ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Master thesis

Automated Processing of GPS/MEMS-IMU Data for Position, Velocity and Attitude Determination

Presented by
Hugues FOURNIER

Supervised by
Dr. Jan SKALOUD
Yannick STEBLER

Swiss Federal Institute of Technology, Lausanne
Geodetic Engineering Laboratory

June 2009

Abstract

This master thesis deals with the problematic of the integration of Global Positioning System (GPS) measurements with inertial data acquired by a Micro-Electro-Mechanical System (MEMS) Inertial Measurement Units (IMU). This technology is employed for the purposes of sport performances assessment, as it enables reconstructing accurately athletes trajectories. Based on a recent development at the TOPO laboratory at the Swiss Federal Institute of Technology of Lausanne (EPFL), a new software's architecture is proposed to ensure an automated treatment of the input data. In its first phase, a Continuous Wavelet Transform (CWT) is performed to split the signal as a function of its dynamic. Then, quasi static periods are automatically identified to initialize the processing. Thereafter, several dynamic ranges can be integrated in order to compute an optimal trajectory. The performances of this new architecture are validated and evaluated using several sport experiments in skiing and biking.

The implemented method is reliable and works correctly. The process offers the capability of bridging lacks of GPS data lasting up to 10 seconds, without any substantial degradation of the trajectory's accuracy. In the frame of this project, a new MEMS-IMU was also engaged, in order to evaluate its navigation performances. It appears that its stochastic model needs to be refined and a specific initialization strategy developed before this sensor finds its place in trajectory reconstructions for downhill skiing.

Résumé

Ce projet de master s'inscrit dans la problématique de l'intégration de mesures GPS (Global Positioning System) avec des données acquises à l'aide de microsystèmes inertiels ou MEMS-IMU (Micro-Electro-Mechanical System Inertial Measurement Units). Cette technologie est employée pour l'évaluation des performances sportives, car elle offre la possibilité de reconstruire de manière précise les trajectoires empruntées par les athlètes. Sur la base d'une récente étude menée au sein du Laboratoire de topométrie de l'Ecole Polytechnique Fédérale de Lausanne (EPFL), une nouvelle architecture de logiciel est proposée afin de rendre possible le traitement automatisé des données collectées avec les sportifs. Dans un premier temps, une Transformation en Ondelettes Continue (TOC) permet de découper le signal en fonction de sa dynamique. Par la suite, les périodes quasi statiques sont automatiquement identifiées à des fins d'initialisation. Plusieurs plages dynamiques peuvent dès lors être intégrées, de manière à obtenir une reconstruction optimale de la trajectoire sousjacente. Les performances de la nouvelle architecture du logiciel sont finalement validées et évaluées à l'aide de nombreuses expériences faites à ski ou à vélo.

A la fin de ce projet, il apparaît que la méthode implémentée est fiable et fonctionne correctement. Ce procédé de traitement possède la capacité de compenser des coupures dans la réception des signaux GPS allant jusqu'à 10 secondes, sans que cela génère des répercussions notables sur la précision de la trajectoire. Un nouveau capteur inertiel a également été utilisé dans ce projet afin d'évaluer ses performances en terme de navigation. Il ressort de cette étude que le modèle stochastique doit être affiné et qu'une nouvelle stratégie d'initialisation se doit d'être développée afin que ce capteur devienne opérationnel pour la reconstruction de trajectoires dans le domaine du ski de compétition.

Remerciements

J'aimerais remercier toutes les personnes qui m'ont aidé dans la réalisation de ce projet, ainsi que tout au long de mon parcours au sein de l'EPFL.

Mes premiers remerciements iront à mon superviseur, le Dr. Jan Skaloud, qui a su m'inculquer la passion pour le domaine de la navigation. Je tiens également à le remercier pour m'avoir offert un encadrement certes exigeant, mais doublé d'une grande disponibilité, de conseils judicieux ainsi que de compétences sans égales. Merci d'avoir créé cet environnement de travail agréable, propice à l'apprentissage et au développement.

Je tiens également à remercier Yannick Stebler pour l'encadrement qu'il m'a offert durant ce projet, pour ses relectures, ses nombreux conseils et enfin pour son caractère fait d'un savant mélange d'humour et de dynamisme. Merci également de m'avoir accompagné lors de la campagne de terrain. Je garderai un excellent souvenir de cette journée à ski peu commune, sur ce qui est certainement le plus beau domaine skiable des Alpes.

Mes remerciements vont aussi à toute l'équipe du Laboratoire de topométrie: Prof. Bertrand Merminod et Pierre-Yves Gilliéron pour avoir éveillé mon intérêt au travers d'un enseignement de qualité, ainsi que pour leur disponibilité, Christian Baumann, Arnaud Deshogues, Philipp Schär et Fabrizio Tappero pour les nombreux coups de main durant ce projet. Merci également pour leur bonne humeur perpétuelle et leur humour, qui ont su me faire garder le sourire. Un merci particulier à Adrian Wägli sans qui ce projet n'aurait certainement pas vu le jour, ainsi que pour son encadrement durant la pré-étude.

Je remercie également mon oncle Alain Tornare pour son aide lors de la campagne de terrain, mais surtout pour la dextérité et la précision dont il a fait preuve pour la création du support des capteurs.

Merci également à tous mes amis et camarades, avec qui j'ai le sentiment d'avoir partagé ce qui va rester comme étant l'une des tranches les plus savoureuses de ma vie.

Finalement je tiens à remercier mes parents pour le soutien et la confiance qu'ils m'ont accordés tout au long de mes études, ainsi que pour m'avoir donné l'opportunité de choisir et d'orienter ma formation en toute liberté.

Florence, merci pour tes encouragements et ton soutien qui m'ont permis de surmonter les embûches rencontrées durant mon séjour à l'EPFL ainsi que de mener à bien ce projet. Merci également pour les relectures critiques et minutieuses très appréciées.

Key Acronyms

CWT	Continuous Wavelet Transform
COM 1	First serial port of a computer
ECEF	Earth Centered Earth Fixed
ECT.	*latin* "et cetera" - and so forth
E.G.	*latin* "exempli gratia" - for example
EKF	Extended Kalman Filter
EPFL	Ecole Polytechnique Fédérale de Lausanne
DGPS	Differential Global Positioning System
FT	Fourier Transform
GPS	Global Positioning System
GUI	Graphical User Interface
I.E.	*latin* "id est" - that is
ID	Identification
IMU	Inertial Measurement Units
INS	Inertial Navigation System
IQR	Interquartile range
L1	First GPS signal (1.57542 GHz)
L2	Second GPS signal (1.2276 GHz)
MEMS	Micro-Electro-Mechanical System
MTB	Mountain Bike
NED	North East Down
PPS	Pulse Per Second
PRN	Pseudo Random Numbers (Satellites ID)
PVA	Position Velocity Attitude
RPM	Revolutions Per Minute
RTS	Rauch-Tung-Striebel (smoother)

SD	Standard Deviation
SNR	Signal-to-Noise Ratio
SURE	Stein's Unbiased Risk Estimate
SV	Space Vehicle
UKF	Unscented Kalman Filter
UNIL	University of Lausanne
USB	Universal Serial Bus

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter aims to present the context, the problematic and the objectives of this master thesis. It also introduces the adopted methodology for reaching the objectives.

1.1 Context

Through the permanent quest to improve their performances, athletes search to have a detailed and accurate appreciation of the deeds they made during a competition. The chosen evaluation's mode is strongly related to the type of sport. Indeed, factors like dynamic, weight, costs or ergonomic constraints are the key parameters for any kind of experimentation.

For activities with a high dynamic, the study of the trajectory is crucial since the final result of a race is often related to the path taken by the athlete. For this reason, a detailed knowledge of the evolution of the Position, the Velocity and the Attitude (PVA) is an important prerequisite for analyzing and subsequently improving performances. Moreover, once the evolution with time of the position is known, other variables associated to the trajectory could also be evaluated, e.g. the heart rate or the RPM of a motor.

The specialists in navigation currently use a combination of various sensors to determine the spatial evolution of a vehicle. For example, a receiver of Global Positioning System (GPS) can be coupled with an Inertial Measurement Unit (IMU). This technique ensures the continuity in the trajectory determination because these sensors are highly complementary. Indeed, the IMU enables to follow the evolution of the trajectory when the GPS signal is too weak or even not present. On the contrary, a good reception of the GPS signals enables to quantify and correct the systematic errors in the inertial sensors that accumulates with time.

This project deals with trajectory reconstruction using satellite and low-cost inertial navigation for sports applications. Due to ergonomic and cost constraints, the traditional, bulky and expensive high-quality GPS/IMU instrumentation are less suitable for this problematic. On the other hand, smaller and less-costly sensors like single-frequency differential GPS and Micro Electro Mechanical System (MEMS) are very attractive options for many sports applications. In the frame of a recent research, the TOPO lab of the Swiss Institute of Technology of Lausanne (EPFL) developed a basic software for inertial navigation and optimal trajectory estimation via GPS/MEMS-IMU data integration. This project builds upon this very recent development with the aim of simplifying and limiting the user interactions.

1.2 Problematic

1.2.1 Technology used

The GPS technology offers the possibility to track second after second, the evolution of the athlete's trajectory. From a theoretical point of view, this technology could be a sufficient tool for tracking outdoors activity. In reality, the reception of the signals generated by the satellites is sometimes disrupted by obstructions due to vegetation, buildings or the terrain. Moreover, other factors as the high dynamic, the changes in the satellite constellation, and multipath effects make difficult to maintain the high accuracy of GPS relative positioning [21, 28]. In order to avoid these inconveniences, a MEMS-IMU is integrated with the GPS data [23, 25]. This integration strategy offers the possibility to reconstitute the trajectory of an athlete even in presence of GPS outages[1].

1.2.2 Integration of MEMS-IMU and GPS data

In order to obtain a continuous and reliable trajectory, the data collected with the MEMS-IMU measurements are integrated with the GPS observations through Extended Kalman Filtering (EKF). The choice of EKF presents the advantages in rapid convergence, good performances, low calculation overhead and easy implementation [28].

The integration is performed in two distinct phases, that can both occur several times during an experiment :

a. The initialization phase

The goal of this phase is to obtain the initial approximation of the PVA states. In other words, the initialization aims to determine the initial position, velocity and orientation of the sensors. To initialize the orientation, the inertial data from the IMU are used. On the other hand, the initial position and velocity are obtained from the GPS measurements. Once the initialization is performed, the navigation phase is enabled. More details about this issue are available in section 3.2.

[1] Cut-off up to 15 [s] for MEMS sensors, in order to keep a sufficient precision for a ski application

b. The navigation phase

Based on the knowledge of the first PVA state of the system, the integration of the data from each sensor is undertaken. The navigation phase is carried out only when the dynamic is above a certain threshold. This is the case when the GPS ground velocity is larger than 0.5 m/s and if the amplitude of the inertial measurements is close to the noise levels of the IMU signals. If the dynamic becomes too benign (e.g. stop of the athlete), the initialization step should be performed again before pursuing with the navigation phase.

1.2.3 Limits of this approach

If the athlete stops during the experimentation, the sensors are in a quasi static state. Thus, the GPS velocity is almost zero. In this phase it is difficult to evaluate the biases of the inertial sensors (more particularly for the gyroscopes). Therefore, during such period the IMU can also drift without significant degradation of the filter covariances. In such situation, the main risk is to integrate quasi static data instead of performing an initialization which will generate a new approximation of the PVA states. Furthermore, note that the convergence of the EKF is not ensured if the integration is performed during a quasi static period. For all these reasons, it is necessary to detect the quasi static periods which could appear during the trajectory, in order to avoid using them during the integration.

1.3 Objectives

In the software developed in the frame of [20], the identification of the quasi-static periods is made manually by the user. This selection is enabled through a Graphical User Interface (GUI). To obtain a more autonomous software requiring less interactions with the user, these periods have to be detected automatically. Furthermore, the tuning of the stochastic parameters used in GPS/INS data integration requires very good knowledges in inertial navigation techniques. This step may be tedious for some users. Moreover, the need for user interaction should be lowered in order to facilitate processing. Four main objectives were identified for reaching this goal:

- To improve the algorithms to automatically detect the dynamic periods within the processed signal, and enhance their independence.
- To propose and implement a new software's architecture for flexible and automated processing.
- To gather new data sets during sport experiments.
- To analyse and validate the performances of the new processing software, compared to the already existing one.

1.4 Methodology and structure of the report

The first step of this master thesis (Chapter 2) consists in the development of an automated process for extracting the dynamic periods within the experimental signal. This is necessary to distinguish the data part serving for the initial orientation estimation from that used for trajectory determination. For that, an analysis with the continuous wavelet transform (CWT) will be applied.

The second step is dedicated to the understanding of the current architecture of the software, as implemented by A.Waegli during his PhD thesis [20]. Then, a new architecture is proposed and implemented (Chapter 3) with the goal of introducing automated data treatment and minimizing the interaction between the user and the software. In this new way of treatment, the entire process is discretized into three independent parts, in order to enhance the software's flexibility. New data sets were also collected during ski and bike experiments, with a newly acquired MEMS-IMU[2] (Chapter 4).

The last step (Chapter 5) deals with the evaluation of the performances of the new processing strategy. This part demonstrates the ability of a fully automated approach for the GPS/IMU integration with low-cost sensors in sports.

[2] Crista Inertial Measurement Unit by CloudCap Technology - www.cloudcaptech.com

Chapter 2

Data pre-processing analyses

This chapter describes the automated detection of the dynamic periods, and its implementation in the new software. Some theoretical basis are also provided to help the reader to understand this process.

As seen in section 1.3, the current implementation of the software requires a manual selection of one dynamic period, which is then treated to reconstitute the PVA evolution. To avoid such intervention by the user, it is necessary to develop a process which can automatically identify the time range of the dynamic periods.

2.1 Principles of detection

2.1.1 Choice of the studied signal

Several sensor signals can be exploited to detect the dynamic periods. The proposed algorithm uses only the inertial measurements at the beginning of the detection's process. For example, the variation and the magnitude of the 3D-acceleration vector (2.1) can be used to detect whether the sensors are in a quasi static state or in a dynamic one.

$$\left\|f^b\right\| = \sqrt{(f_x^b)^2 + (f_y^b)^2 + (f_z^b)^2} \tag{2.1}$$

The angular rate provided by the gyroscopes could also be used. Nevertheless, after analyzing several data sets, it appeared that the gyroscopic observations offer a better potential to analyze the evolution of the PVA with time. Actually, this is due to the fact that the accelerometer signals are more affected by parasitic noise (e.g. the vibrations produced by a motorbike's engine) than the gyroscopic measurements. This is shown on Fig. 2.1.

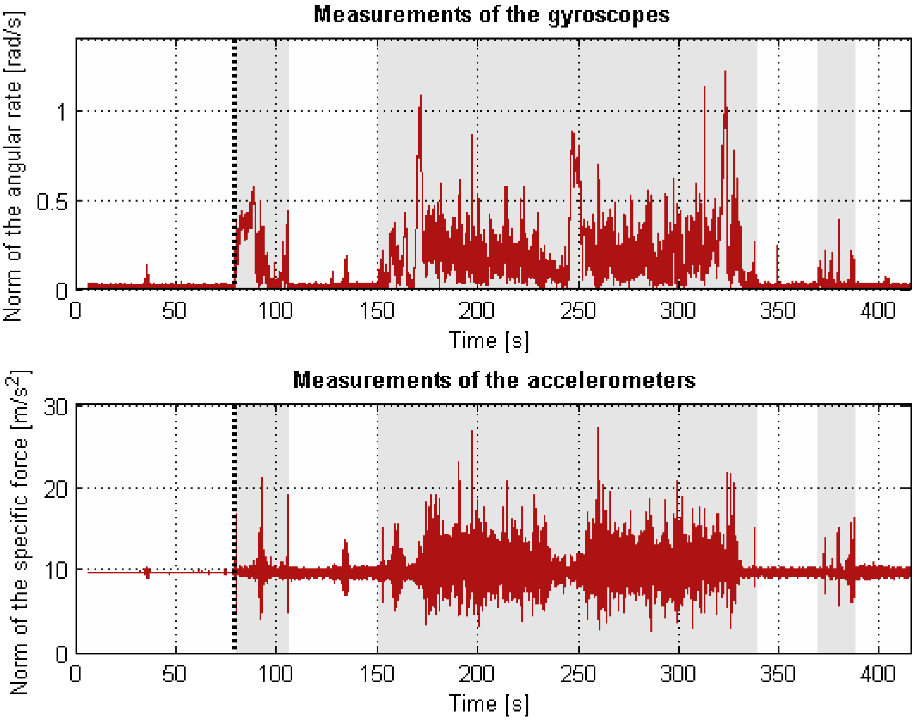


Fig. 2.1: Comparison between the measurements of the gyroscopes and those of the accelerometers. The black vertical doted lines show the motorbike's ignition. The shaded ranges are qualified as dynamic, and the rest as quasi static.

If we analyze the two signals when the motorbike is off, it appears that the accelerometers have a tiny noise compared to the gyroscope outputs. Then, when the engine is turned on (time $t = 86$ s), the noise on the gyroscopes does not change. Contrary to the accelerometers measurements, whose noise is greatly amplified.

In order to develop a flexible software able to deal with different applications (ski, motorbike, car, etc.), it's necessary to avoid that the choice of the carrier disrupts the algorithms. For this reason, all the following procedures for the automatic detection of dynamic periods will be based on the gyroscopic norm (2.2). Where the gyroscopic norm is null and the 3D acceleration vector isn't equal to $\mathbf{g}$, the GPS ground velocity is used to detect the dynamic periods.

$$\|\omega_{lb}^{b}\| = \sqrt{(\omega_{lb,x}^{b})^2 + (\omega_{lb,y}^{b})^2 + (\omega_{lb,z}^{b})^2} \tag{2.2}$$

2.1.2 Dynamic state and associated treatment

The processing mode changes as a function of the dynamic level. This part of the report presents the different states of dynamic together with the associated treatments.

The detection process works in two distinct stages. The first stage searches for dynamic changes by analyzing only the gyroscopic norm. The second one confronts this first classification to the ground velocity measured by the GPS receiver. At this point, the reader should keep in mind that the used data are collected with two different sensors (GPS and MEMS-IMU) which are synchronized on the GPS time. A GPS signal without corresponding inertial informations is not helpful for the complete reconstitution of the trajectory. Thus, such parts of the GPS signal are neglected. In consequence, the time's boundaries of the experiment are set to the begin and the end of the inertial measurements.

In terms of implementation, a numeric ID is associated with each time step. This gives an indication of the dynamic status and thus the required treatment. Fig. 2.2 resumes all the different combinations of the GPS and the MEMS-IMU signals. Each state of dynamic receives an associated treatment: "None", "Initialization" or "Strapdown integration".

