Navigation with UAVs - Assessment of ultra-tight GNSS/INS integration for small UAVs

Olesen, Daniel Haugård

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Navigation with UAVs

- Assessment of ultra-tight GNSS/INS integration for small UAVs

Daniel Olesen
PhD Thesis
November 2017

Supervisors: Per Knudsen and Jakob Jakobsen

DTU Space
National Space Institute

Abstract

This thesis is focusing on improving navigation for UAVs in degraded and harsh GNSS reception environments. Applications with UAVs has encountered an enormous growth in recent years and UAV missions are destined to become completely autonomous without any need for human interaction. Autonomous operation emphasizes the need for a robust navigation system as failures could cause damage of equipment and personal injuries.

The direction taken in this thesis, has been to explore potential benefits of a ultra-tightly coupled GNSS/INS system on small UAVs. This processing strategy has shown clear improvement in GNSS signal tracking and positional accuracy in other types of applications, but has until now not been assessed for small UAVs.

Ultra-tightly coupled systems require direct access to the baseband processing algorithms within a GNSS receiver. This is generally not possible with commercial GNSS receivers, but requires a software-based GNSS receiver implementation. Furthermore, Intermediate Frequency (IF) samples need to be collected and provided as input to the software receiver. Commercial systems for IF data recording do exists, but the available options are generally too large and heavy to be used on small UAVs. To overcome this limitation, a small, low-cost and lightweight system was designed to provide IF data for subsequent experiments. The IF data-recorder was hereafter combined with a MEMS-based IMU and survey-grade GNSS receiver to collect data for the developed ultra-tightly coupled system.

Data was initially recorded and processed from a static test for initial assessment. Hereafter, two kinematic missions were performed. The first mission was completed on a 4-wheeled trolley and featured a passage under dense foliage. Finally, the system was mounted on a small UAV while a mission in harsh GNSS reception conditions was conducted. The processed data was compared with a tightly-coupled GNSS/INS solution and a survey-grade GNSS receiver. The benefits of the ultra-tightly coupled system could clearly be recognized in terms of the sustainment of GNSS signals tracked and magnitude of position errors.

Resumé (Danish)

Denne afhandling fokuserer på, at forbedre navigation med droner under forringede GNSS modtage forhold. Anvendelse af droner har undergået en kraftig stigning i de senere år og drone missioner har udsigt til at blive fuldt autonome i fremtiden. Denne udvikling fremhæver samtidig også behovet for en robust navigationsløsning, da fejl kan medføre skade på udstyr og potentielt mennesker.

Retningen i dette studie har været at undersøge potentielle fordele ved anvendelse af et ultra-tæt koblet GNSS/INS system for små droner. Denne type system har hidtil vist klare forbedringer i tracking af GNSS satelliter og positions nøjagtighed. Indtil nu har ultra-tætte GNSS/INS systemer dog ikke været testet for små droner.

For et ultra-tæt koblet system, kræves der direkte adgang til tracking algoritmer inde i GNSS modtageren. Dette er generelt ikke muligt med kommercielle modtagere, men kræver at en software GNSS modtager bliver anvendt. Derudover, kræver en software modtager adgang til IF data for processering. Kommercielle systemer til opsamling af IF data er tilgængelige, men problemet med eksisterende systemer er at vægten og størrelsen ikke er egnet til montering på små droner. Ud fra denne betragtning har det været nødvendigt at designe et kompakt og let system, således at data kunne indsamles fra en drone.

I denne afhandling blev der indsamlet data fra en statisk test for indlendende system verifikation. Derefter blev der indsamlet data fra to kinematiske missioner. Den første mission bestod af en dataindsamling fra en rullevogn, hvor vognen passerede en sektion overskygget af trækroner. Herefter, blev dataopsamlingssystemet monteret på en drone og en mission under svære GNSS modtage

forhold blev udført. De processerede data blev sammenholdt med en tæt-koblet processering samt data fra en geodætisk GNSS modtager. Fordelene ved den ultra-tætte integration fremstod tydeligt, både i forhold til at vedholde tracking af de enkelte satelliter under degraderede modtageforhold samt i forhold til positionsbestemmelsen.

Acknowledgements

This dissertation would not have been the same without a number of people guiding and supporting me along the way. In this regard, I am grateful for the excellent guidance and sparring I received at my external research stay at the PLAN group in Calgary. Dr. Mark Petovello is acknowledged for his role as my temporary supervisor during this stay and his vast knowledge in GNSS and INS integration as well as GNSS software receivers has really helped me grasp some important aspects and accelerated my research. I am also grateful for the warm reception I got from my fellow PhD students Paul, Chandra, Bernhard and Rakesh during this time.

I am also grateful for the guidance and trust I have received from my supervisors. Per Knudsen always helped steer in the right direction and provide a sense of reality throughout the study. Jakob Jakobsen was always shimmering with enthusiasm and provided encouragement throughout the study. Most of all I am grateful for their trust and for providing me with the opportunity to initiate the study to begin with.

I would also like to acknowledge the Innovation Fund Denmark for the for financial support of this study as a part of the Smart-UAV project. The Idella foundation is acknowledged for the financial support I was offered during my external stay.

Finally, I am grateful for my family and friends, who have always been supportive and understanding during the process.

Nomenclature

C/N_0	Carrier-to-noise density ratio
ADC	Analog-To-Digital Converter
AGC	Automatic Gain Control
BOC	Binary Offset Carrier
BPSK	Binary-Phase Shift Keying
CAF	Cross Ambiguity Function
CDMA	Code Division Multiple Access
CW	Continuous Wave
DCM	Direction Cosine Matrix
DFT	Discrete Fourier Transform
DLL	Delay Lock Loop
DRC	Doppler Removal and Correlation
DSSS	Direct Sequence Spread Spectrum
ECEF	Earth-Centered Earth Fixed
ECI	Earth-Centered Inertial
EIRP	Effective Isotropic Radiated Power
EKF	Extended Kalman Filter

ENU	East-North-Up
FDMA	Frequency Division Multiple Access
FLL	Frequency Lock Loop
FSPL	Free Space Path Loss
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPST	GPS Time
IF	Intermediate Frequency
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
KF	Kalman filter
L.C.	Loosely Coupled
LHCP	Left Hand Circular Polarized
LNA	Low-Noise Amplifier
LO	Local Oscillator
LOS	Line-Of-Sight
LS	Least Squares
LSFR	Linear Feedback Shift Register
MEMS	Microelectromechanical systems
MEO	Medium Earth Orbit
ms	millisecond
NCO	Numerical Controller Oscillator
NED	North-East-Down
NLOS	None Line-Of-Sight
PLI	Phase-Lock Indicator
PLL	Phase Lock Loop
PRN	Pseudo Random Noise

PSD	Power Spectral Density
PVT	Position, Velocity and Time
RF	Radio Frequency
RHCP	Right Hand Circular Polarized
RLG	Ring-Laser Gyro
RTK	Real-Time Kinematic
SBC	Single Board Computer
SIS	Signals in Space
SNR	Signal-to-Noise Ratio
SPI	Serial Peripheral Interface
SV	Satellite Vehicle
T.C.	Tightly Coupled
TEC	Total Electron Content
TOA	Time-Of-Arrival
U.T.C.	Ultra-Tightly Coupled
UAV	Unmanned Aerial Vehicle
VDLL	Vector Delay Lock Loop
XOR	eXclusive OR
ZARU	Zero Angular Rate Update
ZVU	Zero Velocity Update

Contents

Chapter 1

Introduction

Navigation with Unmanned Aerial Vehicles (UAVs) could cover a wide area of research topics. The increasing popularity and availability of small and affordable UAVs has lead to a growing number of applications for UAVs within e.g. safety & rescue operations, intelligent farming, visual inspections, remote sensing, delivery of services and surveying & mapping missions. Depending on the UAV application, there might be different requirements to the accuracy and robustness of the navigation system used. In many cases, a standard code-based Global Navigation Satellite System (GNSS) receiver might deliver sufficient accuracy. In other applications, the UAV could be required to operate indoors, in which case GNSS-based positioning would be significantly degraded or completely unavailable altogether and the UAV would have to rely on other navigation aids. In the case of surveying and mapping, precise positioning is often required with accuracies to within a few centimeters. A key challenge in UAV navigation is that the vehicle is often subject to high vibration levels and can encounter rapid movements. For other navigation objects, it is not uncommon to include constraints in the navigation filters for, e.g., anticipated height variations during the mission, dominant directions of travel, steering geometry of vehicles etc. This liberty is in general not possible for a UAV.

The motivation for this project originates from a collaboration with researchers from DTU Environment about using a small, lightweight commercial UAV to perform measurements of water-levels in lakes and rivers [BJO+17]. Historically, water-level measurements have usually been performed from stationary gauge stations and more recently by airborne or spaceborne altimetry. The advantage of using UAVs compared to the traditional methods is that it is possible to obtain measurements with a much higher spatial and temporal resolution. A typical use-case scenario is illustrated in Figure 1.1.

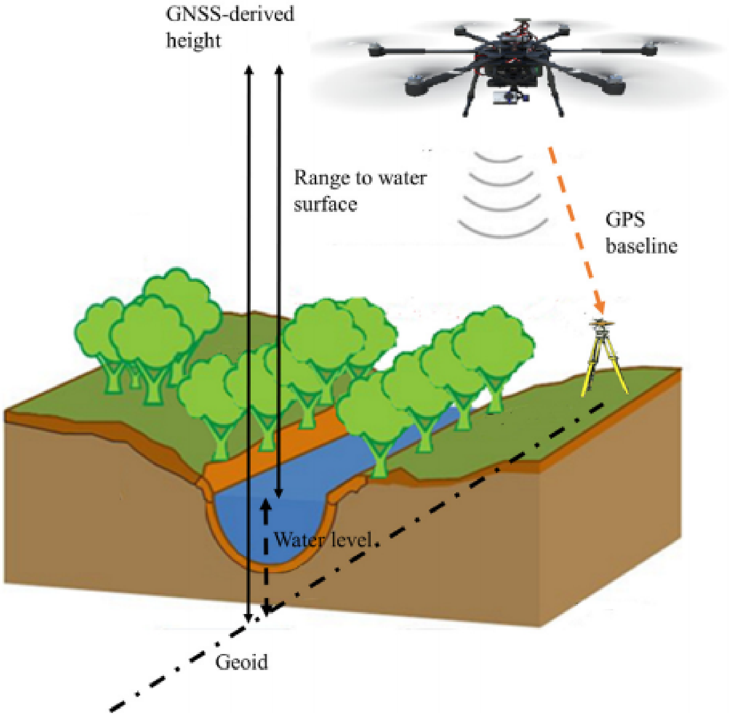
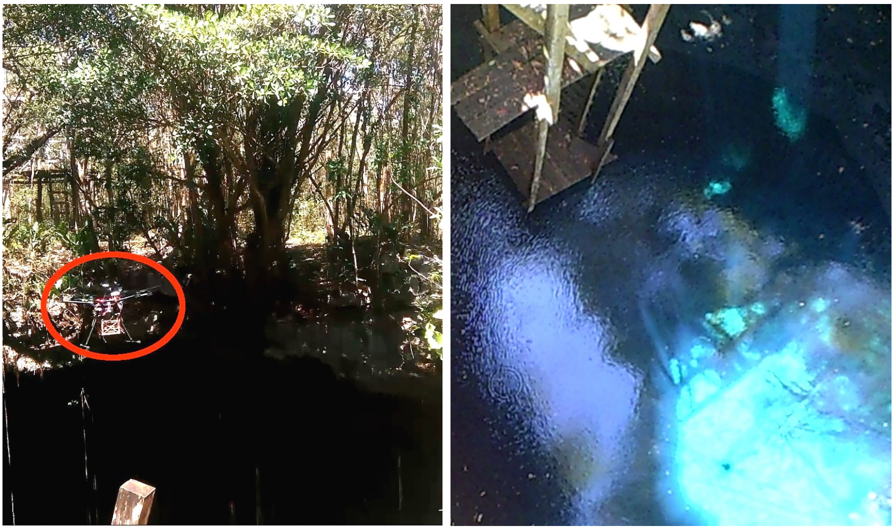


Figure 1.1: Example of how to retrieve water-levels from a UAV with radar ranging and RTK-GNSS. Figure from [BJO+17].

In this example, a Real-Time Kinematic (RTK) GNSS setup is used, where observation data from a GNSS reference station is transmitted to the onboard (roving) GNSS receiver to obtain a precise differential carrier-phase solution. This setup is sufficient in many cases, but requires that the roving receiver has an unobstructed, clear view to the GNSS satellites. This is not always the case as the lake or river might be shadowed by vegation or large tree canopies, which would greatly reduce the obtainable accuracy. A real-world example of this was encountered at a mission in the Yucatán Peninsula of Mexico. The UAV was supposed to measure a water-level in a cenote (sinkhole), which was completely shadowed by vegetation. A picture of the scenario is shown in Figure 1.2.



(a) UAV seen from the cenote rim. (b) Nadir-oriented picture taken from UAV.

Figure 1.2: UAV measuring water-level in mexican cenote. Photo credits: Filippo Bandini.

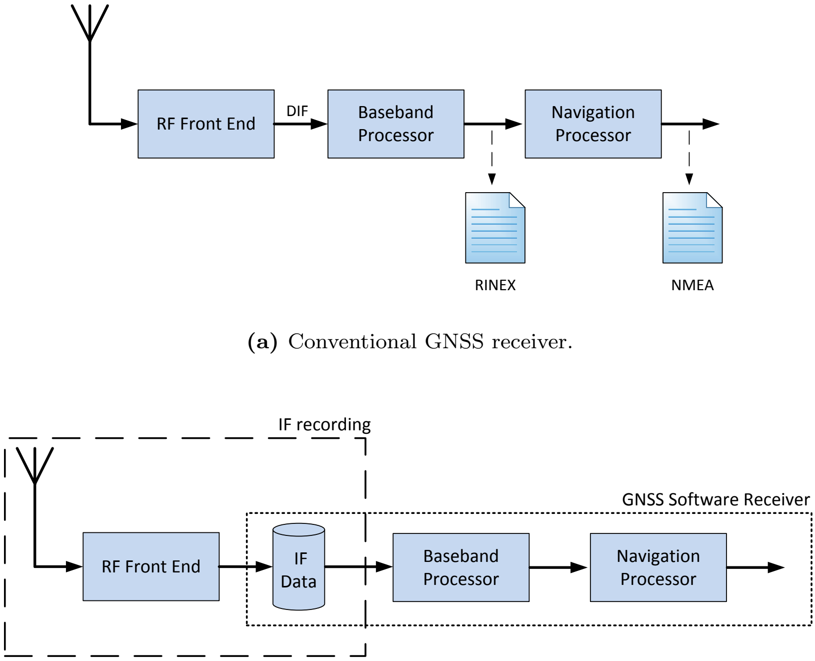
From this particular mission, the GNSS reception was degraded to a point where less than four satellites was tracked continuously, meaning that the receiver was unable to determine a position.

A well known strategy for improving robustness is to combine GNSS with a Inertial Navigation System (INS). Without going to details at this point, the INS is used to constrain the GNSS solution, so that large position outliers are mitigated. An INS is based on measurements from an Inertial Measurement Unit (IMU), which contains an ensemble of accelerometers and gyroscopes. The quality of an INS is largely dependent of the grade of the IMU. Moreover, high quality IMUs tends to be heavy and large, which poses a significant problem for lightweight UAVs.

The UAV which has been used in this study has a payload capacity of 1.5 kg. The ranging technology, which consists of a radar and sonar takes up more than half of that capacity. This has greatly reduced the number of available high-end commercial options for the UAV navigation system.

The main focus in this study has been to investigate GNSS performance in degraded reception environments during UAV missions. This has been accomplished by developing a post-processing software GNSS receiver and a compact and lightweight data-acquisition system to capture downconverted GNSS Intermediate Frequency (IF) samples. Although commercial (hardware) receivers in general are very sophisticated, the major drawback, seen from a research perspective, is that they in essence can be perceived as black boxes producing only a very limited amount of output. In contrary, a software receiver gives insight into all intermediate processing steps from signal reception at the antenna to the generated observables. GNSS software receivers have become increasingly popular in the GNSS research community, since the first implementation from Ohio University was reported back in 1997 [Ako97].

A conceptual comparison of a typical GNSS receiver and a software implementation is illustrated in Figure 1.3.



(a) Conventional GNSS receiver.

(b) Post-processing software GNSS receiver.

Figure 1.3: Conventional GNSS receiver and post-processing software GNSS receiver. Figure adopted from [POJK16].

In a commercial (hardware) receiver, the signal processing chain consists of a Radio Frequency (RF) front-end, which receives the electromagnetic signals from the satellites and downconverts (mix) the original signal to a much lower IF and performs digitization. Hereafter the signal is passed to a baseband processor which aligns the received signal with a receiver-generated replica and continuously tracks the code-phase and Doppler of the signal. The tracked signals can then be used to form traditional observables; pseudoranges, carrier-phase (accumulated Doppler range) and instantaneous Doppler. These observables (or measurements) can subsequently be used to generate a user Position, Velocity and Time (PVT) solution. A software receiver, basically performs the same operational steps, but the main difference is that processing is implemented in software rather than hardware. This offers a much higher level of control and allows for verification of different types of baseband processing algorithms. As the IF data is recorded and stored, this also provides the possibility to reprocess the same dataset infinitely with a variety of algorithms and settings to optimize the obtained solution. [LB16] lists additional benefits of IF data storage compared to that of traditional GNSS observables.

Commercial solutions to record raw IF samples exists, but due to the restricted weight and size constraints imposed from the used UAV, it was necessary to design our own compact and lightweight system.

The main reason for using a software receiver, has been to explore advanced GNSS tracking algorithms to improve the availability and quality of GNSS measurements in degraded environments. A typical GNSS receiver tracks code-phase and carrier-frequency and/or phase of each satellite individually, which provides satisfactory results under normal reception conditions. A more sophisticated architecture, is that of a vector receiver, which uses a feedback from the obtained position and velocity solution, to reduce noise on all channels [Jr.96]. In addition a vector receiver has also proven to increase tracking sensitivity in weak signal environments [LB07]. The drawback of this strategy is that if one channel becomes corrupted, this could lead the entire solution to fail. A more robust approach is to integrate INS measurements into the navigation solution, as this would give an additional redundancy for the navigation solution, when used to steer the individual channels. This approach is known as ultra-tight (or deep) GNSS/INS integration [AL03]. This type of integration, have proven to outperform classical INS and GNSS integration strategies, such as loose- and tight couplings in, e.g., urban environments [LT13], under foliage [POL09] and during scintillations [TFKK14]. In addition ultra-tight implementations have also showed greater immunity to jamming [KJRB$^+$14]. Ultra-tightly coupled GNSS/INS has also been investigated in terms of differential carrier-phase positioning and been reported to outperform classical methods in this regard [POL08a], [POL08b]. Existing research of ultra-tight integration algorithms have largely been based on automotive, pedestrian and (larger) aircraft experiments.

This dissertation extends the use of ultra-tight GNSS/INS integration to the rising field of UAV navigation, which is considered to be the main contribution of the thesis. In addition to the developed GNSS IF data-recorder and post-processing GNSS software receiver, a small real time GPS software receiver was also developed during this study, which potentially can be used for a real-time implementation of the ultra-tight GNSS/INS receiver algorithms.

The dissertation summarizes and extends the results presented in following peer-reviewed papers:

1. Olesen, D., Jakobsen, J. and Knudsen, P. (2016). Low-cost GNSS sampler based on the beaglebone black SBC. *In Proceedings of the 8th ESA Workshop on Satellite Navigation Technologies, and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*
2. Olesen, D., Jakobsen, J., Knudsen, P. (2017). Ultra-Tightly Coupled GNSS/INS for Small UAVs. *In Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, pp. 2587-2602.
3. Olesen, D., Jakobsen, J. and Knudsen, P. (2015). Software-Defined GPS Receiver Implemented on the Parallella-16 Board. *In Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, pp. 3171-3177

Furthermore, the following co-authored paper is also referenced as it provides additional background to the water-level measurement application with UAVs.

- Bandini, F., Jakobsen, J., Olesen, D., Reyna-Gutierrez, J.A., Bauer-Gottwein, P. (2017). Measuring water level in rivers and lakes from lightweight unmanned aerial vehicles. *Journal of Hydrology.* 548:237-250

1.1 Thesis Outline

The thesis is structured in the following way:

Chapter 2 gives a brief overview of the generation of Signals-in-Space (SIS) from the various constellations, but mainly focusing on the GPS L1 C/A signal. The main propagation effects are listed, such as the free space path loss, ionospheric dispersion and trophosheric delay. Finally, degrading effects from the surroundings are considered.

Chapter 3 covers most aspects of GNSS receiver processing. This chapter initially considers the RF front-end, then a brief description of satellite acquisition algorithms is given. Code- and carrier tracking fundamentals are described in terms of classical Phase Lock Loop (PLL) and Delay Lock Loops (DLL). An Extended Kalman filter-based alternative to conventional tracking methods is described afterwards. Typical algorithms for extracting receiver measurements and calculating a position and velocity solution is then given. The chapter concludes by considering the fundamental aspects of a vector tracking GNSS receiver.

Chapter 4 provides a brief overview of Inertial Navigation Systems. This includes the mechanization equations for position, velocity and attitude determination. The chapter concludes by providing an error model of accelerometers and gyroscopes.

Chapter 5 is concerned with sensor fusion or integration of GNSS and INS navigation systems. Classical implementations, such as loose- and tight-couplings are described and finally the basics for ultra-tight integration.

Chapter 6 has been dedicated to describe the software and hardware developements which have been carried out during this study. Initially, the design of the GNSS IF sampler, based on a single board computer (SBC) and a commercial RF front-end is described. Hereafter, the implementation of a ultra-tightly coupled GNSS/INS receiver is described.

Chapter 7 presents the obtained results and provides an subsequent analysis. Initially, a static test was performed on the developed receiver. In this test, the recorded IF samples were artificially degraded with noise and a performance comparison between a ordinary receiver implementation and a vector receiver was made. Hereafter, the ultra-tightly coupled GNSS/INS receiver was validated on a trolley test. The chapter concludes with analysis of a UAV experiment, where the UAV was flying in a degraded reception environment.

Chapter 8 present the conclusions of this research and considers future perspectives.

Chapter 2

GNSS Signals

GNSS is a common term for all satellite navigation systems with global coverage. As of today, there are 2 fully operational systems; The U.S. Global Positioning System (GPS) and the Russian GLONASS system. Two additional systems are furthermore under construction; The Galileo system from the European Union and BeiDou from China. Both of these system are currently providing initial services and expected to be fully operational within a few years.

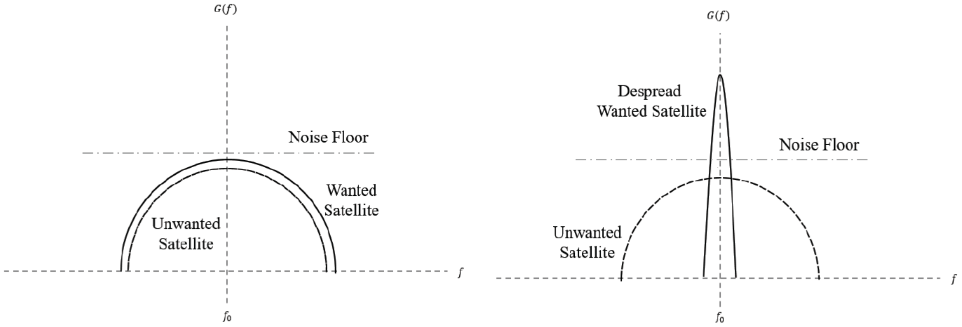
Similar for all systems is the use of Medium Earth Orbit (MEO) satellites, orbiting around the earth in approximately 20.000 kilometers altitude. The number of orbital planes and satellites vary between the different constellations.

The fundamental principle in satellite positioning is the concept of Time-Of-Arrival (TOA) measurements from multiple satellites. Each satellite vehicle (SV) signal contains a unique signature, which makes it distinguishable from other SVs. In addition the transmission of signals from the SVs is synchronized using atomic-scale clocks. In order for the GNSS receiver to obtain a position, at least 4 SVs should be tracked simultaneously. This is required to solve for the user (3D) position and the receiver clock error.

2.1 GNSS Satellite Signals

All GNSS signals makes use of direct sequence spread spectrum (DSSS) modulation. The underlying principle in this modulation technique is to make a wide-band spreading of a low bandwidth datachannel. This is accomplished by modulating the data with a repetetive Pseudo-Random Noise (PRN) sequence. A PRN sequence is purely deterministic, but would appear in the frequency domain as white noise. Some of the benefits in this modulation scheme is robustness to intentional/unintentional jamming. In addition the repetitive spreading sequence also posses good properties for timing applications, due to the higher chip-rate and a repetitiveness of the spreading (PRN) code [Skl12].

GPS, Galileo and Beidou all use a channel-access method known as Code-Division Multiple Access (CDMA). This method allows all SVs within a constellation to use the same carrier frequency (channel) for transmission. Fundamentally, each SV is assigned its own unique PRN code which can make the signals distinguishable for the receiver upon reception. An illustration of the principle can be seen in Figure 2.1.



(a) Power-Spectrum after signal spreading. **(b)** Power-Spectrum after signal despreading.

Figure 2.1: Principle of CDMA.

In the power-spectrum in Figure 2.1a, two SVs are shown to lie on top of each other - both below the thermal noise floor. The receiver demodulates the signal from one of the SV's, by correlating the received signal with the known PRN sequence for that particular SV. This will recreate the original narrow-band data spectrum as illustrated in Figure 2.1b, but not influence the spectrum for the other SVs.

For GLONASS, a channel access method known as Frequency-Division Multiple

Access (FDMA) is employed. For FDMA, the spreading codes for each SV is identical, but the signals are instead separated by using different carrier-frequencies. The downside to this method, is that a GLONASS receiver usually needs to have a higher reception bandwidth increasing the price and complexity of the receiver. The next generation of GLONASS SVs (GLONASS-K) does however feature civilian CDMA signals, thus providing easier interoperability with the other constellations.

2.1.1 GNSS Frequency Bands

The different constellations have been assigned different frequency bands, however some systems share similar slots.

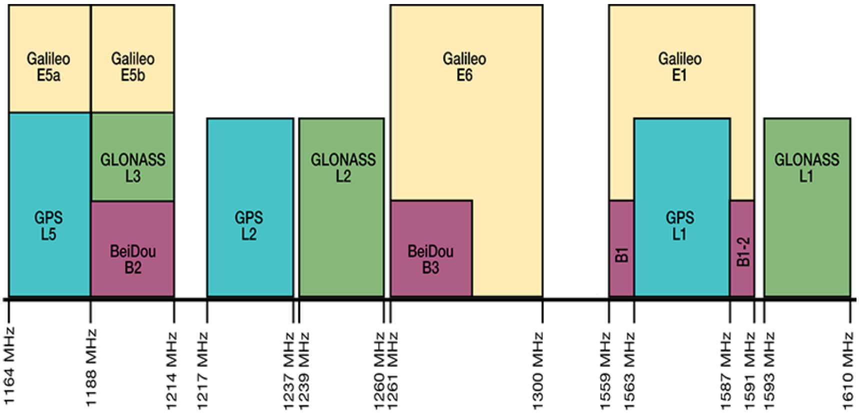


Figure 2.2: GNSS frequency bands. Figure adopted from [Inc17]

In Figure 2.2 an illustration of the different frequency slots for GPS, Galileo, Beidou and GLONASS is shown. Notably for all systems, is that each SV transmit on 3 distinct carrier frequencies. Traditionally, GPS and GLONASS was each allocated their own slots in the L1 and L2 bands, but more recently a third carrier has been introduced; GPS L5 and GLONASS L3. Beidou and Galileo is similarly using 3 different carrier-frequencies.

2.1.2 Transmitted Signals

As explained in the previous section, originally two carrier-frequencies was used for GPS and GLONASS SVs. Back then, two signals was transmitted; a civialian

Coarse / Acquisition (C/A) code signal and a military Precision (P) code signal. Following a modernization effort of GPS and GLONASS, the number of signals have now increased substantially.

The receiver developed for this research have solely been utilizing the GPS L1 C/A code. This signal is described in further detail in the next section.

2.1.2.1 GPS L1 C/A

The GPS L1 C/A code signal is transmitted on the GPS L1 band with a carrier-frequency of 1575.42 MHz. The signal is comprised of a repetitive PRN sequence, denoted the C/A code which has a length of 1023 chips and a chiprate of 1.023 Mchips/s. Following from this, each sequence has a duration of exactly 1 millisecond (ms). The C/A code is unique for each SV and is generated using a Linear Feedback Shift Register (LFSR). Details of C/A code generation can be found in [GPS06]. The C/A code is further modulated with the navigation message of the SVs. This message contains SV ephemeris, clock information and signal propagation parameters. The data rate of the navigation message is 50 bits/s. This modulation is done using modulo-2 addition also known as eXclusive OR (XOR). Finally, the combined C/A code and navigation message is modulated onto the RF carrier by the use of Binary-Phase Shift Keying (BPSK). An illustration of the modulation process is shown in Figure 2.3.

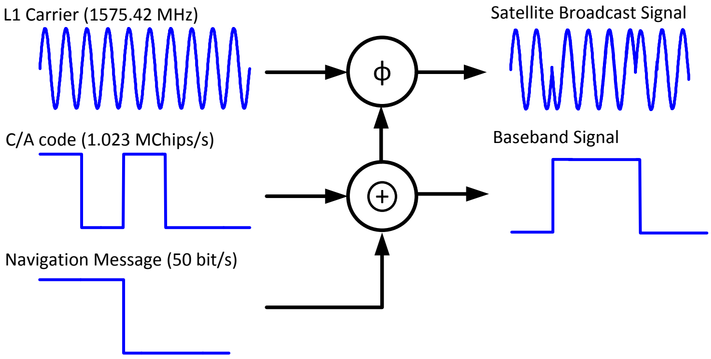

L1 Carrier (1575.42 MHz)
Satellite Broadcast Signal
ϕ
C/A code (1.023 MChips/s)
Baseband Signal
⊕
Navigation Message (50 bit/s)

Figure 2.3: Modulation of GPS L1 C/A signal (not to scale).

Mathematically, the GPS L1 C/A broadcast signal can be defined as:

$$S_k(t) = \sin(2\pi f_L t) \cdot (S_{CA,k}(t) \oplus S_{nav,k}(t)) \tag{2.1}$$

where f_L is the GPS L1 carrier-frequency of 1575.42 MHz, $S_{CA,k}(t)$ is the C/A code sequence (spreading code) and $S_{nav,k}(t)$ the navigation message for the k-th SV.

Equation 2.1 implicitly assumes, that the values of the C/A code sequence and navigation-bit can be either $\{+1, -1\}$. It should be noted, that the GPS L1 carrier is also modulated with the military P-code. The P-code is however transmitted phase-shifted with respect to the C/A code hence the two signals can coexist. In fact, following the modernization effort of GPS, 5 signals are planned to be transmitted on the same carrier. This is visualized in Figure 2.4.

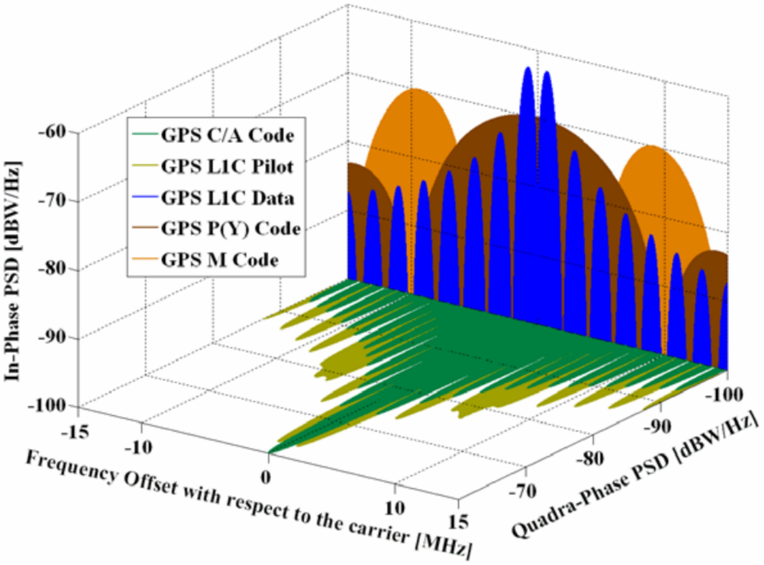


Figure 2.4: I/Q Spectrum of GPS L1 signals. Figure adopted from [Rod08]

This is possible, since the SVs transmit using quadrature channels, utilizing Quadrature Phase Shift Keying (QPSK). In addition, the modernized signals have been designed to ensure spectral separation with the legacy signals, to minimize interference. This has been accomplished with a modulation technique, known as Binary Offset Carrier (BOC) [Bet01]. This technique utilizes one or more subcarriers in addition to spreading code signals to distribute the signal power away from the carrier frequency.

2.2 Link Budget

As GNSS SVs orbit around the earth in approximately 20000 km altitude, the signal is attenuated from Free Space Path Loss (FSPL) by the time of reception. Calculations of FSPL are derived from Friis' transmission formula and is expressed in dB as:

$$FPSL(dB) = 10 \cdot \log_{10}\left(\left(\frac{4\pi df}{c}\right)^2\right) \tag{2.2}$$

where d is the distance between the transmitting and receiving antennas, f is the carrier frequency and c is the speed of light.

For GPS L1, assuming a distance between user and SV of $2 \cdot 10^7$ m, the loss is -182.4 dB [BD99]. In addition to FSPL, attenuation through the atmosphere is assumed to contribute with approx. 2 dB [GPS06]. The transmitted power for the GPS L1 C/A signal has an Effective Isotropic Radiated Power (EIRP) of 478.63 W (26.8 dBW) [BG91]. From the above, the received power for the GPS L1 C/A signal can be calculated as:

$$P_{RX}(dB) = P_{TX} - L_{FS} - L_A = 26.8dB - 182.4dB - 2.0dB = -157.6dB \tag{2.3}$$

The received signal are in fact extremely weak, this can be verified by calculating the resulting Signal-to-Noise Ratio (SNR). The thermal noise floor can be calculated from:

$$P_{noise}(dB) = 10 \cdot \log_{10}(k \cdot t \cdot B) \tag{2.4}$$

where $k = 1.38 \cdot 10^{-23} J/^{\circ}K$ is Boltzmann's constant, t is the noise temperature and B is the noise bandwidth.

The effective noise temperature is a function of ambient temperature, transmission line losses and receiver noise. A typical value for GNSS receivers can be assumed to be 513 K [BD99]. The bandwidth is assumed to be 2 MHz, i.e, the bandwidth of the main-lobe of a C/A code signal. Given the above, the thermal noise floor is -138.5 dB. The SNR can then be found to be:

$$SNR(dB) = P_{RX}(dB) - P_{noise}(dB) = -157.6db - (-138.5db) = -19.1dB \tag{2.5}$$

The weak signal strength is usually not a problem for GNSS receivers, due to the use of spread-spectrum modulation. However since the signal is buried beneath the noise floor, the margin of operation is rather narrow.

2.3 Atmospheric Effects

The signal propagation medium from SV to receiver can for 95% of the distance be regarded as vacuum or free space [ME12]. When the signal reaches the atmosphere, it will be subject to refraction from the Ionosphere and Troposphere layers. The nature of the refraction is although different between the two layers and will be briefly summarized in the following sections.

2.3.1 Ionosphere

The ionosphere is located from about 60 km upto 1000 km in altitude. The ionosphere consists of ionized gases and its state is primarily determined from activity from the sun.

The ionosphere is a dispersive medium, i.e., waves of different frequencies travels with different velocities through the medium. The dispersive property, has the profound effect that the carrier and it's modulation (code) is refracted differently. In fact, the carrierwave will exhibit a phase advance and the modulation will encounter a group delay. This is commonly referred to as code-carrier divergence.

The propagation speed of electromagnetic waves depends on the number of free electrons in the propagation path. This is characterized as the total electron content (TEC). This measure can be defined as:

$$TEC = \int_{S}^{R} n_e(l)\, dl \tag{2.6}$$

where the integration limits, denotes the path from SV to receiver. $n_e(l)$ is the number of electrons, in a 1 m^2 cross sectional area.

Given the TEC value, the resulting range errors from code-phase measurement and carrier-phase measurements can be found as:

$$I_\rho = -I_\phi = \frac{40.3 \cdot TEC}{f_L^2} \tag{2.7}$$

where I_ρ is the delay on the pseudorange in meters and I_ϕ is the delay for carrier-phase in meters. f_L is the carrier-frequency in hertz.

