# Human Motion Modelling for Simulation Testing of GNSS Equipment

Kimon Voutsis

Department of Civil, Environmental and Geomatic Engineering
University College London

July 2017

*Supervisors:*
Dr. Paul GROVES
Dr. Jan BOEHM

Thesis submitted for the degree of Doctor of Philosophy

# Author's Declaration

I, *Kimon Voutsis* confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis. The code which was developed to analyse the data for this thesis, as well as referenced but unpublished technical reports, may be copyrighted both to UCL and Spirent Communications plc.

……………………………………………

# Abstract

Pedestrian motion-induced dynamics along the line-of-sight (LOS) between a GNSS receiver and a satellite, may disrupt the nominal operation of GNSS carrier-tracking loops, by introducing cycle slips and/or false frequency locks. In combination with other factors, e.g. multipath interference, weak signal conditions or limited availability of GNSS signals, the receiver could provide a degraded navigation solution or even lose signal lock. This thesis researches firstly how pedestrian motion affects the operation of carrier phase lock loops (PLLs), used by some GNSS receivers, and frequency lock loops (FLLs), used by all GNSS receivers; and secondly, what is the best way to model pedestrian motion in order to simulate the error effects of pedestrian motion-induced dynamics on a GNSS antenna, via a simulated GNSS carrier phase lock loop (PLL).

The thesis reviews the relevant literature on human biomechanical modelling, path-finding and inertial/GNSS navigation, to design a custom pedestrian motion model (PMM). The PMM validation is supported by motion capture (MoCap) experiments using an inertial/GNSS sensor held by, or attached, on a pedestrian. The thesis also describes an implementation of simulated GNSS carrier-tracking loops (SGCTLs) in Matlab, to assess the effect of human MoCap profiles and synthetic human motion profiles (from the PMM) on the performance of the SGCTLs.

The testing results suggest that GNSS antenna motion dynamics due to typical pedestrian motion can induce excessive cycle slips due to dynamics stress on the simulated PLL and FLL. Therefore, antenna dynamics should be considered when designing GNSS tracking loops and navigation algorithms for pedestrian applications to allow the GNSS receiver track human motion-induced dynamics effectively. The thesis concludes with carrier-tracking bandwidth recommendations for GNSS receiver design, based on the presented evidence. Under good signal conditions (above 40dB-Hz), the minimum recommended bandwidths for PLLs and FLLs are 15Hz and 5Hz, respectively, in order to respond effectively to the dynamic stress induced by typical pedestrian movements. Finally, the results indicate that the PMM can represent the LOS dynamics stress on the SGPLL within an acceptable tolerance. Future work encompasses the analysis of the pedestrian motion effects on real GNSS receivers.

# Acknowledgments

I would like to thank my supervisors, Dr. Paul Groves, especially, and Dr. Jan Boehm for their invaluable support and guidance over the course of this research.

I would also like to thank Dr. Catherine Holloway, Debbie Walter and Nikos Papadosifos of UCL, as well as my former UCL colleagues Dr. Henry Martin and Dr. Lei Wang, for their help with the human motion capture experiments, and also, the other members of the Space Geodesy and Navigation Laboratory for their support and advice. I am also grateful to Dr. Ronald Wong, Colin Ford and, especially, Mark Holbrow of Spirent Communications for their continuous support and advice throughout this research period. Finally, I would like to thank my family for always being a constant source of support, encouragement and inspiration.

This research was co-funded by the Engineering and Physical Sciences Research Council (EPSRC) and Spirent Communications plc.

# List of Abbreviations and Acronyms

| | |
|---|---|
| **1D** | One dimension |
| **2D** | Two dimensions |
| **3D** | Three dimensions |
| **A/D** | Analog-to-digital |
| **ADC** | Analog-to-digital converter |
| **ADR** | Accumulated delta range |
| **AGC** | Automatic gain control |
| **APC** | Antenna phase centre |
| **C/A** | Coarse/acquisition |
| **CASE** | Collaborative award in science and engineering |
| **CD** | Compact disc |
| **CDMA** | Code-division multiple access |
| **CoG** | Centre of gravity |
| **CoM** | Centre of mass |
| **CTM** | Coordinate transformation matrix |
| **CWS** | Comfortable walking speed |
| **DLL** | Delay lock loop |
| **DOF** | Degrees of freedom |
| **DUT** | Device under test |
| **ECEF** | Earth-centered earth-fixed |
| **E-M** | Electro-magnetic |
| **EPSRC** | Engineering and Physical Sciences Research Council |
| **Eq.** | Equation |
| **FFT** | Fast Fourier Transform |
| **FLL** | Frequency lock loop |
| **GBAS** | Ground-based augmentation system |
| **GLONASS** | Global'naya navigatsionnaya sputnikovaya sistema |
| **GNSS** | Global navigation satellite systems |
| **GPS** | Global positioning system |
| **Gyro** | gyroscope |
| **IEEE** | Institute of Electrical and Electronic Engineers |

| | |
|---|---|
| **IF** | Intermediate frequency |
| **IMU** | Inertial measurement unit |
| **ION** | Institute of Navigation |
| **Is and Qs** | I (in-phase) and Q (quadrature-phase) signals |
| **ITRF** | International Terrestrial Reference Frame |
| **KF** | Kalman filter |
| **L1/2/5** | Link 1/2/5 |
| **LBS** | Location based services |
| **LLA** | Latitude, longitude and altitude |
| **LNF** | Local navigation frame |
| **LOS** | Line-of-sight |
| **MCS** | Monte Carlo Simulation |
| **MEMS** | Micro-electro-mechanical systems |
| **NCO** | Numerically controlled oscillator |
| **NLOS** | Non-line-of-sight |
| **NWU** | North, west, up |
| **NED** | North, east, down |
| **PBF** | Pedestrian body frame |
| **Ph.D.** | Doctor of philosophy |
| **PLL** | Phase lock loop |
| **PNT** | Positioning, Navigation and Timing |
| **PRN** | Pseudo-random noise |
| **PVT** | Position, velocity and time |
| **RMS** | Root mean square |
| **RMSE** | Root mean square error |
| **SBAS** | Space-based augmentation system |
| **SD** | Standard deviation |
| **SGCTL** | Simulated GNSS carrier-tracking loop |
| **SGFLL** | Simulated GNSS frequency lock loop |
| **SGPLL** | Simulated GNSS phase lock loop |
| **SI** | Système international d’ unités |
| **SNR** | Signal-to-noise ratio |
| **UCL** | University College London |

| | |
|---|---|
| **UK** | United Kingdom |
| **UMT** | User motion trajectory |
| **USA** | United States of America |
| **UTC** | Coordinated Universal Time |
| **WLAN** | Wireless local area network |
| **w.r.t.** | with respect to |
| **ZVU** | Zero velocity update |

# Nomenclature

This list contains symbols used to denote matrices, vectors, scalars, subscripts and superscripts and qualifiers throughout this thesis. Wherever physical quantities are mentioned, SI (Système international d' unités) units are implied or explicitly used. Quantities noted with bold small letters represent vectors, capital bold letters represent matrices, small letters represent scalars and small letters in subscript or superscript italicised text denote frames of reference or indices (e.g. current epoch).

**Matrices** (upper case bold typeface font)

| | |
|---|---|
| $\mathbf{C}$ | Coordinate Transformation Matrix (3x3) (unitless) |
| $\mathbf{I}_3$ | Identity matrix (3x3) (unitless) |
| $\boldsymbol{\Omega}$ | Skew-symmetric matrix (3x3) of angular rate vector (rad/s) |
| $\Delta\mathbf{C}$ | Attitude update matrix C (3x3) (unitless) |

**Vectors** (lower case bold typeface font)

| | |
|---|---|
| $\mathbf{r}$ | Position vector (3x1) in Cartesian coordinates (m) |
| $\mathbf{p}$ | Position vector (3x1) in Latitude, Longitude, Height (rad, rad, m) |
| $\mathbf{d}$ | Position vector normalised in some length unit (3x1) (unitless) |
| $\mathbf{v}$ | Velocity vector (3x1) (m.s$^{-1}$) |
| $\mathbf{a}$ | Acceleration vector (3x1) (m.s$^{-2}$) |
| $\mathbf{j}$ | Jerk vector (3x1) (m.s$^{-3}$) |
| $\mathbf{f}$ | Specific force (m.s$^{-2}$) |
| $\boldsymbol{\alpha}$ | Attitude increment vector (3x1) (rad) |
| $\boldsymbol{\omega}$ | Angular rate |
| $\boldsymbol{\psi}$ | Attitude vector (3x1) in Euler angles (rad) |

**Scalars** (lower case italicised font)

| | |
|---|---|
| $\rho$ | Range (m) |
| $\varphi$ | Phase (rad) |
| $\varphi$ | Roll (rad) |
| $\theta$ | Pitch (rad) |
| $\psi$ | Yaw (rad) |
| $\tau$ | Time interval (s) |
| $f$ | Frequency (Hz) |
| $B$ | Bandwidth (Hz) |

| | |
|---|---|
| $t$ | Time (s) |
| $c/n_0$ | Carrier power to noise density ratio (unitless) |
| $C/N_0$ | Carrier power to noise density ratio (dB-Hz) |

**Subscripts and superscripts**

| | |
|---|---|
| + | Denotes an updated attitude of a frame |
| - | Denotes an attitude of a frame before being updated |
| $\alpha$ | Reference frame |
| $\beta$ | Body frame |
| $\gamma$ | Resolving frame |
| $i$ | Inertial frame |
| $e$ | ECEF frame |
| $n$ | Local navigation frame |
| $l$ | Local tangent plane |

**Qualifiers**

| | |
|---|---|
| $\exp(x)$ | Exponential of variable x, i.e. $e^x$ |
| $\lvert x \rvert$ | Absolute value of variable x |
| $\lVert \mathbf{x} \rVert$ | Magnitude of vector **x** |
| $dx$ | Differential of variable (or vector) x |
| $\delta x$ | Error in variable (or vector) x |
| $\Delta x$ | Correction or Difference of (variable or vector) x |
| $\tilde{x}$ | Observed variable (or vector) x |
| $\hat{x}$ | Estimated/Predicted variable (or vector) x |
| $\bar{x}$ | Average of variable x |
| $\dot{x}$ | First derivative of x variable (or vector) |
| $\ddot{x}$ | Second derivative of x variable (or vector) |
| $\dddot{x}$ | Third derivative of x variable (or vector) |
| $x(+)$ | Updated variable (or vector) x |
| $x(-)$ | Variable (or vector) x before being updated |
| $(\mathbf{C})^{\mathrm{T}}$ | Transpose of matrix **C** |
| $\left[\mathbf{x}^{\wedge}\right]$ | Skew-symmetric matrix (3x3) of 3D vector **x** (3x1) |

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1. Introduction

This chapter provides an introductory overview of this thesis and is divided in four sections. The first, Section 1.1, introduces the background and motivation of this thesis, the second, Section 1.2, discusses the thesis' aims and objectives, and the third, Section 1.3, briefly discusses a publication resulted from this research work. The chapter concludes with the presentation of the thesis' structure in Section 1.4.

## 1.1 Background and Motivation

Global Navigation Satellite Systems' (GNSS) signal simulation is important as it allows a GNSS-enabled device under test (DUT) to be validated against specific application positioning, velocity and timing (PVT) requirements. Validation testing is essential to ensure that a GNSS DUT behaves as expected and can be relied upon within its operational limits. DUTs which are within the research scope of this thesis encompass all GNSS equipment used for pedestrian applications, e.g. smartphones and wearable equipment, like those used for athletic activity monitoring. It is worth noting that all GNSS receivers use carrier frequency tracking, but only some of them carrier phase tracking, for a more precise determination of the DUT position.

GNSS-enabled devices are used in a wide range of different applications. This range spans from consumer day-to-day applications, such as aided-navigation using a GNSS-enabled smartphone, or location based services (LBS), to military, e.g. ballistic missiles guidance systems; utilities, e.g. time-synchronisation of data networks and electrical power grids; and as life-critical applications, e.g. emergency and rescue services. Knowing the position of a pedestrian with an acceptable degree of accuracy is essential as, naturally, when using or testing positioning, navigation and timing (PNT) equipment, the first question which a user might ask is "can I trust" that PVT navigation solution or, in other words, how *accurate* is the obtained navigation solution against the specific application requirements. From an engineering point of view, it is also worth knowing how, if possible, the PVT solution can be *improved*, with respect to (w.r.t.) an external reference, assumed to be true, e.g. by being at least one order of magnitude more accurate than the DUT test results.

The errors introduced by the motion of the GNSS antenna propagate to the navigation solution and may degrade the PVT solution accuracy. In extreme cases, if the errors due to GNSS antenna motion dynamics are combined with other GNSS vulnerabilities, e.g. multipath interference in urban canyons, then the GNSS equipment can stop tracking satellite signals. This in turn could cause an outage in providing a navigation solution, if the number of available satellite signals is limited.

In this context, suitable pedestrian motion generation for simulation testing is important, because it can replicate the motion dynamics' environment of the GNSS DUT, and assess the DUT's performance under these conditions, in a controlled and repeatable way. This reduces the need to conduct field experiments to assess the GNSS DUT's performance in the presence of a specific motion dynamics' profile. The cost of field experiments may far exceed the cost of a simulation system, despite the typically higher upfront cost of the latter, depending how many different motion profiles need to be captured, and considering the great variation in human activities and specific experimental conditions, e.g. weather, external sources of vibration, interference etc.

Pedestrian motion generation for simulation testing is of interest to Spirent Communications, as it has the potential to increase their product offering in simulation-based testing of (GNSS-enabled) PNT equipment for pedestrian applications. In addition, the outcomes of this thesis, aim to enable Spirent's customer-base, i.e. PNT equipment manufacturers, integrators and/or testers, to validate and/or improve, as appropriate, the robustness of their GNSS-enabled PNT equipment in the presence of pedestrian-induced motion dynamics. This Ph.D. project was sponsored by an EPSRC Industrial CASE studentship and Spirent Communications plc., noting that the aim of the CASE scheme is to allow UK businesses, to take the lead in arranging projects with an academic partner of their choice, so they can mutually benefit from this research collaboration [1].

This thesis shows how real pedestrian motion profiles can degrade the performance of GNSS receivers' carrier-tracking loops, noting that motion dynamics can also affect GNSS receiver oscillators which can be sensitive to vibration, causing phase noise [2], although analysis of vibration effects is out of scope of the thesis. It also shows how a pedestrian motion model (PMM) can be developed in order to recreate the 3D motion dynamics' environment within which the PNT equipment operates, when either held by hand or attached on the human body, e.g. with an arm-band or inside a back-pack. The aim of generating a synthetic motion dynamics' profile is to compare it against real motion (captured by field experiments), and assess whether they have, or not, a similar impact on the performance of GNSS receivers' carrier-tracking loops. It must be underlined that the modelling of pedestrian motion in this thesis aims at creating a test platform for GNSS receivers (in a simulation environment), so it is not intended to generate a complete pedestrian navigation system, which could be used e.g. for validation testing of pedestrian dead-reckoning (PDR) or zero-velocity update (ZVU) algorithms. Also, it is worth noting that human body effects on GNSS antennas, such as signal shadowing (body-masking), diffraction and signal attenuation due to the dielectric properties of human body are not within the scope of this thesis, due to time limitations, but rather form part of a longer-term aspiration for this research.