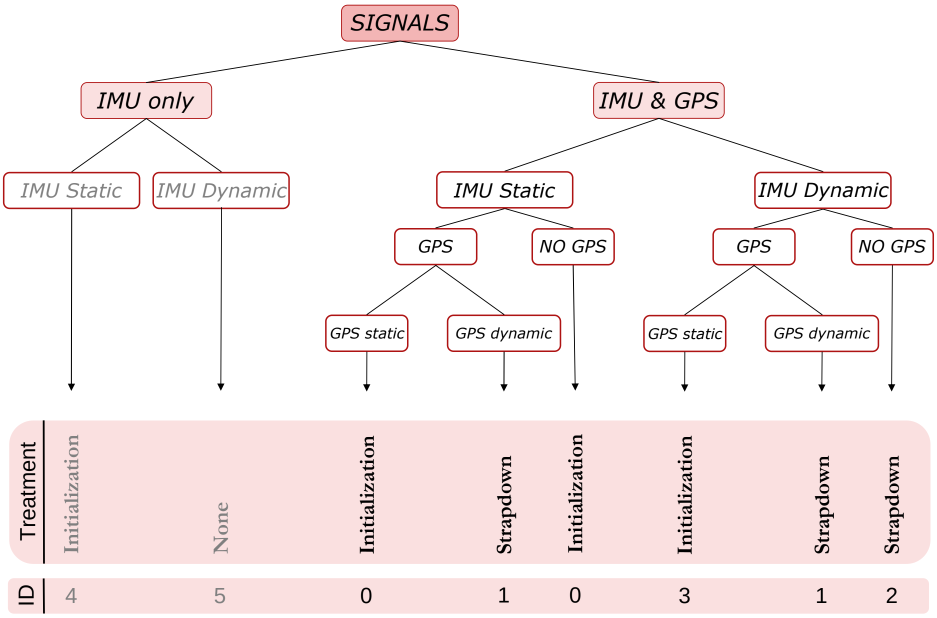


Fig. 2.2: Hierarchical tree of the different combinations that are possible with the GPS and IMU signals. The respective treatments and numerical ID are also depicted.

The first two branches of the tree highlight the fact that there are several ways to combine two signals. On the left side, only the inertial signal is available. This happens when the IMU is turned on but not yet the GPS receiver. The aim of the process reserved for this kind of data set is only to detect if the IMU is in a dynamic or quasi static period. It is important to note that only the last/first quasi static range before/after the period during which both the GPS and the MEMS-IMU are turned on is kept to compute the initialization. The rest of the data classified into the left branche is ignored. More informations on the detection itself is given in section 2.3.2

On the right branche in Fig. 2.2 , both signals are available for the trajectory reconstruction. The first step detects if the IMU is in a dynamic or a quasi static period. Then, based on this first classification, the analysis is refined by using the ground velocity measurements sensed by GPS. Again, several situations can appear. Indeed, whatever the outcome of the first classification based only on MEMS-IMU, the GPS may indicate a dynamic period if the ground speed is $> 0.5\,\mathrm{m/s}$, or a quasi static range when the ground speed is $< 0.5\,\mathrm{m/s}$, or a period without GPS measurements.

If no GPS data are available, then the IMU is trusted. Thus, if the analyses of the inertial measurements indicate dynamic then the data are processed to reconstruct the trajectory. On the contrary, if the IMU is identified as quasi static period, then the data will serve for initialization purpose.

Tab. 2.1: List of the classification categories, of their ID and of their associated treatment[1].

Category	**ID**	**Associated treatments**
IMU Static + GPS Dynamic	1	Strapdown Navigation
IMU Static + GPS Static	0	Initialization
IMU Static + No GPS	0	Initialization
IMU Dynamic + GPS Dynamic	1	Strapdown Navigation
IMU Dynamic + GPS Static	3	Initialization
IMU Dynamic + No GPS	2	Strapdown Navigation
IMU Static only	4	Initialization
IMU Dynamic only	5	None

Once the detection process is performed, six different categories are created, describing the states of the dynamic encountered during the experiment. Each of them has its own ID and represents a specific data treatment. Tab. 2.1 summarizes all the categories used for the classification, their ID and the corresponding treatment.

[1] The case where only GPS data are available is not represented since it is inexploitable for this application.

2.2 The wavelet transform

The Continuous Wavelet Transform (CWT) is used on the gyroscopic norm for the first analysis. This section presents a non-exhaustive theoretical basis on the CWT, which will help the reader in understanding why this operation is used and how it works.

2.2.1 Fundamental principles

Many studies in various research domains such as mathematics, medicine, physics or imaging conclude that the most important part of the informations of a signal is contained in its irregularities [9]. For this reason, tools like the Fourier Transform (FT) or the CWT are frequently used to solve numerous scientific problems.

In a FT, the basic supposition is that each signal can be expressed as a set of co-sinusoidal and sinusoidal functions. The well known FT enables to characterize which frequencies represent the analyzed signal. With this type of analysis, it is however not possible to locate in time a given frequency. A solution to this problem is to cut the signal into subsets and to proceed a local transformation of each of them. This procedure works but doesn't give adequate results every time [17].

The CWT approach, on the other hand, offers an analysis in time and frequency domains [17], that enables accurate localization of the signal's singularities. Schematically, this method consists of sliding a window stepwise along the analyzed signal and computing the spectrum at each stop. The complete time-frequency analysis is obtained by varying the size of the window [17].

2.2.2 Continuous wavelet transform

The first step of the CWT is the choice of a mother wavelet, noted $\psi(t)$. This last one is used as basis function in representing other functions (like the sines and cosines in the FT). The simplest mother function is the Haar wavelet, which is illustrated in Fig. 2.3.

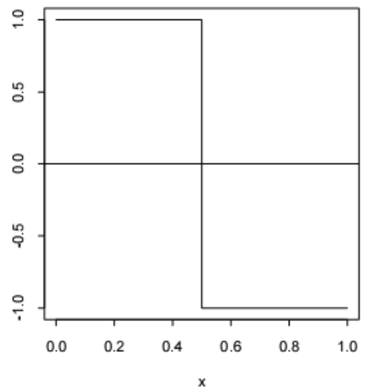


Fig. 2.3: Representation of the Haar wavelet [19].

In order to be an admissible wavelet, a function has to satisfy the relation given in the equation (2.3).

$$\int \psi(t)dt = 0 \tag{2.3}$$

New wavelets can be generated by dilating and translating $\psi(t)$ (2.4). This is depicted in Fig. 2.4. The dilatation enables the analysis of the frequency scales, and the translation is used to analyze the time scale [19].

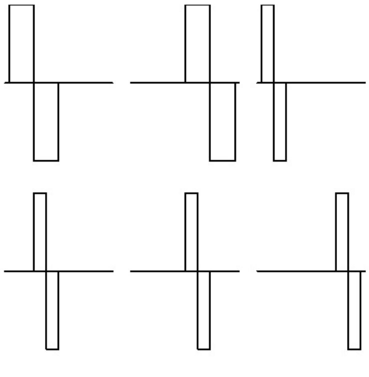


Fig. 2.4: Dilatations and translations of the Haar wavelet on [0,1] range([19]).

$$\psi_{\tau,s}(t) = \frac{1}{s}\psi\left(\frac{t-\tau}{s}\right) \tag{2.4}$$

With:

s	= scale factor
τ	= translation factor
ψ	= mother wavelet
$\psi_{\tau,s}(t)$	= wavelet generated by rescaling and translation

Through this approach it becomes possible to generate various wavelets, and one can describe the analyzed signal as a superposition of all of them (2.5).

$$f(t) = K \int \Gamma(\tau, s)\psi_{\tau,s}(t)\delta\tau\frac{\delta s}{s} \tag{2.5}$$

With:

K	= constant dependant only of $\psi(t)$
$\Gamma(\tau, s)$	= coefficients of the CWT

This last equation can also be inverted to obtain the coefficients of the CWT for a given analyzed function $f(t)$. This is shown in the equation (2.6).

$$\Gamma(\tau, s) = \frac{1}{s}\int f(t) * \psi\left(\frac{t-\tau}{s}\right)\delta t \tag{2.6}$$

With the convolution product expressed as (2.7):

$$f * g(x) = \int_{-\infty}^{+\infty} f(u)g(x-u)\delta u \tag{2.7}$$

Thus, it is possible to obtain a CWT for every signal $f(t)$. The singularities contained in a signal will be traduced by high coefficients in the CWT. Refining the analysis in the frequency scale gives an accurate time localization of the singularities.

2.2.3 Theoretical example

The CWT constitutes a very powerful tool to localize singularities in a signal. To illustrate this, a theoretical signal has been generated and treated with the CWT. The coefficients thereby obtained are then represented in a time-scale space (Fig. 2.5).

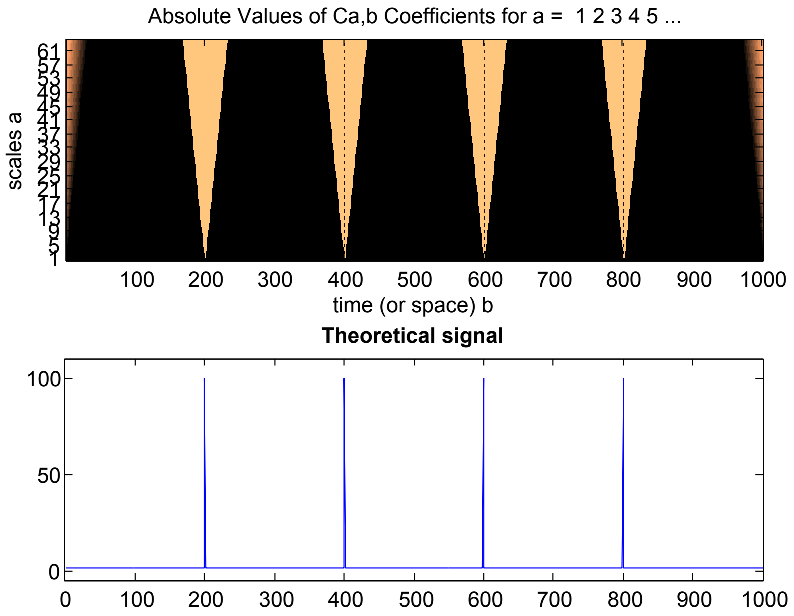


Fig. 2.5: CWT analysis on a theoretical signal.

The lower graph in Fig. 2.5 shows the theoretical signal which is composed of four distinct peaks. By looking at the coefficients (upper graph), it clearly appears that the CWT is able to localize the singularities of a signal over time. Short windows will be able to detect only fast signal variations (i.e. high frequency) while large scales detect slow signal variations (i.e. low frequency) with a smoothing effect .

2.3 Implementation of the automated detection

This section renders a detailed description of the automated analysis. This process aims to detect the parts of the data which will serve to compute the initial orientation and those which will be used for the strapdown navigation. This detection is carried out just before the integration step. The new software can treat many dynamic ranges, which was not the case with the old program.

The algorithm that classifies the data is composed of three main stages based on CWT. In a first time, the input data are de-noised using wavelets. Then, the CWT is applied to this de-noised signal providing the CWT coefficients. These are smoothed with a spline and finally analyzed to detect the singularities of the studied signal. This first classification is done using only the data from the MEMS-IMU. To finish the detection process, these results are confronted with the smoothed ground velocity which was acquired by GPS. After this last step a state vector is created, containing the status of the dynamic at each time step of the experiment. The integration process then follows (see chapter 3).

The individual processing steps are depicted in the work flow of Fig. 2.6. The three main (de-noising, CWT, IMU/GPS dynamic detection) will be studied later on, separately.

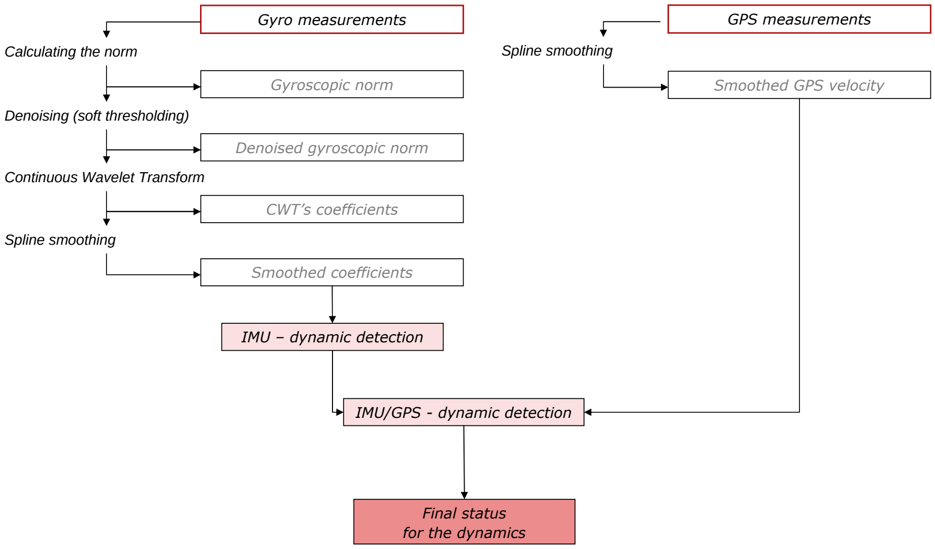


Fig. 2.6: Work flow for the automated detection of the dynamic periods.

2.3.1 De-noising of the signal

The de-noising step is important in order to obtain a sharper detection with the CWT. De-noising by wavelet transform was chosen over classical spline smoothing because it preserves the characteristics of the underlying signal. Moreover, this approach has been shown to work well in a range of situations where many non-wavelets methods have met only partial success [4, 14].

The de-noising by wavelet transform, also called de-noising by soft-thresholding [4], is a procedure that aims to reject noise by thresholding the wavelet coefficients of the noisy signal. This treatment acts in three different stages. First, a wavelet transform analyzes the signal and yields coefficients. The wavelet used for this type of application is a Daubechies 20 (Db20) wavelet, and is represented in Fig. 2.7. This family of wavelet is well adapted because it is less discontinuous and may be better suited at representing smoother variations [3, 14]. Then, the obtained coefficients are thresholded in order to reject the noise. Finally, an inverse wavelet transform rebuilds the de-noised signal from the thresholded coefficients.

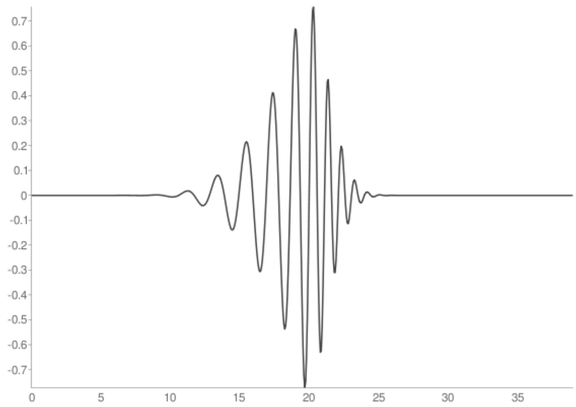


Fig. 2.7: The Daubechies 20 wavelet.

The first and the last stages of the process consist in the application of a filter bank on the signal or on the thresholded coefficients. The theory related to these filters will be exposed in section 2.3.2.

The de-noising by soft thresholding is done using the Matlab function *wden*, which performs an automatic de-noising process of a 1-D signal using wavelets. Six input parameters are required and liste below.

1. **The signal to analyze** which is, as seen on Fig. 2.6, the gyroscopic norm.

2. **The threshold selection rule** which is crucial. For this project a heuristic variant of the principle of Stein's Unbiased Risk Estimate (SURE) was used. This is a near optimal thresholding method using an adaptative threshold selection, which is well suited to recover a function of unknown smoothness from noisy sampled data [5, 14]. Indeed, for each dyadic resolution level of the wavelet decomposition, a specific threshold is assigned by minimizing the SURE for the threshold estimates [5]. Moreover, this method has already been used and empirically tested to de-noise inertial data and works well for such applications [14].

3. **The type of thresholding**. Soft thresholding was chosen for its capacities in reducing noise, by opposition to hard thresholding which is better in preserving edges [11]. Equation (2.8) elucidates the function of the soft thresholding method.

$$D\left(Y,\lambda\right) \equiv \begin{cases} sgn\left(Y\right)\left(\|Y\| - \lambda\right) & \text{if } \|Y\| > \lambda, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

With:

$D\left(\,\cdot\,,\lambda\right)$ = thresholding operator with threshold λ
Y = coefficients of the CWT
sgn = sign of a real number

4. **The threshold rescaling method**. With the chosen one, σ is estimated empirically as the median of the wavelet coefficients at the finest scale, as suggest by Donoho and Johnstone [5]. This method was chosen because no prior knowledge on the noise power is available.

5. **The number of decomposition's levels** was empirically determined using various inertial signals. It has been fixed at five.

6. **The mother wavelet** used for this de-noising is a Db20 wavelet.

The entire process leads to an adaptative de-noising of the gyroscopic norm. Fig. 2.8 illustrates this theoretical part with the result of a de-noising of inertial data by the soft thresholding. It appears clearly that the characteristics of the signal are conserved. This would not be the case with a simple spline smoothing which would distort the signal to a greater extent.

2.3.2 The wavelet decomposition

The next step is the CWT. First it is necessary to choose a mother wavelet. For the detection of fast variations, the Haar wavelet appears to be well adapted, owing to its properties to emphasize discontinuities in the raw data set [3].

As seen before, a wavelet transform is a time-frequency analysis. Therefore the dilatations and the translations must be well chosen to get optimal results. The translations of the wavelet are limited by the duration of the signal. The choice of dilatation is more complexe. Choosing a small scale will not help due to the fact that the coefficients will therefore contain a lot of noise. Contrariwise, a big scale can lead to a less accurate localisation.

To determine the number of scales needed, Valens [17] suggests considering a series of dilated wavelets as a band-pass filter bank. This last one is a structure composed of a set of high-pass and low-pass filters, and was first introduced in the 1980's [15]. Fig. 2.9 shows how a filter bank works.

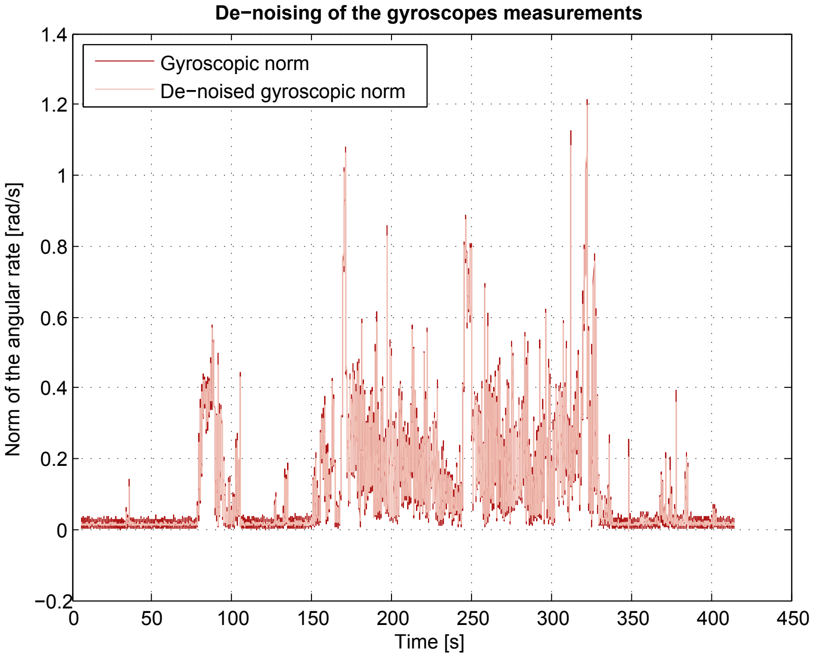


Fig. 2.8: De-noising of inertial data by soft thresholding.

To identify which scales are useful for the treated problematic, the signal spectrum is split with an iterated filter bank. First, the signal is split into two equal parts, a low-pass and a high-pass part. A resampling is done in order to preserve the signal length. The wavelet coefficients are obtained from the high-pass part [3]. The low-pass part is split again because it still contains interesting details. However, this process can be done indefinitely[2].

To reduce the computation as much as possible, it is important to select the number of filters that are really needed. This number is related to the length of the studied signal by the following relation (2.9) [3] :

$$J = \log_2 (N) \tag{2.9}$$

With:

J	= number of useful scales
N	= number of samples in the signal

Then, a new selection is done among these useful scales to retain only the interesting frequencies. Indeed, each time the signal pass through the filters, its frequency is split in two. The frequencies of interest for ski applications are between 0.1 and 10 Hz. Thus, for a signal sampled at 100 Hz, 7 scales are selected (4 to 10). This is illustrated in Tab. 2.2.

