Matlab and NavLab. Particular attention was given to the velocity measurements from the DVL device, as the velocity measurements bounds the velocity error of the navigation system, which in turn only lead to constant increase in position standard deviation.

The method currently used in Kongsberg Maritime is an cartesian integrated DVL. This method, referred to as Method 2, and converts the DVL measurements along x,y,z velocities prior to being inputted to the Kalman filter. However, by inputting the raw DVL measurements to the Kalman filter, navigation position error may increase or decrease compared to Method 2. By implementing such a method, it is possible to assess whether Method 2 can be improved or should be replaced with a new method for performing velocity estimations in a more sufficient manner. Such a method is proposed in this thesis, and is referred to as tightly integrated DVL, herein Method 1.

The investigated challenges were concretized into 3 main objectives:

1. Examine the feasibility of a tightly integrated DVL approach.

2. Compare the performance of the current DVL implementation with the tightly integrated

implementation.

3. Evaluate if a tightly integration requires further testing

Based on the reported performance of Method 1 presented in chapter 8.5this method is still in the development phase and requires further testing with both simulated data and real measurements in order to be able to exceed the method Kongsberg is currently using in terms of performance and accuracy.

Some of the key findings are discussed in the following section.

Measurement noise contribution to the covariance is not tuned, meaning the system deems estimates to be more accurate than what the estimates actually are. Under operation during position update blackouts, i.e. when the position update sensors are not receiving information, Method 2 is far superior to Method 1. This conclusion is based on the navigational error output from Method 2 being almost 10 meters less compared to Method 1. The Kalman filter is also sufficiently tuned for Method 2, meaning that the Kalman filter dynamics are adjusted for optimal performance and minimal navigational error output. It should be emphasized that the Kalman filter in Method 1 is not tuned, meaning that the results presented in chapter 8 are to some extent affected by sources of errors caused by surpassing the step where the Kalman dynamics are tuned.

The Kalman filter is tuned to operate with velocity measurements in x-, y-, and z-direction, as is seen in the standard deviation of the innovations in table 9.1. The standard deviation in x- and y-direction are very close to 1, i.e. filter innovations are approximately Gaussian. Therefore it is of interest to tune the measurement noise towards a standard deviation of 1 in Method 1, and reassess the situation.

The nonlinearities present in the process dynamics for the AUV model for the two methods can in certain situations be of significant magnitude. For such situations, a different Kalman implementation might be required to account for errors caused by higher order nonlinearities.

In the Extended Kalman filter (EKF), the state distribution is estimated using a Gaussian random variable, which is then propagated analytically through the first-order linearization of the nonlinear system. This approach can be the source of large errors in the true posteriori mean and covariance of the transformed Gaussian random variable. Sub-optimal performance and sometimes divergence in the filter may be the result of such errors. The Unscented Kalman Filter (UKF) approaches this issue by using a deterministic sampling approach: Here a minimal set of chosen sample points are used to represents the state distribution, which fully capture the true mean and covariance of the Gaussian random variable. When the variable is processed through the true nonlinear system, the posterior mean and covariance are accurately estimated to the third degree for any nonlinearity Wan and Rudoplh (2000). The UKF is more computationally heavy compared to the EKF, meaning that during times where the non-linearity are of a

low order, the UKF imposes unnecessary burdens on the system, reducing the efficiency of the computations. The nonlinearities are not of sufficient order to warrant the use of a UKF.

The result presented in chapter 7, 8, and 9 are based on recorded data from an AUV mission test, provided by Kongsberg Maritime. Real data from an actual mission allows for ambiguities into the system, such that the values of the estimated velocities, position bias, and error in velocity actually are real values, not values from numerical simulations.

The main focus in this thesis has been on testing the performance of the two methods on real sensor data. It is of further interest to perform simulations using simulated trajectories, velocity changes, position updates, and acceleration changes in order to perform qualitative studies between the two methods to strengthen the evaluation basis. By performing simulations based on simulated measurements it is possible to better assess the performance when applying Method 1 to more complex situations, such as including different trajectories and different output from the DVL and the other sensors, such as position, depth, vehicle orientation. By using simulated values as input for the estimated variables, the true values are known with 100% accuracy and thus a better understanding of the errors and the sources of errors can be deduced.