It should be noted, that the ranging errors due to phase advance and group delay are equal in magnitude, but different in sign.

Due to the frequency dependency of the ranging errors, multi-frequency receivers can estimate and mitigate delay caused by ionosphere penetration. Single frequency GPS users, can compensate for some of the ionospheric delay by using the Klobuchar model. The model consists of 8 parametres, which is transmitted in the SV navigation message and can typically mitigate around 50% of the range error [Klo87].

The state of the ionosphere varies substantially by day and night, this is caused by Ultra-Violet (UV) radiation from the sun, breaking up molecules and freeing electrons. Long term effects, such as solar cycles also influences the ionosphere, especially during peaks. A general note, is that the effects of the ionosphere on GNSS equipment is most profound near the poles or close to equator. For a more comprehensive review of the influence on GNSS signals from the ionosphere, refer to [Klo96].

2.3.2 Troposphere

The troposphere resides between the earth's surface and 60 km altitude. Refraction in the troposphere is caused by 2 effects, a hydrostatic component delay and a wet component delay.

The hydrostatic component delay, also known as the dry delay is caused by dry gases, primarily N_2 and O_2. The wet component delay consist of water vapor.

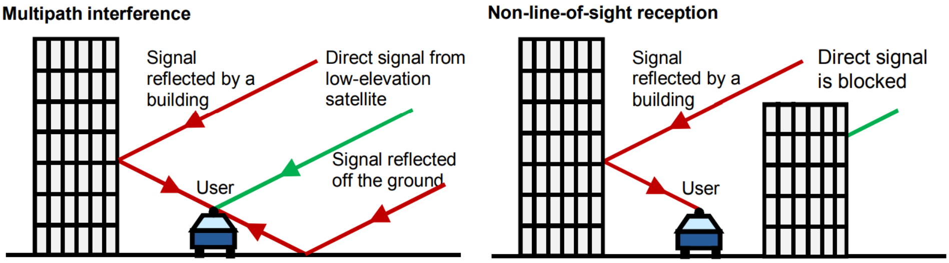
The troposphere is non-dispersive, i.e., frequency independent and can thus not be measured by a GNSS receiver. The path-delay caused by the troposphere typically ranges between 2.5-25 m [ME12], dependent of SV elevation angles. Although the delay can not be measured, it can however be modelled using physical parameters and its effects can then greatly be reduced. Two popular models for compensating tropospheric delay in receiver measurements are the Hopfield [Hop69] and Saastamoinen [Saa72] models.

2.4 Effects from Reception Environment

In addition to atmospheric effects, GNSS signals are also influenced by the reception environment. In the following, a short description of the effects of multipath interference and None Line-Of-Sight (NLOS) reception. Furthermore, a short description of signal effects when the signal is shadowed by vegetation is given.

2.4.1 Multipath and NLOS

Two of the most prominent error sources of GNSS positioning is caused by multipath interference and NLOS signals. These effects both stems from the reception environment and is illustrated in Figure 2.5.


Multipath interference
Signal reflected by a building
Direct signal from low-elevation satellite
Signal reflected off the ground
User
Non-line-of-sight reception
Signal reflected by a building
Direct signal is blocked
User

(a) Multipath Reception.

(b) NLOS Reception.

Figure 2.5: Multipath and None Line-Of-Sight reception. Figures adopted from [GJRS13]

In Figure 2.5a, the antenna receives both the direct signal and two reflected signals. This effectively distorts the signal after despreading and could cause the receiver to perceive a longer (constructive interference) or shorter (destructive interference) distance to the SV than the truth. A similar, although different situation arise in Figure 2.5b. In this case, the direct signal is blocked, but a reflected signal is able to make way to the antenna. This would cause the receiver to perceive a longer distance to the SV.

An illustration of how multipath is perceived from the receiver is illustrated in Figure 2.6.

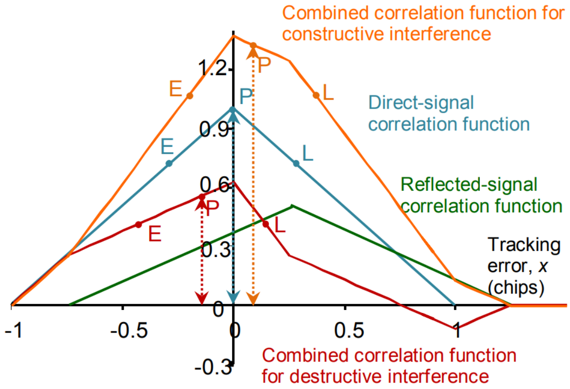


Figure 2.6: Effect of multipath in receiver correlation. The blue curve represents correlation of the direct signal and the green curve represent correlation of the reflected signal. The resulting correlation function can be illustrated by either the orange curve for constructive interference or the red curve for destructive interference. Figure adopted from [GJRS13]

As mentioned, multipath effects can either be constructive or destructive. Constructive multipath occurs, when the carrier phase of the reflected signal w.r.t. to the direct signal is within ±90°. This would cause the correlation functions of the direct and reflected signals to add up. If the phase-difference is around 180°, the resulting effect would be destructive and it would cause the correlation functions to be subtracted. For a stationary user, multipath would also cause ringing or oscillations on the measured signal strength, as the phase-difference and thereby the change between constructive and desctructive interference would vary due to SV motion.

There are different strategies in which multipath and NLOS signals can be mitigated. The simplest form of mitigation is by the use of a choke-ring antenna.

This antenna has a gain-pattern, which attenuates signals arriving from a low or negative incidence angle. For some environments, such as maritime navigation this constitutes the main problem. In Figure 2.5a, a choke-ring would not solve the problem entirely. Another aspect on the antenna, is the polarization sensitivity. GNSS signals are transmitted as Right Hand Circular Polarized (RHCP). Whenever a GNSS signal is reflected from a surface, the polarization changes to Left Hand Circular Polarized (LHCP). For a well designed antenna, the attenuation of reflected LHCP signals should be at least 10 dB on normal incidence angles [Bra96].

2.4.2 Shadowing from Vegetation

When a GNSS receiver is shadowed by vegetation, there are a number of factors which influences the resulting signals. Propagation of RF waves through dense foliage would in general be influenced by scattering, diffraction, absorption and reflection, all of which will contribute to attenuation of the received signal strength [ML10]. An illustration of signal propagation through foliage is shown in Figure 2.7.

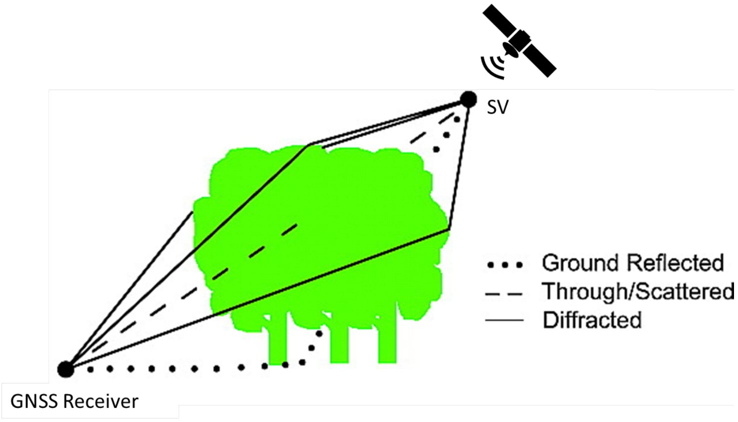


Figure 2.7: RF wave propogation effects through foliage. Figure modified from [SNS+03]

The attenuation from trees are also affected by seasonal effects. The attenuation from a tree in full foliage is aprrox. 35% greater, than a deciduous tree without foliage. This concludes that the majority of attenuation is caused by wood tree limbs, branches and trunk rather than leaves [Spi96].

A number of empirical models which describes attenuation in dB/m of foliage penetration has been proposed. The models are typically dependent on SV elevation angle, leaf state and type of tree.

Chapter 3

GNSS Receiver Processing

As it was stated in the introduction, a GNSS receiver consists of a RF front-end, a baseband processor and a navigation processor. The functional blocks of a GNSS receiver is illustrated in Figure 3.1.

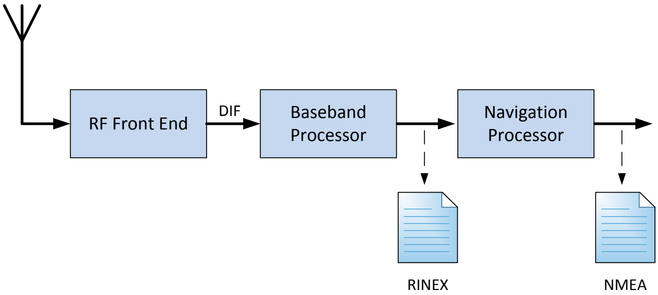


Figure 3.1: Functional blocks of a GNSS receiver.

This chapter introduces the most important concepts for a typical GNSS receiver. The first section is concerned with the RF front-end, where the received signal is amplified, downmixed to IF and digitized. Hereafter, concepts for baseband processing is given. The initial operation in the baseband processor is known as satellite acquisition, where the receiver determines which SVs are in view. Hereafter, the receiver transitions into continuous tracking of the acquired SVs, where separate feedback loops for code- and carrier tracking is employed. The continuous SV tracking is normally accomplished by a DLL and PLL and associative loop filters. An alternative tracking algorithm, which is based on a extended Kalman filter is furthermore introduced. A description of how receiver measurements (or observables) is generated then follows and common algorithms for obtaining a position and velocity solution is then given. The chapter concludes with a description of a vector tracking receiver, which utilizes feedback from the navigation solution to aid the tracking algorithms.

3.1 RF Front-End

GNSS signals are transmitted with carrier frequencies in excess of 1 GHz. The data bandwidth of most signals are however much lower, typically only in the order of a few MHz for non-military legacy signals. GNSS receivers do not process the signals directly at RF, since this would require huge processing capabilities and further require state-of-the-art High Speed Analog-To-Digital converters (ADCs) to digitize the signal. The common approach is to downmix the RF signal to an IF frequency in the MHz range.

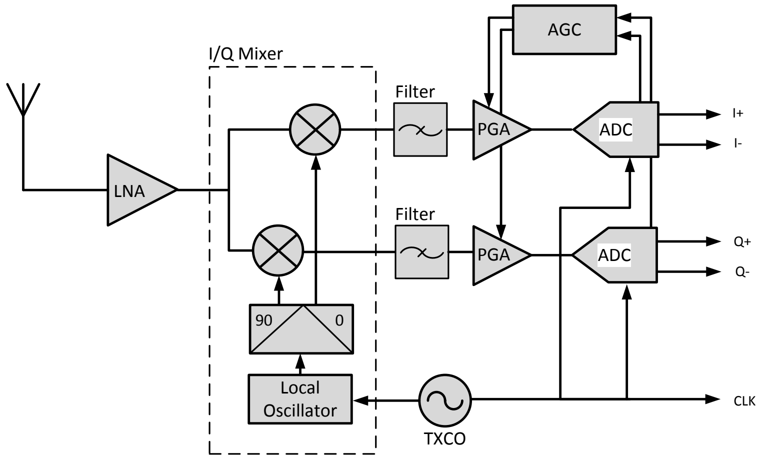


Figure 3.2: Functional block diagram of GNSS RF Front-End.

Figure 3.2 show the elements of a typical GNSS RF Front-End. In this example, the input from the GNSS antenna is passed through a Low-Noise Amplifier (LNA) for initial amplification. The signal is then downconverted by the use of an In-phase/Quadraphase mixer (I/Q Mixer) which translates the original RF spectrum to a (much lower) IF frequency. The IF signal is then either bandpass or lowpass filtered to prevent out-of-band interference and aliasing effects in the subsequent AD conversion. A typical front-end is often equipped with an Automatic Gain Control (AGC) circuit. Such a device is used, to adjust the amplification prior to the ADC for optimum saturation. This is typically achieved by evaluating the statistical distribution from a histogram of output levels in the ADC.

It should be noted, that the visualized front-end architecture should only be perceived as an example, as there could be a number of design variations. This could, for example, be the number of mixing stages, the number of bits in the ADC, the use of I-only (real) or I/Q (complex) mixers and the absence or inclusion of an AGC.

A drastically different approach, is that of a Direct RF Sampling front-end. This implementation differs from our example by not using a mixer-circuit altogether. Instead, a downtranslation of the RF spectrum to IF is achieved by exploiting the bandpass sampling or undersampling technique. This technique samples a bandpass filtered RF signal at a rate below the Nyquist frequency, but is still able to reconstruct the signal from a low-frequency alias of the process. In the following, only traditional front-ends with the use of mixers are considered, but for more information about Direct RF Sampling the reader is referred to, e.g., [PAT03], [SVR08].

3.1.1 Frequency Mixing

A mixer is used for downconverting the original RF spectrum to a (lower) IF spectrum. This is accomplished by multiplication of the incoming signal, $s_{RF}(t)$, with a pure sinusoid generated from a Local Oscillator (LO), $s_{LO}(t)$.

Initially, we only consider the I-component of the mixer from Figure 3.2.

In the time-domain, the mixer operation is defined as:

$$s_{Mix}(t) = s_{RF}(t) \cdot s_{LO}(t) \tag{3.1}$$

By calculating the Fourier transform of eq. (3.1.1), the frequency domain representation can be found.

$$\begin{aligned} S_{Mix}(f) &= \mathcal{F}\{s_{RF}(t) \cdot s_{LO}(t)\} \\ &= \mathcal{F}\{s_{RF}(t)\} * \mathcal{F}\{s_{LO}(t)\} \\ &= S_{RF}(f) * S_{LO}(f) \end{aligned} \tag{3.2}$$

where $*$ is the convolution operator, defined as $f * g(u) = \int f(u) \cdot g(x-u)\,\mathrm{d}u$.

Graphically, the mixing process can be illustrated as shown in Figure 3.3.

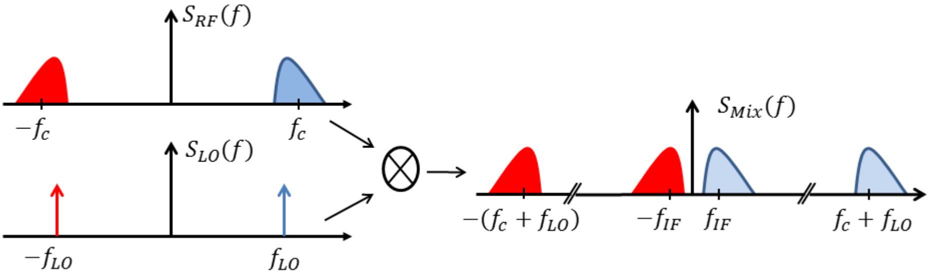


Figure 3.3: Frequency domain representation of mixing process.

From the above, we can conclude that the mixing process generates spectral replicas at:

$$\pm f_c \pm f_{LO} \tag{3.3}$$

where f_c is the carrier frequency from the original RF spectrum and f_{LO} is the LO (mixing) frequency.

At this point, it should be noted that the I-mixer would only function properly for a non-zero IF frequency, i.e. $f_c \neq f_{LO}$. The primary reason for this, is that the spectra residing on negative frequencies in Figure 3.3 are mirrored replicas of the positive spectra. If the LO frequency is chosen equal to the carrier frequency, i.e., $f_{IF} = 0$ Hz, the resulting baseband spectrum would comprise of both the positive spectra (f_{IF}) and the mirrored negative version ($-f_{IF}$) on top of each other, this would cause the original signal to be lost or heavily degraded. An intuitive explanation to the origins of the negative spectra, comes from the trigonometric identity, $cos(2 \cdot \pi \cdot f \cdot t) = cos(2 \cdot \pi \cdot (-f) \cdot t)$, i.e., a positive frequency can not be distinguished by a negative frequency with equal magnitude.

The problem with the negative mirror spectrum, can be overcomed by the use of an IQ mixer. The IQ mixer consist of two mixing blocks, where the LO phase would be shifted exactly 90 degrees between the I- and Q components. For this case, we can represent the LO signal in a complex notation:

$$s_{LO}(t) = cos(2\pi \cdot f_{LO} \cdot t) + i \cdot sin(2\pi \cdot f_{LO} \cdot t) = e^{j \cdot 2\pi \cdot f_{LO} \cdot t} \tag{3.4}$$

The Fourier transform of eq. (3.4) can be found as:

$$S_{LO}(f) = \mathcal{F}\{e^{j \cdot 2\pi \cdot f_{LO} \cdot t}\} = 2\pi\delta(f - f_{LO}) \tag{3.5}$$

where δ is the Dirac delta function.

The result of eq. (3.5), shows that the LO signal now is represented as only one single positive bin in the frequency domain. The absence of a negative spectral component, eliminates the mirrored IF spectrum after the mixing process. The IQ mixing process is illustrated in Figure 3.4.

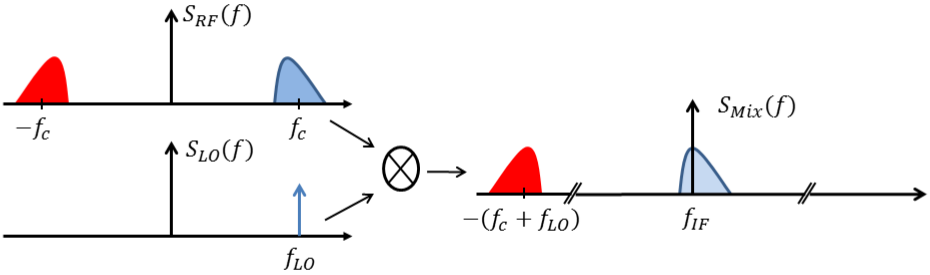


Figure 3.4: Frequency domain representation of Mixing process (IQ-mixing).

A zero IF implementation, is also known as a direct conversion receiver.

3.1.2 Sampling and Filtering

Subsequent to the mixing process, the resulting signal is filtered and sampled by an ADC. The filter would typically be a bandpass filter for a non-zero IF receiver and a lowpass filter for zero IF conversion. The reason to perform filtering is to reduce out-of-band interference and to prevent aliasing of the sampled signal. The minimum required filtering bandwidth to succesfully correlate the incoming signal with a receiver generated replica, in the baseband processing, should not be less than that of the main-lobe of the spreading code (see e.g. Figure 2.4). For GPS L1 C/A code, this would yield a minimum bandwidth of $2 \cdot 1.023$ MHz $= 2.046$ MHz. However, there are some degrading effects by completely filtering out the sidelobes of the signal spectrum. The triangular shape of the autocorrelation function of a BPSK signal, would appear more rounded or smoothened, which can decrease sensitivity if a receiver with narrow correlator spacing is used. [SYG09] describes the effects on the autocorrelation function, as a function of filter bandwidth in more detail.

The number of bits in a ADC typically ranges from 1 to 16 bits for typical receivers. For "normal" operation in which the LOS between receiver and SV is unobstructed a 1 or 2-bit ADC would normally be sufficient. [BAMR03] performed a study on the post-correlation SNR loss as a function of quantization levels of the ADC.

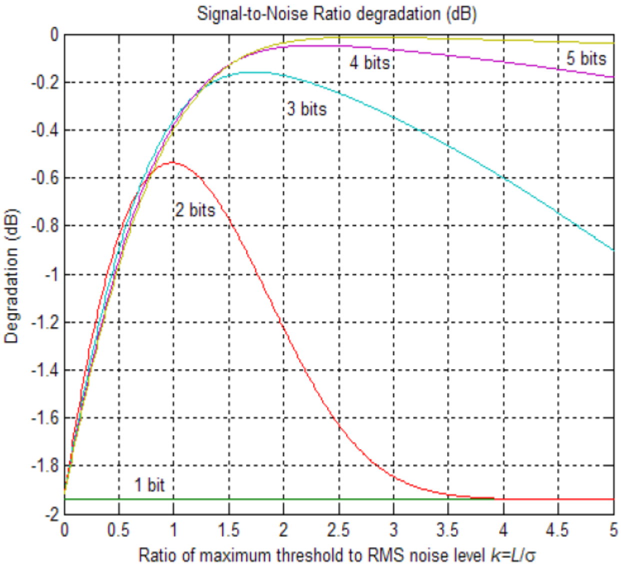


Figure 3.5: SNR degradation from [BAMR03]. The SNR degradation is calculated from different bit-widths of an ADC. L denotes the maximum quantization threshold of the ADC and σ is the standard deviation of the input signal.

Figure 3.5 shows that for a 1-bit (hard-limiter) ADC, the post-correlation SNR degradation is 1.96 dB. By using 2 bits, the SNR degradation increases to 0.54 dB. In multibit ADCs, an AGC to adjust the preamplification coefficient, k, should be used for optimum performance. It should be noted, that the above results are optimistic, in the sense, that filtering effects and degradation from finite sample-rates are not included. For more details of those effects, refer to [Cha82].

A reason for choosing an ADC with more bits, is if the receiver should be made robust to jamming and interference. For this case, a jamming signal would saturate the ADC and must then have sufficient dynamic range in order to preserve and detect the GNSS signal [Tsu00].

The above implicitly assumes the use of non-centered ADC with a uniform (linear) quantization law, i.e, that each quantization step have a constant seperation, which is used most typically. In the presence of Continuous Wave (CW) interference, [Amo83] however showed that an adaptive, non-uniform quantization approach would be better.

3.2 Satellite Acquisition

After the downmixing and digitization in the RF front-end, the first task of a GNSS receiver is to determine, which SVs are in view. Due to the nature of the CDMA scheme, employed by GPS, Galileo and Beidou, multiple SVs are transmitting on the same carrier frequency and hence can not be distinguished by their carriers alone. Instead it is necessary to correlate the incoming signal with individual spreading codes (PRNs) associated with the different SVs. Due to the relative movement of SV and receiver, the carrier frequencies for each SV varies with a Doppler-induced shift. In order to determine the presence of a particular SV, the acquisition algorithm needs to sweep a frequency range for possible Doppler shifts. In addition, the receiver generated code also needs to be aligned with the incoming signal. The different Doppler and code-phase combinations, can be evaluated by the Cross Ambiguity Function (CAF) [Bor08].

$$S_k(\tau, f_D) = \frac{1}{N} \sum_{n=0}^{N-1} r[n] \cdot c(nT_s - \tau) \cdot e^{-j2\pi f_D n T_s} \tag{3.6}$$

where $S_k(\tau, f_D)$ is the CAF as function of combined Doppler and IF, f_D, and code-phase offset, τ, between incoming signal and receiver replica. $r[n]$ is the digitized received signal from the front-end. $c(nT_s - \tau)$ is the receiver generated spreading code, N is the number of samples, which corresponds to the number of samples equaling the length of the spreading code. T_s is the ADC sampling interval. Eq (3.6) is expressed in terms of a complex exponential, in the following it is more convenient to split this term up to an inphase (real) and quadrature (imaginary) part. Using Euler's formula, we obtain:

$$\begin{aligned} S_k(\tau, f_D) =& \frac{1}{N} \sum_{n=0}^{N-1} r[n] \cdot c(nT_s - \tau) \cdot \cos(2\pi f_D n T_s) \\ &- \frac{j}{N} \sum_{n=0}^{N-1} r[n] \cdot c(nT_s - \tau) \cdot \sin(2\pi f_D n T_s) \\ &= S_{k,I} - jS_{k,j} \end{aligned} \tag{3.7}$$

It is clear, that the CAF produces a complex number and hence is impractical for signal detection in the above form. The most common approach is to square and sum the I (real) and Q (imaginary) parts to find the squared absolute value,

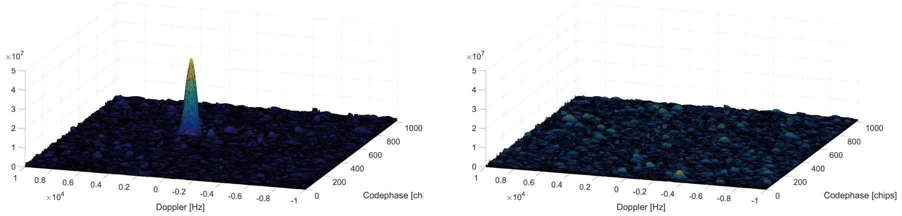
$$|S_k(\tau, f_D)|^2 = S_{k,I}^2 + S_{k,j}^2 \tag{3.8}$$

The output from Eq. (3.8) is compared against a decision threshold to determine if a SV is present from the IF data, i.e.,

$$\max_{\tau, f_D} |S_k(\tau, f_D)|^2 > T \tag{3.9}$$

where T is the decision threshold.

A visual representation of $|S_k(\tau, f_D)|^2$ is given in Figure 3.6.



(a) SV present.

(b) SV not present.

Figure 3.6: CAF evaluation for present and non-present SV.

The figure shows both the case when a SV is present (left) and not present (right). Note, that this example is for an unnormalized CAF, i.e., the results are not scaled with $1/N$.

The search algorithm for code-phase and Doppler can be implemented in different ways. The simplest approach conceptually is to implement a brute-force serial search algorithm which sequentially sweep through all possible combinations. This approach is illustrated in Figure 3.7.

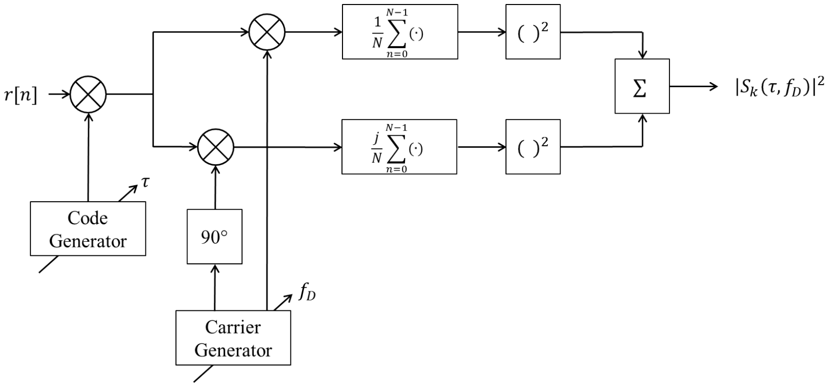


Figure 3.7: Serial search algorithm.

The disadvantage of the serial search algorithm is mainly computational inefficiency, since each combination of f_D and τ is calculated individually. A more sophisticated approach, known as the parallel code-phase search algorithm is shown in Figure 3.8.

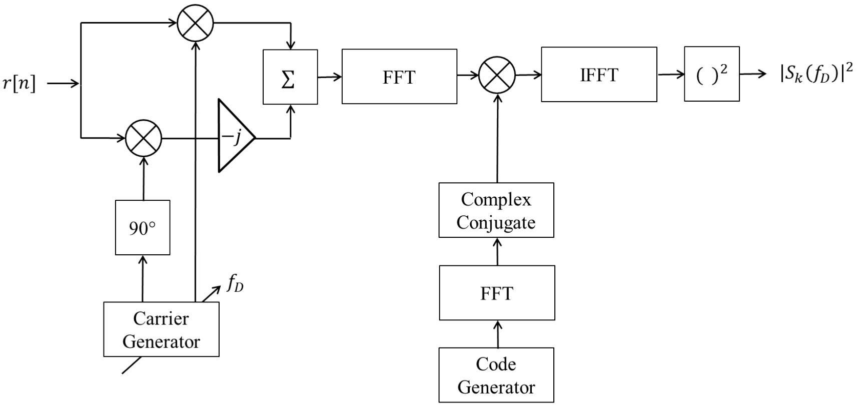


Figure 3.8: Parallel code-phase search algorithm.

It should be apparent, that this algorithm have reduced the search space to one-dimension, i.e., only f_D has to be varied. The resulting output of the algorithm is a $N \times 1$ vector, evaluating response to different code-phase delays, with a resolution of $\delta\tau = 1023/N$ for GPS L1 C/A code. This is possible, due to the circular correlation theorem:

$$(f \star g)[l] = \sum_{n=-0}^{N-1} f^*[n] \cdot g[n+l] = \mathcal{F}^{-1}\{F^* \cdot G\}_l \tag{3.10}$$

where F^* denotes the complex conjugate of the discrete Fourier transform (DFT) of f, and G is the DFT of g.

From substituting $f[n] = c(nT_s)$ and $g[n+l] = r[n+l] \cdot e^{-j2\pi f_D (n+l) Ts}$, the equivalence to eq. (3.6) should be clear. However is should be noted that, the lag operator, l, has been changed from the replica-code to the input-sequence,$r[n]$ and carrier-generator. The improved efficiency of this method is related to the highly optimized algorithms for calculating the DFT, known as the Fast Fourier Transform (FFT).

A third method, known as the the Parallel Frequency search method could also have been used, this is parallelizing the search in the Doppler domain opposed to the code-phase domain, this method is described in more detail in, e.g., [Bor08] or [BAB$^+$07].

In the case, where signals are weak the CAF can be evaluated for succesive spreading-code periods either coherently or non-coherently. For coherent combinations, we have:

$$S_{C,k}(\tau, f_D) = \left| \sum_{i=1}^{M} S_{k,i}(\tau, f_D) \right|^2 = \left| \frac{1}{N} \sum_{n=0}^{MN-1} r[n] \cdot c(nT_s - \tau) \cdot e^{-j2\pi f_D n T_s} \right|^2 \tag{3.11}$$

where $S_{C,k}$ is the coherently combined CAF function, i is an index of the successive code-periods. The coherent combination is the best option for increasing signal-strength and reduce noise variance. The limit is however, that since the spreading code is modulated with a navigation message, the maximum coherent combinations is equal to 20 ms for GPS L1 C/A.

For non-coherent combinations, we have

$$S_{N,k}(\tau, f_D) = \sum_{i=1}^{M} |S_{k,i}(\tau, f_D)|^2 \tag{3.12}$$

where $S_{N,k}$ is the non-coherently combined CAF function, i is an index of the successive coherent integration-periods.

Here the envelope or power of the different CAFs are combined, which is immune to databit modulations from the navigation message. The non-coherent combination are although associated with a squaring-loss penalty [SMMB07].

A practical note, it is common to split the acquisition process in two stages. Firstly a common coarse frequency search can be implemented with frequency separations of e.g. 500 Hz. For the subsequent tracking, this separation would often be too coarse for the tracking to lock-in on the signal. Hence a fine-frequency stage can be implemented with separations of, e.g., 25 Hz for the already detected SVs.

3.3 Tracking

After successful acquisition, the receiver transitions into tracking of the acquired SVs. The receiver tracks the code-phase and the Doppler + nominal IF of the GNSS SVs. As it was also discussed in the previous section, this is largely accomplished by correlation of the incoming signal with receiver generated replicas. An illustration of how correlation is implemented is shown in Figure 3.9.

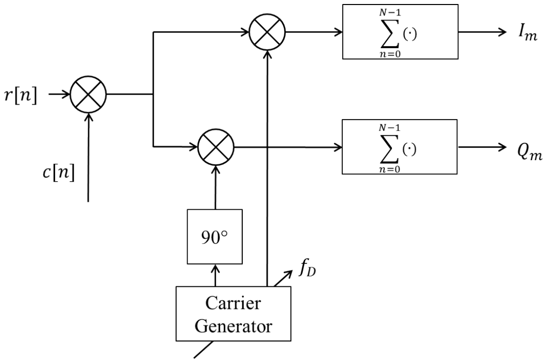


Figure 3.9: Receiver Correlation.

In Figure 3.9, the downconverted and digitized incoming signal $r[n]$, is first multiplied with a receiver generated code signal, $c[n]$. The product is then branched into two arms, where the branched signals are multiplied with a cosine and a sine function respectively. Finally the resulting products are accumulated over N samples before the corresponding outputs are produced.

The first operation, i.e., the multiplication with a receiver generated code replica in Figure 3.9 is also known as code wipeoff, as the (PSK) modulation of the incoming signal is removed and the carrier-wave is restored. This is illustrated in Figure 3.10.

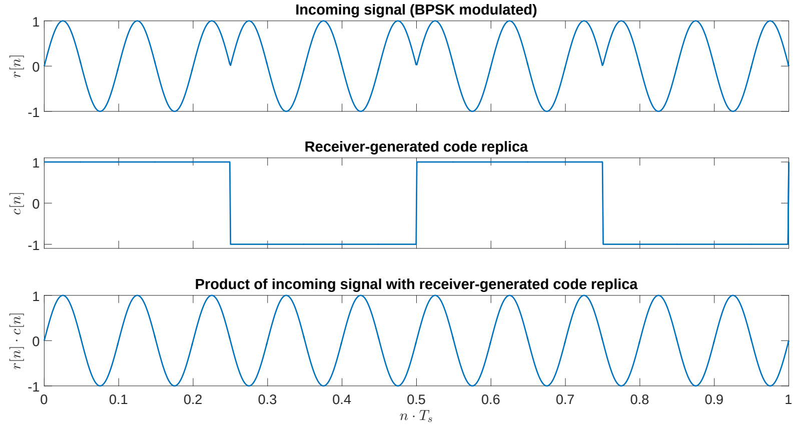


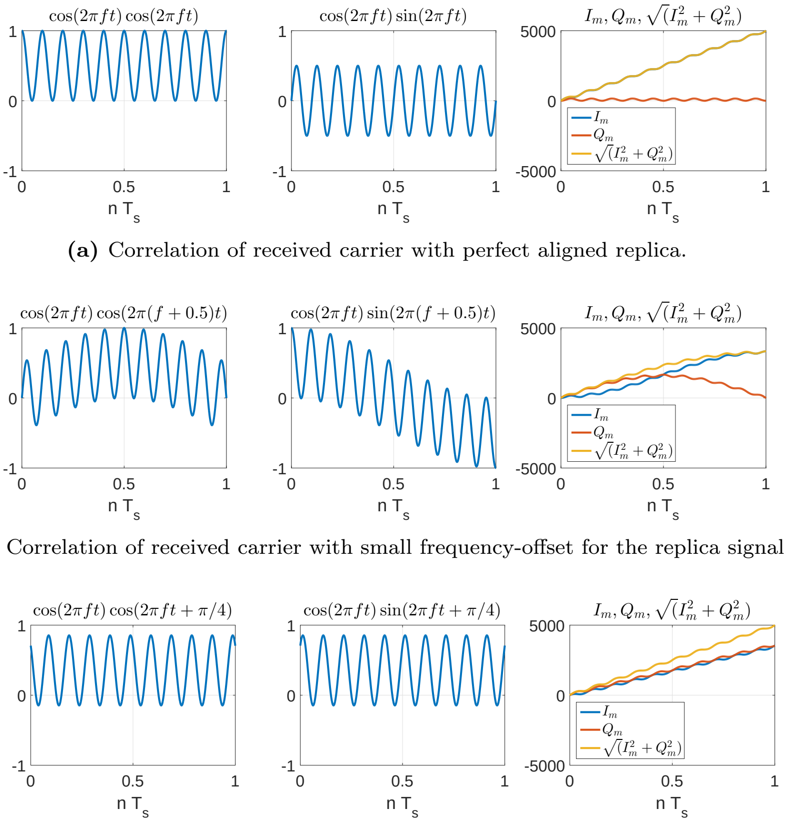
Figure 3.10: Code wipeoff. The phase-modulation of the incoming signal is removed by multiplying the signal with a perfectly aligned code replica.

After code-wipeoff the signal is multiplied with a sine and cosine respectively for each branch. An intuitive example of the resulting product is illustrated in Figure 3.11.

In Figure 3.11, the result of multiplication in the I- and Q arms are illustrated for three cases. In the first case (Figure 3.11a - left), the incoming signal is multiplied with a cosine which exactly matches the frequency and phase of the incoming signal. In this scenario, it can be seen that the resulting signal oscillates between $[1; 0]$. In the middle part of the figure, the product of the incoming signal with a sine wave is shown. In this situation, the resulting signal oscillates between $[-0.5; 0.5]$. The right part of this figure, shows the accumulated sums of the products. Here, it should be clear that all the correlation ”energy” is gathered in the I arm.

In the second case (3.11b), the receiver multiplies the incoming signal with sinusoids which differs slightly in frequency. It should be clear for the cosine- and sine multiplications, that the mean values of both products are time-varying. This becomes even more apparent, when the accumulated sums are inspected.

Here the combined energy, from the I- and Q arms are considerably lower from the previous case, clearly indicating an attenuation due to the frequency offset.



(a) Correlation of received carrier with perfect aligned replica.

(b) Correlation of received carrier with small frequency-offset for the replica signal.

(c) Correlation of received carrier replica signal with a constant phase offset.

Figure 3.11: Multiplication of sinusoids. Figure (a) shows the result of multiplication of sinusoids with identical frequency and phase. Figure (b) shows the multiplication of sinusoids with (slightly) different frequencies. Figure (c), shows multiplication of sinusoids with a constant phase offset.

