

MASTER'S THESIS

L

GNSS/GPS Robustness for UAS

Joshua Stubbs
2016

Master of Science in Engineering Technology
Space Engineering

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering



LULEÅ UNIVERSITY OF TECHNOLOGY

MASTER THESIS REPORT

GNSS/GPS Robustness for UAS

JOSHUA STUBBS

Supervisors:

Associate Professor Lars-Göran WESTERBERG, *Luleå University of Technology*
Professor Dennis AKOS, *University of Colorado*

January 18, 2016

Abstract

UAS (Unmanned Aircraft Systems) are found today in many different environments performing different tasks, which puts different demands on the GNSS (Global Navigation Satellite System) engine. In this thesis three experiments are performed that are related to UAS GNSS. One experiment aims to discover if the onboard electronics cause harmful RFI (Radio Frequency Interference) to the GNSS engine, due to close proximity of various electronic components and the GNSS antenna. The other two experiments evaluate the performance of high accuracy RTK (Real Time Kinematic) and its applicability to UAS, one static experiment and one dynamic experiment. Some comparisons to manned aviation are also made, together with some background and theory to GNSS and RTK. The results show that RFI is present during one of the measurements, and is likely caused by the onboard electronics. The RFI causes a large drop in the carrier-to-noise ratio, which could negatively impact accuracy. Results from RTK experiments show some indication that RTK could successfully be implemented on UAS for navigational purposes. Future work should be focused on performing a combination of the RFI and RTK experiments during a UAS flight, with focus on how RFI impacts RTK measurements.

Acronyms

ADC:	Analog-to-Digital-Converter
AGC:	Automatic Gain Control
C/A:	Coarse/Acquisition
CDMA:	Code Division Multiple Access
CL:	Civilian Long
CM:	Civilian Moderate
CLI:	Command Line Interface
CORS:	Continuously Operating Reference Station
DC:	Direct Current
DGNSS:	Differential GNSS
DLL:	Delay Lock Loop
ECEF:	Earth Centered Earth Fixed
EKF:	Extended Kalman Filter
FAA:	Federal Aviation Administration
FCC:	Federal Communications Commission
FDE:	Fault Detection and Exclusion
FLL:	Frequency Lock Loop
FPGA:	Field Programmable Gate Array
GLONASS:	Globalnaya Navigatsionnaya Sputnikovaya Sistema
GNSS:	Global Navigation Satellite System
GPS:	Global Positioning System
GUI:	Graphical User Interface
ICAO:	International Civil Aviation Organization
IFR:	Instrument Flight Rules
IMU:	Inertial Measurement Unit
INS:	Inertial Navigation System
I/O:	Input/Output
kbps:	kilobits per second
LiPo:	Lithium Polymer
LSE:	Least Square Estimation
Mbps:	Megabits per second
MCS:	Master Control Station
MEO:	Medium Earth Orbit
NAS:	National Airspace System
NTRIP:	Networked Transport of RTCM via Internet Protocol
PLL:	Phase Lock Loop
PPP:	Precis Point Positioning
PRN:	Pseudo-Random Noise
RAIM:	Receiver Autonomous Integrity Monitoring
RC:	Radio Controlled
RFI:	Radio Frequency Interference
RINEX:	Receiver Independent Exchange Format
RNP:	Required Navigation Performance
RTCM:	Radio Technical Commission for Maritime Services
RTK:	Real Time Kinematic
SPP:	Single Point Positioning
UAS:	Unmanned Aircraft System
UART:	Universal Asynchronous Receiver/Transmitter
WAAS:	Wide Area Augmentation System
ZTD:	Zenith Total Delay

Contents

1	Introduction	1
2	Background and Theory	3
2.1	Theory of GNSS/GPS	3
2.1.1	Overview	3
2.1.2	Constellations	4
2.1.3	GPS Structure	4
2.1.4	GPS Signal Structure	4
2.1.5	GPS Receiver Basics	6
2.1.6	Positioning	8
2.1.7	Differential GNSS	10
2.1.8	RTK	11
2.2	Aviation Grade GNSS/GPS Applicability to UAS	13
2.2.1	GPS Standards in Aviation	13
2.2.2	Problem of Dense Electronics on UAS	14
2.3	Example of Receiver: Swift Navigation Piksi	16
2.3.1	Raw Data Collection and Processing	16
2.4	RTKLIB	18
2.4.1	Algorithms in RTKLIB	18
2.4.2	Included Programs	19
2.4.3	Compatibility	20
2.4.4	Example of Use	20
3	Experiments	22
3.1	Self Generated RFI in the UAS GNSS Engine	22
3.1.1	Background	22
3.1.2	Equipment	22
3.1.3	Setup	23
3.1.4	Results	26
3.2	Static RTK Test	29
3.2.1	Background	29
3.2.2	Equipment	29
3.2.3	Setup	30
3.2.4	Results	32
3.3	Dynamic RTK Test	38
3.3.1	Background	38
3.3.2	Equipment	38
3.3.3	Setup	38
3.3.4	Method	38
3.3.5	Results	42
4	Conclusions and Future Work	47
4.1	Conclusions	47
4.1.1	Conclusions from the Self Generated RFI Experiment	47
4.1.2	Conclusions from the Static RTK Experiment	47
4.1.3	Conclusions from the Dynamic RTK Experiment	48
4.1.4	Conclusions about GNSS Usage on UAS	49
4.1.5	Summary of Conclusions	49
4.2	Future Work	50

4.2.1	Self Generated RFI	50
4.2.2	RTK for UAS	50
5	Summary	51
A	RTKLIB Settings for the Static RTK Experiment	52
A.1	Common Settings for all Receivers	52
A.2	Receiver Dependent Settings	53
A.3	Post Processing settings	54
B	RTKLIB Settings for the Dynamic RTK Experiment	55
B.1	Post Processing settings for all receivers	55
B.2	Base-Station Settings	56
	Bibliography	57

Chapter 1

Introduction

The work described in this report is part of the graduation work for the M.Sc. in space engineering, with focus on aerospace engineering, at Luleå University of Technology (Sweden). The work was done at the University of Colorado in Boulder (USA) during April-October 2015.

In this thesis two aspects of UAS (Unmanned Aircraft System) GNSS (Global Navigation Satellite System) implementation will be explored, unintentional RFI (Radio Frequency Interference) effects caused by onboard electronics, and the possibility of using high accuracy RTK (Real Time Kinematic) GNSS for UAS navigation.

It is well known, and accepted, that certain electrical components leak some RFI into the nearby environment. Some of these components are intentional radiators, such as transmitting equipment, however some are unintentional radiators. The impact of unintentional radiators on GNSS UAS engines has previously been largely undocumented. Due to the limited space available on smaller UAS there is often a lot of electronics close to the GNSS antenna, which could affect the GNSS measurements due to RFI. Ochieng et al.[1] highlights some safety concerns when using GPS (Global Positioning System) for manned aviation, but does not draw the same parallel to UAS, while the John A. Volpe National Transportation Systems Center [2] discusses issues with GPS from an infrastructure point of view, and raises the issue of interference affecting the GPS measurements. Unintentional RFI can however also be expected from various sources [3]; all of this in conjuncture points to a possible issue with unintentional RFI affecting the GNSS measurements on UAS.

RTK is used in many applications, like surveying and aerial photography. The use of RTK on UAS has been quite limited, and when it has been used it has mostly been for the payloads onboard, and not for navigational purposes. Huang and Jan [4] present some initial experiments for using an RTK system on a UAS to perform autonomous landings on ships, and Skulstad et al. [5] discusses the possibility of using RTK for autonomous UAS landings in general. However, none of these sources look at using RTK for UAS in general. Some experiments are presented in this thesis that raise the possibility of extending the use of RTK on UAS.

With the growing number of UAS across many different sectors, there is a need for understanding the GNSS environment for UAS. GNSS has been used in traditional manned aircraft successfully for many years, but strict requirements are put on the systems. The need for understanding the RFI environment that affects the GNSS engines can be vital to reduce the risks of possible position related incidents as more UAS crowd the airspace shared with traditional manned aircraft. This could also put a demand on high accuracy systems, such as RTK. It is likely that many demands will be put on UAS as the integration into the airspace continues, and this report is intended to further the development of the GNSS navigation systems used on board, and to highlight some potential issues.

The report is organized as follows; chapter 2 will give theory and background to the subject, chapter 3 will present the details and results of experiments and tests that were carried out and chapter 4 will present conclusions and discuss future work. A summary can be found in chapter 5.

Chapter 2

Background and Theory

This chapter includes a summary of theory needed to understand the work performed in the thesis. General information about GNSS and GPS will also be presented here. There is extensive literature available for those interested in learning more about GNSS and GPS such as references [6, 7, 8].

A receiver, (Swift Navigation's Piksi), that was used for much of the work of the thesis is also described in this chapter. A software suite (RTKLIB) that was used for much of the work is also described here.

2.1 Theory of GNSS/GPS

2.1.1 Overview

GNSS is the umbrella term for satellite navigation systems, and there are many nations and regions that have developed their own systems. The satellites of a system make up a constellation. The similarities shared between the different constellations are many, and the theory of one constellation could be used for another constellation with only a few small changes. The American GPS system will be described most thoroughly in this thesis, as this was the most used system during the course of the thesis work, and it is also the most commonly known system. Figure 2.1.1 shows an image of a GPS satellite.



Figure 2.1.1: GPS Satellite
Public domain image from *Wikimedia Commons*

2.1.2 Constellations

Currently, there are two fully operational GNSS constellations, the American GPS (Global Positioning System) and Russian GLONASS (Globalnaya Navigatsionnaya Sputnikovaya Sistema, or GLObal NAVigation Satellite System). Further constellations are being developed, such as the European Galileo and Chinese BeiDou. Table 2.1.1 shows the number of operational satellites in these constellations.

Table 2.1.1: Number of operational satellites for each constellation as of October 2015

Constellation	Number of Operational Satellites
GPS	31
GLONASS	24
Galileo	3
BeiDou	15

2.1.3 GPS Structure

This section will describe the different constituents of the American GNSS system, GPS. The first American navigation satellite was launched in 1978 as a part of NAVSTAR by the U.S. Department of Defense. The system would not become fully operational until 1993, when a minimum number of satellites (24) were in orbit. This later became GPS, which is owned by the U.S. government and operated by the U.S. air force [9].

GPS consists of three segments, the space segment, the control segment and the user segment.

2.1.3.1 Space Segment

The space segment of GPS consists of the GPS satellites. The constellation requires that 24 satellites are active in MEO (Medium Earth Orbit) at approximately 20 200 km above earth, in order for GPS to be operational. As can be seen in table 2.1.1, there are an additional 7 satellites currently in orbit, which are there for redundancy and improved performance in areas where full sky view may not be possible, such as urban canyons. The satellites belong to different blocks, where each block has different capabilities with regards to signals, power and lifespan. Currently there are four different satellite blocks in use, block IIA, IIR, IIR(M) and IIF, with planned launches of block III in 2016. Each block has different capabilities, such as what signals that can be provided. The on-board clock also varies from block to block. See section 2.1.4 for the different signals available to GPS users [10].

The basic equipment on board the satellites include antennas for transmitting the GPS signals, antennas to receive correction information from control stations, and very accurate atomic clocks to keep time. In order to get an accurate position solution the clocks have to be compensated for both general and special relativity (time dilation due to gravitational effects, as well as the satellite's speed relative to the users) [11].

2.1.3.2 Control Segment

The control segment controls and monitors the GPS constellation. It is run by the U.S. Air Force 2nd Space Operations Squadron in Fort Collins, CO. To perform the necessary tasks they have, at their disposal, one MCS (Master Control Station), one alternate MCS, 12 command and control antennas and 16 monitoring stations. It is the control segment's job to ensure that the satellites are sending the correct messages at the right times, that the performance of GPS is on par and to re-position satellites when necessary [12].

2.1.3.3 User Segment

The user segment of GPS consists of all the hardware necessary to receive and use GPS signals, both military and civilian. Although receivers differ between different types and brands, the basics are the same. For a more detailed description of how a GPS receiver works, see section 2.1.5.

2.1.4 GPS Signal Structure

The GPS constellation is currently broadcasting on three distinct frequencies, L1, L2 and L5. Some receivers can only receive the L1 signal, some L1 and L2 and some L1, L2 and L5. In a similar fashion, some satellites are capable of only broadcasting the L1 signal, some satellites can broadcast the L1 and L2 signal, and some satellites broadcast all three. All 31 operational SVs (Satellite Vehicles) broadcast the L1 signal, and there are currently 17 operational L2 SVs and 10 operational L5 satellites [10].

2.1.4.1 GPS Signal Composition

In general (with some exceptions), each GPS signal consists of three parts. A carrier signal, one or more PRN (Pseudo-Random Noise) codes and the navigation message. The latter two parts are modulated onto the carrier frequency, creating one signal containing three different parts, see figure 2.1.2. Figure 2.1.3 shows the result of modulating a PRN code onto a carrier frequency. The individual parts are briefly explained below.

Carrier Signal The carrier signal is a simple sinusoidal wave with a distinct frequency, the carrier frequency. The carrier signal contains no explicit information. This is the frequency a receiver has to be tuned to in order to receive the GPS information.

PRN Code The PRN code is a unique code that is different for each satellite. The codes are designed to have very little cross correlation with each other, making it harder to mistake one code for another. Since every satellite broadcasts a unique code this is the method for identifying a satellite. The PRN codes are broadcast continuously, and are on constant repeat. The codes act like a key for each satellite, where the code has to be known by the receiver to "unlock" the message within.

Navigation Message The navigation message contains the navigation data, which is the essential part of GPS. It will broadcast Keplerian variables, such as eccentricity and inclination of the orbit, that give information about the orbit of the satellite together with information about the on-board clock. The information contained within the navigation message makes it possible to determine where the satellite was when the message was broadcast, as well as what time the message was broadcast. The navigation message will also give information about other satellites' orbits.

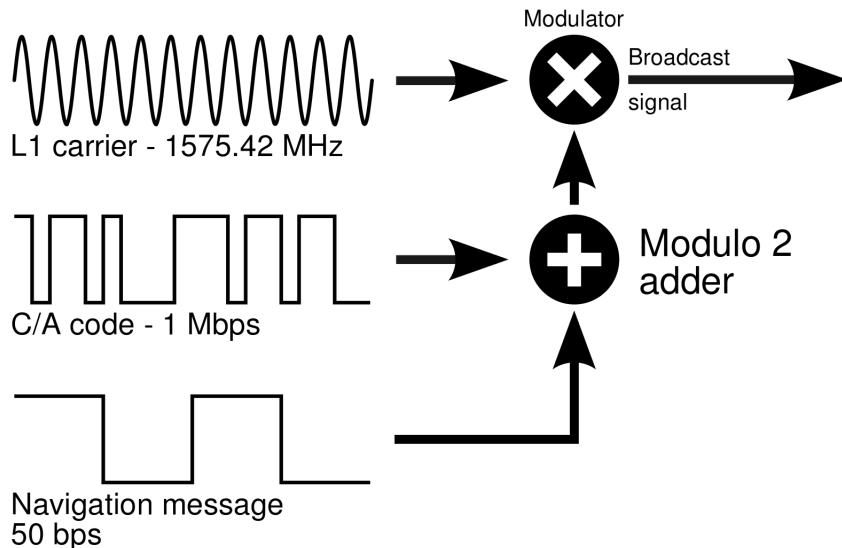


Figure 2.1.2: Parts of the GPS L1 signal, and how it is modulated
 Image from *Wikimedia Commons* under "Creative Commons Share Alike 3.0":
<https://creativecommons.org/licenses/by-sa/3.0/legalcode>

2.1.4.2 GPS Frequencies

The L1 signal consists of three PRN codes that can be broadcast, the P-code, the Y-code and the C/A (Coarse/Acquisition) code. The P and the Y code are military, and are not of interest for the scope of this thesis. The C/A PRN code is the civilian use signal and is 1023 chips (bits) long, and has a period of 1 ms (after which the code repeats itself). The codes are designed to have as little cross-correlation with the other codes as possible, which increases the chances of positive satellite identification. The L1 carrier frequency (upon which the code is modulated) is at 1575.42 MHz [11].

The L2C (C for civilian) is newer than L1 C/A, and consists of two codes, the L2 CM (Civilian Moderate) and L2 CL (Civilian Long) code. The L2 CM code repeats after 20ms and has a chipping rate of 511.5 kbps, while the L2 CL code repeats after 1500ms, but has the same chipping rate as the L2 CM code. The carrier frequency

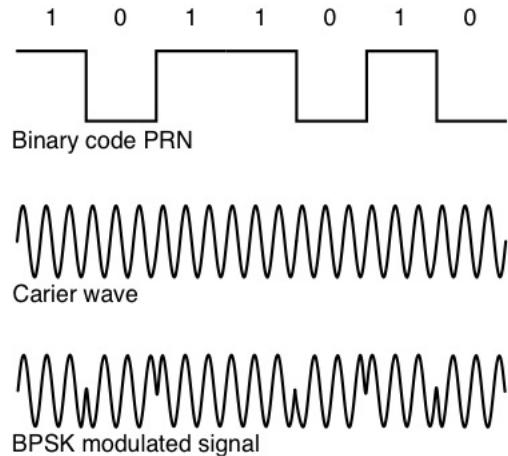


Figure 2.1.3: The result of modulating a PRN code on the carrier signal
Public domain image from *Wikimedia Commons*

of L2 is 1226.6 MHz [11].

The L5 signal is the newest of the GPS signals. Two civilian L5 codes are available, the in-phase L5 code, I5, and the quadrature code, Q5. The I5 code contains navigation data, while the Q5 code does not contain any navigation data, allowing for easier acquisition. The L5 signal is broadcast at 1176.45 MHz, and has a period of 1 ms, broadcasting 10.23 Mbps (Megabits per second) [13].

2.1.5 GPS Receiver Basics

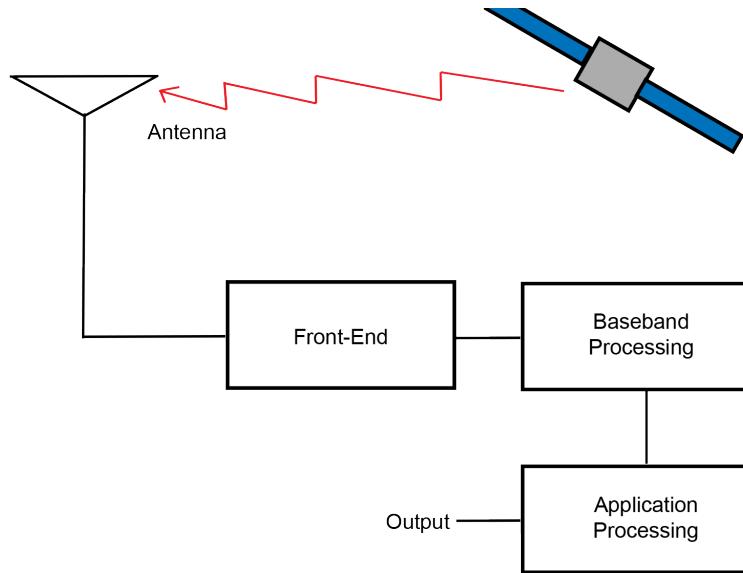


Figure 2.1.4: The basic overview of a GPS receiver

The basic structure of a GNSS receiver can be seen in figure 2.1.4. A radio signal is sent out by the satellite and is captured by the receiver antenna. This signal is then passed on to the front end where some analog processing takes place, and the signal is ultimately converted to a digital signal. The digital signal is processed in the baseband processing stage. Application processing then takes place, which can vary depending on receiver type, and is then outputted to the user in a format that also depends on usage (such as position, or time) [14].

2.1.5.1 Antenna

There are several types of GPS antennas, from small integrated antennas in cell-phones, to external patch antennas and geodetic grade antennas. Each antenna has advantages and disadvantages, and usage of antennas will vary with application. Cell-phones tend to use GPS antennas with low-power and ability to receive signals from many directions, while for a geodetic antenna power needs are not of equal concern, but it is of higher concern to have a well defined antenna pattern to minimize multi-path and interference.

2.1.5.2 Front-End

The receiver front-end has several tasks to accomplish, in order to enable signal processing at the baseband stage. The process described here is generic, and might vary from receiver to receiver.

The signal from the antenna is so far not treated, and could contain out-of-band noise. A filter is often placed in the front-end stage to remove this noise. The received signal is very weak, so an amplification often takes place at this stage.

The signal then needs to be down converted to a lower frequency, the IF (Intermediate Frequency), which is easier to process than the higher frequency signal. There are several techniques for down converting the signal, which front-end designers have to take into consideration.

The signal is then typically converted to a digital signal using an ADC (Analog-to-Digital-Converter). This prepares the signal for the digital signal processing stages, which requires the signal to be digitized. Basically, the ADC takes an analog signal and converts it to a digital format where the signal is represented in a bit-format. There is an inherent signal power loss associated with digitizing the signal, and the more bits used to describe the signal, the smaller the loss. There is an optimal ratio, k , between the maximum quantization threshold, L , and the standard deviation of the noise, σ , described by

$$k = \frac{L}{\sigma} \quad (2.1.1)$$

There are optimal values for k which front end designers aim for, minimizing quantization losses [15].

AGC (Automatic Gain Control) is often incorporated at this stage of the signal processing. The task of the AGC is to increase or decrease the power of the signal, before the ADC. The AGC will try to keep k as close to an optimal value as possible, by varying the gain.

2.1.5.3 Baseband Processing

The baseband processing stage deals with two main tasks, acquisition and tracking.

During the acquisition stage the receiver tries to find individual satellites. GPS uses a CDMA (Code Division Multiple Access) system for its signals, meaning that all satellites broadcast on the same frequency, but use different codes. This is where the PRN code comes into play. In order to find which satellite a receiver is receiving signals from, the receiver generates local copies of the PRN codes and compares these codes to the incoming signal. The locally generated PRN codes are moved backwards and forwards (in relation to the incoming signal) whilst being compared to the incoming signal. If the two signals are similar they are said to have high correlation. The correlation for each delay is evaluated, creating "peaks" where strong correlation exists. If the peak is above a predefined threshold, an acquisition has been made. The travel time of the incoming signal can then be determined by measuring how much the local PRN copy has been moved to match the incoming signal. This gives the receiver the time Δt needed to calculate the pseudo-range. Figure 2.1.5 shows how this delay is used to measure the time. The receiver also has to take into account Doppler shifts of the signal, due to receiver and satellite dynamics. This creates a two-dimensional search space, the receiver has to search for different delays and Doppler shifts. In some cases, such as when satellites recently have been tracked, the receiver can estimate the Doppler shifts, shortening the acquisition stage. If Doppler shift can not be estimated, the receiver searches through all possible combinations of PRN codes and Doppler shifts. The receiver will often have separate channels to do many parallel acquisitions [16].

Tracking does just what the name implies, it tracks the acquired signal. The tracking components track two important aspects of the signal, the code and the phase. The code is tracked in a similar manner as in the acquisition stage, using a locally generated PRN code. In tracking, however, the match between the locally

generated copy and the incoming PRN code is refined using a DLL (Delay Lock Loop). The DLL (in general) uses three versions of the PRN code that are shifted in time, one early version, one prompt version, and one late version. The DLL tries to keep the prompt version aligned with the incoming signal, but also to keep the early and late version equally spaced in time from the prompt version. In doing so, the DLL can track the signal accurately. The results of the DLL is on a feedback loop, constantly feeding information about how to adjust the signal, with regard to time [6].