Generally, PNT equipment can be tested by means of the following four ways [3]:

- *Field trials*, whereby data are collected for a range of representative operational scenarios of the device under test (DUT). The advantage of field trials collection is that the collected data represent very realistically the DUT behaviour under the specific experimental conditions in the field. However, a disadvantage of this method is the difficulty in collecting data for extreme case scenarios, e.g. the effect of radio interference on GNSS signals without disrupting publicly available signals, or if data under adverse weather conditions had to be collected.

- *Recorded (and re-played) data* methods provide advantages compared to field trials, as testing using recorded data can be repeated accurately, allowing different algorithms, e.g. embedded in a device's firmware, and sensors to be tested in bulk under the same conditions and to be compared in terms of relative performance, minimising the testing costs per DUT. Also, this can be beneficial to optimise the design of the hardware or software algorithms, e.g. by detecting data synchronisation issues between hardware and software components, or by monitoring the response of the DUT under specific fault conditions, e.g. when the incoming GNSS RF signal is attenuated/blocked in order to simulate a DUT inside a car which passes under a tunnel.
- *Laboratory testing* enables performance analysis of PNT equipment under controlled physical conditions, e.g. using static or vibration tables to recreate specific vibration conditions, or when capturing pedestrian walking data using a treadmill. Although laboratory testing provides controlled experimental conditions and repeatability of results, a drawback is the limited complexity of conditions which can be recreated compared to reality, i.e. by employing field-trials or recorded and replayed data which can capture, e.g. multipath and blocking/attenuation in GNSS signals while a pedestrian walks in an urban environment. The cost of laboratory testing depends on how many times the laboratory equipment can be re-used, to reduce the cost per experiment.
- *Hardware and software simulation*. This type of simulation can be used to predict the performance of a DUT over a range of conditions which are difficult to recreate in field-trials or even in laboratory testing, e.g. specific characteristics of GNSS signals or interference. Using a simulation environment, we can have an exact knowledge of the simulated input and output can be achieved, noting that the reliability of simulation results depends on how reliable are the input values and the underlying real world model, irrespective of how sophisticated is the results' analysis. Another consideration along these lines has to do with the depth of detail that the underlying model has to represent in order to avoid adding unnecessary information with ensuing time and financial costs; therefore, it is essential to know which parameters impact significantly the performance of the DUT, within the application requirements. Spirent Communications specialise in this type of simulation which is also the focus of this thesis. After an initial cost of installing the necessary testing equipment and training staff, the cost of performing a hardware/software simulation can be less than all other types of testing.

Table 1.1 summarises the four testing approaches. As a degree of uncertainty is always contained in any measurement, the term "truth reference" means a more accurate (or less uncertain) reference than the DUT, against which it can be compared, following the above mentioned methods. In PNT simulation testing, a truth model is called "realistic" when it can recreate the properties of interest of a physical system in question, within the tolerance of the application requirements that it has been designed for.

| | *Field trials* | *Recorded data* | *Laboratory* | *Simulation* |
|---|---|---|---|---|
| *Repeatable* | No | No | Yes | Yes |
| *Testing scenarios availability* | No | No | Limited | Yes |
| *Realistic* | Yes | Yes | Yes | Limited |
| *Cost for single experiment* | Low | Medium | Medium | High |

Table 1.1. Navigation systems' testing methods comparison.

The advantage of real-world data recording is the realistic representation of the parameters we want to capture, e.g. linear acceleration, GNSS PVT etc. However, this approach can only provide *ad-hoc* results, i.e. for the scenario that the data were captured for. Practically, there are many occasions where we would like to generalise the results of our testing to cover a wider range of scenarios. The wide range of possibilities we need to account for renders it difficult, if not impossible, the design of a planned experiment, e.g. to measure GNSS position solution accuracy under specific atmospheric conditions and GNSS satellites' constellation geometry.

In cases like these, the simulation approach can provide us with a timely and cost-effective solution allowing repeatable and controlled experimental conditions. Also, it can provide benchmarking of the results with a pre-defined truth reference. E.g. for a static GNSS receiver, a displacement in the positioning solution would indicate an error in the receiver, therefore we have a performance metric to improve the design of the receiver and reduce the positioning uncertainty. Other advantages of simulation testing in the case of GNSS applications is that it might be the only available testing solution if we want to simulate GNSS signals which are not operational yet, or in the case of certain events which occur rarely e.g. the insertion of a leap second every couple of years to Coordinated Universal Time (UTC) time, in order to account for the Earth's rotation variability [4], i.e. UTC lags GPS time by a specific offset, equal to the sum of the inserted leap seconds.

An important issue which arises at this point is the definition of what constitutes a realistic result of a simulation. Having clear testing requirements is essential in this respect, to decide how to simulate real world phenomena, and select an appropriate testing approach in terms of accuracy and precision. On a similar note, as an analytical model is a simplification of a real-world phenomenon, a simulation can typically only test special cases of that analytical model. Therefore, caution needs to be exercised when interpreting the simulation testing results, in order to avoid extrapolating them beyond the scope of the simulated parameters [5].

With respect to simulation testing of PNT equipment used for pedestrian applications, due to the human motion variation depending on individual characteristics and spatio-temporal context, it is more practical to focus first on finding those aspects of human motion that have the most significant effect on PNT equipment performance and then try to model this behaviour. The chapter continues with the discussion of the aims and objectives of the thesis.

## 1.2 Aims and Objectives

The general aim of this thesis is to identify those aspects of pedestrian motion and sensor location that most affect the performance of GNSS equipment, in terms of disrupting the nominal operation of carrier-tracking functions within the GNSS device under test (DUT). The thesis' aims are to answer these two main questions:

- Which are the key aspects of pedestrian motion that affect the performance of GNSS DUT?
- Which is the best way to generate a pedestrian motion model for simulation testing of GNSS DUT?

The answer to the first research question is examined from a human biomechanics perspective, i.e. by analysing human motion and how it can create an adverse (or not) dynamics' environment for the GNSS antenna, as well as how the GNSS DUT responds to the presence of human-induced antenna motion dynamics. To address the second research question, the thesis discusses different approaches to model human motion and then describes a custom pedestrian motion model (PMM), which is used to simulate the 3D motion dynamics induced to the GNSS antenna, held by hand or attached on the human body. The validation of the PMM (synthetic) output is performed by comparing the effects of the real motion on GNSS sensors' performance (analysed to address research question one) to the effects caused by the synthetic motion.

The following points summarise the specific aims of this project, which are:

- To identify which types of pedestrian motion are relevant to this research in terms of affecting the performance of GNSS receivers. This work supports the design of the thesis' experiments in order to capture real pedestrian motion data in the field. The definition of which pedestrian motion scenarios to capture, is done in collaboration with Spirent within the aims of the thesis.
- To determine how the dynamics induced by pedestrian motion affects the performance of GNSS receiver's tracking loops.
- To find what is the best approach to capture pedestrian motion using IMU/GNSS equipment for examining the effect it has on GNSS receiver's performance.

- To investigate what is the best method to model pedestrian motion for the purposes of recreating a realistic error profile on GNSS carrier-tracking loops.
- An implicit aim of the thesis is to publish the results of this research to appropriate conferences, e.g. the annual Institute of Navigation GNSS conference, or relevant journals. A summary of proposed future research work is provided in Chapter 8.

The results of this research work aim at demonstrating the essential tools needed in order to develop a software tool that generating the GNSS antenna motion dynamics' profile. This motion profile can then be input to Spirent software packages in order to drive the GNSS RF signal generation through Spirent's GNSS RF simulators. This software tool could also assess the impact of the antenna dynamics on the GNSS sensor's performance for the purposes of testing representative GNSS and other signals at the user antenna, as well as simulate IMU outputs.

## 1.3 Publications

As part of this project, the paper "The Importance of Human Motion for GNSS Simulation Testing" was published in ION GNSS+ 2014 conference proceedings. It also received a best presentation award in Session A6 "Simulation and Testing", among 8 presentations in total.

## 1.4 Thesis Structure

As discussed in Section 1.2, the thesis aims to address which human movements most affect the performance of GNSS carrier-tracking loops and which is the best way to simulate human motion in order to recreate the same error behaviour that real motion has in GNSS carrier-tracking loops. To address these two research questions, the thesis progresses as follows.

Chapter 2 reviews the relevant literature, providing the background knowledge required to address the aims of this thesis, discussed in Section 1.2, and is divided into three sections: Section 2.1 introduces some fundamental concepts of pedestrian motion analysis, Section 2.2 provides the reader with the basic principles of inertial and GNSS navigation sensors' operation. Finally, Section 2.3 details methods of human motion capture and modelling, including routing (pathfinding) between two locations. The literature review presented in Chapter 2, provides insight on how different areas of the literature, such as human biomechanics, navigation sensors/systems and pedestrian motion capture/modelling, link together in order to develop a workflow for addressing the thesis' research questions, as shown in Figure 1.1. In particular, Chapter 2 details different ways to capture human motion, which informs the motion capture (MoCap) experiments described in Chapter 3, how to model human motion and develop a pedestrian motion model (PMM) to output synthetic (simulated) human motion (see Chapter 5). Also, Chapter 2 provides the background knowledge of GNSS receivers' operation, used to conduct and analyse the results of MoCap experiments (see Chapter 3), as well as to develop the simulated GNSS carrier-tracking loops (SGCLTs), described in Chapter 4, in order to analyse the effects of real (MoCap) and synthetic (PMM) human motion on the SGCTLs (with the results presented in Chapter 6).

Chapter 3 describes the thesis' human motion capture (MoCap) experiments. The human MoCap experiments provide the real human motion profiles, used to assess the impact of real motion on the performance of GNSS equipment and also, for validating the synthetic output of the pedestrian motion model (PMM). Chapter 3 comprises three sections, the first, Section 3.1, describes the experimental equipment used in the human motion capture experiments, the second, Section 3.2, details the pilot experiments and the last, Section 3.3, presents the main thesis' experiment of human MoCap. The motion capture described in Chapter 3 was essential in order to analyse the effects of real human motion on the performance of SGCTLs (see results in Section 6.2); to validate the synthetic motion output of the PMM (see results in Section 6.3); and also to make recommendations on GNSS carrier tacking loops design (see results in Section 6.4). It must be noted that human MoCap data was not used for human biomechanical modelling, as this was based on other studies in the human biomechanical literature (see Section 5.3).

Chapter 4 details the method of simulating GNSS carrier-tracking loops in Matlab, based on the literature review presented in Chapter 2. Chapter 4 is divided into four sections, the first, Section 4.1, introduces the assumptions made in order to implement the GNSS carrier-tracking loops in Matlab, the second, Section 4.2, explains the simulation of a GNSS carrier phase lock loop and the third, Section 4.3, details the simulation of a GNSS carrier frequency lock loop. Chapter 4 concludes with Section 4.4, which presents the validation results for the simulated GNSS carrier-tracking loops based on a static test case scenario. The work presented in Chapter 4 provides the testbed for testing the effects of real (MoCap) human motion (whose capture process is described in Chapter 3) and the effects of synthetic motion output from the pedestrian motion model (described in Chapter 5) on the simulated GNSS carrier-tracking loops (SGCTLs) developed for this thesis.

Chapter 5 explains the development method of a custom pedestrian motion model (PMM) which simulates the 3D motion dynamics of positioning and navigation equipment held by or attached to a pedestrian. The discussion in Chapter 5 is based on the literature review of human motion analysis, capture and modelling introduced in Chapter 2. Chapter 5 is divided into four sections. The first, Section 5.1, describes the candidate pedestrian motion methods considered in this thesis, justifying why the current approach was selected, the second, Section 5.2, details the implementation of the pedestrian routing model (PRM), which models the movement of human body between two given locations, and the third, Section 5.3, explains the development of a human biomechanical model (HBM), which simulates rotations of human body segments, given an underlying motion (walking) and overlaid gestures on the top of the underlying motion. Chapter 5 concludes with Section 5.4, which explains the synthesis of the HBM and the PRM into the pedestrian motion model (PMM), which outputs the simulated 3D motion dynamics of the positioning and navigation equipment (PNE), when attached on or held by a pedestrian.

The work presented in Chapter 5 is important for the purposes of the thesis, as a custom-made PMM can be manipulated (by means of rotating constituent body segments) to represent human motion, without the need to conduct new MoCap experiments when a new motion profile is required. Although that work was a major time commitment for the thesis, it also facilitated the future productisation of the thesis' outcomes. This is firstly in terms of licensing, i.e. avoiding commercial issues which may arise when using third-party functionality, and secondly, future support and/or tailored enhancements of the PMM, since the PMM code is fully available and understood.

Furthermore, Chapter 5 explains how a tailored human biomechanical model can be controlled using MoCap data in order to produce realistic human motion. This was considered appropriate for the thesis as although modelling human motion using MoCap techniques produces realistic, human-like results, the process can be involved in terms of conducting the experiments end processing the results, e.g. by weighting and smoothing measurements. In addition, combining MoCap segments to synthesise a complete trajectory may involve additional smoothing in order to avoid discontinuities (especially when higher-order quantities, e.g. velocity or acceleration are of interest). Other methods involve imposing physical constrains on human movement, and generate motion without requiring prior MoCap and equipment calibration. However, the result may be unrealistic, due to the high dimensionality of human movement, i.e. there are several ways for human body segments to move and rotate in order to reach from a specific posture to another one. Therefore, these methods have not been considered suitable for this thesis.

Chapter 6 provides the results from the analysis of real human MoCap profiles' effect on the simulated GNSS carrier tracking loops (SGCTLs) and the effect of the synthetic motion from the PMM. Chapter 6 presents the evidence to address the research questions discussed in Section 1.2, and is divided into four sections. The first, Section 6.1, presents the method of (pre)processing the MoCap data from the thesis' main experiment (Section 3.3) and the method of analysing the SGCTLs' results, both for the (input) real and synthetic motion profiles. The second, Section 6.2, addresses the first research question by showing the effect of real MoCap scenarios on the performance of the SGCTLs and the third, Section 6.3, addresses the second research question, by analysing the effect of synthetic motion on the simulated GNSS phase lock loop and comparing this to the real motion effect. Chapter 6 concludes with Section 6.4, which provides a summary of the results presented in this chapter and recommendations on carrier tracking loop bandwidths, for GNSS receiver design in the presence of typical pedestrian motion.