For the final scenario, the receiver-generated sinusoids have equal frequency but has a constant phase offset of 45° compared to the incoming signal. In this case, the correlation energy is equally split between I- and Q arms. The combined energy, is however equal to that of the first case. The above explanation was merely an attempt to give an intuitive understanding of receiver correlation. In the following, a more generic mathematical model is derived.

After code-wipeoff, we can use a trigonometric identity to express the product in the I-arm as:

$$\begin{aligned}\cos(2\pi F^s nT_s + \phi^s) \cdot \cos(2\pi F_r nT_s + \phi_r) = \\ \frac{1}{2}\cos(2\pi F^s nT_s + \phi^s - 2\pi F_r nT_s - \phi_r) + \\ \frac{1}{2}\cos(2\pi F^s nT_s + \phi^s + 2\pi F_r nT_s + \phi_r)\end{aligned} \tag{3.13}$$

where F^s, ϕ^s is the frequency and phase-offset of the received (downconverted) carrier from the SV. F_r, ϕ_r is the frequency and phase of the receiver generated signal.

The above expression can further be rewritten to:

$$\begin{aligned}\frac{1}{2}\cos(2\pi(F^s - F_r)nT_s + (\phi^s - \phi_r)) + \frac{1}{2}\cos(2\pi(F^s + F_r)nT_s + (\phi^s + \phi_r)) = \\ \frac{1}{2}\cos(2\pi\delta f nT_s + \delta\phi) + \frac{1}{2}\cos(2\pi(F^s + F_r)nT_s + (\phi^s + \phi_r))\end{aligned} \tag{3.14}$$

where $\delta f, \delta\phi$ is the frequency and phase difference, between the incoming carrier and the receiver generated carrier. The last term in the above equation, i.e., the frequency and phase sums, can be neglected, due to the subsequent accumulation (low-pass filtering). The I-correlator output can then be expressed as:

$$I_m = \frac{1}{2}\sum_{i=0}^{N-1} \cos(2\pi\delta f nT_s + \delta\phi) \tag{3.15}$$

Eq. (3.15) can also be seen as a Riemann approximation to the following continuos-time integral:

$$I_m = \frac{1}{2T_s}\sum_{i=0}^{N-1} \cos(2\pi\delta f nT_s + \delta\phi)T_s \approx \frac{1}{2T_s}\int_0^T \cos(2\pi\delta f t + \delta\phi)\, dt \tag{3.16}$$

The solution to this integral is, [Die96]:

$$I_m \approx \frac{1}{2T_s} \int_0^T \cos(2\pi\delta f t + \delta\phi)\, dt \approx N \cdot \frac{sin(\pi \cdot \delta f \cdot T)}{\pi \cdot \delta f \cdot T} \cdot cos(\bar{\delta\phi}) \tag{3.17}$$

where $\bar{\delta\phi}$ is the average phase error over the integration interval, i.e., $\delta\phi + \frac{1}{2}\delta f T$. This far, it has been assumed that the code has been completely wiped off the incoming signal. In general, that would not be true. In order to account for effects of code misalignment, the autocorrelation function of the spreading code should be included in the model.

This leads to the following model of the I- and Q correlator outputs, [Die96], [PJ02], [Cur10]:

$$I_m \approx A \cdot N \cdot R(\delta\tau) \cdot \frac{sin(\pi \cdot \delta f \cdot T)}{\pi \cdot \delta f \cdot T} \cdot cos(\bar{\delta\phi}) + \eta_I \tag{3.18}$$

$$Q_m \approx A \cdot N \cdot R(\delta\tau) \cdot \frac{sin(\pi \cdot \delta f \cdot T)}{\pi \cdot \delta f \cdot T} \cdot sin(\bar{\delta\phi}) + \eta_Q \tag{3.19}$$

where $\delta\tau$ is the code-phase misalignment i chips, between the received code and the receiver replica. η_I, η_Q is zero-mean white (gaussian) noise. The noise in the I_m- and Q_m correlators are uncorrelated.

In the above, we have only considered correlation for an open-loop correlation process. Due to constant SV-user movement, the carrier frequency and code-phase needs to be estimated continuously and adjusted to maximize the correlation outputs.

The model presented in eq. (3.18-3.19) is traditionally not used by GNSS receivers to estimate (or track) code- and carrier parameters of the incoming signal. A more common approach is the use of discriminator functions and loop filters. The following sections will introduce these concept for carrier- and code tracking.

3.3.1 Carrier Tracking

Carrier- and code tracking in a GNSS receiver is mutually dependent. As indicated in equations (3.18-3.19), the outputs of both the I- and Q correlators would be pure noise if the code is not tracked. Conversely, if the GNSS receiver is unable to estimate the carrier frequency, the $sin(x)/(x)$ term would greatly attenuate the correlation outputs. Although the code- and carrier tracking are interrelated in terms of correlation output, the tracking of each is normally controlled by independent feedback loops. The carrier tracking sensitivity is although lower than that of the code tracking and is often considered the weakest link [War98].

There are in general two approaches for carrier tracking. Frequency tracking, which is accomplished by a Frequency Lock Loop (FLL) and phase tracking which is based on a Phase Lock Loop (PLL). Both approaches relies on processing the correlation outputs by the means of a discriminator function, which estimates either phase- or frequency errors. The discriminator outputs is subsequently and used to update the estimated carrier frequency for next correlation. The underlying architecture is depicted in Figure 3.12.

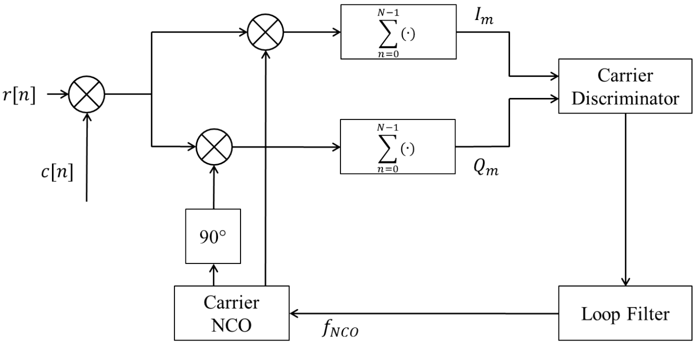


Figure 3.12: Carrier tracking loop.

In the case of phase tracking, the loop implicitly tries to gather all the signal energy in the I-arm. Referencing Figure 3.11c, it was visualized that for a constant phase-offset the signal energy was distributed between both the I- and the Q arms. It should be inherently clear, that the phase offset then can be estimated by comparing the energy between the two correlators.

$$\tan(\bar{\delta\phi}) = \frac{\sin\bar{\delta\phi}}{\cos\bar{\delta\phi}} \propto \frac{Q_m}{I_m} \Rightarrow \bar{\delta\phi} = \tan^{-1}\left(\frac{Q_m}{I_m}\right) \tag{3.20}$$

At this point it should be remembered that the navigation message is modulated on top of the carrier and spreading code. A change in bit polarity causes a 180 degree phase shift of the incoming carrier. As the phase shift induced by navigation-bit transitions is a result of modulation and not LOS dynamics it should not be included in the phase estimate. A type of discriminators which are insensitive to this modulation is known as Costa's discriminators and a list of typical algorithms are shown in Table 3.1. The regular inverse tangent discriminator derived in eq. (3.20) is also insensitive to a 180 degree phase shift, due to the ambiguity; $\tan(\theta + \pi) = \tan(\theta)$.

For some cases, pure PLL discriminators could be employed, i.e., a discriminator that is sensitive in the entire region, $\phi = -180°..180°$. This however requires that a reliable data-bit estimation algorithm is employed, such that navigation-bit transitions would be known apriori. A pure PLL could be used for the pilot component featured in some modernized GNSS signals. The advantage of a pure discriminator PLL is 6dB better tracking sensitivity [WBH06].

Discriminator	Phase Error	Characteristics
$I_m \cdot Q_m$	$\sin 2\phi$	Near Optimal at low SNR
$sign(I_m) \cdot Q_m$	$\sin \phi$	Near Optimal at high SNR
$\tan^{-1}(\frac{Q_m}{I_m})$	ϕ	Optimal (Maximum-likelihood) at both low and high SNR

Table 3.1: Costa's discriminators functions. [WBH06]

The difference between the Costa's discriminators are mainly computational burden. The inverse tangent operator provides the best estimate and is a optimal for both low and high SNRs.

Regarding frequency tracking by a FLL, this is similarly achieved by a discriminator function. The frequency is essentially the derivative of the phase and hence common FLL discriminator functions typically depend on calculating a derivative of I- and Q correlator outputs. If two phase measurements are taken in succesion, we can derive an expression for the frequency change between the correlations, i.e.,

$$\delta f = \frac{\phi_{t_2} - \phi_{t_1}}{t_2 - t_1} = \frac{\tan^{-1}\left(\frac{Q_{m,t_2}}{I_{m,t_2}}\right) - \tan^{-1}\left(\frac{Q_{m,t_1}}{I_{m,t_1}}\right)}{t_2 - t_1} \tag{3.21}$$

The above result, is however more often represented as:

$$\delta f = \tan^{-1}\left(\frac{Q_{m,t_2} I_{m,t_1} - Q_{m,t_1} I_{m,t_2}}{I_{m,t_1} I_{m,t_2} - Q_{m,t_1} I_{m,t_2}}\right) \tag{3.22}$$

The above rearrangement might not seem trivial, but is based on the relation,

$\tan^{-1}(x) + \tan^{-1}(y) = \tan^{-1}\left(\frac{x+y}{1-xy}\right)$.

A few differences of a PLL and FLL should be mentioned; In general, a FLL is more robust and can keep track of a signal to lower power levels than a PLL. A FLL is also more insensitive to LOS dynamics. Conversely, the estimates are more noisy, which affect, e.g., the ability of the receiver to generate accurate carrier-phase observables. It is however also possible to implement a hybrid, known as the FLL assisted PLL [War98], [RGEA08]. This coupling, inherits the robust properties of the FLL, while providing precise (PLL) measurements.

3.3.2 Code Tracking

For the description of carrier tracking, it was assumed that the receiver generated code signal was perfectly aligned with the received signal. In this section, we assume that the receiver generated carrier is in perfectly alignment with the carrier for the incoming signal. The principle in code-tracking is that the receiver generates three time-shifted code replicas, known as the early, late and prompt signals. The receiver correlate the three time-shifted codes with the incoming signal. For the I- branch, the correlators are implemented as shown in Figure 3.13.

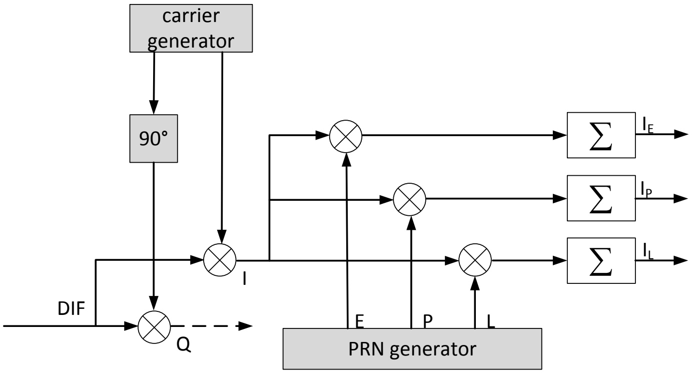


Figure 3.13: Early, Prompt and Late correlations.

The correlators for the Q-branch has a similar layout. The time-difference between the thrree replica signals is here assumed to have a separation of half a chip in the spreading code. An illustration of the resulting correlation output is given in Figure 3.14.

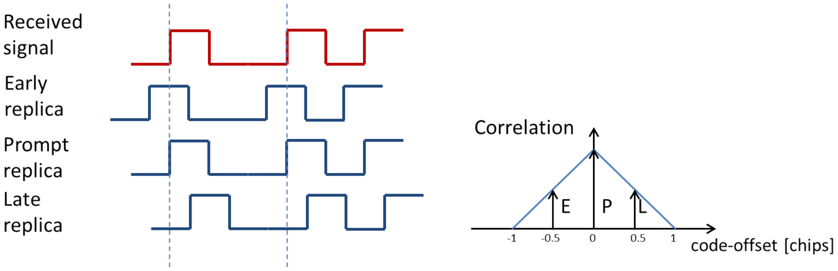


Figure 3.14: Correlation of received signal with receiver-generated replicas.

In the figure, the prompt signal is here illustrated to be perfectly aligned with the incoming sequence. As the spreading code is pseudorandom the correlation peak, will have a normalized value of unity, when one of the replicas exactly matches the incoming code signal. With a half chip seperation, the early and late correlations will have exactly half the energy of the prompt signal. It should be noted, that the correlation function illustrated here is valid for BPSK-modulated signals, such as GPS L1 C/A or GLONASS L1 C/A. The BOC modulation, employed by Galileo and for the coming GPS L1C signal, has a different shape. Another thing to note, is that the sharp edges of the triangular correlation function is negatively degraded by finite sample-rates and bandwidths, as described in section 3.1.2. From the above, it should be intuitively, that an estimate of code misalignment can be deduced by comparing, the early and late correlation results. In Table 3.2 a list of typical DLL discriminators are shown.

Discriminator	**Characteristics**
$I_E - I_L$	Coherent discriminator. Requires that the carrier tracking achieves phase-lock. Provides most accurate measurements.
$\frac{(I_E^2+Q_E^2)-(I_L^2+Q_L^2)}{(I_E^2+Q_E^2)+(I_L^2+Q_L^2)}$	Non-coherent discriminator. Normalized early minus late power.
$I_P(I_E - I_L) + Q_P(Q_E - QL)$	Non-coherent discriminator. The only discriminator that uses all six correlator outputs.

Table 3.2: Common DLL Discriminators. [BAB+07]

The discriminator, that is based on the difference between early and late correlations is the most accurate. This function, unlike the others, requires that phase-tracking is employed and hence need a PLL to operate. The other (non-coherent) variants works with either a FLL or a PLL.

3.3.3 Loop Filters

The carrier and code discriminators are not used directly to control the code- and carrier generation. Instead the discriminator outputs are propagated through a filter inorder to reduce noise. For both code- and carrier tracking, the filtered discriminators are essentially a frequency control signal, that is used as inputs for the Numerical Controlled Oscillators (NCOs) for code- and carrier generation.

A common method to asses closed loop performance of a GNSS carrier- or code tracking loop, is to use a linearized PLL model. A block diagram of the PLL model is shown in Figure 3.15.

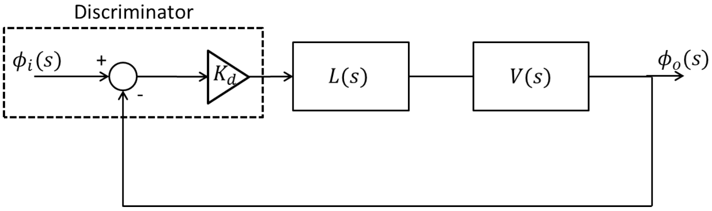


Figure 3.15: Linearized PLL model.

The linear model can accurately describe the behaviour of GNSS carrier- or code tracking performance, under the condition that the phase-error is small, i.e. $\phi \approx sin(\phi)$. In the figure, the transfer-function $V(s)$ denotes a Voltage Controlled Oscillator (VCO), the continuous-time equivalent of an NCO. This is modeled, as an integrator, which in the Laplace-domain is equal to $V(s) = \frac{K_o}{s}$. $L(s)$ denotes the transfer function of the loop filter. From the PLL model, a closed loop transfer function of the PLL can be formed as:

$$\frac{\phi_o(s)}{\phi_i(s)} = \frac{K_d L(s) V(s)}{1 + K_d L(s) V(s)} \tag{3.23}$$

where $\phi_i(s)$ describes the phase error between received signal and receiver replica. $\phi_o(s)$ expresses the loop-response, in terms of replica phase.

Given the closed-loop model, methods from classical control theory can be used to characterize, e.g., settling time, percent overshoot and loop bandwidth. In addition, a steady state error analysis can also be performed for different kinematic scenarios. Another parallel to classical control theory is made by [Cur10], who models different loop filters as a generalized Proportional-Integral (PI) controller. [CCSJ93] provides a more thorough analysis of the linearized PLL model.

The perhaps most important parameter of the closed-loop characteristics is the loop bandwidth. A narrow bandwidth would typically lead to less noise on the estimates. The bandwidth should however also be wide enough to accommodate Doppler shifts induced by LOS dynamics. [WBH06] provide a list of design equations for different orders of loop-filters and their respective trade-offs.

3.3.4 Complete Tracking Block

Until now, individual aspects of code- and carrier tracking has been investigated. In this section, the joint operation of a GNSS receiver tracking block is described.

A complete tracking block for a traditional GNSS receiver is shown in Figure 3.16.

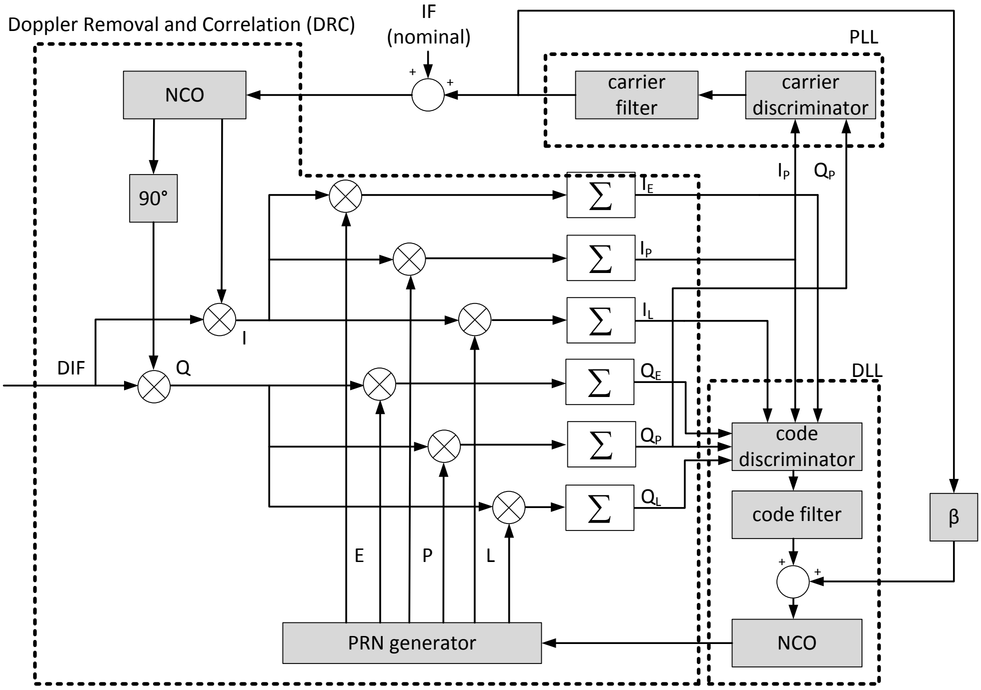


Figure 3.16: Tracking-block for conventional GNSS receiver.

The dashed boxes indicates the various parts of the tracking block. The correlators, carrier-generator and PRN generator is denoted as the Doppler Removal and Correlation (DRC). The PLL consist of a carrier discriminator, carrier loop filter and NCO. The DLL consists of a code discriminator, a code loop filter and a NCO. From the figure, a carrier-aiding of code loop is also included. This has not previously been described, but since receiver dynamics affect both code- and carrier tracking in terms of Doppler variations, it is common to aid the code-tracking from the carrier frequency estimate. A Doppler induced shift however influences the carrier frequency differently than the code-frequency. Inorder to accomodate this, a scaling factor, $\beta = \frac{1.023\text{MChips/s}}{1575.45\text{MHz}}$ is used to translate carrier Doppler into code Doppler. The resulting effect of the aiding, is that the code-loop only needs to track the Code-Carrier divergence effects, as described in section 2.3.1.

The update rate of the PLL and DLL is generally 1 ms, equalling the duration of a GPS L1 C/A code sequence. Since the navigation message is modulated with 50 bit/s, the coherent integration time can although be extended to 20 ms. Longer coherent integration increases the postcorrelation SNR and thereby reduces noise for the tracking estimates. However, as the bits of the navigation message is modulated on top of the spreading code, with 20 ms intervals, bit transitions needs to be identified inorder to utilize the longer coherent integrations. A common algorithm, known as the histogram bit-synchronizer can be utilized for identifying at which integration indices a new data-bit occurs. This algorithm is explained in detail in [Die96].

In addition to bit-synchronization, it is also beneficial to estimate the GNSS signal strength at the antenna input in terms of Carrier-to-noise density ratio, C/N_0. The obtainable strength is related to the noise residing on measurements, hence invaluable in subsequent position determination. A popular algorithm is the narrow-band over wide-band algorithm described in [Die96]. This algorithm requires bit-synchronization and uses a combination of coherent and non-coherent integrations of the prompt I- and Q correlators, to estimate obtained C/N_0. The referenced algorithm works satisfactorily for modest to strong signal strengths, but other algorithms might be better in extreme cases [FPF08].

Finally, it is also useful to implement signal lock-detectors for continuous quality assesments. Lock-detectors can provide a qualitative metric of, e.g., code-lock, phase-lock and frequency-lock. The reader is referred to [Die96] for a more thorough description on different lock detectors. A metric which will later be used in this thesis, is the Phase Lock Indicator (PLI). This detector requires bit-sync and provides an estimate of average phase alignment between incoming signal and receiver replica. The metric can be interpreted as $PLI = \cos(2\bar{\delta\phi})$, where $\bar{\delta\phi}$ represents the average phase-error. If this metric is close to 1, it serves as an indication that perfect phase tracking is achieved.

3.3.5 Kalman Filter-based Tracking

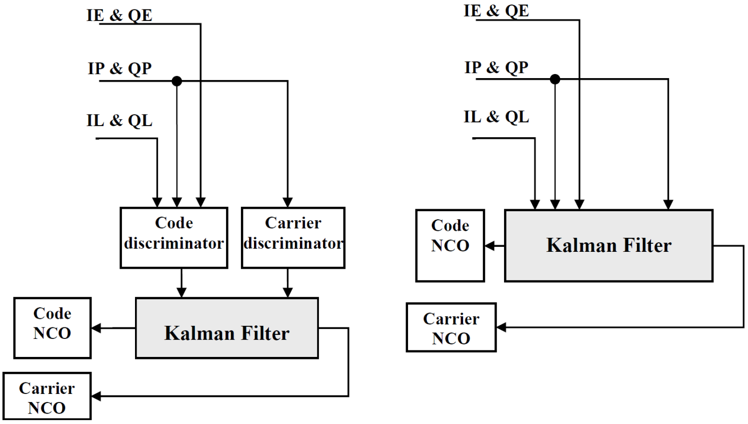
As an alternative to standard tracking loop architectures, which includes discriminators and loop filters, different authors have proposed to use Kalman filters (KF) to drive the code- and carrier NCOs. The KF is also known as Linear Quadratic Estimation and is based on a series of measurements and a dynamic model representing the physical relations between the observed states. The KF is an optimal filter, provided that the dynamical model accurately describes the behavior of the observed system, the measurement noise is white and the covariance is exactly known [BH97].

There has been several suggestions for how a KF based tracking algorithm could be designed. An early attempt of using a KF for GNSS signal tracking was made by [Gus98]. This approach used a hybrid of Maximum-Likelihood and KF estimation methods to design a PLL. Another approach, was taken by [Psi01] where a post-processing algorithm for a DLL and PLL was accomplished by the use of a KF smoother. Combined PLL and DLL algorithms has also been proposed, see e.g. [PJ02], [ZG04]. The main motivation for replacing loop filters with KF variants, is that while the traditional architecture works well in good reception environments, a traditional loop filter is not able to adapt the loop-dynamics under varying circumstances. A KF provides a more adaptive estimation algorithm, which is based on a transition (prediction) model, a measurement/observation model and their associated statistics. The use of KFs instead of traditional tracking filters, have proven to reduce noise on estimates and providing lower tracking thresholds.

There are in general, two options for the inputs to a KF-based tracking design. One approach is to use the non-linear discriminators as measurement to the KF and another is to use the correlation outputs directly. The two architectures are visualized in Figure 3.17.

The primary difference, is that when using the non-linear discriminators as input to the KF, it can normally be implemented as an ordinary linear time-invariant filter. In the second case, when using the correlators as input directly, the filter would typically be implemented as an Extended KF (EKF).

In the following, an algorithm which uses the correlators as input to a 5-state EKF is described. The algorithm was originally proposed by [PJ02] and later revisited by [PL06], [POL08b].



(a) Discriminator-based EKF **(b)** Correlator-based model

Figure 3.17: Correlator-based and Discriminator based inputs to EKF Code- and Carrier tracking. Figure from [Sal10]

3.3.5.1 KF Observation Model

The observation model of the KF is based on the correlator model, previously introduced in eq. (3.18-3.19):

$$I(\Delta) \approx A \cdot N \cdot R(\delta\tau - \Delta) \cdot \frac{sin(\pi \cdot \delta f \cdot T)}{\pi \cdot \delta f \cdot T} \cdot cos(\overline{\delta\phi}) \tag{3.24}$$

$$Q(\Delta) \approx A \cdot N \cdot R(\delta\tau - \Delta) \cdot \frac{sin(\pi \cdot \delta f \cdot T)}{\pi \cdot \delta f \cdot T} \cdot sin(\overline{\delta\phi}) \tag{3.25}$$

where Δ is the correlator spacing. A is the signal amplitude, N is the number of samples in the integration interval, $R()$ is the autocorrelation-function of the spreading code, δf is the carrier frequency error, T is coherent integration time and $\overline{\delta\phi}$ is the average phase error.

From [PJ02], [POL08b], the average phase-error can further be parametrized as:

$$\overline{\delta\phi} = \delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6} \tag{3.26}$$

Notice, that the average phase error here is expanded at the end of an integration interval. Refer to [POL08b] for details of this parametrization.

The non-linear observation model for the KF can be stated as:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} = \mathbf{h}(\bar{A}, \delta\tau, \delta\phi, \delta f, \delta a) + \mathbf{v} \tag{3.27}$$

where $\mathbf{z} = \begin{bmatrix} I_E & I_P & I_L & Q_E & Q_P & Q_L \end{bmatrix}^T$ is the measurement vector, $\mathbf{h}(\mathbf{x})$ is a non-linear function of the state-estimates and $\mathbf{v}$ is a term for measurement noise, which will be clarified later on. The term $\bar{A} = A \cdot \frac{sin(\pi \cdot \delta f \cdot T)}{\pi \cdot \delta f \cdot T}$ combines the frequency attenuation term and the amplitude term. The reason for combining the terms, is that they are otherwise complex to separate. The frequency error can instead be estimated from the average phase error as described above. From eq. (3.24-3.27), $\mathbf{h}(\bar{A}, \delta\tau, \delta\phi, \delta f, \delta a)$ can be defined as:

$$\mathbf{h}(\bar{A}, \delta\tau, \delta\phi, \delta f, \delta a) = \begin{bmatrix} \bar{A} \cdot N \cdot R(\delta\tau - \Delta) \cdot cos(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \\ \bar{A} \cdot N \cdot R(\delta\tau) \cdot cos(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \\ \bar{A} \cdot N \cdot R(\delta\tau + \Delta) \cdot cos(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \\ \bar{A} \cdot N \cdot R(\delta\tau - \Delta) \cdot sin(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \\ \bar{A} \cdot N \cdot R(\delta\tau) \cdot sin(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \\ \bar{A} \cdot N \cdot R(\delta\tau + \Delta) \cdot sin(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \end{bmatrix} \tag{3.28}$$

As eq. (3.27) is non-linear, it needs to be linearized in order to use it for the EKF observation model.

$$\mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial I(\Delta)}{\partial \mathbf{x}} \\ \frac{\partial Q(\Delta)}{\partial \mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial I(\Delta)}{\partial \hat{A}} & \frac{\partial I(\Delta)}{\partial \delta\tau} & \frac{\partial I(\Delta)}{\partial \delta\phi} & \frac{\partial I(\Delta)}{\partial \delta f} & \frac{\partial I(\Delta)}{\partial \delta a} \\ \frac{\partial Q(\Delta)}{\partial \hat{A}} & \frac{\partial Q(\Delta)}{\partial \delta\tau} & \frac{\partial Q(\Delta)}{\partial \delta\phi} & \frac{\partial Q(\Delta)}{\partial \delta f} & \frac{\partial Q(\Delta)}{\partial \delta a} \end{bmatrix} \tag{3.29}$$

where the partial derivatives of $I(\Delta)$ w.r.t. the state vector is found to be:

$$\begin{aligned}
\frac{\partial I}{\partial A} =& N \cdot R(\delta\tau - \Delta) \cdot cos(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \\
\frac{\partial I}{\partial \delta\tau} =& \bar{A} \cdot N \cdot J(\delta\tau - \Delta) \cdot cos(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \\
\frac{\partial I}{\partial \delta\phi} =& \bar{A} \cdot N \cdot R(\delta\tau - \Delta) \cdot \left(-sin(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6})\right) \\
\frac{\partial I}{\partial \delta f} =& \bar{A} \cdot N \cdot R(\delta\tau - \Delta) \cdot \frac{T}{2} \cdot sin(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6}) \\
\frac{\partial I}{\partial \delta a} =& \bar{A} \cdot N \cdot R(\delta\tau - \Delta) \cdot \left(-\frac{T^2}{6} \cdot sin(\delta\phi - \delta f \cdot \frac{T}{2} + \delta a \cdot \frac{T^2}{6})\right)
\end{aligned} \tag{3.30}$$

here $J(x)$ is the derivative function of $R(x)$.

The EKF observation matix, can now be defined as:

$$\mathbf{H} = \begin{bmatrix}
\frac{\partial I_E}{\partial A} & \frac{\partial I_E}{\partial \delta\tau} & \frac{\partial I_E}{\partial \delta\phi} & \frac{\partial I_E}{\partial \delta f} & \frac{\partial I_E}{\partial \delta a} \\
\frac{\partial I_P}{\partial A} & \frac{\partial I_P}{\partial \delta\tau} & \frac{\partial I_P}{\partial \delta\phi} & \frac{\partial I_P}{\partial \delta f} & \frac{\partial I_P}{\partial \delta a} \\
\frac{\partial I_L}{\partial A} & \frac{\partial I_L}{\partial \delta\tau} & \frac{\partial I_L}{\partial \delta\phi} & \frac{\partial Q_P}{\partial \delta f} & \frac{\partial Q_L}{\partial \delta a} \\
\frac{\partial Q_E}{\partial A} & \frac{\partial Q_E}{\partial \delta\tau} & \frac{\partial Q_E}{\partial \delta\phi} & \frac{\partial Q_E}{\partial \delta f} & \frac{\partial Q_E}{\partial \delta a} \\
\frac{\partial Q_P}{\partial A} & \frac{\partial Q_P}{\partial \delta\tau} & \frac{\partial Q_P}{\partial \delta\phi} & \frac{\partial Q_P}{\partial \delta f} & \frac{\partial q_P}{\partial \delta a} \\
\frac{\partial Q_L}{\partial A} & \frac{\partial Q_L}{\partial \delta\tau} & \frac{\partial Q_L}{\partial \delta\phi} & \frac{\partial Q_P}{\partial \delta f} & \frac{\partial Q_L}{\partial \delta a}
\end{bmatrix} \tag{3.31}$$

For the measurement noise matrix, it is assumed, that noise for the I- and Q correlators are uncorrelated and the correlation in each branch are determined from the correlator spacing. The measurement noise covariance matrix, is assumed as [Xie10]:

$$\mathbf{R} = \sigma^2 \begin{bmatrix}
1 & \Delta & 0 & 0 & 0 & 0 \\
\Delta & 1 & \Delta & 0 & 0 & 0 \\
0 & \Delta & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \Delta & 0 \\
0 & 0 & 0 & \Delta & 1 & \Delta \\
0 & 0 & 0 & 0 & \Delta & 1
\end{bmatrix} \tag{3.32}$$

where

$$\sigma^2 = \frac{1}{2 \cdot 10^{0.1 \cdot C/N_0} \cdot T}$$

3.3.5.2 State-transition model

In addition to the observation model, the EKF also consist of a state-transition or prediction model. This model is based on the intermediate relationships between the states and is stated in continuous time as [POL08b]

$$\frac{d}{dt}\begin{bmatrix} \bar{A} \\ \delta\tau \\ \delta\phi \\ \delta f \\ \delta a \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}} \cdot \begin{bmatrix} \bar{A} \\ \delta\tau \\ \delta\phi \\ \delta f \\ \delta a \end{bmatrix} + \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & \beta & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{G}} \cdot \underbrace{\begin{bmatrix} w_A \\ w_{\delta\tau} \\ w_{clk,b} \\ w_{clk,d} \\ w_a \end{bmatrix}}_{\mathbf{W}} \tag{3.33}$$

The model incorporates carrier-aided code tracking, as it can be seen from the β parameter in the **A** and **G** matrices. The model is based on the fundamental relations between phase, frequency and frequency-rate, i.e, $\delta\phi = T \cdot \delta f$ and $\delta f = T \cdot \delta a$. The process noise for the state-transition model, is based on expected variation of signal amplitude, code-carrier divergence, oscillator effects and frequency-rate changes. The latter is related to the dynamic conditions of the receiver. The used oscillator model is treated in more detail later in section 3.5.3.

More specifically about the process noise, then w_A is a white-noise process to account for variations in signal amplitude. The parameter is approximately constant for static applications but can vary greatly in kinematic applications depending on the surrounding environment. $w_{\delta\tau}$ is a white noise process to account for code-carrier divergence. This parameter stems from the fact that code and carrier frequencies are delayed differently in the ionosphere. w_b is the driving noise for the oscillator. w_d is the driving noise for the oscillator drift. Finally, w_a is the driving noise to account for line of sight acceleration.

The EKF is implemented in discrete time, so the continuous representation needs to be transformed by, e.g., van Loans method:

$$\mathbf{N} = \begin{bmatrix} -\mathbf{A} & \mathbf{G}\bar{\mathbf{W}}\mathbf{G}^T \\ \mathbf{0} & \mathbf{A}^T \end{bmatrix} \cdot \Delta t$$

$$\mathbf{M} = expm(\mathbf{N}) = \begin{bmatrix} \cdots & \mathbf{\Phi}_k^{-1}\mathbf{Q}_k \\ \mathbf{0} & \mathbf{\Phi}_k^T \end{bmatrix}$$

where $\bar{\mathbf{W}} = diag(\mathbf{W})$

To this point, the observation matrix, $\mathbf{H}_k$, state-transition matrix $\mathbf{\Phi}_k$, measurement covariance, $\mathbf{R}_k$, and process noise covariance, $\mathbf{Q}_k$, have been defined. The EKF mechanization consist of two steps; a prediction stage and a data-update stage. In the prediction stage, the most current state estimate is propogated forward in time using the state-transition matrix. In addition, an error covariance or state-uncertainty matrix, $\mathbf{P}_k$, is also propogated based on the previous values and the process noise covariance. The prediction step can be summarized as:

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_k\hat{\mathbf{x}}_{k-1}^+ \tag{3.34}$$

$$\mathbf{P}_k^- = \mathbf{\Phi}_k\mathbf{P}_{k-1}^+\mathbf{\Phi}_k^T + \mathbf{Q}_{k-1} \tag{3.35}$$

Here the subscript, k denotes the time index. The notation $^+,^-$ indicates whether the estimate includes the data-update. The hat-accent is included, to emphasize that it is a KF estimate. It should be noted, that strictly $\mathbf{\Phi}_k$ and $\mathbf{Q}_k$ is time-invariant, so the time index could have been omitted.

For the data-update, the measurement vector $\mathbf{z}_k$ is formed from the correlator outputs, and the residual error from the non-linear model, $\mathbf{h}(\mathbf{x})$ is calculated. A Kalman gain $\mathbf{K}_k$ is then calculated based on the linearized observation matrix, the error covariance and the measurement covariance. Finally, the state estimates are updated based on the measurement residual, $\tilde{\mathbf{y}}_k$ and the Kalman gain.

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-) \tag{3.36}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k \tag{3.37}$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{3.38}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k \tag{3.39}$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{3.40}$$

The Kalman gain, $\mathbf{K}_k$, determines how much "trust" is given to the new measurement compared to the prediction model, based on the correspoding covariances, $\mathbf{R}_k$ and $\mathbf{Q}_k$.