The PLL (Phase Lock Loop) does a similar task to the DLL, but instead of comparing PRNs, it compares the incoming carrier signal to a locally generated copy. By doing so, the PLL can estimate both the frequency and the phase of the incoming signal. Through the results, the current Doppler estimate is updated for the receiver, so that it can track the signal better. Some receivers might use a FLL (Frequency Lock Loop), which is similar to a PLL, but does not give an estimation of the carrier phase [6].

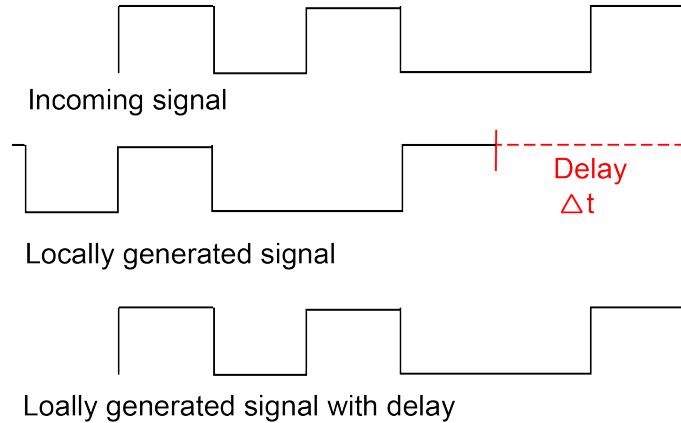


Figure 2.1.5: Concept of measuring travel time of signals using the PRN codes

2.1.5.4 Application Processing

Application processing depends on receiver design, and will perform different tasks depending on intended use of the receiver. It is at this stage that the information from the baseband processing is utilized to calculate receiver position, or whatever else is relevant to calculate from the signals [17].

2.1.6 Positioning

Two methods for determining position will be discussed in this section, code based positioning through reading the data in the navigation message, and carrier phase based positioning. If positioning is done using only one receiver, and with no corrections, this is known as SPP (Single Point Positioning). SPP accuracy of about 3-5 m is common.

2.1.6.1 Pseudo Range

The concept of GNSS positioning is simple. If the time of transmission from a satellite is known, and the time of reception is recorded, the total propagation time for that signal can be calculated using:

$$t = t_{received} - t_{transmitted} \quad (2.1.2)$$

The distance from the satellite can then be calculated using

$$s = c \cdot t \quad (2.1.3)$$

where c is the speed of light. With three measurements, and known satellite positions, a position on earth could be triangulated. However, since clocks on the receiver are seldom very precise, a fourth satellite is needed to solve for the clock bias, b . Effects such as the troposphere and ionosphere will slow the signal down, introducing an error into the measurements. Including these errors, and any other remaining errors as ϵ , we can form the basic equations for satellite navigation, the pseudo-range equation:

$$\rho^{(k)} = \sqrt{(x^{(k)} - x)^2 + (y^{(k)} - y)^2 + (z^{(k)} - z)^2} - b + \epsilon \quad (2.1.4)$$

where k denotes the k :th satellite, $x^{(k)}, y^{(k)}, z^{(k)}$ denotes the k :th satellite's position in space and x, y, z denotes the receiver antennas position. Since the position of the satellites are known through the information broadcast in the navigation message there will be four unknown variables (x, y, z and b). Because of this, information from four satellites is needed to solve for a position and time solution. This range is called the pseudo-range, because it will not be the exact range to the satellite due to the additional delays of the signal by various error sources [6].

2.1.6.2 Carrier Phase Measurements

Carrier phase measurements can be performed for greater accuracy. [8] showed that in an ideal and error-free environment, the range between the satellite and receiver antenna can be described as

$$d = N \cdot \lambda + f \cdot \lambda \quad (2.1.5)$$

where d is the distance between the satellite and receiver antenna, N is the integer number of cycles (or number of wavelengths) between the satellite and receiver antenna, f is the fraction of cycles (or wavelengths) and λ is the wavelength of the signal (approx. 19 cm for the L1 frequency). This equation can be re-written in terms of phase instead as

$$\Phi_r^{(k)}(t_r) = \Phi_r(t_r) - \Phi^{(k)}(t_r) + N \quad (2.1.6)$$

where superscript k denotes the k :th satellite and subscript r the receiver. $\Phi_r^{(k)}$ therefore represents the measured phase between the receiver and the k :th satellite, Φ_r is the phase generated by the receiver oscillator and $\Phi^{(k)}$ denotes the received phase from the satellite. (t_r) shows the phase's dependence on the time at reception. [8] further shows that due to properties of signal phase transmission, $\Phi^{(k)}$ can be re-written as

$$\Phi^{(k)}(t_r) = \Phi_e^{(k)}(t_r - \Delta t) \quad (2.1.7)$$

where $\Phi_e^{(k)}$ is the phase emitted from the satellite. Using the pseudo-range (which is equal to the true range in this idealized case), ρ^k , we can substitute

$$\Delta t = \frac{\rho^{(k)}}{c}. \quad (2.1.8)$$

Equation 2.1.6 can be re-written as

$$\Phi_r^{(k)}(t_r) = \Phi_r(t_r) - \Phi_e^{(k)}(t_r - \Delta t) + N. \quad (2.1.9)$$

Assuming an initial start time of zero and a frequency f we can write

$$\Phi_r(t_r) = f \cdot t_r \quad (2.1.10)$$

and

$$\Phi_e^{(k)}(t_r - \Delta t) = f(t_r - \Delta t). \quad (2.1.11)$$

Equation 2.1.9 can then be re-written as

$$\Phi_r^{(k)}(t_r) = \frac{\rho^{(k)}f}{c} + N. \quad (2.1.12)$$

Introducing the clock bias, b and remaining errors, ϵ and writing them in correct form [8] shows that

$$\Phi_r^{(k)}(t_r) = \frac{\rho^{(k)}f}{c} - fb + N + \frac{\epsilon}{\lambda}. \quad (2.1.13)$$

Using the relationship $f = \frac{c}{\lambda}$ we can re-write the equations as

$$\Phi_r^{(k)}(t_r) = \frac{\rho^{(k)}}{\lambda} - fb + N + \frac{\epsilon}{\lambda} \quad (2.1.14)$$

or

$$\Phi_r^{(k)}(t_r)\lambda = \rho^{(k)} - cb + \lambda N + \epsilon. \quad (2.1.15)$$

This equation can be used by receivers when using carrier-phase based positioning.

2.1.6.3 Error Sources

In the above equation, error sources have been grouped into one term, ϵ . However, there are many sources of error that affect GNSS signals, some of which will be described below:

Ionospheric Errors

The ionosphere is located approximately 60 km above the surface of the earth, and extends up to about 1000 km (these limits are quite dynamic though), and can introduce large errors in GNSS measurements by delaying the signal or increasing the propagation distance. The ionosphere is constantly in motion and changing, which makes it a difficult medium to predict. Solar flares, or geomagnetic storms, can significantly influence the ionosphere as well [18].

Mitigation techniques includes using multiple frequencies to resolve common errors. Since the ionospheric effects are frequency dependent, using multiple frequencies can remove almost all ionospheric effects [8].

Tropospheric Errors

Tropospheric errors occur in the troposphere, which is located from the ground up to about 15 km above the earth's surface. Since tropospheric effects are not dependent on frequency, these effects cannot be removed by using multiple frequencies. The tropospheric errors stem from the refractivity of the troposphere [19].

Multipath Errors

Multipath errors are most often introduced in close vicinity to the receiver. Multipath occurs when the GNSS signal from space reflects off one or several surfaces before reaching the receiver antenna. This elongates the path traveled by the signal, thus delaying it. Multipath errors vary in time as either the receiver or GNSS satellite (or both) move. The surroundings may also move, creating different multipath environments. This makes multipath hard to predict.

GPS signals have right hand circular polarization to mitigate effects of multipath, since when a circular polarized signal is reflected it will change polarization. GPS receiver antennas are made to receive the right hand circular polarized signals, so that any reflected signals are weaker. This does not completely remove the error source though, but it mitigates it to some extent. Multipath can be avoided by taking measurements in open environments, free from possibly reflective surfaces, although that is not always possible [8].

Clock Errors

The atomic clocks onboard the GPS satellites are very accurate, but as with most technical systems, there are some flaws too. The clocks can produce a range error of up to five meters per day, through instability. The clocks are monitored from ground stations, and correction terms are uploaded. These corrections are broadcast in the satellites navigation message, and will reduce the error [20].

2.1.7 Differential GNSS

DGNSS (Differential GNSS) is built upon the basics of GNSS described earlier in this section, but improves the accuracy even further. The most basic form of DGNSS uses a base station with a well known position (such as a surveyed position), a rover (which is a mobile GNSS receiver) and communication equipment that links the base station and the rover. As long as the base station and receiver are relatively close by (a few hundred kilometers), it can be assumed that many of the errors are shared between the base station and rover. However, the accuracy is degraded with increasing baseline (distance between the rover and base station) [21].

Since the position of the base station is well known, the base station can solve the GNSS equations in reverse, and calculate what the pseudo-range should be under ideal conditions. Through this, the base station can figure out the value of the errors affecting the measurements, and broadcast them to any nearby rovers. The rovers can then apply the error correction to solve for a more accurate position solution. Errors that can be resolved through DGNSS include clock errors, tropospheric errors and most of the ionospheric errors. Accuracy of about 1m can be expected when using DGNSS [21].

2.1.7.1 SBAS

Correction sources do not necessarily have to be located on land, but can be located in space too. These sources are called SBAS (Space Based Augmentation System), and several such systems are operational, such as the American system WAAS (Wide Area Augmentation System) and the Japanese system (MSAS). European (EGNOS), Indian (GAGAN), Chinese (SNAS) and Russian (SBCM) systems are being developed.

[22] shows that there are many advantages to using SBAS, such as very large coverage areas. SBAS typically uses geostationary satellites, which have an orbital altitude of approximately 36 000 km, which is significantly higher than the typical GNSS satellite orbit of approximately 20 000 km. Since the satellites are geostationary availability is regional, which has led to the development of several such systems.

A truly unique feature of SBAS is that the corrections are broadcast on the same frequency as the GNSS signals. This means that no extra communication link is needed. All that is needed is some additional features in the receiver itself, to be able to utilize the SBAS corrections [22]. In section 2.2.1 more details of SBAS and WAAS will be given.

2.1.8 RTK

RTK is a form of DGNSS that provides an even more accurate position solution, compared to standard DGNSS. An accuracy of a few centimeters can often be achieved, or even a few mm in certain cases. The main difference between the two methods is that RTK ultimately relies on carrier phase measurements, whereas standard DGNSS is code-based. RTK can be done in real time, as the name implies, or in post processing with recorded data [21]. Figure 2.1.6 shows the basic RTK setup, with one rover and one base station.

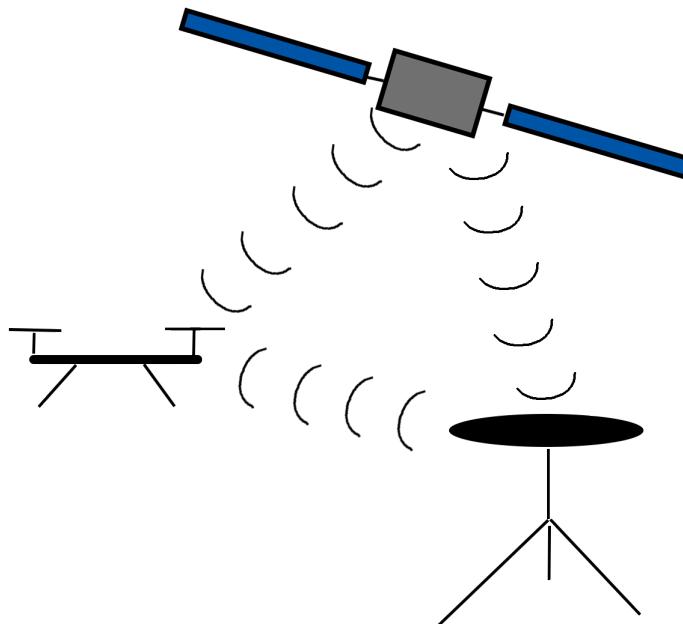


Figure 2.1.6: Basic RTK setup

The increased accuracy of RTK can be attributed to the wavelength used. Standard DGNSS relies on code measurements, which have a wavelength of about 300 m. Position can accurately be determined to within 3 m or less, which is 1 % of the wavelength. The wavelength of the carrier signal, which RTK utilizes, is 0.19 m (for L1). If RTK can determine position to within 1 % of the wavelength this would correspond to a position accurate to within approximately 2 mm (which is possible during very controlled conditions). The wavelength works as a ruler, and using a smaller wavelength would correspond to using a finer graded ruler.

Recall from section 2.1.6.2 that the range to the satellite can be expressed as the number of whole wavelengths, N and the fraction of wavelengths f that separate the receiver and the satellite. One of the main problems inherit in RTK positioning is how to determine N . This is also known as the ambiguity resolution. The ambiguity resolution can either be fixed, where the number of whole cycles has been determined, or it can be undetermined, which is known as a float solution. A fixed ambiguity resolution will provide much better accuracy then a float ambiguity resolution. Since each carrier cycle is identical to the next it is difficult to determine how many whole cycles have passed between the satellite and receiver. The problem is solved with the help of the base station. Through the known position of the base station, the carrier phase measurements of the rover+base station and some advance statistics (which are outside the scope of this thesis) the ambiguity resolution of the receiver can be solved [21].

2.1.8.1 Cycle Slips

Cycle slips can cause big problems for RTK measurements, or any other phase based measurements. When a receiver is measuring phase, such as in RTK, it also counts the number of cycles that have passed by the receiver. Sometimes this count is disrupted, and this is known as a cycle slip. A receiver may miss only one cycle, or many more, depending on circumstances. Cycle slips can be caused by a variety of factors, including sky-blockage, RFI or weak signals. It is vital that the receiver can handle the cycle slips to maintain position. The method for dealing with cycle slips will vary from receiver to receiver [20].

Cycle slips can cause a receiver to lose the fixed ambiguity resolution, returning it to a float resolution. In extreme cases the receiver may not even be able to maintain float resolution. It is common, especially during dynamic scenarios, that the receiver switches between a float solution and a fixed solution many times.

2.2 Aviation Grade GNSS/GPS Applicability to UAS

The FAA (Federal Aviation Administration), the aviation authority in the USA, has been tasked with integrating UAS into the NAS (National Airspace System). Due to the very rapid development of UAS in recent years, there has not been enough time to establish the same kind of standard for UAS that exists for traditional manned aircraft. Because of this, very few American UAS are licensed to operate. Those who wish to operate a UAS in the USA often have to file for exceptions from the current rules and regulations. The FAA wishes to change this, however, and start certifying UAS instead [23].

GPS has been used successfully for a relatively long time for traditional manned aviation. UAS will share the same airspace, and will most likely rely on GPS to some extent. An analysis of how comparable GPS on a manned aircraft to a UAS will therefore be done in this section.

2.2.1 GPS Standards in Aviation

When an aircraft operates under IFR (Instrument Flight Rules) the pilot relies on the available instruments to provide navigational information and orientation. Therefore it is absolutely critical that the flight instruments are providing correct information. Because of this, GPS units that are used for IFR flights have to be certified to certain levels of performance. The performance of the navigation systems that are used are often compared to RNP (Required Navigation Performance). RNP gives a distance of how far the aircraft may be from the reported position. An en-route flight has a RNP of 2 nautical miles, which means that the aircraft cannot be further than two nautical miles away from the reported position. RNP takes the entire system into consideration, which for GPS includes everything from the space-segment to the capabilities of the aircraft and receiver.

In certain areas, SBAS is used to increase precision of GNSS, which supplements traditional GNSS by placing additional satellites in geo-stationary orbits. The American SBAS system, WAAS was the first such system approved by ICAO (International Civil Aviation Organization). However, as the system is geostationary, it is only available for use in the USA, Canada and parts of Mexico. WAAS broadcasts on the GPS L1 frequency at 1575.42 MHz. Not all GPS L1 receivers can receive the WAAS signal though, as some additional hardware and software is required compared to an ordinary GPS receiver. WAAS increases the robustness and reliability of GPS due to several reasons. One reason is that system simply adds more satellites that can be observed by receivers. The main reason, however, is that WAAS sends correction information about GPS to users. WAAS uses a range of ground stations to monitor common errors in the GPS signals. Since the position of the ground stations are known, errors can quickly be determined. The error information is then broadcast to the WAAS satellites, which broadcasts the errors to WAAS enabled receivers [24]. Figure 2.2.1 shows an illustration of WAAS,

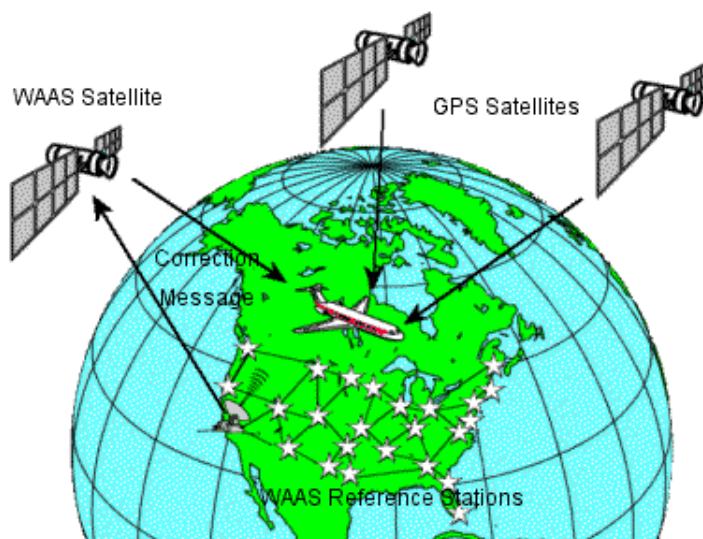


Figure 2.2.1: Illustration of WAAS
Public domain image from *Wikimedia Commons*

Another system that may be used to improve the integrity of aviation GPS is RAIM (Receiver Autonomous Integration Monitoring). RAIM relies on redundant measurements by the receiver, i.e. additional satellites in

view. The algorithms used by RAIM vary from receiver to receiver. All of them, however, aim to at the very least detect faulty measurements due to satellite errors or measurement errors. As satellite errors happen very rarely (estimated to happen every 18 to 24 months), it is usually assumed that only one such error is present at any given time [25].

There are two distinct operating modes for RAIM, that each require a different minimum of satellites in view. The most basic form is only capable of detecting faulty measurements. This requires a minimum of five satellites in view. The receiver then calculates a position and time solution using four of the satellites, and compares this to measurements including the fifth satellite. If this affects the measurements more than some pre-determined threshold the position and time solution is reported as bad to the user. Which group of four satellites that is used can be cycled around, so that satellites are continuously checked. The other operating mode uses a minimum of six satellites. It performs the same check as for the first operating mode, but with the additional satellite it is possible to exclude the satellite associated with the faulty measurements. This is known as RAIM FDE (Fault Detection and Exclusion) [1].

2.2.2 Problem of Dense Electronics on UAS

In general, manned aircraft are larger than their unmanned counterparts due to the space needed for pilots, passengers and their equipment. There are of course exceptions to this, but in general it is easier to fit all the equipment needed for flight in a smaller space on a UAS compared to a manned aircraft. There are many applications where a small aircraft is advantageous, and UAS often fit this role perfectly.

However, this could lead to problems for the GNSS receiver that are not as common on larger aircraft. By placing the GNSS antenna close to a lot of electronics (which is unavoidable sometimes if the aircraft is small), the risk of RFI in the GNSS engine increases. RFI can be characterized as electromagnetic waves interacting with a system it was not intended to interact with. RFI exists in many forms, and will interact with different systems differently.

A commission was founded in 1996 on an order by the U.S. president. It was the commission's task to investigate infrastructure vulnerabilities, which included GPS vulnerabilities. The commission concluded that GPS is vulnerable to intentional and unintentional RFI, and that sources of such should be monitored and reported [2].

Equipment that generate and broadcast electromagnetic waves can be categorized in two ways, intentional and unintentional radiators. Intentional radiators can of course cause problems, but since these sources are expected to generate signals, and characteristics of the signal often are well known (such as power and frequency), RFI effects from intentional radiators can often be avoided by proper planning and analysis.

An unintentional radiator may broadcast signals on a number of frequencies. This is in many cases allowed by regulation, as long as the power of the signal stays under a certain threshold which varies depending on what the center-frequency of the signal is. The FCC (Federal Communications Commission) defines an unintentional radiator as "A device that intentionally generates radio frequency energy for use within the device, or that sends radio frequency signals by conduction to associated equipment via connecting wiring, but which is not intended to emit radio frequency energy by radiation or induction". Such a device may emit up to 300-500 $\mu\text{V/m}$ (depending on what class of device the device falls under) as measured three meters away. If certain conditions prevail, this could be a source of RFI for the GNSS engine [3].

One of the problems with RFI in the GNSS engine is that it can cause false correlation peaks. A false correlation peak can cause problems for the tracking algorithms of the receiver, as a false peak can make the code- and carrier-tracking loops diverge from the actual correlation peak. The receiver would then be giving incorrect feedback internally, which would give a worse position solution [26].

Another potential issue with RFI is that it can effectively "drown out" the GNSS signals by raising the noise floor. Under normal conditions, the noise floor is the same as the thermal noise floor which naturally exists, and is caused by the natural movement of charged particles (such as electrons). The thermal noise floor can be calculated by

$$N = 10 \log_{10}(k T_{eff} B) \quad (2.2.1)$$

where N is the thermal noise floor power in dBW, k is Boltzmann's constant ($1.38064852 \cdot 10^{-23} \text{ J/K}$), T_{eff} is the effective temperature of the receiver front end and B is the measured bandwidth [27].

An effective front-end temperature of 300 K is a good approximation, however this can vary a lot, depending on components, antenna etc. A front-end bandwidth of 4 MHz this gives a value of $N = -137.8 \text{ dBW} =$

-107.8 dBm. The GPS L1 signal strength is about -130 dBm on earth, i.e. the signal is much lower than the thermal noise floor even without the effects of RFI, so added noise from RFI can be critical for the receiver.

2.3 Example of Receiver: Swift Navigation Piksi

A substantial amount of work was done with Swift Navigation's Piksi GNSS receiver. At first the work was aimed towards using the Piksi as a tool for interference detection. It became apparent, though, that there would be several drawbacks to using the Piksi, such as no way of accessing AGC values [28]. With no access to AGC values, interference detection becomes substantially harder. Later, the entire scope of the thesis was changed, and interference detection was no longer the goal. The Piksi was used for the experiments that are described later, and can be used as an example of a receiver and its capabilities. Since the Piksi is marketed towards UAS use, and is capable of performing RTK measurements it is of extra interest to include here.

The Piksi is an open source receiver, meaning that the source code for the processing is publicly available and modifiable by users. The source code for the FPGA (Field Programmable Gate Array) is not open source, however. The Piksi has many I/O (Input/Output) options as well, such as USB and UART (Universal Asynchronous Receiver/Transmitter). There is also an SMA port for connecting an external antenna, although the Piksi also has a built in patch antenna. All of these factors put together makes it a valuable tool for testing, as it is easily modifiable for different applications. A block diagram of the Piksi can be seen in figure 2.3.1.