The thesis continues with Chapter 7, which provides a summary of the conducted research work, and draws conclusions based on the results presented in Chapter 6. The thesis concludes with Chapter 8, which discusses the future work deriving from this research and the productisation of the thesis' outcomes to improve Spirent's product offering in the area of simulation-based testing of positioning, navigation and timing (PNT) equipment for pedestrian applications. Figure 1.1 illustrates how the thesis' (main body) chapters are linked in a workflow chart, in order to address the research questions discussed in Section 1.2.

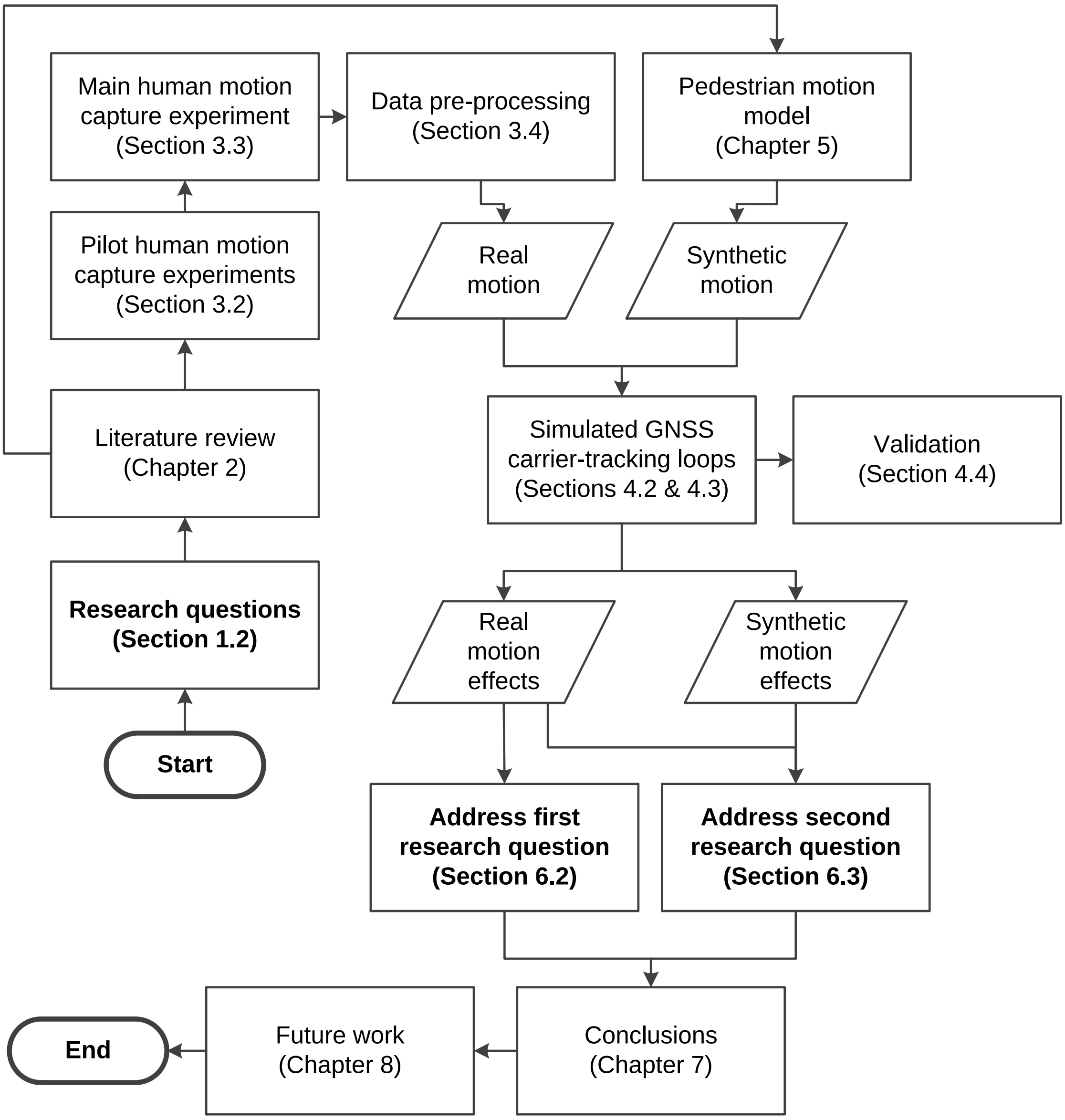
Main human motion capture experiment (Section 3.3)
Data pre-processing (Section 3.4)
Pedestrian motion model (Chapter 5)
Pilot human motion capture experiments (Section 3.2)
Real motion
Synthetic motion
Literature review (Chapter 2)
Simulated GNSS carrier-tracking loops (Sections 4.2 & 4.3)
Validation (Section 4.4)
Research questions (Section 1.2)
Real motion effects
Synthetic motion effects
Start
Address first research question (Section 6.2)
Address second research question (Section 6.3)
End
Future work (Chapter 8)
Conclusions (Chapter 7)

Figure 1.1. Thesis' workflow chart in order to address the research questions

In addition, the thesis includes the following appendices which support the discussion in the thesis' main body chapters. Appendix A presents a method for calculating the human body centre of mass (CoM), based on the 3D coordinates of human segments (which define a specific human body posture). Appendix B supports the discussion about the identified methods for pre-processing the experimental motion capture (MoCap) data in Section 6.1.1. The data pre-processing stage aims at removing any significant artefacts from the MoCap data before they enter the SGCTLs to assess their impact. Appendix B also describes one of the proposed pre-processing methods, i.e. an algorithm for detecting and smoothing transients (see Section 3.2) from the MoCap data.

Appendix C summarises the experimental equipment (Xsens MTI-G IMU/GPS) specifications, while Appendix D presents the frame transformation from Xsens MTi-G default "Forward, Left, Up" body-frame to a "Forward, Right, Down" body-frame, as required by the inertial navigation equations described in Section 6.1.1. Appendix E presents a custom method of deriving acceleration from velocity, constraining the mean velocity and mean acceleration between the input and output datasets to be preserved. Finally, Appendix F contains the Matlab code used in this thesis, which is included in the accompanying CD.

The thesis continues with the literature review in Chapter 2.

# Chapter 2. State of the Art

This chapter reviews the relevant literature, providing the background knowledge required to address the aims of this thesis, discussed in Section 1.2. This chapter is divided into three sections, with Section 2.1 introducing the reader to the fundamental concepts of pedestrian motion analysis. Section 2.2 provides the reader with the basic principles of inertial and GNSS navigation and sensors' operation. Finally, Section 2.3 details methods used to capture and model human motion, including routing (pathfinding) between two locations. This chapter supports the design, development and implementation of methods in the remainder of the thesis, i.e. the design of an appropriate human motion data capture protocol (including usage of the experimental equipment) and pre-processing of the captured data in order to remove artefacts in Section 6.1.1; the development of the simulated GNSS carrier-tracking loops (see Chapter 4) to assess the effect of the real captured motion and the synthetic motion output of the proposed pedestrian motion model (described in Chapter 5). In addition, this chapter aims to provide an overall understanding of the thesis' research area which will support the interpretation of the results (in Chapter 6), as well as the identification of research gaps and longer-term aspirations for this thesis, discussed as part of future work in Chapter 8.

This Chapter provides the justification of the selected methods of analysis in the thesis. In particular, it provides the evidence as to why it was necessary to capture real human motion (see Chapter 3) in order to assess the impact on GNSS carrier-tracking loops and validate the output of the pedestrian motion model (PMM) developed for this thesis (see Chapter 5); and finally, why this tailored PMM for this thesis was the most appropriate method for modelling human motion, instead of combining captured motion sequences or using human biomechanical models with physical movement constraints.

## 2.1 Pedestrian Motion

This section introduces some basic concepts used for the description and analysis of human motion, which supports the discussion of human motion capture and modelling in Section 2.3.

### 2.1.1 Overview

The study of pedestrian (human) motion supports the development of a pedestrian motion model, discussed in Chapter 5. Pedestrian motion is twofold for the purposes of this project; it encompasses routing, i.e. locomotion between two locations and human body movements, e.g. walking, running, and gestures such as arm movements, which fall into the field of human biomechanics. Human biomechanics (or "kinesiology", according to some authors [6]) deals with studying the mechanical properties of human body segments, mainly for performance, injury and rehabilitation analyses [7]. Another implication areas of human biomechanics is studies on the safety of special pedestrian groups against slips and falls [8],[9].

In terms of representing the motion of human body segments, human biomechanics comprise two branches: *kinetics* and *kinematics*. Kinetics investigate what causes a segment/joint to move, so the focus is on forces and torques, e.g. exerted by groups of muscles, while kinematics describe and analyse the resulting movement of a segment/joint, so they employ linear or angular quantities e.g. position, velocity, acceleration etc. for this purpose [10]. In this project, emphasis is given to the results of human motion and not to the causes of that motion; therefore, the kinematic representation of human movement is considered more appropriate.

Any movement of a human segment (or the body as a whole) may encompass up to 6 degrees of freedom (DOF), three for translational (or linear) motion and three for rotational (or angular) motion. As far as the whole human body motion is concerned, the translation and rotation components are referenced along and about, respectively, a right-hand system of three orthogonal axes which comprise the *pedestrian body frame* (PBF), as shown in Figure 2.4. The three orthogonal axes of the PBF are: a. *Anteroposterior* axis denoted with “X” pointing forward, b. *Mediolateral* axis denoted with “Y” pointing right and c. *Longitudinal* (vertical) axis denoted with “Z” pointing down. In turn, these axes form a system of three orthogonal *cardinal* planes: a. the *sagittal* plane which divides the PBF into right and left, b. the *frontal* (or coronal) plane which divides the PBF into front and back, and c. the *transverse* (or longitudinal) plane which divides the PBF into up and down. It follows that segment rotations about the anteroposterior (X) axis are within the frontal plane, rotations about the mediolateral (Y) axis are within the sagittal plane and rotations about the vertical (Z) axis are within the transverse plane. Individual body joints and segments have their own frames of reference and specific motion definitions, as discussed in the next Section, 2.1.2.

The initial posture of the humanoid illustrated in Figure 2.4 is called the *fundamental starting position*, or fundamental posture/position in biomechanical studies [6], noting that these terms are used interchangeably in this thesis. The three orthogonal axes of the PBF intersecting at the human centre of mass (CoM), while at this fundamental position, as illustrated in Figure 2.1.

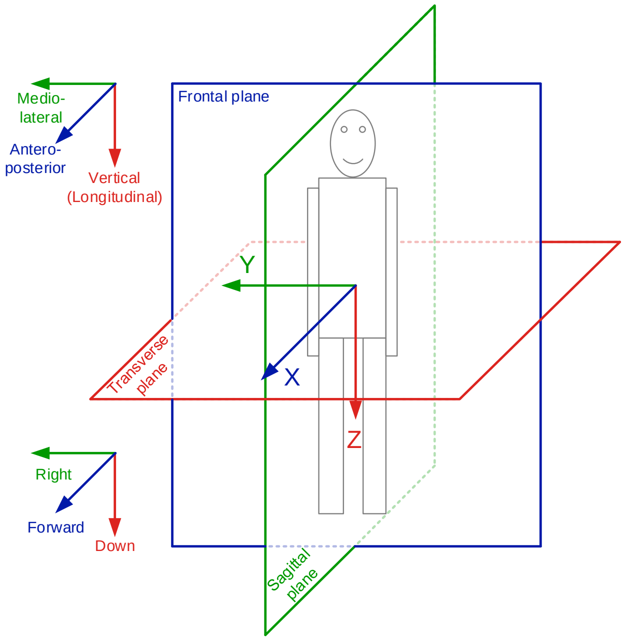

Figure 2.1. Pedestrian body frame (PBF) axes and planes

### 2.1.2 Human body segments rotation

Human kinematics deals with the motion analysis of human body segments in terms of translation and rotation in 3D space using kinematic quantities, e.g. position, velocity, angular velocity etc. Rotational movements of human segments can occur about axes which intersect at a joint, or about external axes to the human body. The ability of a human body segment to rotate about a reference axis constitutes one DOF (degree of freedom). Some joints may have three DOFs of rotation, e.g. a shoulder, while other joints only one DOF, e.g. a knee. Similarly, when a segmental movement involves linear movement, one DOF involves translation of that segment which is registered in one of the cardinal planes (or the local joint planes).

Generally, a segment rotation about the antero-posterior axis of the PBF is called *abduction* or *adduction*, depending on whether the segment moves away or towards, respectively, w.r.t the human body, noting that two segments located in the right and left side of the human body who abduct (or adduct) rotate in opposite directions wr.t. the PBF's antero-posterior (X) axis, i.e. if the right arm abduction is negative about the PBF X-axis, then the abduction of the left arm will be positive about the PBF X-axis. A segment rotation about the mediolateral (Y) axis of the PBF is generally called *flexion* or *extension*, depending on whether the adjacent joint segments' angle decreases or increases, respectively, noting that segments which flex or extend rotate in the same direction w.r.t. PBF, no matter which side of the human body they are located. However, these are only general rules applying to rotational movements w.r.t. the cardinal planes. The naming of motions should be learned on individual joint-by-joint basis due to the fact that in most of the cases the motion of a segment involves both rotational and translational elements which change the axis of rotation; therefore the assigned “axis of rotation” is usually a midpoint between the many instantaneous centres of rotation, i.e. the points around which the segment appears to move in any elementary part of its trajectory [10], although for the purposes of this thesis this translational component is assumed to be negligible.

Each human body segment or joint exhibits between one and three DOFs of angular movement, depending on which direction it can rotate (w.r.t. the PBF or a local segment/joint 3D reference frame). The remainder of this section discusses the angular DOFs for each human body segment, starting from the top of human body.

### *Head*

The human skull is connected with the spine through the topmost cervical vertebra (C1 in Figure 2.2) and its rotation exhibits three DOFs.

### *Thorax (or spine)*

The human spine comprises 7 cervical, 12 thoracic and 5 lumbar vertebrae, as illustrated in Figure 2.2. All thoracic vertebrae are connected to their respective thoracic ribs. In the front of the human thorax, 7 of the ribs are connected directly to the sternum (called ‘true’ ribs) and 5 are not (called ‘false’ ribs). On its lower part, the human spine is connected with the sacrum and then with the pelvis through the two sacroiliac joints. Thoracic (or spine) rotation exhibits three DOFs.