It should be noted, that the data-update mechanizations are iterated because of the highly non-linear nature of the observation model. Another remark, is that the observation matrix, $\mathbf{H}_k$, is rank-deficient. A least-squares solution can hence not produce a unique solution for the carrier phase error, frequency-error and frequency-rate-error. The physical coupling between these measures in the state-transition matrix, does however make the terms observable over time. The state uncertainty matrix, $\mathbf{P}_k$, needs to be initialized when the algorithm is first initiated. A simple way of doing this, would be to employ the tracking filter after an initial period of PLL/DLL tracking. Statistics of the discriminator outputs from the PLL/DLL-based tracking could then give a measure of the expected initial uncertainty. The same method is similar feasible for providing initial state estimates.

3.3.6 Tracking Errors

Code- and carrier tracking errors influences the quality of the subsequent navigation solution. In addition, tracking errors also dictate to which thresholds the receiver can maintain lock on the SVs.

3.3.6.1 Carrier Tracking Errors

A conservative rule of thumb for reliable carrier-phase tracking, is that the phase jitter for an arctangent Costa's discriminator should respect the following inequality, [WBH06]:

$$3\sigma_{PLL} = 3\sigma_j + \theta_e \leq 45^\circ \tag{3.41}$$

where σ_j is the standard deviation of all phase jitter sources, except dynamic stress errors. θ_e denotes the dynamic stress error.

σ_j can further be expressed as:

$$\sigma_j = \sqrt{\sigma_{tPLL}^2 + \sigma_v^2 + \sigma_A^2} \tag{3.42}$$

where σ_{tPLL} is standard deciation of phase jitter caused by thermal noise, σ_v is vibration induced oscillator jitter. σ_A is Allan variance-induced oscillator jitter [All66].

The thermal noise jitter can be calculated as, [WBH06]:

$$\sigma_{tPLL} = \frac{360}{2\pi}\sqrt{\frac{B_n}{10^{\frac{C/N_0}{10}}}\left(1+\frac{1}{2T_c 10^{\frac{C/N_0}{10}}}\right)} \tag{3.43}$$

where B_n is the bandwith of the carrier-tracking loop filter, T_c is coherent integration time.

From eq. (3.43), it should be clear that reducing the loop bandwidth and extending the coherent integration time, would results in a lower phase jitter for the PLL.

The vibration induced oscillator jitter, σ_v, is caused by the output frequency of an oscillator is dependent of acceleration changes. This effect can be modelled as:

$$\sigma_v = \frac{360 f_L S_v}{2\pi}\sqrt{\int_{f_{min}}^{f_{max}} \frac{P(f_m)}{f_m^2}\, df_m} \tag{3.44}$$

where S_v is the oscillation vibration sensitivity (Hz/g), with $g \approx 9.8\ m/s^2$. f_L is the carrier frequency, e.g. 1575.42 MHz for GPS L1. $P(f_m)$ is vibration power as a function of frequency, (g^2/Hz).

The Allan variance induced phase jitter, is determined by the loop filter order. To clarify, the Allan variance is a measure of frequency instability or variance, subject to different averaging times. At short averaging intervals, the instability can be characterized as phase-noise. For medium averaging intervals, the observed instability can be seen as flicker or $1/f$ noise. For longer averaging intervals, the instability characterizes random walk of frequency variations. For a third order filter, σ_A, can be expressed as, [IE02]:

$$\sigma_A = 360\pi f_L^2\left[\frac{\pi^2 h_{\text{-}2}}{3\omega_0} + \frac{\pi h_{\text{-}1}}{3\sqrt{3}\omega_0^2} + \frac{h_0}{6\omega_0}\right] \tag{3.45}$$

where ω_0 is related to the loop bandwidth. For a third order loop, $\omega_0 = B_n/0.7845$. [WBH06]. The parameters h_0, h_{-1}, h_{-2} is asymptotic approximations to the three frequency instability regions.

Dynamic stress error is generally determined by the steady state errors encountered by the loop filter. A first order filter is associated with a steady state error related to step-changes in velocity, a second order filter exhibits steady state errors for step-changes in acceleration and third order filter has a steady state error associated with jerk (derivative of acceleration). The dynamic stress error for a third order PLL is defined as [WBH06]:

$$\theta_e = \frac{d^3R/dt^3}{\omega_0^3} = 0.4828 \cdot \frac{d^3R/dt^3}{B_n^3} \tag{3.46}$$

where R is the distance between receiver and SV.

In summary, the total phase jitter encountered by a PLL is dependent on receiver dynamics, C/N_0, oscillator quality, loop bandwidths and coherent integration time. There are an inherent tradeoff between, e.g., the loop-bandwidth and dynamic performance. The thermal noise jitter is reduced for low bandwidths, but contrary the dynamic stress error is inversely proportional to the bandwidth.

Similar derivations can be made for a FLL, see e.g. [WBH06].

The EKF tracking algorithm is not characterized by a loop bandwidth, hence the above equations do not readily apply. However, upon certain assumptions, the Kalman Gain matrix would typically converges over time to a steady state. [SOL12] provides an analysis of the noise-equivalent bandwidth for a EKF operating in steady-state, which in turn means that the above equations would be made applicable.

3.3.6.2 Code Tracking Errors

Similarly to PLL tracking, a conservative rule for stability in the DLL is that:

$$3\sigma_{DLL} = 3\sigma_{tDLL} + R_e \leq D \tag{3.47}$$

where σ_{tDLL} thermal noise code tracking jitter. R_e is dynamic stress error. D is the early-late spacing, i.e. 2Δ.

The dynamic stress error term, can be completely eliminated by the use of carrier aiding as it was described in section 3.3.4.

The code tracking jitter, arising from thermal noise can be stated as [WBH06]:

$$\sigma_{tDLL} = \begin{cases} \sqrt{\frac{B_n D}{2C/N_0}\left[1+\frac{2f_c}{C/N_0(2-D)}\right]} & \text{if } D \geq \frac{\pi f_c}{B_{fe}} \\ \sqrt{\frac{B_n}{2C/N_0}\left(\frac{f_c}{B_{fe}}+\frac{B_{fe}}{(\pi-1)f_c}\left(D-\frac{f_c}{B_{fe}}\right)^2\right)\left[1+\frac{2f_c}{C/N_0(2-D)}\right]} & \text{if } \frac{f_c}{B_{fe}} \leq D \leq \frac{\pi f_c}{B_{fe}} \\ \sqrt{\frac{B_n}{2C/N_0}\left(\frac{f_c}{B_{fe}}\right)\left[1+\frac{2f_c}{C/N_0}\right]} & \text{if } D \leq \frac{f_c}{B_{fe}} \end{cases} \tag{3.48}$$

where f_c is the chipping-rate of the spreading code, e.g., 1.023 Mchips/s for GPS L1 C/A. B_{fe} is the dual-sided bandwidth of the front-end.

As it has been explained earlier, the front-end bandwidth negatively degrade the correlation output of the receiver. In addition, the front-end bandwidth determines how closely spaced the correlators can be for, e.g., better rejection of multipath. It should be recalled from section 2.4.1, that multipath occurs when a direct LOS signal is received together with a reflected version of the signal. For reflected signal with a long path-delay, the inclucence on the code-correlation can be reduced by selecting a narrow Early-Late correlator spacing. A visual representation of the correlator spacings and corresponding error with respect to multipath is shown in Figure 3.18.

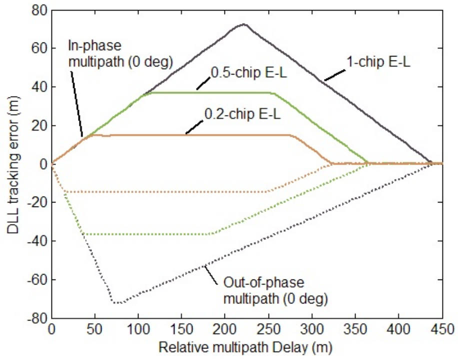


Figure 3.18: Multipath error envelopes. Figure adopted from [Cha11]

It should be noted, that the multipath error envelopes is valid for the BPSK modulation used for GPS L1 C/A code. The BOC modulation employed in, e.g., Galileo signals would lead to different envelopes.

In terms of baseband processing algorithms, there has been an extensive research on multipath mitigation techniques. A popular DLL design for multipath mitigation is based on the double-delta correlator [IE03]. The double-delta correlator extends the normal Early-Prompt-Late correlators with 2 additional correlations, which can be denoted as the Very-Early and Very-Late versions. A coherent DLL dicriminator from [MB99] based on the double-delta correlator can be stated as:

$$(E - L) - \frac{1}{2}(VE - VL) \tag{3.49}$$

where VE, VL is denoting the additional Very-Early and Very-Late correlations.

For a more extensive review of multipath mitigation strategies, refer to [GJRS13].

3.4 GNSS Measurements

A GNSS receiver usually produces 3 fundamental measurements. A pseudorange observable, which is obtained from the relative arrival time of the C/A codes. This measurement is based on the output of the DLL in the receiver tracking loops. The receiver also generates an instantaneous Doppler measurement, based on the tracked carrier-frequency. This quantity is often used to calculate user velocity. Finally, an ambiguous carrier-phase measurement is generated, where the receiver accumulates or integrates the tracked Doppler over time. This observable is the basis of precise GNSS methods such as RTK or Precise Point Positioning (PPP).

3.4.1 Pseudoranges

The pseudorange measurement can be thought as the perceived or apparent distance between receiver and SV and can generally be defined as [ME12]:

$$\rho(t) = c\left[t_u(t) - t^s(t - \tau)\right] \tag{3.50}$$

where $t_u(t)$ is the signal arrival time, measured from the user (receiver) clock. $t^s(t - \tau)$ is the signal emission time, measured from the SV clock. t is the true arrival time and τ is the true travel-time of the signal. c is the speed of light.

From the above equation, it can be immediately seen, that any offset in the user and SV clocks from the true time scale, would affect the apparent range measurement.

By definition of a receiver clock bias $t_u(t) = t + \delta t_u(t)$ and a SV clock bias $t^s(t-\tau) = (t-\tau) + \delta t^s(t-\tau)$ eq. (3.50) can be rewritten to [ME12],

$$\rho(t) = c\tau + c\left[\delta t_u(t) - \delta t^s(t-\tau)\right] + \epsilon_p(t) \tag{3.51}$$

The factor $\epsilon_p(t)$ is added to take unmodelled errors into account. These errors include receiver noise, atmospheric delays and multipath.

A general model for the pseudorange measurement can be stated as [ME12],

$$\rho(t) = r + c\left[\delta t_u - \delta t^s\right] + I_p + T_p + e(t) \tag{3.52}$$

where the reference to the measurement epoch, t, has been left out. r is the true geometric range, I_p, T_p is the ionospheric and tropospheric path-delays. The error term $e(t)$, models multipath and tracking errors.

Generation of pseudoranges

The receiver time-scale is essentially the sample numbers associated with the incoming digitized IF data. The resolution of the timescale is $T_s = 1/f_s$, where f_s is the sample rate of the ADC. There are fundamentally two ways a receiver can generate pseudoranges; either by using the Common Transmission Time or the Common Reception Time [PFP12].

Both methods relies on the fact that GPS SV clocks are synchronized to GPS Time (GPST). For the GPS L1 C/A signal, a navigation message is transmitted to the user from all SVs simultaneously. This message consist of a number of frames, where each frame again has a number of subframes. The frames consist of SV clock correction parameters, orbital parameters, ionospheric correction coeffiecients etc. Each subframe contains a Telemetry Word (TLM), which has a preamble of 8 bits and can be used to detect the beginning of a new subframe.

As the receiver clock is not synchronized to GPST, it is not possible to simply detect the TLM word and then apply eq. (3.50). However, as the TLM word has been broadcasted simultaneously, it is possible to measure the relative time of arrival between the SVs. This strategy is similar for both approaches. The common transmission time method waits until a preamble for the farthest SV is received and then compute the relative time between the receptions. The common reception time method is similar, although this method can generate pseudoranges at any instant. In Figure 3.19, an illustration of how a measurement is generated using the common reception time.

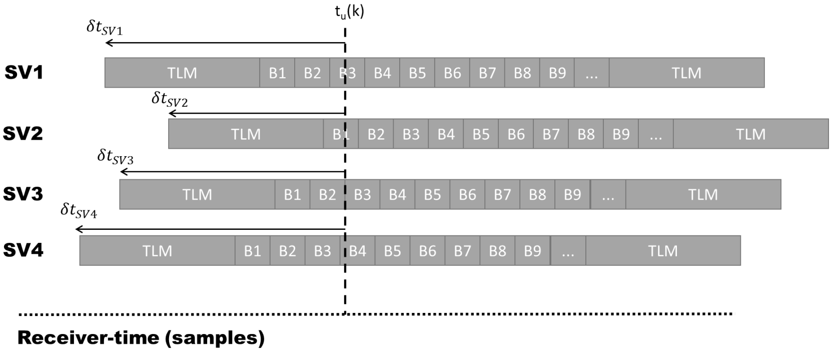

$t_u(k)$
δt_{SV1}
SV1
TLM B1 B2 B3 B4 B5 B6 B7 B8 B9 ... TLM
δt_{SV2}
SV2
TLM B1 B2 B3 B4 B5 B6 B7 B8 B9 ... TLM
δt_{SV3}
SV3
TLM B1 B2 B3 B4 B5 B6 B7 B8 B9 ... TLM
δt_{SV4}
SV4
TLM B1 B2 B3 B4 B5 B6 B7 B8 B9 ... TLM
Receiver-time (samples)

Figure 3.19: Common reception time (not to scale).

For the common reception time, the receiver can generate a measurement at any instant of the receiver clock. From the figure, $t_u(k)$ denotes the reference epoch, in which the measurement is generated. δt_{SVi}, where $i = \{1, 2, 3, 4\}$ denotes the relative delay from the measurement to the starting edge of the TLM word for the SVs. The illustration in Figure 3.19 is a little simplified. In fact, the receiver does not calculate the delay all the way back to the TLM word, but only requires that the measurements are referenced to the same bit-number in the navigation message.

It should be noted, that the relative arrival times does not give us an absolut measure for the real transmission times from SV to receiver. However, it often assumed that the range of transmission time fall within the range of $65 - 85ms$, depending on the location. The receiver would initially, then assume a start offset of approx. $70ms$ and use the SV with the first received TLM word as a reference.

In the subsequent position estimation, the receiver solves for both receiver position and clock-bias, hence the initial start offset can be refined. As mentioned, the resolution of the receiver time scale is inversely proportional to the sampling frequency, as an example if the front-end samples the IF data with 10 MHz, the resulting resolution would be 30 meters. Luckily, this resolution can be improved by taking the residual code-phase into account. The residual code-phase, is a result of two conditions; the first is that the GPS L1 C/A code sequence, due to code Doppler not is exactly 1 ms and secondly since the sampling rate is finite, there would be a mismatch between the actual correlation length in terms of code-phase and what the receiver is able to replicate.

For more information of the TLM word and GPS L1 C/A navigation message, refer to [GPS06]. For more information of pseudorange generation, refer to [LA10], [PFP12], [BAB+07], [Tsu00] and [WBH06].

3.4.2 Doppler

The Doppler measurement also known as the instantaneous Doppler is very simple to extract from the tracking loops. This measurement is generated by subtracting the nominal IF frequency from the tracked carrier-frequency at the receiver measurement epoch. The Doppler measurement is used to solve for user velocity, which is explained in a later section.

3.4.3 Carrier Phase / Accumulated Doppler Range

Measurements from carrier tracking can potentially yield much more precise results, than from the code tracking. The wave-length of the 1575.42 MHz GPS L1 carrier is approximately 19 cm, where in comparison the chip-length of the CA-code corresponds to 293 m. A code-based PVT solution will typical yield accuracies of a few meters, whereas the carrier measuremnt potentially can give cm accuracies.

The fundamental equation of carrier-measurements (in cycles) can be written as [ME12]:

$$\phi(t) = \phi(t_0) + \int_{t_0}^{t} f(s)\, ds \tag{3.53}$$

where $\phi(t)$ is the phase measurement in cycles, $\phi(t_0)$ is the initial phase offset and $f(s)$ is the time-varying Doppler frequency. The measurement can be generated when the receiver has phase-lock on a SV. Conceptually, the receiver continuously computes the number of carrier phase cycles in a given time frame. When a SV is approaching the Doppler frequency is negative, resulting in fewer counted cycles. Conversely, when the SV is moving away from the receiver the Doppler frequency is positive which would give an increase in carrier cycles. For this reason, the carrier phase measurements are also commonly referred to as the integrated or accumulated Doppler range. The general principle is illustrated in Figure 3.20.

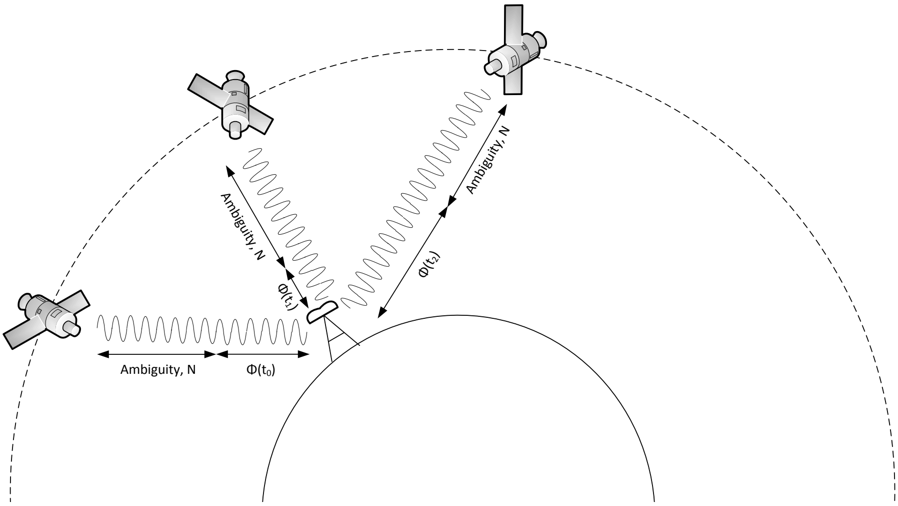


Figure 3.20: Principle in Carrier Phase Measurements.

The Figure shows a GNSS SV at three different epochs. For all measurements a constant number of whole cycles is said to be ambiguous, i.e. the receiver cannot resolve how many wavelengths there is between the SV and receiver, but only find the relative change between epochs.

The carrier-phase measurement can be modelled as [ME12]:

$$\phi = \lambda^{-1}[r + I_\phi + T_\phi] + \frac{c}{\lambda}(\delta t_u - \delta t^s) + N + \epsilon_\phi \tag{3.54}$$

where N is the integer ambiguity, λ is the wavelength, i.e., $\lambda = \frac{c}{1575.42\text{MHz}}$. T_ϕ, I_ϕ is the tropospheric and ionospheric path delays and ϵ_ϕ is measurement noise.

The carrier phase observable, although potentially much more precise than pseudoranges is also a lot more vulnerable. Very brief blockages of the LOS between user and SV would cause the PLL to estimate the carrier-frequency wrongly, hence corrupting the measurement. This effect is known as cycle-slips. A brief temporary blockage does not affect the code-based pseudoranges. In addition to added precision, the carrier phase observable is also much less affected by multipath than pseudoranges. In fact, the maximum disturbance that can be encountered from a multipath signal is only a quarter of the wavelength. Approx

5 cm for GPS L1.

3.5 Navigation Processing

In this section, algorithms for calculating the receiver position and velocity are given. The section will focus on position determination using pseudoranges and velocity determination from the instantaneous Doppler measurement.

3.5.1 Position Determination using Least Squares

The true geometric range between receiver and SV can be stated as:

$$r_i = \sqrt{(X_i - X)^2 + (Y_i - Y)^2 + (Z_i - Z)^2} \tag{3.55}$$

where r_i is the geometric range to the i-th SV, $\{X_i, Y_i, Z_i\}$ represents the Earth Centered Earth Fixed (ECEF) coordinates of the SV, obtained from the navigation message. $\{X, Y, Z\}$ represents the ECEF referenced user position.

The generalized model for the pseudorange from eq. (3.52) is recalled to be:

$$\rho_i = r_i + c\,[\delta t_u - \delta t_i^s] + I_{p,i} + T_{p,i} + e_i \tag{3.56}$$

The ionospheric and tropospheric path-delays can be reduced from, e.g., the Saastamoinen and Klobuchar models. Similar the SV clock offset, δt_i^s, can be obtained from the navigation message. A corrected pseudorange model can thus be stated as:

$$\bar{\rho}_i = r_i + c\delta t_u + \epsilon_i \tag{3.57}$$

where ϵ_i is a noise term, consisting of residual errors from the atmospheric corrections, multipath and tracking errors.

A note regarding the SV coordinates, $\{X_i, Y_i, Z_i\}$, these are generally estimated for the time of transmission. However, as the earth rotates, and the ECEF frame is tied to this rotation, the ECEF frame is different at the time of reception. The coordinates are usually corrected by a rotation matrix, where the propogation time from SV to receiver is used to determine the correct adjustment, see e.g., [ME12] or [SZHP13] for details of this. In the following we assume that SV coordinates are referenced to an ECEF frame at reception time.

From eq. (3.55-3.57), it should be clear that in order to obtain the user position, 4 unknown quantities needs to be estimated. We can define a vector of unknown parameters as:

$$\boldsymbol{\theta} = [\begin{array}{cccc} X & Y & Z & c\delta t_u \end{array}] \tag{3.58}$$

As the geometric range, r_i, is a nonlinear function of the user-coordinates, eq. (3.57) needs to be linearized in order to solve for the user position and receiver clock offset using a least squares (LS) principle. The linearization is performed by choosing an initial starting point (or best guess) of the position and clock offset, and thus linearize eq. (3.57) around this point. We define the starting point as:

$$\boldsymbol{\theta}_0 = [\begin{array}{cccc} X_0 & Y_0 & Z_0 & c\delta t_{u,0} \end{array}]$$

and assume $\boldsymbol{\theta} \approx \boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}$

The linearized variant of eq. (3.57) can be stated as:

$$\bar{\rho}_i(\boldsymbol{\theta}) \approx \bar{\rho}_i(\boldsymbol{\theta}_0) + \frac{\partial \bar{\rho}_i(\boldsymbol{\theta})}{\partial X}\Delta X + \frac{\partial \bar{\rho}_i(\boldsymbol{\theta})}{\partial Y}\Delta Y + \frac{\partial \bar{\rho}_i(\boldsymbol{\theta})}{\partial Z}\Delta Z + \frac{\partial \bar{\rho}_i(\boldsymbol{\theta})}{\partial c\delta t_u}\Delta c\delta t_u \tag{3.59}$$

where the partial derivatives are calculated as:

$$\left.\frac{\partial \bar{\rho}_i(\boldsymbol{\theta})}{\partial X}\right|_{X=X_0} = \frac{X_i - X_0}{r_i}, \quad \left.\frac{\partial \bar{\rho}_i(\boldsymbol{\theta})}{\partial Y}\right|_{Y=Y_0} = \frac{Y_i - Y_0}{r_i}$$

$$\left.\frac{\partial \bar{\rho}_i(\boldsymbol{\theta})}{\partial Z}\right|_{Z=Z_0} = \frac{Z_i - Z_0}{r_i}, \quad \left.\frac{\partial \bar{\rho}_i(\boldsymbol{\theta})}{\partial c\delta t_u}\right|_{c\delta t_u=c\delta t_{u,0}} = 1$$

From the above linearization, we can form a linear set of equations on the form

$\mathbf{b} = \mathbf{A}\mathbf{x}$:

$$\begin{pmatrix} \bar{\rho}_1 - r_1 - c\delta t_{u,0} \\ \bar{\rho}_2 - r_2 - c\delta t_{u,0} \\ \vdots \\ \bar{\rho}_N - r_N - c\delta t_{u,0} \end{pmatrix} = \begin{pmatrix} \frac{X_1 - X_0}{r_1} & \frac{Y_1 - Y_0}{r_1} & \frac{Z_1 - Z_0}{r_1} & 1 \\ \frac{X_2 - X_0}{r_1} & \frac{Y_2 - Y_0}{r_1} & \frac{Z_1 - Z_0}{r_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{X_N - X_0}{r_1} & \frac{Y_N - Y_0}{r_1} & \frac{Z_N - Z_0}{r_1} & 1 \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \Delta c\delta t_u \end{pmatrix} \tag{3.60}$$

The above system of equations can thus be solved using an iterated least-squares algorith, such as Newton-Raphson. The algorithm would typically be repeated, including subsequent linearizations until the step-size of the $\Delta\boldsymbol{\theta}$ estimates reaches a predefined lower bound.

A classical metric, known as Dillution-Of-Precision (DOP) could also be derived from the **A** matrix in eq. (3.60), since the geometry of SVs can be related to the expected accuracy of the position solution, see e.g. [ME12]. The DOP metric does not take local reception conditions, such as multipath and NLOS into account and has not been considered further in this thesis.

3.5.2 Velocity Determination using Least Squares

The most simple way of determining velocity is to simply differentiate subsequent position solutions. This strategy is however inaccurate, due to potentially large errors between epochs. A more accurate algorithm utilizes the instantaneous Doppler measurements, where accuracies in the range of a few cm/s can be obtained. An even more precise approach is to use differentiated carrier-phase measurements between epochs, this can potentially give accuracies within mm/s, but the downside is that cycle-slips can cause problems for that method [FAGT15].

Here the instantaneous Doppler-based method is described. It is common to convert the Doppler measurement into a pseudorange-rate observable, given as:

$$\dot{\rho}_i = -f_{Doppler,i} \frac{c}{f_L} \tag{3.61}$$

where $f_{Doppler,i}$ is the recorded Doppler for the i-th SV, c is the speed of light and f_L is the nominal carrier frequency of the GNSS signal.

We furthermore define a unit-vector, representing the direction of LOS to the i-th SV as:

$$\mathbf{u}_i = \frac{\mathbf{r_i} - \mathbf{r}}{|\mathbf{r_i} - \mathbf{r}|} \tag{3.62}$$

where $\mathbf{r}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$ and $\mathbf{r} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$

Given the unit-vector for LOS, pseudorange-rate and SV velocities we can then solve for user velocity as [ME12]:

$$\begin{pmatrix} \dot{\rho_1} - V_1 \cdot \mathbf{u}_1 \\ \dot{\rho_2} - V_2 \cdot \mathbf{u}_2 \\ \vdots \\ \dot{\rho_N} - V_N \cdot \mathbf{u}_N \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1(1) & \mathbf{u}_1(2) & \mathbf{u}_1(3) & 1 \\ \mathbf{u}_2(1) & \mathbf{u}_2(2) & \mathbf{u}_2(3) & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_3(1) & \mathbf{u}_3(2) & \mathbf{u}_3(3) & 1 \end{pmatrix} \begin{pmatrix} \Delta\dot{X} \\ \Delta\dot{Y} \\ \Delta\dot{Z} \\ \Delta c\dot{\delta} t_u \end{pmatrix} \tag{3.63}$$

where V_i is the SV velocity and $\Delta c\dot{\delta} t_u$ is the receiver clock drift. It should be noted that eq. (3.63) can be solved with ordinary LS, opposed to position determination. The algorithms required for determining SV position and velocities from navigation message parameters are described in, e.g., [Rem04] [Zha07].

3.5.3 Estimating Position and Velocity using a Kalman Filter

The LS methods for estimation position and velocity assumed no apriori information on dynamics of the navigation object. Furthermore, position and velocity was determined by separate algorithms, although a high degree of correlation obviously must exist. The advantages of using a EKF for determining position and velocity, can be summarized as;

- The EKF model exploits the obvious correlation between position and velocity

- The EKF makes a weighted average between the measurements and prediction model based on reported measurement uncertainty.

The algorithm was proposed in [BH97], but is repeated here for completeness. The model is based on a random walk (integrated white-noise) model to describe variations in velocity. The filter is also known as a Position-Velocity (PV) EKF. It furthermore includes a 2-state clock model to describe the imperfections of the receiver clock/oscillator. It should be noted that this model, is derived for a kinematic dynamics profile. For static surveys, a random walk model influencing position only is more truthful. In Figure 3.21, an illustration of the dynamic model for one-dimension is shown:

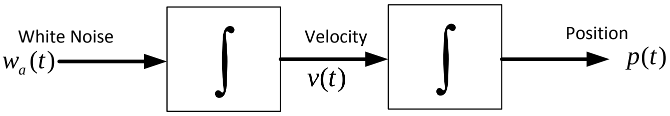


Figure 3.21: PV dynamics model.

The continuos time dynamics model for the PV model (for one dimension) can thus stated as:

$$\dot{p}(t) = v(t)$$

$$\dot{v}(t) = w(t)$$

where $v(t)$ is one-dimensional velocity, $w(t)$ is a white noise process, characterized by $w \sim \mathcal{N}(0, S_p)$. S_p should generally be selected with consideration to the expected dynamics of the navigation platform.

In state-space, where $x(t) = [p(t)\ v(t)]^T$, the model can be expressed as:

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}} x(t) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{G}} w(t) \qquad (3.64)$$

The continous system/dynamics matrix, $\mathbf{A}$, can be converted to a discrete model with sample time Δt as [BH97],

$$\mathbf{\Phi} = e^{\mathbf{A}\cdot\Delta t} = I + \mathbf{A}\Delta t + \frac{(\mathbf{A}\Delta t)^2}{2!} + \frac{(\mathbf{A}\Delta t)^3}{3!} \dots$$

which in our case yields:

$$\mathbf{\Phi}_{PV} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{3.65}$$

The discrete-time process-noise covariance, $\mathbf{Q}_{PV}$ can be computed as:

$$\mathbf{Q}_{PV} = \int_0^{\Delta t} \Phi(\Delta t, \tau) G(\tau) S_p(\tau) G(\tau)^T \Phi(\Delta t, \tau)^T \, d\tau \tag{3.66}$$

$$\mathbf{Q}_{PV} = S_p \int_0^{\Delta t} \begin{bmatrix} \tau^2 & \tau \\ \tau & 1 \end{bmatrix} d\tau = S_p \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \tag{3.67}$$

The clock model, governing the dynamics of the clock behavior is illustrated in Figure 3.22.

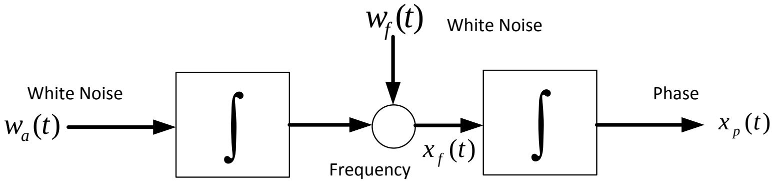


Figure 3.22: 2-state clock model.

The clock model is driven by 2 white noise processes. $w_a(t)$ represent fluctuations in frequency-rate and $w_f(t)$ denotes variations in frequency.

The white noise, processes is characterized by the following stochastics, $w_a \sim \mathcal{N}(0, S_g)$ and $w_f \sim \mathcal{N}(0, S_f)$.

The variances or Power Spectral Densities (PSD) are given as:

$$S_f \approx c^2 \cdot h_0/2 \quad S_g \approx c^2 \cdot 2\pi^2 h_{-2} \tag{3.68}$$

where h_0 and h_{-2} is the Allan variance parameters of the clock (oscillator).

The transition matrix and process noise matrix, can be derived similarly to the PV model and is thus:

$$\mathbf{\Phi}_c = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{3.69}$$

$$\mathbf{Q}_c = \begin{bmatrix} S_f \Delta t + \frac{S_g \Delta t^3}{3} & \frac{S_g \Delta t^2}{2} \\ \frac{S_g \Delta t^2}{2} & S_g \Delta t \end{bmatrix} \tag{3.70}$$

The state-vector and a combined transition model for 3-dimensions and including the clock-model can be stated as:

$$\mathbf{x} = \begin{bmatrix} \Delta X \\ \Delta \dot{X} \\ \Delta Y \\ \Delta \dot{Y} \\ \Delta Z \\ \Delta \dot{Z} \\ \Delta c\delta t_u \\ \Delta c\dot{\delta} t_u \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_{PV} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_{PV} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Phi}_{PV} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\Phi}_c \end{bmatrix} \tag{3.71}$$

Similarly, the process noise covariance can be stated as:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{PV} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{PV} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{PV} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_c \end{bmatrix} \tag{3.72}$$

The measurement model of the EKF, has the general form:

$$
\begin{aligned}
&\begin{bmatrix} \bar{\rho}_1 - r_1 - c\delta t_{u,0} \\ \dot{\rho_1} - V_1 \cdot \mathbf{u}_1 \\ \bar{\rho}_2 - r_2 - c\delta t_{u,0} \\ \dot{\rho_2} - V_2 \cdot \mathbf{u}_2 \\ \vdots \\ \bar{\rho}_N - r_N - c\delta t_{u,0} \\ \dot{\rho_N} - V_N \cdot \mathbf{u}_N \end{bmatrix} = \\
&\begin{bmatrix} \mathbf{u}_1(1) & 0 & \mathbf{u}_1(2) & 0 & \mathbf{u}_1(3) & 0 & 1 & 0 \\ 0 & \mathbf{u}_1(1) & 0 & \mathbf{u}_1(2) & 0 & \mathbf{u}_1(3) & 0 & 1 \\ \mathbf{u}_2(1) & 0 & \mathbf{u}_2(2) & 0 & \mathbf{u}_2(3) & 0 & 1 & 0 \\ 0 & \mathbf{u}_2(1) & 0 & \mathbf{u}_2(2) & 0 & \mathbf{u}_1(3) & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_N(1) & 0 & \mathbf{u}_N(2) & 0 & \mathbf{u}_N(3) & 0 & 1 & 0 \\ 0 & \mathbf{u}_N(1) & 0 & \mathbf{u}_N(2) & 0 & \mathbf{u}_N(3) & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta \dot{X} \\ \Delta Y \\ \Delta \dot{Y} \\ \Delta Z \\ \Delta \dot{Z} \\ \Delta c\delta t_u \\ \Delta c\dot{\delta t}_u \end{bmatrix} + v_k
\end{aligned}
\tag{3.73}
$$

where $v_k \sim \mathcal{N}(0, \mathbf{R})$ is the measurement noise. The measurement noise matrix is an identity matrix consisting of the anticipated uncertainties of the pseudoranges and pseudorange-rates.

The EKF recursions are executed in the same manner, as described for the EKF-based tracking in section 3.3.5.

3.5.4 Weighting of observations

When solving for a position solution, it is common to implement different weights on the individual pseudoranges. Typically, this could be based on elevation angle, as shown below:

$$\sigma_{z_i}^2 = \frac{\sigma_z^2}{sin^2(\epsilon_i)} \tag{3.74}$$

where σ_{zi}^2 is the variance for the i'th SV, ϵ is the elevation angle of the SV and σ_z is the assumed variance for pseudorange measurements in zenith direction. Other schemes are based on the received signal strength or a combination of both.

3.6 Vector Tracking

A vector tracking receiver combines the task of SV tracking and position/velocity estimation. This is in contrast to conventional (scalar) GNSS receivers, where tracking of each SV is performed completely independently for each channel.

The underlying assumption for vector tracking algorithms is that by utilizing the position and velocity solution from the observables, which implicitly determines the coupling between measurements, noise can be reduced for all the channels [Jr.96]. The coupling between measurements can be accounted to be the common receiver clock-bias and the receiver dynamics. As described in section 3.3.6, if the noise bandwidth of the tracking channels can be reduced, this would both lead to lower thresholds in terms of C/N_0 and more precise observables. Vector tracking receivers have also shown to perform better with regards to interference immunity [Ben07], tracking during ionospheric scintillations [DM13] and in weak signal environments [LB07]. Furthermore, Vector tracking has also shown to provide an efficient resistance to multipath and NLOS signal [HJGK15].

The term 'Vector tracking' was first introduced by [Jr.96], where a Vector-DLL (VDLL) implementation was described and the potential benefits highlighted. A VDLL replaces the individual (scalar) DLL's for each tracking channel and use the navigation solution, typically obtained from an EKF, to predict the code-phase of the individual tracked SVs.

At this time, there have been suggested numerous implementation architectures for vector tracking receivers. In Figure 3.23 an overview of different implementation options is given.