[2] Dichotomy paradox by Zeno of Elea (490-430 BC) [17]

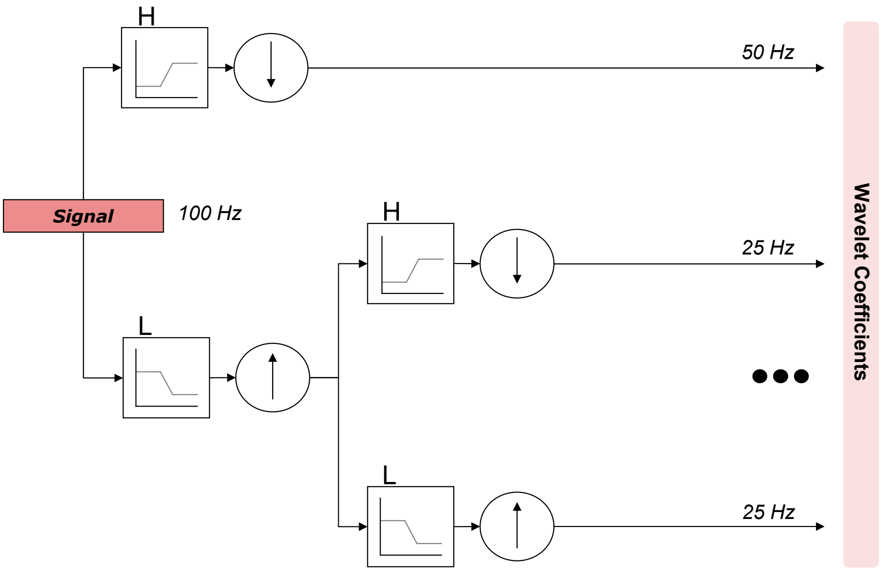


Fig. 2.9: Splitting of the signal spectrum using an iterated filter bank [14]. H represents high-pass filter and L a low-pass filter.

Tab. 2.2: Application of a filter bank on a 100 Hz signal of length N.

Level	**Frequency**		**Samples**
	From [Hz]	To [Hz]	
1	50	100	N/2
2	25	50	N/4
3	12.5	25	N/8
4	**6.25**	**12.6**	**N/16**
5	**3.12**	**6.25**	**N/32**
6	**1.56**	**3.12**	**N/64**
7	**0.78**	**1.56**	**N/128**
8	**0.39**	**0.78**	**N/256**
9	**0.19**	**0.39**	**N/512**
10	**0.09**	**0.19**	**N/1024**
11	0.05	0.09	N/2048
12	0.02	0.05	N/4096
12	0	0.02	N/4096

To summarize, gyroscopic norm is first de-noised. Then, the resulting signal is passed through a filter bank to get the CWT's coefficients. Only the coefficients of the interesting scales are then selected as a function of the frequencies they represent. To localize the singularities with the help of these coefficients, an additional operation is required. As suggested by Capilla [3], the coefficients are squared and then summed to compute the energy contained in the range of frequencies of the studied signal. Indeed, the energy preservation characteristic of the CWT can be expressed in the discrete case with the relation (2.10). Thus, the energy contained at scale j and locations $i = 0$ to $2^{J-j} - 1$ is given by the equation (2.11)[3].

$$\sum_{t=0}^{N-1} f\left(t\right)^2 = \sum_{j=1}^{J} \sum_{i=0}^{2^{J-j}-1} \left[Y\left(j,i\right)\right]^2 \tag{2.10}$$

With:

Y	= coefficients of the CWT
J	= number of scales
N	= number of samples in the signal

$$E(j) = \sum_{i=0}^{2^{J-j}-1} \left[Y\left(j,i\right)\right]^2 \tag{2.11}$$

With:

$E(j)$	= energy contained at scale j
Y	= coefficients of the CWT
J	= number of scales

The summation (2.11) generates a energy's coefficients vector, which has to be smoothed with a spline. A simple thresholding is then applied using the average of the smoothed coefficients as threshold. Values larger than the threshold are considered *dynamic* and those below the threshold *quasi static*. This first classification is finally stored in a state vector for the confrontation with the GPS measurements (section 2.3.3).

2.3.3 Confrontation with GPS

This part of the process confronts the IMU-based classification with the GPS velocity measurements. This process is performed according to the rules of detection listed in Tab. 2.1 and provides the final classification.

2.4 Example of a detection

Fig. 2.10 illustrates the results that are obtained after the automated detection. The processed signal is a gyroscopic norm sensed on a motorcycle which was already used in Fig. 2.1 and Fig. 2.8.

Tab. 2.3: Listed results of the automatic detection. Flag B stands for backward and flag F for forward.

DYNAMIC RANGES -IMU time-

Range	Strapdown_t1	Strapdown_t2	Orient_t1	Orient_t2	Flag
*****	*************	*************	**********	**********	****
1	309171.5580	309186.9480	309186.9580	309193.1480	B
2	309193.1580	309194.4780	309186.9580	309193.1480	F
3	309195.2980	309196.1980	309196.2080	309244.1480	B
4	309244.1580	309422.2480	309196.2080	309244.1480	F
5	309452.1580	309456.4480	309456.4580	309457.0480	B
6	309457.0580	309459.2480	309456.4580	309457.0480	F
7	309495.1580	309496.1480	309496.1580	309506.3680	B

The three graphs of Fig. 2.10 represent the normed measurements from the GPS ground velocity, the accelerometers and the gyroscopes. The green parts of the signals are classified as dynamic, the red as quasi static. It appears that the automated detection works well for a variety of applications (ski, MTB, motorcycle, car, etc.).

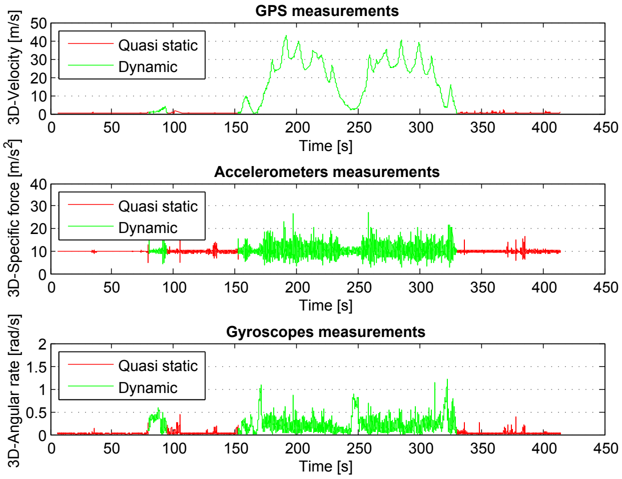


Fig. 2.10: Results of the automated detection on GPS and IMU measurements done with a motorcycle.

The time boundaries of the different periods are saved in a text file as shown in Tab. 2.3. Each identified dynamic range is listed with its start and end time, namely *Strapdown_t1* and *Strapdown_t2*. Moreover, a quasi static period which is adjacent to the dynamic period is also saved to calculate the initialization (*Orient_t1* and *Orient_t2*). The *flag* indicates the direction of the first integration. Subsequently, the inverse direction is used for optimal smoothing.

Chapter 3

Architecture of the software

This chapter describes first the principles of the GPS/INS integration. Then, the previous structure of the integration software is presented. Finally, the new structure capable of automatizing the trajectory's computation is introduced.

3.1 Notation

In the following sections, vectors are represented by a lower case bold letter. Capital bold characters are reserved for the matrices.

A superscript after a vector $\left(\mathbf{f}^{\,b}\right)$, indicates in which reference frame it is presented.

A rotation matrix between two coordinate systems is represented by a capital bold letter, followed by a superscript and a subscript. For exemple, the matrix $\mathbf{R}_b^l$ represents the transformation from the body frame b to the local frame l.

3.2 Principles of the GPS/INS integration

The reader should note that this section provides only an overview of the GPS/INS integration principles. For more detailed information, the reader is refered to the the PhD thesis of A. Waegli [20] or to the book of Titterton and Weston [16]. Good part of this section is taken from these two references.

3.2.1 Requirements for the combination of sensors

In order to combine the measurements of several sensors (e.g. GPS and IMU), they must be first synchronized on a common time scale. The GPS time is often used for this purpose.

Then, the data from one sensor must be expressed in the frame of the other sensor. Usually, the GPS measurements are brought to the center of the IMU. The latter is defined as the intersection of the three axes of the accelerometers. For this reason, the IMUs and the GPS antenna are rigidly fixed on the same structure. The spatial displacement (called lever-arm) between them is measured in the IMU body frame.

Once these operations on the different measurements are completed, an integration strategy is required. As seen in section 1.2.2, the EKF was chosen for this project. Other strategies like the Unscented Kalman Filter (UKF) are potentially more robust for MEMS-IMU applications but have other inconveniences [7, 12, 13, 18, 20]. Due to time constraints, only the EKF solution was implemented in this project.

Trajectory estimation

The trajectory estimation aims optimally determining the navigation states (i.e. PVA) together with the error states (e.g. sensor biases). The state vector combines all states together and the method of Kalman Filtering provides their correction. The trajectory estimation process consists of three steps:

- The first step is the **initialization**, during which the initial PVA state and the covariance matrix are initialized. This part may also include the estimation of the initial sensor's biases. It should be noted that the GPS provides the initial values for the position and the velocity.

- The **prediction** propagates the PVA, based on a propagation model expressed as a set of non-linear equations (i.e. strapdown inertial navigation [16]). The covariance matrix is also propagated at the same time. This happens until the first external measurements become available for an update. The covariance values corresponding to the state vector increase with time until the update, as a function of dynamic and stochastic models.

- The **update** is performed as soon as external informations are available. The propagated estimation of the state is improved by the measurements, in this case GPS position and/or velocity.

All these steps are schematically depicted in Fig. 3.1. It can be seen that the update improves the accuracy of the state vector. This is illustrated by the evolution of the position's standard deviation.

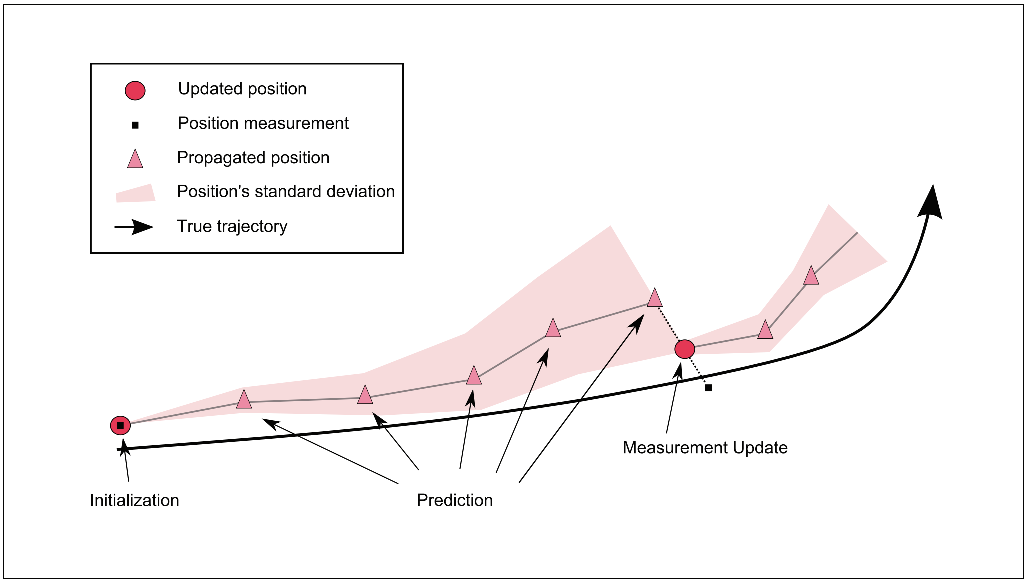


Fig. 3.1: Schematic representation of the GPS/INS integration via Kalman filtering process. Adapted from [20].

3.2.2 Implementation of the integration

Reference frame

Before the integration begins, it is necessary to choose a reference coordinate frame. As shown in Fig. 3.2, various frames are available. Each frame has distinct properties and is more suited for a specific kind of application. All of them are "Cartesian" and right-handed frames.

- The inertial frame has its origin at the center of the Earth and is fixed in space (non-rotating), with its X-axis pointing in direction of vernal equinox. The Z-axis coincides with the mean Earth rotation axis. It is represented by the index i in Fig. 3.2.

- The Earth Centered Earth Fixed (ECEF) frame has its origin at the center of the Earth. Unlike the inertial frame, this reference system rotates with the Earth. The X-axis passes by the intersection of the equatorial plan and the Greenwich meridian. The Z-axis coincides with the mean Earth rotation axis. The Y-axis is set as to form a right-handed frame. The rotation rate between the inertial and the ECEF is noted Ω. The index e is used for this frame.

- The local frame has its origin at the surface of the Earth in the vicinity of the navigation's sensors. In Fig. 3.2, this is represented by point P. The three axes form an orthogonal right-handed system. The X-axis and the Y-axis are aligned respectively with the directions of the north and the east. The Z-axis completes the frame and points down in direction of the center of the Earth. In a particular sequence, this frame is also called NED (North East Down) or navigation frame. It is used as the reference frame for this project and its corresponding index is n.

- The body frame is attached to the studied body (e.g. an athlete), and the axes are aligned with its three rotation axes (roll, pitch and heading). The index b is used to describe everything related to this frame. Unless specified otherwise, it will be assumed that the body frame coincides with the IMU sensor frame.

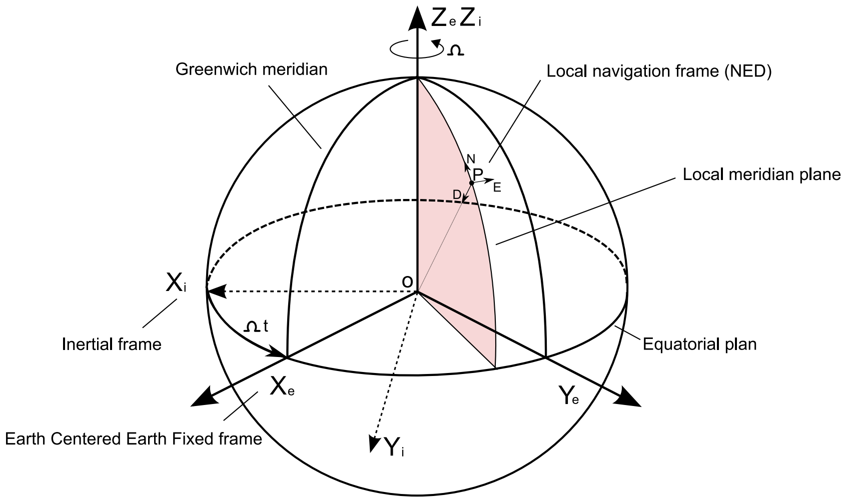


Fig. 3.2: Inertial, ECEF and navigation frame.

Definition of the state vector

The error state vector (3.1) contains the correction terms for the nine navigation parameters which represent the PVA state. In this project, nine additional states are used for the corrections of the systematic errors of inertial and magnetic measurements (3 x accelerometer biases, 3 x gyroscope biases and 3 x magnetic disturbance) [20].

$$\delta x = \begin{bmatrix} \delta\varphi \\ \delta\lambda \\ \delta h \\ \delta v_N \\ \delta v_E \\ \delta v_D \\ \delta\epsilon_N \\ \delta\epsilon_E \\ \delta\epsilon_D \\ \delta b_{f_x} \\ \delta b_{f_y} \\ \delta b_{f_z} \\ \delta b_{\omega_x} \\ \delta b_{\omega_y} \\ \delta b_{\omega_z} \\ \delta d_{m_x} \\ \delta d_{m_y} \\ \delta d_{m_z} \end{bmatrix} \tag{3.1}$$

Computation of the initialization

The initial position and velocity are given by GPS measurements, although it could be also specified by the user (e.g. $v = 0$). The matrix $\mathbf{R}_n^b$ is then computed in order to align the body frame with respect to the navigation frame. The conventional coarse alignment method can not be applied because of the MEMS-IMUs noise level and error characteristics do not allow sensing the Earth's rotation rate [20].

A modified coarse alignment and a quaternion based algorithm have been implemented [20] to exploit the magnetometers for the computation of $\mathbf{R}_n^b$. The main idea is to compare measurements performed in the body frame with the nominal values known in the navigation frame. Only the modified coarse alignment method will be detailed in this section. Theoretical bases on the quaternion estimation algorithm can be found in the work of Bar-itzhack, Psiaki and Waegli ([1, 10, 20]).

The aim of the modified coarse alignment is to compute the $\mathbf{R}_n^b$ matrix without gyroscopic data. However, three known vectors are still required in each frame. The magnetic measurements and the knowledge of the magnetic field at a given location at the surface of the Earth constitue the first pair of vectors. The specific force measurements and the gravity vector form a second one. The last vector is obtained as a crossproduct computed for each frame. The following relation (3.2) shows how the rotation matrix is obtained:

$$\begin{bmatrix} -\mathbf{f}^{\,b} & \mathbf{m}^b & -\mathbf{f}^{\,b} \times \mathbf{m}^b \end{bmatrix} = \mathbf{R}_n^b \begin{bmatrix} \bar{\mathbf{g}}^n & \mathbf{m}^n & \bar{\mathbf{g}}^n \times \mathbf{m}^n \end{bmatrix} \tag{3.2}$$

where $\mathbf{f}^{\,b}$ is the specific force vector, $\mathbf{m}^b$ the magnetic measurement, $\bar{\mathbf{g}}^n$ the normal gravity vector and $\mathbf{m}^n$ the Earth magnetic field deduced from a worldwide model. Note that $\mathbf{f}^{\,b}$ and $\mathbf{m}^b$ can be averaged during static periods [20].

Strapdown inertial navigation

Once the initialization is completed, the strapdown navigation can be started. This technique of navigation is autonomous and thus requires only IMU measurements. No external perturbations (e.g. weather conditions, topographic constraints) can affect the process. Strapdown inertial navigation is composed of three main steps [20]:

- The **orientation prediction**, which propagates the orientation state of the sensor through time.

- The **velocity prediction** with respect to the navigation frame. This value is integrated from the specific force measurement after several corrections.

- The **position prediction by integration of velocity prediction**.

Update with the measurements

Once GPS measurements are available, the update step can take place. In this project, the GPS and INS data are assembled in a loosely coupled approach. Explicitly, GPS coordinates, velocities and sometimes also magnetic measurements are fed in the EKF as external aiding. However, before these data can be used in the filter, they must be brought into the IMU reference system beforehand. This is done by the application of the lever-arm vector that represents the spatial displacement between the center of the IMU and the phase center of the GPS antenna. Fig. 3.3 summarizes the different steps of the loosely coupled integration of GPS/IMU data.