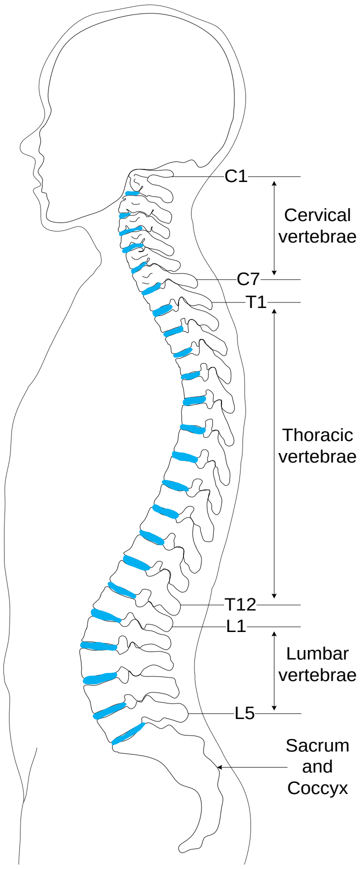

Figure 2.2. Human spine vertebrae

### *Arms*

An arm comprises the upper arm segment which is connected with the thorax through the shoulder joint exhibiting three DOFs of rotation; the forearm segment which is connected with the upper arm through the elbow joint exhibiting one DOF of rotation; and the hand which is connected with the forearm through the wrist joint exhibiting three DOFs of rotation. For further details on the hand structure, the reader may refer to [10].

### *Pelvis*

The pelvis exhibits three DOFs of angular movement during human gait, which are listed below with their respective names:

- Tilt or list which occurs about the mediolateral (Y) axis.
- Obliquity which occurs about the antero-posterior (X) axis.
- Rotation which occurs about the vertical (Z) axis.

According to [11], pelvic list does not decrease the trunk translation along the vertical axis, but it rather increases it. Pelvis rotation amplitude varies with gait speed (i.e. walking or jogging or running speed), while its phase is more coherent with the legs (rotation) phase at higher gait speeds [12]. Also, it is worth noting that the human CoM lies within the pelvis, assuming body is in fundamental position/posture.

### *Legs*

A leg comprises the thigh (femur) segment which is connected with the pelvis through the hip joint and exhibits three DOFs of rotation, the shank segment which is connected with the femur through the knee joint exhibiting one DOF of rotation and the foot which is connected with the shank through the ankle joint exhibiting three DOFs of rotation. For further details on the foot structure, the reader may refer to [10].

Following the discussion on human body segments' and joints' angular DOFs, Table 2.1. summarises the number of DOFs for each of the above items.

| *Segment* | *Angular*<br>*Degrees of Freedom (DOF)* |
|---|---|
| Head | 3 |
| Thorax (spine) | 3 |
| Arms | Shoulder joint: 3<br>Elbow: 1 (Roll w.r.t. PBF)<br>Wrist: 3 |
| Pelvis | 3 |
| Legs | 3 |

Table 2.1. Angular DOFs of human body segments and joints

### 2.1.3 **Human gait cycle**

Human locomotion is the process by which a human subject moves itself from one location to another [13]. From the perspective of functional anatomy, human locomotion comprises nervous stimuli, transferred to groups of muscle fibres which generate tensions to the bones causing the respective human segments to move. More detailed analysis as to which muscle groups are active during the human gait cycle, can be provided by electro-myography (EMG); however, since EMG is an indirect way of detecting muscular activity, the results are difficult to be quantified in terms of muscular contraction and force [6]. EMG has been considered by researchers as and aid to personal navigation algorithms by investigating how muscular activity relates to pedestrian movement [14], or as a part of a pedestrian dead-reckoning algorithm, estimating walking speed [15], or heading [16]. EMG techniques are not readily available to provide accurate quantified results in terms of human motion kinematics, therefore have been considered out of the scope of the present thesis.

Human gait encompasses walking and running and is probably the most common type of human locomotion. In biomechanical terms, human gait is a complex process which could be described as the human body movement where the erect, moving human body is supported first by one leg and then the other. Human gait comprises periodic events, e.g. heel strikes, repeated in a certain pattern, which enable the study of the whole movement through a representative movement section, i.e. one gait cycle, which is equivalent to one period of the whole (periodic) movement. It follows that human gait encompasses walking, jogging and running. A human *gait cycle* (or *stride*) is defined as the movement of the human body between the moment that the heel of a *reference leg* touches the ground, and the moment that the heel of the *reference leg* touches the ground again, to initiate a new gait cycle. According to [17], human gait is a function of principal motion patterns, which involve pelvic rotation, tilt and lateral displacement, as well as knee and foot mechanisms. Therefore, the human gait pattern depends almost entirely on the motion of the *locomotor* unit, which encompasses the pelvis and the legs, while the remaining segments of the human body (head, arms and trunk) comprise the *passenger* unit.

Human gait analysis uses some basic spatio-temporal factors [18], such as step/stride length, step width (shown in Figure 2.3), gait speed, cadence (or step rate/frequency), stance and swing times, which form the pedestrian's stride characteristics [19]. One *step* is defined as the movement of the human body between the moment that the heel of the *reference (or ipsilateral) leg* touches the ground, and the moment that the heel of the *opposite (or contralateral) leg* touches the ground. *Stride* length is the distance covered in one gait cycle, i.e. over two steps. *Step width* is the projected distance along the mediolateral Y axis on the horizontal plane between the point where the right and left leg heels touch when they hit the ground, illustrated in Figure 2.3.

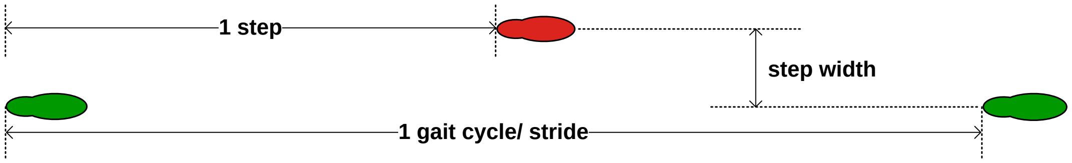

Figure 2.3. Spatio-temporal components of human gait cycle (after [20]).

Gait speed is the distance covered over the time unit measured in m/s (or some authors use non-SI units in m/min). A concept commonly used among biomechanical studies is comfortable (or free) walking speed (CWS), i.e. gait speed which is selected freely by the subject during the experiment. The CWS varies about 10% (1σ) between individuals, depending on their gender, age and limb length [19]. Cadence is the number of steps per time unit (measured in steps/s or steps/min).

Walking speed is a function of the stride length and cadence, following the equation [19]:

$$v = \frac{Lc}{2}, \tag{2.1}$$

where *v* is the gait speed (m/s), *L* is the stride length (m) which comprises two steps, and *c* is the cadence (steps/s). The relation between step length and gait speed is studied in [21], indicating that walking speed has a non-linear relation with the step length even when the step length is normalised w.r.t. the subjects' height (stature). Another finding of the same study was that the CWS was about 1.5m/s for all subjects. Differences in the gait pattern encompass female subjects who tend to increase their cadence (steps over the time unit – or step frequency) as walking speed increases compared to male subjects who tend to increase their step length, and also, that step length and cadence variability was significantly higher among female subjects. However, to generalise the results from gait analysis studies it is essential to sample randomly among groups of different geodemographic characteristics (e.g. gender, nationality) [18]. In [22], the authors found, based on a sample of 800 measurements, that people tend to walk at different speeds depending on the context of the motion, e.g. faster on floors in shopping centres (2Hz cadence, 1.4m/s gait speed) than footbridges (1.8Hz cadence, 1.3m/s gait speed) and confirm that women tend to walk at higher cadence and men walk at higher speed, although they found that the relationship between cadence and walking speed is linear, as shown in Eq. (2.1).

The relationship between subject height, walking speed and step length has been investigated, e.g. in [19] and [21]. These studies suggests that it is preferable to measure experimentally the step length, as the relationship between walking speed and step length is non-linear, even when the step length is normalised in height. Another study [23] suggests that a human trajectory is not planned as a series of successive steps, but there is a common underlying strategy that controls the whole human trajectory w.r.t. the final (target) location. This study showed that different pedestrians walk in variable step lengths, given the same start and end locations and obstacles along their path, although their 2D trajectories look similar. Also, pedestrian behaviour and route choices also depend on interactions with other pedestrians [24][25]. These lines of enquiry could form part of future work, in order to expand the scope of this thesis by comparing the pedestrian motion between different subjects, discussed in Chapter 8.

The events and phases of a human gait cycle are illustrated in Figure 2.4, with the right leg coloured in green and the left leg coloured in red. The motion of the reference leg during one gait cycle can be divided into two phases: *stance* and *swing*, depending on whether the reference leg is in contact with the ground or not, respectively.

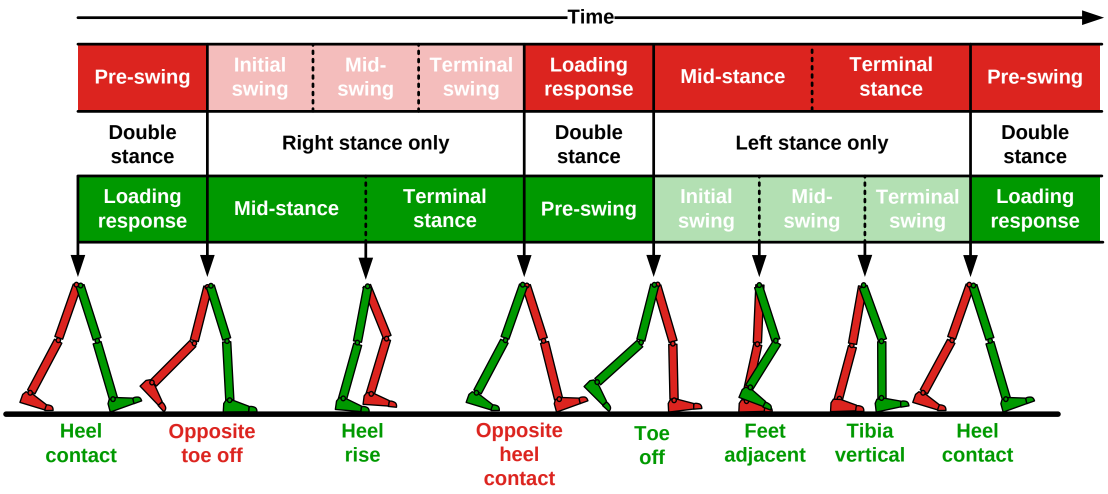

Figure 2.4. Events and phases during one human gait cycle (after [20])

The stance phase begins when the heel of the reference leg touches the ground and ends when the toe of the reference leg leaves the ground. This toe-off event marks the beginning of the swing phase, with the reference leg swinging over the ground and it ends when the heel of the reference leg touches the ground again, which marks the beginning of a new gait cycle. A gait cycle can be divided in 7 phases, which are marked by 8 biomechanical events [13], [20], explained in Table 2.2 and graphically represented in Figure 2.4. The first column of Table 2.2 shows the gait cycle events of the reference leg, the second column the respective gait cycle phases of the reference leg and the third column, the gait cycle phase of the opposite leg.

The stance phase comprises about 60% of one walking gait cycle, while swing the rest 40%, noting that there is no uniform agreement on the exact duration of stance and swing [20] over one gait cycle, as it depends on the particular pedestrian physio-anatomical characteristics and the type of activity. When both legs are in touch with the ground, this is called *double support*, or *double stance* phase, which typically comprises about 20% of the gait cycle. When a pedestrian is jogging or running the phases and their duration change significantly, i.e. the stance phase duration is reduced from about 60% while walking to 30% while running, and 22% when sprinting [26], since the double support phase, decreases as walking speed increases, until it becomes zero. It follows that the distinction between walking and other forms of gait, is that while walking, one leg (either the reference or contralateral one) is always in contact with the ground [7], in other words there is no "flight" phase, i.e. when none of the legs is in touch with the ground.

This brief introduction on the gait cycle phases and events, provides the analytical background to characterise and model human walking in this thesis, as it provides a reference upon which additional pedestrian movements (gestures) can be modelled, as discussed in Section 5.3.

| *Gait cycle event* | *Phase of reference leg during the gait cycle* |
|---|---|
| Reference leg – heel contact | Loading response (Stance)<br>Body weight transferred onto the reference leg. Heel used as a rocker. Knee flexed to absorb the shock. |
| Opposite leg – toe off | Mid-stance (Stance)<br>Reference leg advances over the stationary foot by ankle extension. Knee and hip extend. |
| Reference leg – heel rise | Terminal stance (Stance)<br>Heel rises and reference leg advances over the forefoot rocker. Knee increases its extension and then just begins to flex slightly. Increased hip extension puts the reference leg in a more trailing position. |
| Opposite leg – heel contact | Pre-swing (Stance)<br>Reference leg responds to opposite leg's heel strike with increased ankle plantar-flexion, greater knee flexion and loss of hip extension. |
| Reference leg – toe off | Initial swing (Swing)<br>Reference leg lifted and advanced by hip flexion and increased knee flexion. Ankle partial extension. |
| Both legs – feet adjacent | Mid-swing (Swing)<br>Reference leg advancement anterior to the body weight line by further hip flexion. Knee is allowed to extend in response to gravity while the ankle continues extending to neutral position. |
| Reference leg – tibia vertical | Terminal swing (Swing)<br>Reference leg advancement completed by knee extension. The hip maintains its earlier flexion and the ankle remains extended to neutral position. |
| Reference leg – heel contact | End of previous gait cycle and beginning of a new one. |

Table 2.2. Human walking - gait cycle events and phases.

### 2.1.4 **Human CoM motion during a gait cycle**

The referencing and rotation of human body segments along with the description of the human gait cycle leads to the discussion of a significant characteristic of human gait: the human CoM motion during gait. The CoM is equivalent to CoG and represents the mean average point of mass distribution of the human body particles. During a gait cycle, the human CoM exhibits linear displacement in 3D (usually resolved along the three axes of the PBF, see Section 2.1.1), reflecting the variable change of support of the passenger unit by the locomotor unit.

An interesting property of CoM movement during gait, is that it oscillates in a sinusoidal manner along all three axes of the PBF, as illustrated in Figure 2.5.