In a coherent centralized architecture the correlator outputs are used directly in the measurement model of the Navigation filter. This leads to a heavy processing load, as all correlator outputs from each channel has to be processed within a single filter. [Las09] conducted a number of Monte Carlo simulations for a Vector DLL-FLL (VDFLL) with cascaded (federated) tracking pre-filters and a centralized variant and found the difference to be virtually non-existing. The final implementation option is to propagate discriminator outputs from each tracking channel directly to the filter, this is known as a cascaded non-coherent approach. The most uncommon variant of vector implementation is the Vector PLL (VPLL). The reason, why this variant is seldom used, is if a cycle-slip occurs in one channel that would negative adverse the phase tracking for all channels. This vulnerability is also true for defective or corrupted measurements of code-phase (pseudorange) and/or Doppler (pseudorange-rate). A corrupted measurement in one channel will negatively adverse performance in

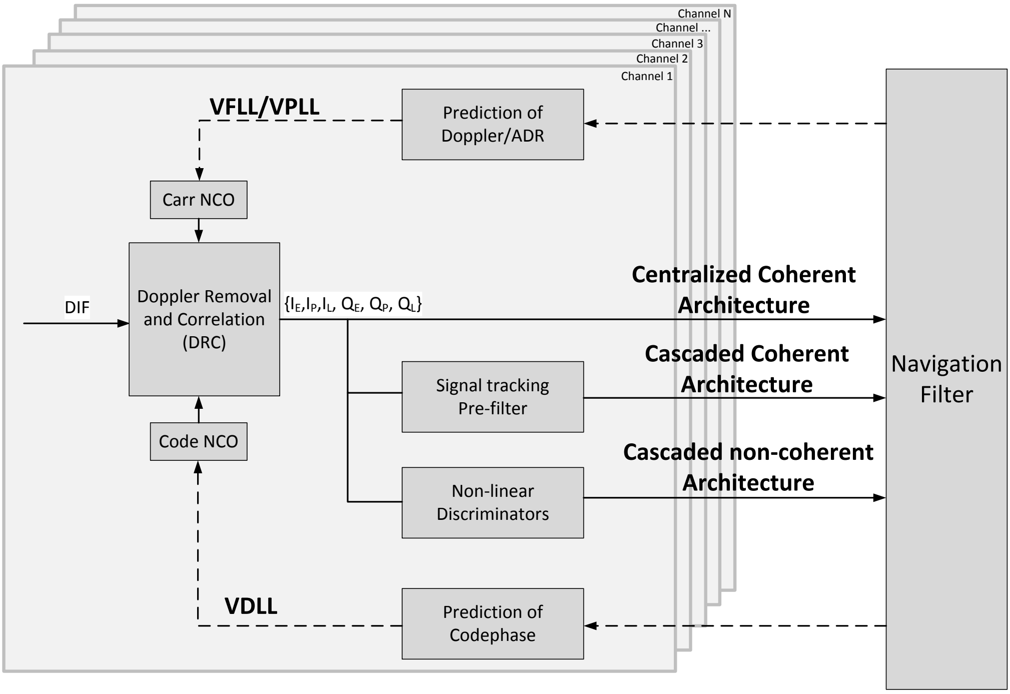

Channel N
Channel ...
Channel 3
Channel 2
Channel 1
VFLL/VPLL
Prediction of Doppler/ADR
Carr NCO
DIF
Doppler Removal and Correlation (DRC)
{I_E,I_P,I_L, Q_E, Q_P, Q_L}
Centralized Coherent Architecture
Cascaded Coherent Architecture
Signal tracking Pre-filter
Navigation Filter
Code NCO
Cascaded non-coherent Architecture
Non-linear Discriminators
VDLL
Prediction of Codephase

Figure 3.23: Vector receiver architecture classification. Figure is inspired from [Bha12]

all channels and in extreme cases could cause instability and loss-of-lock for all SVs. This is often considered the main drawback of vector tracking receivers. However, Doppler and code-phase measurements are fortunately, much more robust in nature than ADR measurements. A note regarding the non-coherent and coherent architectures. The distinction is that coherent architectures uses the correlator outputs directly in subsequent filter(s), where the non-coherent use the discriminator function provide measurements to the subsequent filter. It is also implicitly that coherent architectures maintains carrier phase tracking. Generally, non-coherent variants can sustain tracking at lower C/N_0 than coherent variants. However, the use of non-linear discriminators also introduces non-linear modelling errors and hence generally have noisier measurements than coherent designs. The choice of architecture then constitutes a trade-off between maximum tracking sensitivity and measurement noise [GMM07].

The remaining part of this chapter focuses on a coherent cascaded architecture, which has been implemented for this study. The implementation has been in-

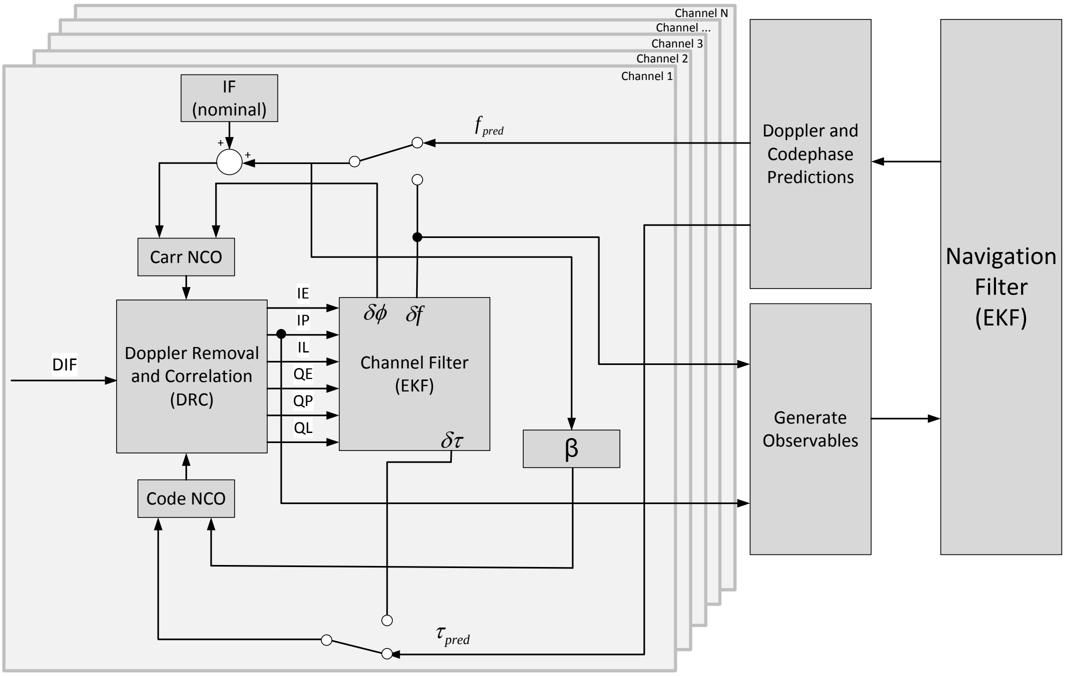


Figure 3.24: Cascaded coherent vector-tracking architecture.

spired from [PL06]. A high-level functional diagram of this system is illustrated in Figure 3.24.

The DRC block operation is the same as Figure 3.16. The channel filter, has the same architecture as described in section 3.3.5. The navigation filter is similarly, identical to the algorithm in section 3.5.3. The binary switches in the figure indicates, that the channel filter can work in standard (or scalar) processing mode, without utilizing the vector architecture.

For traditional tracking based on discriminators and loop-filters, the NCOs to code and carrier generation was based on phase rate-only (or frequency) feedback. In this figure, the NCOs are also steered with phase feedback, as the channel filter estimate both quantities. For the scalar-case, the channel filter controls the DRC process without aiding from the navigation filter. When vector-tracking is utilized, the carrier frequency-error and code-phase errors are instead obtained from a prediction based on the most current PVT solution. It should be noted, that the carrier-phase error in both cases are obtained from the channel filter, for the same reason as was explained regarding a VPLL. The implementation also features carrier aided code tracking, in order to remove LOS dynamics from the code-tracking.

The prediction of code-phase and Doppler for each SV is obtained by predicting a future position solution and receiver clock error for the next scheduled measurement epoch. Figure 3.25 illustrates the timing of the predictions.

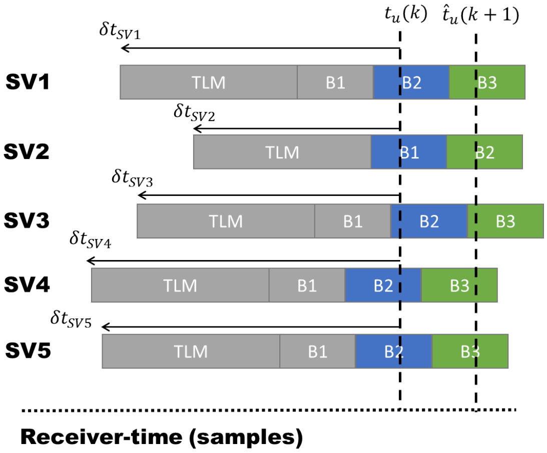


Figure 3.25: Timing for code-phase and Doppler predictions.

$t_u(k)$ represents the current measurement epoch (measured in samples). The navigation bits the receiver is currently tracking is indicated as the blue boxes. The scheduled update is indicated by $\hat{t}_u(k+1)$.

The algorithm uses the prediction model of the EKF Navigation Filter in section 3.5.3 to estimate future position, velocity and clock-states. The estimates are used together with predicted SV position and velocities (obtained from the navigation message) to form predicted pseudoranges and Doppler for each SV. The carrier Doppler can be calculated from predicted velocity and receiver clock drift:

$$\hat{f}^{i,-}_{pred} = -\frac{f_L}{c}\left((\mathbf{v}^i - \hat{\mathbf{v}}^-_u)\cdot\hat{\mathbf{u}}^{i,-}\right) - \frac{f_L}{c}c\hat{\dot{\delta t}}^-_u \tag{3.75}$$

where i denotes the SV index. $\hat{f}^{i,-}_{pred}$, is the predicted Doppler to steer the NCO, $\hat{\mathbf{v}}^-_u$ is the predicted user velocity, $\mathbf{v}^i$ is the SV velocity derived from parameters of the navigation message. $c\hat{\dot{\delta t}}^-_u$ is the predicted clock-drift and $\hat{\mathbf{u}}^{i,-}$ is the predicted LOS vector. f_L is the nominal carrier frequency.

The code-phase prediction is given as:

$$\hat{\tau}_{pred}^{i,-} = \quad \mathrm{mod}\left(\hat{t}_u \cdot f_c - \frac{\|\mathbf{r}^i - \hat{\mathbf{r}}_u\| + T^i + I^i + c\delta\hat{t}_u^- - c\delta t_s^i + cT_{GD}^i}{c/f_c}, L\right) \tag{3.76}$$

where $\hat{\tau}_{pred}^{i,-}$ is the predicted code-phase estimate, $\hat{t}_u$ is the scheduled receiver time for next measurement, $\mathbf{r^i}$ is the predicted SV position and $\hat{\mathbf{r}}_u$ is the predicted receiver position. T^i, I^i is the modelled ionospheric and troposheric delays. $c\delta\hat{t}_u^-$ is the predicted receiver clock-error, $c\delta t_s^i$ is the SV clock error, estimated from the navigation message and cT_{GD}^i is the SV group delay. For GPS L1 C/A code, the chipping frequency and code-length is: $f_c = 1.023 Mchips/s$ and $L = 1023$.

The above equations assumes that the receiver updates the NCOs in the middle of an integration interval. If the updates should be related to the start of an integration interval (see Figure 3.25), the propogation time of the predictions should generally be varied for each SV.

A final note on vector tracking, in general the tracking sensitivity is related to the accuracy of the positioning solution and hence the number of tracked SVs [LB07]. A further benefit, is that vector tracking is able to bridge outage of SVs for shorter intervals, without having to initiate reacquisition.

Chapter 4

Inertial Navigation

In this chapter, a brief overview of Inertial Navigation Systems (INS) is given. For a more comprehensive review of INS, see e.g. [Gro13], [NKG13] and [TW04].

An INS employs accelerometers and gyroscopes to calculate the motion of a platform. Additionally sensors, such as magnetometers or barometers can also be included. The ensemble of sensors for the INS are typically placed in a single enclosure, known as a Inertial Measurement Unit (IMU).

Unlike GNSS, an INS is self-contained, in the sense that no external signals or infrastructure is required for operation. In addition, an INS usually resolves velocity, position and orientation (attitude) of the navigation object. INS is a dead-reckoning system, i.e., the calculation of a position is based on previous positions and measured velocity between updates. A generic diagram of a inertial navigation processor is shown in Figure 4.1.

The INS needs to be initialized before continuous operation, i.e., the position, velocity and orientation must be provided on start-up.

After the initialization process the navigation processor use angular rate and specific force measurements to calculate attitude, velocity and position of the system. It should be noted that the attitude update is the initial step as the sensed accelerations should be referenced to the current attitude. A gravity

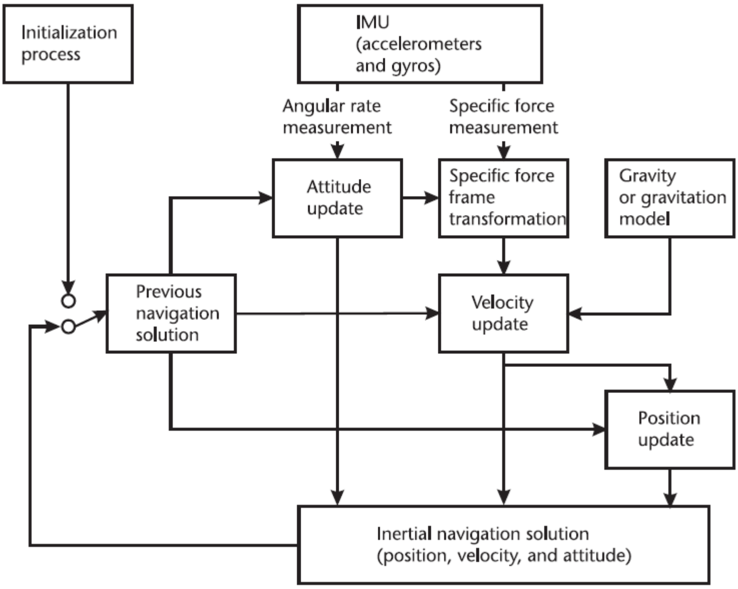

Initialization process
IMU (accelerometers and gyros)
Angular rate measurement
Specific force measurement
Attitude update
Specific force frame transformation
Gravity or gravitation model
Previous navigation solution
Velocity update
Position update
Inertial navigation solution (position, velocity, and attitude)

Figure 4.1: Flowchart of Inertial Navigation System. Figure adopted from [Gro13].

model is also supplied to account for varying gravitational acceleration across the earth. The update equations for attitude, velocity and position is commonly referred to as the INS mechanization or the inertial navigation equations. The equations are dependent of which coordinate frame the system is resolved in. The next section will briefly outline some common coordinate frames.

4.1 Coordinate Frames

In terms of describing motion based on inertial measurements, it is important to realize that an IMU usually is comprised of three perpendicular axes and hence all IMU measurements are referenced to the sensor-frame of the unit. The motion of an object, in this case the IMU, needs to expressed in relatively terms to another coordinate frame. This frame can be either locally or globally founded.

4.1.1 Earth-Centered Inertial Frame (ECI)

An Earth-Centered Inertial (ECI) frame has the following characteristics [GWA07]:

- Origin in the mass center of the earth.
- z-axis is along Earth's rotation axis and passes through the conventional terrestial pole.
- x-y plane spans the equator and the x-axis point towards the vernal equinox.

The ECI frame is unlike the following coordinate frames not rotating with earth. The term inertial strictly means, that the coordinate frame does not accelerate or rotate with respect to the universe. This is obviously not entirely true, as the earth orbits around the sun and the earth's rotation axis also encounters smaller variations. For practical purposes, the ECI frame can thus be treated as inertial frame [Gro13]. The ECI frame is convenient in terms of inertial navigation, as the accelerometers and gyroscopes measures motion with respect to an inertial frame. The ECI frame is also useful as a coordinate frame for satellites, as they move in elleptical orbits independent of Earth's rotation.

4.1.2 Earth-Centered Earth-Fixed Frame (ECEF)

An ECEF frame has similar to the ECI frame the origin at the mass center of the earth. The ECEF frame is however rotating with the earth and the x-axis is defined as going through the Greenwich meridian. The ECEF frame can be expressed in either cartesian coordinates or curvilinear coordinates. GNSS SV coordinates and user position are natively referenced to an ECEF frame. Conversion between cartesian coordinates from the ECI to the ECEF frame is simple, since the center and z-axis is coincident between the ECI and ECEF frame and the rotation rate of the earth is constant. More specifically we have:

$$\mathbf{r}^e = \mathbf{C}_i^e \mathbf{r}^i \tag{4.1}$$

where $\mathbf{r}^e$ is the position vector in ECEF coordinates, $\mathbf{r}^i$ is the position vector in the ECI frame. $\mathbf{C}_i^e$ is the coordinate transformation matrix from ECI to ECEF, this type of matrix is also commonly referred to as a direction cosine matrix

(DCM) and given as [Gro13]:

$$\mathbf{C}_i^e = \begin{bmatrix} \cos\omega_{ie}(t-t_0) & \sin\omega_{ie}(t-t_0) & 0 \\ -\sin\omega_{ie}(t-t_0) & \cos\omega_{ie}(t-t_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

where ω_{ie} is the Earth's rotation rate. t_0 was the latest epoch in which the ECEF x-axis was coincident with the ECI frame.

4.1.3 Navigation Frame

A navigation frame or local level frame, is a coordinate frame useful for expressing the movement of a vehicle. The origin of the frame coincide with the sensor frame of the IMU. The x-axis points towards true north, the y-axis points east and the z-axis points toward the centre of the earth. This convention is known as a north-east-down (NED) frame. Another common convention is the east-north-up (ENU) frame, which is similar but here the z-axis is inverted compared to NED convention. A coordinate transformation matrix from ECEF to NED can be expressed as [Gro13]:

$$\mathbf{C}_e^n = \begin{bmatrix} -\sin L_b\cos\lambda_b & -\sin L_b\sin\lambda_b & \cos L_b \\ -\sin\lambda_b & \cos\lambda_b & 0 \\ \cos L_b\cos\lambda_b & -\cos L_b\sin\lambda_b & -\sin\lambda_b \end{bmatrix} \tag{4.3}$$

where L_b is the geodetic latitude and λ_b is geodetic longitude.

An illustration of how the ECI, ECEF and navigation frame (ENU) is related can be seen in Figure 4.2.

4.1.4 Body Frame

The body frame or vehicle frame is for all strap-down inertial sensors fixed relative to the sensor-frame, i.e., the frame the IMU measurements are referenced to. The origin of the body frame coincide with that of the navigation frame, but the axis are fixed with respect to the body. An illustration of body frame axes for an aircraft is illustrated in Figure 4.3.

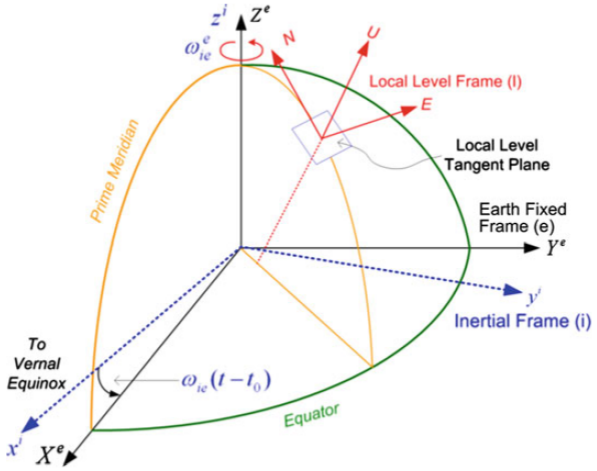


Figure 4.2: Comparison of ECI, ECEF and navigation (ENU) coordinate frames. Figure adopted from [NKG13]

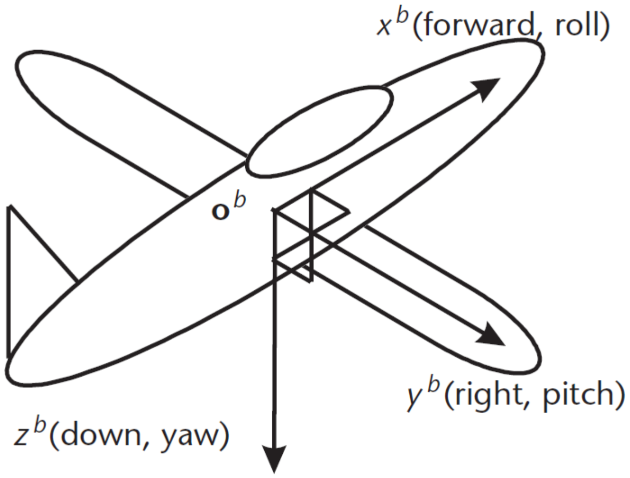


Figure 4.3: Body frame of aircraft. Figure adopted from [Gro13].

The attitude of a body is always expressed relative to another reference frame. Commonly attitude is resolved around the navigation frame, i.e., the heading or yaw angle of a body is usually in relation to north.

4.2 INS Mechanizations in ECEF Frame

In this section the INS mechanizations governing the movement of a body resolved in an ECEF frame are presented. As Figure 4.1 indicated, the mechanizations are based on three main computational steps. This section is largely inspired from [Gro13].

Attitude Update

In continuos time, the attitude update in an ECEF frame is given by the following first order differential equation:

$$\dot{\mathbf{C}}_b^e = \mathbf{C}_b^e \boldsymbol{\Omega}_{eb}^b = \mathbf{C}_b^e \boldsymbol{\Omega}_{ib}^b - \boldsymbol{\Omega}_{ie}^e \mathbf{C}_b^e \tag{4.4}$$

where $\mathbf{C_b^e}$ is the coordinate transformation matrix from body to ECEF. $\boldsymbol{\Omega_{ib}^b}$ is the skew-symmetric form of the sensed angular-rates and $\boldsymbol{\Omega_{ie}^e}$ is the skew symmetric form of the earth's rotation. These terms are given as:

$$\boldsymbol{\Omega_{ib}^b} = \begin{bmatrix} 0 & -\omega_{ib,z}^b & \omega_{ib,y}^b \\ \omega_{ib,z}^b & 0 & -\omega_{ib,x}^b \\ -\omega_{ib,y}^b & \omega_{ib,x}^b & 0 \end{bmatrix}, \quad \boldsymbol{\Omega_{ie}^e} = \begin{bmatrix} 0 & -\omega_{ie} & 0 \\ \omega_{ie} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

where $\{\omega_{ib,x}^b, \omega_{ib,y}^b, \omega_{ib,z}^b\}$ is the sensed angular-rates from the gyroscope and ω_{ie} is the earth's rotation rate.

A discrete approximation of the attitude update equation is given as:

$$\mathbf{C}_b^{e+} \approx \mathbf{C}_b^{e-}(\mathbf{I}_3 + \boldsymbol{\Omega}_{ib}^b \tau_i) - \boldsymbol{\Omega}_{ie}^e \mathbf{C}_b^{e-} \tau_i \tag{4.6}$$

where τ_i is the time between IMU measurements. The superscripts, $^+$,$^-$, denotes the updated and previous solutions respectively. Note that eq. (4.6) assumes that the angular rate is constant during the integration interval.

Velocity Update

The velocity update equation is calculated with respect to the current attitude. In discrete time, the update equation can be approximated as:

$$\mathbf{v}_{eb}^{e+} \approx \mathbf{v}_{eb}^{e-} + (\mathbf{f}_{ib}^{e} + \mathbf{g}_{b}^{e}(\phi^{-}, h^{-}) - 2\mathbf{\Omega}_{ie}^{e}\mathbf{v}_{eb}^{e-})\tau_i \tag{4.7}$$

where

$$\mathbf{f}_{ib}^{e} \approx 0.5(\mathbf{C}_{b}^{e-} + \mathbf{C}_{b}^{e+})\mathbf{f}_{ib}^{b} \tag{4.8}$$

where $\mathbf{f}_{ib}^{b}$ is the sensed specific force, $\mathbf{g}_{b}^{e}(\phi^{-}, h^{-})$ denotes a gravity model as function of height and geodetic latitude.

Position Update

Finally, the position update equation can be approximated as:

$$\mathbf{r}_{eb}^{e+} \approx \mathbf{r}_{eb}^{e-} + \mathbf{v}_{eb}^{e-}\tau_i + (\mathbf{f}_{ib}^{e} + \mathbf{g}_{b}^{e}(\phi^{-}, h^{-}) - 2\mathbf{\Omega}_{ie}^{e}\mathbf{v}_{eb}^{e-})\frac{\tau_i}{2} \tag{4.9}$$

The discrete approximations in the equations above are generally obtained from a truncation to first order of a power-series expansion. Furthermore it was assumed that variations over integration intervals was negligible. For high-end INS systems, the equations above could lead to a less than optimal performance, whereas for lower grade INS the difference between higher order approximations and the presented forms should be neglible [Gro13].

4.3 INS Errors

An INS exhibits errors from three main sources:

1. IMU errors.
2. Initialization errors.
3. INS mechanization errors.

Errors from the accelerometers are double integrated to obtain a position, so even small offsets (biases) will over time severely degrade the solution. Similar angular rates are integrated to obtain an updated attitude of the platform. Attitude-errors would also lead the specific force-transformation to be misaligned with the gravity model, which also seriously would affect the velocity and position estimates. As an INS is a dead-reckoning system, errors in the initialization of the system can also introduce significant degradation of the final solution. The final source of errors is caused by approximations in the INS mechanizations equations.

4.3.1 IMU Errors

All IMUs exhibit errors affecting the gyro and accelerometer measurements. The primary error sources are biases, scale-factor errors and noise. The errors are generally stochastic in nature, but some parts are deterministic. A general error-model for accelerometers and gyroscopes can be defined as [Gro13]:

$$\tilde{\mathbf{f}}_{ib}^{b} = \mathbf{b}_a + (\mathbf{I}_3 + \mathbf{M}_a)\mathbf{f}_{ib}^{b} + \mathbf{w}_a \tag{4.10}$$

where $\tilde{\mathbf{f}}_{ib}^{b}$ is the measured specific force, $\mathbf{b}_a$ is a bias offset, $\mathbf{M}_a$ is matrix which describe scale-factor and cross-coupling errors, $\mathbf{w}_a$ is sensor noise. $\mathbf{f}_{ib}^{b}$ describes the true specific force experienced by the body.

In a similar fashion, the output from the gyroscopes can be modeled as:

$$\tilde{\boldsymbol{\omega}}_{ib}^{b} = \mathbf{b}_g + (\mathbf{I}_3 + \mathbf{M}_g)\boldsymbol{\omega}_{ib}^{b} + \mathbf{G}_g\mathbf{f}_{ib}^{b} + \mathbf{w}_b \tag{4.11}$$

where $\tilde{\boldsymbol{\omega}}_{ib}^{b}$ is the measured angular-rate, $\mathbf{b}_g$, is a bias-offset, $\mathbf{M}_g$ is a matrix to define scale- and cross-coupling errors, $\mathbf{G}_g$ denotes acceleration dependent errors, $\mathbf{w}_b$ is sensor noise and $\boldsymbol{\omega}_{ib}^{b}$ denotes the true encountered angular-rate.

The bias terms in eq. (4.10-4.11) can general be seen as:

$$\mathbf{b}_a = \mathbf{b}_{as} + \mathbf{b}_{ad} \tag{4.12}$$

where $\mathbf{b}_{as}$ is a static term, also known as the turn-on bias or bias repeatability. The term is constant over the operating time, but varies from each time the IMU is turned on. $\mathbf{b}_{ad}$ is a dynamic term, which changes over the operating period. This term is often known as the in-run bias variability or bias instability.

The magnitude of errors greatly determines the overall accuracy which can be obtained with an INS. Errors such as the cross-coupling terms are due to misalignment in the sensor triad and can often be calibrated during a lab-test.

Higher quality IMU's are often calibrated from the factory, so that parts of the errors, which otherwise would degrade the INS solution are mitigated. Most IMUs also exhibit temperature dependent errors and would hence need to be calibrated at a specific operating temperature.

The magnitude of the errors in eq. (4.10-4.11) further depends on the grade of IMU being used. Here, we consider three grades of IMUs, Navigation-grade, Tactical-grade and Automotive, see Table 4.1.

Sensor Error	**Navigation**	**Tactical**	**Automotive**
Gyro Bias $^\circ/h$	0.005-0.010	0.1-10	100+
Gyro Noise $^\circ/h/\sqrt{Hz}$	0.002-0.005	0.2-0.5	-
Acc. Bias mm/s^2	0.05-0.10	2-4	12+
Acc. Noise $mm/s^2/\sqrt{Hz}$	0.05-0.10	2-4	-

Table 4.1: Classification of IMUs [Pet03]

The classification seems not to be universally agreed upon, [Gro13] defines classification in terms of marine, aviation, intermediate, tactical and consumer. The above table also merely serves as an example of expected sensor performance. In addition to the above classification, IMUs are often distinguished by the sensor technology. In general, three types dominates the market; Ring-Laser Gyros (RLG), Fiber-Optic Gyros (FOG) and Microelectromechanical systems (MEMS). The weight and performance of each type differs, where RLGs typically is used for high-end units, but normally heavy and large. MEMS IMUs on the other end, can be manufactured compactly but generally at the expense of performance.

No matter the quality of the IMU, accumulated errors will always build up and degrade the navigation solution. A higher-grade IMU however would give a better performance for a longer duration than that of a lower-grade IMU. It is a common strategy to estimate IMU errors during a mission. Techniques known as zero velocity update (ZVU) and zero angular rate update (ZARU) can be employed, if the user has knowledge about when the system is completely stationary, i.e. not moving. Another approach is to estimate the in-run biases with an auxilary sensor, such as GNSS. The next chapter describes various integration architectures for combined INS and GNSS navigation.

Chapter 5

GNSS/INS Integration

INS generally provides a good short-term position accuracy but can quickly drift from the true position, due to IMU biases. The INS stability, is obviously dependent of the grade of the IMU being used. GNSS on the other hand provides excellent long-term position accuracy, but is dependent on having a unobstructed view to the SVs for optimum performance. The short term stability of GNSS, can be heavily influenced by phenomenas, such as multipath and NLOS reception.

INS and GNSS are widely regarded as complimentary systems and thus is often combined in order to exploit the high update-rate and short-term accuracy of the INS and the long-time stability of GNSS. For some applications, a low grade INS is integrated with GNSS solely for providing an attitude solution.

There are in general 3 levels of GNSS/INS integration; Loosely coupling (L.C.), tightly coupling (T.C.) and ultra-tightly (or deep) coupling (U.T.C.). In Figure 5.1 the difference between the three architectures is shown. For all three cases a closed loop correction of estimated IMU biases are returned to the INS processing. The Navigation Filter (or integration filter) is different between the implementations, but usually implemented as a KF or EKF. For a L.C. approach, the GNSS PVT solution is feeded to the navigation filter to estimate current IMU biases and calculate a position, velocity and attitude solution on basis of both the INS and GNSS. The downside to this strategy, is that at least

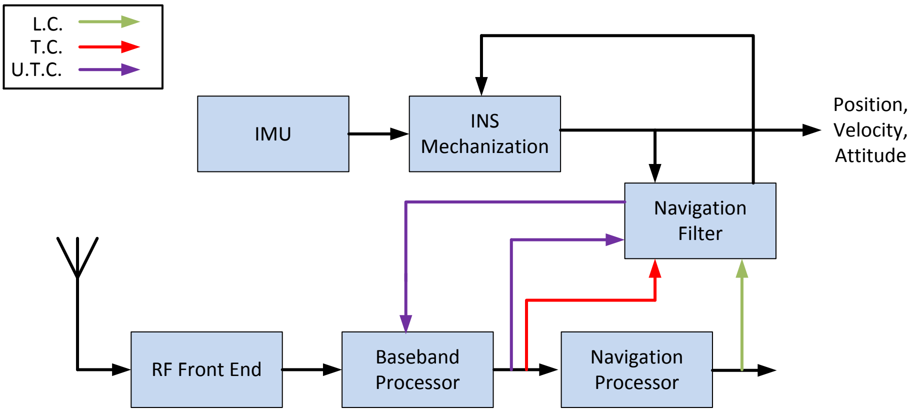


Figure 5.1: Flowchart of loosely-coupled (L.C.), tightly-coupled (T.C.) and ultra-tightly coupled (U.T.C.) GNSS/INS integration.

4 SVs needs to be tracked by the GNSS receiver to produce an input for the navigation filter. In a T.C. solution, the GNSS receiver input pseudoranges and pseudo-range rates to the filter. This strategy is more flexible as, the GNSS receiver would continue to provide information to the navigation filter with as little as one tracked SV. The U.T.C. configuration provides the same input to the navigation filter as the T.C. configuration, but in addition predictions of Doppler and code-phase is fed-back to the baseband processor in the GNSS receiver. An U.T.C. solution is closely related to that of a GNSS vector receiver, as noise is reduced in all channels by using the combined position and velocity solution to drive the tracking loops for each SV. The robustness of the U.T.C. architecture is however better than for a vector receiver, because the INS provides additional measurement redundancy. A U.T.C. implementation normally requires a software receiver, since commercial receivers does not provide a direct interface to the NCOs of the individual tracking loops. In the following, more detailled descriptions of each integration architecture is provided.

In the next section, a description of the INS error state propagation is described, which comprise the prediction or transition model of all the implementation variants. The remaining chapter will only focus on earth-frame (ECEF) implementations. L.C. and T.C. integration methods are well covered in the literature, so this chapter will only serve as a brief overview, for more comprehensive treatments refer to, e.g., [Gro13], [TW04], [NKG13], [GWA07].

5.1 INS State Propagation

A dynamic model of the INS error propogation as well as sensor errors of both systems are used by the filter to estimate biases for the accelerometer and gyroscopes. The KF can be implemented as either a full-state or error-state filter. The difference is that for a full-state implementation the raw IMU inputs are feeded directly into the filter and this needs to implement the INS mechanizations in the prediction model of the KF. For an error-state implementation the INS mechanization is performed seperately and the filter only needs to be run during GNSS measurement updates. There is usually negligible difference in performance between the two strategies, but processor load and algorithm complexity may vary. Herein, only the error-state implementation is considered.

The INS error states are defined as, [Gro13]:

$$\mathbf{x}^e_{INS} = \begin{pmatrix} \delta\boldsymbol{\psi}^e_{eb} \\ \delta\mathbf{v}^e_{eb} \\ \delta\mathbf{r}^e_{eb} \\ \mathbf{b}_a \\ \mathbf{b}_g \end{pmatrix} \tag{5.1}$$

where $\delta\boldsymbol{\psi}^e_{eb}$ is the estimated attitude error, $\delta\mathbf{v}^e_{eb}$ is the estimated velocity error, $\delta\mathbf{r}^e_{eb}$ is estimated position error, $\mathbf{b}_a$ is estimated accelerometer biases and $\mathbf{b}_g$ is estimated gyroscope biases.

The INS state-transition matrix, is defined as, [Gro13]:

$$\boldsymbol{\Phi}^e_{INS} \approx \begin{bmatrix} \mathbf{I}_3 - \boldsymbol{\Omega}^e_{ie}\tau_s & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \hat{\mathbf{C}}^e_b\tau_s \\ \mathbf{F}^e_{21}\tau_s & \mathbf{I}_3 - 2\boldsymbol{\Omega}^e_{ie}\tau_s & \mathbf{F}^e_{23}\tau_s & \hat{\mathbf{C}}^e_b\tau_s & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3\tau_s & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{5.2}$$

where $\hat{\mathbf{C}}^e_b$ is the coordinate transformation matrix (body to ECEF) computed from the INS mechanizations. The hat-accent is used to denote, that the quantity has been corrected from a previous KF error-estimate. τ_s denotes the time between GNSS updates. $\mathbf{F}^e_{21}$ describes the coupling of attitude error to velocity errors. $\mathbf{F}^e_{23}$ describes the sensitivity of the gravitational model with respect to velocity errors.

$$\mathbf{F}_{21}^{e} = \left[- \left(\hat{\mathbf{C}}_{b}^{e} \hat{\mathbf{f}}_{ib}^{b} \right) \wedge \right] \tag{5.3}$$

where $\wedge$ indicates the skew-symmetric form of $\hat{\mathbf{C}}_{b}^{e} \hat{\mathbf{f}}_{ib}^{b}$. $\mathbf{F}_{23}^{e}$ describes the sensitivity of the gravitational model with respect to velocity errors.

$$\mathbf{F}_{21}^{e} = \mathbf{g}_{b}^{e}(\delta \mathbf{r}_{b}^{e}) \tag{5.4}$$

The process noise covariance can be stated as, [Gro13]:

$$Q_{INS}^{e} = \begin{bmatrix} S_{rg}\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & S_{ra}\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & S_{bad}\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & S_{bgd}\mathbf{I}_3 \end{bmatrix} \tau_s \tag{5.5}$$

where S_{rg} is PSD of the gyro noise, S_{ra} is the PSD of the accelerometer noise, S_{bad} is the PSD of the accelerometer in-run bias variation and S_{bgd} is the PSD of the gyro in-run bias variation.

