



Norwegian University of  
Science and Technology

# Integrating GNSS/INS/PVT for robust positioning

**Håkon Leithe**

**Espen Lie**

Master of Science in Cybernetics and Robotics

Submission date: June 2016

Supervisor: Kjetil Bergh Ånonsen, ITK

Co-supervisor: Arild Søråunet, Norbit Subsea AS

Norwegian University of Science and Technology  
Department of Engineering Cybernetics



---

# Abstract

**The thesis objective** was to get a better insight in theories behind global navigation satellite systems, inertial navigation systems and the two integrated. Current technology used to realize such system was also investigated.

**The development** process started by specifying the requirements, designing and implementing a solution.

**The project result** in a Navigation Module for INS with GNSS multi-antenna system and a Kalman filter for estimating position, velocity, attitude and IMU biases.

**Some of the findings** were that a loosely coupled Kalman filter has great advantages compared to a uncoupled filter and is good at estimating filter states through short GNSS outages.

**The conclusion** of the results found in the experiment is that designing a Kalman filter that performs well compared to established systems is achievable. The thesis lead to a hardware and software module that performed as intended.

**For future work** an expansion of the filter can be done. Also, other coupling schemes can be investigated further and implemented.

---

# Preface

This thesis is a continuation of a project with the same title and authors from December 2015. Parts of this thesis are directly copied from the project. It examines possible solutions in inertial navigation systems (INS), global navigation satellite systems (GNSS) and the integration of the two parts.

Norbit Subsea is aiming for a market where GNSS/INS integrated systems are used together with wideband multibeam sonars for seafloor mapping.

The project has been a collaboration between the students Håkon Leithe and Espen Lie. It has been performed in partnership with Norbit Subsea. This report gives a brief overview of background theory used in GNSS, INS and integrated systems. It also contains an experiment where hardware for an integrated system is assembled. GNSS and INS data is gathered and later analyzed and fused in a loosely coupled Kalman Filter. All hardware used during experiments were funded by Norbit Subsea AS and temporary authorization codes for R&D purposes were provided by NovAtel Europe Ltd.

We would like to thank our supervisors Kjetil Bergh Ånonsen from Norwegian University of Science and Technology and Arild Sjøraunet at Norbit Subsea AS for guidance throughout the thesis.

Trondheim, Norway, 2016-06-06

Håkon Leithe

Espen Lie

# Table of Contents

<b>Summary</b>	<b>i</b>
<b>Preface</b>	<b>ii</b>
<b>Table of Contents</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>x</b>
<b>Abbreviations</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Current solution . . . . .	1
1.1.1 External INS . . . . .	1
1.1.2 Applanix integrated . . . . .	2
1.1.3 NovAtel integrated . . . . .	3
1.2 New markets . . . . .	3
1.2.1 Tightly integrated low cost navigation system . . . . .	3
1.3 Project Goals . . . . .	4
1.3.1 Global navigation satellite system . . . . .	4
1.3.2 Inertial navigation system . . . . .	4
1.3.3 Integrating GNSS and INS . . . . .	4
1.3.4 Experiment . . . . .	5
1.3.5 Future work . . . . .	5
<b>2 Global Navigation Satellite Systems</b>	<b>7</b>
2.1 Basic GNSS knowledge . . . . .	8
2.1.1 Estimating position and time . . . . .	8
2.2 GNSS error sources . . . . .	8
2.3 Improving the position accuracy and precision . . . . .	9

---

2.3.1	Estimating the ionospheric delay using a dual-frequency receiver .	9
2.3.2	Differential GNSS . . . . .	9
2.3.3	Real-time kinematic . . . . .	10
2.4	GNSS Attitude determination . . . . .	12
2.5	Current market . . . . .	12
2.5.1	ublox LEA-M8T . . . . .	12
2.5.2	NovAtel OEM617D . . . . .	13
<b>3</b>	<b>Inertial Navigation Systems</b>	<b>15</b>
3.1	Introduction . . . . .	15
3.2	Rotations and strapdown INS equations . . . . .	16
3.2.1	Coordinate frames and transformation . . . . .	16
3.2.2	Navigation equations . . . . .	18
3.3	INS errors . . . . .	20
3.3.1	Error sources . . . . .	20
3.3.2	Bias stability . . . . .	20
3.4	INS Error Dynamics . . . . .	21
3.5	INS Technologies . . . . .	22
3.5.1	Gyroscopes . . . . .	22
3.5.2	Accelerometers . . . . .	23
3.6	Current market . . . . .	24
3.6.1	Sparkfun Razor IMU . . . . .	24
3.6.2	Sensoror STIM300 . . . . .	24
3.6.3	Honeywell HG9900 IMU . . . . .	25
<b>4</b>	<b>Integration of GNSS and INS</b>	<b>27</b>
4.1	Integrated systems . . . . .	27
4.1.1	Uncoupled integration . . . . .	28
4.1.2	Loosely coupled integration . . . . .	28
4.1.3	Tightly coupled integration . . . . .	29
4.1.4	Ultra-Tight integration . . . . .	29
4.1.5	Hardware requirements . . . . .	30
4.2	Kalman Filtering . . . . .	30
4.2.1	Linearized Kalman filter . . . . .	30
4.2.2	Extended Kalman filter . . . . .	31
4.2.3	INS Error Models . . . . .	32
4.2.4	Sensor errors . . . . .	32
4.2.5	Filter design . . . . .	33
4.3	Heave . . . . .	34
4.4	Open-source software . . . . .	34
4.4.1	RTKlib . . . . .	34
4.4.2	INSTK . . . . .	34
4.4.3	Mapping . . . . .	35
4.5	Current market . . . . .	35
4.5.1	Ublox NEO-M8L . . . . .	35
4.5.2	NovAtel . . . . .	35

---

---

4.5.3	Applanix Wavemaster II . . . . .	36
<b>5</b>	<b>Development</b>	<b>37</b>
5.1	Design notes . . . . .	37
5.1.1	Interfaces . . . . .	38
5.1.2	NovAtel OEM617D . . . . .	38
5.1.3	Real-time processor . . . . .	40
5.1.4	STIM300 . . . . .	40
5.1.5	Odroid XU4 . . . . .	41
5.2	Implementation . . . . .	41
5.2.1	Real-time firmware . . . . .	41
5.2.2	Hardware . . . . .	44
5.3	Long term static test of IMU . . . . .	46
5.4	Long term static test of GNSS . . . . .	50
5.5	Data integrity . . . . .	54
5.5.1	Cyclic Redundancy Check . . . . .	54
5.5.2	Software filters . . . . .	55
5.6	Conclusion . . . . .	56
5.6.1	Data integrity . . . . .	56
5.6.2	Timetagging of IMU data . . . . .	56
5.6.3	Sensor analysis . . . . .	56
<b>6</b>	<b>Experiment</b>	<b>57</b>
6.1	Test setup . . . . .	57
6.1.1	Navigation Module . . . . .	57
6.1.2	Alignment of the IMU . . . . .	59
6.1.3	Offsets . . . . .	60
6.1.4	Translation between IMU and GNSS . . . . .	60
6.2	Determining variances . . . . .	61
6.2.1	Determining process noise . . . . .	61
6.2.2	Initial process variance . . . . .	62
6.2.3	Measurement noise matrix . . . . .	63
6.2.4	Measurement mapping matrix . . . . .	63
6.3	Kalman filter implementation . . . . .	64
6.3.1	State vector . . . . .	64
6.3.2	Initialization . . . . .	65
6.3.3	Prediction . . . . .	65
6.3.4	Correction . . . . .	65
6.3.5	Error dynamics . . . . .	66
6.4	Data gathering . . . . .	68
6.5	Uncoupled Kalman filter . . . . .	70
6.5.1	Results . . . . .	70
6.5.2	Discussion . . . . .	70
6.6	Loosely coupled Kalman filter . . . . .	72
6.6.1	Tuning . . . . .	72
6.6.2	Results . . . . .	72

---

---

6.6.3	Discussion . . . . .	73
6.7	Head-to-head performance analysis . . . . .	75
6.7.1	Applanix Wavemaster II . . . . .	75
6.7.2	Hardware setup . . . . .	75
6.7.3	Results . . . . .	76
6.7.4	GNSS outage . . . . .	79
6.7.5	Discussion . . . . .	80
<b>7</b>	<b>Discussion</b>	<b>83</b>
7.1	Findings in theory . . . . .	83
7.2	Findings in experiment . . . . .	83
<b>8</b>	<b>Future work</b>	<b>85</b>
8.1	Hardware integration . . . . .	85
8.2	Expanding the error model . . . . .	85
8.3	Tight coupling . . . . .	85
<b>9</b>	<b>Conclusion</b>	<b>87</b>
9.1	Theory . . . . .	87
9.2	Experiment . . . . .	87
	<b>Bibliography</b>	<b>87</b>



# List of Tables

2.1	Overview of some GNSS constellation parameters. . . . .	8
2.2	Dual-frequency RTK convergence numbers [33] . . . . .	10
2.3	Single-frequency RTK convergence numbers [33] . . . . .	11
3.1	Coordinate system variables index explanation [33] . . . . .	16
3.2	WGS-84 parameters [33]. . . . .	16
3.3	Velocity and angular velocity vectors [33]. . . . .	17
3.4	Position and attitude vectors [33] . . . . .	17
4.1	Discrete extended Kalman filter [33]. . . . .	31
5.1	NovAtel OEM617D performance key figures. . . . .	39
5.2	Sensoror STIM300 performance key figures[26]. . . . .	40
5.3	LED indicator behavior. . . . .	43
5.4	Spectral density of gyroscopes, reading from Figure 5.9 . . . . .	48
5.5	Spectral density of accelerometers, reading from Figure 5.10 . . . . .	49
5.6	Standard deviation of position, velocity and attitude during long-term static test. . . . .	54
6.1	Graph colors overview . . . . .	69
6.2	Applanix Wavemaster II performance key figures. . . . .	75

---

# List of Figures

1.1	Sonar interface unit . . . . .	1
1.2	Wideband Multibeam Sonar without integrated navigation . . . . .	2
1.3	Sonar interface unit with navigation . . . . .	2
1.4	Wideband Multibeam Sonar with IMU mounted in the same bracket . . . . .	2
1.5	Sonar interface unit with navigation . . . . .	3
1.6	Wideband Multibeam Sonar with IMU integrated in the sonar . . . . .	3
2.1	EGNOS RIMS Sites[3] . . . . .	10
2.2	Mid-Norway GNSS base station overview [2]. . . . .	11
2.3	Attitude determination using GNSS . . . . .	12
2.4	u-blox LEA-M8T . . . . .	13
2.5	NovAtel 617D . . . . .	14
3.1	The Sagnac effect . . . . .	22
3.2	Sparkfun Razor IMU . . . . .	24
3.3	Sensoror STIM300 . . . . .	24
3.4	Honeywell HG9900 IMU . . . . .	25
4.1	Uncoupled coupled integration . . . . .	28
4.2	Loosely coupled integration . . . . .	28
4.3	Tightly coupled integration . . . . .	29
4.4	Ultra-Tight integration . . . . .	29
4.5	Ublox NEO-M8L . . . . .	35
4.6	NovAtel integrated solution . . . . .	36
4.7	Applanix wavemaster II . . . . .	36
5.1	Block diagram of components and data flow in the setup . . . . .	37
5.2	The TOV(yellow) and TX(green) outputs @ 125Hz . . . . .	38
5.3	Flowchart of timetag software. . . . .	42
5.4	PPS (yellow) and GPZDA (green) signals. . . . .	43
5.5	The inside of the Navigation Module . . . . .	44

---

5.6	The outside of the Navigation Module. . . . .	45
5.7	Gyro averaging values of $\tau = 1\text{min}$ . . . . .	46
5.8	Accelerometer averaging values of $\tau = 1\text{min}$ . . . . .	47
5.9	Root Allan variance of static gyro measurements . . . . .	48
5.10	Root Allan variance of static accelerometer measurements . . . . .	49
5.11	Setup for long term GNSS static test . . . . .	50
5.12	Heatmap from long term static test GNSS . . . . .	50
5.13	Histogram of position in long-term static test GNSS . . . . .	51
5.14	Histogram of velocity in long term static test GNSS . . . . .	52
5.15	Histogram of attitude in long term static test GNSS . . . . .	53
6.1	Data-flow between components. . . . .	58
6.2	The rotation between IMU and BODY-frame . . . . .	59
6.3	The transformation between GNSS and BODY-frame . . . . .	60
6.4	State machine of a discrete extended Kalman filter. . . . .	64
6.5	Overview of route up the river Nidelva . . . . .	69
6.6	Uncoupled filter velocity . . . . .	70
6.7	Uncoupled filter attitude . . . . .	71
6.8	Unstable roll estimate . . . . .	72
6.9	Loosely coupled filter attitude . . . . .	73
6.10	Loosely coupled filter velocity . . . . .	74
6.11	Estimated biases in the tuned filter . . . . .	74
6.12	To the left, both system are mounted on same rigid frame. To the right a DC block to ensure only one system powers the antennas. . . . .	76
6.13	Velocity compared . . . . .	77
6.14	Attitude Compared. . . . .	78
6.15	Gamle bybro in Nidelva. . . . .	79
6.16	Position during GNSS outage. . . . .	80
6.17	Velocity during GNSS outage. . . . .	81
6.18	Position plotted in map. Black is Applanix and yellow is Kalman filtered solution . . . . .	82
8.1	Suggested setup for future work . . . . .	86

---

# Abbreviations

ADEV	=	Allan deviance
AVAR	=	Allan variance
CLI	=	Command-line user interface
CP	=	Carrier phase
CPOS	=	Centimeter Position
DGNSS	=	Differential GNSS
ECEF	=	Earth-Centered, Earth-Fixed
ECI	=	Earth Centered Inertial
EKF	=	Extended Kalman filter
EGNOS	=	European Geostationary Navigation Overlay Service
FOG	=	Fiber optical gyroscope
GLONASS	=	GLObal NAVigation Satellite System
GNSS	=	Global navigation satellite system
GPS	=	Global positioning system
GUI	=	Graphical user interface
IMU	=	Inertial measurement unit
INS	=	Inertial navigation system
MSAS	=	MTSAT Satellite-based Augmentation System
MTSAT	=	Multi-functional Transport Satellite
NTRIP	=	Networked Transport of RTCM vi Internet Protocol
PPS	=	Pulse per second
PR	=	Pseudoranges
PVT	=	Position, Velocity and Time
RIMS	=	Range and Integrity Monitoring Station
RINEX	=	The Receiver Independent Exchange Format
SSH	=	Secure SHell
TOV	=	Time of validity
TX	=	Transmitter
USART	=	Universal Synchronous/Asynchronous Receiver/Transmitter
VDC	=	Volts of direct current
WAAS	=	Wide Area Augmentation System
WADGNSS	=	Wide Area Differential GNSS
WBMS	=	WideBand Multibeam Sonar
WGS	=	World Geodetic System

---

# Chapter 1

## Introduction

This Chapter describes the current navigation solutions that Norbit Subsea are using today, and the new markets they wish to explore. The project is performed in cooperation with Norbit Subsea AS. Norbit Subsea designs, develops, manufactures and markets wideband multibeam sonars(WBMS) for hydrographic applications, forward-looking applications as well as advanced subsea leakage detection. One of the main products is an integrated system with a wideband multibeam sonar and a GNSS/INS system.

Most of Norbit Subseas customers uses the system for some sort of seafloor mapping. When dealing with applications off this kind, a highly precise and reliable navigation solution with good accuracy is needed. Later in this Chapter, we also present goals for the project.

### 1.1 Current solution

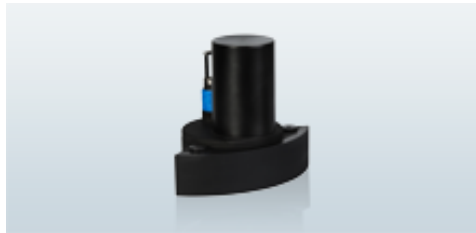
These are the solutions that Norbit Subsea is currently using with the WBMS. To be competitive in the market an accuracy of  $0.05^\circ$  in attitude, and 10 cm in position is required.

#### 1.1.1 External INS

The Norbit sonar can be used with an external INS. The Sonar then needs timing input from the INS, to timestamp its detection points. The interface is \$GPZDA(RS232)[4] + PPS.



**Figure 1.1:** Sonar interface unit



**Figure 1.2:** Wideband Multibeam Sonar without integrated navigation

### 1.1.2 Applanix integrated

The Applanix system uses a dual antenna GNSS receiver and a MEMS based IMU. The IMU is mounted in a separate housing, on the same bracket as the sonar. The sonar interface unit is equipped with Trimble AP20 as the GNSS-Inertial system.



**Figure 1.3:** Sonar interface unit with navigation



**Figure 1.4:** Wideband Multibeam Sonar with IMU mounted in the same bracket

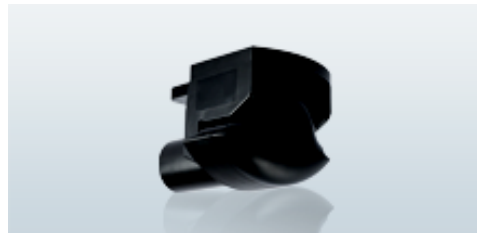


### 1.1.3 NovAtel integrated

By integrating a MEMS IMU, STIM-300, in the Sonar head the need of an extra component is removed. The GNSS receivers, NovAtel 615 and NovAtel 628, are mounted in the Sonar interface unit. The weight and complexity of this system is lower than with the Applanix system.



**Figure 1.5:** Sonar interface unit with navigation



**Figure 1.6:** Wideband Multibeam Sonar with IMU integrated in the sonar

## 1.2 New markets

### 1.2.1 Tightly integrated low cost navigation system

For the future Norbit wants to be able to integrate the navigation solution, used with their sonar kits, tighter with their own software and hardware. Using third-party systems as NovAtel/Sensoror and Applanix/IMU-42 with their integrating software gives constraints in integrating with self-made software and hardware as one needs to rely on third-party systems and their setup.

With a self integrated navigation system, the sale price could be lowered and user experience improved. This could make Norbit Subsea more competitive in rough markets.

## 1.3 Project Goals

In this project we will look into current navigation solutions using GNSS/INS and the theory behind such systems. We will take a look at the two systems separately at first, followed by an integration part.

### 1.3.1 Global navigation satellite system

To get user-position on Earth, a GNSS has the main role. There are several satellite constellations such as:

- Global positioning system (GPS)
- Global navigation satellite system (GLONASS)
- Galileo
- BeiDou

We will look into the basics of a GNSS; signals, frequencies and how user-position is calculated. Also how the precision and accuracy of position can be improved is of interest. Specially using different techniques like differential GNSS (DGNSS), real-time kinematics (RTK), GNSS attitude determination among others. Current receiver solutions from a couple of manufacturers (Applanix, NovAtel, uBlox) will be discussed.

### 1.3.2 Inertial navigation system

An inertial navigation system (INS) is important in integrated systems due to their fast dynamics and high update rates. In the project we will take a further look into basic theories behind inertial navigation calculation with especially the strapdown navigation equations, solutions currently on market (e.g. Sensoror, Honeywell) and sensors used for measurements:

- Ring laser gyroscope
- Fiber optic gyroscope
- MEMS
- Accelerometers

### 1.3.3 Integrating GNSS and INS

There are several types of integration possibilities of GNSS and INS. We want to get a clear overview of today's technology used, as well as the theory behind it. Among other, we will look into:

- Different types of Kalman filtering (loosely coupled vs tightly coupled)
- Open-source software solutions
- Heave measurement/compensation

### 1.3.4 Experiment

During this project we will build an integrated system (hereby called Navigation Module) consisting of an inertial measurement unit (IMU) and a GNSS receiver. The Navigation Module will be made universal so it can be reused for future experiment. When building a hardware module, software for interfacing with the components also needs to be developed. We will write software modules for interfacing with the hardware, for logging data and for analyzing gathered data. In shorthand chronological order we will:

- Develop a hardware module
- Write software for interfacing and logging
- Gather data in the field
- Implement Kalman filter

### 1.3.5 Future work

In the future the error model can be expanded to include more states. A more realistic error model gives a more correct estimate. It can be looked further into and possibly tried out methods for improving the integrated systems precision and accuracy such as:

- Tighter hardware integration
- Further expansion of the error model
- Tightly coupled filter

Verifying precision and quantifying accuracy will also be emphasized.



# Global Navigation Satellite Systems

Global Navigation Satellite Systems(GNSS) provides geo-spatial positioning with global coverage [16]. Global Positioning System (GPS), GLObal NAVigation Satellite System (GLONASS), BeiDou and Galileo are all satellite constellation with operational satellites. They operate in the same frequency range, and receivers often supports all the different systems. They still have some different characteristics and parameters.

GPS is an American system operated by the U.S. Departement of Defense [16]. It was the first system opened for civil applications, when it was opened for civilians in the 1980s. It has been a gateway for other constellations.

The Russian system GLONASS is aslo a fully operational GNSS with a wide range of satellites in operation. It is operated by the Russian Aerospace Defense forces.

In 2003 the first stage of the Galileo program was agreed upon by the European Union and European Space Agency [9]. The system is not fully operational yet. As of 2012, full completion of the Galileo system (24 operational and 6 active spares) is expected by 2020[17].

BeiDou is the chinese GNSS. It consists of two separate satellite constellations - a limited test system, and a full-scale global navigation system that is currently under construction [8].

See Table 2.1 for compared parameters of the constellations.

More information about GNSS can be found in [16] and [18].

Constellation	Satellites in orbit	Orbital period	Inclination	Orbital height
GPS	31	1h 58min	55°	20,200km
GLONASS	28	1h 15min	64,8°	19,100km
Galileo	8	14h 22min	56°	23,222km
BeiDou	20	12h 50min	55°	21,500km

**Table 2.1:** Overview of some GNSS constellation parameters.

## 2.1 Basic GNSS knowledge

### 2.1.1 Estimating position and time

A GNSS receiver measures the pseudoranges (PR) from a GNSS satellite to the receiver. Equation 2.1 shows how PR for a satellite is calculated. The positions dimension is  $[x_u, y_u, z_u]$  and travel time is,  $t_u$ . To estimate position and time at least 4 visible satellites are needed [33].

$$P_i = \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2 + (z_u - z_i)^2} + ct_u \quad (2.1)$$

Position and time is usually estimated using Least Squares method or Kalman Filter of different kinds.

## 2.2 GNSS error sources

GNSS precision and accuracy is very sensitive for timing errors and delays. Some errors can be compensated for and others not. Below are some error sources listed [18].