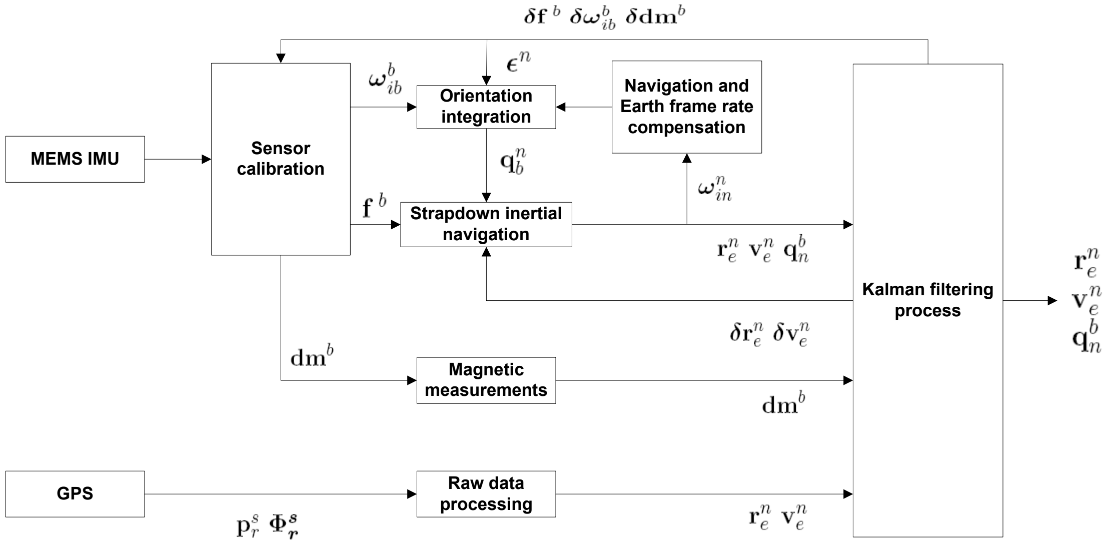

$\delta\mathbf{f}^b$ $\delta\boldsymbol{\omega}_{ib}^b$ $\delta\mathbf{dm}^b$
$\boldsymbol{\epsilon}^n$
$\boldsymbol{\omega}_{ib}^b$
Orientation integration
Navigation and Earth frame rate compensation
MEMS IMU
Sensor calibration
$\mathbf{q}_b^n$
$\boldsymbol{\omega}_{in}^n$
$\mathbf{f}^b$
Strapdown inertial navigation
$\mathbf{r}_e^n$ $\mathbf{v}_e^n$ $\mathbf{q}_n^b$
Kalman filtering process
$\mathbf{r}_e^n$
$\mathbf{v}_e^n$
$\mathbf{q}_n^b$
$\delta\mathbf{r}_e^n$ $\delta\mathbf{v}_e^n$
$\mathbf{dm}^b$
Magnetic measurements
$\mathbf{dm}^b$
GPS
Raw data processing
$\mathbf{p}_r^s$ $\boldsymbol{\Phi}_r^s$
$\mathbf{r}_e^n$ $\mathbf{v}_e^n$

Fig. 3.3: Schematic view of the GPS/IMU loosely coupled integration process. Adapted from [20]

Finally, it is important to note that the integration process can be achieved in a forward sense. In this case, the computation is done in the same direction as time. In case of backward integration, the process starts at the end of the time vector and evolves back in time towards the start. The integration can also be done in both directions, subsequently.

3.3 Previous software structure

Before modifying the software, it is necessary to understand how it works by mapping its structure. This section aims to describe the structure of the software developed by Adrian Waegli [20]. All the steps of the GPS/INS integration described in section 3.2 are implemented with numerous options and parameters. The main program is autonomous and renders a trajectory, starting from raw measurements of the inertial and GPS sensors. The global structure of this software is depicted in Fig. 3.4. More detailed flow charts are given in appendix A.

The first step consists in the creation of a configuration file which contains all the process parameters. This file is loaded and enables the trajectory's computation. The input files (GPS and IMU) are then loaded into memory. The user has to delimit the processing range as well as to specify the initialization periods (for the forward and backward processing) by means of a GUI. Thereafter, the GPS/INS integration process begins for the specified processing range.

The calculated initial orientation is assigned to the first state vector (section 3.2.2). Based on this knowledge, predictions of the next state vectors are made using a dynamic model and its' associated noise characteristics. The evolution of

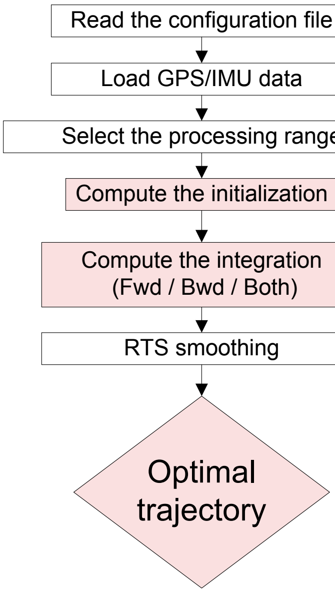


Fig. 3.4: Old structure of the software.

the PVA is thereby followed until GPS measurements are available and the update with the EKF can be made. This process is repeated through the entire processing range with a computation rate specified by the user (in the configuration file). It can be processed in a forward, backward or even bidirectional manner.

Finally, the software executes a fixed-interval smoothing by Rauch-Tung-Striebel (RTS) method to obtain a final trajectory. After this last step, several plots and saving modes become available to represent and analyse the trajectory.

3.3.1 Limitation of the software

The structure of the software developed by A. Waegli is strictly organized and works efficiently. Limitations of the software are that only one processing range can be treated per configuration file and this range has to be identified manually by the user. Moreover, during the intermediate static phases which can occur if the athlete stops, it would be better to recompute an initial vector rather than to continue the integration process. This is principally to avoid the divergences of the EKF (section 1.2).

3.4 Proposed modifications

In order to treat several dynamic ranges together, the software has to be restructured. The entire process is split into three distinct steps which are depicted in Fig. 3.5. A more detailed view of this new architecture is given in appendix B.

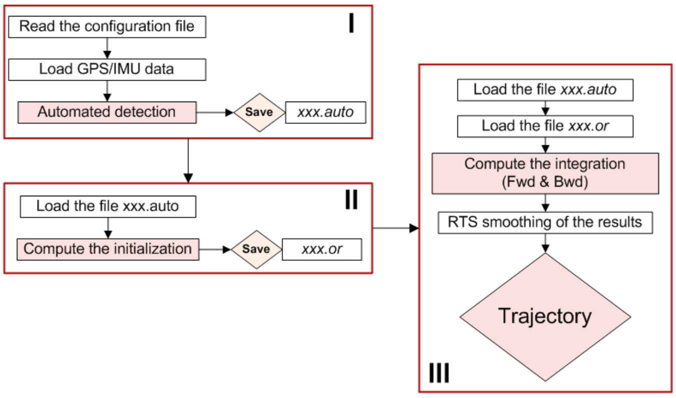


Fig. 3.5: New structure for the automated software.

During the first step, the configuration file, the GPS and the IMU data are loaded. Then, an automated detection of the dynamic ranges (Chapter 2) is done. The results are saved in a text file (Tab.2.3), and the software stops.

The second part of the treatment begins with loading the previously identified dynamic ranges. Based on those values, the initial orientation is computed for each dynamic range and saved in an individual state vector. All the results of the initialization phase are then saved in an orientation file. The software execution stops again.

The third part of the process loads the results from the previous two steps, and computes the integration in both directions per interval. This yields a forward and a backward trajectory for each dynamic range. They are then smoothed with a fixed-interval algorithm and the resulting data are finally concatenated in a unique trajectory vector. This is performed with the goal to generate the same output as the one produced in A. Waegli's software. Hence, the same plotting and saving tools can be used to represent and analyse the trajectory.

When this new software structure is executed, the user can choose to compute each step separately or to run them all in an automatic way. This choice is specified at the beginning as a parameter in the configuration file. This option enables the user to apply modifications between the three processing steps, should it be necessary.

Chapter 4

Practical experiments

This chapter describes the ski and bike experiments together with the used instrumentation.

4.1 Selection of the equipment

Several practical tests were performed to collect data for the evaluation of the new software's architecture. This section briefly presents the material used, before describing the experiments themselves.

For the data acquisition, two MEMS-IMUs were mounted in parallel on a rigid aluminium structure. All parts of the support and the screws are non magnetic to avoid disturbing the magnetometers included in one of the IMUs. A dual-frequency (L1/L2) GPS antenna is also fixed on the aluminium plate, above and close to the IMUs. As all sensors are tightly fixed together it is possible to determine the lever arms between their respective centers. This is important for integrating all the data at the same point in space (e.g. at the center of one IMU).

As shown in Fig. 4.1, the sensors are attached to the athlete's waist by a small bag. A second backpack contains the computer, the GPS receiver and the power for one of the IMUs. To summarize, we can say that three sensors are used:

- The Crista[1] IMU is a new sensor which is tested in this projet. It contains three accelerometers, three gyroscopes and obtains precise GPS time with a Pulse Per Second (PPS) coming out from the dual frequency GPS receiver. The sampling rate of this sensor is 100 Hz.
- The MTi-G[2] IMU contains three accelerometers, three gyroscopes, three magnetometers as well as a single-frequency GPS receiver which uses a small L1

[1] by Cloud Cap Technology (http://www.cloudcaptech.com).

[2] by Xsens (http://www.xsens.com)

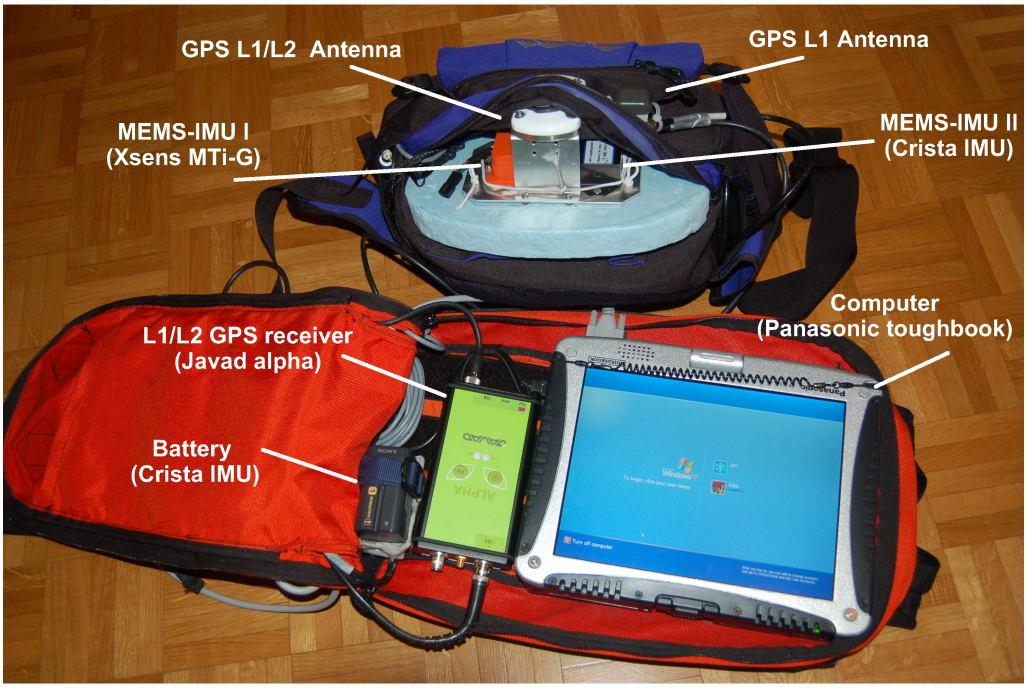


Fig. 4.1: Experimental setup mounted in a backpack.

antenna depicted on Fig. 4.1. For this application, this GPS receiver is only used for obtaining the GPS time for synchronizing the inertial measurements. Similarly to the Crista IMU, the sampling rate of the Xsens IMU is 100 Hz.

- The Javad[3] GPS receiver provides the single-frequency GPS data for the "low-cost" solution, but also the L1/L2 data necessary to build the reference trajectory of higher precision. The latter is crucial for the evaluation of the performances of the "low-cost" system. The selected sampling rate of the GPS receiver is 10 Hz.

Both IMUs are used, because the most recent one has to be tested. The above former is able to obtain a PPS from the GPS dual-frequency receiver, enabling an accurate synchronization of the inertial measurements with the GPS data. However, the Crista IMU doesn't have magnetometers, which are used for the initialization. Thus, the Xsens IMU is still used during the static periods.

Fig. 4.2 presents how the sensors are connected during the practical testing. The computer logs in real time the data from the both IMUs, through a USB or COM1 port. It also provides power for the Xsens MTi-G. The Crista IMU runs on a battery and gets the PPS from the GPS receiver above one pin in its serial connector.

[3] by Xsens (http://www.javad.com)

The dual frequency GPS receiver logs the data in its internal memory and sends the PPS to the Crista IMU. Finally, a second dual frequency GPS receiver is placed in the area of the testing as a base-station for the Differential GPS positioning (DGPS). The base-station data are also logged in the internal memory of the receiver.

All these data, from the various sensors, are finally downloaded and postprocessed to obtain a reconstruction of the athlete's trajectory.

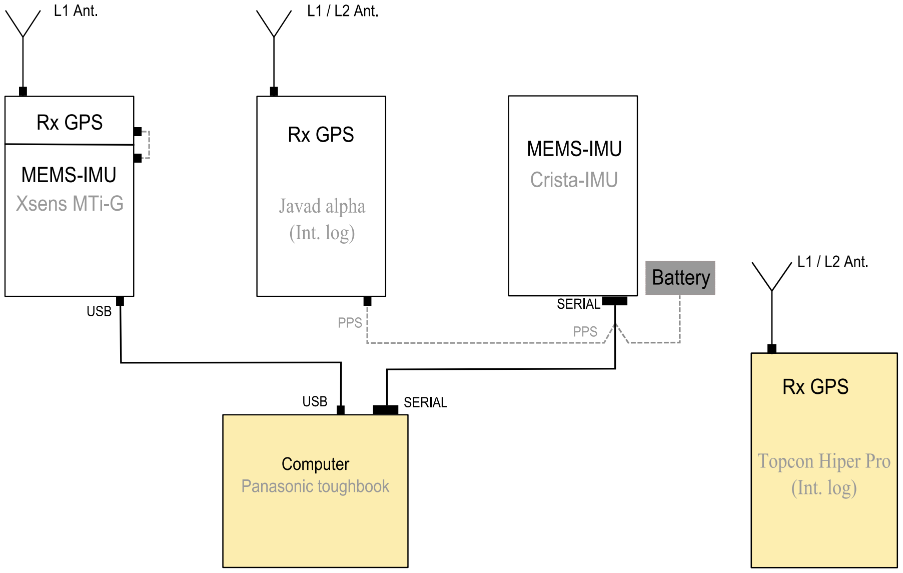


Fig. 4.2: Connections between sensors.

4.2 Ski experiments

4.2.1 Description of the practical tests

The first practical tests were done with a skier (Fig. 4.3) . The selected slope in Haute-Nendaz (Switzerland) is north-faced and not too steep. These conditions where chosen in order to optimize the reception of the GPS signals. Indeed, the GPS antennas are mounted in the back of the skier, thus they are primarily facing southwards during the athlete's descent.

The base station was set on the top off the hill near the slope's summit. With this configuration, the largest separation between the reference and the rover receiver (the athlete) was shorter than 600 m. A ski run composed of twelve gates was completed eight times. It should be noted that the number of visible satellites (SV) varied during the day. For the best periods, it reached a mean of 7 SV per run.

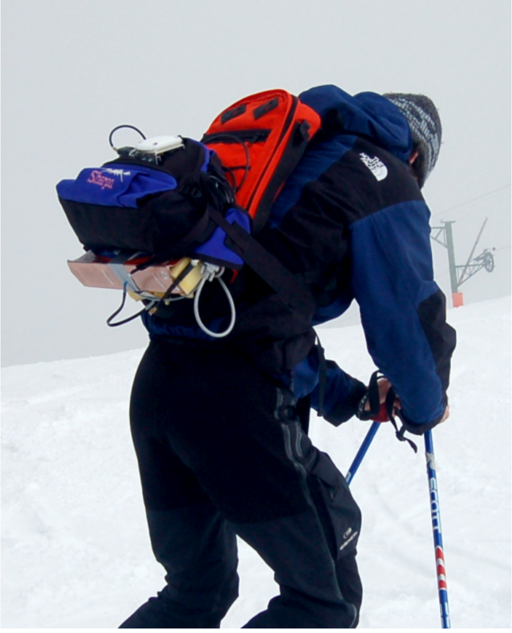


Fig. 4.3: Experimental setup on the back of a skier.

4.2.2 Discussion

It appeared that the entire system does not interfere with the athlete's freedom of movement, although its size could be substantially reduced in "customized" product. Unfortunately, a technical problem was encountered during these tests. The dual frequency GPS antenna, which was mounted on the aluminium structure with the two IMUs, did not work correctly. Indeed, the reception of the signal was weaker than normal, which affected the reception of the second GPS frequency (L2). The result is a lack of L2 data that compromises the accurate determination of the reference trajectory.

Subsequent tests in the office have shown that the antenna has a smaller signal-to-noise ratio (SNR) than a "healthy" antenna under the same conditions. This is illustrated in Tab. 4.1. The reference antenna is slightly bigger and was used for previous experiments with an airborne mapping system.
Tab. 4.1 clearly shows that the SNR of the antenna used for the ski practical tests, is lower than the SNR of the reference antenna. Moreover, for the same satellite (PRN) tracked at the same elevation, the value of the SNR for L2 is systematically lower for the antenna used with the skier. This demonstrates that the lack of L2 frequency is due to the fact that the antenna is defective.

For this reason, the data of the ski experiment are not fully exploitable. Without the second GPS frequency, the reference trajectory cannot be computed and thus only a relative comparison between several L1 trajectories can be carried out. However, this first practical testing was helpful to identify potential problems for

Tab. 4.1: Mean values of SNR on L1 and on L2 GPS frequencies, for a reference antenna and for the antenna used in the ski tests.

Reference antenna											
*****************											
PRN	126	124	120	32	28	23	20	19	11	3	14
Elevation [°]	34	27	32	78	19	5	46	36	80	8	32
L1-SNR	39.1	19.9	40.1	45.9	37.4	33.4	43.0	43.7	44.9	37.5	6.1
L2-SNR	0	0	0	**37.1**	16.8	17.5	**34.4**	**35.6**	**37.5**	20.7	2.0
Ski experiments antenna											
**********************											
PRN	126	124	120	32	28	23	20	19	11	3	
Elevation [°]	34	27	32	79	18	7	47	34	82	7	
L1-SNR	8.1	7.0	36.1	45.5	39.8	40.9	45.6	39.0	43.7	40.7	
L2-SNR	0	0	0	29.9	13.7	17.1	28.7	19.4	**31.0**	14.9	

the following experiments. The antenna was changed to avoid future problems with the reception of the L2 frequency.

4.3 MTB experiments

In order to have new data sets to evaluate the performances of the new software's architecture, two more practical tests were completed with a MTB. The same configuration of the sensors was used for these experiments, only the dual frequency GPS antenna was changed (section 4.2.2). Fig. 4.4 shows the biker with the sensors mounted in the bags on his back.

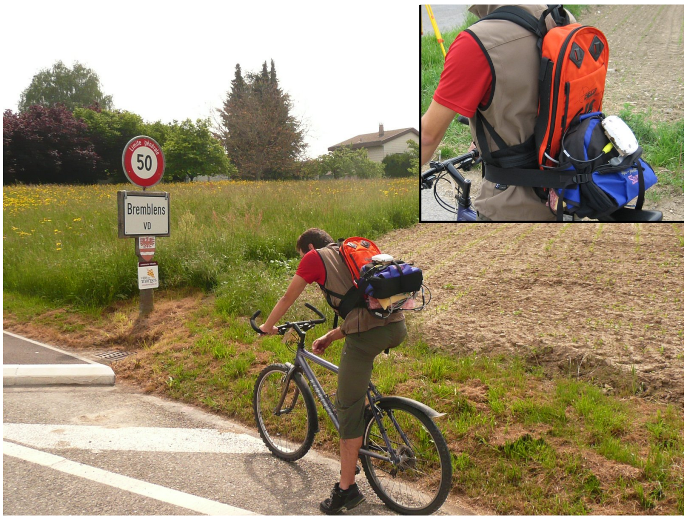


Fig. 4.4: Experimental setup on the back of a biker.