GPS is currently the only available constellation for the Piksi, but planned future updates plan on integrating more constellations. The receiver utilizes the L1 frequency only. The Piksi ships with a communication link to connect the receiver to other Piksis (or other correction sources), so that two Piksis can form a base station and rover pair, enabling easy RTK solutions. The receiver is marketed towards cheap and easy to use RTK [29, 30].

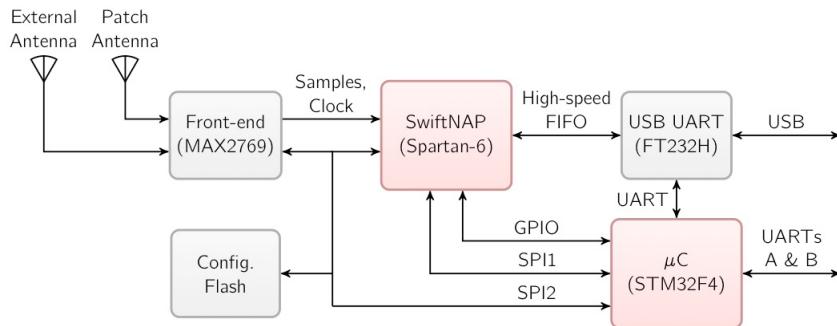


Figure 2.3.1: Diagram of Piksi courtesy of Swift Navigation, Inc. Learn more at www.swiftnav.com

Several different areas and applications of the Piksi receiver were explored, and are detailed below.

2.3.1 Raw Data Collection and Processing

It is possible to make the Piksi output raw GNSS data samples, i.e. GNSS data that has not been interpreted and processed by the receiver. Technically, it is the MAX2769 front end (see figure 2.3.1) that outputs the data after it receives it from the antenna and converts the analog signals to digital signals. However, the MAX2769 cannot directly output data in the Piksi due to how the device has been integrated into the receiver. The data needs to go through large portions of the receiver before it can be outputted. There are a few different ways of accessing the data from the MAX2769, and for the work in the thesis the USB connection was used to do so.

Raw GNSS data can be useful for many applications, particularly research. Many receivers do not have the capability of outputting the raw data, which makes this a unique feature of the Piksi.

Swift Navigation provides a piece of software for receiving raw GNSS samples over the USB connection. The software is called "Piksi Sample Grabber" and is available for free from https://github.com/swift-nav/piksi_sample_grabber. Detailed instructions are provided with the software, but the principal steps will be explained below as well. All data collection was made using a Linux operating system (Mint OS 17, 64 bit).

2.3.1.1 Configuring the Piksi to capture 1-bit raw GNSS samples

At the moment of writing (30/6-2015) the default settings for the MAX2769 and Piksi is to use 1-bit data samples. It is possible to change this, and will be described in section 2.3.1.2.

Once the Piksi Sample Grabber software has been downloaded the Piksi needs to be connected to the computer with the software through a USB connection. The connection then needs to be configured to FIFO (First In First Out) mode from the default UART mode. Included with the software is a script that will perform the necessary changes to the board when the program is run. The program can be started using the Linux terminal. Once run, the receiver can output the raw data samples. The data samples are collected by starting the "piksi_sample_grabber" software through the Linux terminal, with appropriate inputs as specified by the documentation (such as collection time, file output etc.). The documentation calls for the use of the .dat file extension for saving data, but experiments were also successful using the .bin file extension. Once the data collection is finished the receiver can be switched back to using UART using the included script for doing so, or it can be left in FIFO mode if it will be used to collect more raw data samples. UART mode is required for most other applications of the receiver.

A data file, with the name that was specified at the time of collection, will have been created in the same folder as the sample grabber software. For certain applications the data will have to be processed in order to work. Section 2.3.1.3 describes how the data is stored, in an effort to make potential processing easier.

2.3.1.2 Configuring the Piksi to capture 3-bit raw GNSS samples

The method for capturing 3-bit data samples from the Piksi is identical to the method used for 1-bit samples. There is, however, a difference in how to configure the board itself. As mentioned earlier the default settings for the Piksi is to only use 1-bit data samples, even though the MAX2769 is capable of using 3-bit samples.

First of all, the FPGA needs to be configured to utilize 3-bit samples. As the FPGA is not open source this cannot be done without the assistance of Swift Navigation. For the work described here, Swift Navigation provided the necessary firmware to use 3-bit data samples.

Next, the open source firmware was configured to use 3 bits. The firmware can be downloaded from Swift Navigation's GitHub page, <https://github.com/swift-nav>. Once downloaded there is one file that needs to be configured, max2769.c (located at src/board/max2769.c). Table 2.3.1 show what lines (at time of writing, 1/7 2015) that needs changing.

Table 2.3.1: Code in firmware file max2769.c that needs to be modified

Line	Original	Modify to
92	//MAX2769_CONF2_IQEN	MAX2769_CONF2_IQEN
93	MAX2769_CONF2_GAINREF(170)	MAX2769_CONF2_GAINREF(82)
97	MAX_2769_CONF2_BITS_1	MAX_2769_CONF2_BITS_3

Once the firmware has been compiled and uploaded (together with the special 3-bit FPGA firmware) to the board, data capture can be performed in the same way as for the 1-bit case.

2.3.1.3 Data Structure of Stored Raw Data Samples

As stated earlier, the raw sample data retrieved from the Piksi might have to be processed in order to work for certain applications.

The data is stored as 2 samples/byte, together with 1 bit indicating any errors (flag) and 1 bit is unused. When using 1-bit samples only MAX_I1 will contain any data, whereas for 3 bits all data bits are enabled. See table 2.3.2 for how the data is stored in each byte [31].

Table 2.3.2: Data structure of the raw data samples

Bit:	0	1	2	3	4	5	6	7
Description	Flag	Unused	MAX_Q1	MAX_I0	MAX_I1	MAX_Q1	MAX_I0	MAX_I1

For most applications the data has to be stored as 1 sample/byte, and so the data will have to be cut up and then restored in a 8-bit format that fits the application. This can be done in Matlab or similar software.

2.4 RTKLIB

RTKLIB is a collection of programs for GNSS positioning applications. The software package is open source and available from www.rtklib.com. The applications can either be used in real time or for post processing. As the name suggests, RTKLIB can indeed be used for RTK solutions, but can also be used for SPP or PPP (Precise Point Positioning). PPP is another form of precise GNSS positioning (like RTK), that uses corrections, mostly in post processing. Since it is not used for the work in this thesis, it will not be explained here.

Throughout the work of the thesis, that is the basis of this report, RTKLIB was used extensively. This section will therefore briefly explain what RTKLIB is and the basics of how it works in order to give a better understanding of some of the results.

RTKLIB has been built to work on Windows or UNIX machines. For Windows programs with GUIs (Graphical User Interfaces) are provided, and this section will focus on the use of those. For UNIX machines the most essential programs of RTKLIB are provided with a CLI (Command Line Interface).

2.4.1 Algorithms in RTKLIB

This section will briefly describe the fundamentals of what algorithms and systems RTKLIB uses to perform its tasks. All information is collected from Appendix E of the RTKLIB 2.4.2 manual [32].

2.4.1.1 Time

A normal scenario for GPS receivers is to use GPS Week and GPS Time of Week to keep track of time. However, this gives a resolution of $1.3 \cdot 10^{-10}$ s, which translates to a 0.04m resolution in range. Because of the need for high accuracy, RTKLIB also uses fractions of seconds which gives a range resolution of $6.7 \cdot 10^{-8}$ m in range.

2.4.1.2 Coordinates

Internally RTKLIB uses the ECEF (Earth Centered Earth Fixed) coordinate system to perform necessary calculations. The ECEF reference frame used depends on the usage and settings of RTKLIB. RTKLIB contains the necessary equations and models to transform between geodetic, local and ECEF coordinates.

2.4.1.3 Ranging

RTKLIB is capable of determining range through various methods, depending on settings and hardware. RTKLIB is capable of determining the following:

- Pseudorange
- Carrier-phase range
- Geometric range between satellite and receiver
- Elevation and azimuth of satellites

2.4.1.4 Clocks and Ephemerides

RTKLIB supports a variety of different clock and ephemerides models, which includes satellite broadcast clocks and ephemerides, as well as more precise alternatives such as SP3-c or clock information provided in RINEX files. It also allows for the use of SBAS to provide orbit and clock corrections.

2.4.1.5 Atmospheric Corrections

Atmospheric corrections can be modeled and computed through different methods in RTKLIB. Tropospheric corrections can be dealt with through a Saastamoinen estimation, SBAS or ZTD (Zenith Total Delay) estimation. Ionospheric corrections can be done through many different methods, some of which include using the satellite broadcast corrections, SBAS corrections or an ionosphere-free linear combination model.

2.4.1.6 Single Point Positioning

For single point positioning RTKLIB uses an iterated weighted LSE (Least Square Estimation). The programs can handle both linear and non-linear LSE, depending on what is provided by measurements. velocity and clock drift can be estimated by using L1 Doppler measurements. RTKLIB includes an option for excluding potentially bad satellite measurements through RAIM FDE, which works by excluding single satellites and comparing the result. RAIM FDE requires an additional two satellites to the regular four required for a position output.

2.4.1.7 DGNSS/RTK

An EKF (Extended Kalman Filter) is used to obtain position solutions when using DGNSS (Differential GNSS) or one of the RTK modes. If the baseline between the rover and base station is "small" (less than 10 km) a double differencing method can be used, which reduces the error introduced by the satellite clock bias, receiver clock bias, ionospheric effects and tropospheric effects. RTKLIB also has options for the integer ambiguity resolution, and can in some cases lock in certain parameters to improve the fix-to-float ratio, the success of which depends on what setting is used in which situation.

2.4.2 Included Programs

As mentioned earlier, RTKLIB is a collection of programs. Each program has a specific task in the sequence of what is needed to compute a position solution. The software suite can handle multiple GNSS constellations and frequencies, along with a variety of positioning modes. RTKLIB includes all programs necessary to capture raw receiver measurements and store them for later use, plotting a solution and everything in between. Below follows a quick overview (for Windows based systems) of what each program does. Similar programs exist for UNIX based systems, also provided by RTKLIB.

RTKNAVI

RTKNAVI can be used in a few different ways, depending on the situation. It has multiple input, output and logging options that suit a variety of applications. It can be used for real time positioning output (with or without RTK), or it can be used to save raw data from a receiver for post processing. It is possible to input and output files, serial connections or network connections. It also allows a variety of native receiver languages (see section 2.4.3).

RTKCONV

RTKCONV is used to convert RTCM (Radio Technical Commission for Maritime Services) or raw receiver files into RINEX (Receiver Independent Exchange Format), and outputs the results into one or several files. The number and type of files that are outputted depends on what the original file contains, and what the user selects, but output can be in the form of RINEX OBS, NAV, GNAV, HNAV, QNAV, LNAV or SBS. The OBS file provides information about observations for each data point by the receiver (phase, range etc.), while the different NAV files contain the information about the satellites. Each form of NAV file corresponds to a specific constellation (NAV corresponds to GPS, GNAV corresponds to GLONASS etc.). SBS gives information about SBAS, if used.

RTKPOST

RTKPOST is RTKLIB's post processing program. It is capable of post-processing in single receiver mode, RTK mode or PPP mode. The program outputs a position file that can be read in a third party program, but can also be analyzed using RTKLIB (see RTKPLOT). RTKPOST allows several different ways of post processing solutions, and can be setup as to mimic real time processing of a previously recorded data file. RTKPOST is very similar to RTKNAVI, the major difference being that RTKPOST is used for post processing and RTKNAVI for real time.

RTKPLOT

RTKPLOT is used to analyze the information outputted by RTKLIB at various stages. RTKPLOT is capable of plotting ground track, position plots, sky plots and more, depending on what has been inputted to the program. It also has the possibility of plotting two files at the same time for comparison. RTKPLOT also has the capability to calculate and show some statistics.

STRSVR

STRSVR is a program for downloading navigation data over a network. It can for example be used to send base station data to a rover, on the condition that both the base station and rover are connected to a computer with network access.

NTRIP Browser

The NTRIP (Networked Transport of RTCM via Internet Protocol) browser allows the user to browse NTRIP casters around the world. NTRIP casters provide correctional data to use with DGPS or RTK.

RTKGET

RTKGET is a program for downloading GNSS data from various sources. The downloaded data is typically used for PPP or RTK post processing.

2.4.3 Compatibility

RTKLIB is highly compatible with different systems and receivers. As mentioned earlier, the software suite can run on both Windows and UNIX machines. Due to the UNIX compatibility the software can be made to run on small, portable computers, making it very versatile [33].

Many native receiver protocols are supported, which further increases the ability to integrate RTKLIB into a wider variety of setups. As of writing moment the current stable version of RTKLIB is 2.4.2 and supports the following receiver protocols [32]:

- RTCM 2
- RTCM 3
- BINEX
- NovAtel OEM4/V/6
- NovAtel OEM3
- ublox LEA-4T/5T/6T
- Novatel Superstar II
- Hemisphere Crescent, Eclipse
- Skytraq S1315F
- Furuno GW-10-II/III
- JAVAD GRIL/GRIES
- NVS NV08C

2.4.4 Example of Use

RTKLIB can be used at any stage of the process for GNSS application, and may be used from beginning to end, or only for one task, depending on the users needs. For example, it could be used to just convert raw receiver data to RINEX, or it could be used to set up a base-station for another application. For the work described in this report it was used extensively in many different ways. For some experiments it was used to capture, convert and process the data, and for other experiments it was only used to post process the data.

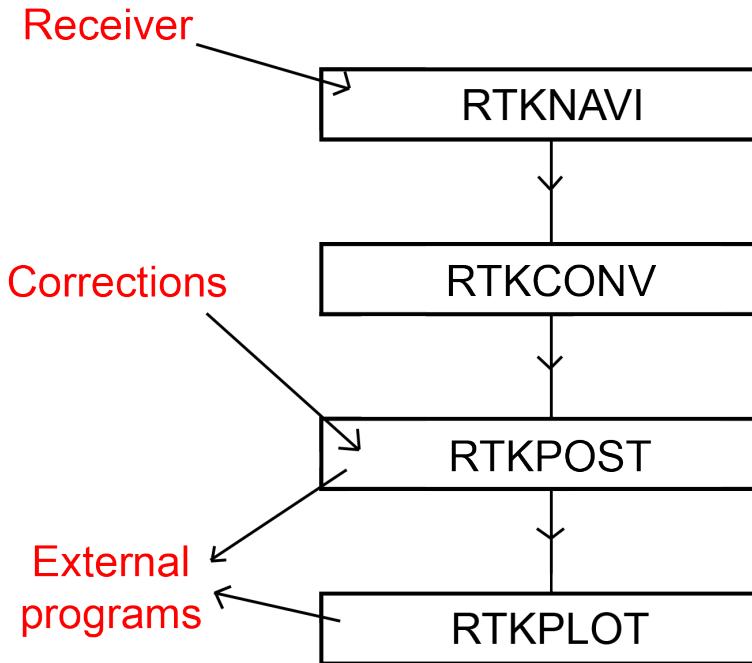


Figure 2.4.1: An example of how RTKLIB was used

Figure 2.4.1 shows a typical example of how RTKLIB was used during the course of the thesis work. This particular example shows how RTKLIB was used for some of the work presented in section 3.2, where a static RTK experiment was performed. A receiver was connected to a laptop through a USB cable. The correct options for the connection was selected in RTKNAVI (see appendix A) along with what receiver language the program should expect. RTKNAVI was then set to record the data and the logging was started. Once the experiment was over, the output files of RTKNAVI was entered into to RTKCONV which converted the information into RINEX format. The RINEX files were entered into RTKPOST together with downloaded RTK corrections from a nearby base station. This generated an output file which was later read by a MATLAB script. The output file was also opened in RTKPLOT to ensure the data had been properly processed, and to generate some plots.

Chapter 3

Experiments

This chapter will present the experiments that were performed as part of the work for the thesis. First an experiment to determine if typical UAS electronics will produce harmful GPS RFI is presented. Then two RTK experiments are presented, one static and one dynamic. The purpose of the RFI experiment was to asses the feasibility of the original thesis idea, to be able to detect harmful RFI from a UAS. However, due to the results of this test, focus of the thesis was shifted. The results from the RFI tests are still of interest with regards to the robustness of GNSS measurements on UAS. The RTK experiments were performed to determine if high accuracy GNSS solutions can be implemented on UAS, while maintaining system robustness.

3.1 Self Generated RFI in the UAS GNSS Engine

3.1.1 Background

As described in section 2.2.2 there is a risk of RFI affecting the GNSS engine due to intentional and unintentional radiators. An experiment was set up to determine if a standard small UAS would generate notable GPS RFI.

With a UAS borrowed from the RECUV team at the University of Colorado in Boulder the test was set up. The UAS was a basic fixed-wing setup of a small remote controlled aircraft. The aircraft had been fitted with various electronic components to make it autonomous. It was chosen because it is considered a common setup. An image of the UAS can be seen in figure 3.1.1. The UAS remained stationary on the ground throughout the tests.

3.1.2 Equipment

UAS

The UAS that was tested can be seen in figure 3.1.1. It was custom built by the RECUV team at the University of Colorado in Boulder. Some of the equipment of interest that was on-board was:

- Standard RC (Radio Controlled) aircraft equipment
 - RC receiver
 - Servos
 - DC (Direct Current) electric motor
 - LiPo (Lithium Polymer) battery
- A Pixhawk flight controller
- A 915 MHz communications transmitter/receiver
- A 2.4 GHz communications transmitter/receiver

Testing Equipment

- ublox EVK-6T L1 GPS receiver
- Laptop computer for logging data from the ublox receiver



Figure 3.1.1: The UAS that was used for the testing

- R&S FSH4 Spectrum Analyzer
- 40 dB in-line amplifier
- ublox L1 patch antenna (27 dB amplification)
- 9V DC power for antenna and amplifier
- SMA cables and signal splitters

3.1.3 Setup

The patch antenna was first placed on the nose of the aircraft for some reference tests, and then in the belly of the aircraft at a common antenna location, see figure 3.1.2. The wire covering the antenna in the figure was removed for the tests. Figure 3.1.3 shows an image of the test equipment.

The patch antenna was connected to a bias tee which enabled a 9V battery to feed DC electricity to the antenna and an amplifier. The resulting signal was then split twice, one end went to the ublox receiver and the other to the spectrum analyzer. A block diagram of the setup can be seen in figure 3.1.4.

Three tests were planned and performed:

1. A reference test where the antenna was placed outside the airframe (on the nose of the aircraft) with all equipment turned off (by disconnecting the battery)
2. A second reference test with the antenna placed inside the airframe (see figure 3.1.2), and all the equipment switched off (by disconnecting the battery)
3. The RFI test with the antenna inside the airframe (see figure 3.1.2) and all the equipment running (battery connected, flight computer on, transmitters/receivers on, engine running and servos moving).

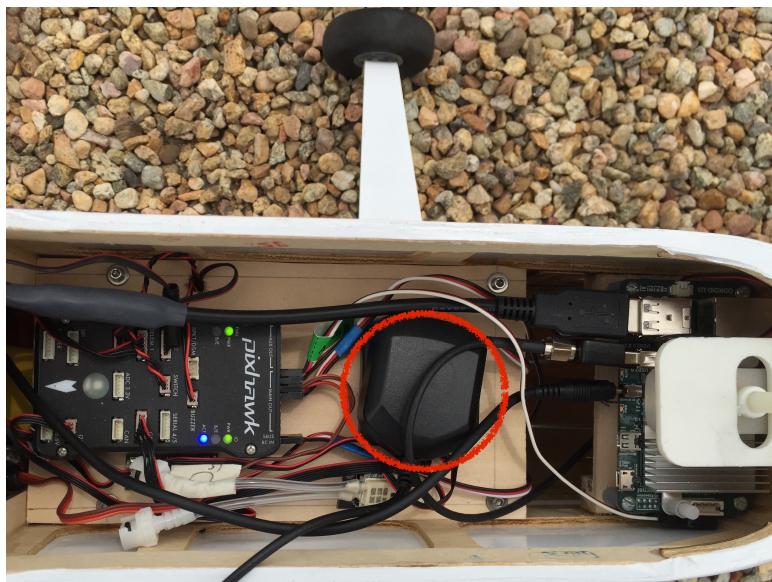


Figure 3.1.2: Antenna location inside the aircraft



Figure 3.1.3: Test equipment

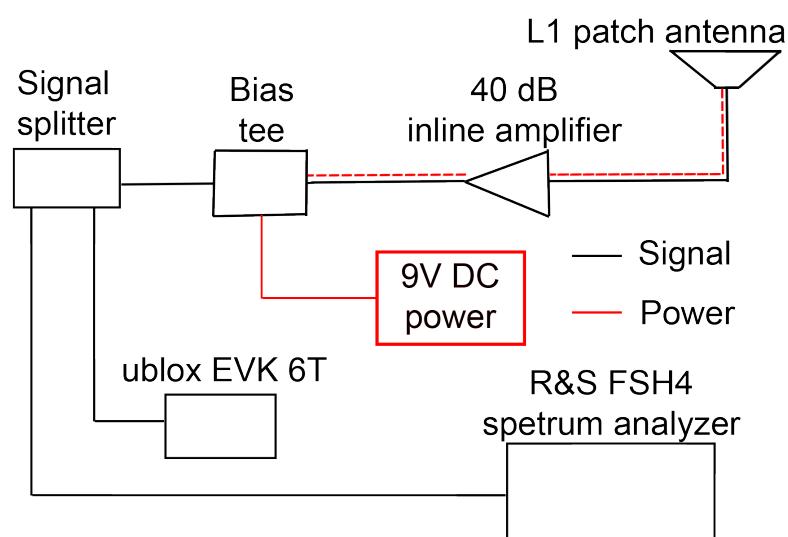


Figure 3.1.4: Block diagram of the setup for testing the presence of RFI

3.1.4 Results

Figure 3.1.5 shows a graph of how many satellites were tracked, plotted against time, by the ublox receiver for each test. As the number was the same for all tests, all lines coincide.

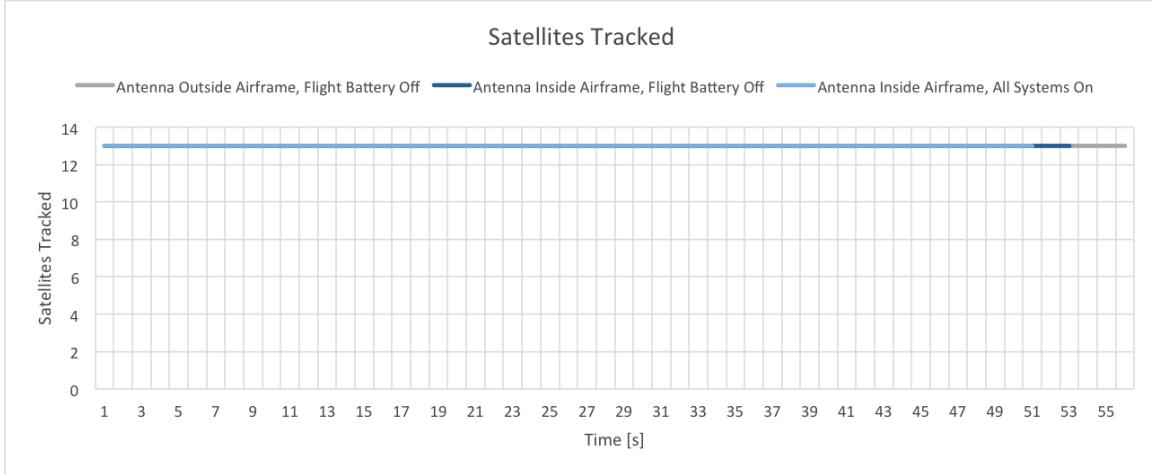


Figure 3.1.5: Number of satellites tracked for each RFI test

Figure 3.1.6 shows a graph of the carrier-noise-ratio, C/N_0 , plotted against time. The ratio was averaged over all received satellites.