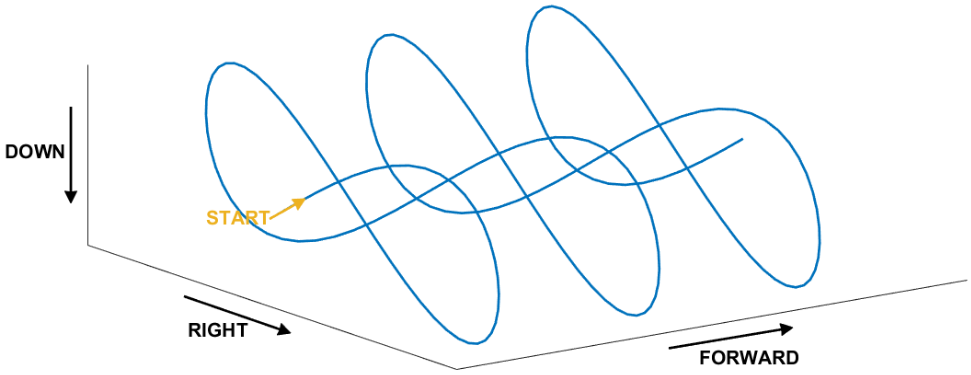

Figure 2.5. Example of 3D sinusoidal motion of the human CoM, expressed in PBF axes

This finding is common in many studies of human biomechanics [13], [27], [28], [29], [30], [31]. More specifically, the CoM describes two vertical and one mediolateral oscillations per gait cycle independently of the gait type, e.g. walking or running [32]. The CoM movement during gait can be modelled by a simple mechanical model of two massless legs in [33]. In this study, the energy cost (calculated as the kinetic energy of the modelled human body CoM) was associated with gait spatio-temporal parameters (speed and step length) and minimised using optimisation techniques. This study [33] found that for lower gait speeds (i.e. walking) the work done by each leg in order to move the (modelled) human body CoM forward, say *C*, is minimised using an inverted walking pendulum model, illustrated in Figure 2.6, which encompasses two moving (massless) legs, with the CoM modelled at the point that they are connected. At higher speeds (where the flight phase is dominant), *C* can be minimised by a bouncing/ impulsive run model, which can bounce between running flight phases, as a response to the force exerted to the leg when the heel touches the ground; and at intermediate speeds, *C* can be minimised by a hybrid model of a pendulum run, which follows the inverted walking pendulum at stance phase and the impulsive run model at flight phase. This model is similar to [30], which models the difference in compression of the CoM during heel strike to the ground between walking and running gaits. This study found that the touch-down angle of the heel striking to the ground and the leg compression during the stance phase of the gait cycle (as the leg in question is the single support of the human body weight), should be taken into account to determine the CoM trajectory and whether the gait refers to walking or running.

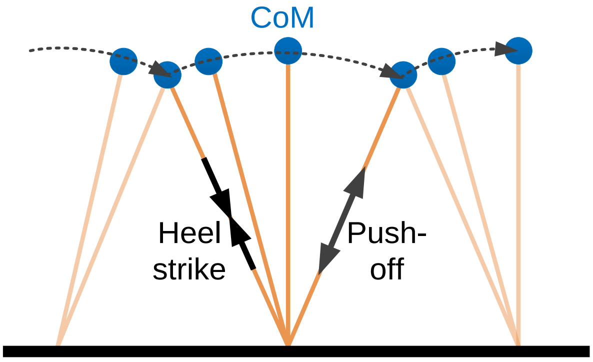

Figure 2.6. Example of an inverted pendulum model for a single leg walking (after [33])

In practice, human gait is far more complex than simple mechanical models can represent, since joints and segments present additional translational and rotational DOFs during gait, e.g. the relative linear motion between the pelvis (encompassing the CoM) and the trunk [29]. During one gait cycle, the pelvis tilts about 4° in the sagittal plane, drops/rises about 4° in the frontal (coronal) plane and rotates about 10° in the transverse plane [19]. Trunk relative translational and rotational movements with respect to (w.r.t.) the CoM are investigated in [32], during walking and running over consecutive gait cycles. The results are consistent with [34], which also found that the sagittal rotation (inclination) of the trunk depends on the walking speed. Also, evidence suggests that swing of the arms does not affect gait, but it helps a human subject to recover more effectively from external perturbation during walking [35]. The evidence from another paper [23], suggests that the head rotation (towards a target location) precedes the rotation of the trunk and pelvis as the human subjects walk along a curved trajectory, especially on paths with higher curvature, showing that the head anticipates the change of the instantaneous walking direction, in order to drive the whole-body movement in space. This paper [23] also studies the spatial variability between steps and whole trajectories among different subjects trying to reach a range of targets, and suggests that the formation of the trajectory (pathfinding between two locations) is not planned as a series of successive steps, but there is rather a common underlying strategy to plan the whole trajectory, as the paths among different subjects were very similar, while the feet positions (as a result the step lengths) showed significant variability.

Various methods have been employed to calculate accurately the displacement of CoM during walking or running [36]. In [37], three methods of CoM calculation were compared, namely: optical tracking of a reflective marker placed above the tail-bone (sacrum); segmental analysis; and double integration of acceleration derived from force platform data. The authors of this study found that the sacral reflective marker method did not produce consistent results for walking speeds higher than 0.8 m/s, which was attributed to the relative dynamics between the pelvis, which was tracked by the fixed sacral reflective marker, and head, arms and trunk. Force platforms have also been used for human motion analysis, e.g. in [38], in order to measure ground reaction forces in 3D, although simple designs measure along the vertical axis only. Typically, they are placed on the floor as tiles, above which a human subject stands or moves. In [28], the authors found that the CoM displacement along the vertical axis is proportional to the gait speed, with an average CoM vertical excursion 2.7cm and 4.8cm at 0.7m/s and 1.6m/s walking speeds, respectively. The CoM displacement along the mediolateral axis was found to be inversely proportional to gait speed, with an average CoM mediolateral excursion of 3.8cm and 7cm at 0.7m/s and 1.6ms gait speeds, respectively. Another finding of this study [28] is that that stride and step lengths increase with gait speed but stride width decreases, although in a non-linear manner.

Other papers study the pelvic rotation about the vertical axis of the PBF, because it reduces significantly the vertical displacement of the CoM [7], although other authors reported that it is minimal and only about 10-12% of the total CoM vertical displacement [9],[11].

A proposed test to show how well the human body CoM motion follows a sinusoidal motion in 3D, about a nominal centre of oscillation, is now described. The human body CoM coordinates, $\tilde{\mathbf{r}}^{p}_{pCoM}$ , were calculated as part of the PMM using the method explained in Appendix A. A model of three sinusoids can be used to fit the calculated CoM from the PMM and the calculate the relevant statistics describing how good is the fitting, e.g. by calculating the root mean square error (RMSE) between the model and the underlying CoM coordinates form the PMM. The proposed model of calculating the CoM coordinates analytically is:

$$\hat{\mathbf{r}}^{p}_{pCoM}\Big|_{k} = \begin{pmatrix} \hat{r}^{p}_{pCoM,x}\big|_{k} \\ \hat{r}^{p}_{pCoM,y}\big|_{k} \\ \hat{r}^{p}_{pCoM,z}\big|_{k} \end{pmatrix} = \mathbf{r}^{p}_{pCoM}\Big|_{1} + \begin{pmatrix} A_x \cdot \sin\left(2\hat{f}_{gait}t_k + \hat{\phi}_x\right) \\ A_y \cdot \sin\left(\hat{f}_{gait}t_k + \hat{\phi}_y\right) \\ A_z \cdot \sin\left(2\hat{f}_{gait}t_k + \hat{\phi}_z\right) \end{pmatrix}, \qquad (2.2)$$

$$t_k = (k-1)\tau \quad , \quad k = 2\ldots m$$

where $\tau$ is the epoch time interval (s), assuming discrete sampling, $m$ is the number of epochs within one full gait cycle, $f_{gait}$ is the gait frequency (Hz) i.e. how many gait cycles are contained within one second, $A$ is the amplitude (m) of the CoM displacement along each of the three PBF axes w.r.t. the initial CoM coordinates at the first epoch ($k$ = 1) of the gait cycle, and $\phi$ is the phase (rad) of the CoM w.r.t. a full gait cycle (2π rad). It is worth noting that the period of lateral displacement of the CoM (along the $y$ PBF axis) is half the period of the CoM displacement along the other two PBF axes ($x$ and $z$), since the body weight oscillates once per gait cycle swaying once to the right side and once to the left (or vice versa), while the oscillation of the CoM along e.g. the vertical $z$ axis occurs twice per gait cycle [19], once per each leg at stance phase (see Section 2.1.3). The underlying assumption is that due to the repetitive pattern of the gait cycle, the last epoch ($k = m$) of a previous gait cycle is the first epoch of the next one.

The chapter continues with the discussion of navigation sensor and systems' fundamental concepts and operational principles.

## 2.2 Navigation Sensors and Systems

This section introduces briefly the operation of positioning and navigation sensors and systems, their error sources and sensor integration. The reader may refer to detailed navigation textbooks for further information, e.g. [3], [39], [40], [41], [42], [43]. This section supports the simulation of GNSS tracking loops in Matlab (see Chapter 4), which forms the testing and validation platform of analysing the effects of human motion on the SGCTLs, as well as the validation of the pedestrian motion model discussed in Chapter 5.

### 2.2.1 Frames of reference

A navigation solution typically provides the user with position, velocity and attitude information of a body frame w.r.t a reference frame. The navigation solution may be expressed in a third frame's axes, called the resolving frame. All frames comprise a right-hand set of three orthogonal axes. In geodesic texts, a frame (term used in this thesis, as well as navigation textbooks) is equivalent to the realisation of a coordinate reference system, i.e. a coordinate reference frame. Throughout this thesis, the following definitions of frames are used:

- An *Earth-Centred Earth-Fixed* (ECEF) *frame*, is based on a coordinate system of three orthogonal axes (X, Y and Z – following the right-hand rule) with an origin at the Earth's centre of mass with its Z axis pointing along the Earth's axis of rotation to the geodetic North pole and its X axis X pointing to the 0° longitude prime meridian and Y axis to the 90° East meridian. In this thesis, the origin and Z axis are defined using the current realisation of the WGS84, which follows closely (at centimetre-level accuracy) the International Terrestrial Reference Frame (ITRF) current realisation of the International Terrestrial Reference System (ITRS) [44]. For further information on how the ITRS is defined, the reader may refer to geodesic texts, e.g. [44], [45].
- A *Local tangent plane* (LTP) is based on a coordinate system of three orthogonal axes (X, Y and Z – following the right-hand rule), with an origin at the Earth's surface, modelled as an ellipsoid defined by the current realisation of WGS84. The X and Y axes intersect at the origin and form a plane tangential to the current realisation of WGS84 ellipsoid at the origin point. The X axis points towards the geodetic North pole and the Y axis towards West (orthogonal to X axis). The Z axis is vertical to the X-Y plane and intersects with X and Y axes at the origin

pointing upwards, forming a North, West, Up (NWU) system of axes, following the right-hand rule. The LTP is a convenient frame to express the navigation solution of a body frame, e.g. a landing aircraft (body frame) on an airway (LTP).

- A *Body frame,* is based on a coordinate system of three orthogonal axes (X, Y and Z – following the right-hand rule), with an origin fixed w.r.t. the object whose navigation solution is described. The origin can be, e.g. the proximal joint of a human body segment (see Chapter 5 for examples on how the human biomechanical body segments and the pedestrian body frames are defined), or for an inertial measurement unit (IMU) the point in space at which it measures motion. The IMU axes point along which direction specific force is measured and about which direction rotation is measured.

### 2.2.2 Inertial Navigation Systems

An inertial navigation system (INS) provide an inertial solution, i.e. a solution based on the output (measurements) from inertial sensors only, i.e. accelerometers and gyros as a minimum. The inertial solution typically encompasses position, velocity and attitude information. INS are self-sufficient as they do not require external infrastructure to operate e.g. satellites transmitting signals, but they have to be initialised in terms of position, velocity and attitude from external sources, e.g. GNSS, magnetometers, pre-surveyed position etc. Also, an INS requires initial calibration and regular correction. An INS typically comprises an inertial measurement unit (IMU), an inertial navigation processor and a power supply unit. IMUs are platforms containing inertial sensors, typically a triad of accelerometers and a triad of gyroscopes (gyros), along with a temperature sensor used for their temperature-dependent calibration. In this thesis, emphasis is given on IMUs which are based on MEMS (Micro Electronic Mechanical Systems) technology, which contain electromechanical parts whose dimension is at the micro-meter level (as opposed to other inertial technologies employing large mechanical parts, e.g. gimbals) and can be produced cheaply in large quantities, which is the reason that they can be found in many daily-used electronic devices, such as smartphones. IMUs measure specific force (accelerometers) and angular rate (gyros). The position, velocity and attitude solutions are obtained by processing inertial sensors' outputs, without the need of any other external sources of information, e.g. radio signals.

IMUs' range of applications spans from everyday use, e.g. inside car navigation systems or smartphones, to advanced military applications, e.g. part of the navigation system of ballistic missiles. IMUs are used for the study and analysis of human motion, with their suitability depending on the particular application, e.g. clinical applications involving walking [46]. For a more detailed review of accelerometer and gyros technologies the reader may refer to [42]. Due to the wide range of applications that employ IMUs with different performance requirements, IMU performance grades may be categorised into consumer or automotive, tactical, intermediate, aviation, and marine grades, as detailed in [3], although there is no uniform definition of these categories among authors.

The minimum number of sensors required to obtain a 3D inertial position solution is a triad of accelerometers and a triad of gyros, provided that the initial position and orientation of the sensor w.r.t. a reference frame are known. A temperature sensor is used for calibration of the inertial sensors' raw/analog-to-digital (A/D) outputs. An IMU processor converts the raw sensor readings to physical units using the device's calibration parameters, usually embedded by the manufacturer in the non-volatile memory of the IMU, although they can be applied during the post-processing. The IMU processor also synchronises the inertial sensors' outputs using an internal clock as time reference (typically a crystal oscillator), as illustrated in Figure 2.7. Some IMU configurations employ additional sensors, such as a triad of magnetometers to initialise the attitude of the device w.r.t. ECEF, or a barometer to aid the height calculation. Details about how integrated navigation systems work are discussed in Section 2.2.4.

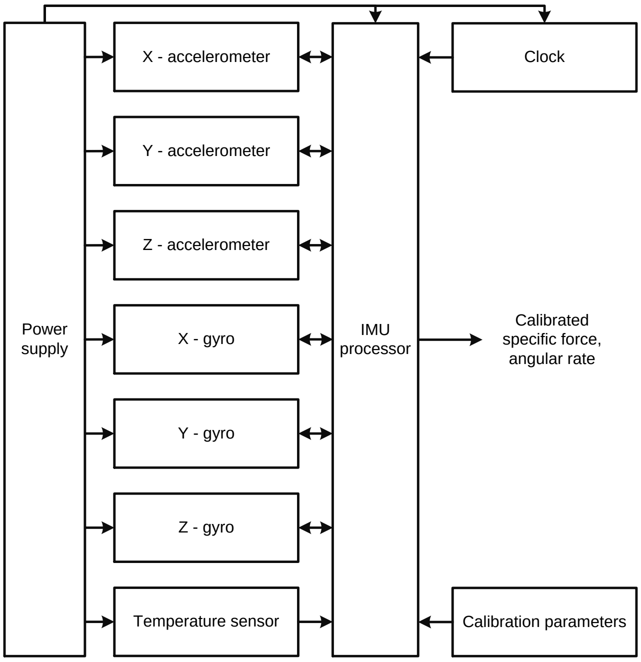

Figure 2.7. Generic IMU diagram (after [3]).