10.2 Conclusions

Accurate real time velocity approximations are important to ensure restricted position drift of underwater vehicles (UV). Inaccurate velocity measurements is a major source of errors which causes biases in the position estimates. A tightly integrated Doppler Velocity Log (DVL) solution was implemented and tested on real data, as an alternative to the already implemented DVL solution. The proposed implementation was proven to be a feasible method, as the Extended Kalman filter (EKF) was able to estimate velocities in transducer beams with an approximate mean errors of 0.2%. The navigational error did not decrease when using Method 1, relative to Method 2. However, the EKF was not tuned for Method 1, meaning that the internal Kalman filter dynamics for Method 1 presented in this thesis are not sufficiently accounted for.

The EKF was not tuned for Method 1, meaning that the Kalman filter performs incorrect weighting of the internal states in the filter, which is a source of error affecting the filter output. Based on this, further testing should be done to evaluate the applicability and reliability of Method 1.

Whether Method 1 is superior to Method 2 is inconclusive in this study. However, the important issue of Extended Kalman filter tuning in Method 1 has been illuminated, and the tightly integrated Doppler Velocity Log was tested on real data and successfully demonstrates velocity estimation in transducer beams. The work presented in this thesis forms a solid foundation for further research within the field of velocity and position estimation for AUVs in Kongsberg Maritime and continuous advancements of operations in the multitude of fields that benefit from

AUVs and DVL system navigation.

10.3 Recommendations for Further Work

As the Kalman filter dynamics for Method 1 presented in this thesis are not satisfactory accounted for , further testing should be done to evaluate the applicability, reliability, and robustness of Method 1. With the groundwork of this thesis in mind, the testing should include the use of a simulator and finer tuning of the EKF. A normalized innovation squared (NIS) test should be done to further evaluate the consistency of accurate estimates from the Kalman filter, as the test in section 2.5.2 did not account for the values off the diagonal.

Appendix A

Supplementary information

A.1 The Doppler effect

The Doppler effect is a change in the observed sound pitch that results from relative motion Indstruments (1996). An example to illustrate this is a honking car moving towards an observer. As the car approaches, the honk has a higher pitch, and a lower pitch when the car has passed. As the change in pitch is proportional to the velocity of the car, it is possible to calculate the velocity of the car. The standard equation for calculating the frequency shift is:

$$\Delta f = f_0 \cdot \frac{v}{c} \tag{A.1}$$

where Δf is the frequency shift caused my relative motion, f_0 is the frequency to the outgoing signal, v is the relative velocity from the signal source to an observer, and c is the speed of sound.

A.2 Gyrocompassing

Gyrocompassing is a method of finding orientation about the vertical axis of the Earth. The method is non-metalic as it uses gyros instead of the magnetic field of the Earth. Orientation is found by measuring the direction of Earth's axis of rotation relative to inertial space $\vec{\omega}_{IE}$.

As gyrocompassing does not rely on magnetism, the method has significant advantages over regular compasses: true (geographical) north is found as opposed to magnetic north, and the gyroscopic measurement is not affected by local magnetic fields, such created by the iron in a ships steel hull.

A.3 Differentiating of coordinate vectors

A coordinate vector is time differentiated by time differentiating the components of the vector

$$\dot{\mathbf{u}}^L = \frac{d}{dt}(\mathbf{u}^L) = \frac{d}{dt}\begin{bmatrix} \dot{u}_1^L \\ \dot{u}_2^L \end{bmatrix} \tag{A.2}$$

The time derivative in frame L in relation to frame B is found defined by the equation

$$\mathbf{k}^L = \mathbf{R}_B^L \mathbf{k}^B \tag{A.3}$$

when time differentiated becomes

$$\dot{\mathbf{k}}^L = \mathbf{R}_B^L \dot{\mathbf{k}}^B + \dot{\mathbf{R}}_B^L \mathbf{k}^B \tag{A.4}$$

Inserting $\dot{\mathbf{R}}_B^L = \mathbf{R}_B^L \mathbf{S}(\omega_{LB}^B \mathbf{k}^B)$ from (Egeland and Gravdahl, 2002) and replacing $\mathbf{k}$ with the position vector $\mathbf{p}$ yield the expression

$$\mathbf{v}^L = \mathbf{R}_B^L(\dot{\mathbf{v}}^B + \mathbf{S}(\omega_{LB}^B)\mathbf{p}^B) \tag{A.5}$$

where $\mathbf{p}$ is a coordinate distance vector and $\mathbf{v}$ is a coordinate velocity vector.