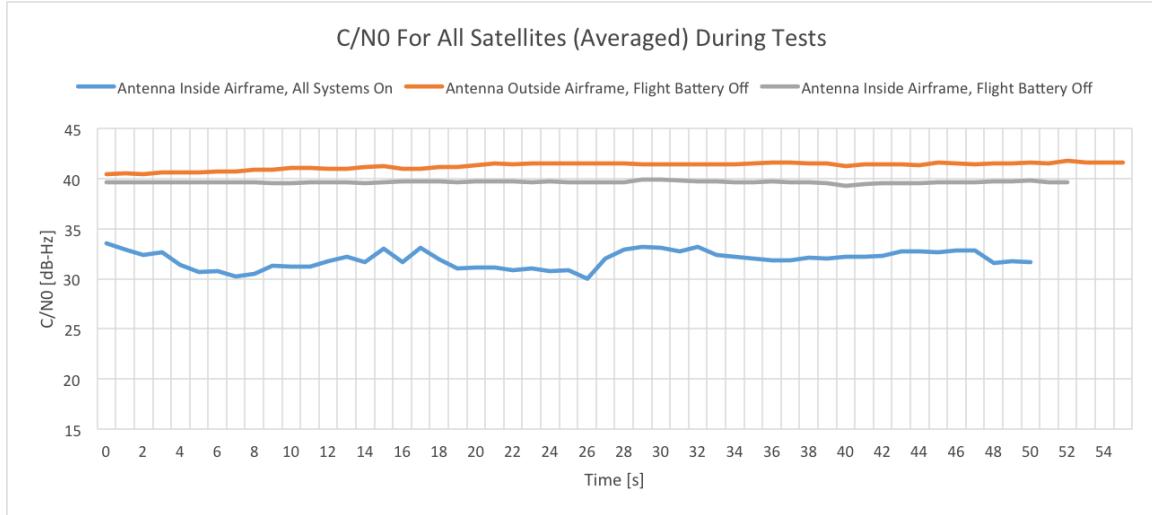


Figure 3.1.6: Average C/N_0 ratio for each RFI test

Table 3.1.1 shows numerical results from the ublox receiver. The values were *not* averaged over all satellites before being entered into the table, hence the differences from figure 3.1.6.

Figures 3.1.7, 3.1.8 and 3.1.9 show screen shots from the spectrum analyzer of tests 1, 2, 3 (see setup, section 3.1.3, for a description of the tests). The center frequency of the spectrum analyzer was set to 1575,42 MHz, the span (the distance from the center frequency to either edge) was 30 MHz. A resolution band-width of 30 kHz was used. The plots show the average of 50 sweeps. **Note that the y-axis is not correctly adjusted. Relative measurements are still valid, i.e. a 10 dB change in the graph is still a 10 dB change.**

Table 3.1.1: Numerical results from the ublox receiver, values have *not* been averaged over all satellites

Test:	Max C/N ₀ [dB-Hz]	Min C/N ₀ [dB-Hz]	Mean C/N ₀ [dB-Hz]	Median C/N ₀ [dB-Hz]
Antenna outside airframe, battery disconnected	47	27	41.3	41.6
Antenna inside airframe, battery disconnected	37	27	39.1	39.1
Antenna inside airframe, battery connected and all systems running	41	15	31.1	30.8

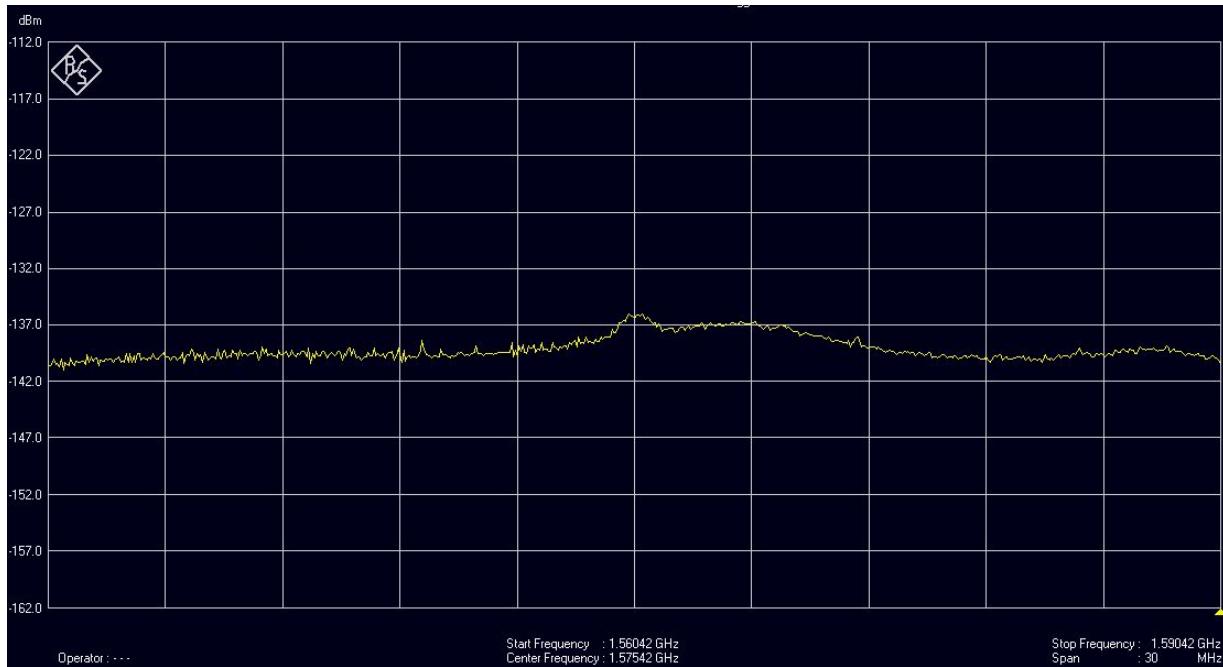


Figure 3.1.7: Test 1 (antenna placed on the outside of the airframe, battery disconnected), screen shot from spectrum analyzer

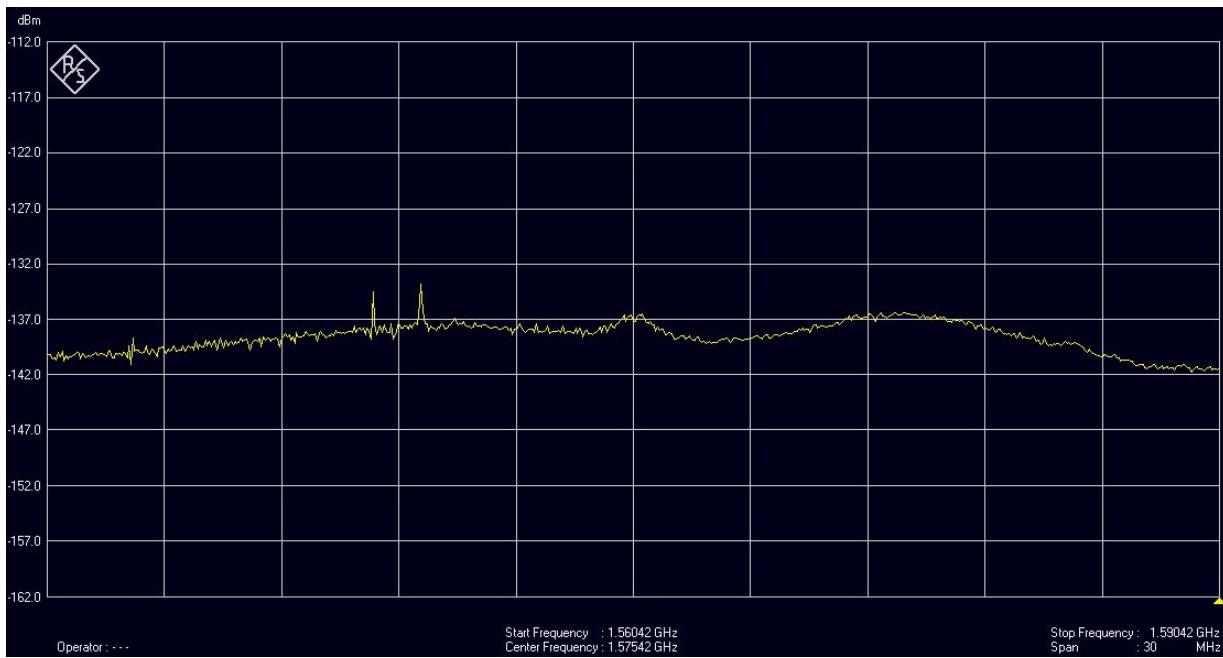


Figure 3.1.8: Test 2 (antenna placed on the inside of the airframe, battery disconnected), screen shot from spectrum analyzer

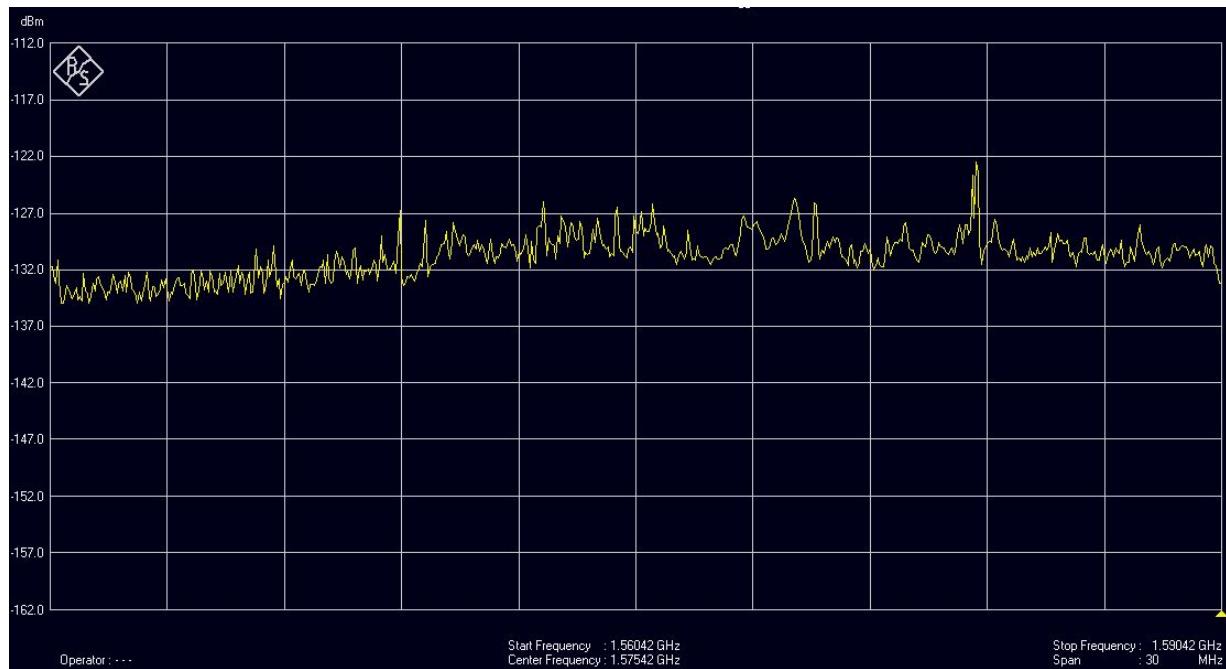


Figure 3.1.9: Test 3 (antenna placed on the inside of the airframe, battery connected and all systems running), screen shot from spectrum analyzer

3.2 Static RTK Test

An experiment was performed with several GNSS receivers to perform RTK under static conditions (i.e., the antenna is stationary). The test was carried out at the university of Colorado in Boulder. This section will describe the experiment and the results.

3.2.1 Background

The test was setup as a first test of the capabilities of RTK and RTKLIB. Although the results of the test was not directly relevant to the RTK-related goals of the thesis, it was a step a long the way to learn how RTK can be implemented on UAS.

Many of the methods and results were utilized in a more advanced, dynamic test that will be described in section 3.3.

The goals of the test was to:

- Learn how to use RTKLIB in a real scenario
- Gain vital skills necessary to perform a dynamic test
- Compare performance of receivers
- Compare performance of RTK during different scenarios

Due to technical issues with RTKLIB, only GPS data was used.

3.2.2 Equipment

Four RTK capable receivers were used. The receivers were all connected to a laptop computer in order to record the data. Below follows a brief summary of the receivers' capabilities, and a quick background to the receiver.

3.2.2.1 Swift Navigation Piksi

The Piksi is an open source receiver, designed to perform high accuracy RTK at a low cost. It is marketed towards use on UAS, and is therefore of interest to test. [29] For a more detailed description of the Piksi, see section 2.3.

Capabilites:

GPS: L1

GLONASS: L1 (Not yet implemented)

Galileo: E1 (Not yet implemented)

Beidou: N/A

Other Systems: SBAS (Not yet implemented)

[30]

3.2.2.2 ublox EVK-6T

The EVK part of the name indicates that it is an evaluation kit [34], and is not an actual receiver. The receiver itself is a ublox LEA-6T which sits inside the EVK-6T, and it is the LEA-6T that is capable of carrier-phase measurements. The LEA-6T is relatively cheap and light-weight, and is found on some UAS, and is therefore of interest to test.

Capabilites:

GPS: L1

GLONASS: L1

Galileo: Open Service capable

Beidou: N/A

Other Systems: SBAS

[35]

3.2.2.3 Novatel Flexpak 6

The Novatel Flexpak 6 is a well rounded receiver, capable of handling most constellations and frequencies. It was included in the tests because of its relative light weight and high number of signals it can receive.

Capabilites:

GPS: L1/L2/L2C/L5

GLONASS: L1/L2/L2C

Galileo: E1/E5a/E5b/AltBOC

Beidou: B1/B2

Other Systems: SBAS, QZSS, L-Band

[36]

3.2.2.4 Trimble NETR9

The Trimble NETR9 was included because it was recommended as very precise receiver by experienced GNSS users and professors at the University of Colorado in Boulder.

Capabilites:

GPS: L1/C/L2E (Trimble method for tracking unencrypted L2P)/L2C/L5

GLONASS: L1 C/A and unencrypted P code/L2 C/A and unencrypted P code/L3 CDMA

Galileo: L1 CBOC/E5a/E5b/E5AltBOC

Beidou: B1/B2/B3

Other Systems: SBAS, QZSS, L-band

[37]

For much of the data logging, and all of the post processing, RTKLIB was used. A description of RTKLIB can be found in section 2.4.

3.2.3 Setup

The receivers were set up in the basement GNSS lab at the University of Colorado, Boulder. A Trimble Zephyr antenna was placed on the roof of the building, with a cable running down to the basement lab. The signal was boosted by a 20dB amplifier in order to get clearer results. The signal was then split six times. Two of the signals were used in a separate experiment, and the remaining four went to the receivers described above. See figure 3.2.1 for a diagram over the setup, and how the signal was split.

All receivers, except for the Piksi, were connected to RTKNAVI by the connection type shown in figure 3.2.1. The Piksi does not support any of the type of messages that RTKLIB supports, and therefore, the Piksi was set to log data through the Piksi Console software provided by Swift Navigation. For each receiver a separate RTKNAVI client was started. The relevant settings for RTKNAVI are shown in appendix A.

Logging was enabled in RTKNAVI, and the data collection was then started at approx. 2232 UTC on 27/8-2015, and lasted for approx. 2 hours.

3.2.3.1 Post Processing:

The data collected using RTKNAVI is saved in a raw format that needs to be converted to RINEX before it can be post processed. RTKCONV was used for the conversion. To convert the files, the raw receiver file was inputted, the correct receiver language was chosen and the conversion was then executed. RTKCONV automatically detects what constellations are available, and creates appropriate RINEX files. The data recorded by the Piksi Console did not have to be converted as it recorded in RINEX format (.obs and .eph).

The converted RINEX files were then processed with RTKPOST as post-processed RTK. The base-station data was downloaded from the CORS (Continuously Operating Reference Station) website (geodesy.noaa.gov/CORS/). Data from two nearby base stations were downloaded, P041 and DSRC. When processing with RTKPOST, the .obs file from the base station was used, together with the converted .obs and .nav file from the receiver. The settings used for RTKPOST can be seen in section A.3.

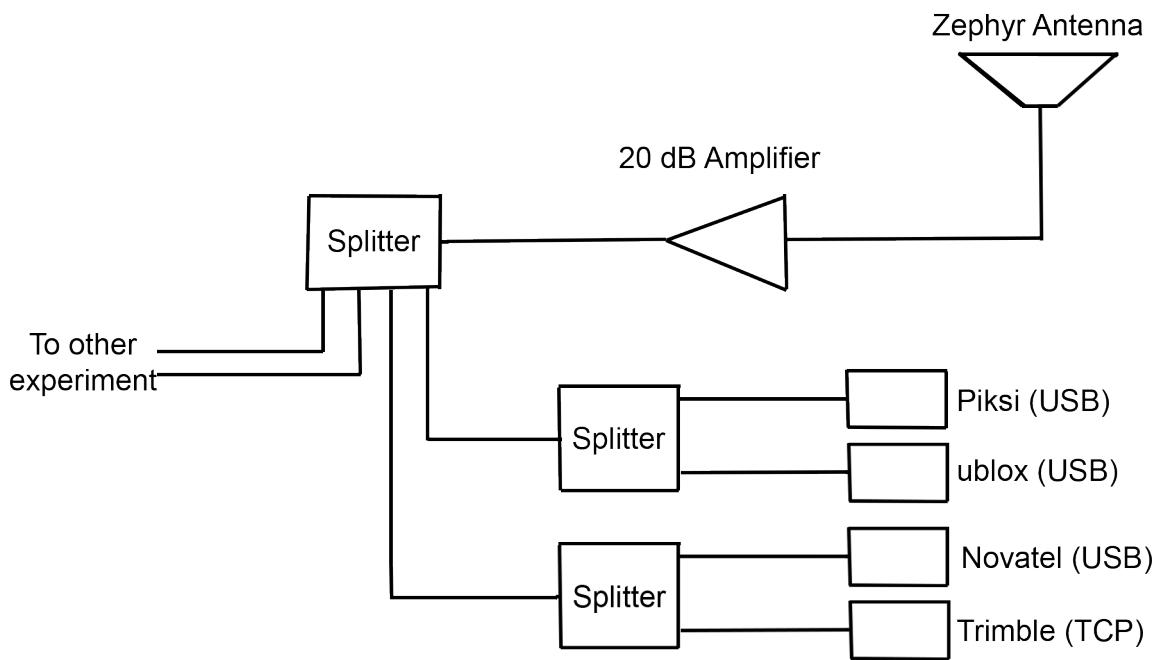


Figure 3.2.1: Diagram of receiver setup and PC connections

The .eph file (which in theory could replace the .nav file) generated by the Piksi Console cannot be read by RTKLIB, and therefore the .nav file from a base station had to be used instead. The effect of using the base station .nav file should not be noticeable as it contains information about the observed satellites, which should be the same for the base station and the receiver.

The results from the post processing were plotted in Matlab, and can be seen in section 3.2.4 below.

3.2.3.2 Attempt to Decide True Position of Antenna

The exact position of the antenna had to be determined in order to have a truth reference to compare the results against. Several methods were tried to determine the exact position. A method for determining truth that was tried but not included here, was NASA's APPS (Automatic Precise Positioning Service). It was not included because it was deemed to be inaccurate.

RTKPOST was eventually used to decide the position of the antenna. In order to get as good of a position estimate as possible, some changes were made to the methods used to get the RTK positions of the receivers. These changes in RTKPOST were:

- Used .nav file from base station instead of .nav file from receiver. As mentioned earlier there should be no difference between the two. As the base stations typically have been operating for a long time, any problems in generating the .nav file should have been solved, and so using the .nav file from the base station is more of a precautionary move.
- Used forward+backward filter type instead of forward filter type (Appendix A, setting 2). This smoothed the results further by processing the data an extra time after initial processing, smoothing the original results (this is only possible in post-processing modes).
- Only data from the Trimble and Novatel was used for the truth reference, as these support multiple frequencies giving them more accessible data to use, which should improve the accuracy.
- Only fixed RTK data from receivers was used, i.e. data points that were the ambiguity resolution was not fixed were discarded.

This was done with the two base stations and the two receivers, totaling four positions. The average value (over the entire data set) of each receiver-base station pair was entered into Microsoft Excel and plotted. The results

can be seen in section 3.2.4 below. The average of the four positions was used as the truth reference, against which the other receivers were compared.