4.3.1 Location 1: Saint-Saphorin sur Morges

Test description

The tests were performed on a road between Saint-Saphorin sur Morges and Bremblens (Switzerland). The road faces east and the surrounding topography offers good satellite visibility. Two small forests on the side of the road cause gaps in the GPS signal reception with maximum duration of ten seconds. This can be overcome by the IMU's measurements. The road was biked six times with an average visibility of 8 SV. The biker made several stops along the track to simulate quasi static states, which are later used to test the capacities of the algorithm detecting dynamic ranges.

The base-station was set at one end of the road with an excellent satellite coverage (mean of 13 SV). The largest baseline lengths was 1.25 Km.

Discussion

The first two run were done with a sufficient GPS coverage. Both frequencies (L1/L2) have been regularly sensed, confirming the antenna defectively during the ski experiments.

During the third run, the computer entered a standby mode, which generated a conflict with the real time logging of the Xsens inertial data. Thereafter, runs were logged using the default mode of the IMU software. These data are unfortunately unexploitable. However, the first two tracks can be used for the evaluation of the new software's capabilities.

4.3.2 Location 2: UNIL's sports field

Test description

This test was carried out on an open sports field with nearly no satellite masking above 20°. The runs were shorter and included at least one stop, with an average of 6 SV visible for the rover receiver. For this experiment no path was imposed and the biker made random curves.

The base-station was set at 1.2 Km of the sports field. With an average of 10 SV, the satellite coverage was good.

Discussion

Seven runs were completed without any complication. The satellite constellation was slightly poorer than planned, thus it is hard to have a fixed solution when computing the reference trajectory for some runs. The runs six and seven are used in the following sections for the evaluation of the performances of the new software.

Chapter 5

Performances evaluation

This chapter presents the evaluation of the new software's capabilities for automatized processing. First, the new software functionality will be validated. Then, the performances and the merits of the automatized approach will be assessed using trajectory comparisons as well as simulations of GPS outages.

5.1 Preliminary note

To reconstitute the athlete's trajectory, GPS observations are integrated with the MEMS-IMU measurements in loosely-coupled mode. Hence, before the integration, the GPS data are post-processed for obtaining positions and velocities. Three carrier-phase ambiguities processing approaches are available (i.e. fixed, float or carrier-phase smoothed pseudoranges):

- The reference trajectories are computed with the dual-frequency GPS measurements, using commercial software package (Novatel-Waypoint[1]). Under good measurement conditions, the carrier-phase ambiguities are fixed during 95 % of the time [20], resulting in an accurate GPS trajectory ($1\sigma < 10cm$).

- Single-frequency GPS measurements are used to compute "low-cost" GPS trajectories. This treatment can be achieved by using commercial software packages or an algorithm implemented by A. Waegli in the frame of his PhD thesis. The main difference between this two methods stands in the way they use the single-frequency data. The commercial solution considers float ambiguities.

[1] http://www.waypnt.com

- The algorithm presented in [20] uses the carrier-phase to smooth the code pseudoranges and do not attempt to solve the ambiguities. Practical tests have shown that the accuracy is improved by 30 - 50% when float ambiguities are estimated instead of using carrier-phase smoothed pseudoranges [20].

In the frame of this project, the reference as well as the single-frequency solutions are computed with a commercial software package. Moreover, the single-frequency measurements are taken from a geodetic receiver. Therefore, the resulting positions are most likely accurate than those derived from a "low-cost" GPS receiver. In other words, the so called "low-cost trajectories" in the next section, are expected to be more accurate than a "real low-cost" solution.

5.2 Validation of the new software's architecture

5.2.1 Principles

The main objective of this section is the validation of the new software's architecture. This step is crucial to ensure that the integration process works correctly before evaluating its performances.

The validation is based on the comparison of trajectories. The trajectory comparison software developed by A.Waegli [20] is used for this purpose. This needs as an input a reference trajectory and at least one other trajectory to be compared with. For the purpose of validation, the reference trajectory is computed with the old software's architecture. The same input data are then processed with the new software's version. That gives a second trajectory, which is confronted with the former. The validation will be qualified as successful if the differences between the two trajectories are negligible. Indeed, this means that both versions of the software lead to the same results and that the new software works correctly. Fig. 5.1 summarizes the principles of this validation procedure.

5.2.2 Results and consequences

Several data sets of various types (motorcycle, MTB and ski) were processed for the purpose of validation. For each of them, two trajectories were obtained and confronted. The differences in terms of position and velocity were saved and analyzed.

The different characteristics, i.e. the mean, the median, the standard deviation (SD) and the robust InterQuartile Range (IQR), in Tab. 5.1 represent the offsets on the east coordinates (δE), the north coordinates (δN), the altitude (δh) and the lateral displacement (δLD). Then, the differences for the vertical velocity (δvh) as well as the 3D-velocity (δv 3D) are given. Since the IQR is a less usual statistic estimator, it is presented in detail in appendix C.

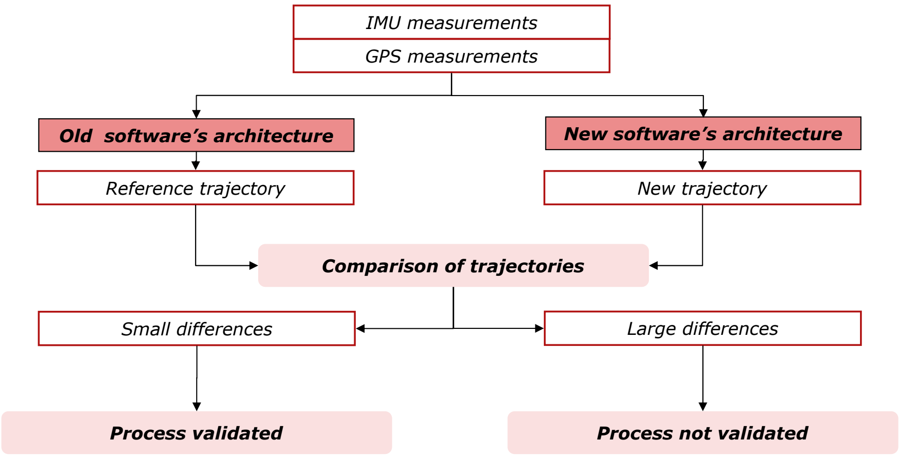


Fig. 5.1: Flow chart of the validation process.

It appears clearly that the differences between the reference trajectory and the one computed with the automatized software are very small. Indeed, the differences in terms of position have a subcentimetric mean value, with a standard deviation smaller than two centimeters. The mean difference for the 3D velocity (0.004 m/s) and its relative SD (0.052 m/s), are small compared to the threshold value used earlier for the detection of the dynamic periods (0.5 m/s).

The results obtained with the automatized version of the GPS/INS software are very close to those of its predecessor. These observations lead to the conclusion that the new software works correctly and is therefore validated for the following analyses.

Tab. 5.1: Comparisons of trajectories computed with the old as well as the new software's architecture.

		Mean	**Median**	**SD**	**IQR**
δ**E**	[m]	0.001	0.001	0.014	0.003
δ**N**	[m]	-0.003	-0.001	0.010	0.004
δ**h**	[m]	0.005	0.005	0.006	0.002
δ**LD**	[m]	0.008	0.005	0.018	0.005
δ**vh**	[m/s]	-0.001	-0.001	0.017	0.008
δ**v 3D**	[m/s]	0.004	0.005	0.052	0.021

5.3 Performances of the GPS/INS integration

The automatized processing was implemented in order to reconstruct the athletes trajectories. For sport applications characterized by a relatively short duration (e.g. ski competition), the parameter estimation by EKF used in the GPS/INS integration must converge quickly. This means that the position, velocity and orientation accuracy must increase rapidly after the start of the athlete, while the bias terms for the gyroscopes and the accelerometers converge with the variations of dynamic. Actually, the change of orientation and the variation of acceleration enable the decorrelation of the biases terms from the errors in attitude initialization.

Fig. 5.2 shows that the orientation accuracy increases rapidly after the start of the skier. It appears clearly that this increase is slower for the heading term. This is due to the limited observability of the heading angle, under limited motion [20]. The convergence of the bias terms is depicted in Fig. 5.3. It appears that the implemented EKF filter works correctly and converges quickly, in less than 15 s. Since the integration is performed automatically in both directions, the trajectory can be smoothed with a fixed interval smoothing (i.e. RTS method). The benefits of this process for the orientation accuracy are represented in Fig. 5.4. They are more significant towards the processing starts, due to the optimal combination of the forward and the backward solutions.

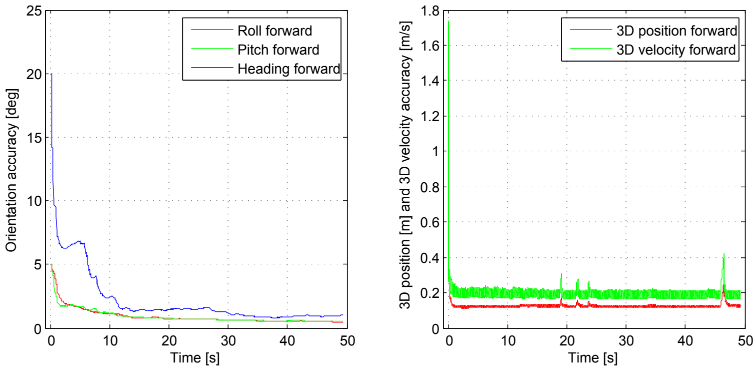


Fig. 5.2: Orientation accuracy (left) and 3D position and 3D velocity accuracies (right) in a ski experiment.

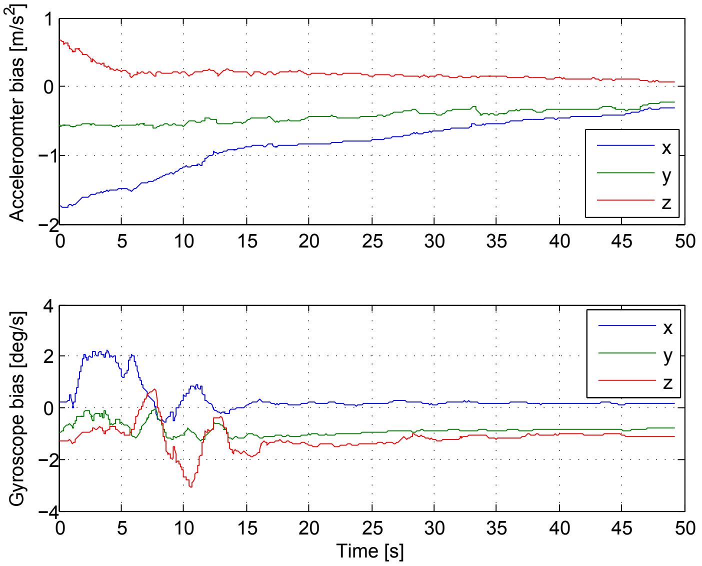


Fig. 5.3: Convergence of the accelerometer and gyroscope biases after the start of the skier.

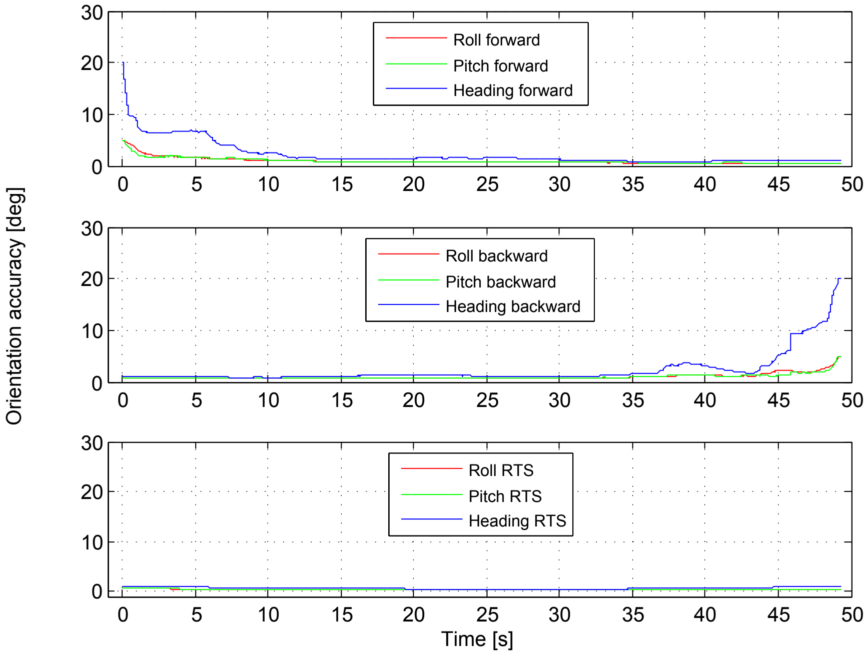


Fig. 5.4: Benefits of the fixed interval smoothing. The figure represents the orientation accuracy calculated in forward direction (top), backward direction (middle) and smoothed (bottom).

5.4 Evaluation of the software's performances

5.4.1 Evaluation of "low-cost" L1 trajectories

A "low-cost" trajectory is understood when only the single-frequency GPS measurements are used with the MEMS-IMU data during the GPS/INS integration. For evaluation purposes, a reference trajectory is also computed with the L1 and L2 GPS frequencies. These two trajectories ("low-cost" and reference) are then compared with the trajectory comparison software developed by A.Waegli [20]. For each experiment, two trajectories are thus processed.

The aim of this evaluation is to quantify the deviations between a "low-cost" trajectory and a trajectory obtained with a dual-frequency GPS receiver. Small differences mean that the automatized reconstruction of trajectories using MEMS-IMU and a single-frequency GPS receiver is reasonable for sport applications. As in section 5.2, the position and the velocity differences are obtained and analyzed. The same statistical estimators are used. Several comparisons were computed, based on the measurements collected during the MTB experiments. A summary of these results is given in Tab. 5.2.

Tab. 5.2: Comparisons of "low-cost" and reference trajectories, computed with the automatized integration software, using MTB experiments.

		Mean	**Median**	**SD**	**IQR**
δ**E**	[m]	0.113	0.095	0.194	0.213
δ**N**	[m]	0.161	0.210	0.253	0.343
δ**h**	[m]	0.183	0.186	0.101	0.110
δ**LD**	[m]	0.333	0.352	0.199	0.277
δ**vh**	[m/s]	0.002	0.006	0.209	0.171
δ**v 3D**	[m/s]	-0.004	-0.001	0.135	0.082

The mean values of the position differences between the reference and the tested trajectory range between 0.1 m and 0.33 m. The associated SD is between 0.1 m and 0.25 m and the robust IQR between 0.11 m and 0.34 m . The previous study concluded that the position accuracy achieved with the integrated system of this type varies from decimeter level (dual-frequency DGPS) to 0.7 m ("low-cost" L1 DGPS) [20]. The differences mean values for the vertical and 3D velocities are even smaller ($\delta\mathbf{v3D} = -0.004\,\mathrm{m/s}$) with a SD between 0.13 m and 0.21 m. Thus, the obtained differences depicted in Tab. 5.2 are not significant.

Fig. 5.5 presents the histograms of the differences calculated between a "low-cost" and a reference trajectory. The trajectory represents a bike run of 4 minutes with three stops. The lateral displacement and the altitude differences are slightly skewed. This could mean, that one trajectory is higher than the other one during the whole run. Thus, more attention should be given to the spreading of the data rather than to the absolute mean value. The same goes for the lateral displacement.

Finally, these results lead to the conclusion that the automatized software works correctly, even if only the single-frequency GPS measurements are used.

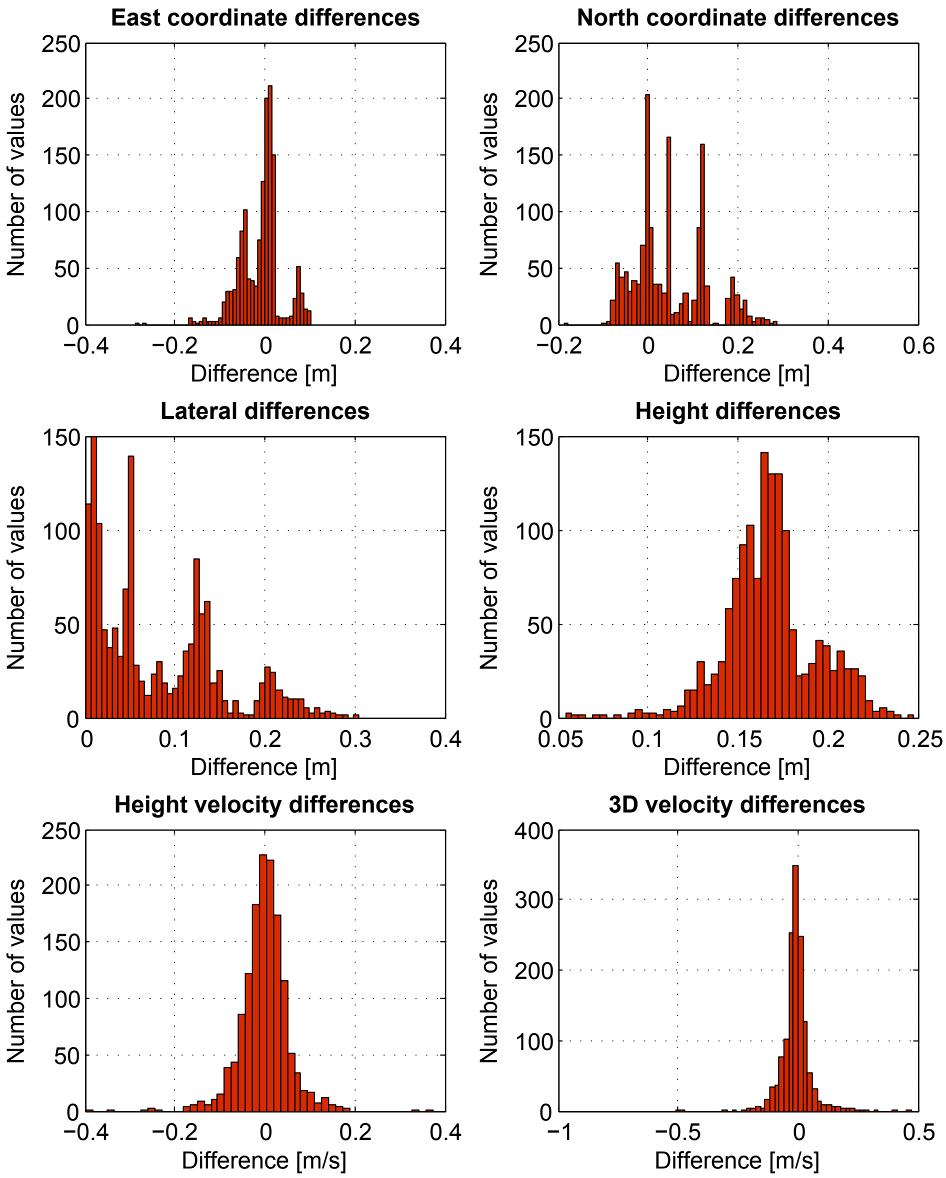


Fig. 5.5: Histograms of the differences between a "low-cost" and a reference trajectory. These data represent a 4 minutes-long bike run with 3 stops.

5.4.2 Trajectory comparisons

Now, once the functionnality of the new software's architecture has been proven, it is interesting to compare several ski descents. The fastest trajectory calculated with the single-frequency GPS measurements is set as reference. Two other downhills are then confronted to the reference trajectory. This analysis highlights the differences in terms of trajectories, and thus for example, the repercussions on the athlete's velocity. The planimetric and the 3D velocity comparison between these three descents are presented in Fig. 5.6.