### 1. Satellite

- *Satellite Clock*: Error in the satellites internal clock
- *Ephermis data*: The satellite drifts out of orbital route

### 2. Atmosphere

- *Tropospheric Delay*: 0-50 km above sea-level
- *Ionospheric Delay*: 50-200 km above sea-level

### 3. Detection errors

- *Receiver clock*: Receiver clock drift
- *Multipath*: Signal bounces via nearby structures
- *Signal masking*: Signal is blocked by nearby structures
- *Jamming/Spoofing*: Signal is jammed/spoofed by nearby transmitter

## 2.3 Improving the position accuracy and precision

The position accuracy can be improved in many ways. Some of the most efficient methods for seafloor mapping are mentioned in this Chapter.

### 2.3.1 Estimating the ionospheric delay using a dual-frequency receiver

When a dual-frequency receiver is outputting raw data, both L1 and L2 pseudorange are available. The difference in the pseudorange can be used to calculate the ionospheric delay. [33]

$$\Delta^{iono}(f_{L1}) = \frac{(r_{L1} - r_{L2})f_{L1}^2}{1 - \left(\frac{f_{L1}}{f_{L2}}\right)^2} \quad (2.2)$$

where

$\Delta^{iono}(f_{L1})$  = Pseudorange error for L1(m)

$r_{L1}$  = Measured range on L1 frequency(m)

$r_{L2}$  = Measured range on L2 frequency(m)

$f_{L1}$  = L1 frequency = 1572.42 MHz

$f_{L2}$  = L2 frequency = 1227.6 MHz

The corrected pseudorange can be found by subtracting the ionospheric delay.

$$r_{L1,L2} = r_{L1} - \Delta^{iono}(f_{L1}) \quad (2.3)$$

This method is one of many methods to remove ionospheric delay [16].

### 2.3.2 Differential GNSS

#### DGNSS

By receiving Pseudorange (PR) corrections from a local reference station, the receiver is able to remove common errors.

#### WADGNSS

The development of Wide Area DGNSS (WADGNSS) systems, such as Wide Area Augmentation system (WAAS) in US, European Geostationary Navigation Overlay Service (EGNOS) in EU and MTSAT satellite Augmentation System (MSAS) in Japan, are replacing the need of a local reference station. By gathering data from multiple Ranging Integrity Monitoring Systems (RIMS) on the ground, the atmosphere over entire continents can be modelled. In fact, 10 cm level accuracies have been reported for up to 750 km baselines [33].

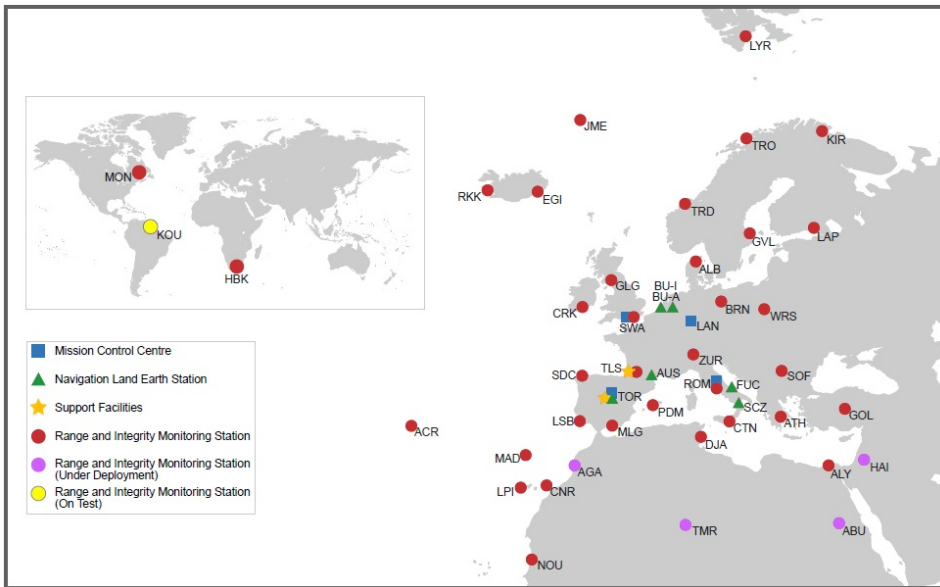


Figure 2.1: EGNOS RIMS Sites[3]

### 2.3.3 Real-time kinematic

Real-time kinematic (RTK) receivers utilize the carrier differential measurements in addition to the PR-corrections in DGNSS to achieve position accuracy in the order of a few centimeters. To achieve the best possible result, the distance to reference receiver should be as low as possible. With a baseline  $>1$  km, it is crucial to have a dual-frequency receiver. The expected performance using a single reference station can be seen in Table 2.2 and 2.3. The same technique with multiple sites can be applied for RTK as it is done with DGNSS.

Baseline	Search time with at least 6 satellites	Horizontal accuracy
< 10 km	120 sec + 1.5sec/km	2 cm + 0.5 ppm
< 15 km	8 min	8 cm
< 25 km	14 min	10 cm
< 35 km	20 min	40 cm

Table 2.2: Dual-frequency RTK convergence numbers [33]

Providers of RTK data can be the national authorities of the country. In Norway it is the national mapping service Kartverket that provides RTK among a variety of other positioning services. Kartverket offers correction services at a different degree of precision. The CPOS service provides measurement accuracy in a centimeter level through Network-RTK. Network-RTK differs from traditional RTK because it utilizes one or more Continuous Operation Reference Stations (CORS) to provide the RTK data. In a traditional RTK



setup one usually need to find a suitable place for having a GNSS base, which sometimes can be challenging. Also, there is usually need for UHF radio communication between the station and the rover. The main drawback of using Network-RTK in advance of the traditional RTK is the dependence of being inside of CORS coverage areas. The CPOS service consists of GNSS correction data received in real-time via Networked Transport of RTCM via Internet Protocol (NTRIP) client through an internet connection. CPOS supports RTCM versions 2.3 and 3.1 which is a widely used protocol. The system uses multiple CORS to calculate a virtual reference station (VRS) for the user [1], and the main purpose to do this is to reduce the baseline distance between the rover and the reference station in order to efficiently remove spatially correlated errors using differential processing. The user can not distinguish between VRS and a real reference station.

CPOS is aimed for professional users and has a user base consisting of 3000 users. For applications that do not require centimeter level accuracy, Kartverkets service DPOS provides accuracy at desimeter level through DGNSS.

Currently, Kartverkets GNSS network consists of more than 170 GNSS reference stations. They are located both on the mainland and on remote islands, such as Jan Mayen and Svalbard. Full coverage for real-time services through station network in the entire country was reached in 2011. In Figure 2.2 an overview of the GNSS base stations situated around Trondheim is shown.

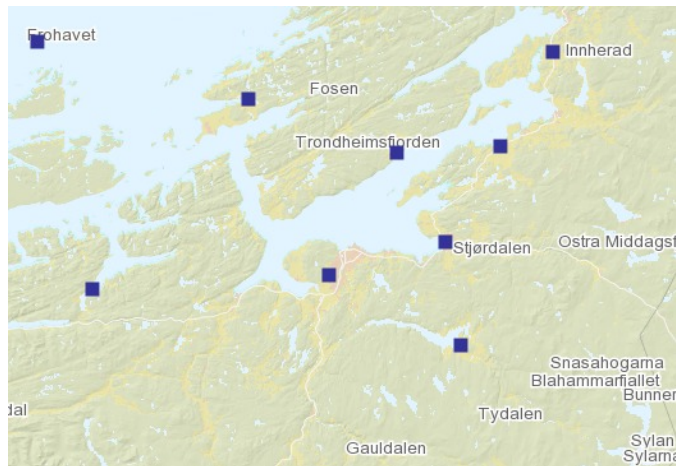


Figure 2.2: Mid-Norway GNSS base station overview [2].

Baseline	Search time with at least 5 satellites	Horizontal accuracy
1 km	1 - 600 sec	100 to 25 cm
1 km	600 - 3000 sec	25 to 5 cm
1 km	> 3000 sec	< 5 cm

Table 2.3: Single-frequency RTK convergence numbers [33]

## 2.4 GNSS Attitude determination

By measuring the carrier phase using multiple antennas, the attitude can be determined. With three antennas, roll, pitch and yaw angles can be estimated, given that the antennas are not placed on a straight line. With two antennas placed as in Figure 2.5, pitch and yaw can be estimated. It is crucial that the receiver clocks are synced for this to be possible. Expected accuracy of GNSS Attitude determination is shown in Equation 2.4

$$Accuracy = \frac{0.2}{baseline} \text{ } ^\circ \quad (2.4)$$

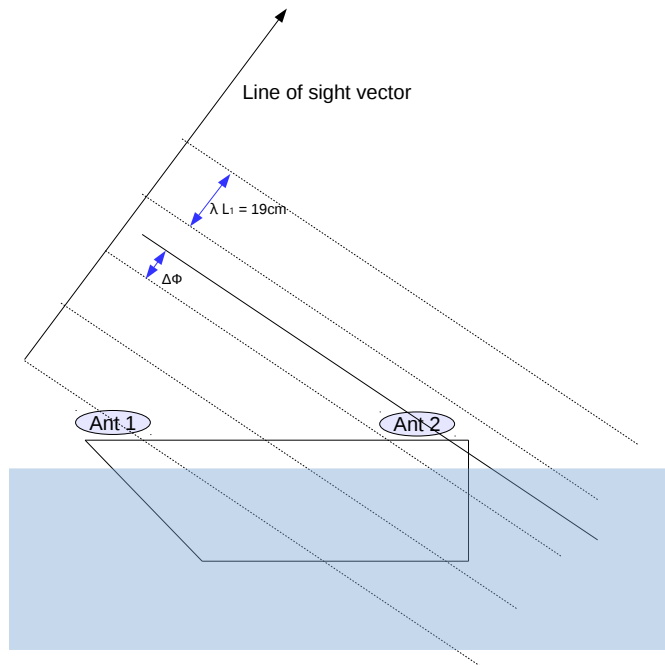


Figure 2.3: Attitude determination using GNSS

## 2.5 Current market

In this Section, a couple of GNSS receivers are listed, with some specifications and price. This is meant to get an overview of what high-end and low-end receivers can offer and their pricing.

### 2.5.1 ublox LEA-M8T

The ublox LEA-M8T is a low cost receiver for the consumer market. It supports all the GNSS-systems, GPS, QZSS, GLONASS Galileo and BeiDou, but it lacks dual-frequency.

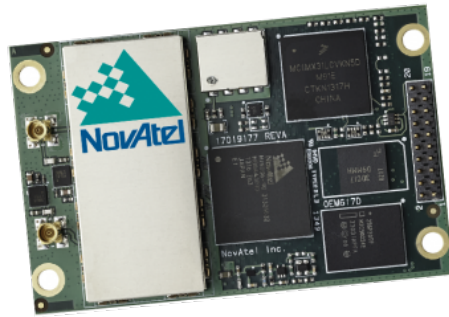
It has possibility for outputting raw data and precise timing, which makes it a suitable receiver to use when learning about GNSS or when high accuracy is not required. It can be used with DGNSS, but not with RTK nor attitude determination. The price of a development board, with chip is around \$150.



**Figure 2.4:** u-blox LEA-M8T

### **2.5.2 NovAtel OEM617D**

The NovAtel OEM617D is a dual-frequency, dual-antenna receiver. It is suited for both RTK and GNSS Attitude determination. It supports all the usual GNSS systems and the raw data is accessed easy through one of the three COM ports. The price is around \$2.500.



**Figure 2.5:** NovAtel 617D

# Chapter 3

## Inertial Navigation Systems

In this Chapter we will look into the basics of inertial navigation systems (INS). Fundamental theory will be presented as well as transformations between desired frames used for navigation. We will also look at common accelerometer and gyroscope technologies used by manufacturers today in INS.

### 3.1 Introduction

The theory behind pure inertial navigation is to integrate acceleration to obtain velocity and position in a desired coordinate frame. Attitude is obtained in the same manner using gyroscope measurements. Since all measurements from the IMU are given in the frame of reference, the body frame (often called vehicle frame), they must be transformed to the desired navigation frame.

There are two types of INS platforms; strapdown and stabilized-platform (gimbaled systems). In strapdown INS the unit is rigidly fixed to the moving body. Since strapdown systems move with the body, their gyroscopes will also experience and measure the same changes in angular rate as the body in motion. In gimbaled systems on the other hand, gimbals are used to isolate the inertial sensor assembly from vehicle rotation [20]. Today, strapdown systems are the dominating type of system. In later years, MEMS based sensors has improved enough to make this the dominating platform. Though, for application that requires extreme precision, strapdown system will not be sufficient. Example of application may be a nuclear submarine which may operate without GNSS fixed position for several days.

In this Chapter we will go deeper into strapdown systems and strapdown navigation equations.

## 3.2 Rotations and strapdown INS equations

The same naming scheme and parameters for equations as in [33] is used here. Also the same notation for vectors and matrices is used.

### 3.2.1 Coordinate frames and transformation

While navigating, a position is obtained in a certain coordinate frame. There are several types of coordinate frames, e.g. Earth Centered Coordinate Systems, Geographic Coordinate Systems and Vehicle Coordinate Systems. A brief overview of the different coordinate systems and their parameters is presented in Table 3.1.

Index	Coordinate system	Components
$i$	ECI	$x_i, y_i, z_i$
$e$	ECEF Cartesian	$x_e, y_e, z_e$
$n$	NED	$N, E, D$
$t$	Tangent frame	$x_t, y_t, z_t$
$b$	BODY	$x_b, y_b, z_b$
$a$	General frame	-

**Table 3.1:** Coordinate system variables index explanation [33]

Further information on the coordinate systems can be found in [18], [33]. While measurements from GNSS are computed in ECEF, this does not give much value for the end-user. In order to present position in terrestrial position, the coordinates are transformed into the ellipsoidal parameters longitude, latitude and height. Reference ellipsoid used for this transformation is known as WGS-84 (World Geodetic System 1984). Parameters used for the transformation are shown in 3.2.

Parameter	Comments
$r_e = 6378137m$	Equatorial radius of ellipsoid.
$r_p = 6356752m$	Polar axis radius of ellipsoid.
$\omega_e = 7292115 \cdot 10^{-11} rad/s$	Angular velocity of the Earth.
$\mu_g = 3986005 \cdot 10^8 m^3/s^2$	Gravitational constant of Earth.
$e = 0.0818$	Eccentricity of ellipsoid.

**Table 3.2:** WGS-84 parameters [33].

Variables used in the different systems are presented in Table 3.3 and 3.4. They will be used throughout this Chapter.

In order to transform coordinates from one coordinate system to another, rotation matrices are used. A rotation matrix is an orthogonal  $3 \times 3$  matrix with some special properties [33]. See [29] for detailed information on rotation matrices and their properties.

Rotation matrix from NED to BODY is given by:

Frame	Vectors	Comments
BODY	$\mathbf{v}^b = [u, v, w] \in \mathbb{R}^3$	Surge, sway and heave.
BODY/NED	$\boldsymbol{\omega}_{nb}^b = [p, q, r] \in \mathbb{R}^3$	Roll, pitch and yaw rate
NED	$\mathbf{v}^n = [v_N, v_E, v_D] \in \mathbb{R}^3$	North, east and down velocity
NED/ECEF	$\boldsymbol{\omega}_{en}^n = [\omega_N, \omega_E, \omega_D] \in \mathbb{R}^3$	Function of change in long. and lat.
ECEF	$\mathbf{v}^e = [v_{x_e}, v_{y_e}, v_{z_e}] \in \mathbb{R}^3$	Velocity in fram given by GPS.
ECEF/ECI	$\boldsymbol{\omega}_{ie}^e = [0, 0, \omega_e] \in \mathbb{R}^3$	$\omega_e$ is angular velocity of earh.

**Table 3.3:** Velocity and angular velocity vectors [33].

Frame	Vectors	Comments
BODY/NED	$\Theta = [\phi, \theta, \psi] \in \mathbb{S}^3$	Roll, pitch and yaw angles.
NED	$\mathbf{r}^n = [x_N, y_N, z_N] \in \mathbb{R}^3$	Position relative to a t-frame origin.
ECEF	$\Psi = [l, \mu] \in \mathbb{S}^2$	Geodetic longitude and latitude.
ECEF	$\mathbf{r}^r = [x_e, y_e, z_e] \in \mathbb{R}^3$	Cartesian ECEF position.

**Table 3.4:** Position and attitude vectors [33]

$$\begin{aligned}
 \mathbf{R}_b^n &= \mathbf{R}_{z,\psi} \mathbf{R}_{y,\theta} \mathbf{R}_{x,\phi} \\
 &= \begin{bmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\theta + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\theta + s\psi s\theta s\phi & -c\psi s\theta + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (3.1)
 \end{aligned}$$

where  $c = \cos$  and  $s = \sin$ .

Rotation between ECEF and NED frame is:

$$\begin{aligned}
 \mathbf{R}_n^e(\Psi) &= \mathbf{R}_{z_e, l} \mathbf{R}_{y_e, -\mu - \frac{\pi}{2}} \\
 &= \begin{bmatrix} -s\mu s l & -s l & -c\mu c l \\ -s\mu s l & c l & -c\mu s l \\ c\mu & 0 & -s\mu \end{bmatrix} \quad (3.2)
 \end{aligned}$$

where  $c = \cos$  and  $s = \sin$ .

### 3.2.2 Navigation equations

The vector  $\omega_{ib}^b = [p, q, r]$  is the inertial angular rate of the BODY frame expressed in body coordinates, which would be measured by an ideal set of platform-mounted gyros. Although the inertial-frame implementation results in the most straight-forward differential equations [18], it is not commonly used. There are two main reasons; the complications of calculation  $g$  and the desire for the navigation system to produce earth-relative position and velocity.

Relation between NED angular velocity, and longitude and latitude is found with [33]:

$$\begin{aligned}\omega_{nb}^b &= \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_{x,\phi}^T \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}_{x,\phi}^T \mathbf{R}_{y,\theta}^T \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \equiv \mathbf{T}_{\Theta}(\Theta)^{-1} \dot{\Theta}\end{aligned}\quad (3.3)$$

Continuing with the strapdown inertial navigation equations; see [33] for full derivation of the equations.

The accelerometers measures absolute acceleration and can not distinguish between the acceleration of gravity and inertial acceleration[18], Therefore the user must compensate the specific force output for the effects of gravity in the equations. Under acceleration, the specific force will be [33]:

$$\mathbf{f}^b = \mathbf{a}^b - \mathbf{R}_n^b \mathbf{g}^n \quad (3.4)$$

where  $\mathbf{a}^b$  is the variable of interest.

The NED velocity differential equation can be written [33]:

$$\dot{\mathbf{v}}^n = \mathbf{R}_b^n \mathbf{f}^b + \mathbf{g}^n - [2\mathbf{S}(\omega_{ie}^n) + \mathbf{S}(\omega_{en}^n)] \mathbf{v}^n \quad (3.5)$$

where  $\mathbf{S}$  denotes the skew symmetric matrix [29] and [33]. The rotation differential equation is then found using Equation 3.6

$$\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \mathbf{S}(\omega_{nb}^b) \quad (3.6)$$

By using  $\mathbf{v}^n$  calculations of latitude, longitude and height can be done using:

$$\begin{bmatrix} \dot{\mu} \\ \dot{l} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{r_{mer}+h} & 0 & 0 \\ 0 & \frac{1}{(r_{prime}+h)\cos\mu} & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{v}^n \quad (3.7)$$

where  $r_{prime}$  and  $r_{mer}$  are the prime and meridian curvatures of Earth defined in Table 3.2

$$r_{prime} = \frac{r_e(1-e^2)}{[1-e^2\sin^2\mu]^{1.5}}$$



$$r_{mer} = \frac{r_e}{[1 - e^2 \sin^2 \mu]^{0.5}}$$

Position can also be expressed in ECEF Cartesian coordinates:

$$\mathbf{r}^e = \mathbf{R}_n^e \mathbf{v}^n \quad (3.8)$$

This formula is specially useful when integrating INS with GNSS measurements, which is given in Cartesian ECEF coordinates.

Rotation in NED frame can be calculated directly from the NED velocities by using the following equations [33]:

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} \omega_N \\ \omega_E \\ \omega_D \end{bmatrix} = \begin{bmatrix} \frac{1}{r_{prime}+h} v_E \\ -\frac{1}{r_{mer}+h} v_N \\ -\frac{\tan \mu}{r_{prime}+h} v_E \end{bmatrix} = \mathbf{U} \mathbf{v}^n \quad (3.9)$$

where

$$\mathbf{U} = \begin{bmatrix} 0 & \frac{1}{r_{prime}+h} & 0 \\ -\frac{1}{r_{mer}+h} & 0 & 0 \\ 0 & -\frac{\tan \mu}{r_{prime}+h} & 0 \end{bmatrix} \quad (3.10)$$

## 3.3 INS errors

There are several error sources related to the IMU and the alignment. Some of the most important are mentioned in this Section. There is also a more detailed explanation on bias stability.

### 3.3.1 Error sources

A brief overview of accelerometer and gyroscope error sources are found in the list below [18].

1. Accelerometer errors

- *Accelerometer measurement noise*: Random error added to measurements
- *Accelerometer bias*: Bias in specific force measurements
- *Accelerometer scale factor*: Error in the accelerometer calibrated scale factor
- *Accelerometer nonlinearity*: Deviation from desired linear input/output relationship

2. Gyroscope errors

- *Gyro measurement noise*: Random error added to measurements
- *Gyro drift (bias)*: Bias in angular rate measurements
- *Gyro scale factor*: Error in gyros calibrated scaling factor
- *Gyro alignment*: Misalignment of the gyro measurement axis from the orthogonal platform axis
- *Gyro g sensitivity*: Sensitivity of the gyro output due to force applied along or perpendicular to the sensitive axis of the gyro

3. Gravity anomalies and deflection of the vertical

4. Numeric computation error

### 3.3.2 Bias stability

Bias is a long term average of measurement data from a IMU. The biases tends to change, it is called bias instability. It is meaningless in terms of single data points to look at bias instability, longer datasets are needed. If the bias is constant, it can easily be cancelled out by subtracting it from the measurements. Biases in gyros and accelerometers tends to vary. The change in bias measurements is often a part of the specification of IMUs given in  $^{\circ}/hour$  or  $^{\circ}/sec$  for gyroscopes and  $\frac{m}{s^2}$  or  $g$  for accelerometers. A bias stability measurement tells you how stable the bias of a sensor is over a certain period of time.