Accelerometers measure specific force (units in m/s$^2$), which is the force per unit mass exerted on a body mass without any gravitational components, e.g. for a free-falling object in vacuum, specific force is zero as only gravitational forces act on it. For a static body on a desk near the Earth's surface, the specific force acting on the object is the reaction force from the desk which acts against gravity; therefore, specific force will have equal magnitude and opposite direction to the gravity vector of that object, noting that the gravity vector encompasses the centrifugal force due to the Earths' rotation.

To understand better the way accelerometers measure specific force, let's imagine a typical accelerometer that comprises a fixed chassis (i.e. the body frame) with an enclosed proof mass (of known mass) inside an electromagnetic (E-M) field. The chassis movement (along the accelerometer's sensitive axis) and the proof mass inertia result to a relative movement between them (the proof mass will appear to be moving at an opposite direction w.r.t. the chassis), so the E-M field changes (the proof mass material has specifically been chosen by the manufacturer to affect the E-M field). Pick-off sensors sense this E-M field change, which is translated into voltage, measured by the accelerometer's pick-offs. A feedback loop compensates the proof mass displacement within the chassis by applying a restoring voltage to keep the proof mass in the centre of the chassis. A triad of accelerometers is needed to measure specific force in 3D. The additional voltage required to offset the movement of the proof mass (which is the output of the feedback loop), is integrated and dumped over a specific time interval and the analog output is then digitised by an analog-to-digital converter (ADC), typically as an integer value. This digital reading will then be fed into the IMU processor for calibration and conversion to physical units of specific force or integrated specific force (or " delta-v's ", units in m/s), depending on the IMU design. A single accelerometer typically senses the displacement of a proof mass along one direction, so three accelerometers, with their sensitive axes placed perpendicularly w.r.t. each other are needed to sense linear (translational) motion in 3D.

Early gyros using MEMS technology, measure angular rate (units in rad/s) about a sensitive axis using a vibratory element under applied voltage, e.g. a string, a tuning fork or cylinder [47], mounted along the sensitive axis. The vibration of the mounted element takes place in the plane defined by the sensitive axis and the perpendicular drive axis, i.e. the axis along which it vibrates. This is like a guitar string vibrating on a specific plane. If the vibratory element oscillates on the XZ plane, then a rotation of the gyroscope chassis (w.r.t. inertial frame) causes a position displacement of the vibratory element along the Y axis due to Coriolis acceleration, typically sensed by the gyroscope pick-offs and a feedback loop applying an opposite voltage to eliminate that displacement and restore the vibration on the XZ plane. This additional voltage is measured over a time interval and the analog value is input to the ADC for digitisation. This digital reading it is then fed to the IMU processor for calibration and conversion to physical units, e.g. angular rate or accumulated angular rate (attitude increment or "delta-θ"s, measured in rad), depending on the IMU design. A single gyro typically senses the rotation of a vibratory element about a single sensitive axis, so three gyros placed orthogonally w.r.t. each other are needed to sense rotations in 3D.

### *Errors*

Inertial sensors' measurements contain deterministic (or systematic) errors and noise. The systematic errors are due to the physical limitations of inertial sensors and can be calibrated. Noise can only be identified and characterised, e.g. using Allan variance analysis [48]. This type of analysis represents in a diagram the root mean square error (RMSE) in the vertical axis and the averaging time intervals on the horizontal axis (also called Sigma-Tau diagrams). The inverse of the averaged time interval is frequency, so these diagrams provide valuable information about what type of noise dominates a specific frequency band, e.g. quantisation noise may be the dominant underlying noise process over a specific frequency band, while bias instability or random walk dominate over another. The main types of errors which inertial sensors may exhibit, are listed below [3]:

- *Bias* errors are deterministic encompassing a static and a dynamic component and are independent of the underlying specific force or angular rate (in other words they are "g-independent"). The static component is constant throughout the IMU operating period and varies from run to run. The dynamic component (which is typically about 10% of the static one) varies over periods of a few minutes and is temperature-dependent. Some manufacturers use the term bias stability to characterise the variation of bias over time of the calibrated analog-to-digital (A/D) readings from the IMU processor [49].
- *Scale factor* errors (deterministic), which incorporate the deviation of the input-output gradient of the sensor from unity. These errors are proportional to the true underlying specific force or angular rate. It is worth mentioning that scale factor errors are non-linear and asymmetrical around zero point, although for simplicity the scale factor error can be approximated by a linear equation between input and output measured quantities.
- *Cross-coupling* errors (deterministic), are errors proportional to the specific force and angular rate, from axes orthogonal to the sensitive axis of an accelerometer or gyro, respectively, so they are also called cross-axis sensitivity. This could be caused due to e.g. vibration in vibratory inertial sensors; however, one major factor that causes cross-coupling errors is that the mounting of the sensors is not exactly orthogonal w.r.t. the IMU frame, due to manufacturing limitations. It follows that misalignment errors create additional scale factor errors, which are typically two to four orders of magnitude lower than misalignment errors. These misalignment errors are deterministic, since the misalignment is measurable and known through the manufacturer's calibration parameters; therefore, misalignments can be compensated through calibration.
- *Random noise* (stochastic), e.g. electrical noise in the IMU circuit which limits the resolution of the inertial sensor. The random noise standard deviation (SD), is usually represented by the square root of the noise power spectrum density multiplied by the IMU output rate; therefore, the noise SD, is proportional to the square root of the output rate of the IMU [49].
- *Vibration-induced* errors, as a result of external sources (e.g. pedestrian motion) at high frequencies near the resonant frequencies of the inertial sensors. Due to the non-linearity and asymmetry of the scale factor and misalignment errors, vibration-induced errors leave a remaining component called vibration rectification error.

Another source of error for accelerometers is the *size effect*, which can be easily calibrated through. The size effect is due to the fact that their origin points cannot be collocated at the nominal reference point for the whole triad. This causes centripetal accelerations to be sensed, which are a function of the distance between each accelerometer's origin and the nominal point of reference. The size effect is deterministic and can be calibrated, since the origin positions of the accelerometers w.r.t. the nominal point of reference are known by the manufacturer. The size effect does not constitute a problem for gyros as the angular rate they sense is invariant to their origin point's position along the sensitive axis [3], similar to a rotating rod whose points exhibit different linear but the same angular velocity.

Gyros are affected, however, by *g-dependent* deterministic errors, since their vibratory elements are sensitive to the underlying specific force due to imbalances of their masses. Other types of errors affecting both accelerometers and gyros stem from the quantisation process (stochastic in nature) during converting the analog outputs to digital, as well as externally-induced vibration near the internal resonant frequencies of the inertial sensors. It is worth noting that all inertial sensor errors are referenced to an inertial frame and resolved in body-frame axes.

### *Inertial navigation equations and error propagation*

The inertial navigation processor applies the inertial navigation equations using information from the calibrated specific force and angular rate measurements of the IMU (and any other aiding information from other sensors), and the initial conditions of the navigation solution, e.g. position, velocity and attitude of the body-frame w.r.t. the reference frame. The execution of the inertial navigation equations is an iterative process applied at discrete time instants (epochs), with next epoch's solution depending on the previous one for first-order systems. It follows that the accuracy with which the navigation solution is known at initialisation is crucial. Due to the iterative nature of inertial navigation equations implementation, any errors from previous epochs can propagate over the next epochs. This error propagation is known as drift of the inertial navigation solution with its magnitude depending on the INS design and quality of internal sensors used. For best results, other sensors which are not subject to similar types of errors at the same experimental conditions need to be employed by means of a fusion algorithm, e.g. Kalman Filter, which is discussed in Section 2.2.4.

Depending on the design of the INS, the inertial navigation solution may only use new information from the inertial sensors for each epoch (and weight appropriately information from previous measurements). In that case, the update rate of the navigation solution will be equal to the lowest sampling rate $f_{out}$ (Hz) employed by any of the used inertial sensors, depending on their integration time interval. If the navigation solution is meant to provide outputs at a lower rate than $f_{out}$, if the required processing power is not available.

Inertial navigation equations are updated by the following four steps [3]:

- Attitude update of the body-frame w.r.t. a preferred reference frame, by integrating the angular rate.
- Specific force obtained from the accelerometers transformed from body-frame resolving axes into a preferred resolving frame, using the attitude solution.
- Velocity update which involves calculating the true acceleration of the body-frame, by adding any gravitation or gravity to the specific force measured by the accelerometers. This uses a gravitation model if the navigation solution is referenced to inertial frame, or a gravity model (including gravitational and centrifugal acceleration on components) if the navigation solution is referenced to a rotating frame, such as the Earth frame. The true acceleration is then integrated over the update step time interval to obtain an updated velocity solution.
- Position update, by integrating the velocity solution over the update step time interval.

The implementation of inertial navigation equations in the discrete time domains, implies that prior and updated position, velocity and orientation are valid at specific time *instants*, while angular rate (or "delta-θ"s), specific force (or "delta-v"s) and acceleration represent the average values over the time *interval* for each inertial navigation equations' iteration. The time validity of these quantities is illustrated in Figure 2.8. It is worth noting that the Xsens MTi-G IMU/GPS used to capture human motion in this thesis (see Chapter 3), outputs angular rate, specific force and magnetometer outputs in a time-stamped data packet, along with position, velocity and attitude for consistency, so the time of validity for all these quantities, refers to the timestamp (i.e. a time instant) of that data packet.

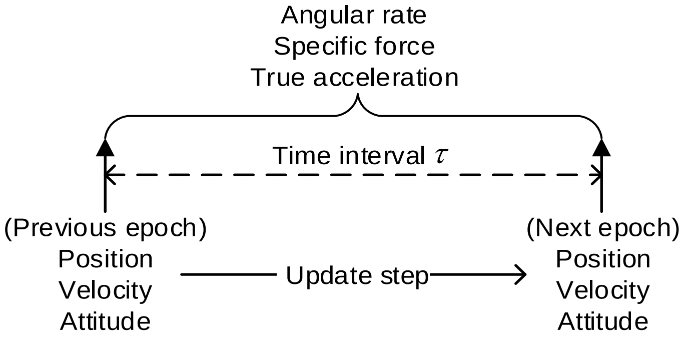

Figure 2.8. Time of validity of inertial navigation parameters (from [3]).

The calculated inertial navigation solution contains errors as a result of inertial sensors' measurement errors, initialisation errors and processing approximations. The latter category encompasses errors due to quantisation, internal clock errors and time referencing between hardware components, as well as coning and sculling errors due to undetected out-of-phase high-frequency vibration along orthogonal axes between two gyros or a gyro and an accelerometer, respectively [3], [42]. Un-calibrated errors occurring in one step of the update process can have an effect on other steps, e.g. attitude drift will result in inaccurate application of the gravitation/gravity model to transform specific force and, as a result, velocity and position drift.

As an example, assume a body-frame (XYZ) travelling towards north along a straight line at a constant speed with its attitude constantly aligned with the NED frame and that the gravity vector remains unchanged for the duration of this motion. Assuming also perfect accelerometers, specific force should be sensed only along the down axis, opposite to the vector of gravity. However, if the actual and assumed attitude differs, e.g. due to tilt errors (roll and pitch errors), the inertial navigation equations will apply the gravity correction using the incorrectly observed attitude, i.e. coupling the gravity correction along the North and East axes, which will manifest as a constant acceleration bias along these axes. In turn, this acceleration bias will propagate to the next steps causing a linear velocity error which grows as a linear function of time and a position error which grows as a quadratic function of time. If the gyros measuring roll and pitch in the above example are also subject to a linear attitude drift, then they will cause a quadratic velocity error and a cubic position error. It follows that in the case of the gyros' errors, more sophisticated calibration is required in order to compensate for those higher-order errors in the position and velocity solutions.

### 2.2.3 Global Navigation Satellite Systems (GNSS)

GNSS encompass three segments: space, ground and user/receiver equipment. GNSS receivers depend on existing infrastructure to receive signals transmitted from satellite constellations, e.g. the US GPS, the Russian GLONASS, European GALILEO, Chinese BEIDOU, Japanese QZSS and Indian IRNSS. The different GNSS space segments (constellations) are operated by ground control stations, who upload the satellites with updated information enabling them to broadcast it in their navigation messages (called *ephemeris*) which are modulated on specific carrier frequencies, e.g. L1, L2, L5. The user equipment receives and decodes these messages to obtain information about the satellites' motion (e.g. position) and status. By combining this information with (at least) four pseudo-range measurements, obtained from the receiver's code-tracking function (see Section 2.2.3.4), the receiver can calculate its position, velocity and timing (PVT) solution. GNSS navigation messages can be unencrypted (civilian) and available to all users, or encrypted so only authorised users can decode them, e.g. military. GNSS signals may also be assisted by additional ground-based or space-based augmentation systems (GBAS and SBAS, respectively), to improve PVT performance in terms of accuracy, integrity continuity and availability. For a more detailed discussion of GBAS and SBAS the reader may refer to [50], [51].

#### 2.2.3.1 Overview

The accuracy of a GNSS PVT solution depends on many factors, e.g. satellite constellation health status and geometry, signal-in-space error (mainly due to satellite clocks), space weather affecting ionospheric propagation delays, atmospheric scintillation (more commonly around equatorial regions), variations in tropospheric propagation delays, multipath, signal refraction (due to ionosphere/troposphere or around edges of buildings), diffraction and non-line-of-sight (NLOS), as well as interference which may be due to unintentional interference, or intentional jamming and spoofing. At the user side, contributing factors to GNSS PVT errors are receiver design and relative motion dynamics along the line-of-sight (LOS) between the GNSS equipment and the satellites. For indoor environments, GNSS signal availability is more problematic, as the signals are attenuated by building structures, e.g. walls, with multipath and NLOS effects increased.

The basic principle of GNSS operation is that each satellite vehicle transmits a unique PRN (ranging) code, and a navigation message with its orbital parameters (from which the receiver calculates the satellite's coordinates), uploaded to the satellite by a ground control station, and a timing reference, provided by a set of atomic clocks mounted on its platform. The PRN code is received and processed by the GNSS receiver which then estimates the distance (pseudo-range) from the respective satellite by comparing the ranging code of the incoming signal with its own internal timing PRN code reference, typically a crystal oscillator, given that the signal propagation speed is known. To obtain a PVT solution or "fix", at least four pseudo-range measurements are needed, three for estimating roughly the 3D position of the receiver (via trilateration) and another one to correct for timing discrepancy (receiver clock bias) between the receiver and satellite clocks, which is contained in the pseudo-ranges. More than four satellites are needed in order to provide better accuracy (depending on satellite geometry), redundancy and employment of advanced techniques for signal integrity-checking, e.g. Receiver Autonomous Integrity Monitoring (RAIM) which detects and excludes a satellite which degrades the navigation solution, noting that at least five satellites are required for detection and 6 for exclusion, in order to provide redundancy for service continuity after the satellite exclusion.