5.2 Loosely-Coupled Integration

In a L.C. implementation the position and velocity measurements of both the GNSS and the INS system is combined in a single KF. The KF algorithm is typically executed when a new GNSS measurement becomes available. An illustration of the L.C. architecture is depicted in Figure 5.2.

The measurement innovation, $\delta \mathbf{z}$, is formed as the position and velocity differences between the INS and GNSS and is forwarded to the KF.

The measurement vector for the KF is, [Gro13]:

$$\delta \mathbf{z} = \begin{bmatrix} \mathbf{r}_{GNSS}^{e} - \mathbf{r}_{INS}^{e} - \hat{\mathbf{C}}_{b}^{e} \mathbf{l}_{ba}^{b} \\ \mathbf{v}_{GNSS}^{e} - \mathbf{v}_{INS}^{e} + \mathbf{\Omega}_{ie}^{e} \hat{\mathbf{C}}_{b}^{e} \mathbf{l}_{ba}^{b} \end{bmatrix} \tag{5.6}$$

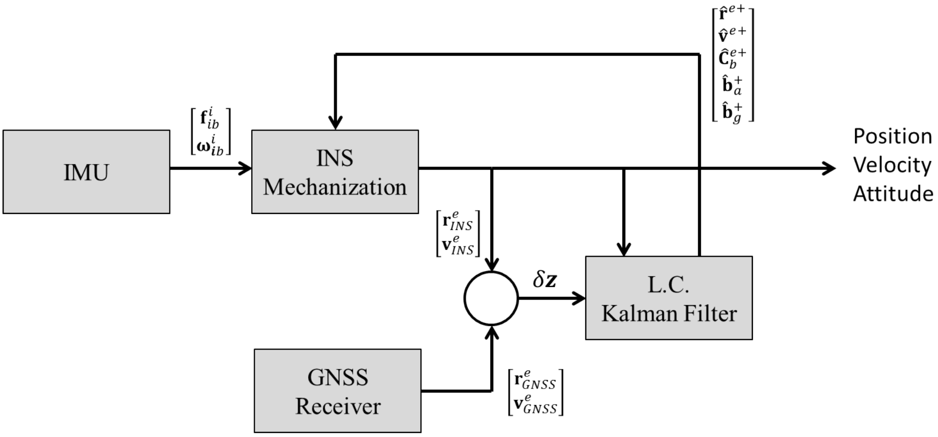

IMU
$\begin{bmatrix}\mathbf{f}^i_{ib}\\ \boldsymbol{\omega}^i_{ib}\end{bmatrix}$
INS Mechanization
$\begin{bmatrix}\hat{\mathbf{f}}^{e+}\\ \hat{\mathbf{v}}^{e+}\\ \hat{\mathbf{C}}^{e+}_b\\ \hat{\mathbf{b}}^+_a\\ \hat{\mathbf{b}}^+_g\end{bmatrix}$
Position
Velocity
Attitude
$\begin{bmatrix}\mathbf{r}^e_{INS}\\ \mathbf{v}^e_{INS}\end{bmatrix}$
$\delta\mathbf{z}$
L.C. Kalman Filter
GNSS Receiver
$\begin{bmatrix}\mathbf{r}^e_{GNSS}\\ \mathbf{v}^e_{GNSS}\end{bmatrix}$

Figure 5.2: Architecture of L.C. GNSS/INS integration with closed-loop corrections.

where $\mathbf{l}^a_{ba}$ is the lever-arm between the phase-center of GNSS antenna and origin of the IMU sensor frame. This quantity can for most applications be measured prior to operation. The importance of including this term is described in [Pet03].

The associated measurement error matrix, $\mathbf{R}$, has been determined as:

$$\mathbf{R} = \begin{bmatrix} \sigma^2_{xyz}\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma^2_{v,xyz}\mathbf{I}_3 \end{bmatrix} \tag{5.7}$$

where σ^2_{xyz} is the (single-dimension) variance of the GNSS position provided by the receiver. $\sigma^2_{v,xyz}$ is the variance of the GNSS velocity. Note, that if the receiver is able to calculate the position- and velocity uncertainty for separate dimensions, these values should be used instead.

The observation matrix, $\mathbf{H}$ is given as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{5.8}$$

It should be noted that, the L.C. filter is an ordinary KF as no linearization steps has been involved. The KF recursions are performed as described in section 3.3.5. It should although be noted, that the filter resets the error state estimates after the data-update due to closed-loop corrections. This essentially means that the prediction of the error-states becomes zero. The superscript, $^+$, in the corrections from Figure 5.2 indicates that the estimates are computed after the data-update. The state-uncertainty matrix or error-covariance, $\mathbf{P}$, should be initialized based on expected uncertainties in position, velocity and attitude. Furthermore, the run-to-run bias variations of the accelerometers and gyroscopes should also be accounted for.

5.3 Tightly-Coupled Integration

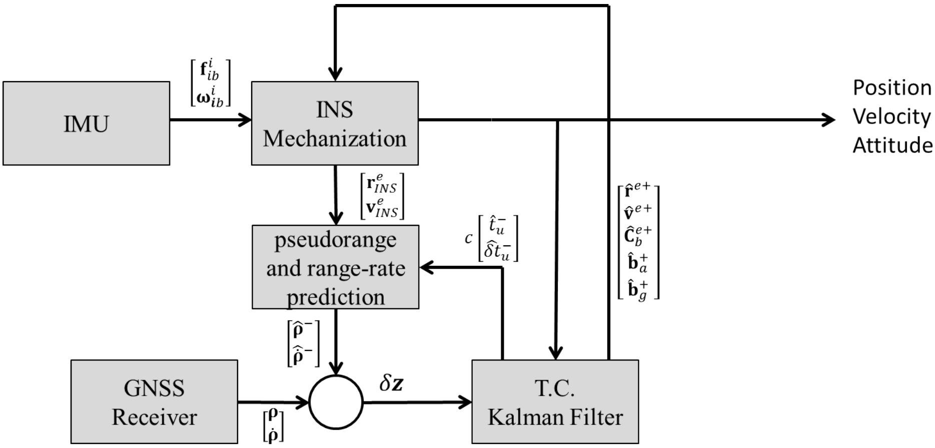

IMU
$\begin{bmatrix} \mathbf{f}_{ib}^{i} \\ \boldsymbol{\omega}_{ib}^{i} \end{bmatrix}$
INS Mechanization
Position
Velocity
Attitude
$\begin{bmatrix} \mathbf{r}_{INS}^{e} \\ \mathbf{v}_{INS}^{e} \end{bmatrix}$
$\begin{bmatrix} \hat{\mathbf{r}}^{e+} \\ \hat{\mathbf{v}}^{e+} \\ \hat{\mathbf{C}}_{b}^{e+} \\ \hat{\mathbf{b}}_{a}^{+} \\ \hat{\mathbf{b}}_{g}^{+} \end{bmatrix}$
$c\begin{bmatrix} \hat{t}_u^- \\ \widehat{\delta t}_u^- \end{bmatrix}$
pseudorange and range-rate prediction
$\begin{bmatrix} \hat{\boldsymbol{\rho}}^- \\ \hat{\dot{\boldsymbol{\rho}}}^- \end{bmatrix}$
GNSS Receiver
$\begin{bmatrix} \boldsymbol{\rho} \\ \dot{\boldsymbol{\rho}} \end{bmatrix}$
$\delta\mathbf{z}$
T.C. Kalman Filter

Figure 5.3: Architecture of T.C. GNSS/INS integration with closed-loop correction.

In a T.C. implementation, the transition model for the integration filter is largely the same as for the L.C. architecture, however since this architecture solves GNSS user position from pseudoranges and pseudorange-rates, two additional states are augmeted to the state-vector to model clock-bias and clock-drift. The state vector takes the form:

$$\mathbf{x}^e = \begin{bmatrix} \mathbf{x}_{INS}^e \\ \mathbf{x}_{GNSS}^e \end{bmatrix}, \ \mathbf{x}_{GNSS}^e = \begin{bmatrix} c\delta t_u \\ c\delta \dot{t}_u \end{bmatrix} \tag{5.9}$$

Similarly, the transition matrix is augmented as:

$$\boldsymbol{\Phi}^e = \begin{bmatrix} \boldsymbol{\Phi}_{INS}^e & \mathbf{0}_2 \\ \mathbf{0}_2 & \boldsymbol{\Phi}_{GNSS}^e \end{bmatrix}, \ \boldsymbol{\Phi}_{GNSS}^e = \begin{bmatrix} 1 & \tau_s \\ 0 & 1 \end{bmatrix} \tag{5.10}$$

and finally, the process noise covariance, $\mathbf{Q}$ is augmented as:

$$\mathbf{Q}^e = \begin{bmatrix} \mathbf{Q}^e_{INS} & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{Q}^e_{GNSS} \end{bmatrix}, \; \mathbf{Q}^e_{GNSS} = \begin{bmatrix} S_f\tau_s + \frac{1}{3}S_g\tau_s^3 & \frac{1}{2}S_g\tau_s^2 \\ \frac{1}{2}S_g\tau_s^2 & \frac{1}{2}S_g\tau_s \end{bmatrix} \tag{5.11}$$

where the PSDs, S_f and S_g, can be calculated from the oscillator's Allan variance parameters as, [BH97],

$$S_f \approx h_0/2 \cdot c^2 \;\; S_g \approx 2\pi^2 h_{-2} \cdot c^2$$

The measurement vector of the T.C. filter is defined as:

$$\delta\mathbf{z} = \begin{bmatrix} \mathbf{z}_{pr} \\ \mathbf{z}_{prr} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\rho} - \hat{\boldsymbol{\rho}}^- \\ \dot{\boldsymbol{\rho}} - \hat{\dot{\boldsymbol{\rho}}}^- \end{bmatrix} \tag{5.12}$$

where $\boldsymbol{\rho}$ denotes a vector of measured pseudoranges from the GNSS receiver, $\dot{\boldsymbol{\rho}}$ denotes the vector of measured pseudorange-rates. $\hat{\boldsymbol{\rho}}^-$,$\hat{\dot{\boldsymbol{\rho}}}^-$ denotes predicted values from the INS solution and the predicted KF estimates of clock bias and drift, hence is given for the k-th SV as:

$$\hat{\rho}_k^- = \sqrt{(X_k^e - X_{INS}^e)^2 + (Y_k^e - Y_{INS}^e)^2 + (Z_k^e - Z_{INS}^e)^2} + c\hat{\delta t}_u \tag{5.13}$$

$$\hat{\dot{\rho}}_k^- = \mathbf{u}_k^T \left[\mathbf{v}_k^e - \mathbf{v}_{INS}^e\right] + c\hat{\delta t}_u \tag{5.14}$$

with $\mathbf{u}_k^T$ is the LOS unit-vector from receiver to SV, defined as:

$$\mathbf{u}_k = \frac{\mathbf{r}_k^e - \mathbf{r}_{INS}^e}{|\mathbf{r}_k^e - \mathbf{r}_{INS}^e|}$$

It should be noted, that the GNSS pseudoranges in eq. (5.12) is implicitly assumed to have been corrected for atmospheric errors.

Finally, the observation matrix can be found as the partial derivatives of

$$\mathbf{H} = \begin{bmatrix} \frac{\partial\mathbf{z}_{pr}}{\partial\delta\boldsymbol{\Psi}^e_{eb}} & \frac{\partial\mathbf{z}_{pr}}{\partial\delta\mathbf{v}^e_{eb}} & \frac{\partial\mathbf{z}_{pr}}{\partial\delta\mathbf{r}^e_{eb}} & \frac{\partial\mathbf{z}_{pr}}{\partial\mathbf{b}_a} & \frac{\partial\mathbf{z}_{pr}}{\partial\mathbf{b}_g} & \frac{\partial\mathbf{z}_{pr}}{\partial ct_u} & \frac{\partial\mathbf{z}_{pr}}{\partial c\delta t_u} \\ \frac{\partial\mathbf{z}_{prr}}{\partial\delta\boldsymbol{\Psi}^e_{eb}} & \frac{\partial\mathbf{z}_{prr}}{\partial\delta\mathbf{v}^e_{eb}} & \frac{\partial\mathbf{z}_{prr}}{\partial\delta\mathbf{r}^e_{eb}} & \frac{\partial\mathbf{z}_{prr}}{\partial\mathbf{b}_a} & \frac{\partial\mathbf{z}_{prr}}{\partial\mathbf{b}_g} & \frac{\partial\mathbf{z}_{prr}}{\partial ct_u} & \frac{\partial\mathbf{z}_{prr}}{\partial c\delta t_u} \end{bmatrix} \tag{5.15}$$

which can be approximated as:

$$\mathbf{H} = \left[\begin{array}{ccccccc} \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{u}_1^T & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 1 & 0 \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{u}_2^T & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{u}_n^T & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 1 & 0 \\ \hdashline \mathbf{0}_{1,3} & \mathbf{u}_1^T & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 0 & 1 \\ \mathbf{0}_{1,3} & \mathbf{u}_2^T & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{1,3} & \mathbf{u}_n^T & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & \mathbf{0}_{1,3} & 0 & 1 \end{array}\right] \tag{5.16}$$

The approximation does not feature a coupling between attitude error, velocity and gyro errors to the pseudorange and pseudorange-rates, since this influence is considered neglible [Gro13].

The final part of the T.C. model is the measurement noise covariance matrix, **R**, which describes the uncertainty of the GNSS pseudorange and pseudorange-rate measurements. The errors in measurements are assumed to be uncorrelated. The filter implementation is an EKF, since linearization is required.

5.4 Ultra-Tightly Coupled Integration

As mentioned, the U.T.C. implementation is closely related to that of a vector receiver. In fact, it can be seen as a fusion between a T.C. GNSS/INS integration and a vector receiver. For that reason, the U.T.C. receiver implementation options (excluding the navigation filter), is the same as described in section 3.6.

U.T.C. GNSS/INS receivers are although superior, since they features the added benefit of having an integrated INS to constrain the position solution. A vector-receiver needs to track the LOS dynamics for each SV, whereas for U.T.C. systems only the residual errors from the INS solution needs to be tracked.

A figure of the U.T.C. implementation used in this work is illustrated in Figure 5.4.

It can be noted, that the figure closely resembles the block-diagram of the vector receiver in Figure 3.24. In fact, the only difference is the inclusion of the INS. For the vector receiver, the code-phase and Doppler predictions was obtained from propagating the (GNSS) position and velocity forward in time. For an

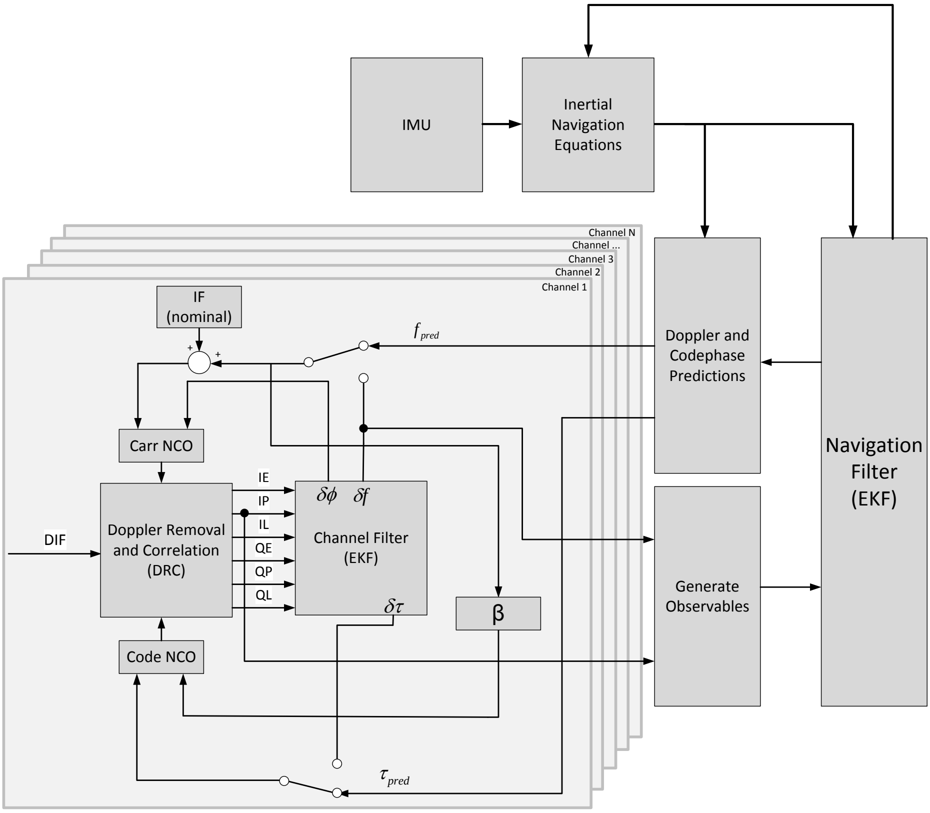


Figure 5.4: Flowchart of U.T.C. GNSS/INS integration.

U.T.C. implementation, this is not necessary as the INS solution can be used instead prior to generating the observables for the T.C. filter. The clock-bias and drift however still need to be propagated from the prediction model of the T.C. filter.

Chapter 6

Experimental Set-Up

This chapter describes the hardware and software developed during the study. This includes a UAV navigation payload, comprising of a purpose built GNSS IF recorder, a MEMS-based IMU and a reference GNSS receiver. Afterwards, the description of an implemented post-processing U.T.C. GNSS/INS receiver is given.

6.1 GPS/GLONASS IF Recorder

The system presented in this section, was originally published in [OJK16]. A brief summary of the design is however provided here for completeness.

As stated in the introduction, commercial systems for GNSS IF data recording do exists, but none of the available products have been found suited for UAV testing primarily due to size and weight constraints. Commercial products are often constructed as streaming devices, which downmix, filters and samples IF data and provide a USB data stream to a PC for storage. Although this approach is flexible and more than adequate for a large number of applications, it is not an ideal solution for applications with severe weight restrictions.

To overcome this issue, a system was designed based on a small single-board computer (SBC) and two commercial GNSS RF front-ends. The used SBC is the BeagleBone Black, which is based on a 1 GHz ARM Cortex A8 processor. The SBC further includes two 200 MHz coprocessors, which have been fundamental in this design. The RF front-ends are the MAX2769 [Max10] from Maxim Integrated. A functional diagram of the MAX2769 front-end is visualized in Figure 6.1.

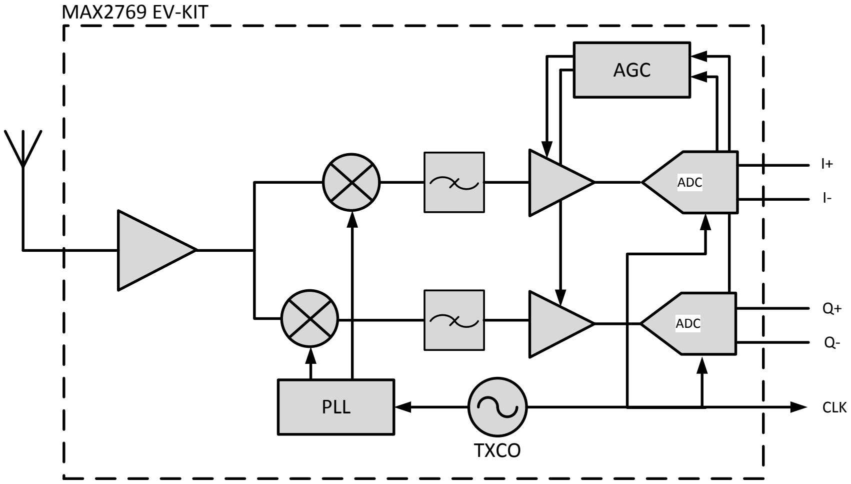


Figure 6.1: Blockdiagram of MAX2769.

The MAX2769 is an integrated circuit, which includes a complete RF processing chain and can be configured for either GPS L1, GLONASS L1 and Galileo E1 reception.

The RF front-end features a programmable single-conversion stage and supports both the use of active and passive antennas. The RF front-end is available as an evaluation kit (EV-kit) equipped with a number of SMA connectors for evaluation of separate stages of the front-end circuitry. This kit also comes with a 16.368 MHz Temperature Compensated Crystal Oscillator (TCXO) and in addition features an SMA input for use with an external oscillator, hence it is possible to evaluate the system using different grades of reference oscillators. The antenna input is first amplified though a Low Noise Amplifier (LNA). The received signal then propagates through a quadrature mixer, where the RF signal is converted to IF. The PLL Synthesizer for the mixing-stage can be programmed to a LO frequency in the range 1550-1610 MHz with approximately 40 Hz separations. After the mixing process, the IF signal passes through a configurable filter and a Programmable Gain Amplifier (PGA) before AD conversion. The RF front-end is also equipped with an AGC circuit, which adjusts the PGAs for ideal saturation of the ADC. The ADC has a configurable sample rate of up to 50 MSamples/s and has the ability to quantize the signal with up to 3 bits precision for real samples and (2+2) bit precision for I/Q sampling. The configuration of the mixer, filters etc. is done using a Serial Peripheral Interface (SPI).

The use of two front-ends has been necessary to capture IF data for GPS L1 C/A code and GLONASS L1 C/A code simultaneously. The front-end for GPS reception has been configured to an IF frequency of 4.092 MHz, the subsequent filter was configured as a band-pass filter with a center-frequency equal to the IF frequency and a 2.5 MHz bandwidth. As GLONASS use frequency-division (FDMA) to distinguish SVs and is transmitting on 14 frequency slots within the range between 1597.5515 - 1605.886 MHz, a wide bandwith configuration has been facilitated. The LO was programmed to a frequency of 1600.995 MHz, and a lowpass filter with a dual sided bandwidth of 18 MHz was employed. For both front-ends, the onboard 16.368 MHz TXCOs have been used as the clock reference for the frequency synthesizer and the ADC. For the GPS front-end, only 2-bits of the I- output of the ADC was used. As the front-end for GLONASS, have spectral components in both the positive and negative sidebands, 2 bits from each of the I and Q outputs has been connected to the SBC. In addition to the ADC data outputs, the sample clock from the ADCs has also been provided to the SBC.

A block-diagram of the SBC and interface connections from the front-ends are shown in Figure 6.2.

The SBC is based on Texas Instruments Sitara AM335x Processor [TIS13]. This features an ARM Cortex-A8 CPU running with a clock frequency of 1GHz. In addition the processor is also equipped with two coprocessors, here termed as Programmable Real-time Units (PRU). This module consist of two 32-bit CPUs which runs with clock-frequencies of 200 MHz. Each PRU core has an allocation of 8 Kb of Instruction Memory and 8 Kb Data memory. In addition the PRU cores also shares 12 Kb of memory. The PRUs features an interrupt controller which is directly connected to the interrupt controller of the hostsystem (ARM). The PRUs are especially useful in timing critical applications, as they operate independently from the linux operating system on the ARM processor. In our design we have utilized both PRU cores to transfer data from the two connected front-ends. As mentioned, each front-end provides a parallel interface providing the ADC data-outputs and the sampleclock, these inputs are connected to digital IOs of the SBC. In the SBC two 4 MByte circular buffers has been allocated in DDR memory, where the received data is temporarily stored. The PRUs fetch data from the IOs, according to the flowchart in Figure 6.3.

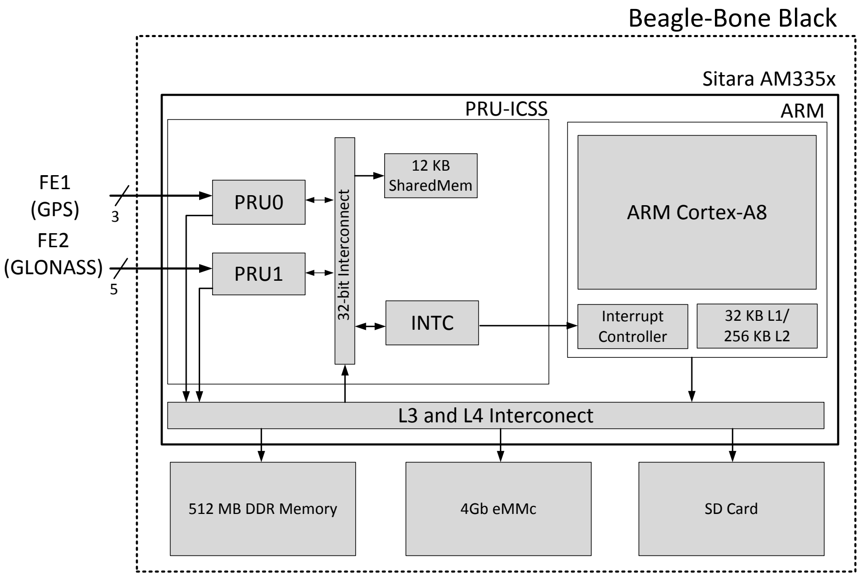


Figure 6.2: Blockdiagram of the BeagleBone Black (SBC) and interface to RF front-ends.

The program for the PRUs are written in assembly language. The PRUs essentially waits for a high clock edge of the ADC sampling clock. After an edge has been detected, the state of the ADC databits are read from the IO and moved in to a local buffer within the PRU data memory. Whenever 32-bits of data

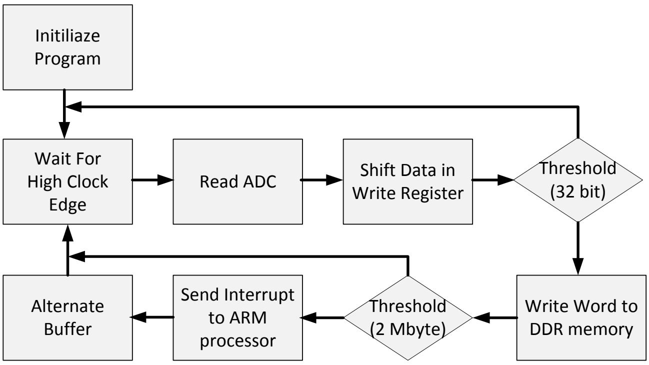


Figure 6.3: Flowchart of PRU operation.

has been received, the PRU initiates a transfer to the circular buffers residing in the DDR memory. This process continues until a threshold of 2 Mbytes has been received, corresponding to half of the buffer allocation. At this point, the PRU triggers an interrupt to the host processor (ARM), which then transfer the received data onto an SD card for permanent storage. Whilst the buffer is being emptied, the PRU are storing data in the second portion of the circular buffer to ensure data continuity.

Both connected front-ends have a sample clock-rate of 16.368 MSamples/s. From empirical tests, is was found that the maximum ADC clock frequency that could have been used is around 20 MHz. The main bottleneck is caused by the operation to alternate between buffers and generate interrupt in the PRU cores. In terms of storage, data from each front-end are stored in separate binary files. The GPS data file packs 4 (2-bits) samples into one byte, whereas for GLONASS a byte can only store 2 samples (2 bit I + 2 bit Q). The data throughput rate is approximately 4 MB/s for GPS and 8 MB/s for GLONASS. For a 32GB SD card this would be approximately 45 minutes of IF data.

A picture of the entire prototype is shown in Figure 6.4.

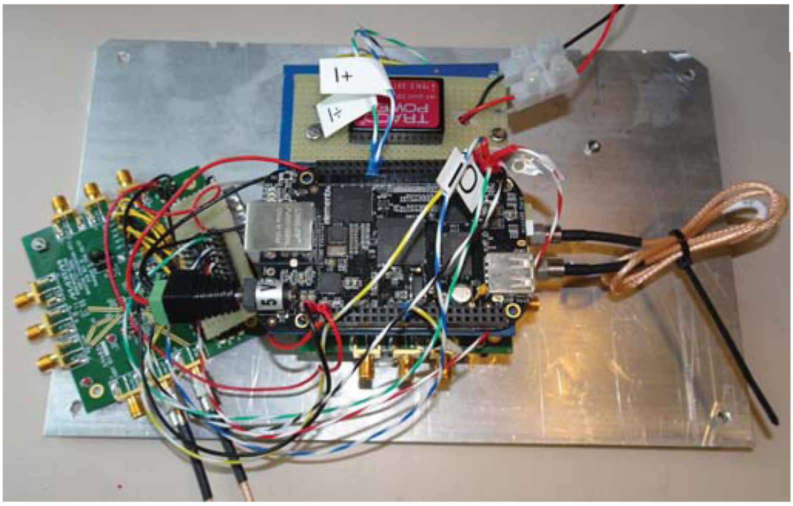
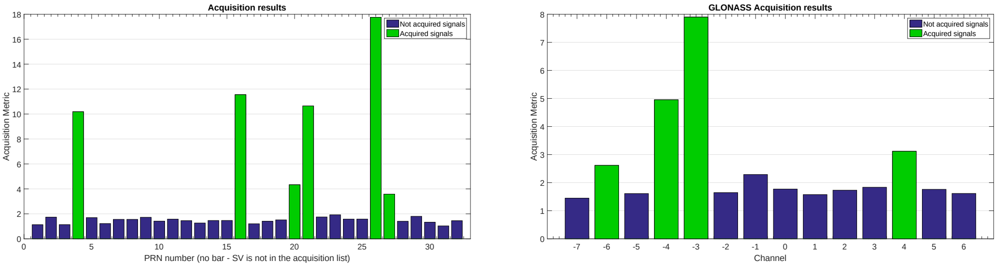


Figure 6.4: Picture of GNSS sampler prototype.

The picture shows the SBC, the two RF front-ends and a 5V DCDC converter used for powering the system. In order to validate the GNSS sampler, a static data-collection was performed the 3rd of July 2016. The GNSS sampler was connected to a roof-mounted choke ring antenna at the DTU Space Institute building. Acquisition results for GPS is shown in Figure 6.5a and for GLONASS in Figure 6.5b.



(a) GPS Acquisition. **(b) GLONASS Acquisition.**

Figure 6.5: GPS and GLONASS Acquisition.

From the figures, it can be seen that 6 GPS SVs and 4 GLONASS SVs was acquired. The software receiver used to process the IF data was a modified version of [BAB+07], which has been changed to support complex IF data and with modified algorithms to acquire and track GLONASS SVs.

The novelty of this design is that the IF recorder is built from commercial off-the-shelf components and hence only required a minimal effort in terms of hardware engineering. The design has been made possible due to the coprocessors embedded in the Sitara AM335x processor. This architecture furthermore has the additional benefit, that the ARM processor also are left with resources to perform other tasks.

6.2 Navigation Payload for UAV testing

The GNSS IF recorder from previous section has not been tested on a UAV in the described form. As the end goal has been to perform an U.T.C. GNSS/INS integration on a UAV, this section describe the design of the data-acquisition setup, which herein is referred to as the navigation payload. The setup consists of the GNSS IF recorder described in the previous section, although the RF front-end intended for GLONASS data collection was dropped in order to save weight and reduce complexity. The GPS RF front-end was furthermore reconfigured to use complex sampling with a nominal IF frequency of 0.42 MHz. In addition the filter was changed to a low-pass filter, with a dual-sided bandwidth of 8.4 MHz.

The navigation payload furthermore includes a MEMS-based IMU from Xsens and a survey-grade multi-frequency, multi-constellation GNSS receiver from Novatel for reference measurements. A high-level blockdiagram of the setup is shown in Figure 6.6.

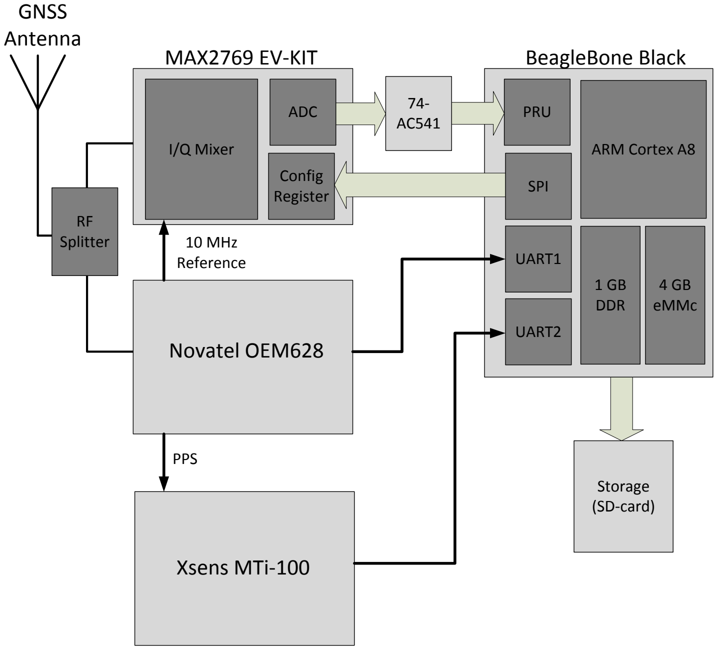


Figure 6.6: High-level blockdiagram of Navigation Payload.

In the designed system, the BeagleBone Black is used for interfacing the RF front-end, but in addition also provides an interface and storage for the IMU and the reference receiver. The clock-reference for the RF front-end has been taken from the reference receiver primarily for two reasons; The first reason is by sharing the same reference, the influence of oscillator behavior affects both receivers and hence anomalies in receiver measurements should be present in both systems. The second reason is that the embedded TXCO from the RF front-end has been found to be quite poor, which was especially profound with coherent integration times of 20 ms and while the system was experiencing movement. A final benefit in using the oscillator from the reference receiver, is that Allan variance parameters of the internal oscillator is well documented [Nov15].

In general for INS and GNSS integration, the time syncronization of measurements are important. In this case, the IMU has a Pulse-Per-Second (PPS) input, so that the internal timescale can be aligned with GPS Time (GPST).

The PPS is provided by the reference receiver. In terms of communication, the IMU and the Novatel receiver communicates with the BeagleBone Black using a universal asynchronous receiver-transmitter (UART) channel. In the SBC a multi-threaded program has been developed in C++, which stores the respective inputs from the IMU and reference receiver in separate ASCII files to the SD card.

The performance specifications of the IMU and Novatel reference receiver is briefly summarized below.

Xsens MTi-100

The IMU is an MEMS-based IMU. In addition to measurements of angular-rates and specific force, it also includes a 3-axis magnetometer and a barometer. The IMU can output calibrated measurements up to 400 Hz and raw sensor readings with up to 2 kHz. The IMU has the following specifications for gyroscopes and accelerometers [Xse15]:

Gyro Specifications	**Typical**	**Max**
Turn-on bias $[°/s]$	0.2	0.5
In-run bias $[°/h]$	10	-
Noise-density $[°/s/\sqrt{Hz}]$	0.01	0.015
g-sensitivity $[°/s/g]$	0.003	0.015

Table 6.1: MTi-100 Gyroscope specifications.

Acc. Specifications	**Typical**	**Max**
Turn-on bias $[m/s^2]$	0.03	0.05
In-run bias $[\mu g]$	40	-
Noise-density $[\mu g/\sqrt{Hz}]$	80	150

Table 6.2: MTi-100 Accelerometer specifications.

The IMU is delivered as being factory calibrated, which includes compensation of scale-factor, misalignment (cross-couling) and temperature induced errors.

Novatel OEM628

The reference receiver is a multi-frequency and multi-constellation receiver. The receiver features 120 tracking channels and have support for all current GNSS signals. The oscillator for this model is a Voltage-Controlled TXCO (VCTXCO), which has the following Allan variance parameters:

$\mathbf{h_0}$	$\mathbf{h_{-1}}$	$\mathbf{h_{-2}}$
$1.0 \cdot 10^{-21}$	$1.0 \cdot 10^{-20}$	$1.0 \cdot 10^{-20}$

Table 6.3: VCTXCO Allan variance parameters

The receiver normally incorporates clock-steering, i.e., it continuously varies the oscillator frequency to steer the clock-bias towards zero. This feature, has however been disabled as it caused problem when used as a reference clock for the RF front-end.

A picture of the assembled system is shown in Figure 6.7.