To determine an IMUs bias parameters, the Allan variance (also known as two-sample variance) can be calculated. The Allan variance (AVAR) is a measure of frequency stability in clocks, oscillators and amplifiers [7]. It is defined as:

$$\begin{aligned}\sigma_y^2(\tau) &= \langle \sigma_y^2(2, \tau, \tau) \rangle \\ &= \frac{1}{2} \langle (\bar{y}_{n+1} - \bar{y}_n)^2 \rangle\end{aligned}\quad (3.11)$$

where  $\tau$  is the observation period,  $\bar{y}_n$  is the  $n$ th fractional frequency average over the observation time  $\tau$ . Then, the Allan deviation (ADEV), often called root Allan variance, is given by [7]:

$$\sigma_y(\tau) = \sqrt{\sigma_y^2} \quad (3.12)$$

From Allan deviation, angle random walk (ARW) for gyroscopes and velocity random walk (VRW) for accelerometers can be estimated. ARW and VRW is a noise specification, in units of  $^\circ/\sqrt{\text{hour}}$  for ARW and  $\text{m/s}/\sqrt{\text{hour}}$  for VRW, that is directly applicable to angle calculations. It describes the average deviation or error that will occur when you integrate the signal.

### 3.4 INS Error Dynamics

The errors of the navigation parameters are defined as:

$$\begin{aligned}\delta r^n &= r^n - \hat{r}^n \\ \delta v^n &= v^n - \hat{v}^n \\ \hat{R}_b^n &= (I - E^n)R_b^n\end{aligned}\quad (3.13)$$

Where the elements with the  $\hat{\phantom{x}}$  are the estimated values and the symbol  $\delta$  is the error.  $E^n$  is the skew-symmetric vector of the attitude errors in the ENU frame.

The behaviour of the navigation parameter errors is obtained using the navigation equations in Section 3.2.2.

$$\begin{bmatrix} \delta \dot{r}^n \\ \delta \dot{v}^n \\ \dot{\epsilon}^n \end{bmatrix} = \begin{bmatrix} F_{rr} & F_{rv} & F_{r\epsilon} \\ F_{vr} & F_{vv} & F_{v\epsilon} \\ F_{\epsilon r} & F_{\epsilon v} & F_{\epsilon\epsilon} \end{bmatrix} \begin{bmatrix} \delta r^n \\ \delta v^n \\ \epsilon^n \end{bmatrix} + \begin{bmatrix} 0_{3 \times 1} \\ R_b^n \delta f^b \\ R_b^n \delta \omega^b \end{bmatrix}\quad (3.14)$$

Where

- $\delta r^b$  is the vector of errors in the measured position,
- $\delta v^b$  is the vector of errors in the measured velocity,
- $\epsilon^n$  is the vector of attitude errors in the ENU frame,
- $\delta f^b$  is the vector of errors in the measured acceleration,
- $\delta \omega^b$  is the vector of errors in the measured angular rate

The F matrix consists of velocity integration, gravity, Coriolis, centrifugal, Earthrate compensation and leveling. The full derivation of the equation is in [20], [27].

## 3.5 INS Technologies

### 3.5.1 Gyroscopes

For integrated navigation solutions using GNSS and INS, the most widely used gyroscopes are MEMS gyroscopes, FOG and RLG. MEMS gyroscopes are known to be low-cost and are easily produced in large numbers. In high-end integrated systems FOGs and RLGs are preferred due to their accuracy compared to MEMS gyroscopes.

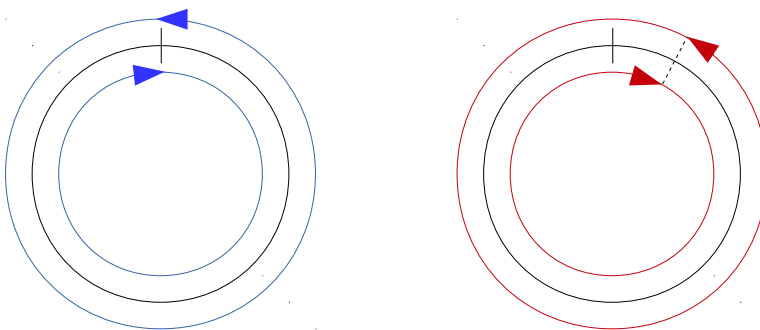
#### Microelectromechanical systems

Microelectromechanical systems (MEMS) is a kind of vibrating structure gyroscope and is named Coriolis vibratory gyroscope (CVG) by IEEE [11]. MEMS gyroscopes are known to be inexpensive but also less accurate than gyroscopes using fiberoptical technologies.

Manufacturers that provide IMUs which uses the MEMS technology are Panasonic, Analog Devices and Sensoror. In Chapter 6, Sensorors STIM300 is used for the inertial navigation part of the system.

#### Fiber optic gyroscope

A Fiber Optic Gyroscope (FOG) uses the interference of light to detect rotation of the body. This is called the Sagnac effect. A beam of light is split up and the two beams are made to follow the same path, but in opposite direction. In Figure 3.1 the principle of the Sagnac effect is demonstrated. To the left in the figure, there is no rotation. To the right, on the other hand, there is a clockwise rotation of the apparatus. The relative phases of the two beams, and thus the position of the interference fringes, are shifted according to the angular velocity of the apparatus [10].



**Figure 3.1:** The Sagnac effect

In FOGs the light beam is being sent through a coil of optical fibre which can be as long as 5 km. Advantages of the FOG vs a MEMS gyroscope is that it provides extremely precise rotational rate information.

**Ring laser gyroscope**

Ring Laser Gyroscope (RLG) uses the same Sagnac effect as FOG to detect rotation. Instead of running the light through a fiber coil, the light beam is sent in a closed-loop path with mirrors in the corners.

**3.5.2 Accelerometers**

An accelerometer is a device that measures proper acceleration (i.e. the physical acceleration experienced by an object) [6]. There is a wide range of technologies used in today's accelerometers, but the most important types of accelerometers are mechanical and vibratory accelerometers [33]. Mechanical accelerometers exploit Newton's laws to measure the specific force on the unit. Vibratory accelerometers measure shift in frequency due to changes in tension [33]. We will not look further into the two technologies here.

## 3.6 Current market

### 3.6.1 Sparkfun Razor IMU

For \$80 one can buy the entrance-level IMU from Sparkfun. Gyroscopes used are ITG-3200. At a zero rate they have a typically output of  $\pm 40^\circ/sec$ . Accelerometers used are ADXL345 which have a bias instability of typically  $\pm 40mg$ . Boards like this Sparkfun IMU board is often used onboard of small quadcopters for navigation.



Figure 3.2: Sparkfun Razor IMU

### 3.6.2 Sensoror STIM300

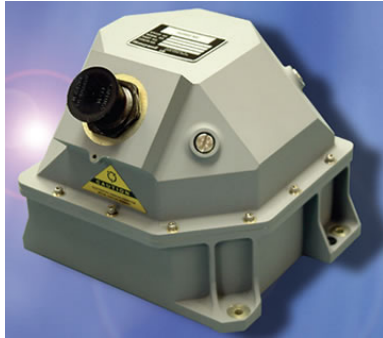
STIM300 is a mid-range IMU with pricings around \$7500. MEMS gyroscopes are cheaper to manufacture, but also not as accurate as high-end IMUs using RLG or FOG technologies. With a smaller form factor than IMUs using RLG or FOG they can fit into a wider range of applications. The STIM300 is specified with nominal ARW  $0.15^\circ/\sqrt{hour}$  and bias instability  $0.5^\circ/hour$  for the gyroscopes. For the accelerometers it has a nominal VRW of  $0.06 m/s/\sqrt{hour}$  and bias instability  $0.05 mg$  [26].



Figure 3.3: Sensoror STIM300

### 3.6.3 Honeywell HG9900 IMU

This is a high-end IMU from Honeywell. It is well-performing with specifications like bias stability  $< 0.003 \text{ }^\circ/\text{hour}$  and ARW  $< 0.002 \text{ }^\circ/\sqrt{\text{hour}}$  for the gyroscopes. For accelerometers the specification for bias stability is  $< 25 \text{ } \mu\text{g}$ . Gyroscopes of high accuracy can be used for north-seeking. The price for a IMU of this type is around \$100k.



**Figure 3.4:** Honeywell HG9900 IMU





# Integration of GNSS and INS

A pure inertial navigation system will suffer from unbounded error growth [18] in the position and velocity errors due to the integration of inertial measurements which will contain various types of errors. INS could not be trusted for as a highly reliable stand-alone navigation solution over longer time periods [18].

With the long term accuracy of GNSS in combination with the high output rate, high dynamic capability, robustness and reliability of INS, one can offer a more precise and accurate navigation system by integration of the two. Integration is done by applying different kinds of Kalman filtering techniques. In 1960, Rudolf Kalman presented a new approach to linear filtering and prediction problems [23] which have evolved to be a cornerstone in implementations of navigation systems.

When discussing integration of GNSS and INS systems, the term coupling is widely used. There are several ways of integrating the two system - it ranges from uncoupled to ultra-tight coupled. There are no clear definition of what is defining many of the types. Next we will look at a few coupling methods, and define what they contain.

## 4.1 Integrated systems

GNSS is the most common method for navigation users worldwide. Some applications require a more accurate and robust position- and attitude-estimate. Inertial sensors can help improve the system bandwidth, provide error estimations and even allow navigation during brief GNSS outages.

The Integration architectures are usually divided into four terms. Uncoupled, loosely coupled, tightly coupled and ultra-tight coupled. Ultra-tight is the most complex integration, and requires a more complex hardware than the others. In this Section a more detailed explanation of the couplings will follow.

### 4.1.1 Uncoupled integration

The least complicated integration method for GNSS and INS, is the uncoupled integration. As the name states, the two systems are not coupled. The uncoupled scheme uses a linearized filter and estimates the errors when GNSS measurements are received. GNSS and INS produce independent navigation solution with no influence on the other. An uncoupled system tends to drift rapidly according to the grade of the IMU in absence of GNSS data. For low-cost MEMS IMU it will propagate relatively large navigation errors in a small time interval. Without a feedback loop for INS strapdown calculations high error values can spoil the small error assumptions made in an INS error model, which will make the linearization not valid. It could potentially degrade system performance.

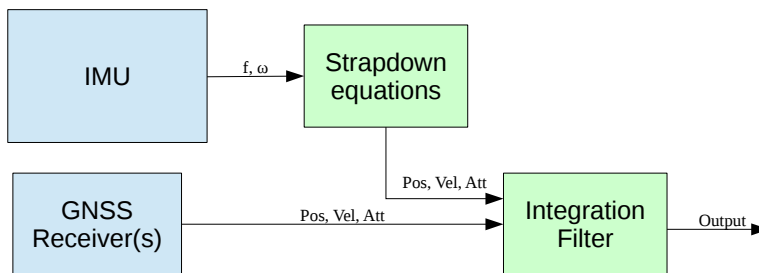


Figure 4.1: Uncoupled coupled integration

### 4.1.2 Loosely coupled integration

Loosely coupling is the simplest integration method. In comparance with a standalone GNSS receiver it has higher bandwidth, better attitude estimation and is fairly good during GNSS outages. It is the least complicated closed-loop solution where the INS errors is corrected by the integration filter. As we can see from Figure 4.2 it uses fixed values (Pos, Vel, Att) from the GNSS in the integration filter. All raw GNSS measurements are independently processed in the receiver.

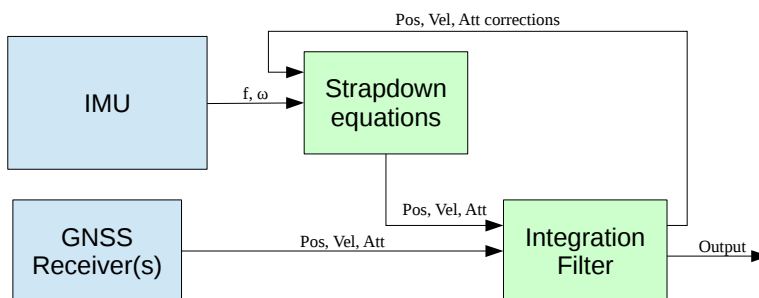


Figure 4.2: Loosely coupled integration

### 4.1.3 Tightly coupled integration

Tightly coupled is a much more robust integration. It is often referred to as a "centralized" filter. By using raw data (PR, CP and Doppler) from GNSS, the integration filter can estimate errors for every single GNSS measurement. The tightly integrated filter can utilize its knowledge of GNSS raw data propagation to predict future measurements. In occasions where the receiver has lock on less than 4 satellites it is not able to calculate position. A tight integration is able to utilize the measurements from even one single satellite. The advantage of this is particularly applicable where GNSS coverage is poor. Loose and tight integrations basically differ in the type of information that is shared between the systems. A lot more computation is carried out in the Kalman filter in case of the tightly coupled filter.

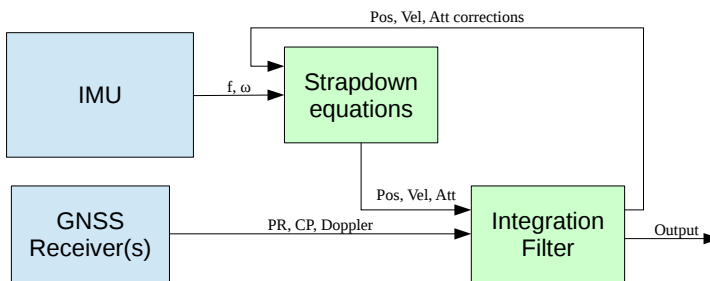


Figure 4.3: Tightly coupled integration

### 4.1.4 Ultra-Tight integration

In the Ultra-Tight integration strategy, the GNSS and INS devices do not work as independent systems anymore. Measurements from GNSS are used to estimate INS errors and measurements from INS are used to aid the GNSS receiver tracking loops. This is clearly at a deeper level than the other integration strategies and it requires access to firmware in the receiver to be able to achieve Ultra-Tight integration. The Ultra-Tight integration scheme is shown in Figure 4.4.

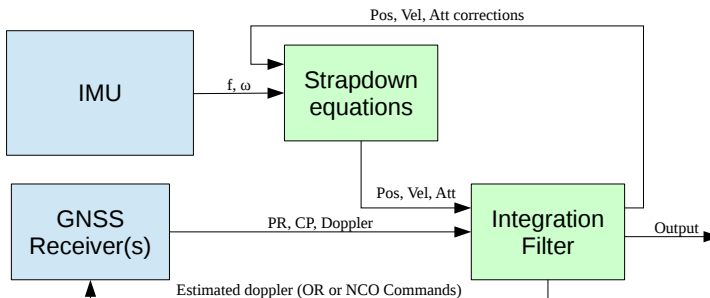


Figure 4.4: Ultra-Tight integration

### **4.1.5 Hardware requirements**

#### **Loosely coupled integration**

For a loosely coupled system the GNSS receiver outputs position only. This is easily done using the NMEA standard sentences[4]. When fusing data from IMU together with GNSS, the time needs to be synchronized. As data lines from the receiver are not timing critical, they should not be used for timing purposes alone. The receiver needs a timing output, e.g. PPS, to be used for synchronizing the data. The IMU-data could be synced with the GNSS clock using either external trigger on the IMU or timetagging on time of validity(TOV) output.

#### **Tightly coupled integration**

A Tightly coupled integration requires raw data from the GNSS in addition to the synchronization features with the IMU. Raw data is easily exchanged using RINEX(The Receiver Independent Exchange Format).

## **4.2 Kalman Filtering**

After Rudolf Kalman presented a new approach to linear filtering and prediction problems [23], later known as the Kalman filter, in 1960 researchers has used the work of Kalman to adapt to their control problems. Since the Kalman filter is only the optimal solution if it is a linear problem, linearization has been important considering that many of the early applications were navigation where the measurements are nonlinear.

There are two common approaches to the nonlinear systems, that is the linearized Kalman filter and the extended Kalman filter. Also, it has been presented some newer extensions of the basic filter the recent years [14]. Three of these are the ensemble, unscented and particle filter. We will not go further into the the newer extensions here. Only the linearized Kalman filter and the extended Kalman filter will be presented here, with the most emphasis on the extended Kalman filter.

### **4.2.1 Linearized Kalman filter**

The linearized Kalman filter linearize the non-linear equations about some nominal trajectory in state space that does not depend on the measurement data. That means it assumes the trajectory is known in advance, such that the Kalman gain can be computed offline and stored on a computer [33].

### 4.2.2 Extended Kalman filter

The Extended Kalman filter (EKF) is one of the earliest extensions of the basic Kalman filter and also probably the most widely used. Due to non-linearities in process dynamics, in the measurements or both, linearization is needed in order to solve the Kalman filter equations for these systems.

In the extended Kalman filter, the equations are expanded about the best current estimate by using the partial derivatives of the nonlinear equations. The equations for the filter algorithm can be found in 4.1.

Design matrices	$\mathbf{Q}(k) = \mathbf{G}^\top(k) > 0, \mathbf{R}(k) = \mathbf{R}^\top > 0$
Initial conditions	$\bar{\mathbf{x}}(0) = \mathbf{x}_0$ $\bar{\mathbf{P}}(0) = E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^\top] = \mathbf{P}_0$
Kalman gain matrix State estimate update Error covariance update	$\mathbf{H}(k) = \left. \frac{\delta \mathbf{h}}{\delta \mathbf{x}} \right _{\mathbf{x}=\bar{\mathbf{x}}(k)}$ $\mathbf{K}(k) = \bar{\mathbf{P}}(k) \mathbf{H}^\top(k) [\mathbf{H}(k) \bar{\mathbf{P}}(k) \mathbf{H}^\top(k) + \mathbf{R}(k)]^{-1}$ $\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k) + \mathbf{K}(k) [\mathbf{y}(k) - \mathbf{h}(\bar{\mathbf{x}}(k), k)]$ $\hat{\mathbf{P}}(k) = [\mathbf{I} - \mathbf{K}(k) \mathbf{H}(k)] \bar{\mathbf{P}}(k) [\mathbf{I} - \mathbf{K}(k) \mathbf{H}(k)]^\top + \mathbf{K}(k) \mathbf{R}(k) \mathbf{K}^\top(k)$
State estimation propagation Error covariance propagation	$\Phi(k) = \left. \frac{\delta \mathbf{f}}{\delta \mathbf{x}} \right _{\mathbf{x}=\hat{\mathbf{x}}(k)}$ $\bar{\mathbf{x}}(k+1) = \mathbf{f}(\hat{\mathbf{x}}(k), \mathbf{u}(k), k)$ $\bar{\mathbf{P}}(k+1) = \Phi(k) \hat{\mathbf{P}}(k) \Phi^\top(k) + \Gamma(k) \mathbf{Q}(k) \Gamma^\top(k)$

**Table 4.1:** Discrete extended Kalman filter [33].

Estimating the error states of the integrated system is a common approach for the PVT problem. When error states are estimated in the filter we call it an indirect extended Kalman filter. On the other hand, if the states are estimated directly in the filter it is a direct method.

### 4.2.3 INS Error Models

Uncertainties in sensors and gravity field will cause the navigation parameters obtained from the INS strapdown equations to contain errors. In order to obtain a good navigation solution, these errors need to be modeled and compensated for. Many models to describe the time-dependent behaviour of INS sensor error have been developed. The choice of which to use is mainly dependent on the application. Here, we will take a closer look at the Phi-Angle Error Model.

The Phi-Angle Error Model is given in the true n-frame (navigation frame, here NED). This is a classical approach that has the benefit of perturbing navigation parameters in respect to the true n-frame.

Phi-Angle perturbation model can be written as follows [27]:

$$\delta \dot{\mathbf{r}}^n = -\omega_{en}^n \times \delta \mathbf{r}^n + \delta \theta \times \mathbf{v}^n + \delta \mathbf{v}^n \quad (4.1)$$

$$\delta \dot{\mathbf{v}}^N = \mathbf{C}_b^n \delta \mathbf{f}^b + \mathbf{C}_b^n \mathbf{f}^b \times \phi + \delta \mathbf{g}^n - (\omega_{ie}^n - \omega_{in}^n) \times \delta \mathbf{v}^n - (\delta \omega_{ie}^n + \delta \omega_{in}^n \times \mathbf{v}^n) \quad (4.2)$$

$$\dot{\phi} = -\omega_{in}^n \times \phi + \delta \omega_{in}^n - \mathbf{C}_b^n \delta \omega_{ib}^b \quad (4.3)$$

### 4.2.4 Sensor errors

In low-cost IMUs, the turn-on errors can be significant from time to time. In order to get uncertainties down, we need to determine these errors. This is done as much as possible through calibration. But even after calibration, specially in low-cost MEMS IMUs, there will still exist error variations. Therefore we need to model this in the state vector of the navigation filter in addition to the core variables (position, velocity and attitude errors).

Estimation of sensor errors is done by using simplified models. In reality, the error propagation is more complicated than the models implemented. Modeling IMU sensor errors can be done in a various ways, but modeling sensor bias and scale factor as shown in Equation 4.4 is a common approach. This is investigated to a deeper extend in [18].

The evolution of the sensor errors can be expressed as follows [27]:

$$\dot{\mathbf{b}}_a = \text{diag}(\mathbf{c}_{ab})\mathbf{b}_a + \mathbf{w}_{ab}, \quad (4.4a)$$

$$\dot{\mathbf{b}}_g = \text{diag}(\mathbf{c}_{gb})\mathbf{b}_g + \mathbf{w}_{gb}, \quad (4.4b)$$

$$\dot{\mathbf{s}}_a = \text{diag}(\mathbf{c}_{as})\mathbf{s}_a + \mathbf{w}_{as}, \quad (4.4c)$$

$$\dot{\mathbf{s}}_g = \text{diag}(\mathbf{c}_{gs})\mathbf{s}_g + \mathbf{w}_{gs}, \quad (4.4d)$$

where  $\mathbf{c}_{ab}$ ,  $\mathbf{c}_{gb}$ ,  $\mathbf{c}_{as}$  and  $\mathbf{c}_{gs}$  denotes the continuous-time sensor error model parameters for the random walk and the random constant. All  $\mathbf{w}$ 's are first-order Gauss-Markov processes

for their respective state.

The investigation of the behaviour of the sensor errors depends on the given application to the system. If the system operation time is very short, one can consider treating all sensor errors practically as constants. On the other hand, if the system is going to be in operation over a longer period of time, the IMU error behaviour must be investigated further.

### 4.2.5 Filter design

In this Section we will take a further look at Kalman Filter design, and develop the discrete-time extended Kalman filter as presented in 4.1. First, we will look at the discrete-time system model, then we will dive into a linearized measurement model for the error state model and look at the measurement space for the aiding sensors.

The linear continuous-time model for the system is in the following form:

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t), \quad (4.5)$$

where  $\mathbf{F}$  is the process error dynamics matrix,  $\delta\mathbf{x}$  is the error state vector,  $\mathbf{G}$  is the noise-input mapping matrix and  $\mathbf{w}$  is the noise vector.