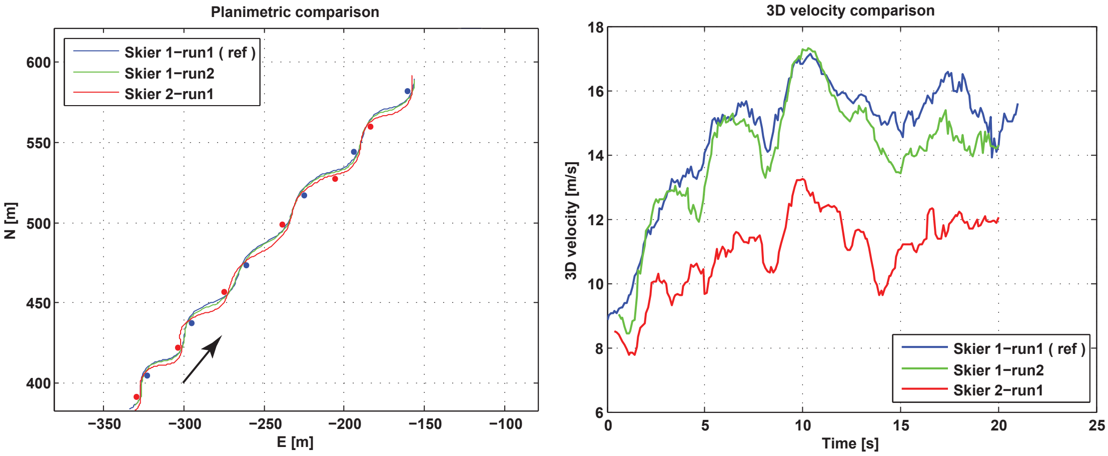


Fig. 5.6: Planimetric comparison (left) and 3D velocity comparison (right) of three ski runs done by two skiers.

The second skier (red) has adopted an anticipative strategy, thus he turns before the gate. As consequence his turns are more pronounced and the path is longer. By opposite, the first skier (blue and green) took more direct trajectories. This strategy is less anticipative, thus the turns are sometimes finished after the gate.

The 3D velocity comparison shows that the fastest strategy is the direct one, even if this is not optimal from the theoretical point of view. Indeed, professional skiers carve at the location of the gate in order to accelerate at the end of the turn, without compromising catching the next gate. Fig. 5.7 shows the 3D velocity differences as well as the East and North coordinate differences between the reference and the compared trajectories. These graphs enable to detect if the skier is faster or slower than the reference athlete, and to locate in time the largest position differences.

A more complete analysis is certainly possible with this application in order to understand why one descent was better than the other. However, this is not the main interest of this project. Indeed, this section aims to illustrate the capabilities provided by the GPS/INS technology and the software implemented or used for this project.

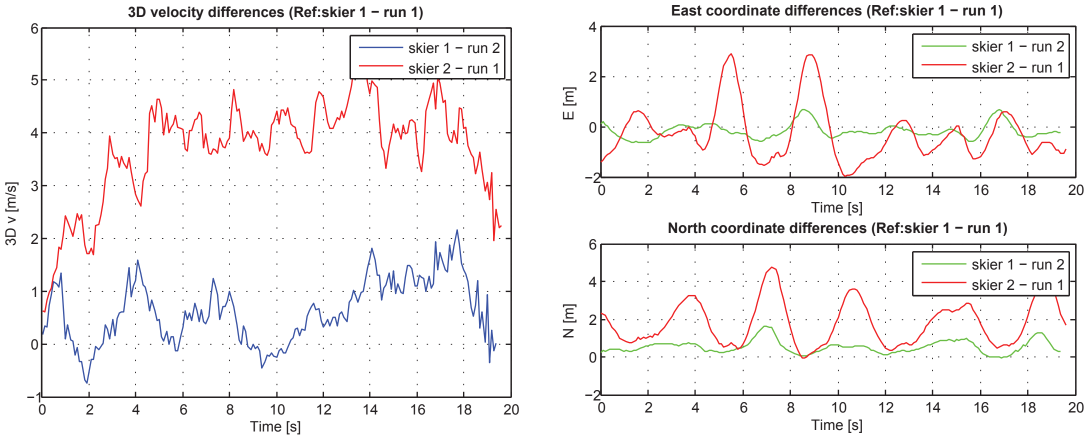


Fig. 5.7: 3D velocity differences (left) and the East and North coordinate differences (right) between the reference and the compared trajectories. A negative velocity difference means that the run is faster than the reference one.

5.4.3 Simulation of GPS outages

This section presents the results of GPS outages simulation. As seen in section 1.1, the IMU measurements are integrated with the single-frequency GPS data in order to bridge possible lacks in the GPS signal reception. To evaluate the performances of the integrated inertial navigation system, GPS outages of various duration are simulated. This was performed with the data collected during a ski run (skier1 - run1 (ref), Fig. 5.6). The same trajectory is computed with gaps duration of 2 s, 5 s, 10 s and 15 s in the GPS measurements. The "low-cost" trajectory computed without any outages is considered as a reference in the comparison.

Fig. 5.8 shows that the heading accuracy is more affected by lack of GPS updates than the roll and pitch accuracies. This could be caused by the fact that the observability of misalignment in heading is limited. Thus, when the outages begin (t = 13.4 s), the heading misalignment is not as well determined as that in roll and pitch. Its dependence on dynamic then generates a faster degradation of the heading accuracy during the outage.

Outages of 2 s and 5 s durations do not imply large variations in the orientation accuracy. Moreover, thanks to the RTS smoothing, the accuracy is refined rapidly at the end of the outage, and reaches the level obtained without the gaps in the GPS signal. For longer outages (10 s and 15 s), the orientation accuracy differences

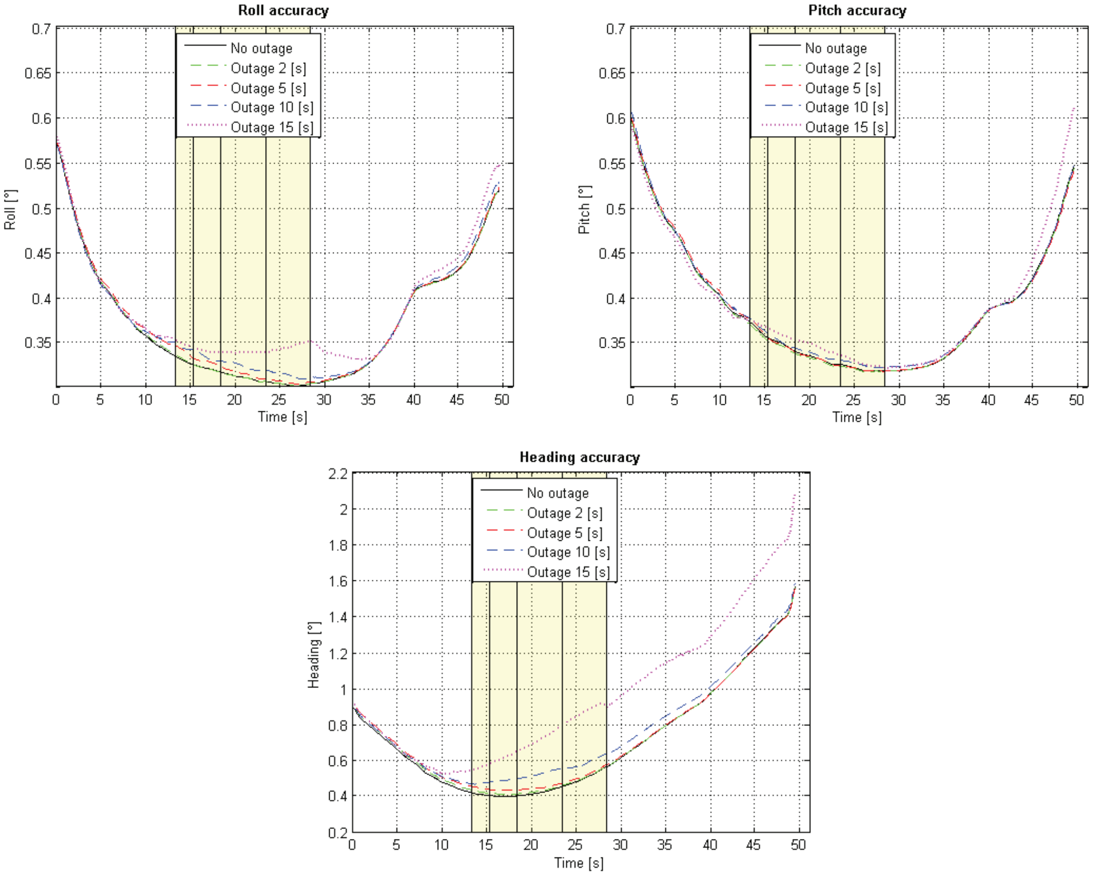


Fig. 5.8: Evolution of the RTS smoothed orientation accuracy, as a function of GPS signal outages, in ski experiment. The vertical lines represent the simulated lacks of GPS updates.

are larger. The gap which lasting 15 s makes the heading accuracy decreasing since the beginning of the outage, even when the RTS smoothing is applied.

The relatively short duration of the experiment implies a decrease of the orientation accuracy for each angle after 30 s. This is the consequence of the RTS smoothing, which combines the "low" accuracy of the backward solution at this portion of the data set, with the forward "high" accuracy (see Fig. 5.4).

Tab.5.3 presents the differences in terms of position and velocity, resulting from the comparison of "low-cost" trajectories computed with the various GPS outages. As in section 5.2, the differences are expressed using statistical estimators (i.e. mean, median, SD, IQR). The maximum difference is also listed. Note that these estimators are computed using only the trajectory differences during the outages.

Outages of short duration (2 s and 5 s) are well bridged by the inertial navigation system. Indeed, the maximum position differences range from 0.05 m to 0.8 m

Tab. 5.3: Comparison of "low-cost"' trajectories computed with GPS outages of various duration in the ski experiment. Gaps of 2 s, 5 s, 10 s and 15 s are simulated in the GPS measurements.

Outage 2[s]		**Max.**	**Mean**	**Median**	**SD**	**IQR**
$\delta\mathbf{E}$	[m]	0.133	0.068	0.071	0.038	0.035
$\delta\mathbf{N}$	[m]	0.038	-0.029	-0.030	0.037	0.044
$\delta\mathbf{h}$	[m]	0.164	0.078	0.086	0.056	0.076
$\delta\mathbf{LD}$	[m]	0.145	0.082	0.095	0.040	0.048
$\delta\mathbf{vh}$	[m/s]	0.026	-0.068	-0.057	0.073	0.074
$\delta\mathbf{v\ 3D}$	[m/s]	0.132	-0.018	-0.018	0.096	0.101

Outage 5[s]		**Max.**	**Mean**	**Median**	**SD**	**IQR**
$\delta\mathbf{E}$	[m]	0.553	0.254	0.311	0.258	0.321
$\delta\mathbf{N}$	[m]	0.158	-0.183	-0.086	0.238	0.250
$\delta\mathbf{h}$	[m]	0.84	0.347	0.274	0.248	0.270
$\delta\mathbf{LD}$	[m]	0.83	0.401	0.398	0.243	0.297
$\delta\mathbf{vh}$	[m/s]	0.471	0.072	0.073	0.216	0.272
$\delta\mathbf{v\ 3D}$	[m/s]	0.235	-0.315	-0.328	0.286	0.348

Outage 10[s]		**Max.**	**Mean**	**Median**	**SD**	**IQR**
$\delta\mathbf{E}$	[m]	2.815	1.419	1.468	0.818	0.888
$\delta\mathbf{N}$	[m]	-0.041	-1.414	-0.958	1.151	1.553
$\delta\mathbf{h}$	[m]	2.508	0.384	0.153	0.841	0.751
$\delta\mathbf{LD}$	[m]	4.112	2.073	2.035	1.306	1.761
$\delta\mathbf{vh}$	[m/s]	0.52	-0.203	-0.173	0.389	0.344
$\delta\mathbf{v\ 3D}$	[m/s]	2.021	0.596	0.737	0.663	0.786

Outage 15[s]		**Max.**	**Mean**	**Median**	**SD**	**IQR**
$\delta\mathbf{E}$	[m]	7.045	1.579	1.128	2.680	3.426
$\delta\mathbf{N}$	[m]	1.976	-1.543	-1.000	2.347	2.295
$\delta\mathbf{h}$	[m]	2.286	-0.629	-0.523	1.584	1.910
$\delta\mathbf{LD}$	[m]	8.754	3.444	2.690	2.380	2.622
$\delta\mathbf{vh}$	[m/s]	0.546	-1.383	-1.358	0.857	1.016
$\delta\mathbf{v\ 3D}$	[m/s]	5.372	0.960	0.751	2.343	3.059

depending on the analyzed parameter ((δE), (δN), (δh), (δLD)). The associated SD is smaller than 0.05 m for the outage of 2 s and ranges from 0.21 m to 0.28 m for the gaps of 5 s. The maximum velocity differences are lower than 0.5 m/s, which was also the dynamic range detection threshold used in the automated processing. Those values are of the same order of magnitude as the differences observed in the comparison between a dual and a single-frequency trajectory (Tab. 5.2). Moreover, the position accuracy of a "low-cost" trajectory is roughly 0.7 m [20]. These results show that the inertial navigation performed in the automatized software is able to bridge GPS outages of 5 s without compromising its accuracy.

Longer outages have more influence on the trajectories. Indeed the maximum lateral displacement difference for an outage of 10 s is 4 m. The mean differences are smaller and range from 0.38 m to 2 m. The velocity differences are lower than 1 m/s and the associated SD is smaller than 0.66 m/s. These results show that it is hard to compensate an outage longer than 10 s with the MEMS-IMU of this type without accuracy degradation. However, it should be kept in mind that the outage are simulated at time t = 13.4 s and that the orientation accuracy has just become reasonable at this time.

Fig. 5.9, shows the repercussions of the GPS outages on the trajectory by providing a planimetric and an altimetric comparison of the 5 "low-cost" trajectories. It appears clearly that small outages (2 s and 5 s) do not distort the trajectory significantly.

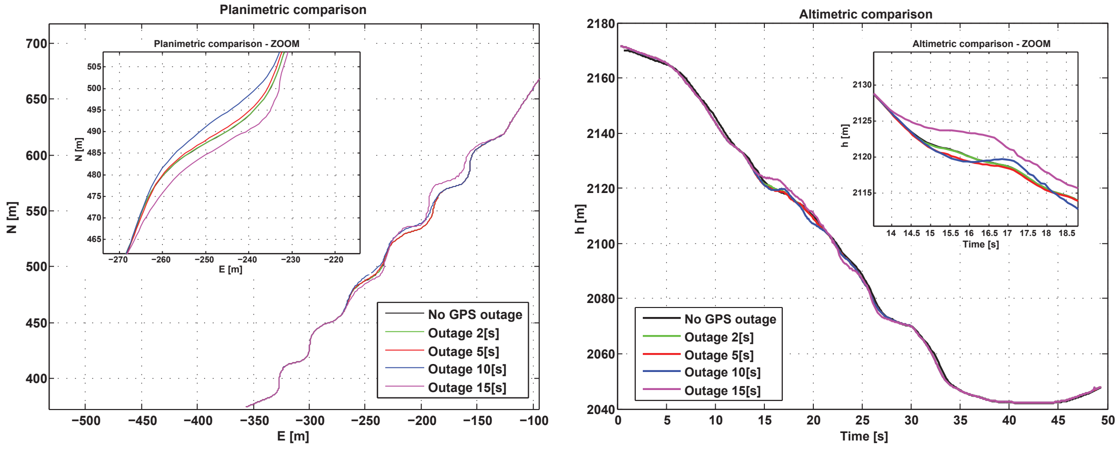


Fig. 5.9: Planimetric and altimetric comparison for GPS signal outages of 2 s, 5 s, 10 s and 15 s.

Outages of 15 s may generate even larger differences (maximum position difference of 8 m). That may not be suitable for athlete's performances analyse.

In summary we can say that the automatized software using strapdown inertial navigation works correctly. Its performance depends on the quality of the GPS and IMU observations as well as the dynamic of the trajectory. In the particular case of the ski experiment, the trajectory determination accuracy was satisfactory up to 10 s of GPS data absence, on contrary to the 15 s reported in [20].

5.5 Usage of the new Crista IMU

This section aims to evaluate the integration of the Crista IMU data. The comparisons between single and dual-frequencies trajectories as well as between "Xsens trajectories" and "Crista trajectories" are performed.

5.5.1 Evaluation of trajectories

The method used for this evaluation is similar to the one applied with the Xsens IMU measurements. The dual-frequency GPS data are used to generate a reference trajectory. The trajectory obtained with the single-frequency GPS measurements is then compared to the reference, leading to a set of position and velocity differences. A GPS update rate of 10 Hz was used to compute these trajectories. The obtained results are presented in Tab.5.4.

Tab. 5.4: Comparisons of "low-cost" and reference trajectories, computed with the automatized integration software. The inertial measurements are provided by the Crista IMU.

		Mean	**Median**	**SD**	**IQR**
δ**E**	[m]	0.128	0.087	0.159	0.176
δ**N**	[m]	0.009	0.000	0.136	0.157
δ**h**	[m]	0.231	0.253	0.196	0.272
δ**LD**	[m]	0.182	0.178	0.165	0.236
δ**vh**	[m/s]	0.003	0.000	0.250	0.083
δ**v 3D**	[m/s]	0.005	0.001	0.153	0.065

Tab.5.4 shows that the differences between the reference and the "low-cost" trajectory computed with the Crista IMU are small. These results are very closed to those obtained using the Xsens IMU (Tab. 5.2). The mean values of the position differences range between 0.01 m and 0.23 m with an associated SD smaller than 0.2 m. The differences for the velocities are also very close to what was previously obtained in Tab. 5.2. Thus, these differences are not significant. This means that computing trajectories using single-frequency GPS measurements together with the Crista IMU data is reasonable and works correctly.

5.5.2 Comparison to Xsens IMU

A comparison between "low-cost" trajectories computed with the Crista and the Xsens MEMS-IMU measurements on two bike runs, lead to the differences in terms of position and velocity listed in Tab. 5.5. The planimetric comparison shows that the results are very close, regardless the inertial sensor used. Indeed, Tab. 5.5 shows that the average position differences range between 0 m and 0.1 m. The mean differences for the velocities are even smaller with a maximum of 0.02 m/s. These results mean that the integration with the new sensor works correctly. However, it should be kept in mind that the results in terms of position and velocity are not very influenced by the choice of the MEMS-IMU hardware, if the GPS updates are

Tab. 5.5: Comparison of "low-cost" trajectories computed using the Crista and the Xsens inertial measurements. This differences are computed with the data from two bike runs.

		Mean	**Median**	**SD**	**IQR**
δ**E**	[m]	0.000	-0.012	0.072	0.096
δ**N**	[m]	0.060	0.077	0.044	0.036
δ**h**	[m]	0.010	0.008	0.015	0.015
δ**LD**	[m]	0.101	0.100	0.022	0.011
δ**vh**	[m/s]	0.020	0.002	0.162	0.055
δ**v 3D**	[m/s]	-0.013	0.023	0.230	0.113

available at a high rate (in this case 10 Hz). Indeed, a high GPS update rate will compensate potential orientation imprecision or sensor systematic error before its larger impact is observed. For this reason, an other analysis is presented.

5.5.3 Simulation of a GPS outage

This section aims to evaluate the "standalone" performances of the Crista MEMS-IMU by simulating a 10 s long GPS outage in the data of a ski run. During the period without GPS signal, no updates are performed. The resulting trajectory is built using only the inertial navigation. Therefore, the differences in terms of orientation misalignment and biases are well highlighted. In other words, if the orientation accuracy or the bias determination is not reasonable at the beginning of the outage, then the trajectory errors will increase rapidly.

Fig. 5.10 shows that the orientation accuracy for the Crista MEMS-IMU is comparable to the accuracy of the Xsens MEMS-IMU and to the reference computed without GPS outage. This is logical since the stochastic models are similar for both IMU's, and the same dynamic is encountered.