3.2.4 Results

3.2.4.1 General

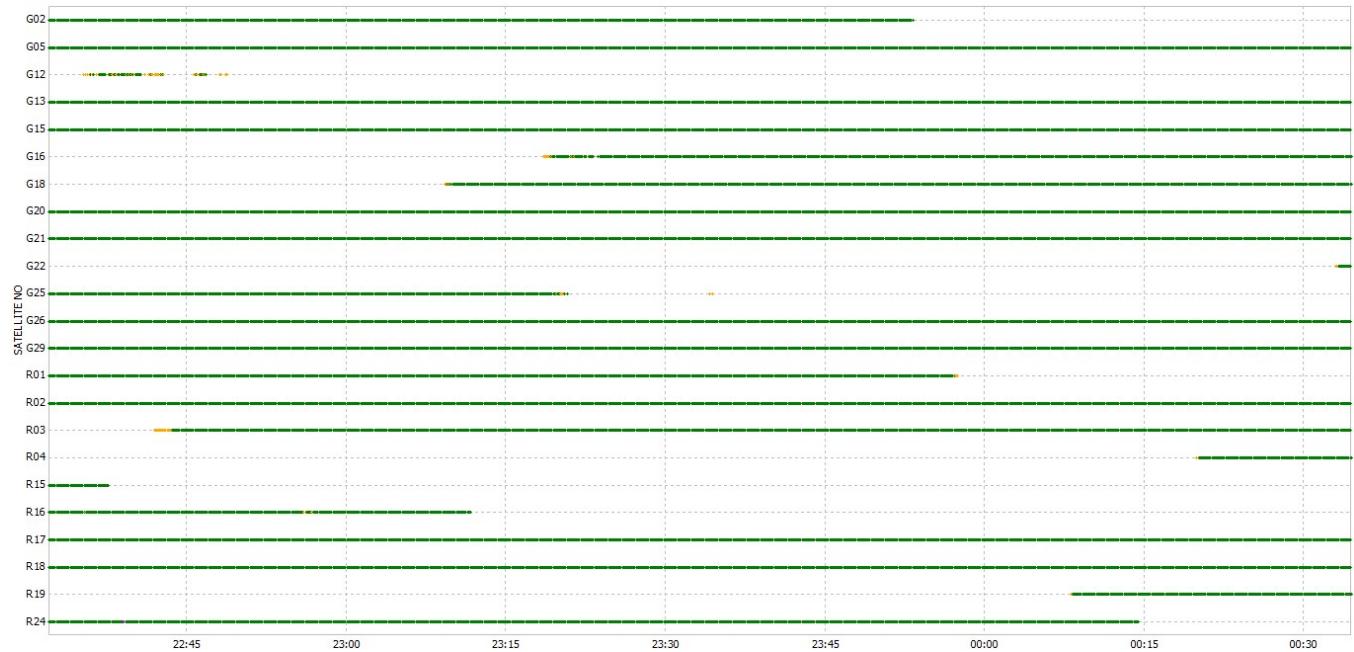


Figure 3.2.2: Visibility of all GPS satellites received by the Novatel receiver during the entire session

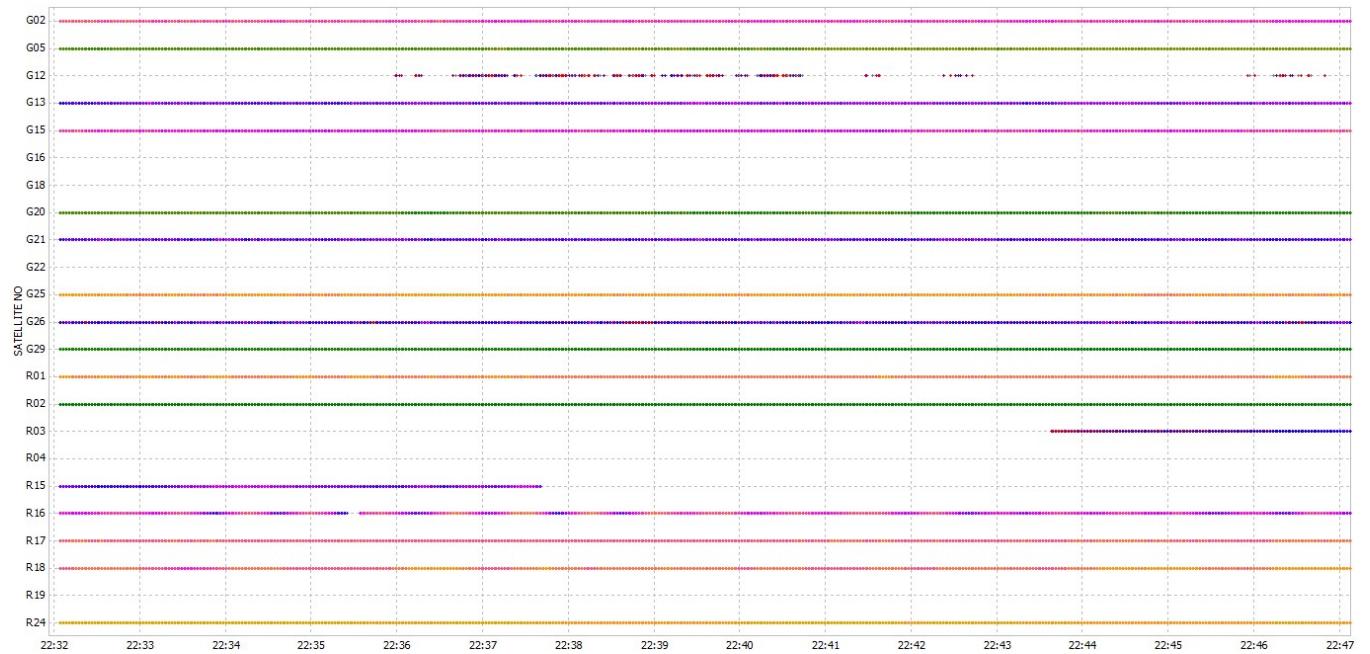


Figure 3.2.3: Visibility of the L2-Satellites received by the Novatel receiver during the first 900 seconds of the session

3.2.4.2 Truth Reference

The results from the attempts to decide the truth reference for the location of the antenna can be seen in figure 3.2.4. The average of all the positions are used for the other graphs. The coordinates and their average can be seen in table 3.2.1.

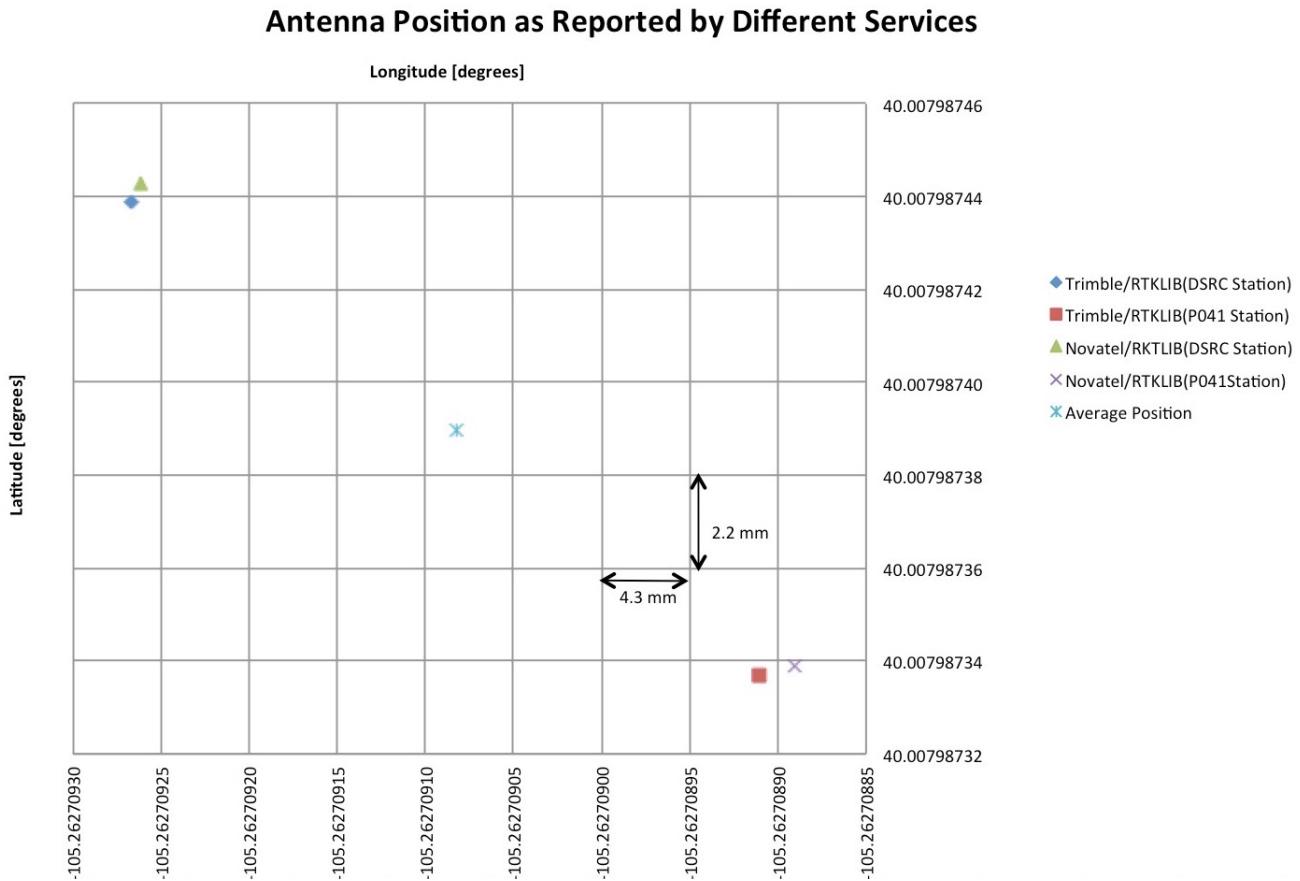


Figure 3.2.4: Coordinates of antenna, according to different methods. Average position also included

Table 3.2.1: Antenna position as calculated by using different receivers and base stations with RTKLIB. Average position also included.

Receiver/Base Station	Latitude [degrees]	Longitude [degrees]	Altitude [m]
Trimble/DSRC	40.007987439	-105.262709267	1624.6402
Trimble/P041	40.007987337	-105.262708911	1624.6097
Novatel/DSRC	40.007987443	-105.262709261	1624.6400
Novatel/P041	40.007987339	-105.262708890	1624.6034
Average	40.007987390	-105.262709082	1624.6233

3.2.4.3 Base Stations

The performance of base stations was compared. The Trimble receiver was used to compare the accuracy of the base stations, and can be seen in figure 3.2.5. Figure 3.2.6 shows the same data, but only the first 900 seconds.

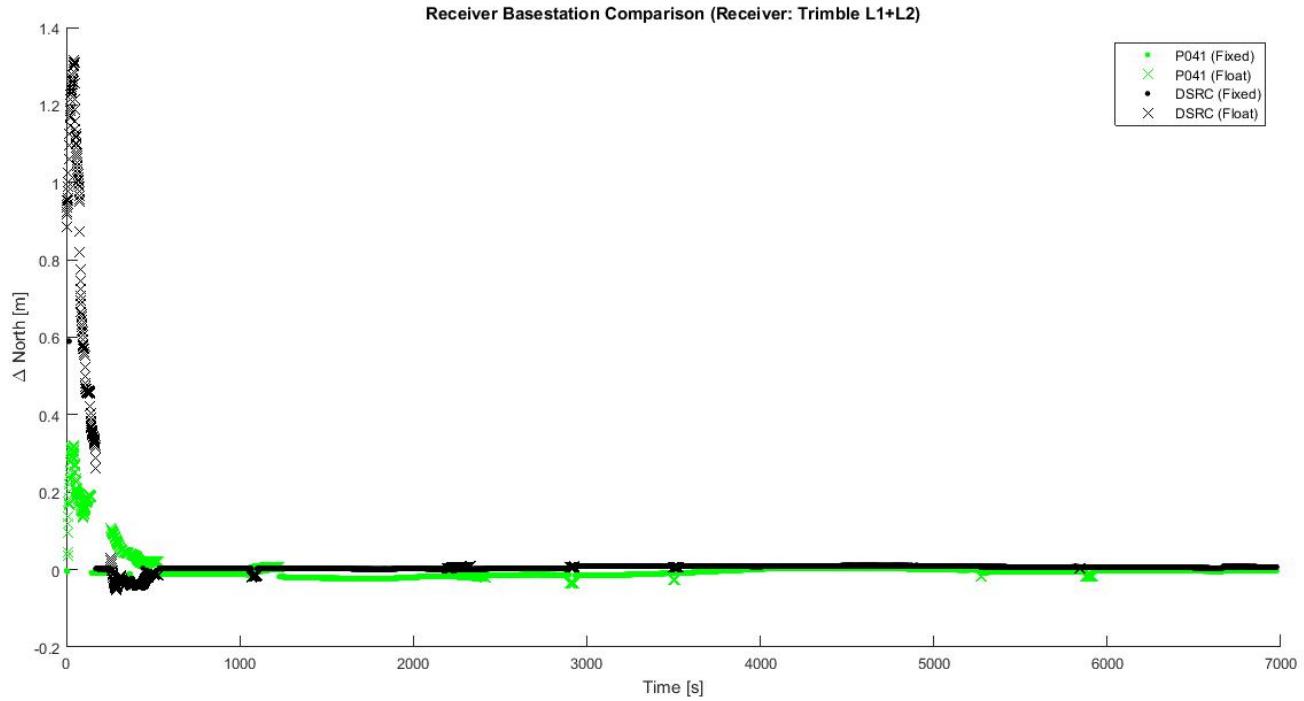


Figure 3.2.5: Difference (in m) north from estimated truth reference

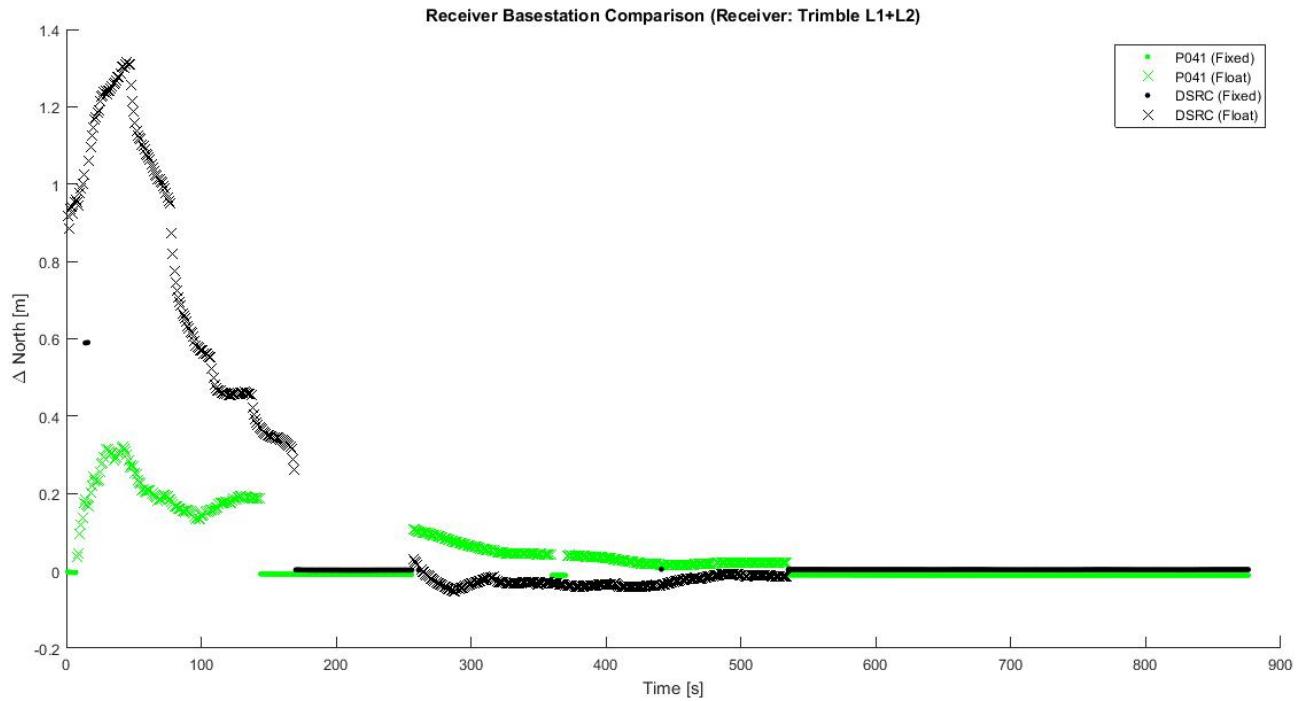


Figure 3.2.6: Difference (in m) north from estimated truth reference, first 900 seconds

3.2.4.4 Receiver Comparisons

The performance of individual receivers was compared. Figure 3.2.7 shows how each receiver performed when only using the L1 GPS frequency. Figure 3.2.8 shows the same data, but without the Piksi, and only the first 900 seconds of data.

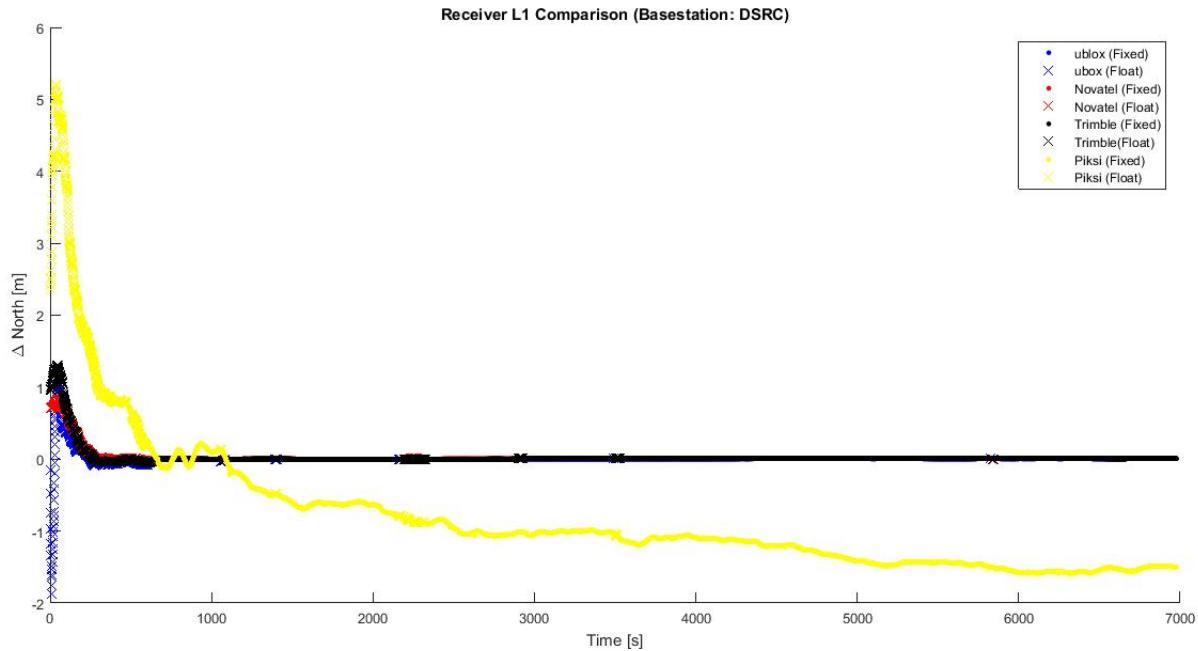


Figure 3.2.7: Difference (in m) north from estimated truth reference for L1 GPS

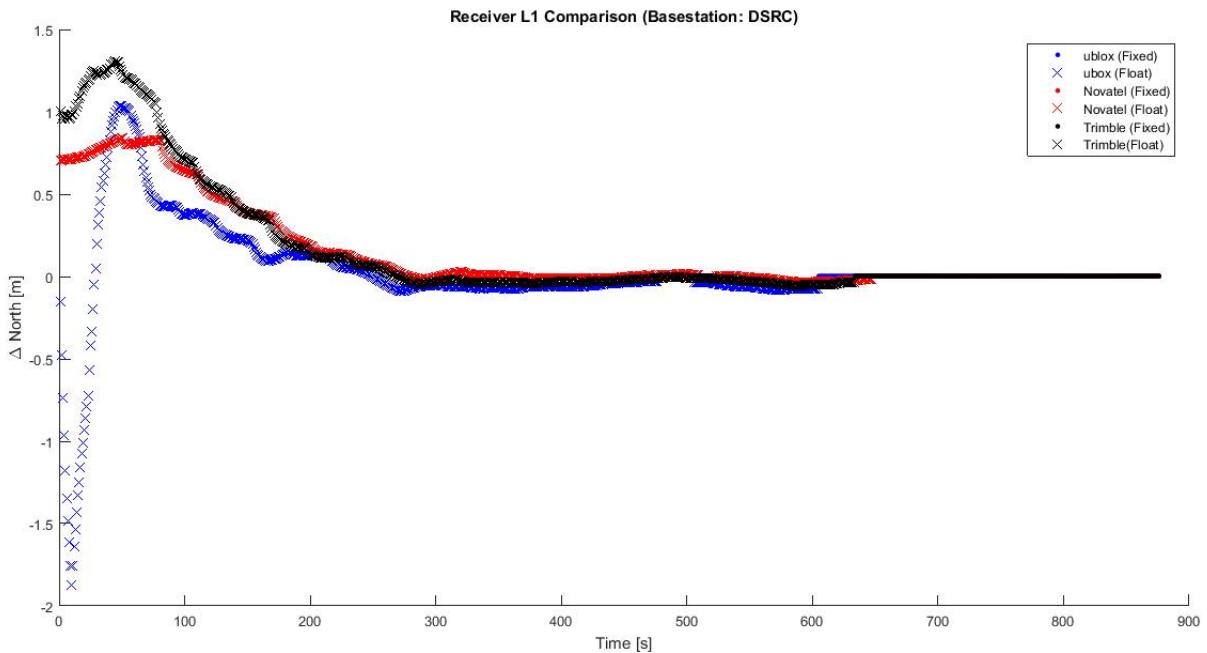


Figure 3.2.8: Difference (in m) north from estimated truth reference for L1 GPS, first 900 seconds

Figure 3.2.9 shows a comparison of L1+L2 capable receivers, and figure 3.2.10 shows the first 900 seconds of the same data.

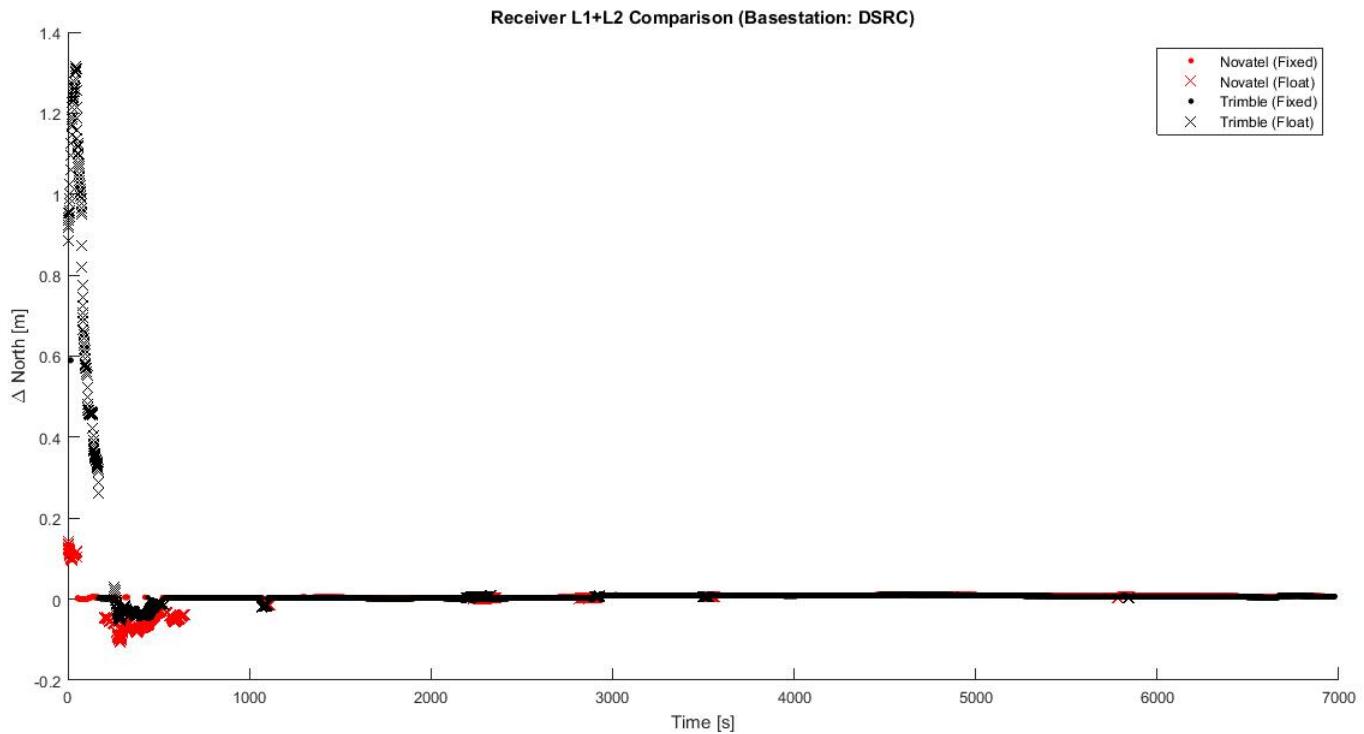


Figure 3.2.9: Difference (in m) north from estimated truth reference for L1+L2 GPS

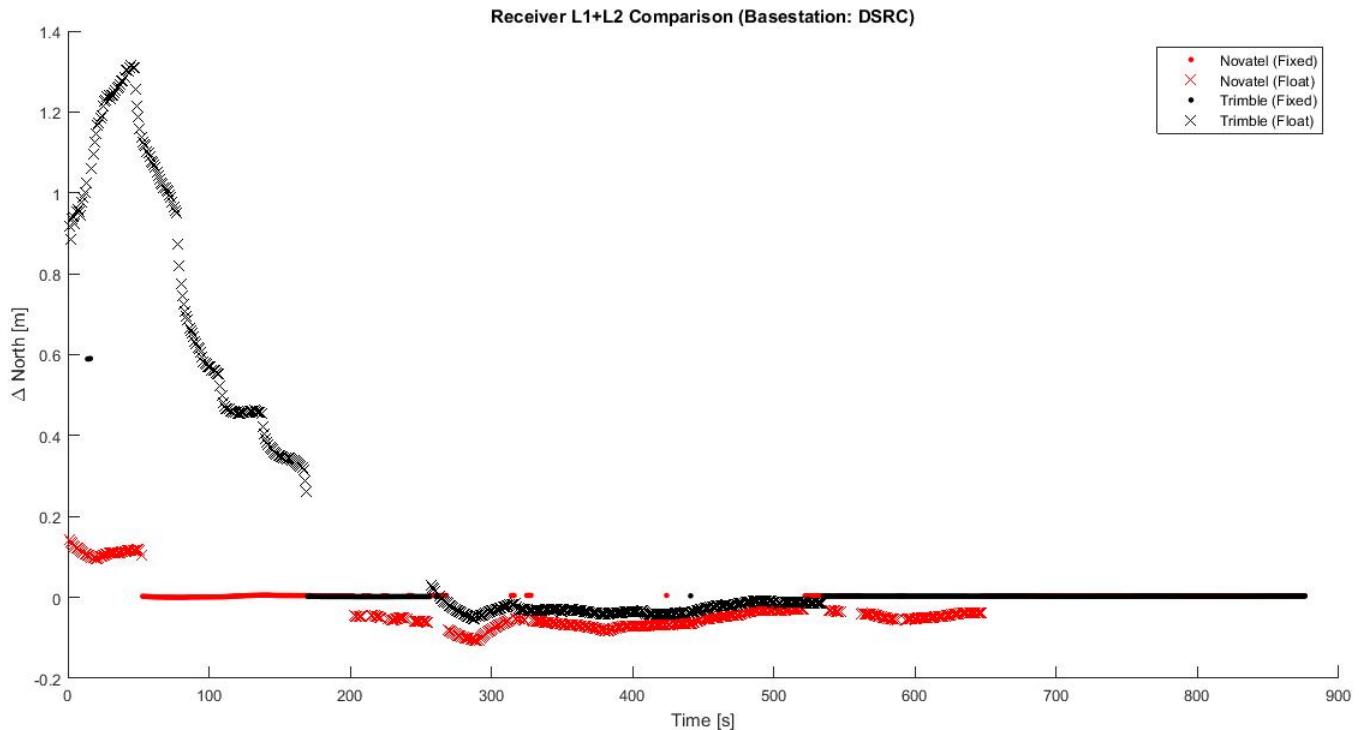


Figure 3.2.10: Difference (in m) north from estimated truth reference for L1+L2 GPS, first 900 seconds

Figure 3.2.11 shows a comparison of only using the GPS L1 frequency vs. using the L1+L2 frequency on the Trimble receiver. Figure 3.2.12 shows the first 900 seconds of the same data.