Since strapdown inertial systems not are continuous-time systems, but usually implemented with a high-rate sampled data, we need a corresponding discrete-time form for the continuous-time form above 4.5:

$$\delta\mathbf{x}(t_{k+1}) = \Phi(t_{k+1}, t_k)\delta\mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)\mathbf{G}(\tau)\mathbf{w}(\tau)d\tau, \quad (4.6)$$

where the state transition matrix  $\Phi$  is the first order approximation by using:

$$\Phi(\mathbf{t}) = e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t \quad (4.7)$$

In abbreviated form, the system can be obtained:

$$\delta\mathbf{x}_{k+1} = \Phi_k\delta\mathbf{x}_k + \mathbf{w}_k, \quad (4.8)$$

where  $\Phi_k$  is the state transition matrix as before, and  $\mathbf{w}_k$  is the driven response at  $t_{k+1}$  due to the presence of the input white noise during current time interval [14].

In order to achieve a well performing extended Kalman filter, the choice of  $\mathbf{Q}$  is crucial. Too small value for elements in  $\mathbf{Q}$  will lead to overconfidence in the accuracy of the estimates and, as a consequence, the filter will not pay attention to new measurements (Kalman gain gets to small). This can cause filter divergence.

## 4.3 Heave

Heave (vertical boat movement resulting from wave action) is especially destructive when doing hydrographic surveys, as it affects the depth measurements directly. As GNSS have high dilution of precision in the vertical axis, extra precautions have to be taken. To ensure the best result INS computation together with tide tables is applied. The tide table is applied to cancel out any biases in heave calculation. When good RTK is present, heave is usually no issue.

## 4.4 Open-source software

There are many open-source communities around the world that develop software packages free to use. They often encourage users to contribute, which can be a great learning possibility. Many of these packages are generic and it can be hard to adapt functionalities due to different hardware requirements. They are often best suited when working with simulated datasets. They can be a great tool when learning about positioning systems and as a supplement when developing an integrated system.

### 4.4.1 RTKlib

RTKlib is an open-source software package that provides a wide selection of navigation and positioning possibilities. The software supports all major satellite constellations and major data formats. It has both a command-line user interface (CLI) and a graphical user interface (GUI). Executing GUI programs requires a Windows platform, while the CLIs also work for Linux [25]. It does not have support for integrating GNSS with an INS, though it is on the RTKLIBs To Do List [5]. RTKlib has two main modules; RTKNAV and RTKPOST.

#### RTKNAVI

RTKNAVI is a module for real-time positioning. It executes navigation processing in real-time taking raw observation data of GPS/GNSS as inputs. The real-time module supports a wide range of data formats such as u-Blox and NovAtel binary formats and has a wide range of configuring possibilities. There is also a function for converting from different binary formats to RINEX format.

#### RTKPOST

If raw GNSS data is collected, it is possible to post-process the trace with RTKPOST. Together with other inputs like precise ephemeris or correction messages, the accuracy of positioning can be improved. Data must be converted to RINEX format before post-processing.

### 4.4.2 INSTK

INSTK provides a toolkit for developing an inertial navigation system. It consists of a large number of octave/matlab scripts which implement the core navigation functions and algorithms. It also includes integration of GNSS through aided INS.



### 4.4.3 Mapping

Inspired by the success of Wikipedia, OpenStreetMap was created in 2004 with the aim to create a free editable map of the world. Since the start it has grown to over 2 million registered users. Map data is collected from scratch by volunteers performing systematic ground surveys [? ]. The amount of work put into this project is significant.

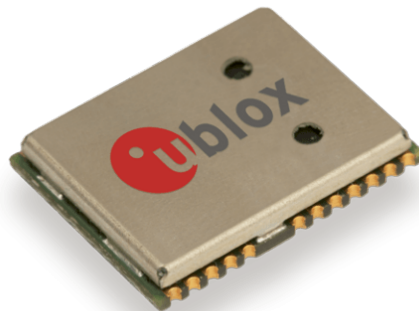
For plotting position data in maps, there are several open-source track drawing web services. One of them is uMap. uMap lets you create a map with OpenStreetMap layers and easily embed them in other web based software solutions.

## 4.5 Current market

The current market in integrated positioning solution offers a wide range of products from cheap all-in-one circuit boards to high-end systems containing state of the art INS with GNSS integration. In this Section we will present a brief overview of whats offered.

### 4.5.1 Ublox NEO-M8L

The NEO-M8L is a very low cost GNSS receiver with integrated 3-axis gyroscope and accelerometers[32]. It has an output rate up to 20 Hz, and can receive wheeltick-pulses from external sensors, to estimate sensor biases during GNSS outage.



**Figure 4.5:** Ublox NEO-M8L

### 4.5.2 NovAtel

Figure 4.6 shows how the NovAtel is integrated in the WBMS. The NovAtel 615 is for GNSS attitude determination with second antenna, and the MEMS Interface Card is mounted inside the WBMS for powering and time-sync the STIM300.

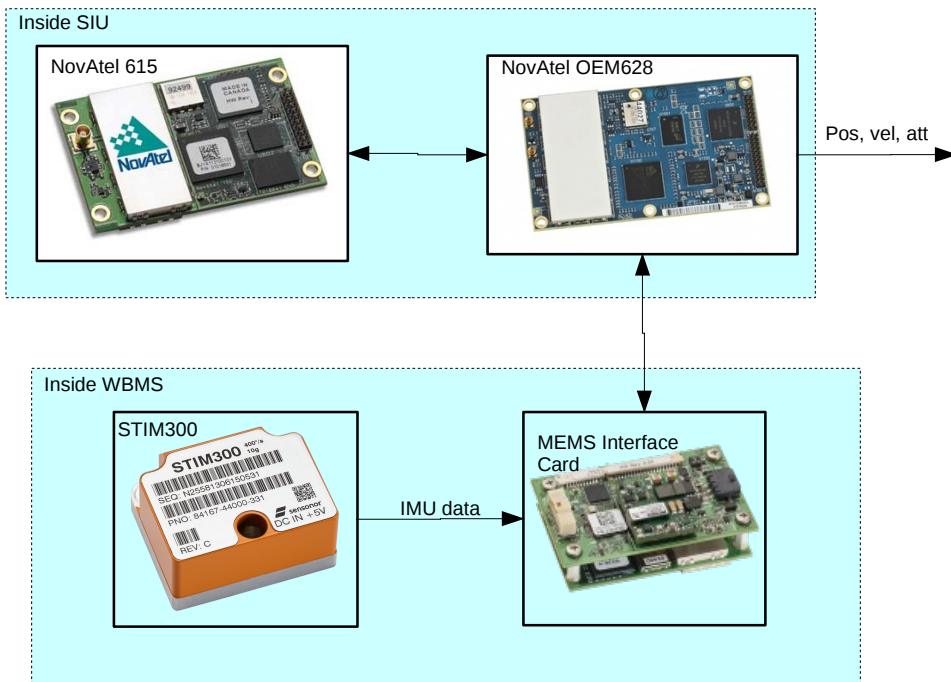


Figure 4.6: NovAtel integrated solution

### 4.5.3 Applanix Wavemaster II

The Applanix Wavemaster II is similar to Norbit's Applanix integrated system in Section 1.1.2. It is often used as external INS together with the WBMS. It consists of two GNSS antennas and one IMU. It outputs Real-time (up to 200Hz) over ethernet. It can do Tightly coupled integration, and RTK.



Figure 4.7: Applanix wavemaster II

# Chapter 5

## Development

This Chapter describes how a device containing a dual-frequency, dual-antenna GNSS receiver and a IMU was developed. The device will be referred to as the Navigation Module. The main intention of the module is to get time-synced GNSS and IMU measurements which will ease the process of implementing a Kalman filter later. In addition there will also be embedded a Odroid XU4, which is a small computer for data gathering and processing purposes.

### 5.1 Design notes

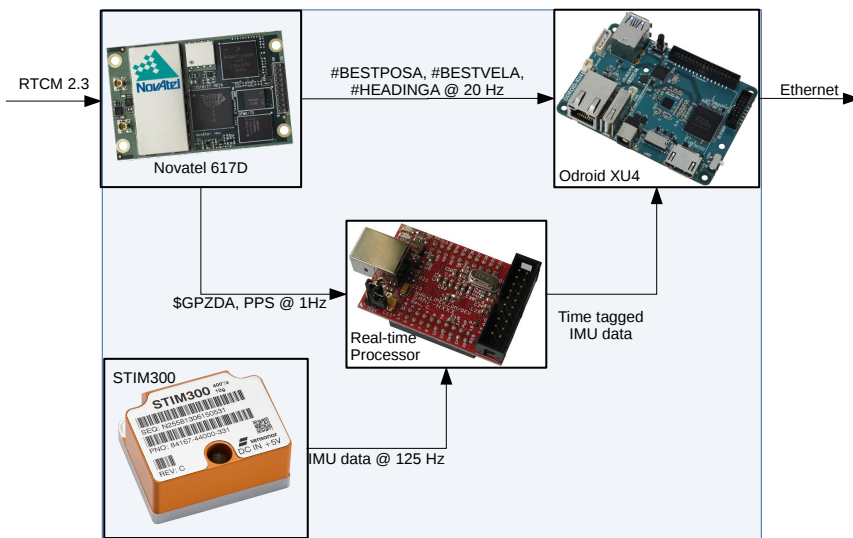


Figure 5.1: Block diagram of components and data flow in the setup

### 5.1.1 Interfaces

- **NovAtel OEM617D to Odroid XU4**

The Odroid and the NovAtel is connected through LVTTL.

- **NovAtel OEM617D to RTK source**

One of the LVTTL ports on the NovAtel goes through a RS232 transceiver and is connected to a DSUB9 on the front panel. Any RTK source with RS232 can be used.

- **NovAtel 617D to Real-time processor**

A LVTTL and PPS output from the NovAtel goes directly to the Real-time processor.

- **IMU to Real-time processor**

This data consists of RS422 and TOV (Time of Validity).

The RS422 (TX+) is used as output to the Real-time processor.

The TOV goes directly into Real-time processor, see Figure 5.2 for more details.

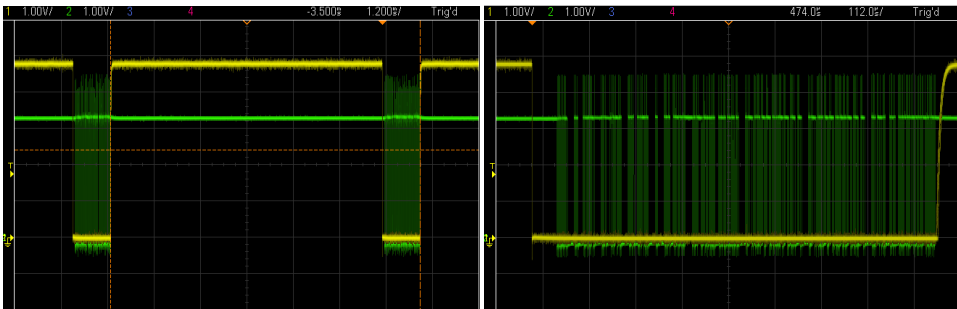


Figure 5.2: The TOV(yellow) and TX(green) outputs @ 125Hz

- **Real-time Processor to Odroid**

The TTL output goes through a TTL-232R-3V3 and into a USB port on the Odroid.

- **Output from Odroid**

The Odroid is connected to the world via 100Mbit ethernet.

### 5.1.2 NovAtel OEM617D

The GNSS receiver used in the Navigation Module is a NovAtel OEM617D. It has dual-antenna input and supports dual-frequency for GPS, GLONASS, BeiDou and Galileo. OEM617D comes on a PCB with antenna and communication (COM) connectors mounted. There are 3 LVTTL for communication with the unit. Some key figures from NovAtel datasheet on performance are mentioned in 5.1

The receiver uses the OEM6 family firmware and has a wide range of real-time output logging options. In this experiment, there where used NovAtel ASCII log sentences for position, velocity and attitude. The sentences are timetagged, and they contain information

	<b>RMS</b>	<b>Unit</b>
Single point L1/L2 position accuracy	1.2	m
DGPS position accuracy	0.4	m
RTK position accuracy	0.01 + 1ppm	m
Yaw accuracy 2m baseline	0.08	°
Velocity accuracy	0.03	m/s

**Table 5.1:** NovAtel OEM617D performance key figures.

about the measurement accuracy in terms of a standard deviation. The attitude measurements contains yaw and pitch. The receiver also has possibilities for outputting log in a binary format. For detailed description see [24].

The setup in the Navigation Module requires two LVTTL COM ports and a PPS output from the GNSS receiver board. If RTK is used during logging, all three COM ports are required. It is configured as follows:

- **COM1: NMEA \$GPZDA @ 1 Hz**  
Sent to real-time processor for timetagging purposes.
- **COM2: NovAtel ASCII log @ 20 Hz**  
A variety of NovAtel ASCII sentences for logging position, velocity and attitude on computer.
- **COM3: RTK input, RTCM 2.3 @ 1 Hz and \$GPGGA output**  
RTK input to receiver to enhance the precision of data derived from GNSS.
- **PPS @ 1 Hz**  
The PPS is configured to 1 Hz, 100ms pulse length, rising edge. It is used by the Real-time Processor for timing precision.

### 5.1.3 Real-time processor

The Real-time processor (AT91SAM7S256) is mounted on a development board (Olimex SAM7-H256). The development board is fitted with a 18.432 Mhz crystal, for precise timing. The Real-time processor is used to timestamp all data packets from the IMU.

During test of the timetagging function, it was observed that the crystal accuracy to keep track on time between each PPS was not sufficient. The number of ticks in crystal per second, varies enough to end outside the second marks. According to the datasheet the crystal's frequency may deviate with 30ppm. That means the counter could end outside its bounds during longer times of operation. This will cause non linearities in the IMU datastream. An adaptive tick rate has been implemented to avoid these situations. The tick rate is set to number of ticks between two PPS edges, which is the reference for timing.

- **UART1**  
Receives \$GPZDA from GNSS.
- **UART2**  
Receives IMU data from IMU.  
Delivers Time stamped IMU data to Odroid.
- **PPS**  
Used as a timestamp for the \$GPZDA.
- **TOV**  
Used as a timestamp for the IMU data.

### 5.1.4 STIM300

The inertial measurement unit (IMU) used in the Navigation Module is Sensorors STIM300. Expected performance is seen in Table 5.2. It outputs the following:

	<b>Gyros</b>	<b>Accelerometers</b>
Bias Range	$\pm 250^\circ/h$	
Bias on/off repeatability		$\pm 0.75mg$
Bias instability	$0.5^\circ/h$	$0.05mg$
Random walk	$0.15^\circ/\sqrt{hr}$	$0.06m/s/\sqrt{hr}$

**Table 5.2:** Sensoror STIM300 performance key figures[26].

- **Rate**  
X, Y and Z-axis gyro output.
- **Acceleration**  
X, Y and Z-axis accelerometer output.
- **TOV**  
Indicates the time of when the datagram should be timestamped.

### 5.1.5 Odroid XU4

The Odroid XU4 from Hardkernel co., Ltd is a single board computer. It runs Debian "Wheezy" GNU/LINUX 7. It has a 8-core processor from ARM, 2 GB DDR3 RAM and should be able to run a Kalman filter in real-time.

While testing the logging software with both GNSS receiver and IMU plugged in, it was discovered a lot of discrepancies in measurements from IMU. Approximately every 100th-300th measurement was a spike outside valid measurement range. This happens only when running simuntaneously from both UART ports, called SAC0 and SAC2. It could have been a kernel interrupt or hardware related bug. Further investigation to find the core of the bug has not been done, but the problem was solved by using a FTDI UART-USB adapter for IMU data.

Unfortunately, the stock Linux kernel for ARM that is precompiled does not include all hardware drivers which is needed for interfacing with serial port adapters. A custom Linux kernel with all drivers enabled was compiled.

## 5.2 Implementation

### 5.2.1 Real-time firmware

The software is intended for time-tagging of the IMU-data, so that the IMU-data can be synced with GNSS-data.

#### STIM300 interfacing

Sensoror STIM300 transmits measured data as packages of fixed format, called datagrams. Packages are transmitted at intervals given by the sample rate which can be set by the user (min. 125Hz, max. 2000Hz). For this experiment a sample rate of 125 Hz is used. Internally in the STIM300, the sample rate is 2000Hz. The outputted values at 125Hz is averaged over the internal sample rate. The datagram is binary coded, and needs to be decoded in the receiver end. STIM300 provides a selection of different measurements, such as gyro, accelerometer, inclinometer and sensor temperatures. See [26] for more information on datagram formats and interfacing with the unit.

#### Time tagging of IMU data

For best possible accuracy of IMU and GNSS data, they need to be synced in time. The IMU has no timing, so in order to get time synchronisation of IMU data we need to timestamp datagrams transmitted from STIM300. For that purpose an Atmel SAM7S256 microcontroller is used. It has two Universal Synchronous/Asynchronous Receiver/Transmitter (called USART1 and USART2 from now) channels [13]. One channel is used to receive GNSS/timing (USART1) info and the other IMU datagrams (USART2) as indicated in Figure 5.1.

A pulse per second (PPS) provided by the GNSS receiver is used as second mark for precise timing. When a PPS is received by the microcontroller, it resets an internal counter

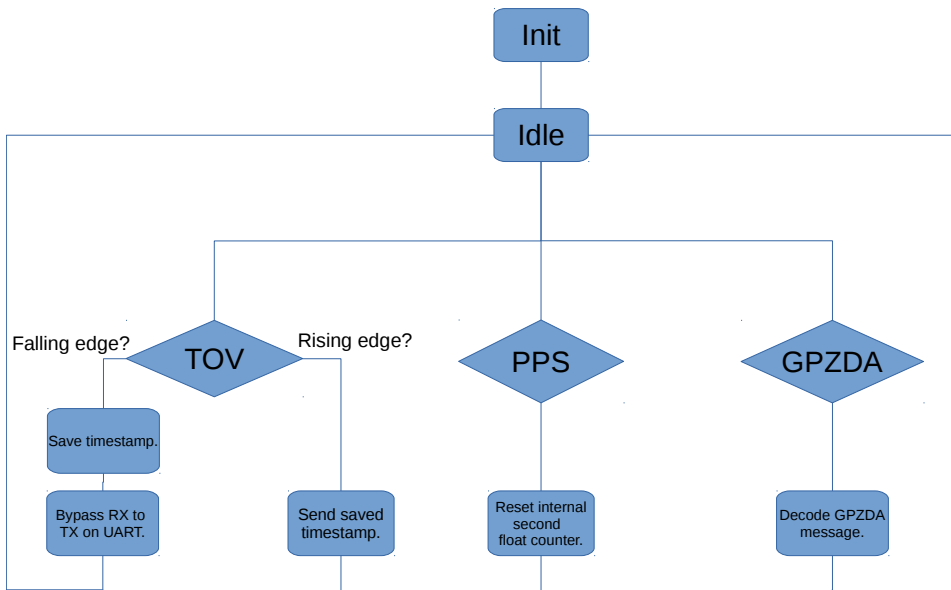


Figure 5.3: Flowchart of timetag software.

which keeps track of the decimal point of the second by using the internal system clock. The counter has 16 bits which theoretically will give a resolution of

$$\frac{1}{2^{16}} s = 1.53 \cdot 10^{-5} s \quad (5.1)$$

but that depends on a desired clock frequency offered by the main system clock. The best resolution we achieved with our was given by some division of the system clock. It has 46,924 cycles per second, which gives the resolution 0.0000213 second. This is a satisfying resolution compared to the output rate of IMU with a resolution of

$$\frac{1}{125} s = 8 \cdot 10^{-3} s \quad (5.2)$$

As we can see from Figure 5.2 TOV signal is normally high, and drops low when the datagram is being transmitted. From [26] we know that TOV is set low when the datagram is ready to be transmitted.

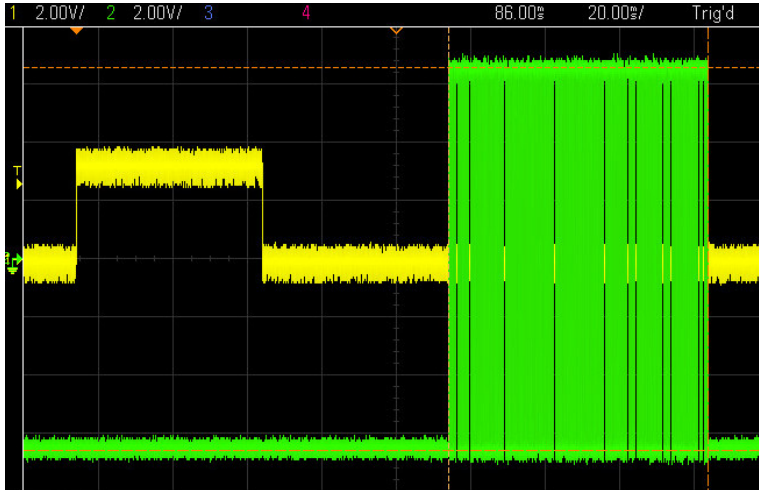
The intention with USART2 in the first place was to continuously read data from the channel while TOV is low, and then append the desired timestamp. The microcontroller was not able to handle the data rate while it also was supposed to read from the other USART channel and save timestamp.

Solution to this problem was to set the USART2 channel in bypass mode (See [13] for further information on different operation modes of AT91SAM7 USART channels) while TOV is low. Then, no data is handled by the code - everything is bypassed from RX to TX. When the rising edge of TOV is received (which means STIM300 is finished transmitting



datagram), bypass mode is turned off and the timestamp is sent in ASCII encoding.

The microcontroller continuously reads USART line from GNSS and checks for incoming GPZDA messages. GPZDA is a NMEA sentence that provides GNSS timing data. For more information on NMEA sentences and their structure, see [4]. GPZDA is transmitted by the GNSS receiver after PPS is sent. In order to know the correct time simultaneously as the PPS is received, the previous received timestamp from GPZDA needs to be incremented on incoming PPS. Figure 5.4 demonstrates a PPS and GPZDA signal stream.



**Figure 5.4:** PPS (yellow) and GPZDA (green) signals.

A flowchart diagram of the operating software for timetagging is shown in Figure 5.3. An internal interrupt handler of the microcontroller controls the scheduling when it receives inputs on different channels. Receiving PPS has the highest priority, then follows TOV. Reading from USART1 happens while no interrupts are received.

Two LED indicators are connected to the microcontroller and mounted above each connector as shown in Figure 5.6. The intention with the LED lights is to clarify for the user in which operational mode the Navigation Module is in. LED indicator behaviour is described in Table 5.3. Since datagrams are bypassed by the microcontroller, IMU LED indicates if TOV signal is received.