Typically, the mean accuracy of GNSS equipment inside smartphones is about 5 meters [52] under open-sky conditions outdoors, with variable signal availability in urban areas, as well as indoors [53]. GNSS receivers' typical output rate is limited to 10Hz [54], therefore, inertial sensors, or other technologies which exploit appropriate signals (e.g. wireless local access network – WLAN), need to be integrated with GNSS sensors in order to increase the navigation solution availability in more challenging areas. GPS uses the WGS84 coordinate reference frame which is an ECEF (rotating) frame, as it is easier to define a user's position and velocity with respect to the Earth's surface (WGS 84 ellipsoid). Figure 2.9 illustrates how a generic GNSS receiver works, with each of these functional blocks explained in the remainder of this section.

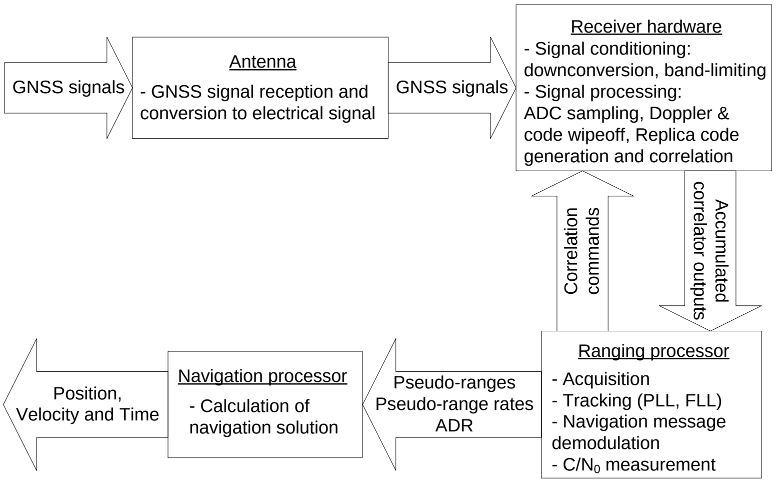
GNSS signals
Antenna
- GNSS signal reception and conversion to electrical signal
GNSS signals
Receiver hardware
- Signal conditioning: downconversion, band-limiting
- Signal processing: ADC sampling, Doppler & code wipeoff, Replica code generation and correlation
Correlation commands
Accumulated correlator outputs
Ranging processor
- Acquisition
- Tracking (PLL, FLL)
- Navigation message demodulation
- C/N0 measurement
Pseudo-ranges
Pseudo-range rates
ADR
Navigation processor
- Calculation of navigation solution
Position, Velocity and Time

Figure 2.9. GNSS receiver functional diagram.

### 2.2.3.2 GNSS signals and antenna

A generic GNSS signal comprises a ranging code and a navigation message modulated over a carrier signal transmitted at (approximately) 3 central frequencies per GNSS, e.g. in L1/L2/L5 frequency bands for GPS, L1/L2 frequency bands for GLONASS and E1/E5/E6 frequency bands for Galileo, to allow for better calibration of the ionospheric delay and greater robustness against interference [3]. A pseudo-random noise (PRN) code is modulated on GPS signals using code-division multiple access (CDMA) in order to allow different satellites to transmit in the same band without interfering with each other and also, spreading the signal spectrum to propagate better through the atmosphere. GNSS signals transmitted on different central frequencies exhibit different atmospheric refraction properties, so a GNSS receiver which operates in at least two different bands, e.g. L1 and L2, can compensate for most of the ionospheric delay of the signals [55]. For further reference to GNSS signals structure (including sub-carriers) and recently introduced GNSS signals the reader may refer to [3], [56], [50].

Reception of transmitted GNSS signals depends on the nature and density of obstructions along the line-of-sight (LOS) between the receiver and the satellite as well as the antenna design. Tall buildings in urban canyons or multiple layers of indoor walls, for example, can obstruct GNSS signals completely. Attenuation and refraction of the GNSS signals crossing the atmosphere depend on the satellites' elevation angle, i.e. how much above the horizon they are. With regards to pedestrian applications, GNSS signal observability depends on the antenna location on the user's body, due to body-masking effects [57], and signal attenuation due to human body segments dielectric properties [58]. When GNSS E-M signals are received by the receiver's antenna, they are converted to electrical current for input to the receiver's circuitry. The antenna design therefore should allow maximum gain of GNSS signals at different elevation angles.

### 2.2.3.3 Receiver hardware

The function of the receiver's hardware can be divided to signal conditioning, taking place in the analog domain and signal processing, occurring in the digital domain.

#### *Signal conditioning*

The main aim of signal conditioning is to prepare the GNSS signals for digital conversion allowing more robust signal processing. The converted GNSS electrical signals are conditioned by the front-end receiver hardware which encompasses amplification, and at least two stages of downconversion to lower (intermediate) frequencies in order to reduce the processing load with the receiver. After downconversion, the signals are filtered to remove created harmonics and out-of-band interference and amplified before entering the automatic gain control (AGC) loop which adjusts the signal strength in order to maintain a constant level of noise. The AGC function is similar to adjusting the volume between tracks when listening to a music album. After the signal amplitude is adjusted by the AGC, the signal enters the analog-to-digital converter (ADC) to be digitised. It is worth noting that throughout the signal conditioning process, the spectral characteristics of the incoming signals, e.g. Doppler shift, do not change, but are shifted onto intermediate frequencies' (IF) carriers [3].

#### *Signal processing*

The ADC samples the analog signal to convert it to digital at a specific rate. The pre-correlation bandwidth must be at least two times higher (Nyquist frequency) than the chipping rate of the PRN code modulated on the incoming signal [59], in order to contain enough information to reconstruct the original waveform. Higher pre-correlation bandwidths may provide better signal acquisition and tracking but they introduce extra processing load to the receiver; therefore, these two factors need to be balanced appropriately. Other considerations for ADC designers are that the sampling rate which needs to be asynchronous with the chipping rate of the PRN code, the rate of the navigation message and the IF bandwidth to cover properly the incoming signal envelope and the quantisation level (ADC resolution) [60].

The next stage of signal processing is the prompt sampling of the incoming GNSS signal in-phase ($I_P$) and quadraphase ($Q_P$), i.e. with a phase delay of $\pi/2$ from the in-phase signal, noting that these are components of the signal processing sampling process inside a GNSS receiver, which are different to the C/A and P(Y) quadrature signals (i.e. with a 90° phase difference) transmitted by GPS satellites. This process is similar to expressing a 2D point in x and y Cartesian coordinates. The $I_P$ and $Q_P$ vector sum provides the amplitude of the sampled signal while the arctangent of their ratio ($Q_P/I_P$) provides the phase of the signal [40]. The prompt I and Q signals, which are two sinusoidal signals with 90° phase difference between them, are correlated via correlators with replica prompt I and Q signals, generated locally by the receiver and controlled by the numerically controlled oscillator (NCO), noting that the NCO is part of the baseband signal processor, controlled by the ranging processor. By changing the frequency only of the replica prompt I and Q signals, the ranging processor can effectively control the frequency and phase of the generated prompt replica signals in order to align them with the incoming ones. This stage is called Doppler-wipeoff or carrier-wipeoff, as the carrier signal (including the Doppler shift which is measured in the ranging processor), is removed from the $I_p$ and $Q_p$ signals, leaving only the modulated PRN code and navigation message, along with the noise.

The receiver also generates prompt, early and late PRN code replica signals, whose frequency is also driven by an NCO, different to the one driving the carrier tracking loops and noting that this NCO is also part of the baseband signal processor, controlled by the ranging processor. The early and late replica PRN code spacing may be between 0.1 and 1 chip long, i.e. narrow and standard correlator spacing respectively [61]. The receiver correlates the replica PRN codes with the incoming codes in the prompt, early and late I and Q signals, which is known as code wipeoff and takes place in the GNSS receiver's signal processor over the coherent time interval [3], [40], [2]. A perfect alignment of the incoming and replica GNSS signals, results in a complete removal of the PRN code from the GNSS signal (code wipe-off). The incoming GNSS signal still contains though the modulated navigation message, which in the case of L1 C/A code is modulated at a rate of 50bit/s, using a binary phase shift keying (BPSK) modulation, i.e. a navigation bit is modulated by altering the signal phase by π rad.

The outputs of the correlators are then accumulated over the coherent time interval (typically 20ms). These correlator-accumulated outputs are, perhaps a bit ambiguously, also known as "Is and Qs" or "I and Q signals". The code tracking loop acts to equalise the observed power in the early and late correlators, which is equal to the sum of the Is and Qs squares. Typically, an accumulation time interval is 20ms [3], which is the maximum time interval that can be used to avoid navigation data bit transitions (modulated at a 50bit/s rate). The accumulation time interval needs to be selected carefully by the receiver designer to ensure a good signal to noise ratio over the accumulation time interval and that data bit transitions are detected properly [3]. If the accumulation spans over two navigation bits then, if the navigation bits having opposite signs, the accumulated signal power will be minimised and the output Is and Qs may be too weak for further processing.

During this stage, the signal processor also uses the Is and Qs to calculate the signal carrier power to noise density ($C/N_0$) in dB-Hz, noting that it needs to be measured in advance of signal acquisition and tracking, in order to inform the ranging processor. This is the ratio between the received carrier power over the noise power density (noise power per Hz of signal bandwidth) of the expected received signal under optimal conditions. $C/N_0$ describes the ratio of the signal strength divided by noise power per Hz of signal bandwidth (and measured in Hz), i.e. it does not depend on the bandwidth of the filters that the signal goes through in the receiver's circuitry, contrary to the signal carrier power to noise ratio (SNR) which changes depending on the bandwidth of the last filtering stage [62].

### 2.2.3.4 Ranging processor

To maintain a precise alignment between the incoming and replica signals and allow correct demodulation and time of arrival estimation of the navigation message, the receiver needs to continuously monitor the frequency (Doppler) and code phase of the incoming signal. The ranging processor aids this process by estimating the time delay (phase) between the transmission and arrival of the PRN code, at the satellite and GNSS receiver antennas, respectively, as well as the frequency shift (Doppler) of the PRN code and its carrier. For more precise positioning, the carrier phase shift is also estimated by dedicated carrier tracking loops. The ranging processor uses the Is and Qs of the conditioned signal to estimate the pseudo-range (code delay), pseudo-range rate (Doppler shift) and carrier phase (carrier phase shift). Alternatively, some receivers may output delta range (integral of pseudo-range rate since the last measurement) or accumulated delta range (also known as carrier phase) which is the carrier phase observable equivalent in the range domain [3].

#### *Acquisition*

The ranging processor needs first to estimate roughly (initialise) the code frequency and phase delay and then fine-tune them for more precise alignment between the incoming and replica signals. To achieve this, when the receiver is switched on, it selects from an array a combination (called a cell) of candidate Doppler frequencies and code phases in order to generate a replica PRN code signal, which then correlates with the incoming (digital) signal, aiming for maximum correlation. This process is called acquisition and its aim is to provide the maximum correlation between the incoming code signal and the replica code in order to initiate properly the code and carrier frequency tracking [61].

The ability of a receiver to acquire an incoming signal quickly and robustly depends on the $C/N_0$, processing power limitations (e.g. number of correlation channels of the size of an FFT-based acquisition engine), as well as the travelled path of the signal, the characteristics of the receiver's clock (reference oscillator) and the relative dynamics between the receiver and the satellite [3]. During acquisition, signal detection involves comparing the sum of the Is and Qs squares with a specific threshold, which can be lowered in order to ensure maximum sensitivity.

#### *Tracking*

Following the acquisition of a GNSS signal, the receiver needs to maintain maximum correlation (alignment) between the received and internally generated PRN codes, as well as compensate continuously for the Doppler shift in the code and carrier frequencies. This process is called tracking and it can be divided into code and carrier tracking. Tracking is normally implemented via loops with variable gain depending on $C/N_0$ which use the accumulated correlated outputs (Is and Qs) from the signal processor in order to track code, carrier frequency and phase errors between the incoming and replica signals, e.g. [63]. Carrier tracking can also be based on Fast Fourier Transform (FFT) algorithm to track carrier phase in the frequency domain in the case of static receivers [64]. Other methods employ a Kalman Filter instead of tracking loops [65][66][67], but they are out of the scope of this thesis.

Code-tracking is an iterative process that takes place inside a delay lock loop (DLL) which tries to minimise the code phase error between the incoming and replica PRN codes during the correlation time interval, typically 20ms. A generic DLL compares early and late accumulated correlated outputs (Is and Qs) from the signal processor to measure the current code phase error. Then, using the code phase error measurement, the DLL loop filter down-weights the code phase error measurement using the loop gain and then used this to estimate the current code phase of the incoming signal, i.e. the timing of signal arrival, and also predict the code phase of the next iteration (epoch). This code phase prediction informs the NCO to control the frequency of the generated replica PRN code in the signal processor and maintain better alignment between the incoming and replica PRN codes. The DLL bandwidth is typically between 0.05Hz and 1Hz, which refers to the interval over which the commands to the NCO are statistically independent, noting that the commands (code phase predictions) to the NCO are typically at a rate of 50Hz (or higher) [39]. Lower bandwidths are typically used for geodesy and surveying applications and higher bandwidths for kinematic applications, e.g. pedestrian navigation or location based services (LBS).

To explain how a DLL works, the Is and Qs from the signal processor, i.e. the part of the GNSS receiver between the ADC and the generation of Is and Qs feeding the ranging processor as shown in Figure 2.9, are input to the DLL's code phase discriminator which then associates them with a code phase error measured in chips, which may be normalised taking into account the measurements' standard deviation, the accumulation coherent time interval and $C/N_0$ [3]. A typical coherent time interval for the accumulators is 20ms (i.e. the time interval between navigation bits transitions at 50bit/s), allowing for a maximum tolerable pseudo-range rate error of 4.2m/s in the L1 band [3].

The time interval over which the code phase discriminator function accumulates the Is and Qs may be fixed, or variable – depending on the observed $C/N_0$ by the signal processor. The estimated code tracking error is then used to update the current code phase estimate in the DLL loop filter, in order to estimate the time difference between the code NCO time and the current receiver time (which is used to calculate the pseudo-range) and, along with Doppler shift information from carrier tracking aiding, to predict the code phase for the next DLL iteration. The code phase prediction is differenced with the current code phase estimate and sent to the NCO of the receiver to control the replica code generation frequency, noting that this NCO command correction will be applied in the next iteration. The loop gain determines the magnitude of the response of the DLL to the code phase discriminator output (code phase error estimate), which is down-weighted in order to estimate the current code phase. The higher the loop gain, the faster the tracking loop will respond to changes, but the more sensitive it will be to tracking noise, as the input Is and Qs are less smoothed. The adjustment of DLL bandwidth needs to be optimally balanced between noise reduction and fast response to dynamics.