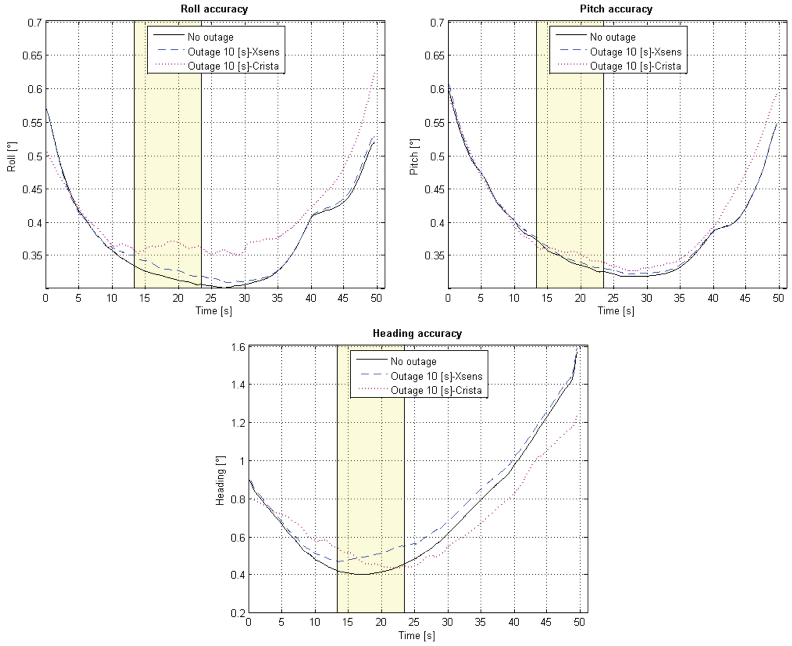


Fig. 5.10: Comparison of the orientation accuracy for the Crista and the Xsens IMU during a GPS outage lasting 10 s. The yellow bar represents the period with the lack of GPS measurement.

Fig. 5.11 represents the evolution of the Crista biases and their accuracy. It appears that the biases accuracy decreases in the backward integration. The biases on the accelerometers are large but converge towards a steady state. This is less apparent for the gyroscopic biases, which estimation is disturbed by the outage.

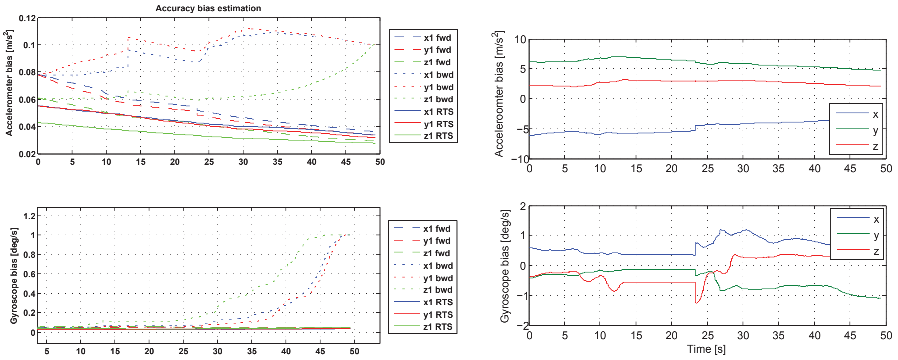


Fig. 5.11: Evolution of the Crista IMU biases (right) and of their respective accuracy (left).

In order to better evaluate the Crista IMU "standalone" performances, Fig. 5.12 represents the evolution of the planimetric errors with their associated accuracies (dashed red lines). The reference trajectory was computed using the Xsens inertial measurements, without gap in the GPS data set. Then a "Xsens trajectory" with a GPS gap during 10 s, as well as a "Crista trajectory"’ with the same lack of GPS data, are compared to the reference.

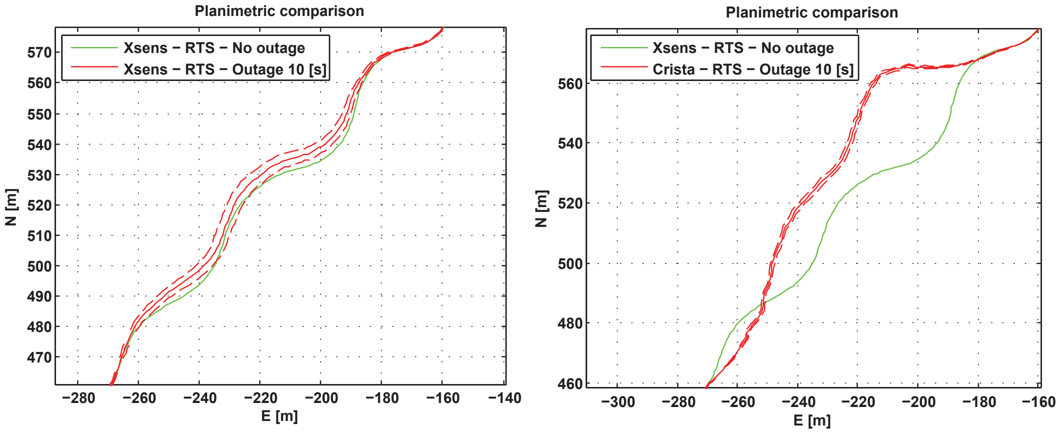


Fig. 5.12: Planimetric comparison of Xsens trajectories (left) and of Xsens and Crista trajectories (right). The reference is computed with the Xsens data, without GPS outage. The compared trajectories are computed with a GPS outage during 10 s. The dashed lines show the confidence level at 2.58σ (95%).

Fig. 5.12 shows that the trajectory integrated with the Xsens measurements and a GPS outage of 10 s duration is reliable. Indeed, the reference trajectory is close to or even included within the confidence level of 2.58σ of the tested trajectory during most of the GPS outage. On the other hand, the position error for the Crista trajectory is much larger than predicted by its confidence level. Thus, this trajectory is precise but not accurate. The large position errors can be explained by several factors:

- Usage of a too optimistic stochastic model.

- Presence of a large misalignment at the beginning of the GPS outage.

- Incorrectly estimated bias terms.

To enhance the navigation performance with the Crista MEMS-IMU, the sensor's stochastic model needs to be refined. Moreover, a specific strategy for the computation of the initial orientation and the initial approximation of the bias terms is needed.

Chapter 6

Conclusion and perspectives

This chapter provides a synthesis of the project and of its main results. Some perspectives for further research are also given.

Synthesis

In the first place, this project aimed to implement a flexible algorithm to detect automatically the dynamic ranges within MEMS-IMU/GPS signal. The chosen approach confronts the results of a CWT applied on the gyroscopes measurements with the smoothed GPS ground velocity, and highlights the dynamic periods of an experiment. From this part of the project, the following conclusions can be drawn:

- The proposed algorithm works correctly and provides an efficient detection of the dynamic parts of a studied signal.

- The CWT enables to adapt the detection criteria as a function of the input signal. This is not possible with a classical approach, which requires the user to tune the thresholding process according to the dynamic encountered during the experiment. The new algorithm overcomes this inconvenience, and thus represents a flexible tool across different applications.

- The selection of the frequency range to be analyzed enables working with noisy inertial signal. This is a great advantage for applications during which a parasitic signal (e.g. vibrations) interferes with the inertial measurements (e.g. the motorized sports).

In the second place, the results from the first objective were combined with the work of A. Waegli [20] to establish a new software's architecture for reconstituting athlete's trajectories via GPS/MEMS-IMU integration. An automated processing was created and several sports experiments were performed for the purposes of software's validation and performances evaluation. The measurements of a new MEMS-IMU sensor were also processed in order to evaluate its suitability. From this work, the following conclusions can be made:

- The new software's architecture needs less interventions from the user due to its high level of automation.

- The new architecture of the processing software was validated and had proved its reliability. Using a single-frequency geodetic GPS receiver and the measurements of a MEMS-IMU, the obtained results corresponds to those of the old software's architecture.

- The performance of an integrated system was then studied by simulating discontinuities in satellite signal reception. The automated processing has shown to be able to bridge gaps lasting up to 10 s in the GPS measurements without compromising the accuracy of the restituted trajectory. Nevertheless, it should be kept in mind that these results depend from the quality of the GPS and IMU observations, as well as from the dynamic of the trajectory.

- In the limited experience, the "standalone" accuracy of the newly acquired MEMS-IMU sensor was not suitable for bridging the GPS outages.

Perspectives

The new software's architecture developed during this project represents a further step towards a "black box" processing, designed for performances assessment in sports via GPS/INS integration. However, a lot of work is still required to reach this ultimate goal.

Further developments could be made on the automated detection to enhance the computation of the inertial orientation. The implementation of a quality index qualifying the detected "quasi static" periods could be an interesting option to better isolate the static ranges suitable for the sensors' bias estimation.

Indeed, the estimation should be computed on static periods, no matter where they are located into the processed signal. On the other hand, the initialization of the orientation should be done at the beginning of the dynamic range, of almost instantaneous way.

As the integration is done in both directions by default, the initialization should be computed once at the beginning and once at the end of the dynamic range. This will avoid starting the backward integration by using the last PVA state computed during the forward integration, when this last has diverged.

The results presented in this report are based on several short data sets, resulting from experiments in various sports applications. It is therefore recommended to acquire additional and also longer data sets, to continue the evaluation and the enhancement of the software.

Finally, the integration of the Crista's inertial measurements represents an interesting field of research. Indeed, this sensor has a good potential due to its larger bandwidth and perfect synchronisation with GPS time. However its performances in the frame of the GPS/IMU navigation are still limited, unless its own stochastic model as well as the computation strategy of its initial orientation are not enhanced.

Bibliography

[1] I.Y. Bar-itzhack. Request - a new recursive algorithm for attitude determination. In *National Technical Meeting of the Institute of Navigation*, pages 699 – 706, 1996.

[2] A. Bryman and D. Cramer. *Quantitative Data Analysis with SPSS Release 8 for Windows: For Social Scientists*. Routledge, New York, NY, 10001, 1999.

[3] C. Capilla. Application of the haar wavelet transform to detect microseismic signal arrivals. *Journal of Applied Geophysics*, 59:36–46, May 2006.

[4] D. L. Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41:613–627, 1995.

[5] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90:1200–1224, 1995.

[6] L. R. Jaisingh. *Statistics for the Utterly Confused, 2nd edition (Utterly Confused)*. McGraw-Hill Pub. Co., 2005.

[7] S.J. Julier and J.K. Uhlmann. New extension of the kalman filter to nonlinear systems. In I. Kadar, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3068 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 182–193, July 1997.

[8] John Maindonald and John Braun. *Data Analysis and Graphics Using R: An Example-based Approach (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge University Press, New York, NY, USA, 2006.

[9] S. Mallat and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory*, 38:617–643, 1992.

[10] Mark L. Psiaki. Extended quest attitude determination filtering, 1999.

[11] J. Qian. *Denoising by wavelet transform.* PhD thesis, Rice University, Department of electrical engineering, 2000.

[12] E-H. Shin. A quaternion-based unscented kalman filter for the integration of gps and mems ins. In *17th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GNSS 2004*, 2004.

[13] E-H. Shin. *Estimation Techniques for Low-Cost Inertial Navigation.* PhD thesis, Department of Geomatics Engineering, University of Calgary, Alberta, Canada, 2005.

[14] J. Skaloud. *Optimizing georeferencing of airborne survey systems by INS/DGPS.* PhD thesis, Department of Geomatics Engineering, University of Calgary, Alberta, Canada, 1999.

[15] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, 1997.

[16] D.H. Titterton and J.L. Weston. *Strapdown inertial navigation technology.* Peter Peregrinus Ltd., 1997.

[17] C. Valens. A Really Friendly Guide to Wavelets. 1999.

[18] R. Van der Merwe and E. A. Wan. The square-root unscented kalman filter for state and parameter-estimation. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 6, pages 3461–3464 vol.6, 2001.

[19] B. Vidakovic and P. Mueller. Wavelets for kids, A Tutorial Introduction. *Discussion Paper 94-13*, 1994.

[20] A. Waegli. *Trajectory Determination and Analysis in Sports by Satellite and Inertial Navigation.* PhD thesis, Thesis 4288, EPFL, 2009.

[21] A. Waegli, J.M. Bonnaz, and J. Skaloud. L'analyse de performance sportive à l'aide d'un système GPS/INS low-cost: évaluation de capteurs inertiels de type MEMS. *Revue XYZ*, 113:19–24, 2007.

[22] A. Waegli, S. Guerrier, and J. Skaloud. Redundant MEMS-IMU integrated with GPS for Performance Assessment in Sports. In *IEEE/ION PLANS 2008*, 2008.

[23] A. Waegli, F. Meyer, S. Ducret, J. Skaloud, and R. Pesty. Assessment of Timing and Performance based on Trajectories from low-cost GPS/INS Positioning. In *International Congress on Science and Skiing*, 2007.

[24] A. Waegli, A. Schorderet, C. Prongué, and J. Skaloud. Accurate Trajectory and Orientation of a Motorcycle derived from low-cost Satellite and Inertial Measurement Systems. In *7th ISEA CONFERENCE 2008*, 2008.

[25] A. Waegli and J. Skaloud. Assessment of GPS/MEMS-IMU Integration Performance in Ski Racing. In *TimeNav'07*, 2007.

[26] A. Waegli and J. Skaloud. Turning Point - Trajectory Analysis for Skiers. *Inside GNSS*, Spring 2007:24–34, 2007.

[27] A. Waegli and J. Skaloud. Optimization of two GPS/MEMS-IMU, Integration Stategies with Application to Sports. 2008.

[28] A. Waegli, J. Skaloud, P. Tomé, and J.M. Bonnaz. Assessment of the Integration Strategy between GPS and Body-Worn MEMS Sensors with Application to Sports. In *ION-GNSS 2007*, 2007.

[29] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

Appendices

Appendix A

The old software's architecture

This appendix presents the structure of the software developed by A. Waegli during his PhD [20]. Appendix A.1 provides a global flow chart of the entire process. The next appendices (A.2 and A.3) show detailed views of the two main parts of the trajectory's computation process.