LED	Steady on	Flash @ 1Hz	Flash @ 5Hz	Off
PPS	No GPZDA	Timing OK	No PPS	No GPZDA or PPS
AUX		TOV OK		No TOV

**Table 5.3:** LED indicator behavior.

## 5.2.2 Hardware

The module was built in a water-tight aluminium housing found on the shelf at Norbit. It was already fitted with all the needed mounting holes for connectors and an interface board that eased the integration with the hardware used in this thesis. Cables, mounting holes and some PCB had to be custom built to connect all the transmission lines.

### Transmission line protection

To ensure no high currents on the data lines,  $1K\Omega$  resistors were added in series with the data lines.

### Power

The power supply (Traco Power TEN 8-2411WI) is galvanic isolated and rated for 9-36 VDC. During operation, the module runs on  $\approx 12W$ . That gives more than 50 hours of operation on a standard car battery (12V, 40Ah).



Figure 5.5: The inside of the Navigation Module

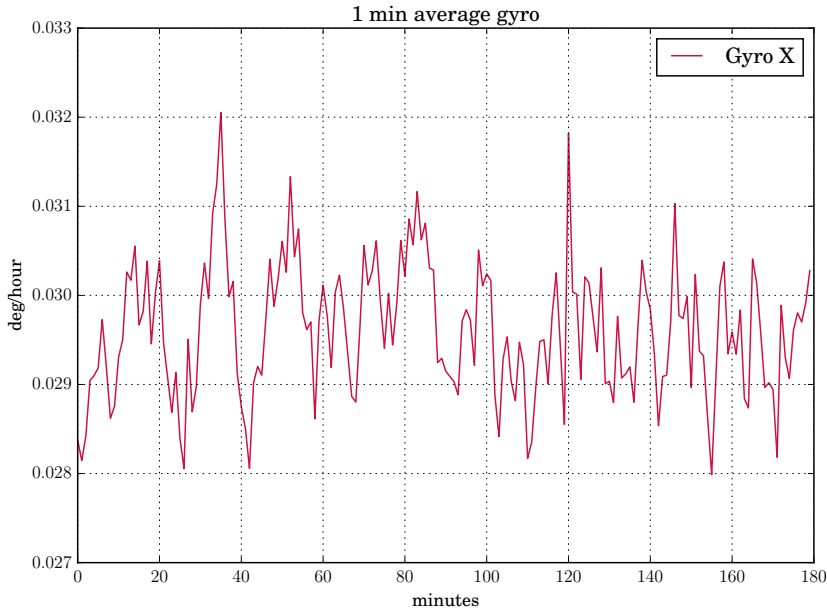


**Figure 5.6:** The outside of the Navigation Module.

### 5.3 Long term static test of IMU

One of the main drawback of MEMS based IMUs is the instability over period of time. By sampling data from the IMU under static circumstances over a long time period, the bias stability can be investigated. Data samples from during 5 hour long time period is used. Sampling frequency was, as mentioned earlier in this Chapter, 125 Hz.

Equation 3.11 was used on the dataset, to obtain Figure 5.9 and 5.10.

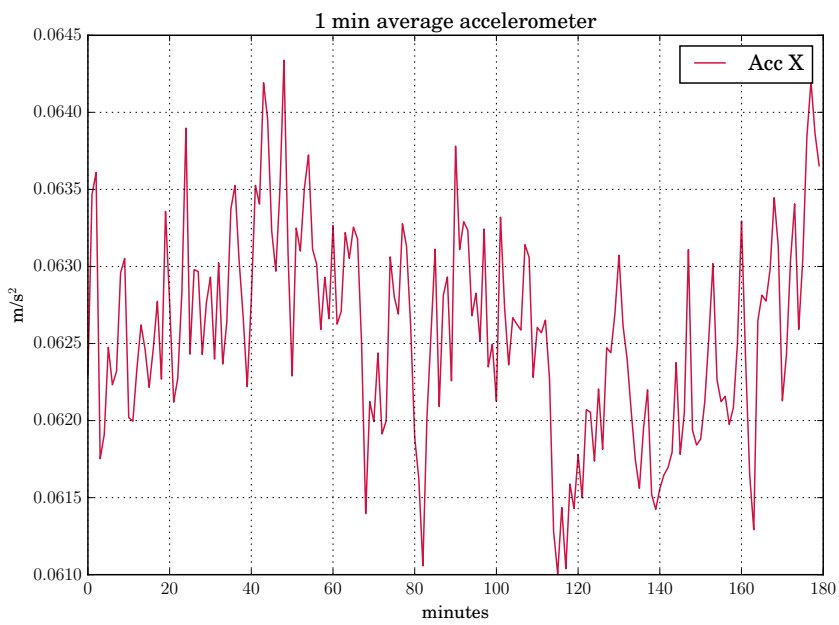


**Figure 5.7:** Gyro averaging values of  $\tau = 1$  min.

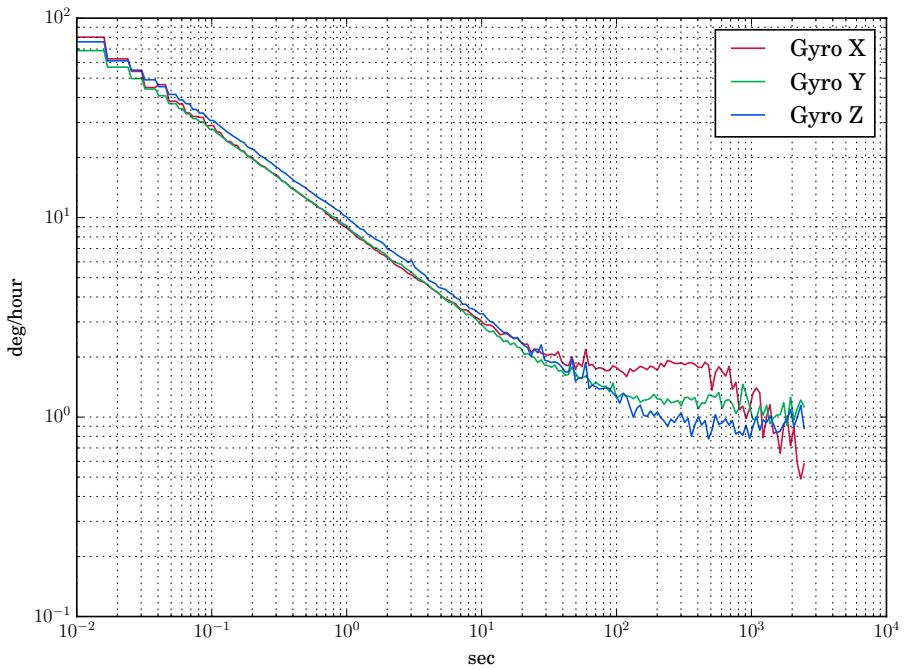
Figure 5.7 and 5.8 shows snippets from the dataset when computing Allan variance from respectively x axis gyroscope and accelerometer. Figures shows averaging values with bin size ( $\tau$ ) 1 minute.

Different error sources can be read out from the Allan deviation plot:

- Quantization noise: slope -1
- Velocity random walk: slope -0.5
- Bias instability: slope 0
- Acceleration random walk: slope 0.5
- Acceleration ramp: slope 1



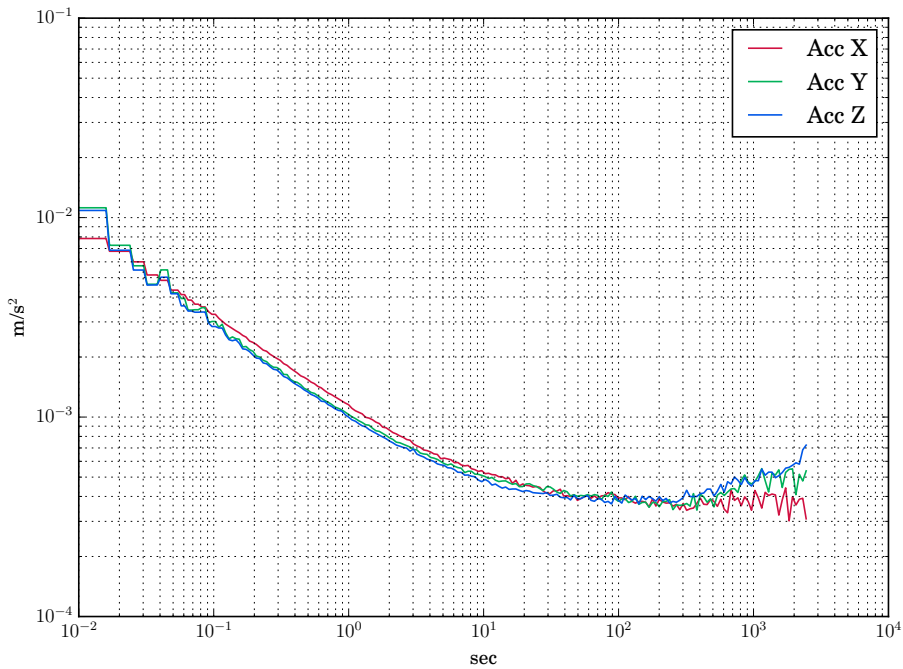
**Figure 5.8:** Accelerometer averaging values of  $\tau = 1\text{min}$ .



**Figure 5.9:** Root Allan variance of static gyro measurements

Parameter	Measurement	x	y	z	units
Angle Random Walk	AllVar plot at T = 1 sec	0.15	0.15	0.17	$^{\circ}/\sqrt{hour}$
Bias Instability	AllVar plot at slope = 0	1.7	1.3	0.9	$^{\circ}/h$
Correlation time	Block size at slope = 0	200	300	500	s

**Table 5.4:** Spectral density of gyroscopes, reading from Figure 5.9



**Figure 5.10:** Root Allan variance of static accelerometer measurements

Parameter	Measurement	x	y	z	units
Vel Random Walk	AllVar plot at $T = 1$ sec	0.072	0.06	0.06	$m/s/\sqrt{hour}$
Bias Instability	AllVar plot at slope = 0	0.0004	0.0004	0.0004	$m/s^2$
Correlation time	Block size at slope = 0	100	100	100	s

**Table 5.5:** Spectral density of accelerometers, reading from Figure 5.10

By analyzing Figure 5.9 and 5.10 bias instability can be estimated.

By comparing Tables 5.2, 5.4 and 5.5, it is concluded that all measurements are close to be as specified, except gyro bias instability. It was measured to be 2-3 times larger than specified.

The testing conditions while sampling static data was probably not ideal and the duration should have been longer. There may have been some small vibrations picked up by the IMU which have affected the results. The test in STIM300 datasheet [26] assumes a test temperature of 25°C. Room temperature during static data sampling was around 23°C.

## 5.4 Long term static test of GNSS

To get a understanding of the GNSS performance a long term static test was performed. Two roof-mounted antennas with 0.75m baseline were used and CPOS(RTK) from Kartverket was connected. The test setup can be seen in Figure 5.11, and the results in Figure 5.12, 5.13, 5.14 and 5.15. The test ran for approximately 510 minutes at 20 Hz and consists of 612237 data points. The key figures are gathered in Table 5.6 and are within the specification of the NovAtel OEM617D[22].

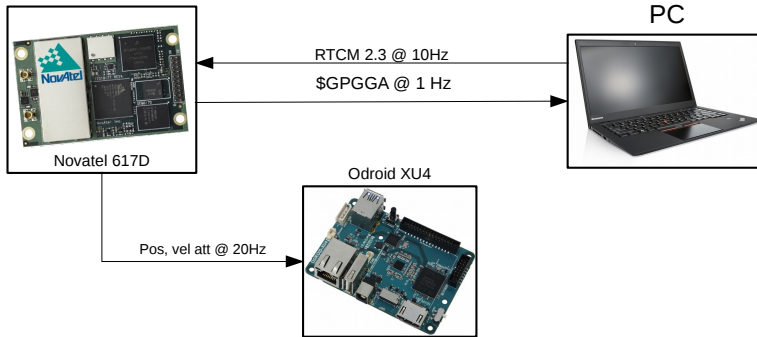


Figure 5.11: Setup for long term GNSS static test

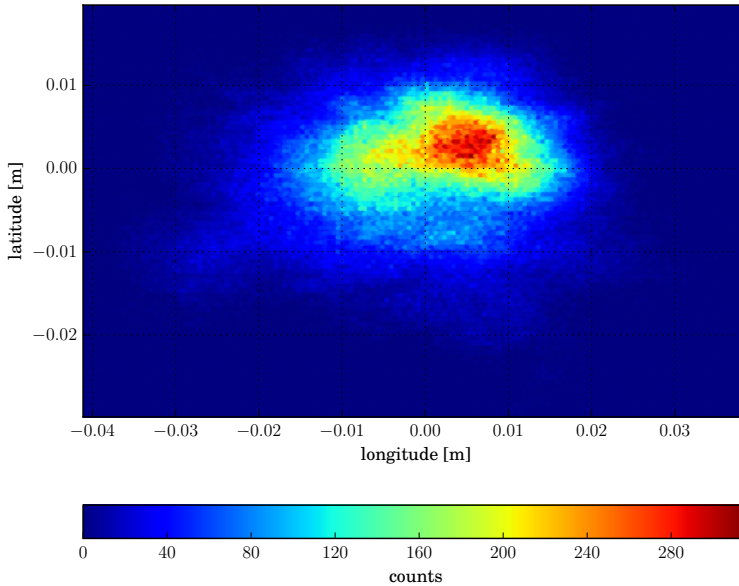
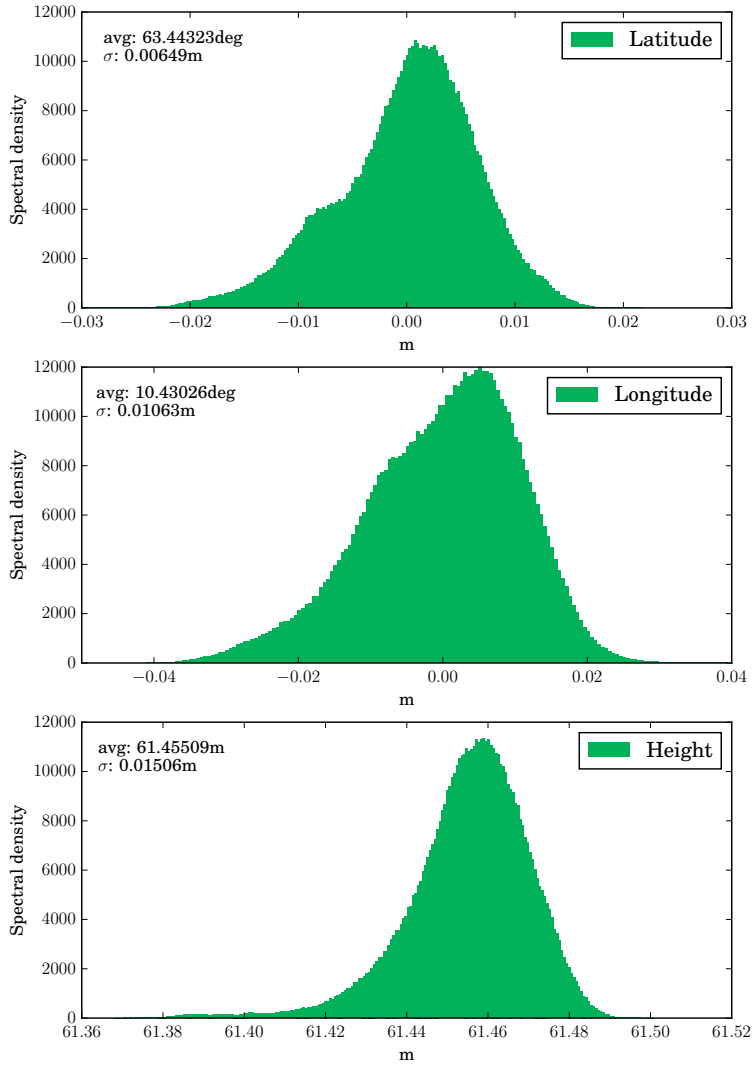


Figure 5.12: Heatmap from long term static test GNSS





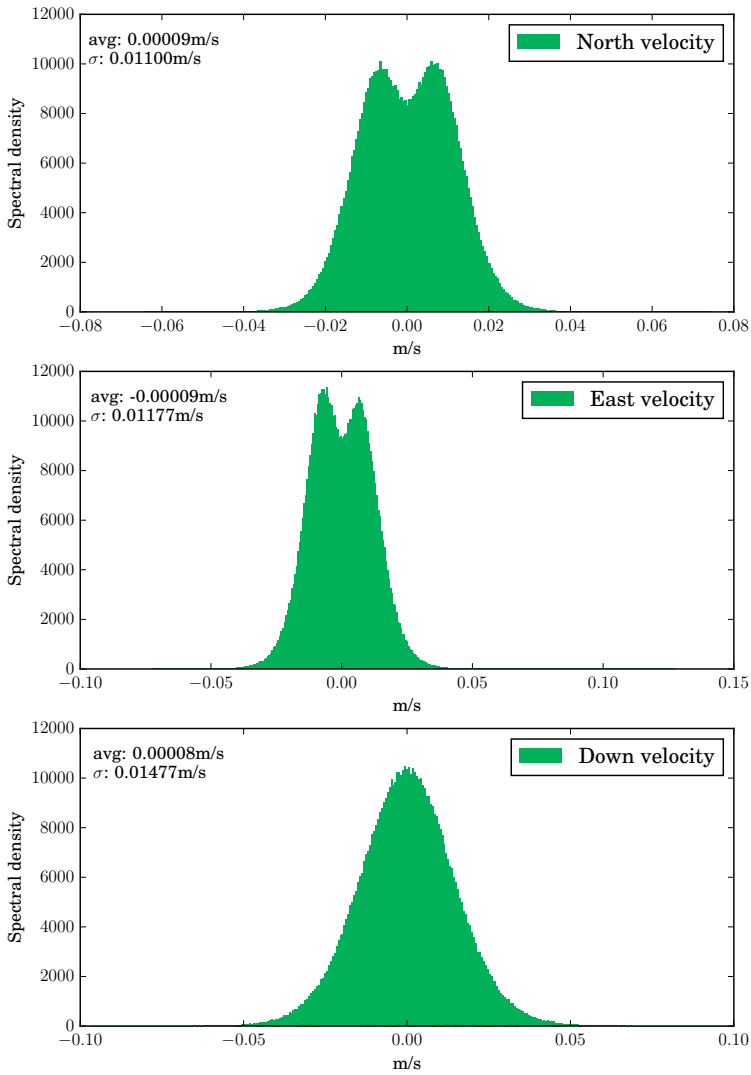
**Figure 5.13:** Histogram of position in long-term static test GNSS

The horizontal velocity is calculated from 'hor spd' and 'trk gnd' where

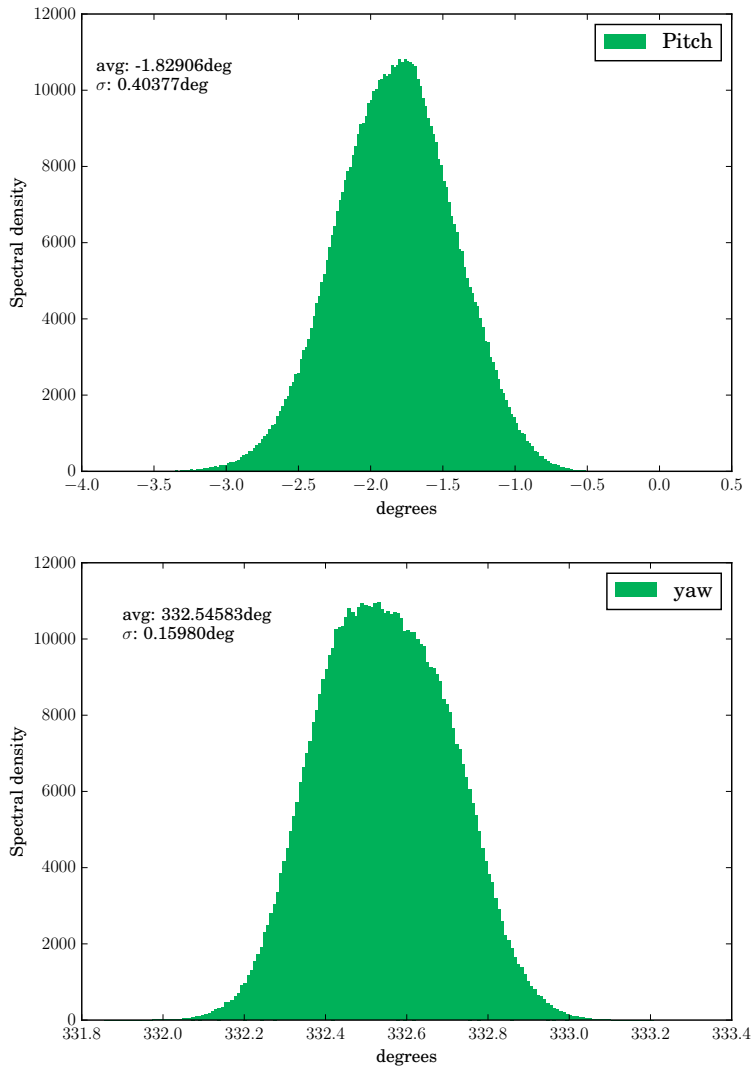
$$v_n = \sin(trkgnd)(horspd) \tag{5.3}$$

and

$$v_e = \cos(trkgnd)(horspd) \tag{5.4}$$



**Figure 5.14:** Histogram of velocity in long term static test GNSS



**Figure 5.15:** Histogram of attitude in long term static test GNSS

The histogram plots in Figure 5.13, 5.14 and 5.15 can be well fitted under a probability density function to underline that all are close to be Gaussians. This is a fundamental property, and a key reason why Kalman filters are computationally feasible.

Parameter	Standard deviation	Unit
Latitude	0.00649	m
Longitude	0.01063	m
Height	0.01506	m
North velocity	0.01100	m/s
East velocity	0.01177	m/s
Down velocity	0.01477	m/s
Pitch	0.40377	deg
Yaw	0.15980	deg

**Table 5.6:** Standard deviation of position, velocity and attitude during long-term static test.

As we can see from all the distribution plots of both position, velocity and attitude they are all close to Gaussians with an expected value:

$$E[X] = 0$$

and a standard deviation. Key figures from the long-term test can be found in Table 5.6.

## 5.5 Data integrity

Corrupt data may cause a degraded system. There are several approaches to the data integrity problem. Mainly a hardware and a software approach. The hardware part of data integrity has not been emphasized during the implementation, though common sense was used during placement of components. With the means of placing noisy components such as power supplies as far away as possible from sensitive transmission lines.