A.4 Calculation of angular velocity ω_{EL}^L if necessary

(Gade, 1997) shows that for the wandering azimuth L-system (AZI) to only rotate around its x- and y- coordinates in relation to the Earth, the vertical part of $\vec{\omega}_{EL}$ must always be equal to zero. If the Earth is modeled as a sphere, the angular velocity of L with respect to Earth, is

$$\omega_{EL}^L = \frac{1}{r_{EB}}(\mathbf{u}_{EB}^L \times \mathbf{v}_{EB}^L) = \frac{1}{r_{EB}}(\mathbf{S}(\mathbf{u}_{EB}^L)\mathbf{v}_{EB}^L) \tag{A.6}$$

In some navigation standards, namely WGS-84[1], the Earth is modeled as a ellipsoid. r_{EB} is the length of the vector spanning from the center of the Earth, to the center of the Body-system. In other words, r_{EB} is equal to the radius of the Earth, minus the depth of the vehicle.

$$r_{EB} = 6371km - z$$

where z is the depth of the AUV. Recall that depth is positive downwards. See appendix A.5 for clarification why the depth of the water is not included.

The vector $\mathbf{u}_{EB}^L$ is the unit vector from the origin of the body-system, to the origin of the

[1] World Geodetic System

earth-system. This vector is equivalent to the unit vector from the origin of the L-system, hence

$$\mathbf{u}_{EB}^{L} = \mathbf{u}_{EL}^{L} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^{T}$$

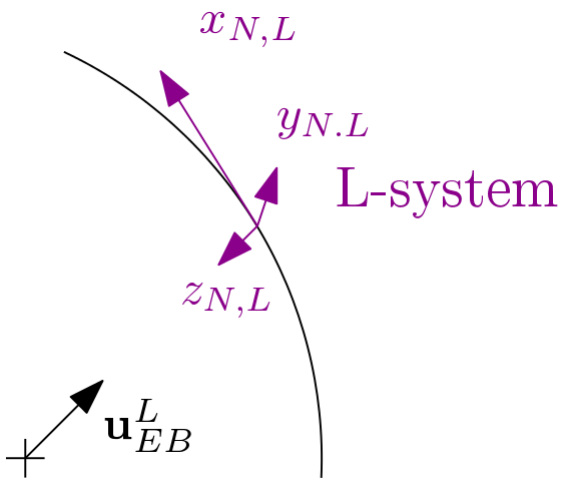


Figure A.1: The unit vector $\mathbf{u}_{EB}^{L}$ together with the orientation of the L-system.

A.5 Distance from center of Earth to AUV

The actual expression for the r_{EB} is

$$r_{EB} = r_{Earth} + d - z \tag{A.7}$$

where the values are as seen in figure A.2. However, the water depth d can be removed from the

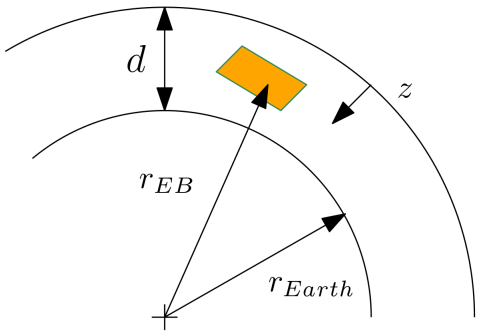


Figure A.2

calculation of z as it is so small when compared to the radius of the Earth. Consider if the tokt was done in Norways deepest fjord, Sognefjorden (with a depth of 1303 meters). The depth of the AUV is set to 150 meters under the water surface,

$$r_{EB} = \begin{cases} 6371 * 10^3 m + 1303 m - 150 m = 6372153 m & \text{if d included} \\ 6371 * 10^3 m - 150 m = 6370850 m & \text{without d} \end{cases} \tag{A.8}$$

The difference in r_{EB} is approximately 0.02%. Such a small error does not warrant the depth of the water to be included when calculating r_{EB}.