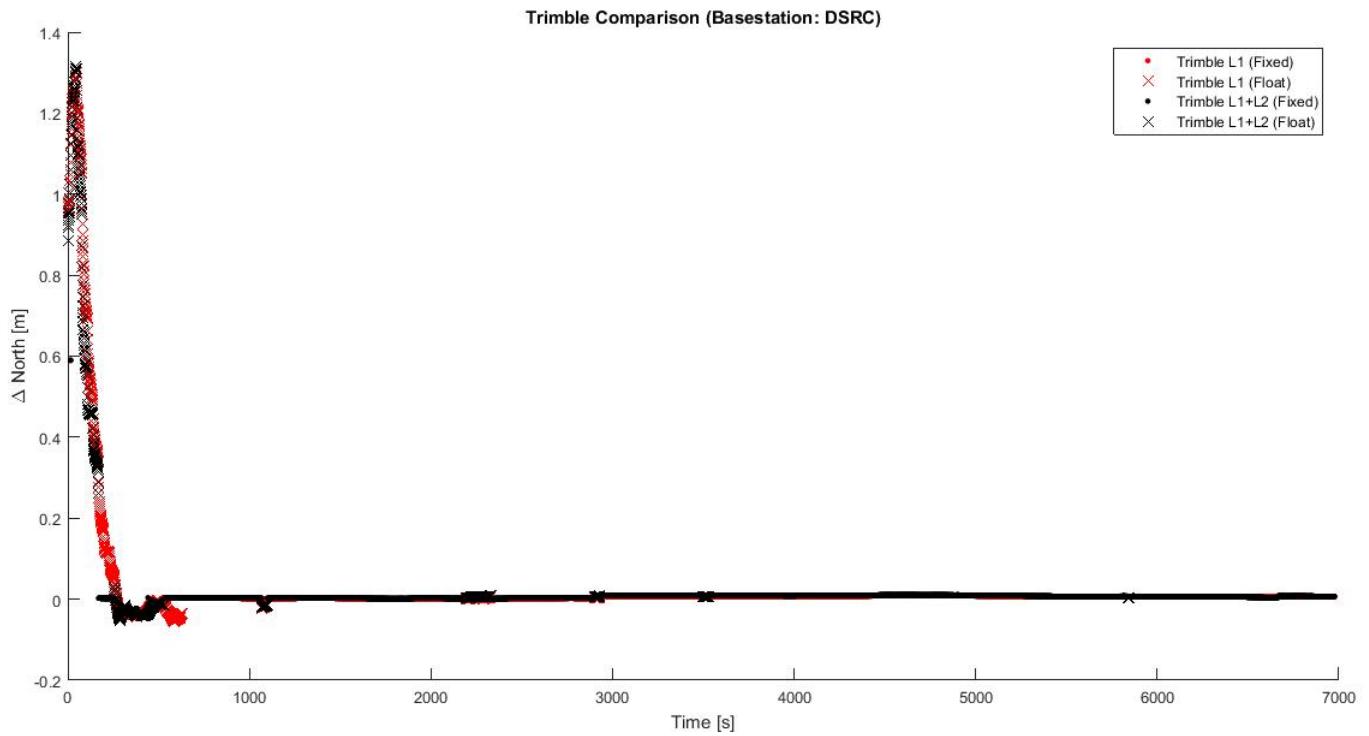


Figure 3.2.11: Difference (in m) north from estimated truth reference

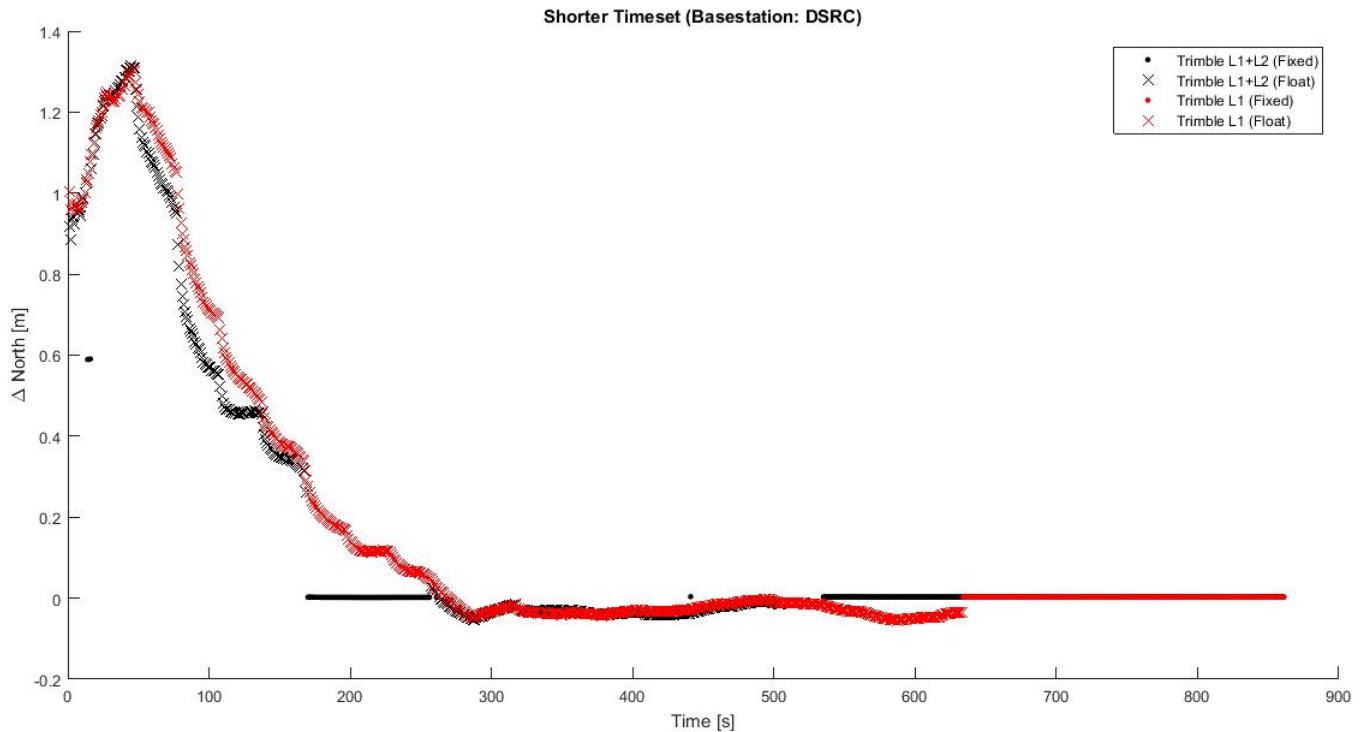


Figure 3.2.12: Difference (in m) north from estimated truth reference, first 900 seconds

3.3 Dynamic RTK Test

3.3.1 Background

The dynamic RTK test was done as a continuation of the static RTK test (section 3.2). It was performed in order to further evaluate the use of RTK for UAS. However, as no UAS was available to use the test was performed using a car to simulate a UAS in flight. The goals of the test were to:

- Asses how different receivers handle dynamic scenarios
- Asses how single-frequency RTK compares to multi-frequency RTK in a dynamic scenario
- Asses how SPP compares to RTK in a dynamic scenario
- Gather data in order to determine if a UAS could use RTK for navigation

A SPAN-SE receiver and IMU (Inertial Measurement Unit) was used for truth reference, against which all other receivers were compared.

3.3.2 Equipment

The equipment used was the same as in the static RTK experiment, with an extra addition of a Novatel SPAN-SE and IMU (described below). See section 3.2.2 for more information of the Swift Navigation Piksi, u-blox EVK-6T, Novatel Flexpak 6 and Trimble NET-R9.

3.3.2.1 Novatel Span-SE

The Novatel Span-SE is a high accuracy GNSS receiver, capable of being integrated with an IMU to produce a combined GNSS/INS (Inertial Navigation System) solution. An IMU uses gyros and accelerometers to measure forces affecting the IMU. The combined system allows the receiver to perform navigation tasks in GNSS challenged areas [38].

3.3.3 Setup

All the receivers in the car were setup to log data through their native software, unlike the static experiment where the data was recorded using RTKLIB. This meant using "Piksi Console" for the Piksi, "u-center" for the ublox and internal logging for the Trimble. This was done to ensure that other groups at the university could use the data once the test was over, as the native formats are more common to work with. GPS was the only constellation enabled on the receivers, due to technical errors with other constellations that had not been resolved from the static test.

receivers and equipment were setup in a car for the experiment. A Novatel L1+L2+L5 antenna was placed on the roof of the car using a magnetic mount to hold it in place. A custom mount for the IMU was constructed and placed on the roof. Pictures of the setup can be seen in figures 3.3.1 and 3.3.2.

A block diagram of the setup can be seen in figure 3.3.3. The only equipment that needed external power is the Span-SE and the IMU. A power inverter (which converts direct current to alternating current) was connected to the car's battery for this purpose. The antenna port of the SPAN provided power to the antenna on the roof.

3.3.4 Method

The driven route can be seen in figure 3.3.4. The route included roads in and around the University of Colorado (Boulder) campus. The path included a parking lot (a very open area), parts of a highway (an open area), major roads (open area with parts covered by trees), residential areas (with many trees covering the sky) and a parking garage (with complete sky-blockage). The total length of the route was approximately 10 km. The start and end-point was a parking area on the east side of the engineering building.



Figure 3.3.1: Test vehicle with GNSS antenna and IMU visible

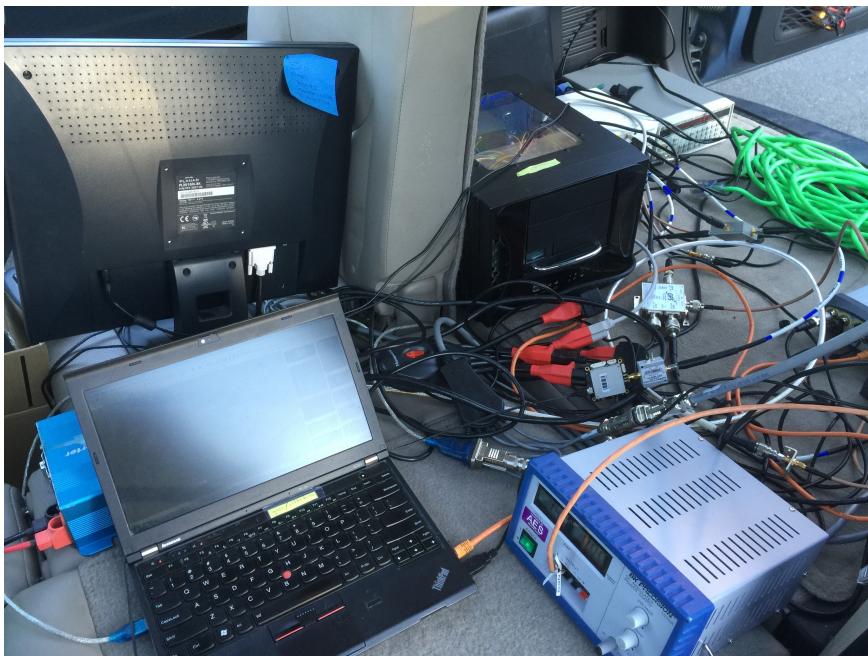


Figure 3.3.2: Inside of the test-vehicle. Some of the equipment (stationary computer and screen) was being used by another group

3.3.4.1 The Drive

To ensure that all data was being logged correctly, a custom made checklist was followed. Once the equipment was securely fastened the car was started to ensure that sufficient power would be provided to all equipment. The SPAN was then started, as this would power the antenna. The IMU was then powered up, and offsets (distances) from the antenna were entered. The IMU was then allowed to spin up and align correctly. The other receivers were then set to log data as indicated in figure 3.3.3 in blue text. Several laps were then driven around a parking lot to make sure all the equipment was operating as planned. The drive was then commenced, following the path shown in figure 3.3.4. An image from the last part of the drive in the parking garage can be seen in the lower part of figure 3.3.5. The garage was entered on the northern side, then the car was driven up four floors to the top where the sky was clear. The start of the drive can be seen in the top part of figure 3.3.5. An alley was driven through at one point. The southern part of the alley, visible at the bottom part of figure

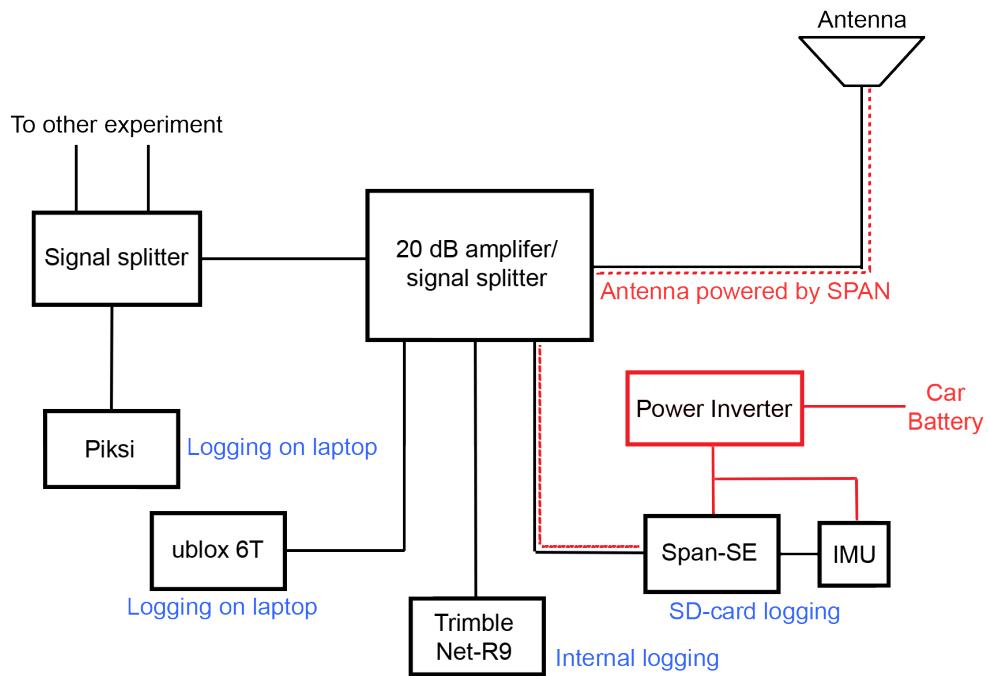
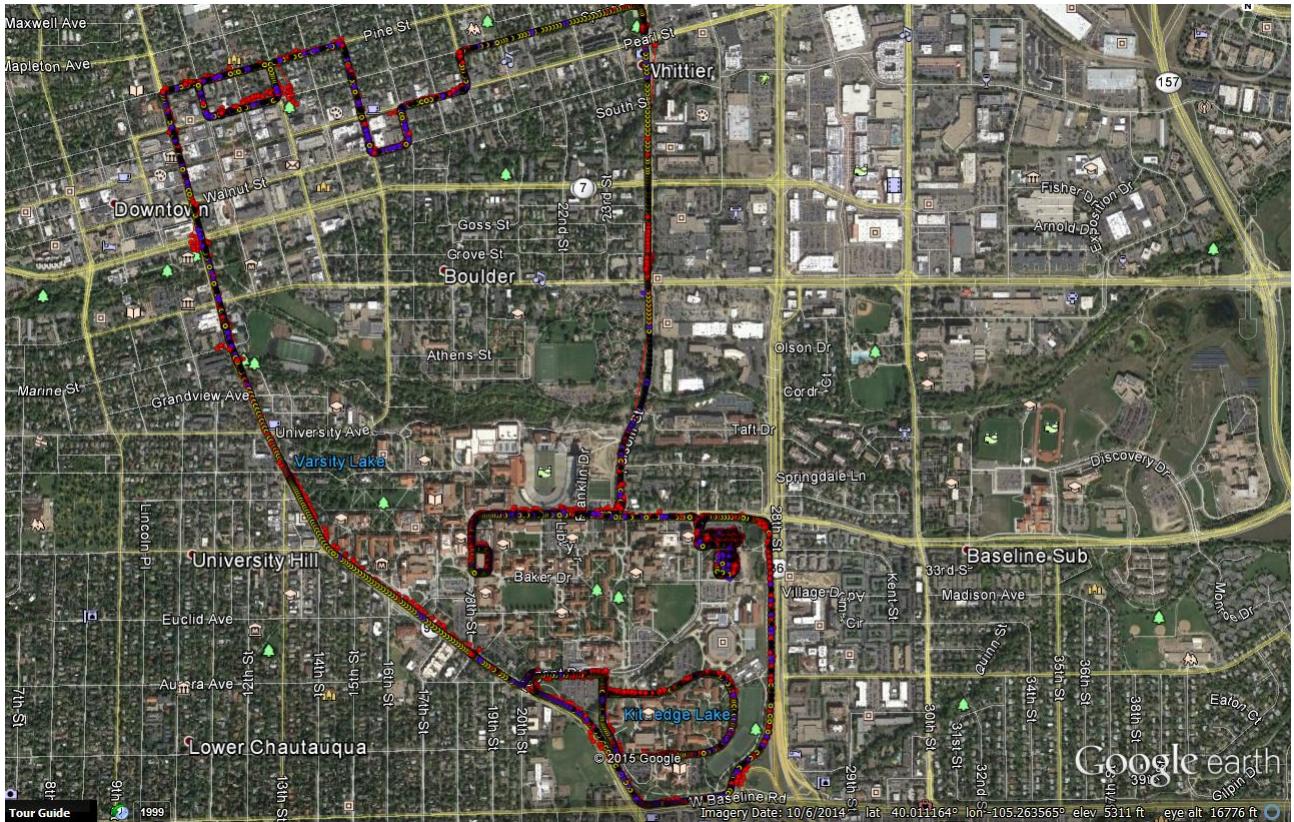