On the other hand, the software part has been laid much emphasis on. In the system, there are data integrity checks on both transport layer and application layer. On the transport layer, the STIM 300 supplies a Cyclic Redundancy Check (CRC). Both the NovAtel GNSS receiver and the STIM 300 IMU provides various kinds of status messages which makes the receiver able to filter on the message quality in software filters.

### 5.5.1 Cyclic Redundancy Check

Cyclic redundancy check (CRC) is an error detection technique that was invented in 1962 and is widely used in various forms in network and storage devices to detect accidental changes to raw data, i.e. bit errors. Protocols like TCP uses CRC checksums. The aim of an error detection technique is to make the receiver of a message transmitted over a noisy channel to determine whether the message has been corrupted during the transmission [34]. In order to do this, the transmitter constructs a value (called a checksum) that is a function of the message and appends it to the message. A CRC is called a n-bit CRC when

its checksum is  $n$  bits long. For each  $n$ -bit CRC, there are several options of checksum polynomial. CRC uses polynomial arithmetic for computing a checksum of the message being transmitted.

The STIM 300 provides a CRC checksum at the end of each datagram. This checksum is 32 bits, and the method has the name CRC-32/MPEG-2. This basically tells the receiver which polynomial and what conditions that is used to compute the checksum value. With all conditions right, it is trivial to the user how to check for data transmission errors.

Like the STIM 300, the NovAtel OEM617D also provides a 32-bit CRC for data verification. The NovAtel does not use the same polynomial for calculation, it uses one named CRC-32/ADCCP.

CRCs are not suitable for protecting against intentional data corruption. There is no authentication, a man-in-the-middle can easily modify the data and recompute the CRC. CRCs are also reversible functions. In closed systems like the one implemented in this thesis, these problems are not considered critical. Since it is easy applicable and computer cost-efficient, the method is suitable for detecting accidental changes to raw data.

### **5.5.2 Software filters**

In addition to Cycle Redundancy Check (CRC), the STIM 300 has a status byte for each sensor. So even though the checksum calculation is correct, received data might not be reliable. This makes it possible to filter out bad measurements of individual sensor from the IMU.

As mentioned earlier in this Chapter, NovAtel OEM617D provides ASCII messages for position and velocity updates. These messages contains various status flags explaining the message quality. A software filtration algorithm can be used to filter out bad messages.

## 5.6 Conclusion

In the end, software and hardware performed as intended, and the experiment could take place. We will in this Section go further into some lessons learned from building a hardware module and also the importance of a good analysis of sensors before they are integrated to higher-level applications.

### 5.6.1 Data integrity

The implementation of CRC in the receiver software seems to be important. When analyzing the 5 hour static test data set, a quick verification of how corrupt data could have been sent to the Kalman filter without a CRC could have looked was done. It showed angular rates beyond  $200deg/sec$  and accelerometer values exceeding  $10g$  when the unit was stationary. This kind of corrupt data could have caused trouble for the Kalman filter behaviour. During the static test, the density of bad CRC was 4 ppm.

### 5.6.2 Timetagging of IMU data

By making the main clock of microcontroller run at a frequency divisible by a higher baudrate, the reading and writing from USART is faster and ensure it to be less busy.

Sending timestamp in ASCII encoding makes the data package unnecessary large (16 bytes). If the timestamp had been sent binary coded as in unix time format with a floating part it could have been 6 bytes (32 bit unix time + 16 bit floating part).

### 5.6.3 Sensor analysis

Long-term static test of both NovAtel GNSS receiver and Sensoror IMU has proved that the manufacturers specification might not be correct. The static test of NovAtel OEM617D shows close to Gaussian distributed datasets with negligible offset and standard deviation withing valid range according to datasheet. During static test of STIM300 IMU, characteristics of the sensors was determined. Angular random walk, was found not to be within specified range according to datasheet specification. However, as the test setup and environment were not perfect no further conclusion can be made.

This makes a good foundation for integration and Kalman filter implementation using the the given GNSS receiver and IMU. Sensor characteristics derived for all sensors will be the base for the Kalman filter implementation and tuning.

# Chapter 6

## Experiment

During this Chapter a detailed description of how data was gathered, processed and analyzed will follow. The intention was to use the hardware and software developed in Chapter 5 to implement a uncoupled and loosely coupled Kalman filter.

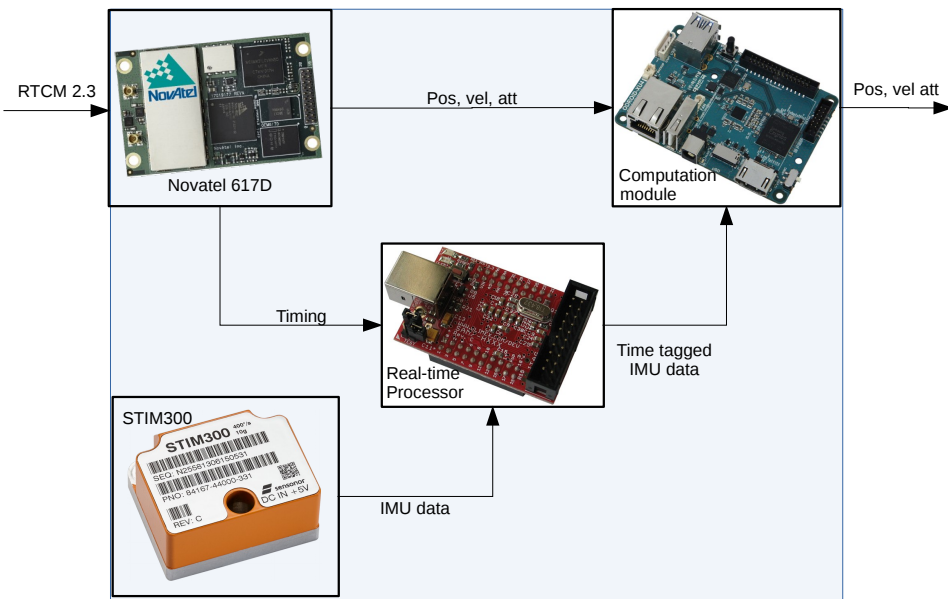
As a performance measure, a field test comparing the integrated system developed in this thesis with a commercial off-the-shelf (COTS) integrated system from Applanix has been conducted. The Applanix system has a good standing in marine section and is considered to be a good benchmark for performance tests.

### 6.1 Test setup

#### 6.1.1 Navigation Module

The Navigation Module described in Chapter 5 has been used for all the data gathering in this experiment. The flow of data is shown in Figure 6.1. Navigation Module with IMU inside was mounted on a rigid aluminum frame containing two GNSS antennas with a baseline of 1.35 meters. An illustration of the frame can be seen in 6.3.

The logging computer embedded in the Navigation Module used for data gathering from IMU and GNSS can be interfaced through ethernet. Logging is easily started and monitored over a SSH session.

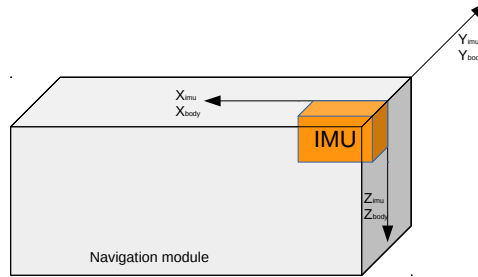


**Figure 6.1:** Data-flow between components.



### 6.1.2 Alignment of the IMU

The alignment of the IMU is shown in 6.3. The IMU frame and body frame is the same. The simple rotation matrix between IMU and BODY frame is shown in equation 6.1.



**Figure 6.2:** The rotation between IMU and BODY-frame

$$\mathbf{R}_{imu}^{body} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.1)$$

$$\boldsymbol{\omega}_{ib}^b = \mathbf{R}_{imu}^{body} \boldsymbol{\omega}_{imu} \quad (6.2)$$

The IMU measures the rotational velocity of the BODY frame relative to the ECI frame decomposed into the BODY frame.

The same procedure is done on the acceleration measurements.

$$\mathbf{f}_b = \mathbf{R}_{imu}^{body} \mathbf{f}_{imu} \quad (6.3)$$

### 6.1.3 Offsets

The position computed by the GNSS-receiver is computed at the primary antenna. The secondary antenna is for computing yaw and pitch. The transformation between the GNSS and IMU is given by Equation 6.4

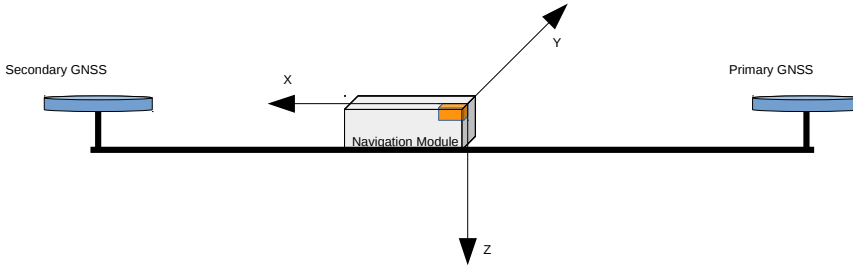


Figure 6.3: The transformation between GNSS and BODY-frame

$$\mathbf{T}_{IMU}^{GNSS} = \begin{bmatrix} 1 & 0 & 0 & 0.74 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.02 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

### 6.1.4 Translation between IMU and GNSS

#### Position

$$\mathbf{p}_{imu} = \mathbf{p}_{GNSS} + \begin{bmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{R}_n^b \begin{bmatrix} 0.74 \\ 0 \\ 0.02 \end{bmatrix} \quad (6.5)$$

#### Velocity

$$\mathbf{v}_{imu} = \mathbf{v}_{GNSS} + S(\omega) \begin{bmatrix} 0.74 \\ 0 \\ 0.02 \end{bmatrix} \quad (6.6)$$

Where  $S(\omega)$  is the skewmatrix of the measured angular velocity, found by using Euler derivative on the previous estimates.

#### Attitude

$$\mathbf{R}_{imu} = \mathbf{R}_{GNSS} \quad (6.7)$$

## 6.2 Determining variances

### 6.2.1 Determining process noise

The  $\mathbf{Q}$ -matrix is the matrix for the dynamic process noise and reflects how much noise from the IMU is affecting the process. Statistical data gathered makes a good foundation for determine those values. The elements of the spectral density is derived in Chapter 5 and can be found in Table 5.5 and Table 5.4. Be aware that the filter is using other units.

$$\mathbf{Q}_k = \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & q_a & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & q_g & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & q_{ba} & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & q_{bg} \end{bmatrix} \quad (6.8)$$

$$\mathbf{q}_a = \begin{bmatrix} 0.0012^2 & 0 & 0 \\ 0 & 0.001^2 & 0 \\ 0 & 0 & 0.001^2 \end{bmatrix} \quad (6.9)$$

$$\mathbf{q}_g = \begin{bmatrix} (4.36 \times 10^{-5})^2 & 0 & 0 \\ 0 & (4.36 \times 10^{-5})^2 & 0 \\ 0 & 0 & (4.84 \times 10^{-5})^2 \end{bmatrix} \quad (6.10)$$

The spectral density of the Gauss-Markov process is computed using equation 6.11.

$$q_{bai} = \frac{2\sigma_{bai}^2}{\tau_{bai}}$$

$$q_{bgi} = \frac{2\sigma_{bgi}^2}{\tau_{bgi}} \quad (6.11)$$

$$\mathbf{q}_{ba} = \begin{bmatrix} \frac{2(4 \times 10^{-4})^2}{100} & 0 & 0 \\ 0 & \frac{2(4 \times 10^{-4})^2}{100} & 0 \\ 0 & 0 & \frac{2(4 \times 10^{-4})^2}{100} \end{bmatrix} \quad (6.12)$$

$$\mathbf{q}_{bg} = \begin{bmatrix} \frac{2(8.24 \times 10^{-6})^2}{200} & 0 & 0 \\ 0 & \frac{2(6.30 \times 10^{-6})^2}{300} & 0 \\ 0 & 0 & \frac{2(4.85 \times 10^{-6})^2}{500} \end{bmatrix} \quad (6.13)$$

Setting values in  $\mathbf{Q}$ -matrix is a crucial step when implementing a Kalman filter. If  $\mathbf{Q}$  is too small then the filter will be overconfident in its prediction model, and will diverge from the actual solution. If it is too large then the filter will be unduly influenced by the noise in the measurements and perform sub-optimally.

## 6.2.2 Initial process variance

The initial process variance is determined by characteristics given in datasheets, long term static tests and manual tuning on a variety of datasets.

$$P_0 = \begin{bmatrix} \sigma_{\delta\mathbf{r}}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \sigma_{\delta\mathbf{v}}^2 & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\delta\mathbf{e}}^2 & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\delta\mathbf{b}_a}^2 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \sigma_{\delta\mathbf{b}_g}^2 \end{bmatrix} \quad (6.14)$$

$P_0$  is assumed diagonal for lack of sufficient statistical information to evaluate its off-diagonal elements.

- $\sigma_{\delta\mathbf{r}}$   
Set from the reported standard deviation of the first position estimate from the GNSS receiver. Typical accuracy is 0.01 to 0.02 meters.
- $\sigma_{\delta\mathbf{v}}$   
Set to  $0.03m/s$ , according to expected accuracy [24].
- $\sigma_{\delta\mathbf{e}}$   
Roll is set to  $0.8^\circ$ .  
Pitch and yaw is set from the reported standard deviation of the first yaw estimate from the GNSS receiver. Typical accuracy for pitch is  $0.6^\circ$  and  $0.4^\circ$  for yaw.
- $\sigma_{\delta\mathbf{b}_a}$   
Set to  $0.05m/s^2$ , which is expected bias from the IMU [35]. It is about 4 times higher than specified in [26]
- $\sigma_{\delta\mathbf{b}_g}$   
Set to  $0.0012rad/sec$ , which is specified max bias from the IMU [26]. It is about 2-3 times higher than measured in [35].

### 6.2.3 Measurement noise matrix

It is required that the measurement noise matrix is updated according to the standard deviation reported from the GNSS receiver for each measurement. Standard deviation of the position and attitude is reported continuously by the GNSS receiver. The velocity standard deviation is not reported continuously by the receiver, but is set to expected value  $0.03m/s$  according to [24].  $0.03m/s$  is a larger value than obtained during the long-term static test of GNSS velocity in Chapter 5. Since we do not have any statistical data from GNSS receiver during movement, the value from NovAtel datasheet is used.

$$\mathbf{R} = \begin{bmatrix} r_p & 0_{3 \times 3} & 0_{3 \times 2} \\ 0_{3 \times 3} & r_v & 0_{3 \times 2} \\ 0_{2 \times 3} & 0_{2 \times 3} & r_a \end{bmatrix} \quad (6.15)$$

$$\mathbf{r}_p = \begin{bmatrix} \sigma_\mu^2 & 0 & 0 \\ 0 & \sigma_l^2 & 0 \\ 0 & 0 & \sigma_h^2 \end{bmatrix} \quad (6.16)$$

$$\mathbf{r}_p = \begin{bmatrix} \sigma_\mu^2 & 0 & 0 \\ 0 & \sigma_l^2 & 0 \\ 0 & 0 & \sigma_h^2 \end{bmatrix} \quad (6.17)$$

$$\mathbf{r}_a = \begin{bmatrix} \sigma_\psi^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \quad (6.18)$$

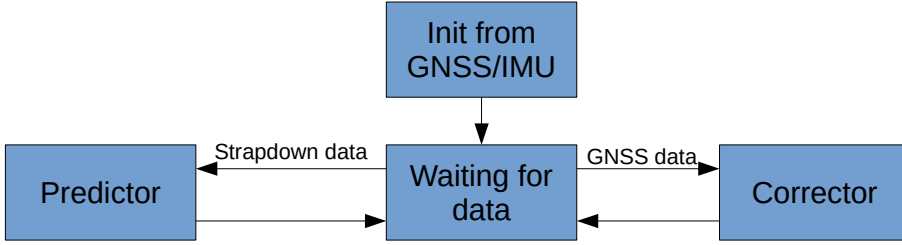
### 6.2.4 Measurement mapping matrix

The  $\mathbf{H}$ -matrix indicates which states that are observable through measurements provided, in this case, by the GNSS receiver. This matrix shows that position, velocity, pitch and yaw is observable through measurements.

$$\mathbf{H} = \begin{bmatrix} I_3 & 0_3 & 0 & 0 & 0 & 0_6 \\ 0_3 & I_3 & 0 & 0 & 0 & 0_6 \\ 0_3 & 0_3 & 0 & 1 & 0 & 0_6 \\ 0_3 & 0_3 & 0 & 0 & 1 & 0_6 \end{bmatrix} \quad (6.19)$$

## 6.3 Kalman filter implementation

This Chapter shows how a discrete extended Kalman filter is integrated to the hardware and implemented. A mathematical description for a discrete-time extended Kalman filter is shown in Chapter 4.2, and an equation overview can be found in 4.1. The intention of this Section is to show the integration of the GNSS receiver and the IMU to those equations. In figure 6.4 the filter flow is shown in form of a Kalman filter state machine.



**Figure 6.4:** State machine of a discrete extended Kalman filter.

### 6.3.1 State vector

The state vector consists of nine navigation parameters and six sensor bias parameters. For the navigation parameters we have three position errors, three velocity errors and three attitude errors. For the sensor bias parameters we have three gyro bias estimates and three accelerometer bias estimates. The complete state vector is given by 6.20.

$$\mathbf{x} = [\delta\mathbf{r} \quad \delta\mathbf{v} \quad \delta\mathbf{e} \quad \mathbf{b}_a \quad \mathbf{b}_g]^\top \quad (6.20)$$

where

$$\delta\mathbf{r} = [\delta r_N \quad \delta r_E \quad \delta r_D]^\top \quad (6.21)$$

$$\delta\mathbf{v} = [\delta v_N \quad \delta v_E \quad \delta v_D]^\top \quad (6.22)$$

$$\delta\mathbf{e} = [\delta\phi \quad \delta\theta \quad \delta\psi]^\top \quad (6.23)$$

$$\mathbf{b}_a = [b_{ax} \quad b_{ay} \quad b_{az}]^\top \quad (6.24)$$

$$\mathbf{b}_g = [b_{gx} \quad b_{gy} \quad b_{gz}]^\top \quad (6.25)$$

is position, velocity and attitude error followed by accelerometer and gyroscope biases respectively. Attitude is given in the well-known Euler angles.

### 6.3.2 Initialization

Initialization of the filter is important for future filter performance. The initial state as well as the initial belief in the state needs to be computed. A bad initial estimate may cause filter divergence or sub-optimal performance.

The strapdown equation is initialized with the initial estimate from the GNSS. Initial roll is estimated by assuming that the initial force contains only gravity.

The filter state estimate  $\hat{\mathbf{x}}$  is initially modelled as:

$$\hat{\mathbf{x}} = [0_{15 \times 1}] \quad (6.26)$$

The process noise matrix  $\mathbf{Q}$  is from Equation 6.8, the measurement mapping matrix in  $\mathbf{H}$  is stated in Equation 6.19.  $\mathbf{P}_0$  is initial covariance matrix and its initialization is showed in Equation 6.14.

### 6.3.3 Prediction

When new IMU-data is received, a new prediction of the state estimate and the estimated covariance is done in the Kalman filter. This is done by applying the propagation equations in 4.1:

$$\bar{\mathbf{x}}(k+1) = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}(k), k) \quad (6.27)$$

$$\Phi(k) = \mathbf{I} + \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}(k), k) \quad (6.28)$$

$$\bar{\mathbf{P}}(k+1) = \Phi(k)\hat{\mathbf{P}}\Phi^T(k) + \mathbf{Q}(k) \quad (6.29)$$

### 6.3.4 Correction

GNSS receivers usually transmit PVT data at a lower rate than IMUs.

By applying the update equations in 4.1, a new Kalman gain and a correction of the state estimate and the estimated covariances is being calculated based on a GNSS measurement update with its associated variances. This step is done when GNSS-data is received.

$$\mathbf{S}(k) = \mathbf{H}\bar{\mathbf{P}}(k)\mathbf{H}^T + \mathbf{R}(k) \quad (6.30)$$

$$\mathbf{K}(k) = \bar{\mathbf{P}}(k)\mathbf{H}^T\mathbf{S}(k)^{-1} \quad (6.31)$$

$$\hat{\mathbf{P}}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}]\bar{\mathbf{P}}(k)[\mathbf{I} - \mathbf{K}(k)\mathbf{H}]^T + \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}^T(k) \quad (6.32)$$

$$\mathbf{z}(k) = \mathbf{ss}(k) - \mathbf{y}(k) \quad (6.33)$$

where  $\mathbf{ss}$  is the current state of the strapdown equations.  $\mathbf{y}(k)$  is the GNSS measurements. Then we have the new state estimate by:

$$\hat{\mathbf{x}}(k) = \mathbf{x} + \mathbf{K}(k)[\mathbf{z}(k) - \mathbf{x}(k)\mathbf{H}] \quad (6.34)$$

### 6.3.5 Error dynamics

Process error dynamics is given by the  $\mathbf{F}$  matrix. This is the model where all state errors are propagated. It is a somehow complex matrix, so it has been split in to sub matrices for clearer presentation. The derivation of the error dynamics equation is shown as written in [27].