A.1 Global structure

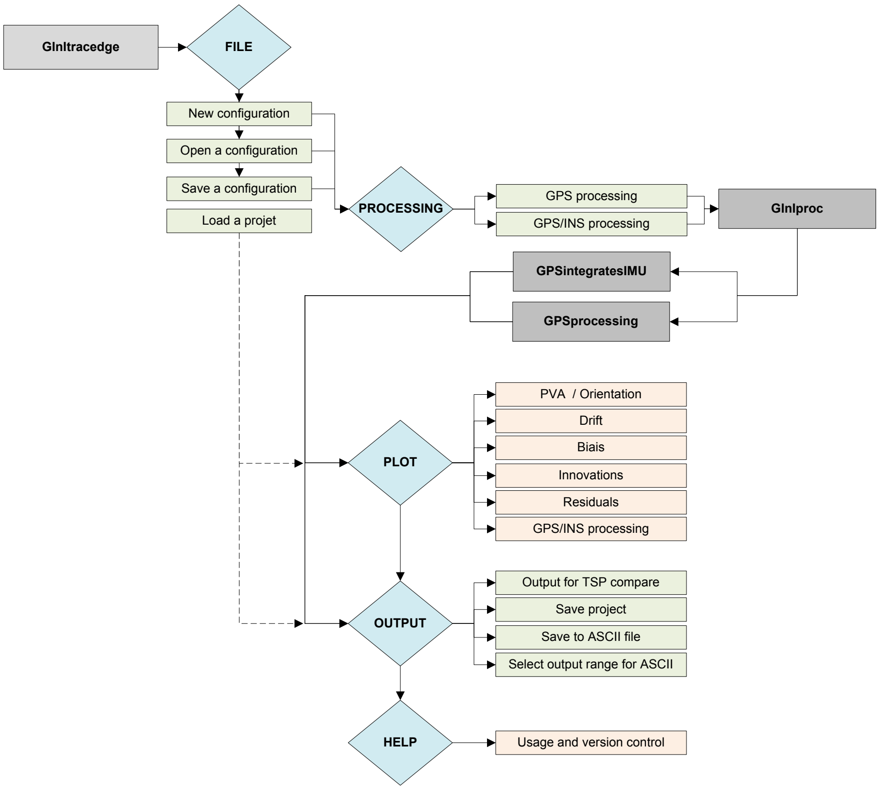


Fig. A.1: Backbone of the software

A.2 Process main code

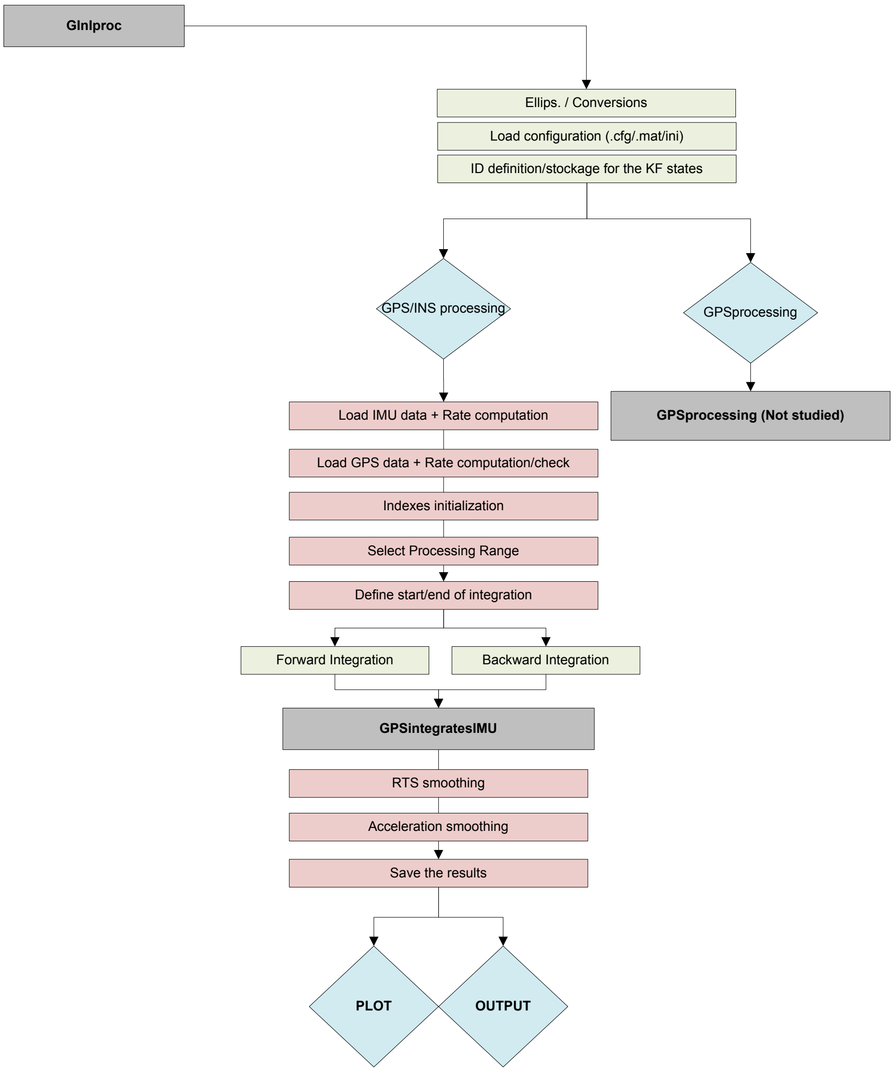


Fig. A.2: Flow chart of the software's main algorithm

A.3 GPS/INS integration code

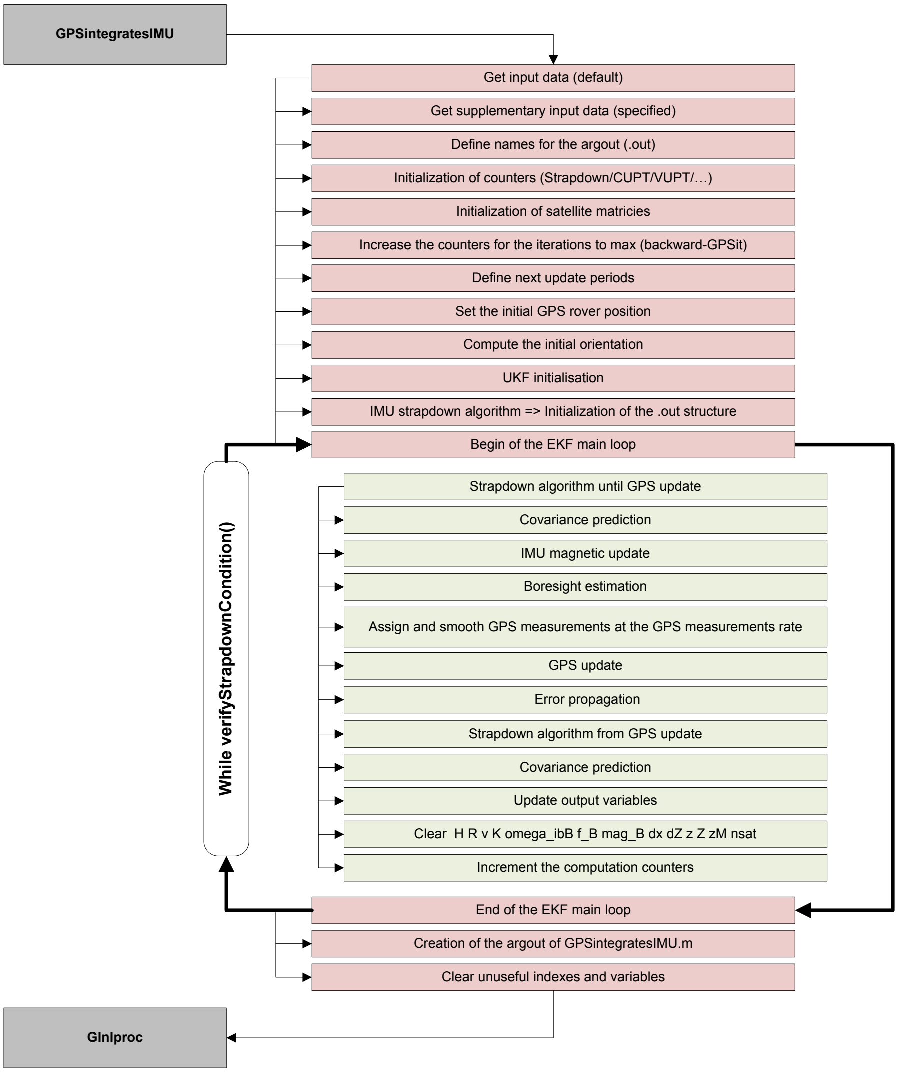
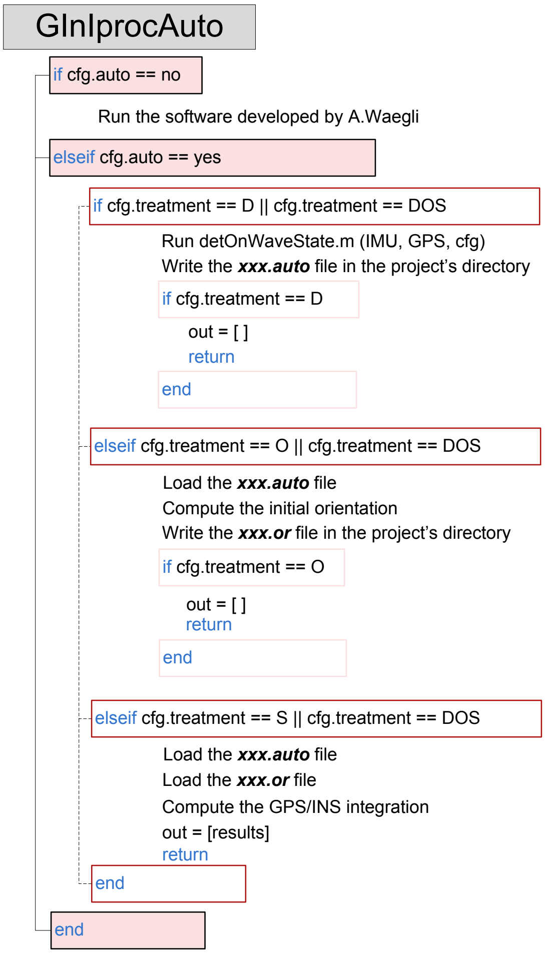

GPSintegratesIMU
Get input data (default)
Get supplementary input data (specified)
Define names for the argout (.out)
Initialization of counters (Strapdown/CUPT/VUPT/…)
Initialization of satellite matricies
Increase the counters for the iterations to max (backward-GPSit)
Define next update periods
Set the initial GPS rover position
Compute the initial orientation
UKF initialisation
IMU strapdown algorithm => Initialization of the .out structure
Begin of the EKF main loop
While verifyStrapdownCondition()
Strapdown algorithm until GPS update
Covariance prediction
IMU magnetic update
Boresight estimation
Assign and smooth GPS measurements at the GPS measurements rate
GPS update
Error propagation
Strapdown algorithm from GPS update
Covariance prediction
Update output variables
Clear H R v K omega_ibB f_B mag_B dx dZ z Z zM nsat
Increment the computation counters
End of the EKF main loop
Creation of the argout of GPSintegratesIMU.m
Clear unuseful indexes and variables
GInIproc

Fig. A.3: Flow chart of the GPS/INS integration

Appendix B

The new software's architecture

This flow chart shows how the new software's architecture works. The code GInIprocAuto replaces the old GInIProc depicted in appendix A.2.



```
GInIprocAuto
if cfg.auto == no
    Run the software developed by A.Waegli
elseif cfg.auto == yes
    if cfg.treatment == D || cfg.treatment == DOS
        Run detOnWaveState.m (IMU, GPS, cfg)
        Write the xxx.auto file in the project's directory
        if cfg.treatment == D
            out = [ ]
            return
        end
    elseif cfg.treatment == O || cfg.treatment == DOS
        Load the xxx.auto file
        Compute the initial orientation
        Write the xxx.or file in the project's directory
        if cfg.treatment == O
            out = [ ]
            return
        end
    elseif cfg.treatment == S || cfg.treatment == DOS
        Load the xxx.auto file
        Load the xxx.or file
        Compute the GPS/INS integration
        out = [results]
        return
    end
end
```

Fig. B.1: Detailed flow chart of the new software's architecture

Appendix C

InterQuartile Range

The InterQartile Range (IQR) is a robust statistical estimator. It gives a measure of spread that is not influenced by any extreme values (outliers) in the data set. In other words, the IQR measures the spread of the middle 50 percent of a data set ordered from the smallest to the largest value. It is the difference between the third (Q_3) and first (Q_1) quartiles, as follows (C.1) [6].

$$IQR = Q_3 - Q_1 \tag{C.1}$$

Fig. C.1 illustrates the concept of the IQR.

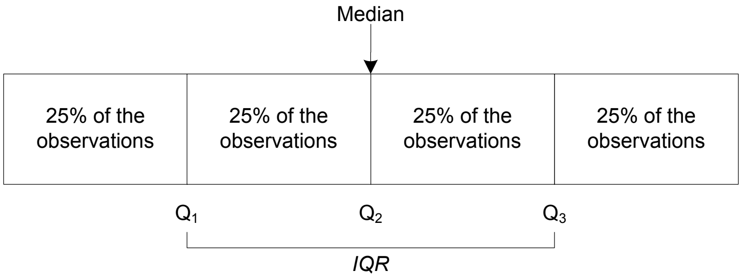


Fig. C.1: Interquantile range. Adapted from [2]

In order to compare the IQR with the SD for a same data set, the IQR is rescaled. Assuming one-dimensional data are normally distributed, the standard deviation on either side of the mean includes roughly 68% of the data. However, the region between the lower and upper quartiles contains 50% of the data. Thus, the following relation makes both estimators comparable (C.2) [8]. The values presented in the various tables of this report are computed with this scale factor.

$$SD = \left(\frac{0.5}{0.68}\right) IQR \approx 0.75 IQR \tag{C.2}$$

Appendix D

Typical configuration file

This appendix presents a typical configuration file used with the automated software.

Tab. D.1: Typical configuration file

GPS Files		
filePath;	V:\Ski\ski6\;	file path
GPSfile;	ski6.cmb;	GPS file (for loosely coupled integration)
refFileName;	.;	raw GPS data from master station
roverFileName;	.;	raw GPS data from rover receiver
ephemerisFile;	.;	ephemeris file
IMU File		
IMUfile;	ski6Xsens.mat;	IMU file (Xsens)
frame;	xFyRzD;	frame orientation
mobileOrientRl;	0.00;	[deg]
mobileOrientPt;	0.00;	[deg]
mobileOrientHd;	0.00;	[deg]
IMU2GPS_B_x;	0.015;	[m]
IMU2GPS_B_y;	0.051;	[m]
IMU2GPS_B_z;	0.055;	[m]
IMU2IMU_B_x;	0.000;	[m]
IMU2IMU_B_y;	0.000;	[m]
IMU2IMU_B_z;	0.000;	[m]
IMU2IMU_B_pt;	0.00;	[deg]
IMU2IMU_B_hd;	0.00;	[deg]
IMUDelay2GPS;	0.00;	[s]
redIMUintegration;	ext;	Redundant IMU integration(synth or ext)
IMUsensorUsed_f;	all;	Used accelerometers (separated by / or all)
IMUsensorUsed_r;	all;	Used gyroscopes (separated by / or all)
IMUsensorUsed_m;	all;	Used magnetometers (separated by / or all)

Typical configuration file - Continued...

Crista IMU file		
IMUfileC;	ski6Crista.mat;	IMU file
UseCrista;	no;	(yes/no)
frameC;	xFyRzD;	frame orientation
mobileOrientRlC;	0.00;	[deg]
mobileOrientPtC;	0.00;	[deg]
mobileOrientHdC;	0.00;	[deg]
IMU2GPS_B_xC;	-0.008;	[m]
IMU2GPS_B_yC;	-0.046;	[m]
IMU2GPS_B_zC;	0.053;	[m]

General parameters		
FilterType;	EKF;	(EKF/UKF/SR-UKF)
GPSintegration;	loosely;	(loosely/closely)
REFmeasRate;	10.00;	[Hz]
GPSmeasRate;	10.00;	[Hz]
IMUmeasRate;	100.00;	[Hz]
covPredRate;	100.00;	[Hz]
strapdownRate;	100.00;	[Hz]
MUPTrate;	10.00;	[Hz]
GPSuptRate;	5.00;	[Hz]
boresightRate;	1.00;	[Hz]
magMeas;	no;	(yes/no)
biasEstimation;	yes;	(yes/no)
boresight;	no;	(yes/no)
CP;	46.16;	[deg]
CM;	7.29;	[deg]
smooth;	0.99;	[-]

Typical configuration file - Continued...

smoothv;	0.75;	[-]
curvatureType;	spline;	(spline)
sampling_f;	1.00;	[-]
processDir;	both;	(forward/backward/both)
RTSsmooth;	yes;	(yes/no)
LPFmag;	yes;	(yes/no)
extrapolation;	yes;	(yes/no)
cycleSlipTol;	1000;	[m]
phaseSmoothInd;	yes;	(yes/no)
useDoppler;	yes;	(yes/no)
cutOffAngle;	0;	[deg]
leapSec;	15;	[s]
itADJtol;	0.010;	[m]
EKFoutputTol;	2.00;	[m] and [m/s]
GPSprocOnly;	no;	(yes/no)
FilterTypeGPSproc;	EKF;	(EKF/ADJ)
satRemoveMan;	.;	(PRN number separated by /)
iniOrient;	quest;	(PhT/coarse/quest)
useOrientIniPeriod;	no;	(yes/no)
directNoiseEst;	no;	(yes/no)
dirNoiseEstAvTime;	1.00;	[s]
dirNoiseEstInt;	1.00;	[s]
Output		
selectProcRange;	true;	(true/false)
defFirstLastEpo;	no;	(yes/no)
firstEpo;	0.00;	(GPS sec)
lastEpo;	604800.00;	(GPS sec)

Typical configuration file - Continued...

saveOutStruct;	no;	(outFile/projectSave/no)
saveGPSrawData;	no;	(yes/no)
useGPSrawData;	no;	(yes/no)
SaveRawIMU;	no;	(yes/no)
UseRawIMU;	no;	(yes/no)
autoName;	false;	(true/false)
useSlopeModel;	false;	(true/false)
outFlagTime;	yes;	(yes/no)
outFlagPos;	no;	(yes/no)
outFlagVel;	no;	(yes/no)
outFlagAcc;	no;	(yes/no)
outFlagOrient;	no;	(yes/no)
outFlagMobileOrient;	no;	(yes/no)
outFlagCurv;	no;	(yes/no)
outFlagAzi;	no;	(yes/no)
outFlagDrift;	no;	(yes/no)
Reference station coordinate		
ref_lat;	0/0/0.0000000;	[dms]
ref_long;	0/0/0.0000000;	[dms]
ref_h;	0.000;	[m]
Initial state IMU		
lat;	0/0/0.0000000;	[dms]
long;	0/0/0.0000000;	[dms]
h;	0.000;	[m]
vN;	0.000;	[m/s]
vE;	0.000;	[m/s]

Typical configuration file - Continued...

vU;	0.000;	[m/s]
rl;	0.00;	[deg]
pt;	0.00;	[deg]
hd;	0.00;	[deg]
t1FwdOrient;	0.00;	[s]
t2FwdOrient;	0.00;	[s]
biasGyro_x;	0.00;	[deg/s]
biasGyro_y;	0.00;	[deg/s]
biasGyro_z;	0.00;	[deg/s]
biasAcc_x;	0.00;	[m/s^2]
biasAcc_y;	0.00;	[m/s^2]
biasAcc_z;	0.00;	[m/s^2]
magnDist_x;	0.00;	[-]
magnDist_y;	0.00;	[-]
magnDist_z;	0.00;	[-]
Initial state IMU - backward computation		
lat_back;	0/0/0.0000000;	[dms]
long_back;	0/0/0.0000000;	[dms]
h_back;	0.000;	[m]
vN_back;	0.000;	[m/s]
vE_back;	0.000;	[m/s]
vU_back;	0.000;	[m/s]
rl_back;	0.00;	[deg]
pt_back;	0.00;	[deg]
hd_back;	0.00;	[deg]
t1BwdOrient;	0.00;	[s]
t2BwdOrient;	0.00;	[s]

Typical configuration file - Continued...

Calibrated sensor biases and scale factors IMU		
calib_biasGyro_x;	0.0000;	[deg/s]
calib_biasGyro_y;	0.0000;	[deg/s]
calib_biasGyro_z;	0.0000;	[deg/s]
calib_biasAcc_x;	0.0000;	[m/s^2]
calib_biasAcc_y;	0.0000;	[m/s^2]
calib_biasAcc_z;	0.0000;	[m/s^2]
calib_magnDist_x;	0.0000;	[uT]
calib_magnDist_y;	0.0000;	[uT]
calib_magnDist_z;	0.0000;	[uT]
calib_sfGyro_x;	0.0000;	[-]
calib_sfGyro_y;	0.0000;	[-]
calib_sfGyro_z;	0.0000;	[-]
calib_sfAcc_x;	0.0000;	[-]
calib_sfAcc_y;	0.0000;	[-]
calib_sfAcc_z;	0.0000;	[-]

Q matrix (driving noise GPS/INS processing)		
whiteNoiseLat;	0.1000;	[m/sqrt(Hz)]
whiteNoiseLong;	0.1000;	[m/sqrt(Hz)]
whiteNoiseH;	0.1000;	[m/sqrt(Hz)]
whiteNoisevN;	0.1500;	[m/s/sqrt(Hz)]
whiteNoisevE;	0.15000;	[m/s/sqrt(Hz)]
whiteNoisevD;	0.1500;	[m/s/sqrt(Hz)]
whiteNoiseE;	0.0100;	[deg/sqrt(Hz)]
whiteNoiseHd;	0.0100;	[deg/sqrt(Hz)]
whiteNoiseAcc;	0.2000;	[m/s^2/sqrt(Hz)]

Typical configuration file - Continued...

whiteNoiseBf;	0.0400;	[m/s^2/sqrt(Hz)]
whiteNoiseGyro;	0.1000;	[deg/s/sqrt(Hz)]
whiteNoiseBw;	0.0500;	[deg/s/sqrt(Hz)]
whiteNoiseDm;	100.000;	[-]
whiteNoiseBore;	0.1000;	[deg]
beta_bf;	0.0100;	[1/s]
beta_bOmega;	0.0010;	[1/s]
beta_dm;	0.1000;	[1/s]
beta_bore;	0.010000;	[1/s]
Q matrix (driving noise, GPS processing only)		
whiteNoiseLatGPS;	1.0000;	[m/sqrt(Hz)]
whiteNoiseLongGPS;	1.0000;	[m/sqrt(Hz)]
whiteNoiseHGPS;	1.0000;	[m/sqrt(Hz)]
whiteNoisevNGPS;	1.0000;	[m/s/sqrt(Hz)]
whiteNoisevEGPS;	1.0000;	[m/s/sqrt(Hz)]
whiteNoisevDGPS;	1.0000;	[m/s/sqrt(Hz)]
UKF variables		
UKFbeta;	2.0000;	[-]
UKFomega_0;	0.5000;	[-]
UKFalpha;	0.0010;	[-]
UKFnoiseTyp;	non;	(add/non)
cholMod;	no;	(yes/no)
Tolerances for magnetic disturbance		
magDistNormTol;	0.1000;	[uT]
magDistAngleTol;	2.0000;	[deg]

Typical configuration file - Continued...

Initial P matrix		
sigmaLat;	1.0000;	[m]
sigmaLong;	1.0000;	[m]
sigmaH;	1.0000;	[m]
sigmavN;	1.0000;	[m/s]
sigmavE;	1.0000;	[m/s]
sigmavD;	1.0000;	[m/s]
sigmaE;	5.0000;	[deg]
sigmaHd;	20.0000;	[deg]
sigmaAcc;	5.0000;	[m/s^2]
sigmaBf;	0.1000;	[m/s^2]
sigmaBw;	0.5000;	[deg]
sigmaDm;	5.0000;	[uT]
sigmaBore;	5.0000;	[deg]

Typical configuration file - Continued...

Initial P matrix (GPS processing only)		
sigmaLatGPS;	0.0000;	[m]
sigmaLongGPS;	0.0000;	[m]
sigmaHGPS;	1.0000;	[m]
sigmavNGPS;	1.0000;	[m/s]
sigmavEGPS;	1.0000;	[m/s]
sigmavDGPS;	1.0000;	[m/s]

R Matrix		
stdpos;	0.1000;	[m]
stdvel;	0.1500;	[m/s]
stdmag;	5.000;	[uT]
stdbore;	2.0000;	[deg]
pseudoRangeStd;	4.0000;	[m]
phaseRateStd;	0.0400;	[m/s]

Rejection Criteria		
Position;	1.5000;	[m]
Altitude;	2.0000;	[m]
Velocity;	3.0000;	[m/s]

Automation		
FreqRangeUp;	10.0000;	[Hz]
FreqRangeDown;	0.1000;	[Hz]
Automation;	yes;	(yes/no)
Treatment;	DOS;	(D/O/S/DOS)