Figure 3.3.3: Block diagram of experiment setup



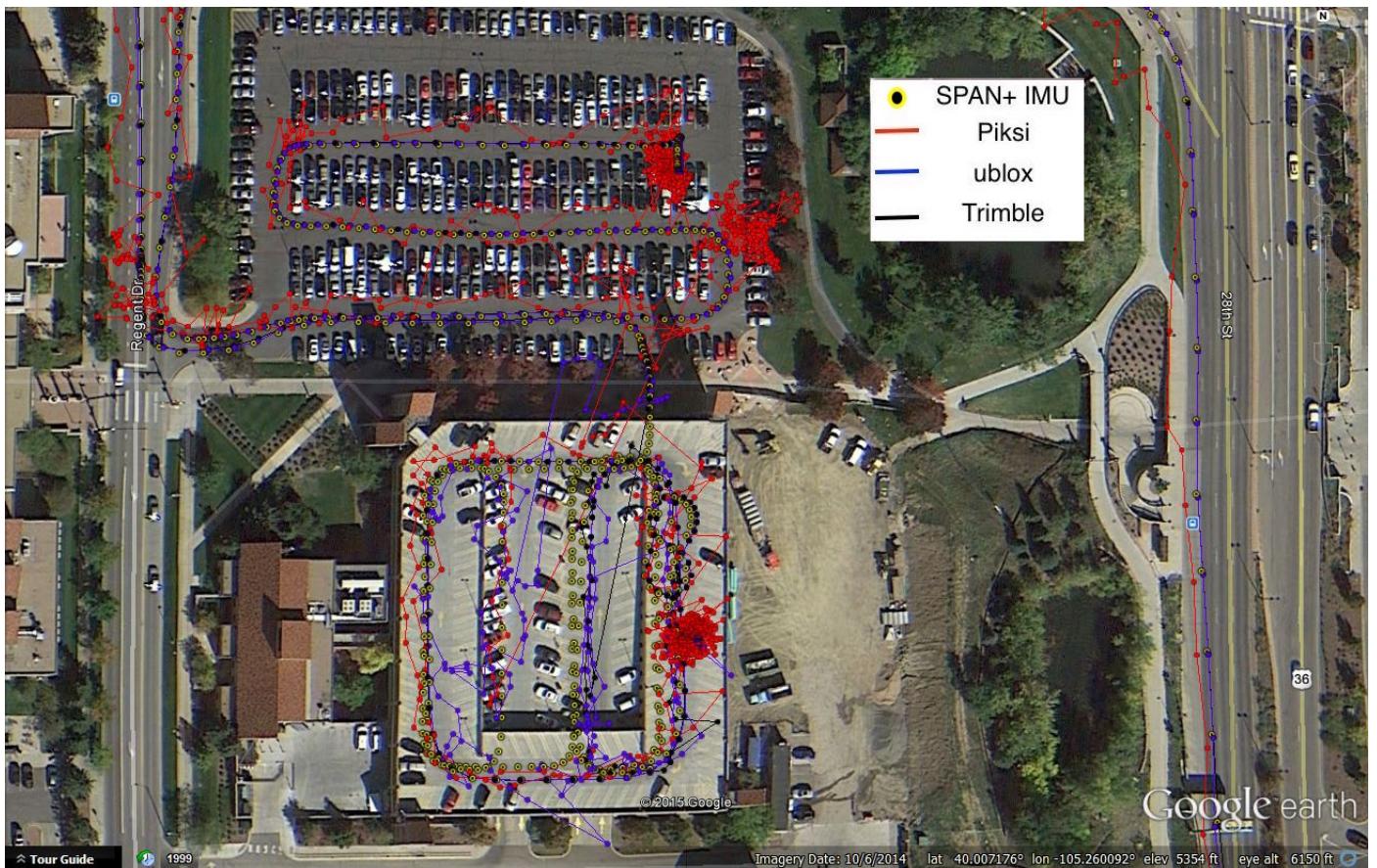


Figure 3.3.5: Logged path at beginning and end of drive

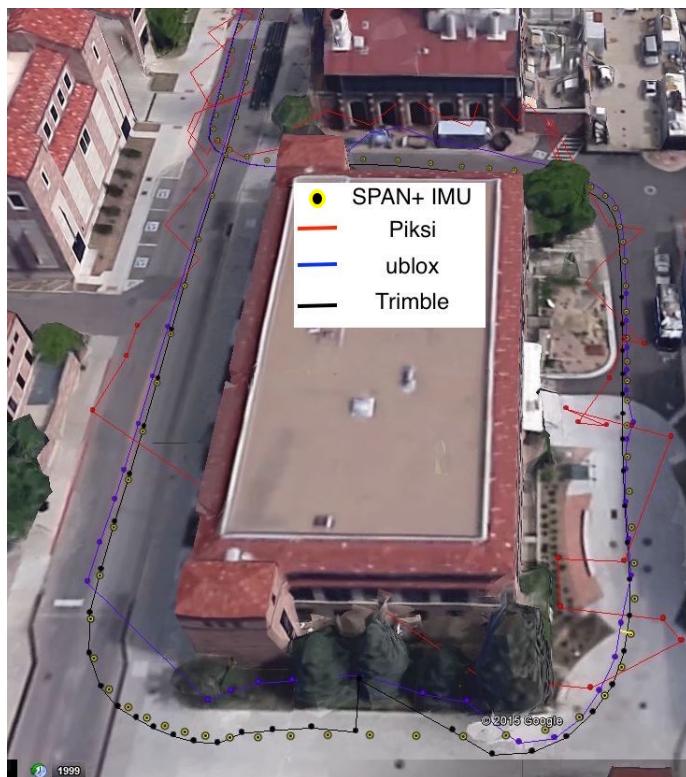


Figure 3.3.6: Logged path through the alley

3.3.4.2 Base Station Data

The Novatel Flexpak 6 (mentioned above and in section 3.2) was used to collect base station data. The receiver was set up in the same basement lab as used for the static experiment in section 3.2, and was continuously logging data using RTKLIB during the entire drive. Once the data was collected it was converted into RINEX using RTKLIB, to be able to be used for post processing. The position of the antenna was assumed to be the truth reference that was calculated for the static experiment.

3.3.4.3 Post Processing

The raw data was post processed to be used for comparisons and plotting. The method for each receiver is described below.

SPAN + IMU: The data collected from the SPAN and IMU was post processed using a program from Novatel called Waypoint. The data was processed as PPP using correction data from several nearby stations. The INS data was added into the software after the initial post processing, which smoothed and corrected the data even further (especially during periods where the sky was partially or completely blocked). The GNSS+INS solution was then exported as a data-file. This file was read by a custom made MATLAB script. The MATLAB script was able to read the data from the other receivers as well, creating a platform where all the data could be compared.

Other Receivers: The other receivers were all post processed using RTKLIB. The first step was to ensure the data was in RINEX format. The Piksi's data was recorded in this format by the logging software. The ublox data was converted to RINEX using RTKLIB's RTKCONV. The Trimble data could not be converted by RTKLIB, even after applying a patch to the software that should have enabled the function. Software provided by Trimble had to be used instead.

The data from these three receivers was then post processed as GPS RTK using RTKLIB. The Piksi and ublox only support L1, so that was the only frequency used for them. The Trimble supports L1+L2+L5, and all frequencies were used. The base station data was supplied by the Novatel Flexpak 6 as described above. The results of the post processing created a file that could be opened by the aforementioned MATLAB script.

Data from the ublox was also processed as SPP in order to compare SPP to RTK. The process was the same as described above, but without using the base station (as none is needed for SPP).

3.3.5 Results

Below the results from the different receivers can be seen. The alley was entered at around $t = 4.980 \cdot 10^5$ s and the parking garage was entered at around $t = 4.983 \cdot 10^5$ s. These points can be seen quite clearly in the figures below, as there is a noticeable increase in the 2D error.

Figure 3.3.7 show a plot of the absolute value of the 2D error in m for all receivers as compared to the SPAN+IMU solution. Each receiver is presented more clearly in separate figures below. Each point represents a measurement, and are spaced by 1 s. Figure 3.3.8 shows the 2D error in m for the Piksi receiver. The part of the drive through the alley and the parking garage is clearly visible on the graph at $4.980 \cdot 10^5$ s and $4.983 \cdot 10^5$ s. Figures 3.3.9 and 3.3.10 show the same plots but for the ublox and Trimble receiver (the ublox figure contains a SPP comparison as well).

Tabulated numerical results from the test can be seen in table 3.3.1. 2D RMS was the root mean square of only longitude and latitude. The 3D RMS was the root mean square of longitude, latitude and height. Availability was defined as

$$\text{availability} = \frac{\text{number of recorded seconds}}{\text{total number of seconds during experiment}} \quad (3.3.1)$$

RTK solution is defined as how many of the available data points that were able to be processed as RTK. Fix and float solution indicates how much of the RTK data was able to be processed as fix vs. float.

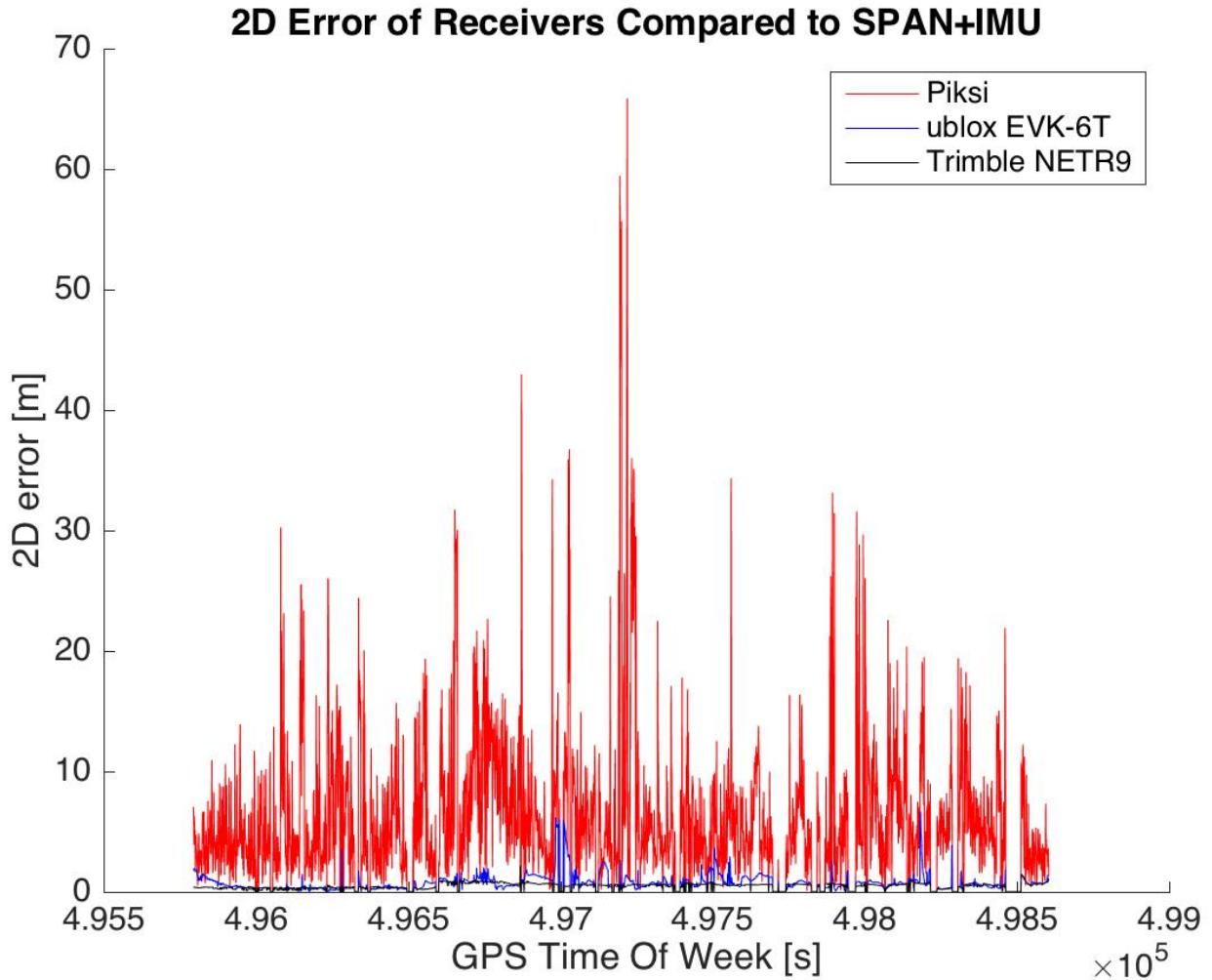


Figure 3.3.7: 2D error of all receivers compared to SPAN+IMU

Table 3.3.1: Numerical results from the dynamic RTK experiment

	Piksi RTK	ublox RTK	ublox SPP	Trimble RTK
2D RMS:	9.14 m	2.98 m	3.25 m	0.72 m
3D RMS:	23.78 m	5.99 m	6.96 m	1.43
Availability:	83 %	97 %	97 %	94 %
RTK Solution (fix or float):	17 %	100 %	N/A	98 %
Fix Solution:	12 %	13 %	N/A	61 %
Float Solution:	88 %	87 %	N/A	39 %

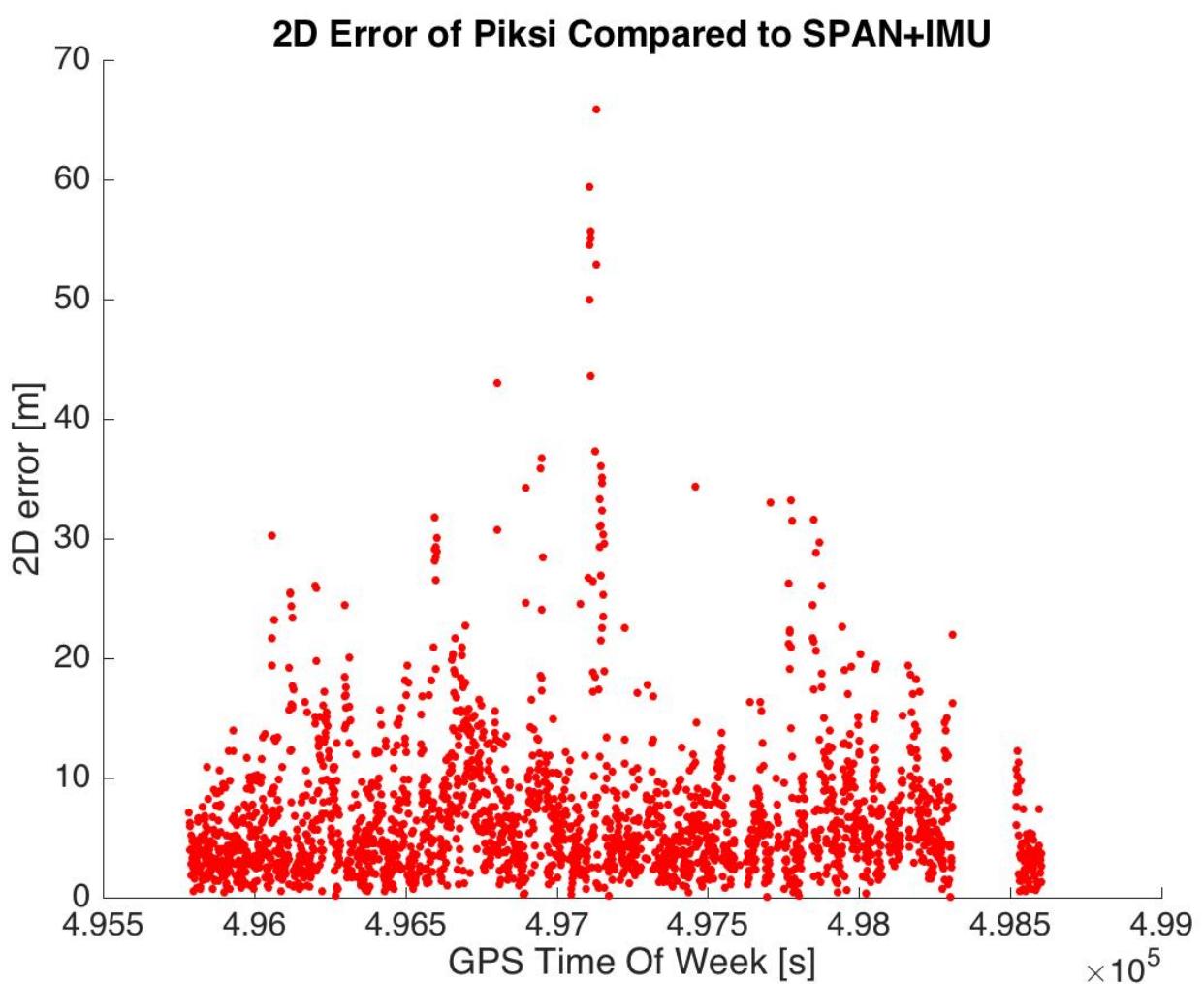


Figure 3.3.8: 2D error of Piksi compared to SPAN+IMU

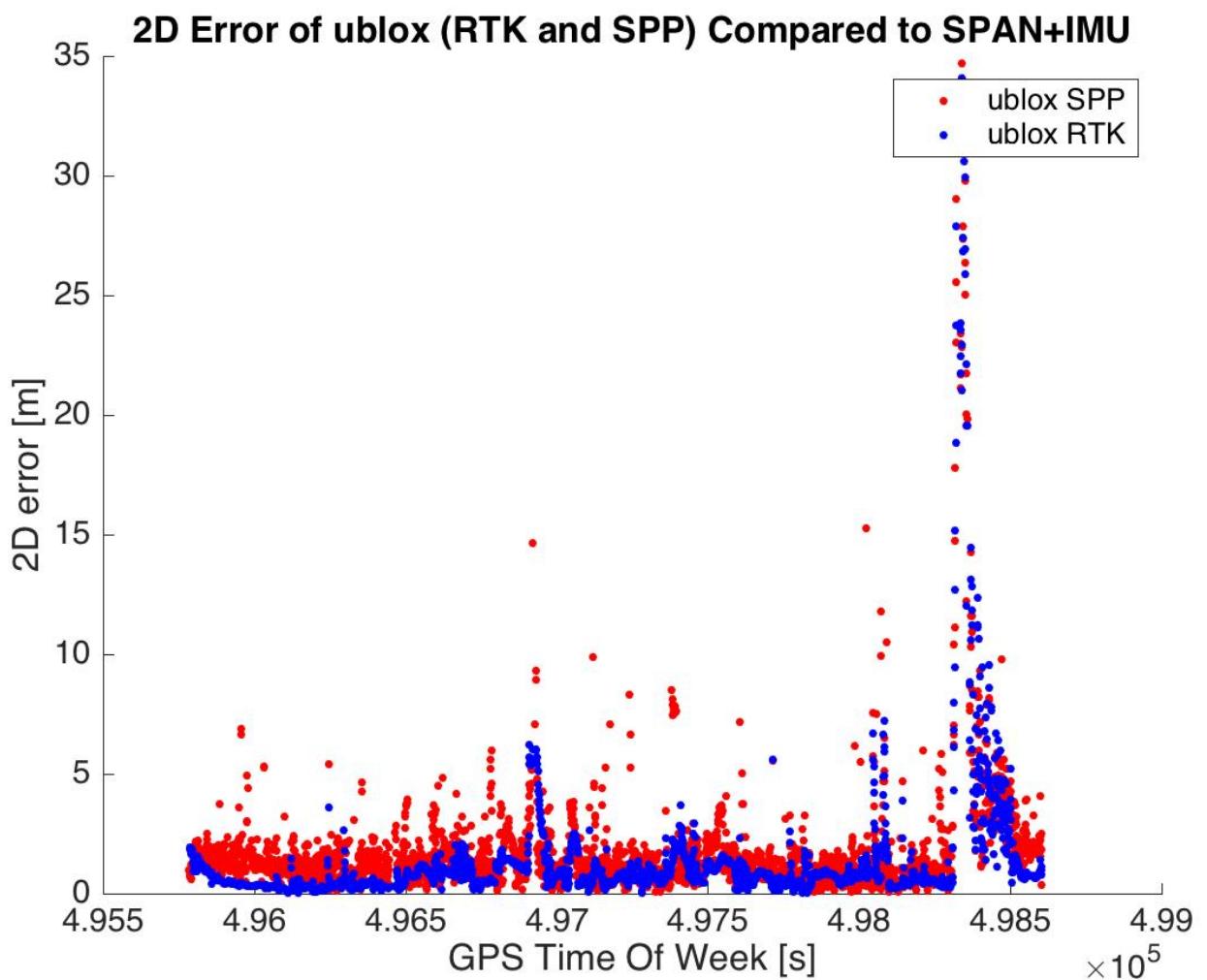


Figure 3.3.9: 2D error of ublox compared to SPAN+IMU, both RTK and SPP

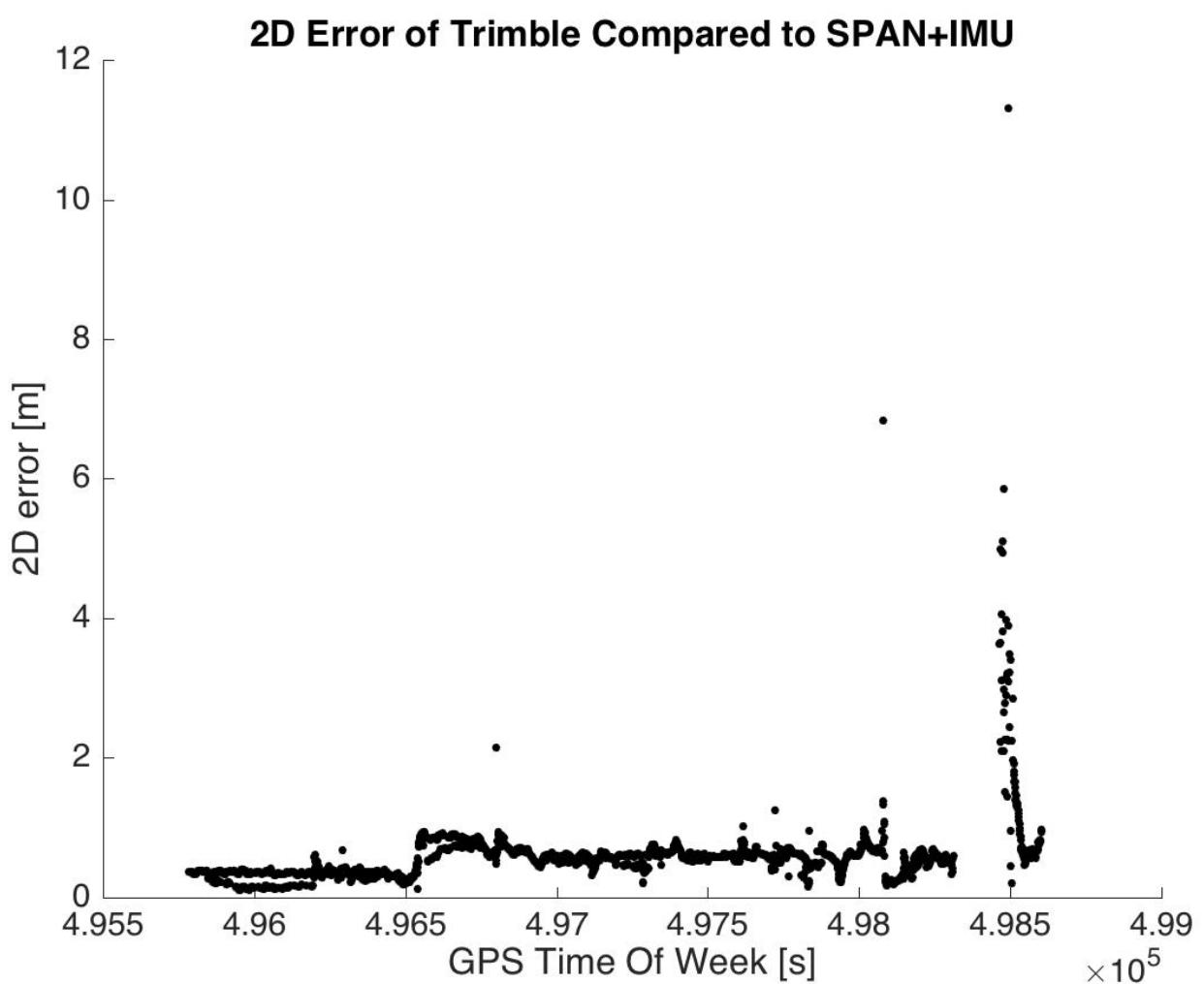


Figure 3.3.10: 2D error of Trimble compared to SPAN+IMU

Chapter 4

Conclusions and Future Work

In this chapter the conclusions from the theoretical and experimental analysis is presented. Ideas on how the experiments could have been performed differently is also given. Suggestions for future continuation of the work is given at the end of the chapter.

4.1 Conclusions

4.1.1 Conclusions from the Self Generated RFI Experiment

The presence of RFI can clearly be seen in figure 3.1.6. Turning the equipment on causes a drop in C/N_0 of approximately 10 dB, which is quite significant. This effect can also clearly be seen in table 3.1.1. Test 2 (with the antenna inside the airframe and the battery disconnected) proves that it is not just the attenuation from being placed inside the airframe that is causing this effect, and therefore it must be RFI causing the large drop in C/N_0 . As the only transmitting equipment is operating on 915 and 2400 MHz, there are no immediate harmonic effects affecting the GNSS antenna. The conclusion from this must be that there is something other than the broadcast frequencies affecting the measurement. However, the source of this interference remains unknown, as it was not tested for.

The RFI can also be seen by comparing figure 3.1.7, 3.1.8 and 3.1.9 (the results from the spectrum analyzer). The small bump in figure 3.1.7 and 3.1.8 in the center is caused by the additive power of the GPS signals. However, in figure 3.1.9 this "GPS bump" is completely disguised by the noise, and it cannot be distinguished. The noise floor has clearly risen too. The spectrum is visually more noisy as well, as there are many small spikes throughout the spectrum, whereas the spectrum in figure 3.1.7 is relatively clean or smooth.