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{rr} & \mathbf{F}_{rv} & 0 & 0 & 0 \\ \mathbf{F}_{vr} & \mathbf{F}_{vv} & \mathbf{F}_{ve} & \mathbf{R}_n^b & 0 \\ \mathbf{F}_{er} & \mathbf{F}_{ev} & \mathbf{F}_{ee} & 0 & \mathbf{R}_n^b \\ 0 & 0 & 0 & \dot{\mathbf{b}}_a & 0 \\ 0 & 0 & 0 & 0 & \dot{\mathbf{b}}_g \end{bmatrix} \quad (6.35)$$

Submatrices in  $\mathbf{F}$  are listed below:

$$\mathbf{F}_{rr} = \begin{bmatrix} 0 & 0 & \frac{-v_N}{(M+h)^2} \\ \frac{v_E \sin(\varphi)}{(N+h)\cos^2(\varphi)} & 0 & \frac{-v_E}{(N+h)^2 \cos(\varphi)} \\ 0 & 0 & 0 \end{bmatrix} \quad (6.36)$$

$$\mathbf{F}_{rv} = \begin{bmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos(\varphi)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (6.37)$$

$$\mathbf{F}_{vr} = \begin{bmatrix} -2v_E\omega_e \cos(\varphi) - \frac{v_E^2}{(N+h)\cos^2(\varphi)} & 0 & \frac{-v_N v_D}{(M+h)^2} + \frac{v_E^2 \tan(\varphi)}{(N+h)^2} \\ 2\omega_e(v_N \cos(\varphi) - v_D \sin(\varphi)) + \frac{v_N v_E}{(N+h)\cos^2(\varphi)} & 0 & \frac{-v_E v_D}{(N+h)^2} - \frac{v_N \tan(\varphi)}{(N+h)^2} \\ 2v_E\omega_e \sin(\varphi) & 0 & \frac{v_E^2}{(N+h)^2} + \frac{v_N^2}{(M+h)^2} - \frac{2G}{R-h} \end{bmatrix} \quad (6.38)$$



$$\mathbf{F}_{\mathbf{v}\mathbf{v}} = \begin{bmatrix} \frac{v_D}{M+h} & -2\omega_e \sin(\varphi) - \frac{2v_E \tan(\varphi)}{N+h} & \frac{v_N}{M+h} \\ 2\omega_e \sin(\varphi) + \frac{v_E \tan(\varphi)}{N+h} & \frac{v_D + v_N \tan(\varphi)}{N+h} & 2\omega_e \cos(\varphi) + \frac{v_E}{N+h} \\ \frac{-2v_N}{M+h} & -2\omega_e \cos(\varphi) - \frac{2v_E}{N+h} & 0 \end{bmatrix} \quad (6.39)$$

$$\mathbf{F}_{\mathbf{v}\mathbf{e}} = \text{skew}(\mathbf{f}) \quad (6.40)$$

$$\mathbf{F}_{\mathbf{e}\mathbf{r}} = \begin{bmatrix} -\omega_e \sin(\varphi) & 0 & \frac{-v_E}{(N+h)^2} \\ 0 & 0 & \frac{v_N}{(M+h)^2} \\ -\omega_e \cos(\varphi) - \frac{v_E}{(N+h)\cos^2(\varphi)} & 0 & \frac{v_E \tan(\varphi)}{(N+h)^2} \end{bmatrix} \quad (6.41)$$

$$\mathbf{F}_{\mathbf{e}\mathbf{v}} = \begin{bmatrix} 0 & \frac{1}{N+h} & 0 \\ \frac{-1}{M+h} & 0 & 0 \\ 0 & \frac{-\tan(\varphi)}{N+h} & 0 \end{bmatrix} \quad (6.42)$$

$$\mathbf{F}_{\mathbf{e}\mathbf{e}} = \begin{bmatrix} 0 & -\omega_e \sin(\varphi) - \frac{v_E \tan(\varphi)}{N+h} & \frac{v_N}{M+h} \\ \omega_e \sin(\varphi) + \frac{v_E \tan(\varphi)}{N+h} & 0 & \omega_e \cos(\varphi) + \frac{v_E}{N+h} \\ \frac{-v_N}{M+h} & -\omega_e \cos(\varphi) - \frac{v_E}{N+h} & 0 \end{bmatrix} \quad (6.43)$$

$$\mathbf{R}_{\mathbf{n}}^{\mathbf{b}} = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)c(\theta) & -s(\theta) \\ -s(\psi)c(\varphi) + c(\psi)s(\theta)s(\varphi) & c(\psi)c(\varphi) + s(\psi)s(\theta)s(\varphi) & c(\theta)s(\varphi) \\ s(\psi)s(\varphi) + c(\psi)s(\theta)c(\varphi) & -c(\psi)s(\varphi) + s(\psi)s(\theta)c(\varphi) & c(\theta)c(\varphi) \end{bmatrix} \quad (6.44)$$

$$\dot{\mathbf{b}}_{\mathbf{a}} = \begin{bmatrix} \frac{1}{\tau_{ba_x}} & 0 & 0 \\ 0 & \frac{1}{\tau_{ba_y}} & 0 \\ 0 & 0 & \frac{1}{\tau_{ba_z}} \end{bmatrix} \quad (6.45)$$

$$\dot{\mathbf{b}}_{\mathbf{g}} = \begin{bmatrix} \frac{1}{\tau_{bg_x}} & 0 & 0 \\ 0 & \frac{1}{\tau_{bg_y}} & 0 \\ 0 & 0 & \frac{1}{\tau_{bg_z}} \end{bmatrix} \quad (6.46)$$

where  $N$  is given by:

$$N = \frac{6378137(1 - 0.0818^2)}{((1 - 0.0818^2)\sin^2(\varphi))^{1.5}} \quad (6.47)$$

and is the radius of curvature in prime vertical. The meridian radius of the curvature in prime horizontal is given by  $M$ :

$$M = \frac{6378137}{((1 - 0.0818^2)\sin^2(\varphi))^{0.5}} \quad (6.48)$$

and is used to translate from latitude and longitude from meters to radians.

Earth rotation is denoted by  $\omega_e$ . It is assumed to be constant with the value:

$$\omega_e = 7.292115 \times 10^{-5} \text{ rad/sec} \quad (6.49)$$

For gravitation model, the following is used:

$$G = g_d \left( \frac{R}{R + v_N} \right)^2 \quad (6.50)$$

where:

$$R = \sqrt{NM} \quad (6.51)$$

is the Gaussian mean Earth radius of curvature[27], and  $g_d$  is the normal gravity at  $h = 0$ . This given by the STIM300 datasheet to be  $9.80665 \frac{m}{s^2}$ .

## 6.4 Data gathering

The rigid frame shown in Figure 6.3 was mounted to a inflatable boat with a small motor, and the equipment were powered using batteries. In order to get some realistic challenging situations a tour up and down the river Nidelva has been done in addition to a roundtrip on the fjord. Due to several bridge passes and narrow straits in the river, the GNSS coverage were all from poor to absent. This will be presented in later sections. The tour taken up the river can be seen in Figure 6.5.

When sailing on the fjord, there were quite some more dynamics with rapid manouvering and some mild seas. Datasets from the fjord has been used with most of the filter performance analysis.

The results from different setups and filters are usually presented as graphs comparing two or more systems. To avoid confusion, graph colors for the respective systems are consistent. The colors used are found in Table 6.1. Also, in some cases there is a colored area around the graph. This area indicates one standard deviaton from the result.

In the following chapters, different cases will be presented. A uncoupled Kalman filter implementation will first be presented, followed by a loosely coupled filter. Finally, a head-to-head performance analysis with the Applanix Wavemaster II is presented.



**Figure 6.5:** Overview of route up the river Nidelva

System	Graph color
Kalman filter	Blue
NovAtel GNSS receiver	Green
IMU strapdown equations	Orange
Applanix Wavemaster II	Red

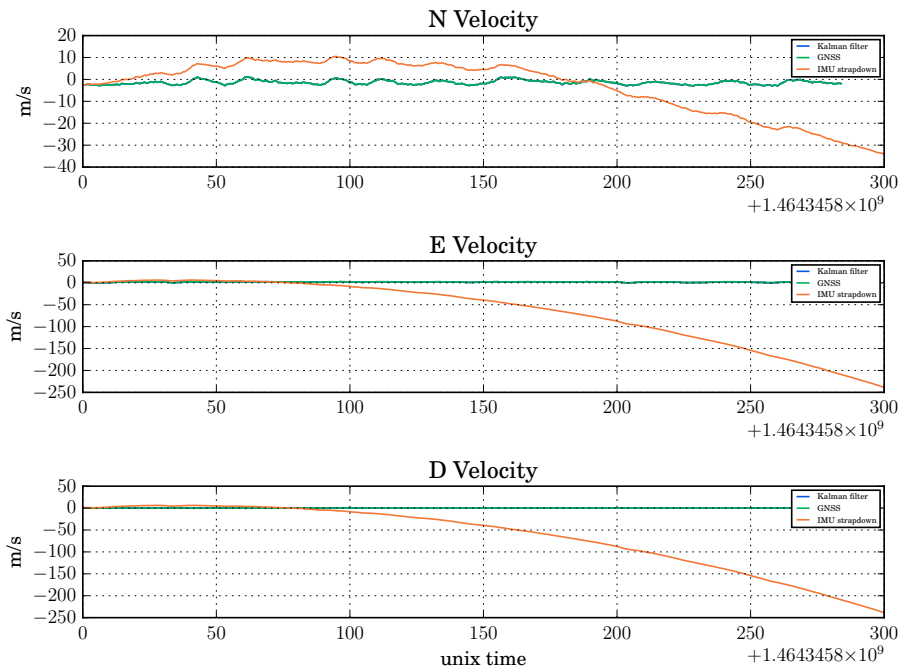
**Table 6.1:** Graph colors overview

## 6.5 Uncoupled Kalman filter

This Section describes how the uncoupled filter was implemented, results and discussion.

### 6.5.1 Results

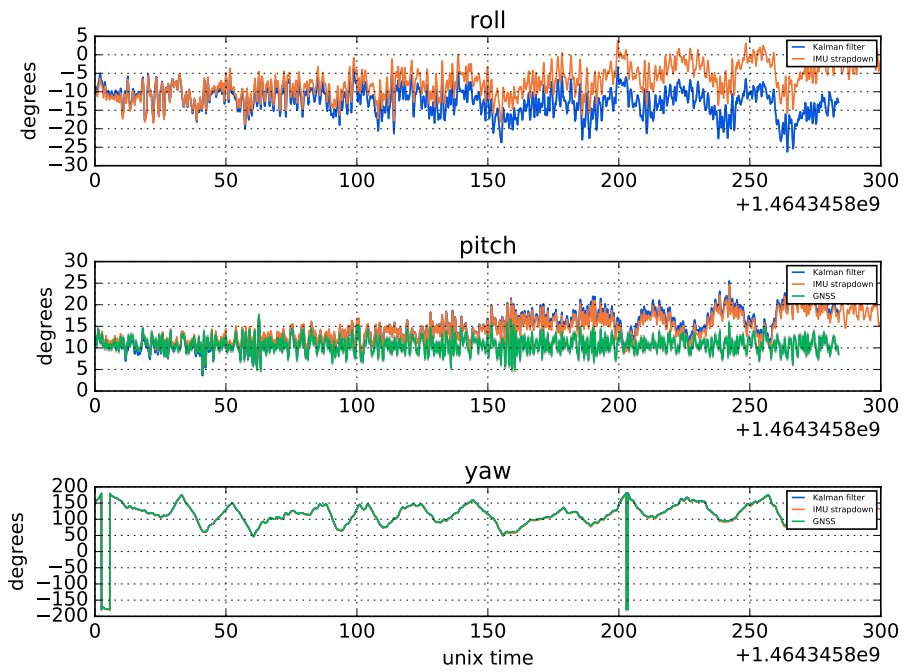
The filter is able to follow the GNSS receiver on position and velocity. However, as the IMU strapdown equation drifts of at the speed of sound after less than 300 seconds the error is nonlinear and is hard to keep track of. Changes in attitude is especially hard to keep track of, as the IMU strapdowns frame differs from the real frame.



**Figure 6.6:** Uncoupled filter velocity

### 6.5.2 Discussion

Letting the IMU strapdown equation drift off is unstable, and a feedback from the Kalman filter should be implemented. This can be done by changing coupling scheme to a loosely coupled scheme, which introduces a feedback loop. In the next Section this will be done.

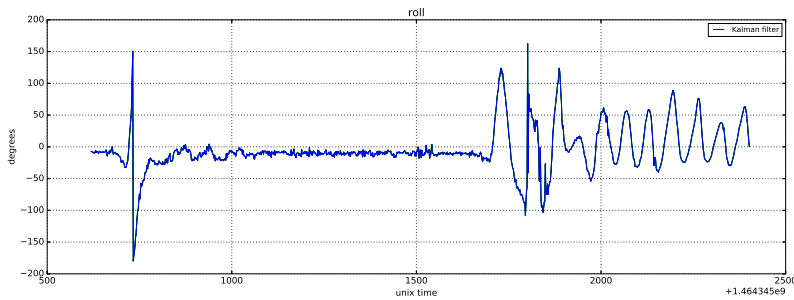
**Figure 6.7:** Uncoupled filter attitude

## 6.6 Loosely coupled Kalman filter

The uncoupled filter from previous Section has been extended to contain a feedback loop for the IMU navigation equations. The strapdown state is updated by the corrector, and the error estimate is set to zero. The bias estimate is kept in the state matrix and runs in an open loop.

### 6.6.1 Tuning

When using the tuning parameters from Chapter 6.2, the filter is unstable. See Figure 6.8



**Figure 6.8:** Unstable roll estimate

This is a divergence problem, and they are commonly caused by round-off errors, gross modelling errors or observability problems[14].

Observability of INS with GNSS multi-antenna system is proven to be observable[28] and is not considered an issue.

The Bayes method shown in Equation 6.32 will always produce symmetric results and is the most important way of avoiding divergence due to computational frailties [14]. All values in the Kalman filter is computed using `numpy.float64`, which is double precision float: sign bit, 11 bits exponent, 52 bits mantissa. It ensures minimal round-off errors.

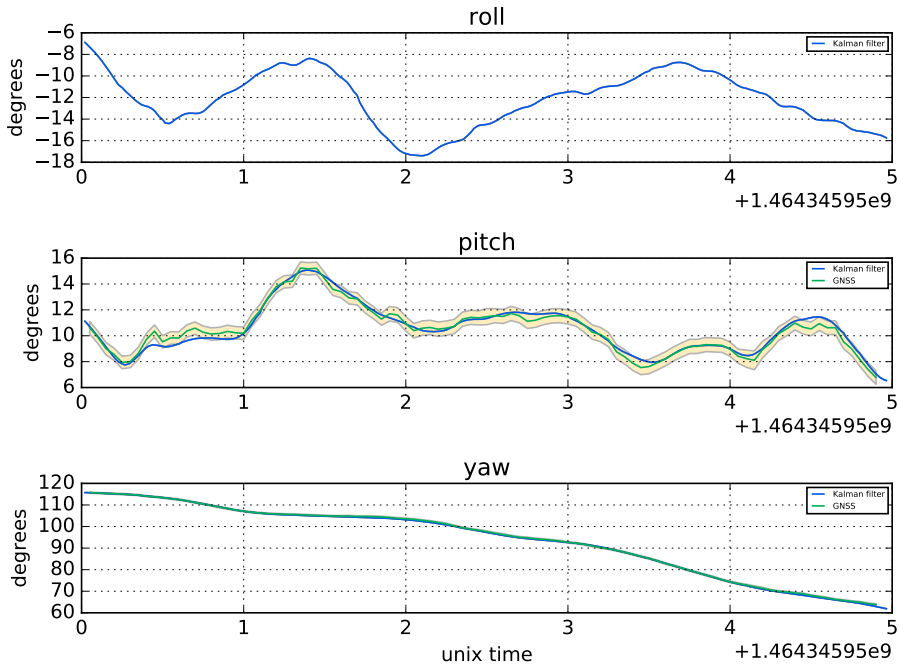
Gross modelling errors may cause divergence because of inaccurate modeling of the process being estimated[14]. This occurs when the process does not behave as the Kalman filter is "told". The obvious solution for this type of divergence problem is always to insert some process noise. It can cause a suboptimal filter, but makes a much safer filter than otherwise.

By increasing  $\mathbf{q}_a$  and  $\mathbf{q}_g$  to  $2\sigma$  the Kalman filter became stable. Further tuning of  $\mathbf{R}$  and  $\mathbf{P}_0$  was also tried without satisfying results.

### 6.6.2 Results

A Kalman filter can be considered good if input states and filter states are within eachothers field of variance. As seen in Figure 6.9 and 6.10, the Kalman filtered solution is a smoothed

version of the GNSS-measurements. The GNSS attitude is plotted with variance. The Kalman filtered pitch moves very delicately within the GNSS variance.



**Figure 6.9:** Loosely coupled filter attitude

As described in Chapter 6.2.2, biases for the accelerometers are expected to be within  $0.05m/s^2$  and  $0.0012rad/sec = 250deg/h$  for the gyros. All estimated biases are within the expected values, Figure 6.11.

### 6.6.3 Discussion

The loosely coupled Kalman filter needs careful tuning to become optimal and prevent divergence. When stable, state estimates are within the expected values.

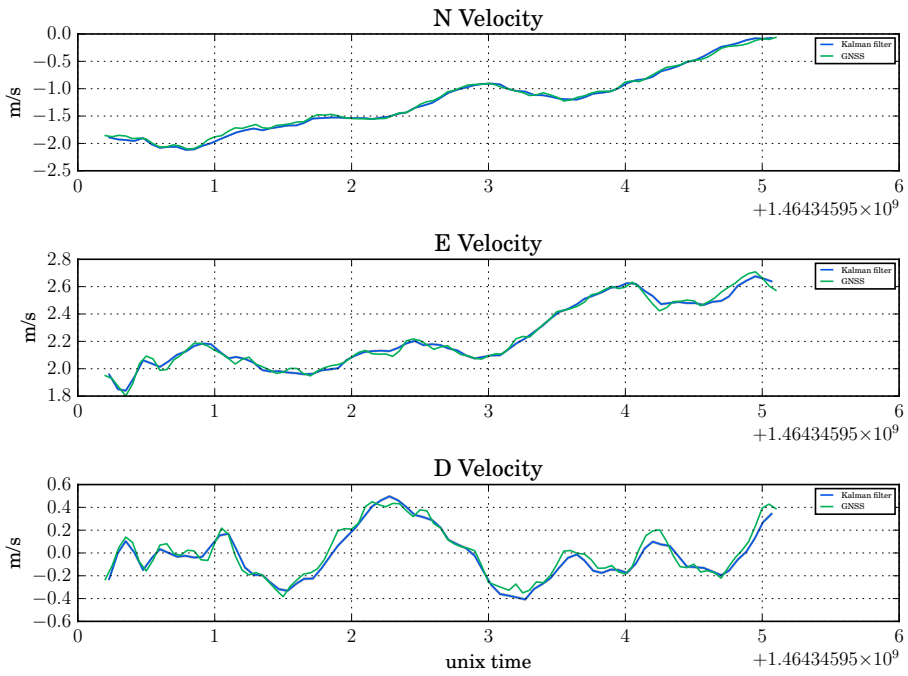


Figure 6.10: Loosely coupled filter velocity

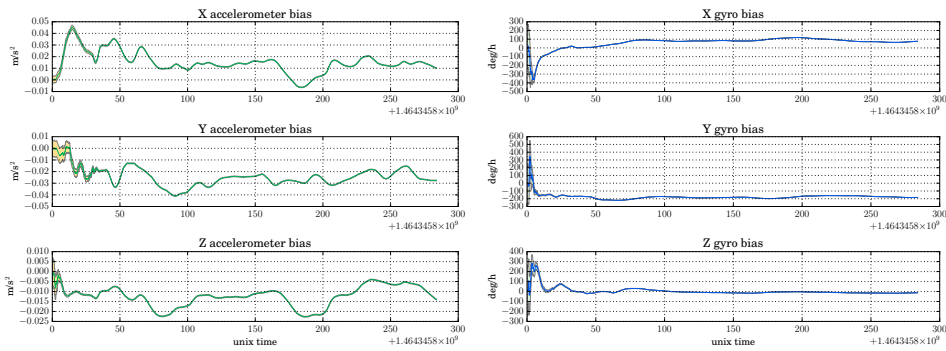


Figure 6.11: Estimated biases in the tuned filter



## 6.7 Head-to-head performance analysis

We had the opportunity to do some head-to-head tests with an Applanix integrated system. The system is a Applanix Wavemaster II integrated into Norbit Subsea's mechanical solution. This is a high-end system with dual-band dual-frequency GNSS, and it consists of an IMU with MEMS based gyros. Two different test approaches were used. One under circumstances with a good coverage of satellites. One where the unit was run under a bridge, for bad GNSS coverage. As mentioned before, several GNSS outages were obtained under bridges.

### 6.7.1 Applanix Wavemaster II

Some performance key figures can be seen in Table 6.2. According to the Applanix manual, the integrated solution can run both tightly coupled and loosely coupled mode. The POS MV software will automatically switch between the tightly coupled and loosely coupled algorithms to ensure maximum performance. During the tests performed in this experiment a loosely coupled solution with RTK was used in the Applanix POS MV software.

	<b>RMS</b>	<b>Unit</b>
Single point L1/L2 position accuracy	1.2	m
DGPS position accuracy	0.4	m
Yaw accuracy 2m baseline	0.08	°
Velocity accuracy	0.03	m/s

**Table 6.2:** Applanix Wavemaster II performance key figures.

The integrated system used for reference, Applanix Wavemaster II, comes with a software solution for configuring and monitoring the system. The software is called POS MV (**P**osition and **O**rientation **S**ystem for **M**arine **V**essels) as mentioned above.

POS MV has a variety of logging possibilities, both binary and ASCII encoded NMEA sentences. Binary logging format was chosen in lack of sufficient information regarding the accuracy and precision of the system in NMEA sentences. Binary logging also allows for a higher sampling rate from POS MV software. Data was logged at 100 Hz in this experiment.

### 6.7.2 Hardware setup

In this head-to-head test, both integrated systems were mounted on the same rigid aluminum frame as mentioned earlier in Section 6.4. The complete test setup can be seen to the left in Figure 6.12.

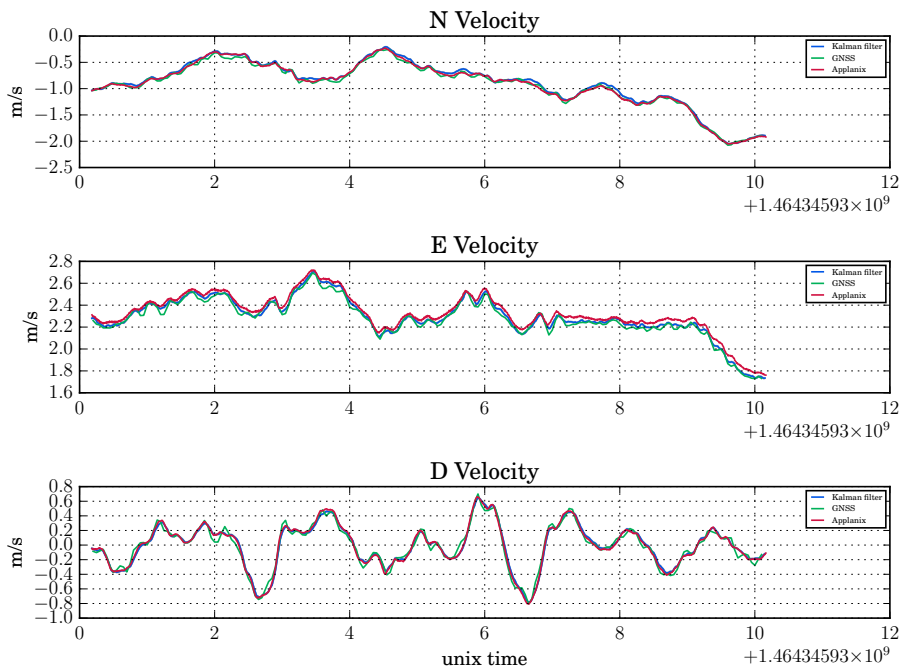
Only one set of antennas were used, so no transformation between the two systems had to be done. The signal from the GNSS antennas was split, so both receivers had the same signal. A DC block was installed on one of the receivers to avoid both receivers powering the antennas. Pictures of this solution can be seen in 6.12.