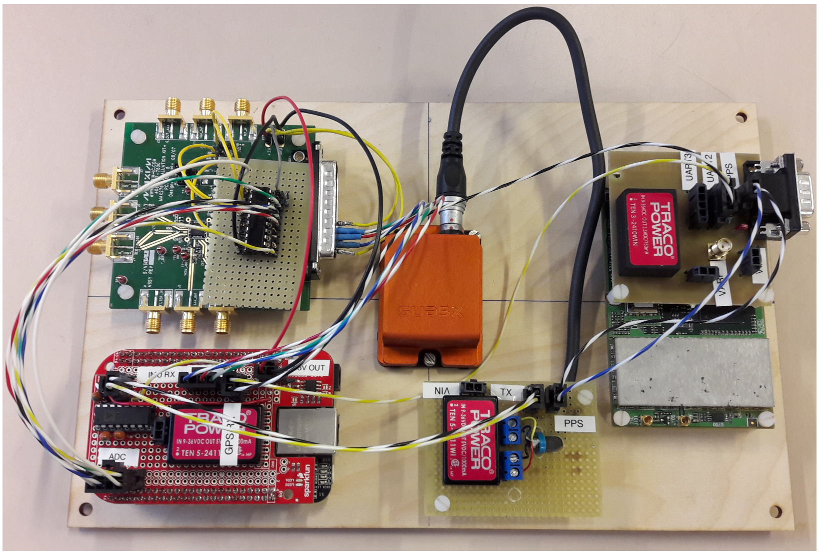


Figure 6.7: Picture of data-collection system mounted on UAV payload plate.

The IMU is placed in the center of the UAV payload plate in order to reference the inertial measurements, as closely to the center of mass of the platform. The Novatel receiver is placed on the right, the SBC in the bottom left and the MAX2769 RF front-end in the top left corner. The payload plate has the dimensions 17 x 27 cm and the total setup (excluding GNSS antenna and RF splitter) is 450 grams.

6.3 Post-Processing GNSS Software Receiver

There is a rising number of open-source GNSS software receivers implementations becoming available, perhaps most notably the GNSS-SDR developed by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC). This implementation is based on the GNU Radio framework and provides multi-frequency and multi-constellation tracking. Another notable implementation is the MATLAB-based GPS L1 C/A code receiver from [BAB+07], although this implementation is very simplistic it serves as an excellent starting point for experimenting with GNSS software receivers. In addition there also exist a few commercial options, e.g., SX3 from IFEN GmbH or University of Calgary's GSNRx receiver. The problem with using a commercial solution is however, that typically only binaries would be provided and it would be limited which modifications the user would be able make. Despite the impressive capabilities of the GNSS-SDR, it was although decided to initiate development of our own receiver based on MATLAB. There are several reasons for this; None of the open-source implementations provides functionality for, e.g., vector-tracking or U.T.C. GNSS/INS integration, so this would have to be developed in either case. Secondly, since these advanced concepts require additional interfaces between the navigation processing and tracking modules, it was considered to be easier to design the software architecture with this in mind from the outset, rather than trying to retrofit functionality on existing software.

From the earliest development phase, the receiver was inspired by the code of [BAB+07], but has now substantial differences in capabilities and architecture.

A high-level flowchart of the implemented receiver is shown in Figure 6.8.

The first operation of the receiver is to search for available SVs. This is accomplished in the acquisition module. After acquisition, the coarse estimates of code-phase and Doppler for the acquired SVs are then sent to the tracking module. The tracking module spawns a number of parallel worker threads, using the MATLAB Parallel Computing Toolbox [Mat17b]. Initially the receiver tracks the SVs for a predefined amount of time, to ensure a stable lock on all channels. Hereafter it transitions into an epoch-by-epoch tracking state, where the signals are tracked for a duration equal to the position update rate. In normal (scalar) operation, a PVT solution is calculated based on the tracking results. For vector- or U.T.C. processing, there is an additional step, where current PVT solution is used to predict Doppler and code-phase values for all SVs, which is then fed back to the tracking modules before processing the next block of IF samples.

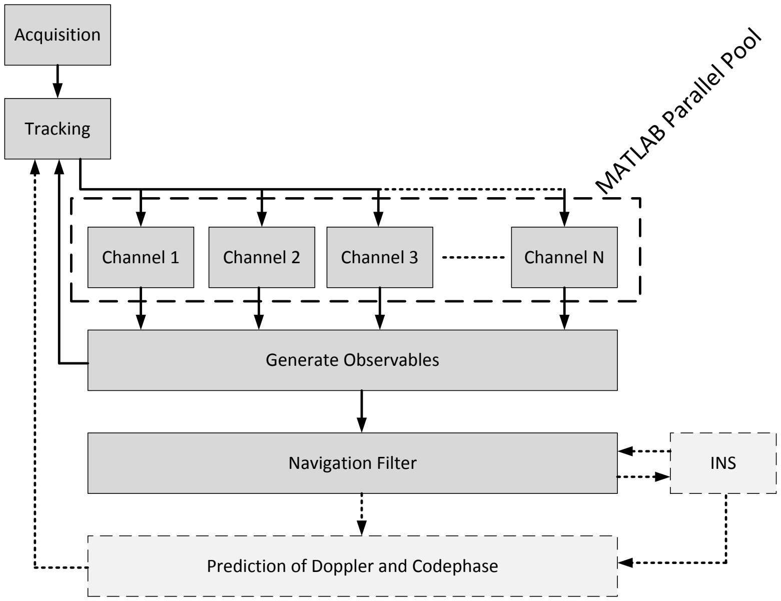


Figure 6.8: High-level flowchart of MATLAB software receiver. The dashed lines and blocks represent additional dataflow for U.T.C. GNSS/INS processing.

6.3.1 Acquisition

The acquisition module of the developed receiver is based on the parallel code-phase search algorithm as described in Section 3.2. The module is divided into two parts; A coarse estimation of Doppler and code-phase and later a fine-search procedure to further refine the estimates. The acquisition module uses largely the same code as described in [BAB$^+$07], although modifications has been made has been to support complex IF data from the RF front-end.

6.3.2 Tracking

The tracking for each channel is implemented as visualized in Figure 6.9.

The code- and carrier estimation block consist of a 5-state Finite State Machine, where the coherent integration time and loop bandwidths can be varied for each

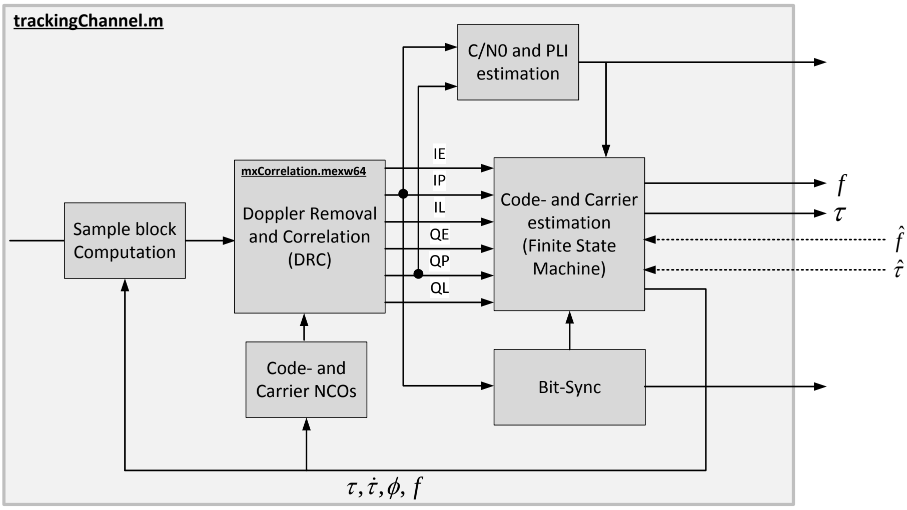


Figure 6.9: Tracking channel.

state. The initial 3 states, employs a traditional PLL/DLL architecture and the 4th state uses the EKF-based algorithm described in section 3.3.5. The 5th state is used for U.T.C./vector processing, where prediction of Doppler- and code-phase derived from the PVT solution is used for tracking. A bit-synchronization routine has also been implemented, which statistically detects whenever a new data-bit is transmitted. This functionality is especially important for coherent integration times up to 20 ms, which is the limit for GPS L1 C/A code tracking and in addition locations of the data-bit transitions are also required for the following generation of observables.

The Doppler removal and correlation block has been implemented as a MATLAB EXecutable (MEX) file [Mat17a]. The source code of this file has been written in C++ and compiled into a dynamical linked subroutine which can be called within the MATLAB scripting environment. Although outright computational efficiency is not a main priority for this receiver implementation, this measure does drastically improve processing speed.

6.3.3 Navigation Processing

In terms of obtaining a PVT solution, the filter implementation depends on which processing mode is used, i.e., scalar tracking, vector tracking or U.T.C.

For the scalar case, two algorithms has been implemented. The receiver can either determine the position and velocity using the LS approach desribed in sections (3.5.1-3.5.2). Optionally, the algorithm can use elevation dependent weighting on the observations, in which case a weighted LS estimate is computed. In addition the EKF navigation filter from section 3.5.3 can also be used for this mode.

For the vector receiver, the EKF navigation filter is employed. From the current PVT solution, the filter predicts user position, velocity, clock-bias and clock-drift forward in time using the prediction model of the EKF. Hereafter, the NCO commands for code- and carrier tracking are calculated as described in section 3.6.

6.3.3.1 Ultra-Tightly Coupled GNSS/INS Processing

The U.T.C. processing consists of three continuous processing steps; INS mechanizations, GNSS/INS EKF filter execution and Doppler and code-phase prediction for feedback to tracking NCOs.

The INS mechanizations is performed as described in section 4.2. The mechanizations are resolved in an ECEF frame and utilizes the most recent error and bias estimates from integration filter. Whenever new GNSS measurements becomes available, a 17-state EKF filter is executed to estimate errors in position, velocity and attitude. The filter also estimates accelerometer and gyroscope biases and lastly estimate clock-bias and clock-drift for the receiver. The navigation filter, is implemented as the T.C. variant described in section 5.3. The accompanying MATLAB-scripts provided from [Gro13] has been used as inspiration.

The final step is to predict the code-phase and Doppler for feedback to receiver tracking. This is largely accomplished as for the vector receiver, but only the clock-bias and clock-drift is here needed to be propagated forward in time. The position and velocity estimates, needed for the prediction is obtained from the INS mechanization.

INS initialization

As it was mentioned in Chapter 4, the INS needs to be initialized prior to operation. The position and velocity initialization is obtained from GNSS, but the attitude angles can not be initialized from GNSS. A levelling process is used for roll and pitch angle initialization from accelerometer measurements [Gro13],

$$\theta = \arctan\left(\frac{f^b_{ib,x}}{\sqrt{(f^b_{ib,y})^2 + (f^b_{ib,z})^2}}\right), \quad \phi = \arctan_2(-f^b_{ib,y},\ -f^b_{ib,z}) \qquad (6.1)$$

where θ represents the pitch angle and ϕ represent roll (in radians).

The heading (or yaw) angle, has in this study been initialized by an external measurement from a magnetic compass.

Chapter 7

Results and Analysis

In this chapter, a lab-test was conducted using a geodetic grade roof-mounted choke-ring antenna. The purpose of this test was to evaluate the developed software receiver in a controlled environment. For this test, an artificial jammer was used in order to asses tracking thresholds of the PLL/DLL based tracking, the EKF-based tracking algorithm and the vector-implementation. The U.T.C GNSS/INS solution is furthermore examined during two kinematic missions. In the first mission, the system was evaluated using a dataset acquired from a 4-wheeled trolley. The mission includes a passage below trees to asses the quality of the position solution in harsh surroundings. Finally, the U.T.C. solution is tested on a dataset acquired from a DJI S900 spreading wings hexacopter.

7.1 Comparison of PLL/DLL, EKF-based Tracking and Vector Receiver for Static Dataset

For initial verification and comparison of the implemented receivers, IF data was recorded using the navigation payload the 3rd of July, 2017. The used antenna was a Trimble TRM29659 choke-ring antenna, placed at the top of the DTU Space main building, at Elektrovej in Kgs. Lyngby, Denmark. The exact position of this antenna is known, and hence position errors from the three implemented receivers could be evaluated.

The recorded dataset is approximately 4 minutes long and was processed by the software receiver using a PLL/DLL, the EKF-based tracking algorithm and finally vector tracking. The position was determined using the LS approach described in section 3.5.1 and an elevation dependent weighting scheme. This approach was chosen here in favour of a EKF position solution, since the true (unfiltered) variability should be assessed. A total of 6 GPS SVs was acquired and tracked during the test.

For all the receivers, an early-late correlator spacing of 0.2 chips was used. The DLL was of second order and a bandwidth of 0.2 Hz was chosen. The PLL was of 3rd order with a bandwidth of 10 Hz. The parameters for the EKF-based tracking and the vector receiver is listed in Table 7.1.

Parameter	Value
w_A $(dB/s/\sqrt{(Hz)})$	0.5
$w_{\delta\tau}$ $(m/s/\sqrt{Hz})$	0.1
$w_{clk,b}$ $(cycles/s/\sqrt{Hz})$	$2\pi f_{L1} \cdot \sqrt{(h_0/2)}$
$w_{clk,d}$ $(Hz/s/\sqrt{Hz})$	$2\pi f_{L1} \cdot \sqrt{(2\pi^2 h_{-2})}$
w_a $(m/s^3/\sqrt{Hz})$	2
S_p $(m/s^2/\sqrt{Hz})$	0.1

Table 7.1: Processing parameters of EKF-based tracking and vector receiver.

where the Allan-variance parameters, h_0 ,h_{-2}, for the clock is given in Table 6.3. The S_p parameter is for the EKF navigation filter from section 3.5.3. The vector receiver is "tied" to this option, as the filter is needed to provide a predicted position, velocity and clock parameters as basis for the code-phase and Doppler predictions. The position is although still computed from a LS estimate, as the generated observables after vector receiver processing was recorded and subsequently used in a LS position estimation.

The errors from the true position of the antenna using the EKF-based tracking algorithm is plotted in Figure 7.1 in a local ENU frame. The mean errors in each dimension are between 2 and 4 meters. The standard deviations are approximately 1 meter for the horizontal dimensions and 2 meters vertically.

The mean errors was found to be similar for the three receivers, but the main difference was apparent in the obtained standard deviations. These values are summarized in Table 7.2.

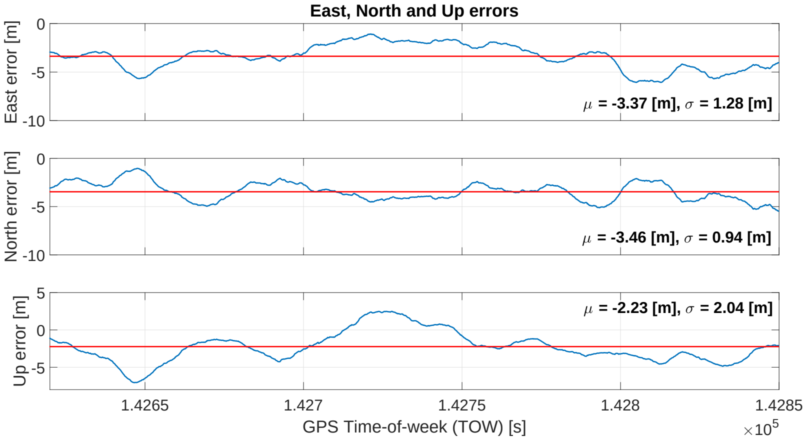


Figure 7.1: East, North and Up position errors from known antenna placement (EKF-based tracking).

Tracking Algorithm	σ_E $[m]$	σ_N $[m]$	σ_U $[m]$
PLL/DLL	2.03	2.01	4.44
EKF	1.28	0.94	2.04
Vector	1.26	0.99	2.38

Table 7.2: Position errors (1σ) from PLL/DLL, EKF-based tracking and vector tracking.

The EKF-based tracking obtains the least error variation. Secondly, the vector receiver and finally the traditional PLL/DLL option performed the worst. The difference between the EKF-based tracking and vector tracking are however minuscule.

The reason for using a vector receiver is often associated with weak signal tracking. In that regard, the original IF data was artificially corrupted with white noise being added to simulate the effect of a jammer. A test was conducted where the noise was gradually ramped ud to give a maximum attenuation of approximate 17 db-Hz of the original signal.

All three receivers could cope with this degradation without losing SVs. The mean position errors was largely unaffected, but the degradation of IF data increased the standard deviations of the position errors as seen from Table 7.3.

Tracking Algorithm	σ_E $[m]$	σ_N $[m]$	σ_U $[m]$
PLL/DLL	4.55	5.62	11.08
EKF	1.56	2.14	2.78
Vector	1.54	2.25	2.57

Table 7.3: Position errors (1σ) from PLL/DLL, EKF and vector tracking (Attenuated 17.5 db-Hz).

The PLL/DLL based tracking exhibit an error variation significantly larger than for the undistorted case. The standard deviations in each dimension has more than doubled for this receiver. The other two options has also been affected by the degraded signal but to a much more modest degree.

In order to asses the sensitivity limits of the PLL/DLL, EKF-based tracking and vector receiver, the IF data was again artificially degraded, but this time up to an attenuation of approx. 30 db-Hz. The result of this experiment in terms of C/N_0 is shown in Figure 7.2.

The PLL/DLL-based tracking is seen to lose lock on SV 15,16 and 21 during the jamming operation. This happens immediately after the jammer reaches maximum power. The threshold for tracking can be seen to be approximately 22.5 db-Hz for this receiver. The EKF-based tracking immediately lose SV 15, but keeps the remaining SVs in lock for longer. Around 120 seconds into the dataset, SV 16 is also lost. The remaining 4 SVs are maintained throughout the datarecord. The tracking threshold for the EKF-based tracking algorithm is approx. 20db-Hz. Finally, the vector receiver is seen to keep lock on all SVs during the entire mission.

As the vector receiver was the only option to sustain tracking on all SVs, the position error variations was calculated only for this. The results are summarized in Table 7.4.

Tracking Algorithm	σ_E $[m]$	σ_N $[m]$	σ_U $[m]$
Vector	2.00	2.16	5.58

Table 7.4: Position errors (1σ) from vector tracking (Attenuated 30 db-Hz).

Despite the excessive attenuation, the vector receiver obtains less error variance than what the PLL/DLL did under much more modest attenuation levels. It should however be emphasized that the navigation filter for the vector receiver effectively was configured for a (near) static scenario and that the initial position was calculated prior to the "jammed" reception.

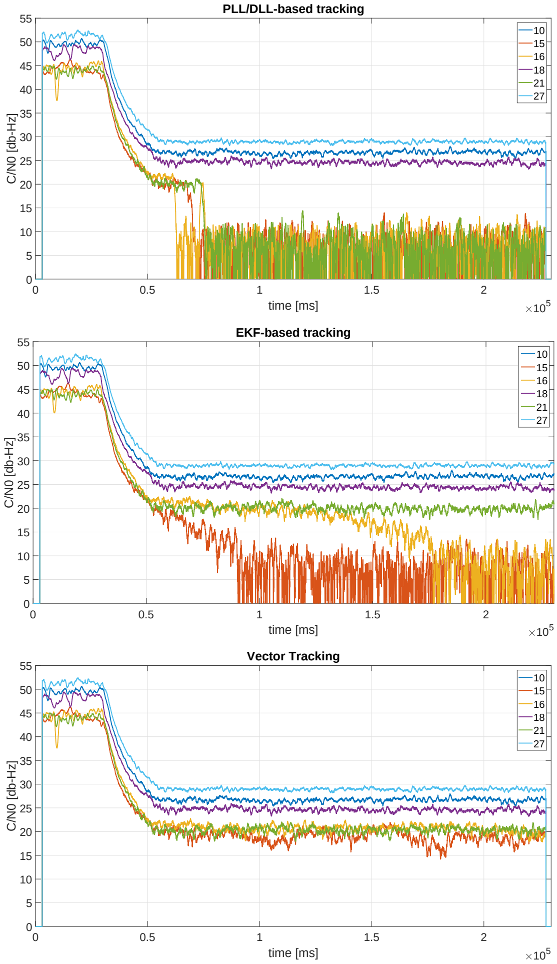
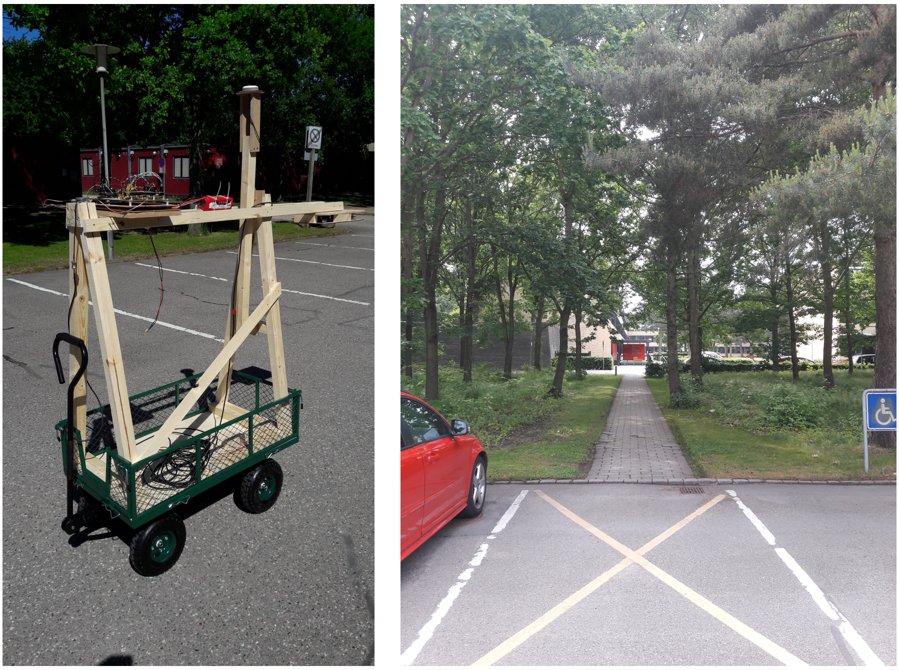


Figure 7.2: C/N_0 estimates from the PLL/DLL-based tracking (top), EKF-based tracking (middle) and Vector tracking (bottom) under ”jamming” test. (Time-scale relative to GPS TOW 142580.)

7.2 Ultra-Tightly coupled GNSS/INS for Dataset Acquired on a 4-wheeled Trolley in a Dense Foliage Environment

Although the end goal has been to assess the performance on the U.T.C. GNSS/INS on a UAV, a preliminary experiment was conducted on a 4-wheel trolley for initial system verification. Some of the results presented here, has also been published in [OJK17]. The analysis here has although been expanded to also include vector processing. In addition, a comparison with a T.C. solution has also been added.

The navigation payload was mounted on the 4-wheeled trolley as shown in Figure 7.3a.



(a) 4-wheeled trolley with navigation payload.

(b) Foliage passage.

Figure 7.3: Test vehicle and foliage passage.

The antenna used for this test was a Tallysman TW372, Tri-band GNSS antenna. The antenna was mounted in approximately 1.7 meters height on the trolley, in order to reduce shadowing effects from the "driver". The mission includes both open-areas (with good SV visibility) and a passage below dense foliage and tree canopies. The trees in this section are a mix of deciduous and coniferous species. The foliage section is pictured in Figure 7.3b.

The processing parameters for the U.T.C. comprises of settings for the EKF-based tracking filters and the T.C. integration filter. The settings for the EKF-based tracking filter are shown in table 7.5.

Parameter	**Value**
w_A $(dB/s/\sqrt{(Hz)})$	1
$w_{\delta\tau}$ $(m/s/\sqrt{Hz})$	0.1
$w_{clk,b}$ $(cycles/s/\sqrt{Hz})$	$2\pi f_{L1} \cdot \sqrt{(h_0/2)}$
$w_{clk,d}$ $(Hz/s/\sqrt{Hz})$	$2\pi f_{L1} \cdot \sqrt{(2\pi^2 h_{-2})}$
w_a $(m/s^3/\sqrt{Hz})$	5

Table 7.5: Processing parameters of EKF-based tracking algorithm.

The settings for the T.C. integration filter was based on the following settings; The standard deviations of pseudorange and pseudorange-rate measurements was selected to 5 m and 0.5 m/s respectively. The settings for in-run bias variability and noise parameters for the gyroscope and accelerometers was taken from table 6.1 and 6.2. The IMU update rate was 100 Hz and the GNSS update-rate was 10 Hz.

The mission was conducted the 9th of June 2017 at a parking lot on Elektrovej, Kgs. Lyngby, Denmark. It was not possible obtain a ground truth model of the experiment, but the Novatel receiver on the navigation payload has been used as a reference. In addition to the reference GNSS solution, a T.C. solution based on GPS L1 measurements from the reference receiver has also been computed. The reason for constraining this solution to only include GPS L1, is to provide a fairer comparision to the U.T.C. system. The reported horizontal trajectory from all three solutions is shown in Figure 7.4. A rough (manually drawn) truth estimate has also been added to the figure.

The start of the mission was in the open area in the left. The system was initially kept stationary for about 60 seconds to ensure stable lock conditions for the GNSS tracking and to allow time for proper INS alignment. After initialization, the trolley is dragged at walking pace in the depicted trajectory. The end of the mission is between the two buildings on the right side of the figure. It is clear from the three solutions, that the U.T.C. receiver produces the smoothest trajectory. The T.C. solution from the reference receiver, is also fairly smooth,

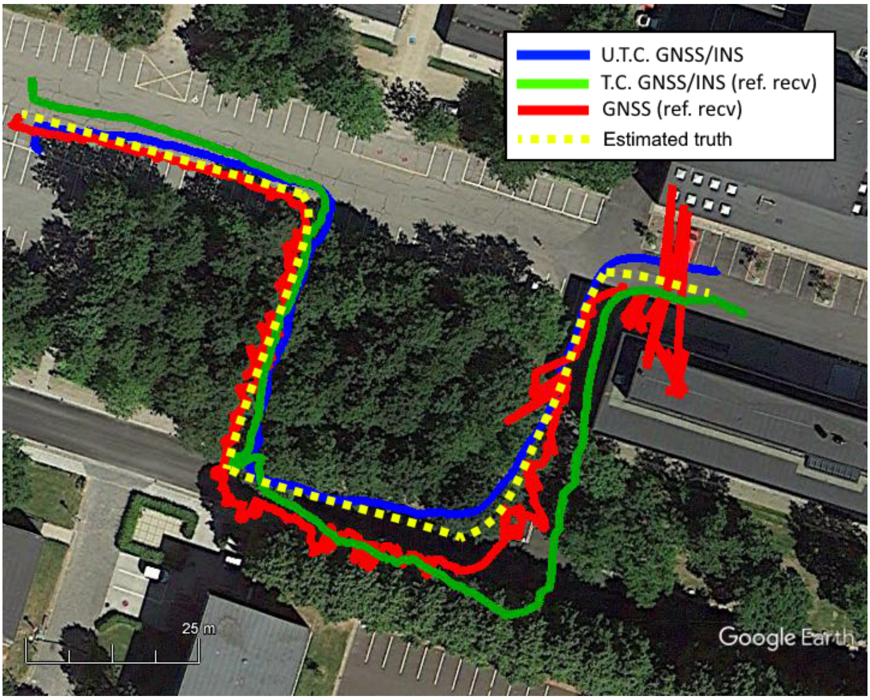


Figure 7.4: Google Earth trajectory for reference receiver (red line), T.C. solution (green line) and U.T.C. solution (blue line).

but has fluctuations. The trajectory from the reference receiver is extracted as the PSRXYZ log [Nov15]. It should be noted that this estimate, is based on multi-frequency and multi-constellation measurements and thus can rely on more SVs for the position solution. In addition, the reference receiver also has the added benefit of being able to estimate the ionospheric delay compared to the other two solutions.

The height estimation of the three systems was also investigated and the results are shown in Figure 7.5. The elevation changes encountered during the mission has been marginal, so ideally the height estimates should be approximately constant.

It can be seen that the reference solution has severe fluctuations towards the end of the mission. The estimates varies from 70 to 110 meters. This is unsurprisingly, as severe multipath and possibly NLOS is expected close to the buildings. The T.C. solution has similar variation in magnitude, but here it is apparent, that the INS filters the estimate, such that the changes are seen much more

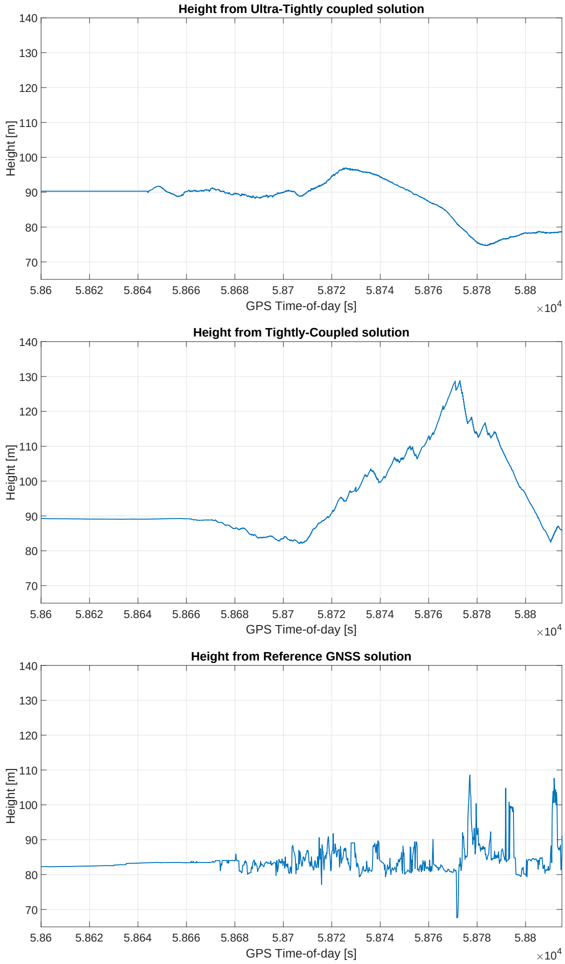


Figure 7.5: Height-solution from U.T.C. solution (top), T.C. solution (middle) and reference GNSS solution (bottom).

gradually. The U.T.C. solution, has far less variations indicating this solution provides a more robust defence against multipath and NLOS.

In addition to assesment in the position domain, it was also investigated how the systems compared with respect to signal tracking quality. For this analysis, the results has also been compared with processing results of the vector-receiver.

In Figure 7.6, the measured C/N_0 has been evaluated for the U.T.C. system, a vector receiver, a scalar receiver and the reference receiver.

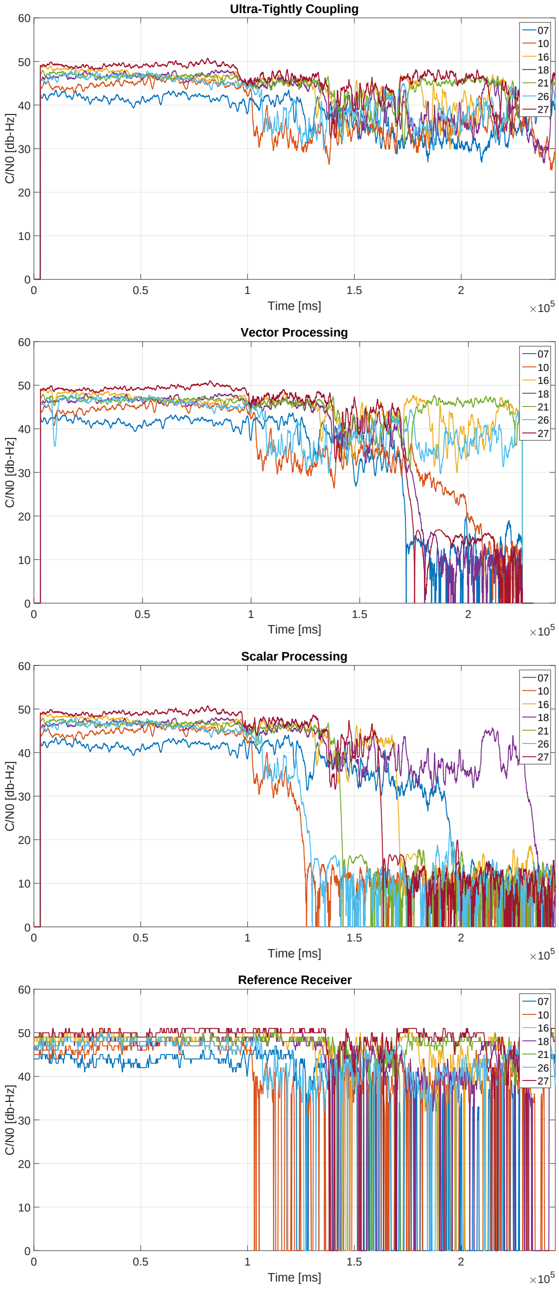


Figure 7.6: C/N_0 estimates from U.T.C. solution, vector processing, scalar processing and reference receiver. (Time-scale relative to GPS Time-of-day 58570.)

During initialization, it can be seen that the four receivers obtains similar C/N_0 values. After approximate 100 seconds into the dataset, large fluctuations starts to become apparent for all the receivers. This marks the beginning of the tree-section passage. It can be seen that the scalar-receiver gradually loses lock on SVs in this phase. The same observation can be made for the reference, but this receiver is reacquiring the SVs shortly after. The vector receiver and U.T.C. is similar affected by the degrading environment but both is able to maintain lock through this phase. An interesting observation could be made for the vector receiver, approximately 170 seconds into the dataset, 4 SVs are lost in a short time-span. This marks, the vulnerability with this type of receiver, that if measurements are allowed to corrupt the navigation filter, this would rapidly degrade the other channels.

In conclusion, the U.T.C. GNSS/INS system outperformed the vector receiver, the T.C. reference solution and the stand-alone GNSS solution. The differences was especially noticeable in terms of errors in height determination and the ability to maintain lock on SVs in a degraded environment. In the next section, the system is tested on a UAV.

7.3 Ultra-Tightly coupled GNSS/INS for a UAV flying below Foliage

In this section, a similar experiment to the trolley test is performed, but here with a UAV as a platform. Some of the results presented here, was likewise also published in [OJK17]. The results and analysis is although expanded to also include height-estimations and comparison with a T.C. solution and a vector receiver.

The mission was designed to resemble harsh reception environments which could be encountered for the water-level measurement application as stated in the introduction. For the mission, the navigation payload was mounted in the payload bay of a DJI S900 spreading wing hexacopter. This bay is suspended with rubber dampers to reduce vibrations originating from the UAV. The used antenna, an Antcom 3GNSSA-XT-1 Triple Frequency model, was mounted approximately 5 cm above the top plate of the platform. The dataset was recorded the 28th of June 2017, in Kgs. Lyngby Denmark. The dataset included the same tree passage as for the trolley test. In Figure 7.7, pictures of the UAV flying through this passage is shown.

The processing parameters for the U.T.C. was the same as for the trolley-test.

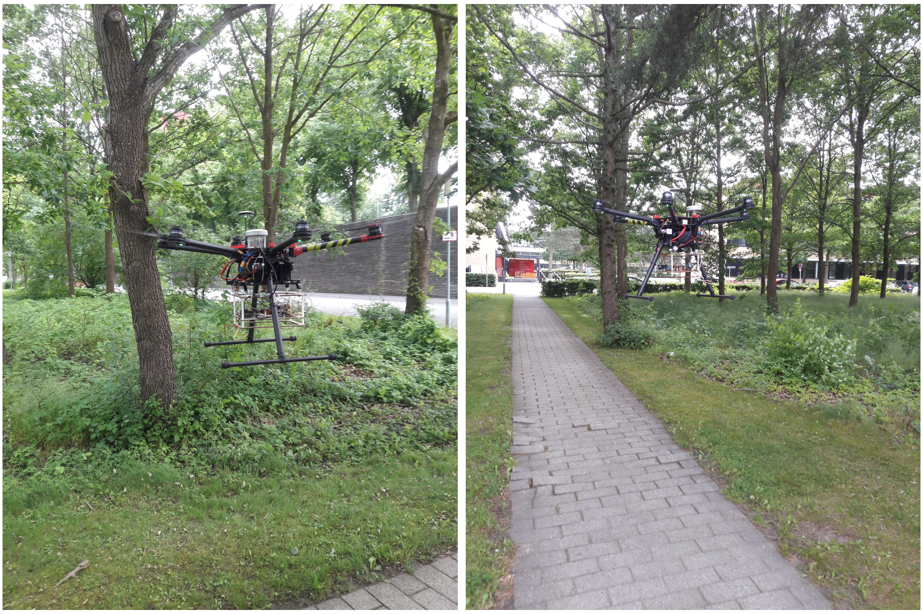


Figure 7.7: UAV flying below foliage.

The trajectory generated from the U.T.C. solution, the T.C. solution (based on GPS L1 measurements from the reference receiver) and the GNSS solution from the reference receiver is shown in Figure 7.8. A rough (manually drawn) truth estimate has also been added to the figure.

The route was flown with the UAV in a approximate height of 2 meters above the ground. The starting point was in close vicinity to that of the trolley test and includes the same foliage passage. After the passage, the UAV is piloted west along the road and through a secondary passage and the flight is terminated approximately 10 meters from the starting point.