Interesting to note is that the number of tracked satellites, figure 3.1.5, does not change between tests, even with the large drop in C/N_0 described above. Table 3.1.1 also shows that at one point during test 3 (antenna inside the airframe, battery connected and all systems running), one satellite provided a C/N_0 of only 15 dB-Hz, and still the receiver did not loose track of the satellite.

These two effects in conjuncture, the large drop in C/N_0 and the consistency in the number of tracked satellites could be a bad combination for regular GPS users, especially if the GPS is supplying navigational information for a UAS. If only the number of tracked satellites are monitored (which is a common quality evaluation when using GPS), everything would appear fine. However, due to the lowered signal strength, the navigation solution would not be as good, even though there would be no clear indication of it.

It is possible that the static conditions of the test was advantageous for the receiver. Dynamic conditions could make it more difficult for filters to aid the tracking of satellites during similar circumstances.

4.1.2 Conclusions from the Static RTK Experiment

4.1.2.1 Truth Reference

As can be seen from figure 3.2.5 and table 3.2.1, the reported antenna position is heavily dependent on which base station is used, as indicated by the grouping of coordinates by base station. Without access to survey data, the exact location of the antenna is next to impossible to determine, and this is the reason for using the average location of the calculated coordinates.

An interesting observation can also be made about the accuracy depending on which type of coordinate that is used. The coordinates vary much less in latitude than they do in longitude. This could be due to local geometry of the measurement area having less sky blockage in the north-south direction, allowing for a more accurate latitude position.

As there is nothing "better" to compare with, the average position will have to be assumed true. The truth becomes a difficult concept when dealing with mm accuracy.

4.1.2.2 Base Stations

It is difficult to draw any solid conclusions from figures 3.2.5 and 3.2.6 (the figures showing the comparisons of basestations). As stated in the section above, it is hard to determine which base station gives results that are closer to the "truth" in the two graphs since $\Delta North$ is measured from an average of the two base-stations. The DSRC base-station gives a larger initial error, however this is not really of concern as once RTK-fix has been obtained, the difference between the two stations are very small.

4.1.2.3 Receivers

From figure 3.2.7 it is clear that the Piksi's measurements are sub-par with regards to accuracy, compared to the other L1 receivers. Why the Piksi performs so much worse than the other receivers is not clear, and would need to be investigated further. Ignoring the Piksi's data, and looking closer at the first 900 seconds of data in figure 3.2.8 we see that all receivers obtain RTK fix within a minute of each other, and once a fix has been obtained the accuracy is approximately the same for all receivers.

Observing figure 3.2.9 and 3.2.10 (the comparisons of the two L1+L2 receivers) we can see some interesting effects; Both receivers are able to obtain a RTK fix relatively early (when compared to the L1 only set), with the Novatel getting a fix approx. 1,5 minutes before the Trimble. However, both receivers then loose the fix and return to float mode. This is possibly caused by satellite G12 which cuts in and out of reception around the same time (see figure 3.2.2). The L2 signal for G12 is weak (see figure 3.2.3), which agrees with the theory that G12 could be the culprit for the loss of the RTK fix, as both the Trimble and the Novatel are capable of receiving the L2 signal. Figure 3.2.11 and 3.2.12 further shows this effect, where the early lock can be seen for the L1+L2 set, whereas the L1 set only obtains lock much later. The L1 signal for G12 is stronger during this time (no figure included for L1 power), which also indicates that G12 could be causing the loss of lock for the L1+L2 set.

4.1.3 Conclusions from the Dynamic RTK Experiment

4.1.3.1 Comparison of RTK receivers

Figure 3.3.7 makes it clear that the error of the Piksi is much larger than the error of the other receivers. Not only is the error relatively large at almost all times, but there are several spikes of exceptionally large errors. The very large spikes for the Piksi in the middle of figure 3.3.8 does not seem to exist in any of the other receiver plots. At the time of the spike the car was driving through a residential area with some trees that could be covering parts of the sky, which could be a part of the explanation of the spike. Another part of the explanation could be that the Piksi was not receiving as strong of a signal as the other receivers, as the Piksi's connection was split more than the others (see figure 2.3.1). This was unfortunately unavoidable due to hardware limitations during the experiment. However, during the static experiment in section 3.2 the Piksi's connection was split as many times as the other connections, and the Piksi still performed much worse than the other receivers. A further explanation could be poor compatibility with 3rd party programs. The Piksi was the only receiver that could not be used with RTKLIB directly, which could be an indication of poor interfacing in the Piksi software.

The effects of the alley and parking garage (visible at $t = 4.980 \cdot 10^5$ s and $t = 4.983 \cdot 10^5$ s in the graphs) are interesting. The effects are most clearly seen in the Piksi's error plot (figure 3.3.8). The alley causes the minimum error to rise, while the parking garage causes the receiver to lose position completely. The effects were similar for the Trimble (figure 3.3.10), but errors were a lot smaller and the period of the signal outage in the parking garage was not as long. The ublox did relatively well in both cases, and hardly lost any data during the parking garage part, which was surprising as the sky was completely blocked for several minutes. The errors during this time rises significantly, though.

Table 3.3.1 further confirms the performance of the individual receivers. The Piksi had poor accuracy and

availability, as shown by the graphs. The Trimble had a very good accuracy, which could be expected due to the multiple frequencies used, and it also had quite good availability. However, the ublox had an even better availability, which further confirms the ublox's ability to hang on to satellites during GNSS challenged periods. The accuracy of the ublox was not as good as the Trimble though, but very good compared to the Piksi.

4.1.3.2 Comparison of SPP and RTK

Figure 3.3.9 shows the 2D error of the ublox in both SPP and RTK mode. From the figure it seems like the major difference between the two modes is the error of the position. Both modes seem to react similarly to the alley and the parking garage. Table 3.3.1 confirms this. The 3D RMS error is approx. 1 m larger for SPP, but for the 2D error the difference is only approx. 25 cm (for the whole data set). The availability is the same for both modes. So it would seem that it is only advantageous to use RTK. However, as RTK is a more complicated system with more steps involved, users may not be enticed to switch from SPP with such a small increase in accuracy.

4.1.4 Conclusions about GNSS Usage on UAS

As described in section 2.2.1 there are strict rules and regulations on performance for manned aircraft using GNSS for navigation. This is positive, as many lives can be at stake, on board and on the ground, if something were to go wrong. UAS are by their definition, unmanned, so the risk of anyone onboard perishing in an accident is not an issue. UAS still pose a threat to people and property not only on the ground, but in the air. A collision with an aircraft could be fatal. But not all UAS will operate in the vicinity of other people or aircraft, which opens the door to a unique certification opportunity for UAS GNSS. There could be several tiers of GNSS certification for UAS, and depending on the size, weight and operating environment of the UAS, different levels of performance could be required. This could open up doors for smaller/cheaper GNSS units on UAS, making the industry less complicated and more available. There is really no need for a small UAS that is only intended to operate in very remote areas at low altitude to have a high accuracy GNSS unit onboard. For a larger UAS capable of reaching high altitude it is of utmost importance to be able to guarantee separation from other aircraft.

4.1.4.1 UAS RTK

From the experiments presented in this report it seems like using RTK for UAS is very plausible. There are of course many more aspects that need to be analyzed, but the results of the ublox during the dynamic experiment shows only positive effects when compared to SPP.

One aspect that has not been analyzed that needs more attention is how RFI affects the RTK measurements, especially during dynamic flight conditions.

4.1.5 Summary of Conclusions

While it is clear that RFI is present during one of the experiments, the source of the RFI was not determined. It is likely caused by some of the electronic components onboard the UAS. As SPP positioning was used for the first experiment, it is not known how RTK measurements would be affected by the same level of interference. As RTK relies on carrier phase measurements (as opposed to code-based measurements for SPP), the results might be different. It is also possible that the choice of receiver affected the results of the RFI experiment, as the ublox performed very well with regards to maintaining tracking of satellites during the dynamic RTK experiment, when compared to other receivers.

Although RFI was not tested for during the RTK experiments, it was likely present to some degree due to the variety of electronic components in the vehicle and experiment-setup. The results of the RTK experiments showed that RTK may in fact be a viable option to use for UAS. However, as RFI was not accounted for and no test was performed with an actual UAS, more experiments are needed.

4.2 Future Work

4.2.1 Self Generated RFI

Many more experiments can be done to further the work to examine self generated GNSS RFI from UAS. These include:

- Compare individual UAS components to investigate what components constitute the greatest RFI sources
- Test several standard UAS setups for GNSS RFI
- Measure RFI during an actual UAS flight
- Compare receivers to investigate how different receivers handle RFI
- Investigate how to mitigate GNSS RFI effects through shielding, placement of antenna, removal of certain equipment or a combination of the above.

4.2.2 RTK for UAS

To further investigate how RTK can be utilized, more experiments will be required. The author recommends doing the following:

- An investigation into how RFI affects RTK measurements onboard a UAS is needed to guarantee acceptable performance during flights
- Perform UAS flights with real time RTK measurements
- Investigate the possibility of programming a flight controller to use RTK during a UAS flight, with fall-back modes to SPP for safety

Chapter 5

Summary

The goal of the thesis was to further the research and knowledge for GNSS implementation on UAS, with focus on self generated RFI caused by onboard electronics and use of high precision RTK GNSS techniques. Three experiments were performed and presented in the thesis, highlighting these areas. Results from the RFI experiment showed that RFI is indeed present on a UAS equipped with electronics typically found on smaller UAS, and the RFI is most likely caused by the onboard electronics. The RTK experiments showed promising initial results for using RTK as a primary navigation technique for UAS, however RFI effects were neglected during these experiments. Both areas need further investigation, however, and some recommendations are given on how to proceed from these initial results.

Appendix A

RTKLIB Settings for the Static RTK Experiment

A.1 Common Settings for all Receivers

The following settings were entered into RTKNAVI for the experiment:

Setting 1

<i>Positioning Mode:</i>	Static
<i>Frequencies/Filter Type:</i>	Receiver Dependent / N/A
<i>Elevation Mask (°) / SNR Mask (dBHz):</i>	15 / None
<i>Rec Dynamics / Earth Tide Correction:</i>	OFF / OFF
<i>Ionosphere Correction:</i>	Broadcast
<i>Troposphere Correction:</i>	Saastamoinen
<i>Satellite Ephemeris/Clock:</i>	Broadcast
<i>RAIM FDE:</i>	Off
<i>Excluded Satellites:</i>	None
<i>Constellations:</i>	Receiver Dependent

Setting 2

<i>Integer Ambiguity Res (GPS/GLO):</i>	Continuous / OFF
<i>Min Ratio to Fix Ambiguity:</i>	3.0
<i>Min Lock / Elevation (°) to Fix Amb:</i>	0 / 0
<i>Outage to Reset Amb / Slip Thres (m):</i>	5 / 0.050
<i>Max Age of Diff (s) / Sync Solution:</i>	30.0 / OFF
<i>Reject Threshold of GDOP / Innov (m):</i>	30.0 / 30.0
<i>Number of Filter Iteration:</i>	1

Output

Settings do not matter, as the stand-alone output will not be used

Statistics

Settings do not matter, as built in statistics aren't used

Positions

Settings do not matter, as RTK is done post processing

Files

Settings do not matter, as no external files are used

Misc

<i>Process Cycle:</i>	10 / 32768
<i>Timeout / Reconnect Interval (ms):</i>	10000 / 10000
<i>NMEA Cycle (ms) / File Swap Margin (s):</i>	5000 / 30
<i>Solution Buffer / Log Size (epochs):</i>	1000 / 100
<i>Navigation Message Selection:</i>	All
<i>SBAS Sat Selection / Monitor Port:</i>	0 / 52001

<i>HTTP/NTRIP Proxy:</i>	None
<i>TLE Data:</i>	None
<i>Sat No:</i>	None

A.2 Receiver Dependent Settings

ublox:

Setting 1:

Frequencies: L1

Constellations: GPS, SBAS

Input:

Rover: Enabled

Type: Serial

Cmd: See below

Format: u-blox

Commands to send at startup:

!UBX CFG-MSG 2 12 0 0 0 1 0

!UBX CFG-MSG 2 11 0 0 0 1 0

Commands to send at shutdown:

-

Novatel:

Setting 1:

Frequencies: L1+L2

Constellations: GPS, SBAS

Input:

Rover: Enabled

Type: Serial

Cmd: See below

Format: NovAtel OEM6

Commands to send at startup:

unlog all

log rangecmpb ontime 1

log rawephemb onnew

log ionutcb onnew

log rawwaasframb onnew

log gloephemerisb onnew

Commands to send at shutdown:

unlog rangecmpb

unlog rawephemb

unlog ionutcb

unlog rawwaasframb

unlog gloephemerisb

Trimble:

Setting 1:

Frequencies: L1+L2

Constellations: GPS,GLONASS,SBAS

Input:

Rover: Enabled

Type: TCP

Cmd: See below

Format: BINEX

Commands to send at startup:

-

Commands to send at shutdown:

A.3 Post Processing settings

The following settings were entered into RTKPOST. All settings were the same as above, except for: **Setting 1**

Frequencies/Filter Type: Receiver Dependent / Forward **Setting 2**

Integer Ambiguity Res (GPS/GLO): Continuous / OFF

Min Ratio to Fix Ambiguity: 3.0

Min Lock / Elevation (°) to Fix Amb: 0 / 0

Outage to Reset Amb / Slip Thres (m): 5 / 0.050

Max Age of Diff (s) / Sync Solution: 30.0 / N/A

Reject Threshold of GDOP / Innov (m): 30.0 / 30.0

Number of Filter Iteration: 1

Appendix B

RTKLIB Settings for the Dynamic RTK Experiment

B.1 Post Processing settings for all receivers

The following settings were entered into RTKPOST for post-processing the data.

Setting 1

<i>Positioning Mode:</i>	Dynamic
<i>Frequencies/Filter Type:</i>	Receiver Dependent / Forward
<i>Elevation Mask (°) / SNR Mask (dBHz):</i>	5 / None
<i>Rec Dynamics / Earth Tide Correction:</i>	OFF / OFF
<i>Ionosphere Correction:</i>	Broadcast
<i>Troposphere Correction:</i>	Saastamoinen
<i>Satellite Ephemeris/Clock:</i>	Broadcast
<i>RAIM FDE:</i>	Off
<i>Excluded Satellites:</i>	None
<i>Constellations:</i>	GPS

Setting 2

<i>Integer Ambiguity Res (GPS/GLO):</i>	Continuous / OFF
<i>Min Ratio to Fix Ambiguity:</i>	3.0
<i>Min Lock / Elevation (°) to Fix Amb:</i>	0 / 0
<i>Outage to Reset Amb / Slip Thres (m):</i>	5 / 0.050
<i>Max Age of Diff (s) / Sync Solution:</i>	30.0 / OFF
<i>Reject Threshold of GDOP / Innov (m):</i>	30.0 / 30.0
<i>Number of Filter Iteration:</i>	1

Output

<i>Solution Format:</i>	Lat/Lon/Height
<i>Output Header / Processing Options:</i>	ON/ON
<i>Time Format / # of Decimals:</i>	ww ssss GPST / 3
<i>Latitude / Longitude Format:</i>	ddd.dddddddd
<i>Field Separator:</i>	
<i>Datum / Height:</i>	WGS84 / Ellipsoidal
<i>Geoid Model:</i>	N/A
<i>Solution for Static Mode:</i>	N/A
<i>NMEA Interval.. :</i>	N/A
<i>Output Solution Status / Debug Trace:</i>	Residuals / Level2

Statistics

Settings do not matter, as built in statistics aren't used

Positions

<i>Rover:</i>	N/A
---------------	-----

Base Station: Lat/Lon/Height (dms/m)
40.007987390 -105.262709082 1624.6233

Antenna Type: Enabled
*

Delta-E/N/U (m): 0 / 0 / 0

Station Position File:

Files

Settings do not matter, as no external files are used

Misc

<i>Process Cycle:</i>	10 / 32768
<i>Timeout / Reconnect Interval (ms):</i>	10000 / 10000
<i>NMEA Cycle (ms) / File Swap Margin (s):</i>	5000 / 30
<i>Solution Buffer / Log Size (epochs):</i>	1000 / 100
<i>Navigation Message Selection:</i>	All
<i>SBAS Sat Selection / Monitor Port:</i>	0 / 52001
<i>HTTP/NTRIP Proxy:</i>	None
<i>TLE Data:</i>	None
<i>Sat No:</i>	None

B.2 Base-Station Settings

The following settings were entered into RTKNAVI for logging base-station data with the Novatel Flexpak 6. Only differencing setting from the settings above are shown here:

Novatel:

Setting 1:

Frequencies: L1+L2+L5
Constellations: GPS, SBAS

Input:

Base Station: Enabled
Type: Serial
Cmd: See below
Format: NovAtel OEM6

Commands to send at startup:

```
unlog all
log rangecmpb ontime 1
log rawephemb onnew
log ionutcb onnew
log rawwaasframb onnew
```

Commands to send at shutdown:

```
unlog rangecmpb
unlog rawephemb
unlog ionutcb
unlog rawwaasframb
```

Bibliography

- [1] W. Ochieng, K. Sauer, D. Walsh, G. Brodin, S. Griffin, and M. Denney, “GPS Integrity and Potential Impact on Aviation Safety,” vol. 56, pp. 51–65, 2003.
- [2] John A. Volpe National Transportation Systems Center, “Vulnerability Assessment of the Transportation Infrastructure Relying on the Global Positioning system,” 2001.
- [3] FCC, *Rules and Regulations for Title 47*, 2015, part 15.
- [4] C.-J. Huang and S.-S. Jan. (2014) Uav shipboard landing with rtk. [Online]. Available: <http://gpsworld.com/uav-shipboard-landing-with-rtk/>
- [5] R. Skulstad, C. Syversen, M. Merz, N. Sokolova, T. Fossen, and T. Johansen, “Net recovery of UAV with single-frequency RTK GPS,” in *Aerospace Conference, 2015 IEEE*, March 2015, pp. 1–10.
- [6] P. Misra and P. Enge, *Global Positioning System. Signals, Measurements, and Performance*, 2nd ed. Ganga-Jamuna Press, 2006.
- [7] E. D. Kaplan and C. J. Hegarty, Eds., *Understanding GPS, Principles and Applications*, 2nd ed. Artech House, Inc., 2006.
- [8] G. Xu, *GPS Theory, Algorithms and Applications*, 2nd ed. Springer, 2007.
- [9] T. Mai. (2015) Global Positioning System History. [Online]. Available: http://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_History.html
- [10] National Coordination Office for Space-Based Positioning, Navigation, and Timing, “Space Segment,” 2015. [Online]. Available: <http://www.gps.gov/systems/gps/space/>
- [11] IS-GPS-200H, *Global Positioning Systems Directorate. Systems Engineering & Integration*, 2013, Navstar GPS Space Segment/Navigation User Interfaces.
- [12] National Coordination Office for Space-Based Positioning, Navigation, and Timing, “Control Segment,” 2015. [Online]. Available: <http://www.gps.gov/systems/gps/control/>
- [13] IS-GPS-705D, *Global Positioning Systems Directorate. Systems Engineering & Integration*, 2013, Navstar GPS Space Segment/User Segment L5 Interfaces.
- [14] GMV, “Generic Receiver Description,” 2011. [Online]. Available: http://www.navipedia.net/index.php/Generic_Receiver_Description
- [15] ——, “Front End,” 2011. [Online]. Available: http://www.navipedia.net/index.php/Front_End
- [16] ——, “Baseband Processing,” 2011. [Online]. Available: http://www.navipedia.net/index.php/Baseband_Processing
- [17] ——, “Generic Receiver Description,” 2011. [Online]. Available: http://www.navipedia.net/index.php/System_Design_Details
- [18] M. Codrescu, “The Influence of the Ionosphere on GPS Operations,” Space Weather Prediction Center, Tech. Rep., 2007.
- [19] S. M. Shrestha, “Investigations into the Estimation of Tropospheric Delay and Wet Refractivity Using GPS Measurements,” Master’s thesis, 2003.
- [20] A. El-Rabbany, *Introduction to GPS*. Artech House, 2002.
- [21] Trimble, *GPS The First Global Satellite Navigation System*. Trimble, 2007.

- [22] GMV, “SBAS Fundamentals,” 2011. [Online]. Available: http://www.navipedia.net/index.php/SBAS_Fundamentals
- [23] USDOT and FAA, “Integration of Civil Unmanned Aircraft Systems (UAS) in the National Airspace System (NAS) Roadmap,” 2013.
- [24] ——, “Global Positioning System Wide Area Augmentation System (WAAS) Performance Standard,” 2008.
- [25] A. Rakipi, B. Kamo, S. Cakaj, V. Kolici, A. Lala, and I. Shinko, “Integrity Monitoring in Navigation Systems: Fault Detection and Exclusion RAIM Algorithm Implementation,” vol. 3, pp. 25–33, 2015.
- [26] T. Balaei, J. Barnes, and A. Dempster, “Characterization of Interference Effects on GPS Signals Carrier-Phase Error,” in *SSC 2005 Spatial Intelligence, Innovation and Praxis: The national biennial Conference of the Spatial Science Institute*, 2005.
- [27] F. Van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.
- [28] H. Hallam, “Utilizing the 3 bits of the max2769,” 2015. [Online]. Available: https://groups.google.com/forum/#!topic/swiftnav-discuss/xLE_9y_P458
- [29] Swift Navigation Inc, “Piksi,” 2014. [Online]. Available: <http://www.swiftnav.com/piksi.html>
- [30] Swift Navigation Inc., “Piksi Datasheet,” 2013. [Online]. Available: http://docs.swiftnav.com/pdfs/piksi_datasheet_v2.3.1.pdf
- [31] Maxim Integrated, “MAX2769 Universal GPS Receiver,” 2010. [Online]. Available: <http://datasheets.maximintegrated.com/en/ds/MAX2769.pdf>
- [32] T. Takasu, *RTKLIB ver. 2.4.2 Manual*, 2013. [Online]. Available: http://www.rtklib.com/prog/manual_2.4.2.pdf
- [33] T. Takasu and A. Yasuda, “Development of the Low-Cost RTK-GPS Receiver with an Open Source Program Package RTKLIB,” *international symposium on GPS/GNSS*, pp. 4–6, 2009.
- [34] ublox, *EVK-6 u-blox Evaluation Kits User Guide*, 2012. [Online]. Available: https://www.u-blox.com/sites/default/files/products/documents/EVK-6_UserGuide_%28GPS.G6-EK-10040%29.pdf
- [35] ——, *LEA-6 u-blox 6 GPS Modules Data Sheet*, 2014. [Online]. Available: https://www2.u-blox.com/images/downloads/Product_Docs/LEA-6_DataSheet_%28GPS.G6-HW-09004%29.pdf
- [36] Novatel, *Enclosures FlexPak6*, 2015. [Online]. Available: <http://www.novatel.com/assets/Documents/Papers/FlexPak6.pdf>
- [37] Trimble, *Datasheet*, 2011. [Online]. Available: http://www.ascscientific.com/NetR9_DataSheet.pdf
- [38] Novatel, *SPAN-SE User Manual*, 2012. [Online]. Available: <http://www.novatel.com/assets/Documents/Manuals/om-20000124.pdf>