**Figure 6.12:** To the left, both system are mounted on same rigid frame. To the right a DC block to ensure only one system powers the antennas.

### 6.7.3 Results

In this Chapter, two plots of the Kalman filtered solution and the Applanix solution are compared. The position is not very interesting, as it is hard to see the difference between the solutions. The RTK-solution is very accurate and the position estimates are assumed to be beyond cm accuracy. The velocity comparison in Figure 6.13 shows that velocity is very well estimated. The accuracy of the GNSS receivers doppler measurements are very accurate and corrects the solution if needed. In the attitude plot, Figure 6.14 it can be seen that the Kalman filtered solutions roll is approximately  $0.5^\circ$  below the Applanix. It is probably due to misalignment in the mounting. The filter is able to cancel out the forces by estimating a negative bias on the y-axis accelerometer, as seen in Figure 6.11

**Figure 6.13:** Velocity compared

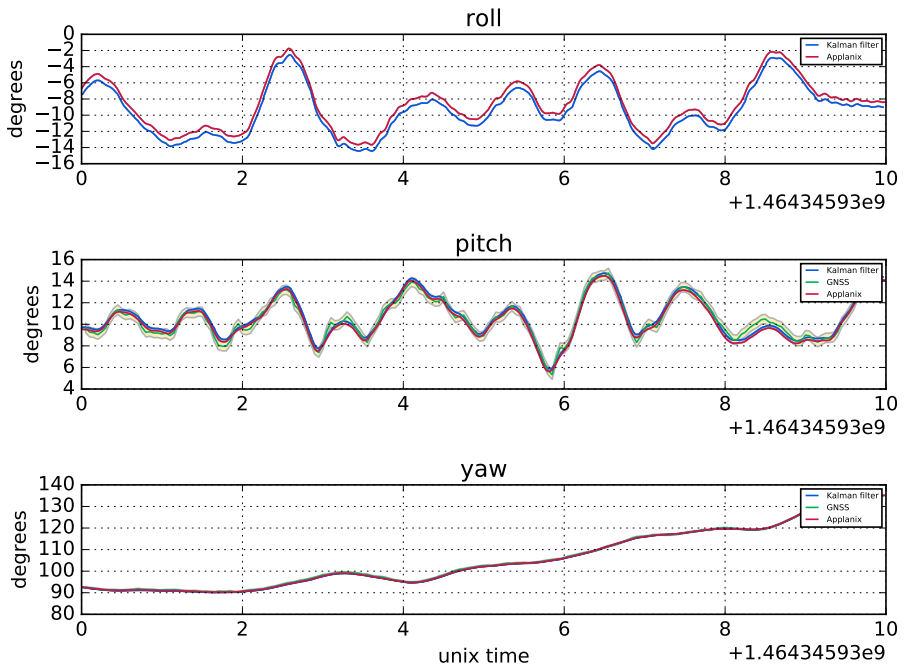


Figure 6.14: Attitude Compared.

### 6.7.4 GNSS outage

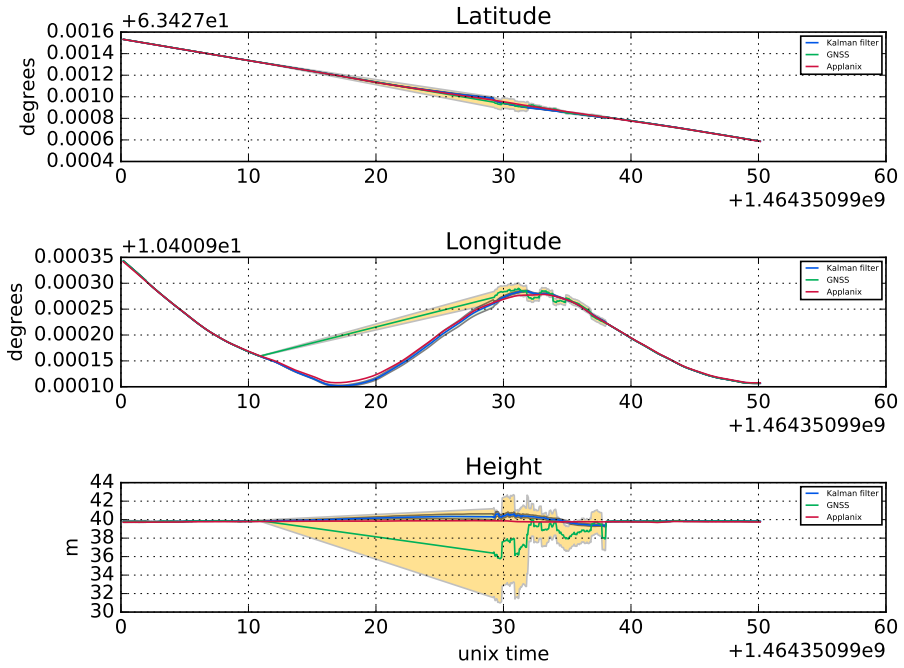
When sailing under Gamle bybro the GNSS coverage was lost for about 20 seconds. In this Chapter the filters accuracy during outage was compared to the Applanix. The solutions are so close to each other that it is hard to tell them apart. The biggest difference can be seen in the North velocity and indicates that the Kalman filter position is about to drift away. The GNSS coverage reappeared back before the difference is visible in the attached plots. In Figure 6.18 .kml files were generated and uploaded to a uMap which is a track drawing web site using OpenStreetMap.



**Figure 6.15:** Gamle bybro in Nidelva.

### 6.7.5 Discussion

The head-to-head test with the Applanix proves that the Kalman filter estimates states and biases correctly. The concept has been proven to work, and no further analysis is going to be done in this thesis.



**Figure 6.16:** Position during GNSS outage.

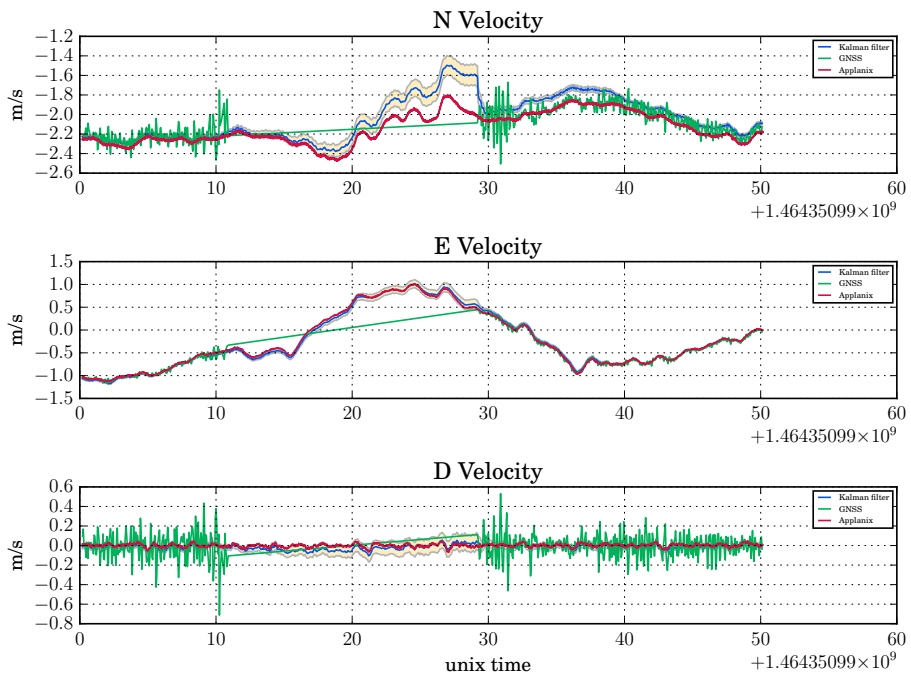
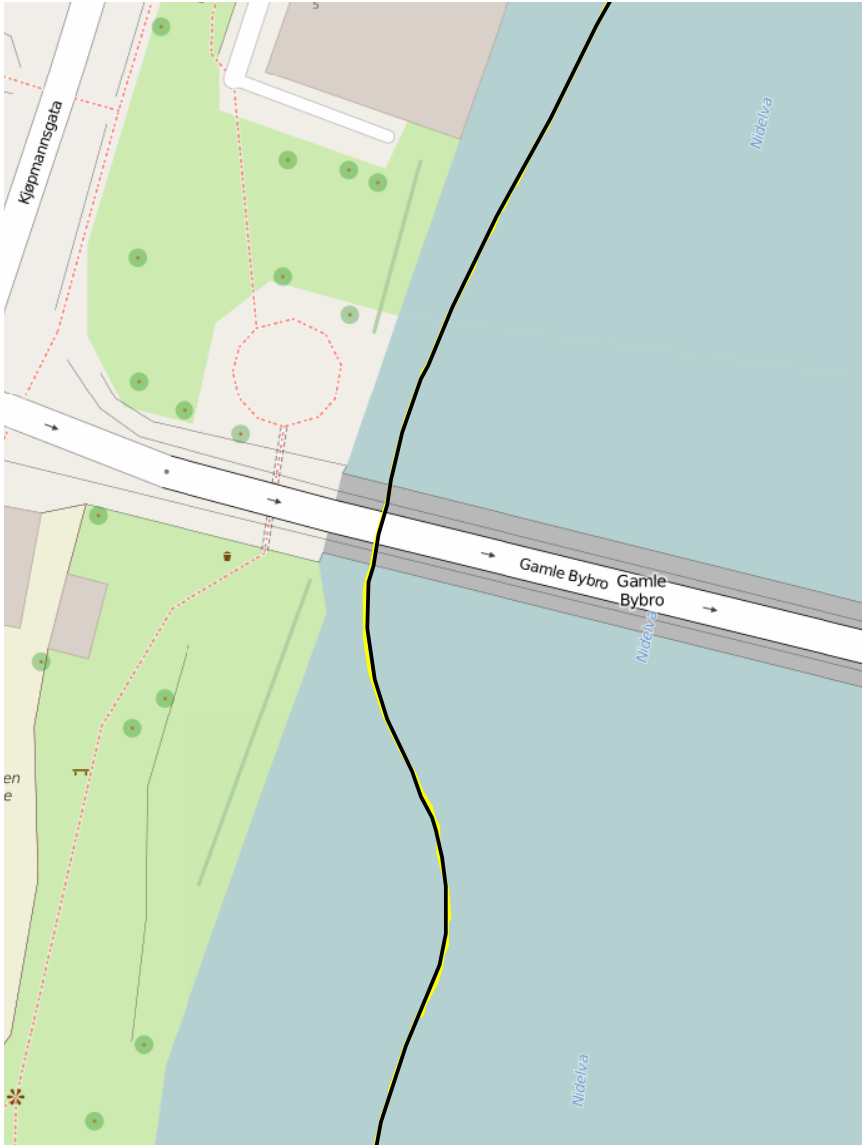


Figure 6.17: Velocity during GNSS outage.



**Figure 6.18:** Position plotted in map. Black is Applanix and yellow is Kalman filtered solution



# Discussion

## 7.1 Findings in theory

The task was to integrate GNSS and INS for robust positioning. Obviously, this is a significant task, and needs a good amount of theory in basis. There has been written about GNSS and INS separately, and also a chapter about integration. Importance of precise and accurate GNSS position has been studied, and it has been shown that GNSS aided attitude determination, RTK and heave estimation are all improving the solution. A Kalman filter is known for its superior precision when a error dynamic model and sensor error models are present. By using complex statistics and a state for every estimate it is able to utilize the best part of all sensors.

## 7.2 Findings in experiment

In Section 5 a lot of effort were put into analysing the sensors. The errors that was looked into were close to Gauss-Markov processes that is a fundamental property, and a key reason why Kalman filters are computationally feasible.

### Uncoupled filter

A uncoupled filter is unstable as the IMU solution drifts of. It is not suitable for navigation. The loosely coupled filter resets the IMU strapdown equations and removes the nonlinearities.

### Sensor knowledge

A good knowledge of the sensors error model is crucial for the Kalman filter to work. The Kalman filter depends on knowing the accuracy of every measurement to estimate its process variance and computing the correctors gain. The measurements in Chapter 5 were crucial when tuning the filter.

### **Scale factor**

Scale factor is the relation between input and output. For application on a boat, the dynamics are low and the IMU is working in a small part of its sensor range. Most of the scale factor error is then compensated for in the bias estimate. By expanding the error model with scale factor and scale factor linearity the solution can be improved.

### **Turn-on Bias**

An IMU, including the STIM300, has a significant constant turn-on bias. It can be found very easy by implementing a static initialization. As static initialization is impossible on water, a In-Motion alignment is needed. A method for the Kalman filter to better estimate the Turn-on Bias, is to increase the Bias Instability in the Covariance matrix,  $\mathbf{q}_{ba}$ ,  $\mathbf{q}_{bg}$  [27].

### **IMU Non-orthogonality**

The IMU mounting has 6 degrees of freedom, 3 axes in position and 3 axes in rotation. Equation 6.4 describes the offsets in our setup. It is impossible to measure the offsets correct manually and our Kalman filter has no property of estimating misalignments. To get a more correct solution, the Error dynamics can be expanded to estimate misalignment.

### **GNSS bias**

In this experiment RTK corrections from Kartverket were used, and as proven in Chapter 5.4 errors are expected to be on centimeter level. If the GNSS solution is calculated without RTK, the expected accuracy is much lower, and large biases may occur. When such is present, bias estimation can be crucial.

### **Tight coupling**

In a loosely coupled filter the GNSS solution is computed separate from the IMU corrections. By outputting raw data from the GNSS-receiver, the kalman filter can estimate the error for every single satellite, assisted by the IMU. In areas with bad GNSS coverage, it can utilize even a single satellite, while the loosely coupled filter loses the GNSS lock if less than 4 satellites are present.

## Future work

This project is a good base for future work.

### 8.1 Hardware integration

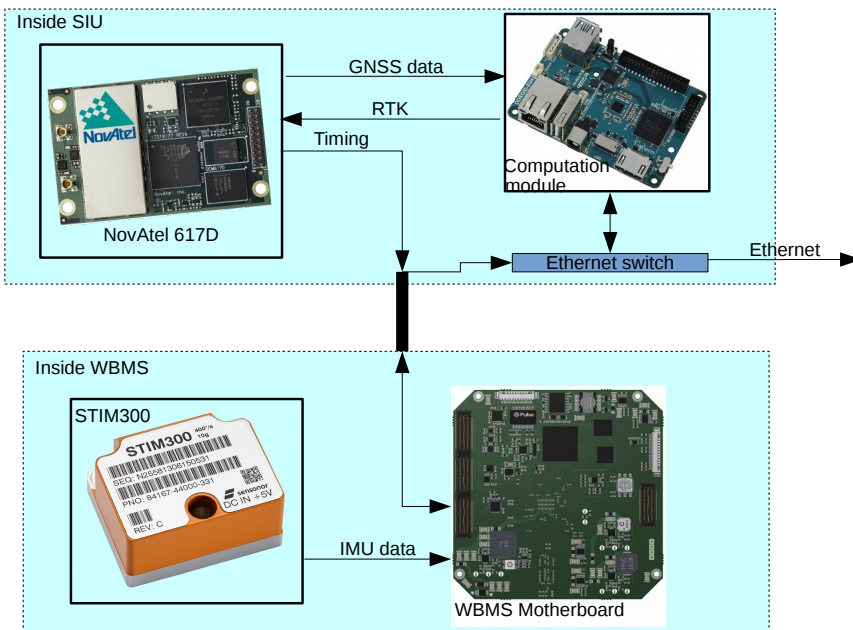
A tighter hardware integration with few componets and cables is appreciated by customers. By integrating the IMU into the sonar, the number of components and cables can be reduced, Figure 8.1. The alignment measurements are simplified for the end user, as the Sonar and IMU are in the same rigid module. The IMU data can be timetagged by the WBMS Motherboard and transferred through the already existing 100 Mbit ethernet interface. The end user can connect to ethernet and interface with the Computation module and the WBMS. RTK can be handled through ethernet by the Computation module to reduce the number of needed connections.

### 8.2 Expanding the error model

As mentioned in the discussion, Chapter 7, the error model has potential for expansion. By implementing GNSS bias, Turn-on bias, scale factor and IMU Non-orthogonality the Kalmans filter estimates are more accurate. A more accurate model will give a more accurate solution.

### 8.3 Tight coupling

By implementing a tight coupling, the accuracy can be improved, especially in areas with bad GNSS coverage.



**Figure 8.1:** Suggested setup for future work

# Conclusion

## 9.1 Theory

A literature survey gave important insight in the complexity of navigation. It became evident that very delicate processes has to take place to estimate all errors.

## 9.2 Experiment

### **Kalman filter integration**

A extended Kalman filter has been developed to use in the integration of the INS and GNSS error models. Integration schemes and initialization of the filter states are explained and argued for. A number of real-world measured datasets are gathered and put through the Kalman filter. They put the filter through more realistic tests than any simulation can do and shows that the Kalman filter is working as intended.

### **Head-to-Head test**

It is proven that today's established solutions can be challenged by something new. In this thesis hardware and software has been developed and shown results comparable to an established system with good reputation.

### **Future work**

The implementation done in Chapter 6 does have many possibilities for further system extensions which, if implemented, will make it more robust and reliable.

It is the author's assessment that all objectives of this thesis has been met.



# Bibliography

- [1] Cpos. <http://www.kartverket.no/Posisjonstjenester/CPOS/>. Accessed: 2016-05-21.
- [2] Gns base stations overview. [http://norgeskart.no/fastmerker/#6/261029/7067728/-land/+PGS\\_stasjoner](http://norgeskart.no/fastmerker/#6/261029/7067728/-land/+PGS_stasjoner). Accessed: 2016-05-21.
- [3] Navipedia. [http://www.navipedia.net/images/6/60/RIMS\\_sites.JPG](http://www.navipedia.net/images/6/60/RIMS_sites.JPG). Accessed: 2015-12-14.
- [4] Nmea data. <http://www.gpsinformation.org/dale/nmea.htm>. Accessed: 2015-12-12.
- [5] Rtklib: To do list. [http://www.rtklib.com/rtklib\\_todo.htm](http://www.rtklib.com/rtklib_todo.htm). Accessed: 2015-12-18.
- [6] Wikipedia, accelerometer. <https://en.wikipedia.org/wiki/Accelerometer>. Accessed: 2015-12-16.
- [7] Wikipedia, allan variance. [https://en.wikipedia.org/wiki/Allan\\_variance](https://en.wikipedia.org/wiki/Allan_variance). Accessed: 2015-12-14.
- [8] Wikipedia, beidou navigation satellite system. [https://en.wikipedia.org/wiki/BeiDou\\_Navigation\\_Satellite\\_System](https://en.wikipedia.org/wiki/BeiDou_Navigation_Satellite_System). Accessed: 2015-12-16.
- [9] Wikipedia, galileo (satellite navigation). [https://en.wikipedia.org/wiki/Galileo\\_\(satellite\\_navigation\)](https://en.wikipedia.org/wiki/Galileo_(satellite_navigation)). Accessed: 2015-12-16.
- [10] Wikipedia, sagnac effect. [https://en.wikipedia.org/wiki/Sagnac\\_effect](https://en.wikipedia.org/wiki/Sagnac_effect). Accessed: 2015-12-15.
- [11] IEEE Aerospace and Electronic Systems Society. *IEEE Standard Specification Format Guide and Test Procedure for Coriolis Vibratory Gyros*. IEEE Standards, 1969.
- [12] Santiago Alban, Dennis M. Akos, Stephen M. Rock, and Demoz Gebre-Egziabher. *Performance analysis and architectures for ins-aided gps tracking loops*, 2003.

- 
- [13] Atmel. *AT91SAM7S Series Datasheet*, October 2012.
- [14] Robert G. Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, 2012.
- [15] Electronic Industries Association. Engineering Department. *EIA standard RS-232-C*. EIA, 1969.
- [16] Per Enge and Pratap Misra. *Global Positioning System*. Ganga-Jamuna Press, second edition, 2006. Signals, Measurements, and Performance.
- [17] ESA. Galileo fact sheet. [http://download.esa.int/docs/Galileo\\_IOV\\_Launch/Galileo\\_factsheet\\_2012.pdf](http://download.esa.int/docs/Galileo_IOV_Launch/Galileo_factsheet_2012.pdf), February 2012.
- [18] Jay A. Farrell and Matthew Barth. *The Global Positioning System & Inertial Navigation*. McGraw Hill, 1999.
- [19] Future Technology Devices International Ltd. *TTL-232R*, September 2010.
- [20] Mohinder S. Grewal, Angus P. Andrews, and Chris G. Bartone. *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. Wiley, third edition, 2013.
- [21] João P. Hespanha. *Linear Systems Theory*. Princeton University Press, third edition, 2009.
- [22] NovAtel Inc. Oem 617d, November 2014.
- [23] Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Research Institute for Advanced Study*, 1960.
- [24] Novatel. *OEM6® Family Firmware Reference Manual*, January 2015.
- [25] RTKLIB. *RTKLIB ver. 2.4.2 Manual*, April 2013.
- [26] Sensoror. *STIM300 Inertial Measurement Unit Datasheet*, October 2015.
- [27] Eun-Hwan Shin. Estimation techniques for low-cost inertial navigation. The University of Calgary, 2005.
- [28] Jose A. Rios Sinpyo Hong, Man Hyung Lee and Jason L. Speyer. Observability analysis of ins with a gps multi-antenna system. *KSME International Journal*, 2002.
- [29] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling And Control*. Wiley, 2006.
- [30] Telecommunications Industry Association. Standards and Technology Department. *TIA/EIA-422-B*. TIA, 1994.
- [31] u-blox. *u-blox M8 Receiver Description - Manual*, 2014. Including Protocol Specification.
- [32] u-blox. *NEO-M8L*, 2015. u-blox M8 ADR module including 3D sensors.



- 
- [33] Bjørnar Vik. Integrated satellite and inertial navigation systems. Norwegian University of Science and Technology, 2014.
- [34] Ross N. Williams. *A painless guide to CRC error detection algorithms*, August 1993.
- [35] Kjetil Bergh Ånonsen and Atle Skaugen. Rate table tests of low-cost inertial measurement units. Technical Report 2014/01970, Forsvarest forskningsinstitutt, 2014.

---