The reference receiver has some major outliers under the tree passage. A few places the horizontal error is towards 15 meters. There is a disagreement between the T.C. solution and U.T.C. solution in terms of starting point of around 10 meters. This displacement is constant throughout the mission. The reference receiver best resembles the true starting point.

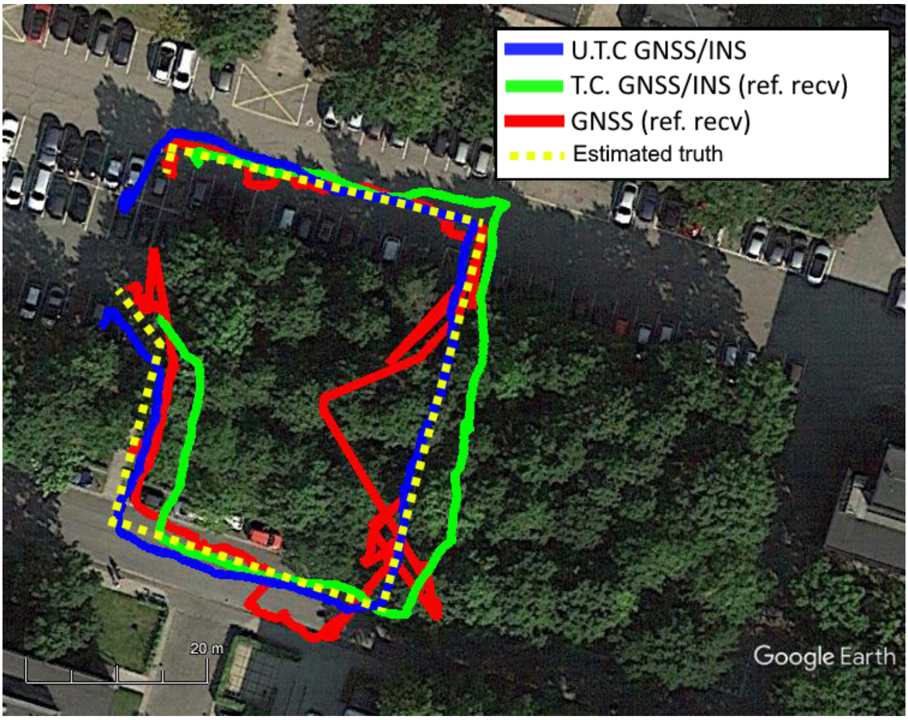
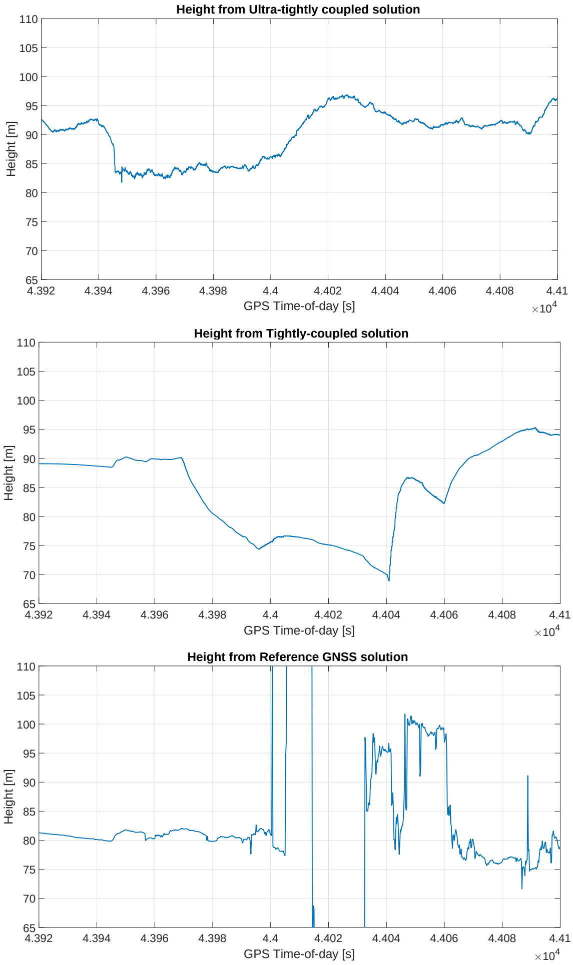


Figure 7.8: Google Earth trajectory for reference receiver (red line), T.C. solution (green line) and U.T.C. solution (blue line)

As for the trolley-test, the different solutions has also been compared in terms of reported height. The estimates are shown in Figure 7.9.

The U.T.C. solution encounters a sudden decline in height of around 7 meters. This happens around 25 seconds into the dataset, immediately before the UAV was setting of. The cause of this is believed to be related to a rapid change in clock-drift occuring at that instant. The time of take-off was at 43945 at GPS time-of-day. It is apparent that an increase in height of approximately 2 meters occurs here for the T.C. solution and from the GNSS reference receiver. During the mission, the height estimates steadily drops for the T.C. solution and reaches a minimum 20 meters below the initial height, which likely is an effect of tracking less than 4 SVs and possible disturbances from multipath signals. The reference receiver exhibits the largest variations in height. At the end of the mission, the reference receiver converges to the same height before take-off. Despite the initial drop in height from the U.T.C. solution, this system manages to produce the most consistent estimates throughout the mission.


Height from Ultra-tightly coupled solution

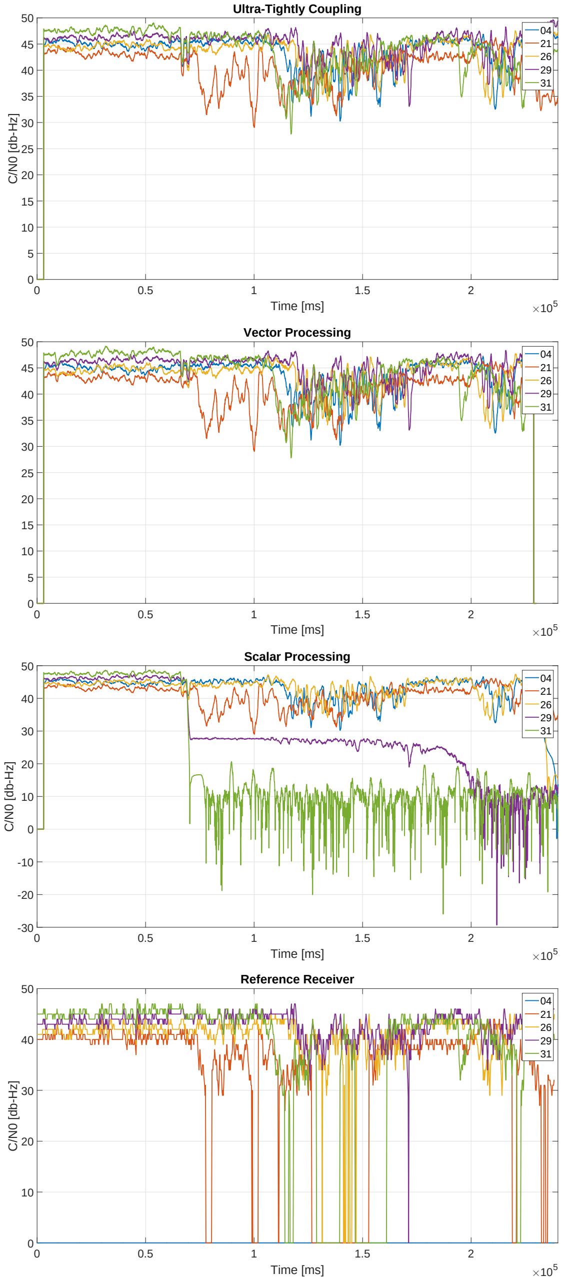
Height from Tightly-coupled solution

Height from Reference GNSS solution

Figure 7.9: Height-solution from U.T.C. solution (top), T.C. solution (middle) and reference GNSS solution (bottom).

In terms of signal tracking, the U.T.C. was compared with a vector receiver, a scalar receiver and the reference receiver. C/N_0 estimates from all systems are shown in Figure 7.10.

The U.T.C. system exhibits a similar behavior to the trolley-test as it is able to sustain tracking of all SVs through the mission. The vector receiver interestingly accomplishes the same. For the scalar receiver, a SV is immediately lost during takeoff and another SV goes into a false carrier lock condition. The behavior


Ultra-Tightly Coupling

Vector Processing

Scalar Processing

Reference Receiver

Figure 7.10: C/N_0 estimates from U.T.C. solution, vector processing, scalar processing and reference receiver. (Time-scale relative to GPS Time-of-day 43860.)

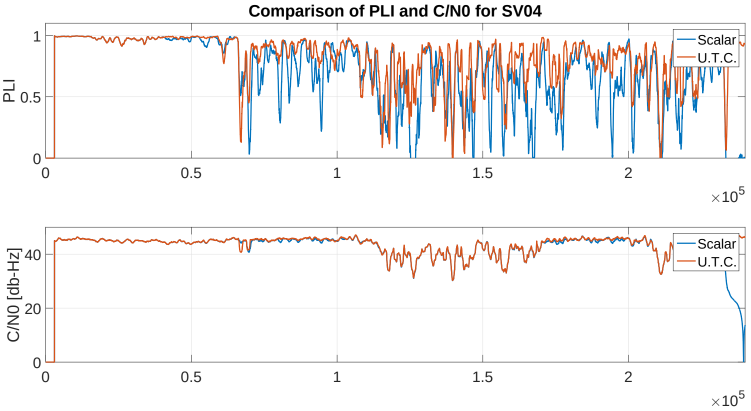


Figure 7.11: Comparison of C/N_0 and PLI for SV04 with scalar and U.T.C. implementation. (Time-scale relative to GPS Time-of-day 43860.)

is believed to be caused by a jerk-stress effect, during the initial maneuvering. The reference receiver can be seen to cope with the initial jerk well, but during the foliage passage SVs are rapidly lost and reacquired.

In addition to C/N_0 measurement, it is also of interest to inspect the achieved Phase Lock Indicators (PLI), as this indicates how well the receiver is tracking the carrier phase. A plot of PLI for SV04 is shown in Figure 7.11 for the scalar and U.T.C. receivers.

It should be immediately clear, that the U.T.C. receiver obtains superior performance. The mean values of PLI for all tracked SVs have also been calculated for both implementations and are summarized in Table 7.6

SV	**Mean PLI (scalar)**	**Mean PLI (U.T.C.)**
04	0.669	0.814
21	0.634	0.757
26	0.660	0.821
29	0.488	0.836
31	0.248	0.778

Table 7.6: Mean PLI comparison between scalar and U.T.C. GNSS/INS receiver.

The comparisons of SV29 and SV31 are although, not representative as these

SVs was lost in the scalar receiver during the mission. The improvement in PLI gives an indication of that the U.T.C. solution has a more robust carrier-phase tracking and hence could yield improvement for (precise) differential carrier phase positioning.

7.3.1 Comparison of clock-drift for trolley- and UAV missions

As stated, a rapid change in clock-drift was believe to cause the sudden drop in calculated height for the U.T.C. solution. To further analyze this behaviour, the clock-drift estimate from the integration filter is plotted in Figure 7.12.

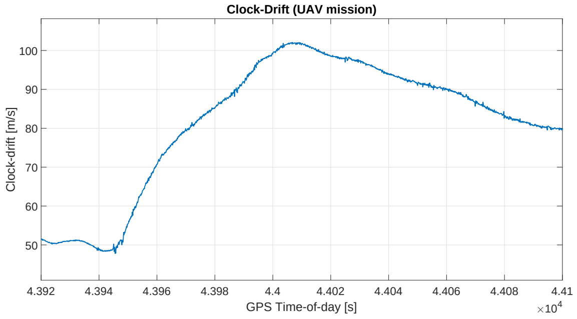


Figure 7.12: Estimated clock-drift from UAV mission.

It can be seen that initially, the clock-drift is relatively stable at approximately 50 m/s. 25 seconds into the mission, the rate starts to increase rapidly. This correlates well with the reported behaviour of the height for that instant. Approximately at the half-way point, the drift begins to decrease again. A change in the height estimate, although less sudden, can also be noticed for this point in time. The behavior of the clock drift, is assumed to be caused by vibration induced oscillator jitter as described in section 3.3.6.1. The navigation payload was suspended in rubber dampers, but it might be possible to further optimize the damping system. The reference receiver did apparently not suffer for changing drift-rates, hence it should be possible also to optimize how the observables are generated to compensate this behavior.

The height-estimates for the trolley-test has in general been more well-behaved. This can at least partly be explained by a much more stable oscillator behaviour as seen in Figure 7.13.

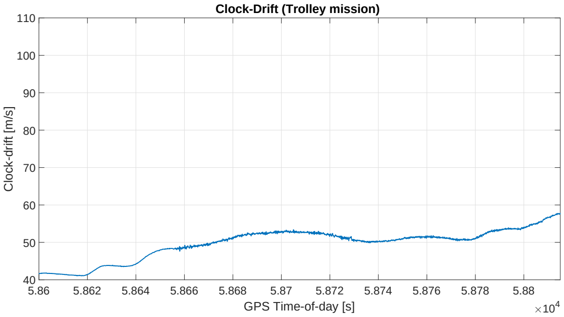


Figure 7.13: Estimated clock-drift from trolley mission.

7.4 Summary

In this chapter, the developed receiver(s) was assessed in a static-scenario, where the position accuracy was measured for a PLL/DLL, EKF-based tracking and vector implementations. The EKF-based tracking algorithm and vector receiver outperformed a classical PLL/DLL implementation during both strong and weak signal scenarios. From this test it was concluded that the PLL/DLL could function to C/N_0 values down to approx 22.5 dB-Hz, the EKF based tracking could maintain lock marginally better and the vector receiver could sustain tracking at least down to 20 db-Hz. The developed U.T.C. system were tested on a trolley-test and during a UAV mission under harsh conditions. For these tests, a ground truth trajectory was not provided, ideally a high-end commercial GNSS/INS solution would have been used for reference, but such a system was not available, neither would it have been possible to attach on the UAV. The results then has to be evaluated qualitatively. Both tests featured a passage below trees and was conducted in environments prone to multipath. In both cases, the U.T.C. solution compared favorably against a T.C. system and a survey grade GNSS reference receiver. Especially, the ability to maintain lock on SVs during the entire missions was noted. In addition, the height estimates also had less variations due to more tracked SVs.

Chapter 8

Conclusion

As the number of UAV applications increases, the importance of a robust navigation system becomes evermore important. Although the mission conducted in this work, was with a remotely piloted vehicle, completely autonomous flights are destined for the future and in that sense a failing navigation system could cause significant damage or personal injuries.

UAV navigation has also sparked a growing interest from the academic community, several navigation conferences have created dedicated UAV navigation tracks, where researchers can share their results. To the author's best knowledge, ultra-tightly coupled GNSS/INS has thus far not been explored for small UAVs. This is believed, to be caused by the lack of readily available GNSS IF recording options, which could be fitted onto the different small UAVs.

During this study, a GNSS IF data recorder was developed which can be mounted on a small UAV. The design was utilized with commercially available components and required only a minimal hardware design effort. The main challenge, was to utilize the embedded coprocessors to ensure continuous datatransfer from the ADC of the RF front-end.

Even though GNSS software receivers are very popular in the GNSS community and a few open-source implementations exists. There are none which support vector-tracking capability and integrated GNSS and INS processing. Because of this, a substantial effort was also put into the design of our own implementation.

The engineering and development effort required to build the GNSS IF datarecorder and the ultra-tightly coupled GNSS/INS solution, did however put a constraint on the number of tests which could be conducted within the time frame of the study. However, the initial results have been encouraging, clearly indicating that a ultra-tightly coupled GNSS/INS most definitely can bring improvements for UAV navigation. An important aspect for UAVs is height determination, this is often overlooked in other types of GNSS research. On this front, it was clearly proved that compared to a tightly-coupled GNSS/INS system, the ultra-tight processing did improve consistent height estimation. This was likely a direct consequence of having more SVs available for more of the time. The test scenarios in this dissertation investigated positioning performance in harsh environments, such as below dense foliage and close to buildings. Under these circumstances the position estimation showed improvement compared to a tightly-coupled system and a survey-grade GNSS receiver.

8.1 Future Research

The current system has only utilized GPS L1 C/A code signals. It should improve overall performance to include signals from other constellations as well. The GNSS IF data recorder used for the missions also captures the BOC(1,1) component of the Galileo E1 OS signal. Hence, it should just be a matter of modifying the software receiver to utilize this signal. In addition, the original IF recorder utilizing two RF front-ends could also capture GLONASS L1 C/A code signals, but due to weight constraints this was dropped for the navigation payload. An improved prototype of that system with the RF front-end evaluation kits replaced by smaller PCBs is currently under developement, but a picture of the proposed system is shown in [OJK16].

The water-level measurement application from the introduction requires either a RTK or PPP solution to obtain the necessary accuracy. Although, differential carrier phase processing has not been utilized during this study, there have been reportings that ultra-tightly coupled systems can provide more robust carrier-phase estimates, see e.g. [POL08a], [POL08b]. In addition, [BCL15] showed that by modelling the g-sensitivity of the receiver oscillator even further improvements can be obtained. Although carrier phase positioning was not assessed directly for this work, the PLI comparisons between the scalar and ultra-tightly coupled system for the UAV mission did indicate that an improvement in this regard already has been achieved, but it needs to be verified by additional tests.

Another perspective, that further could be explored is to use the developed software receiver for analysis of multipath from water with different grades of antennas. Choke-ring antennas is known to suppress reflected signals coming from below, but this type of antenna is normally heavy and thus not suited for small UAVs.

Visual Odometry (VO) is also considered as a future direction. VO systems are either based on monocular or stereo cameras and the basic concept is obtain a translation and rotation from tracked image feature points between subsequent frames. VO systems are similar to INS a dead-reckoning system and already a popular topic within UAV navigation. VO has also been exploited in ultra-tightly fusion with GNSS [APL15], [RKJB+14], but currently not for small UAVs.

As a final note, the ability to perform real-time processing on a UAV is obviously of great concern, especially if the ultra-tightly coupled system should be used for positioning during autonomous flights. In [OJK15], the concept of a 4-channel real-time GPS receiver was demonstrated. This was based on the kick-starter funded Parallella-16 SBC and the same RF front-end as used for the navigation payload. Although 4 channels are too limited for practical use, the design however demonstrated a novel way to implement a miniature real time GPS receiver.

Bibliography

[Ako97] D. M. Akos. *A Software Radio Approach To GNSS Receiver Design (PhD Thesis)*. Ohio University, 1997.

[AL03] A.S. Abbott and W.E. Lillo. Global positioning systems and inertial measuring unit ultratight coupling method, 02 2003. US Patent 6516021.

[All66] D. W. Allan. Should the classical variance be used as a basic measure in standards metrology? *IEEE Trans. on Instrumentation and Measurement*, IM-36:85–95, 1966.

[Amo83] F. Amoroso. Adaptive a/d converter to suppress cw interference in dspn spread-spectrum communications. *IEEE Transactions on communications*, vol. 31, 1983.

[APL15] B. Aumayer, M. Petovello, and G. Lachapelle. Automotive navigation using ultra-tightly coupled vision/gnss. In *Proceedings of International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015.

[BAB$^+$07] K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen. *A Software-Defined GPS and Gallileo Receiver: A Single-frequency Approach*. Birkhauser, 2007.

[BAMR03] F Bastide, D Akos, C. Macabiau, and B. Roturier. Automatic gain control (agc) as an interference assessment tool. In *Proceedings of the 16th International Technical Meeting of the Satellite Division (ION GPS/GNSS)*, pages 2042–2053, 2003.

[BCL15] S. Bhaskar, J. T. Curran, and G. Lachapelle. Improving gnss carrier phase tracking via oscillator g-sensitivity compensation. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, no. 4.:2641–2654, 2015.

[BD99] M. S. Braasch and A. J. V. Dierendonck. Gps receiver architectures and measurements. *Proceedings of the IEEE*, 87, no.1:48–62, 1999.

[Ben07] D. Benson. Interference benefits of a vector delay lock loop (vdll) gps receiver. In *Proceedings of the 63rd Annual Meeting of The Institute of Navigation*, pages 749–756, 2007.

[Bet01] J. W. Betz. Binary offset carrier modulations for radionavigation. *Navigation: Journal of the Institute of Navigation.*, vol. 48, no. 4:227–246, 2001.

[BG91] M. Braasch and F. V. Grass. Guidance accuracy considerations for realtime gps interferometry. In *Proceedings of the 4th International Technical Meeting Satellite Division of the Institute of Navigation*, pages 373–386, 1991.

[BH97] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering, 3rd ed.* Wiley, 1997.

[Bha12] S. Bhattacharyya. *Performance and Integrity Analysis of the Vector Tracking Architecture of GNSS Receivers (PhD thesis).* University of Minnesota, 2012.

[BJO$^+$17] F. Bandini, J. Jakobsen, D. Olesen, J. Antonio Reyna-Gutierrez, and P. Bauer-Gottwein. Measuring water level in rivers and lakes from lightweight unmanned aerial vehicles. *Journal of Hydrology*, 548:237–250, 2017.

[Bor08] D. Borio. *A Statistical Theory for GNSS Signal Acquisition (PhD Thesis).* Politecnico di Torino, 2008.

[Bra96] M. S. Braasch. Multipath effects. In B. Parkinson and J. J. Spilker Jr, editors, *Global Positioning System: Theory and Applications*, chapter 14, pages 547–568. American Institute of Aeronautics and Astronautics, 1996.

[CCSJ93] B.Y. Chung, C. Chien, H. Samueli, and R. Jain. Performance analysis of an all-digital bpsk direct sequence spread-spectrum if receiver architecture. *IEEE Journal On Selected Areas In Communications*, vol. 11, no. 7:1096–1107, 1993.

[Cha82] H Chang. Presampling, filtering, sampling and quantization effects on the digital matched filter performance. In *Proceedings of the International Telemetering Conference*, 1982.

[Cha11] C. L. Chang. Anti-multipath filter with multiple correlators in gnss receivers. In L. Garcia, editor, *Adaptive Filtering Applications*, chapter 18, pages 380–401. IntechOpen, 2011.

[Cur10] J. T. Curran. *Weak Signal Digital GNSS Tracking Algorithms (PhD Thesis).* National University of York, 2010.

[Die96] A. J. V. Dierendonck. Gps receivers. In B. Parkinson and J. J. Spilker Jr, editors, *Global Positioning System: Theory and Applications*, chapter 8, pages 329–407. American Institute of Aeronautics and Astronautics, 1996.

[DM13] D. Deambrogio and C. Macabiau. Vector tracking aiding for carrier phase estimation in the presence of ionospheric scintillation. In *Proceedings of the International Technical Meeting of the Institute of Navigation*, pages 333–342, 2013.

[FAGT15] P. Freda, A. Angrisano, S. Gaglione, and S. Trosi. Time-differenced carrier phases technique for precise gnss velocity estimation. *GPS Solutions*, vol. 19, no. 2:335–341, 2015.

[FPF08] E. Falletti, M. Pini, and M. Fantino. Performance evaluation of c/n_0 estimators using a real time gnss software receiver. In *Proceedings of the International Symposium on Spread-Spectrum TEchnologies and Applications (ISSSTA 2008)*, 2008.

[GJRS13] P. D. Groves, Z Jiang, M. Rudi, and P. Strode. A portfolio approach to nlos and multipath mitigation in dense urban areas. In *Proceedings of the 26th International Technical Meeting of the Satellite Division (ION GNSS+)*, pages 3231–3247, 2013.

[GMM07] P. D. Groves, C. J. Mather, and A. A. Macaulay. Demonstration of non-coherent deep ins/gps integration for optimised signal-to-noise performance. In *Proceedings of the 20th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+)*, pages 2627–2638, 2007.

[GPS06] *Interface Specification: IS-GPS-200, rev. D.* GPS Joint Program Office, 2006.

[Gro13] P. D. Groves. *Principles of GNSS, Inertial and Multisensor Navigation, 2nd Edition.* Artech House, 2013.

[Gus98] D. Gustafson. Gps signal tracking using maximum- likelihood parameter estimation. *Navigation: Journal of the Institute of Navigation*, vol. 45, no. 4:287–295, 1998.

[GWA07] M. S. Grewal, L. R. Weill, and A.P. Andrews. *Global positioning systems, inertial navigation, and integration, 2nd ed.* Wiley, 2007.

[HJGK15] L. T. Hsu, S. S. Jan, P. D. Groves, and N. Kubo. Multipath mitigation and nlos detection using vector tracking in urban environments. *GPS Solutions*, vol. 19, no. 2:249–262, 2015.

[Hop69] H. S. Hopfield. Two-quartic tropospheric refractivity profile for correcting satellite data. *Journal of Geophysical Research*, vol. 74, no. 18:4487–4499, 1969.

[IE02] M. Irsigler and B. Eissfeller. Pll tracking performance in the presence of oscillator phase noise. *GPS Solutions*, vol. 5, no. 4:45–57, 2002.

[IE03] M. Irsigler and B. Eissfeller. Comparison of multipath mitigation techniques with consideration of future signal structures. In *Proceedings of the 17th International Technical Meeting of the Satellite Division (ION GPS/GNSS 2003)*, pages 2584–2592, 2003.

[Inc17] Novatel Inc. Gnss frequency bands, 2017. https://www.novatel.com/industries/autonomous-vehicles/technology/ (2017-09-01).

[Jr.96] J. J. Spilker Jr. Fundamentals of signal tracking theory. In B. Parkinson and J. J. Spilker Jr, editors, *Global Positioning System: Theory and Applications*, chapter 7, pages 245–328. American Institute of Aeronautics and Astronautics, 1996.

[KJRB$^+$14] M. Kirkko-Jaakkola, L. Ruotsalainen, M. Z. H. Bhuiyan, S. Söderholm, S. Thombre, and H. Kuusniemi. Performance of a mems imu deeply coupled with a gnss receiver under jamming. In *Proceedings of Ubiquitous Positioning, Indoor Navigation, and Location-Based Services (UPINLBS)*, 2014.

[Klo87] J.A. Klobuchar. Ionospheric time-delay algorithm for single-frequency gps users. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23, no. 3.:325–331, 1987.

[Klo96] J.A. Klobuchar. Ionospheric effects on gps. In B. Parkinson and J. J. Spilker Jr, editors, *Global Positioning System: Theory and Applications*, chapter 12, pages 485–515. American Institute of Aeronautics and Astronautics, 1996.

[LA10] X. Li and D. Akos. Implementation and performance of clock steering in a software gps l1 single frequency receiver. *NAVIGATION, Journal of The Institute of Navigation*, 57:69–85, 2010.

[Las09] M. Lashley. *Modeling and Performance Analysis of GPS Vector Tracking Algorithms (PhD Thesis).* Auburn University, 2009.

[LB07] M. Lashley and D. Bevly. Comparison of traditional tracking loops and vector based tracking loops for weak gps signals. In *Proceedings of the 20th International Technical Meeting of the Satellite Division (ION GNSS+)*, 2007.

[LB16] G. Lachapelle and A. Broumandan. Benefits of gnss if data recording. In *Proceedings of the European Navigation Conference*, 2016.

[LT13] M. Langer and G. F. Trommer. Comparison of tightly coupled and deeply coupled gps/ins integration for automotive application using a software defined gnss receiver framework. In *In Proceedings of the 26th International Technical Meeting of the Satellite Division (ION GNSS+)*, pages 1308–1316, 2013.

[Mat17a] *MATLAB: C/C++, Fortran, Java and Python API Reference.* MathWorks Inc., 2017.

[Mat17b] *MATLAB: Parallel Computing Toolbox^{TM} - User's Guide.* MathWorks Inc., 2017.

[Max10] *MAX2769 Datasheet.* Maxim Integrated, 2010.

[MB99] G. A. Mcgraw and M. S. Braasch. Gnss multipath mitigation using gated and high resolution correlator concepts. In *Proceedings of the National Technical Meeting of the Satellite Division of the Institute of Navigation (ION NTM)*, pages 333–342, 1999.

[ME12] P. Misra and P. Enge. *Global Positioning System - Signals, Measurements and Performance (Revised 2nd edition).* Ganga-Jamuna Press, 2012.

[ML10] Y. S. Meng and Y. H. Lee. Investigations of foliage effect on modern wireless communication systems: A review. *Progress In Electromagnetics Research*, 105:313–332, 2010.

[NKG13] A. Noureldin, T. B. Karamat, and J. Georgy. *Fundamentals of Inertial Navigation, Sattelite-based Positioning and their Integration.* Springer, 2013.

[Nov15] *OEM6® Family Firmware Reference Manual, rev. 8.* Novatel Inc., 2015.

[OJK15] D. Olesen, J. Jakobsen, and P. Knudsen. Software-defined gps receiver implemented on the parallella-16 board. In *Proceedings of the 28th International Technical Meeting of the Satellite Division (ION GNSS+)*, pages 3171–3177, 2015.

[OJK16] D. Olesen, J. Jakobsen, and P. Knudsen. Low-cost gnss sampler based on the beaglebone black sbc. In *Proceedings of the 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, 2016.

[OJK17] D. Olesen, J. Jakobsen, and P. Knudsen. Ultra-tightly coupled gnss/ins for small uavs. In *Proceedings of the 30th International Technical Meeting of the Satellite Division (ION GNSS+)*, pages 2587–2602, 2017.

[PAT03] M. L. Psiaki, D. M. Akos, and J. Thor. A comparison of direct rf sampling and down-convert & sampling gnss receiver architectures. In *In Proceedings of the 16th International Technical Meeting of the Satellite Division (ION GPS/GNSS)*, pages 1941–1952, 2003.

[Pet03] M. G. Petovello. *Real-Time Integration of a Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation (PhD Thesis)*. University of Calgary, 2003.

[PFP12] M. Pini, G. Falco, and L. L. Presti. Estimation of satellite-user ranges through gnss code phase measurements. In S. Jin, editor, *Global Navigation Satellite Systems: Signal, Theory and Applications*, chapter 5, pages 107–127. IntechOpen, 2012.

[PJ02] M. L. Psiaki and H Jung. Extended kalman filter methods for tracking weak gps signals. In *Proceedings of the 15th International Technical Meeting of the Satellite Division (ION GPS)*, pages 2539–2553, 2002.

[PL06] M. G. Petovello and G Lachapelle. Comparison of vector-based software receiver implementations with application to ultra-tight gps/ins integration. In *Proceedings of the 19th International Technical Meeting of the Satellite Division (ION GNSS+)*, pages 1790–1799, 2006.

[POJK16] M. Petovello, D. Olesen, J. Jakobsen, and P. Knudsen. Are there low-cost and low-weight options for gnss if storage? *InsideGNSS*, pages 40–45, 2016.

[POL08a] M. G. Petovello, C. O'Driscoll, and G Lachapelle. Carrier phase tracking of weak signals using different receicer architectures. In *Proceedings of the National Technical Meeting of the Satellite Division of the Institute of Navigation (ION NTM)*, pages 781–791, 2008.

[POL08b] M. G. Petovello, C. O'Driscoll, and G Lachapelle. Weak signal carrier tracking using extended coherent integration with an ultra-tight gnss/imu receiver. In *Proceedings of the European Navigation Conference*, 2008.

[POL09] M. Petovello, C. O'Driscoll, and G. Lachapelle. Ultra-tight integration of an imu with gps/glonass. In *Proceedings of 13th IAIN World Congress*, 2009.

[Psi01] M. Psiaki. Smoother-based gps signal tracking in a software receiver. In *Proceedings of the 14th International Technical Meeting of the Satellite Division (ION GPS 2001)*, pages 2900–2913, 2001.

[Rem04] B. W. Remondi. Computing satellite velocity using the broadcast ephemeris. *GPS Solutions*, 8:181–183, 2004.

[RGEA08] A. Razavi, D. Gebre-Egziabherm, and D. M. Akos. Carrier loop architectures for tracking weak gps signals. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 44, no. 4.:697–710, 2008.

[RKJB$^+$14] L. Ruotsalainen, M. Kirkko-Jaakkola, M. Z. H. Bhuiyan, S. Söderholm, S. Thombre, and H. Kuusniemi. Deeply coupled gnss, ins and visual sensor integration for interference mitigation. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+)*, pages 2243–2249, 2014.

[Rod08] J. Á. Á. Rodriguez. *On Generalized Signal Waveforms for Satellite Navigation (PhD Thesis).* UNIVERSITY FAF MUNICH, 2008.

[Saa72] J. Saastamoinen. Atmospheric correction for the troposphere and stratosphere in radio ranging satellites. In S. Henriksen, A. Mancini, and B. Chovitz, editors, *The use of Artificial Satellites for Geodesy.* American Geophysical Union, 1972.

[Sal10] D. Salem. *Approaches for the Combined Tracking of GPS L1/L5 Signals (PhD Thesis).* University of Calgary, 2010.

[Skl12] B. Sklar. *DIGITAL COMMUNICATIONS - Fundamentals and Applications (2nd edition).* Prentice Hall, 2012.

[SMMB07] C. Strässle, D. Megnet, Heinz Mathis, and C. Bürgi. The squaring-loss paradox. In *Proceedings of the 20th International Technical Meeting of the Satellite Division (ION GNSS)*, pages 2715–2722, 2007.

[SNS$^+$03] N. Savage, D. Ndzi, A. Seville, E. Vilar, and J. Austin. Radio wave propagation through vegetation: Factors influencing signal attenuation. *Radio Science*, 38(5):n/a–n/a, 2003. 1088.

[SOL12] D. R. Salem, C. O'Driscoll, and G. Lachapelle. Methodology for comparing two carrier phase tracking techniques. *GPS Solutions*, vol. 16, no. 2:197–207, 2012.

[Spi96] J. J. Spilker. Foliage attenuation for land mobile userss. In B. Parkinson and J. J. Spilker Jr, editors, *Global Positioning System: Theory and Applications*, chapter 15, pages 569–583. American Institute of Aeronautics and Astronautics, 1996.

[SVR08] V. Syrjala, M. Valkama, and M. Renfors. Design considerations for direct rf sampling receiver in gnss environment. In *Proceedings of 5th Workshop on Positioning, Navigation and Communication, WPNC 2008*, 2008.

[SYG09] I. Sharp, K. Yu, and Y.J. Guo. Peak and leading edge detection for time-of-arrival estimation in band-limited positioning systems. In *IET Communications*, 2009.

[SZHP13] J. Sanz. Subirana, J.M. Juan Zornoza, and M. Hernández-Pajares. *GNSS Data Processing: Fundamentals and Algorithms.* ESA Communications, 2013.

[TFKK14] T. Tsujii, T. Fujiwara, T. Kubota, and Y. Kubo. Testing of an ultra-tightly coupled gps/ins under strong ionospheric scintillation. *Transactions of the Institue of Systems, Control and Information Engineers*, vol. 27, no. 12:476–484, 2014.

[TIS13] *Sitara AM335x ARM Cortex-8 Microprocessors.* Texas Instruments, 2013.

[Tsu00] J. B.Y. Tsui. *Fundamentals of Global Positioning System Receivers: A Software Approach.* John Wiley & Sons, 2000.

[TW04] D. H. Titterton and J. L. Weston. *Strapdown Inertial Navigation Technology, 2nd ed.* The Institution of Electrical Engineers, 2004.

[War98] P. W. Ward. Performance comparisons between fll, pll and a novel fll-assisted-pll carrier tracking loop under rf interference conditions. In *Proceedings of the 11th International Technical Meeting of the Satellite Division (ION GPS)*, pages 783–795, 1998.

[WBH06] P. W. Ward, J. W. Betz, and C. J. Hegarty. Satellite signal acquisition, tracking, and data demodulation. In E. D. Kaplan and C. J. Hegarty, editors, *Understanding GPS: principles and applications*, chapter 5, pages 153–241. Artech House, 2006.

[Xie10] P. Xie. *Improving Carrier Phase Tracking Reacquisition Time using Advanced Receiver Architectures (MSc Thesis).* University of Calgary, 2010.

[Xse15] *MTi 10-series and MTi 100-series user manual, rev. F.* Xsens Technologies B. V., 2015.

[ZG04] N. I. Ziedan and J. L. Garrison. Extended kalman filter-based tracking of weak gps signals under high dynamic conditions. In *Proceedings of the 17th International Technical Meeting of the Satellite Division (ION GNSS 2004)*, pages 20–31, 2004.

[Zha07] J. Zhang. *Precise Velocity and Acceleration Determination Using a Standalone GPS Receiver in Real Time (PhD Thesis).* Royal Melbourne Institute of Technology, 2007.

ISBN: 978-87-91694-36-3

DTU Space
Elektrovej 328
2800 Kgs. Lyngby
Tlf. +45 45259500