A.6 Normalized Innovation Squared (NIS) test

The Normalized Innovation Squared (NIS) test for Kalman filters are according to Houts, Sarah E et al. (2013) based on the assumption that the normalized innovations will during normal conditions (no sensor dropouts) have unit variance. The Kalman filter innovation is defined as the difference between the measurement and the estimated measurement. This is defined by the following expression.

$$\nu_k = \tilde{y}_k - \hat{y}_k \tag{A.9}$$

The covariance S_k is equal to

$$S_k = R_k + \mathbf{H}_k \hat{P}_k (\mathbf{H})_k^T \tag{A.10}$$

where $\mathbf{R}_k$ is the measurement covariance shown in 2.5a, $\mathbf{H}_k$ is the measurement vector calculated in 5.15, $\mathbf{P}_k$ is the predicted state covariance as seen in 2.9.

The normalized innovation is then defined for time step k as

$$NIS_k = \nu_k S_k^{-1} (\nu_k)^T \tag{A.11}$$

The null-hypothesis is defined as

$$H_0 : E[\nu_k] = 0 \tag{A.12}$$

The null-hypothesis assumes that samples observation are produced by randomly, such that the expected value of the innovations are zero. The null-hypothesis are tested against an alternative hypothesis

$$H_a : E[\nu_k] \neq 0 \tag{A.13}$$

The statistic test NIS_k follows an χ^2 distribution with the probability relationship

$$P\{NIS_k \leq \chi^2_{m,1-\alpha} | H_0\} = 1 - \alpha \tag{A.14}$$

where m is the number of measurements (degrees of freedom) α is the significance level. If

$NIS_k < \chi^2_{m,1-\alpha}$ then equation A.12 is accepted. Gamse et al. (2014) concluded that if equation A.12, then there are no significant discrepancy between a system prediction and measurement. Subsequently, if the statement expressed in A.12 is rejected, the innovation discrepancy may be rooted in the measurements, the model estimate, or in the system model used in the Kalman filter.

Appendix B

Verifications

B.1 The measurement matrix H_k

The measurement matrix H_k can be verified by controlling the expression

$$h(\hat{\mathbf{R}}_B^L, \hat{\mathbf{v}}_{EB}^L) - h(\mathbf{R}_B^L, \mathbf{v}_{EB}^L) \approx H\delta\mathbf{x} \tag{B.1}$$

The procedure for doing so is

1. Choose reasonable[1] $\mathbf{R}_B^L$, $\mathbf{v}_{EB}^L$, $\delta\mathbf{v}_{EB}^L$, and $\mathbf{e}_{LB}^L$
2. Calculate $\hat{\mathbf{v}}_{EB}^L = \mathbf{v}_{EB}^L + \delta\mathbf{v}_{EB}^L$
3. Calculate $\hat{\mathbf{R}}_B^L = \mathbf{S}(\mathbf{e}_{LB}^L) \cdot \mathbf{R}_B^L$
4. Find $\delta\mathbf{x} = [\mathbf{e}_{LB}^L; \delta\mathbf{v}_{EB}^L]$
5. Controll equation B.1

Verification with plausible parameters

Say that

$$\mathbf{R}_B^L = \begin{bmatrix} cos45 & -sin45 & 0 \\ sin45 & cos45 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{v}_{EB}^L = \begin{bmatrix} 10 \\ 4 \\ 2 \end{bmatrix}$$

[1] $\mathbf{e}_{LB}^L$ and $\delta\mathbf{v}_{EB}^L$ should be in the $10^{-5} range$

$$\delta \mathbf{v}_{EB}^{L} = 10^{*-5} \begin{bmatrix} 5 \\ 0.5 \\ 0.15 \end{bmatrix}$$

$$\mathbf{e}_{LB}^{L} = 10^{*-5} \begin{bmatrix} 8.256 \\ 2 \\ 56 \end{bmatrix}$$

The estimated velocity $\hat{\mathbf{v}}_{EB}^{L}$ is then

$$\hat{\mathbf{v}}_{EB}^{L} = \mathbf{v}_{EB}^{L} + \delta \mathbf{v}_{EB}^{L} = \begin{bmatrix} 10.00005 \\ 4.000005 \\ 2.0000015 \end{bmatrix} \tag{B.3}$$

The estimated rotation matrix $\hat{\mathbf{R}}_{B}^{L}$ is

$$\hat{\mathbf{R}}_{B}^{L} = \mathbf{R}_{B}^{L} + \mathbf{S}(\mathbf{e}_{LB}^{L})\mathbf{R}_{B}^{L} = \begin{bmatrix} 0.7067 & -0.7075 & 2.0000 * 10^{-5} \\ 0.7075 & 0.7067 & -8.2560 * 10^{-5} \\ 4.4237 * 10^{-5} & 7.2521 * 10^{-5} & 1 \end{bmatrix} \tag{B.4}$$

δx is

$$\delta \mathbf{x} = \begin{bmatrix} (\mathbf{e}_{LB}^{L})^{T} \\ \delta \mathbf{v}_{EB}^{L\ T} \end{bmatrix} \tag{B.5}$$

Insertion of the aforementioned values into B.1 yields that both the right hand side and the left hand side is approximately equal to 0.0024. The measurements matrix H_k is verified.

Bibliography

Attaway, S. (2013). *A Practical Introduction to Programming and Problem Solving.*

Brown, R. G. and Hwang, P. Y. C. (2012). *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB exercises.* John Wiley & Sons, Inc.

Chen, C.-T. (2013). *Linear System Theory and Design.* Oxford University Press.

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

Gade (1997). INTEGRERING AV TREGHETSNAVIGASJON I EN AUTONOM UNDERVANNSFARKOST. *Forsvarets Forskningsinstitutt.*

Gade, K. (2005). NavLab, a generic simulation and post-processing tool for navigation. *Modeling, Identification and Control*, 26(3):135–150.

Gamse, S., Nobakht-Ersi, F., and Sharifi, M. (2014). Statistical Process Control of a Kalman Filter Model. *Sensors*, 14(10):18053–18074.

Hegrenæs, Ø. and Berglund, E. (2009). Doppler Water-Track Aided Inertial Navigation for Autonomous Underwater Vehicle. *IEEE*, 25:223–236.

Hegrenæs, Ø., Gade, K., Hagen, O. K., and Hagen, P. E. (2009). Underwater Transponder Positioning and Navigation of Autonomous Underwater Vehicles. *OCEANS 2009.*

Houts, Sarah E, Dektor, Shandor G, and Rock Stephen, M (2013). A Robust Framework for Failure Detection and Recovery for Terrain-Relative Navigation. Unmanned Untethered Submersible Technology 2013.

Indstruments, R. D. (1996). Principles of Operation: A Practical Primer.

Jalving, B., Gade, K., Svartveit, K., Willumsen, A., and Sorhagen, R. (2004). DVL Velocity Aiding in the HUGIN 1000 Integrated Inertial Navigation System. *Modeling, Identification and Control*, 25:223–236.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35.

Kongsberg Maritime (2017). Autonomous Underwater Vehicle: The HUGIN Family. page 11.

McGee Leonard, A. and Schmidt Stanley, F. (1985). Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry.

NORTEK (2018). DVL 1MHz/500kHz.

Rudolph, D. and Wilson, T. A. (2012). Doppler Velocity Log Theory and Preliminary Considerations for Design and Construction. *IEEE Southeastcon*.

Vickery, K. (1998). ACOUSTIC POSITIONING SYSTEMS "A PRACTICAL OVERVIEW OF CURRENT SYSTEMS". Dynamic Positioning Cinference.

Vik, B. (2014). Integrated Satellite and Inertial Navigation Systems.

Wan, E. A. and Rudoplh, v. d. M. (2000). The Unscented Kalman Filter for Nonlinear Estimation. *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*.

Willumsen, A. B. and Hegrenaes, O. (2009). The joys of smoothing. In *OCEANS 2009-EUROPE*, pages 1–7. IEEE.