For a code tracking bandwidth of $B_{co}$ = 1Hz, i.e. fast response of the code-tracking function to motion dynamics and a maximum recoverable error of $x$ = 0.5 chip, the pseudo-range rate tolerance (difference between the actual and expected pseudo-range rate) along the LOS is about 117.3m/s, using [3]:

$$x=\frac{f_{co}}{4B_{co}c}\Delta\dot{\rho}\Leftrightarrow\Delta\dot{\rho}=\frac{4B_{co}c}{f_{co}}x \quad , \tag{2.3}$$

where $\Delta\dot{\rho}$ the range rate error, $x$ the code phase tracking error in chips, $f_{co}$ the code frequency (1.023 Mchip/s), $c$ the speed of light in vacuum, and $B_{co}$ the DLL bandwidth. which can be considered non-realistic for pedestrian applications. Therefore, it can be safely assumed for the purposes of this thesis that typical pedestrian motion, considered for the purposes of this thesis, cannot introduce any errors due to dynamics stress to the code tracking loop.

Carrier tracking of GNSS signals encompasses frequency and phase tracking by respective frequency lock loops (FLLs) and phase lock loops (PLLs). In this thesis, (see details in Chapter 4) they are used independently, i.e. the FLL does not providing any aiding information (Doppler) to the PLL. The aim of carrier tracking is to estimate accurately the carrier Doppler shift and, in some receivers, the carrier phase of the incoming signals, in order to drive more precisely the NCO and generate replica code and carrier signals that align better with the incoming signal. Carrier tracking is normally implemented via tracking loops, although other methods may encompass a Kalman Filter [65][66][67][68][69][70], or Fast Fourier Transform algorithm to track carrier phase and frequency in the frequency domain [64].

FLLs track carrier frequency (equivalent to pseudo-range rate in the time-domain). Typically, FLLs are more robust than PLLs in terms of tolerance to motion dynamics along the line-of-sight (LOS) between the user equipment and the GNSS satellite, as well as weak $C/N_0$ [3], [40], [2]. However, PLLs allow more precise estimation of the carrier phase and the time of the GNSS signal arrival. Due to their different design and performance characteristics, some GNSS receivers use a PLL aided by an FLL, in order to maintain lock of the signal [64],[71]. First-order FLLs can track carrier frequency error only, assuming it remains constant over two subsequent epochs and second-order FLLs can track carrier frequency and frequency rate of change errors between received and internally generated signals, i.e. they can predict the frequency error in the next epoch by using the frequency and frequency rate measurements of the previous one, which makes them more robust then PLLs in terms of retaining lock of signal.

First-order PLLs cannot be used for GNSS as they require high $C/N_0$ to support the bandwidth for tracking the phase changes between the received and internally generated signals, due to the relative motion dynamics between the receiver and the satellite along the LOS and the receiver clock jitter. Second-order PLLs can track carrier phase (range) and frequency (range rate) errors, while third-order PLLs can track carrier range, range rate and range acceleration (frequency rate of change) errors, which makes them more appropriate for GNSS signal tracking.

The purpose of a PLL is to estimate the carrier range error and inform the code and carrier NCOs which control the frequency of the replica Is and Qs generation in order to achieve better alignment with the received signals over the coherent correlation time interval [3]. An example of a third-order PLL is illustrated in Figure 2.10. The way it works is that the correlator outputs (Is and Qs) are fed into the carrier range (phase) error discriminator that estimates the carrier range error, and then normalises it depending on the measurement standard deviation, the correlator accumulation time interval $\tau_\alpha$ and the $C/N_0$ [3], [40]. The Is and Qs accumulation interval, prior to enter the carrier range discriminator, defines the PLL bandwidth, i.e. the rate at which the PLL updates the carrier NCO.

The carrier range error estimate is then smoothed inside the PLL filter and used to update the current carrier range, range rate and range acceleration estimates, noting that the range acceleration estimate in a third-order PLL remains unchanged between subsequent loop iterations (although being updated during these iterations). The carrier range error from this stage can also be fed into the ranging processor for the accumulated delta range (ADR) or carrier phase calculation. The carrier range-rate error can also be fed into the DLL to estimate the code phase and inform the pseudo-range-rate calculation in the ranging processor. In the next processing stage, the (current) range estimates are used to predict the range and range rate that will inform the PLL filter in the next loop iteration, i.e. after a lag of a $\tau_\alpha$ time interval. The range predictions are fed into the carrier NCO and used to generate the replica prompt, early and late I and Q signals entering the correlation function of the signal processor.

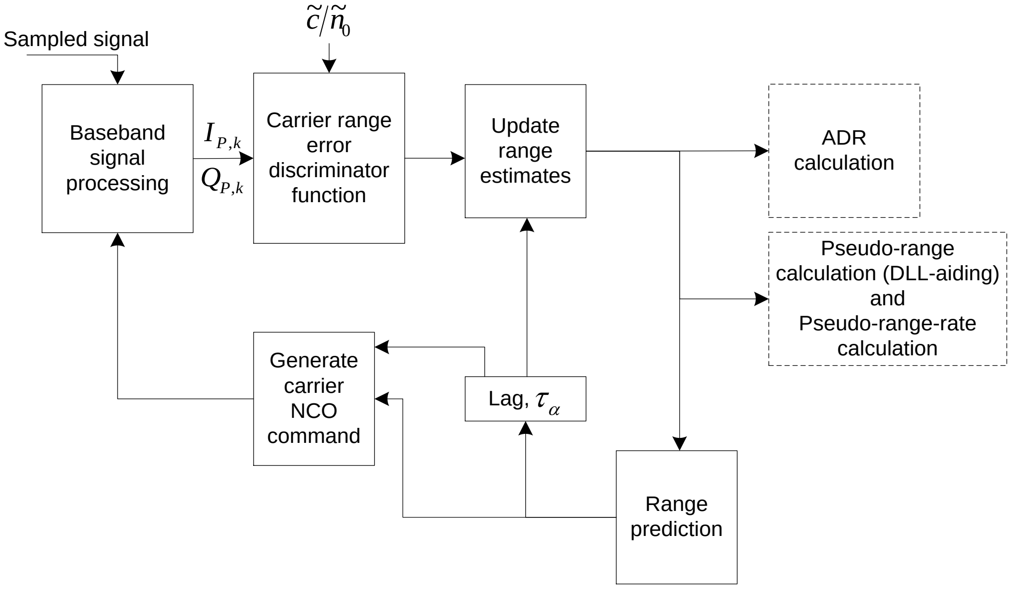
Sampled signal
$\tilde{c}/\tilde{n}_0$
Baseband signal processing
$I_{P,k}$
$Q_{P,k}$
Carrier range error discriminator function
Update range estimates
ADR calculation
Pseudo-range calculation (DLL-aiding) and Pseudo-range-rate calculation
Generate carrier NCO command
Lag, $\tau_\alpha$
Range prediction

Figure 2.10. An example of a phase lock loop (after [3]).

Although a PLL allows for precise estimation of the incoming GNSS signal's carrier phase, it is less robust in terms of sensitivity to environmental radio-frequency (RF) noise and dynamic stress [3], [40], which can introduce phase and frequency steps exceeding the input range of the carrier phase and frequency discriminators, respectively, causing tracking slips (cycle slips and frequency false locks, respectively), which, in extreme cases, can cause the receiver to lose track of a satellite signal. The error sources of carrier tracking loops (PLL/FLL) can be summarised to the following three categories [72]:

- Vibration-induced phase noise to the NCO; this can be internal noise cause by frequency standard instabilities of the reference oscillator. These effects are also explained in [2].
- Thermal noise, which is always present in electronic circuits and is independent of the carrier tracking loop order.
- RF noise, which includes the other GNSS signals on the same frequency and other interference sources. A more detailed study of the effects of thermal noise and oscillator phase noise to the performance of PLLs is presented in [73].
- Dynamic stress error due to the relative motion between the satellite and the GNSS receiver which tracks the signal of that satellite. This depends on the order of the PLL/FLL, with second-order FLLs and third-order PLLs sensitive to jerk stress (range acceleration rate of change over the time unit). This is because second-order FLLs and third-order PLLs, do not update a range jerk estimate in their loop filter, as a higher-order FLL/PLL would be required for that. Therefore, these carrier tracking loops cannot predict the rate of change of range acceleration in the next loop iteration. In other words, they only update the range acceleration

estimate (which is their loop filters response to the range acceleration), but they do not predict the range acceleration onto the next loop iteration.

It is also worth noting that attenuation of the received signal, increases the receiver's sensitivity to both RF and thermal noise, as it "buries" the signal deeper into the noise. Vibration-induced and multipath effects may also affect carrier tracking loops' performance [2], but they remain out of the scope of this thesis, as it is not expected that for pedestrian applications requiring tracking of LOS motion dynamics, the receiver design would need to be set into a higher tracking bandwidth in order to respond fast to the changing LOS phase and frequency errors (induced due to dynamic stress). Typically, PLL carrier phase errors are in the order of 1.4mm (1$\sigma$) under good $C/N_0$, e.g. above 40dB-Hz, for a 20Hz PLL effective bandwidth and a $\tau$=10ms correlators' accumulation time interval [40].

A PLL drives the NCO with a phase (and sometimes frequency) estimate in order to aid the input and replica code alignment in the signal processor, noting that often the NCO command derives from tracking loops' change of phase (or code). To measure the current phase error, i.e. the range error in the time-domain, the PLL uses a range error discriminator which compares the input and replica GNSS signals and estimates their range difference (error). A special case is Costas discriminators which have half of the typical range error input range, i.e. half of a carrier wavelength instead of a wavelength, so they are not sensitive to navigation bit transitions that invert the phase of the incoming signal by $\pi/2$, i.e. half of a wavelength.

The gain of an arctan discriminator (slope of the phase error input-output curve) is linear over the carrier tracking error input range, as illustrated in Figure 2.11, therefore it can be considered more robust than other types of discriminators. If the carrier range tracking error exceeds the pull-in range of a Costas discriminator, it will cause the discriminator to "jump" an integer number of cycles, due to the cyclic nature of the discriminator, and incur a range error which is a multiple of half of a carrier wavelength for a Costas (two-quadrant) arctan discriminator, as shown in Figure 2.11, or a multiple of a carrier wavelength for a four-quadrant arctan PLL discriminator [3], [40]. This error at the discriminator output in the presence of a range error exceeding its pull-in range is called a cycle slip. Cycle slips may occur due to scintillation, low $C/N_0$ conditions or antenna motion dynamics, with the oscillator jitter contributing as well but not causing cycle slips on its own. Cycle slips affect the navigation message demodulation and carrier-based ranging [3] which are critical for precise positioning. It is worth noting that if the receiver loses lock of a signal, then until the signal is tracked again, many cycle-slips will have occurred, due to the change in pseudo-range between the receiver and the satellite. Cycle slips may also occur due to reference oscillator errors, which are out of the scope of this thesis, e.g. depending on temperature variations, crystal physical characteristics and aging, sensitivity to specific force causing frequency variations and vibration sensitivity causing phase noise in the oscillator and random high-frequency phase noise [2].

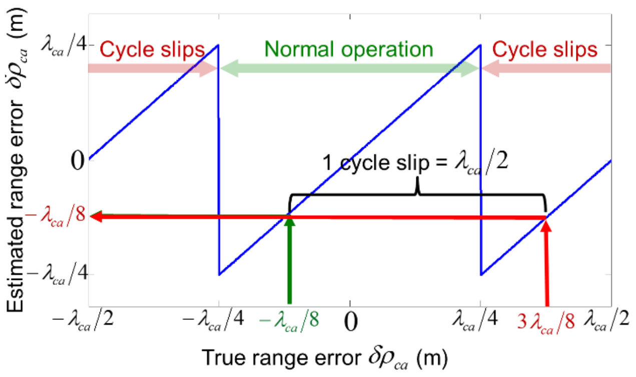

Figure 2.11. Example of a cycle slip in a Costas Arctan (two-quadrant) discriminator.

A phase lock loop (PLL), outputs measurements typically at a rate between 5Hz and 20Hz effective bandwidth, noting that this is the same as the equivalent noise bandwidth up to a level where the amplitude response of the PLL (considered for this purpose as a low-pass filter from a digital signal processing perspective) drops below a level of -3dB. For applications that do not require fast response to high dynamics, e.g. static receivers used for surveying, then a low PLL bandwidth may be used [61]. For high dynamics applications such as human motion modelling, a PLL with a high effective bandwidth is more appropriate. The effective bandwidth of a PLL determines the tolerance against antenna motion jerk dynamics, over the coherent time interval (typically 0.02s) [3]:

$$|\dddot{\rho}| < \frac{\left(1.2B_{L\_CA}\right)^{3} c}{4 f_{ca}} \quad , \tag{2.4}$$

where $B_{L\_CA}$ the effective PLL bandwidth (Hz), $c$ the speed of light (m/s$^2$) and $f_{ca}$ the carrier central frequency (Hz). From Eq. (2.4), a third-order PLL with 20Hz effective bandwidth can tolerate up to ±657m/s$^3$ jerk along the LOS, about ±82m/s$^3$ with a 10Hz effective bandwidth and about ±10m/s$^3$ with a 5Hz effective bandwidth, i.e. slow response to motion jerk dynamics, noting that larger (absolute) jerk dynamics may occur over smaller time intervals (less than 0.02s) without disrupting carrier tracking by causing cycle slips [3]. Typically, carrier phase tracking can be maintained when $C/N_0$ is above 24dB-Hz depending on the PLL bandwidth and motion dynamics [2]. A typical PLL bandwidth for a high-grade receiver is 15Hz (adjustable between 1Hz and 100Hz), and 0.25Hz for a DLL (adjustable between 0.01Hz and 5Hz) [74], noting that for the Xsens MTi-G GPS/IMU sensor used in this thesis (see Chapter 3) this information is not commercially available, which is typically the case for lower-grade receivers, such as those used in smartphones and wearables for pedestrian applications.

A frequency lock loop (FLL) tracks the carrier frequency error between the true Doppler shift of the carrier and the observed Doppler shift obtained by the carrier tracking loop. The FLL refines the Doppler shift prediction (equivalent to carrier range rate error in the range domain) of the DLL, maintaining a more robust correlation between incoming and replica PRN code in the signal processor, as well as aiding the PLL when phase lock is lost due to a low $C/N_0$. An FLL tracks Doppler shift (equivalent to range rate error in the range domain) in a similar way that PLL tracks the carrier phase (equivalent to range) error, as illustrated in Figure 2.12.

The operation of an FLL begins with the carrier frequency error discriminator, which detects the carrier range rate (frequency) error by computing the Is and Qs of the current and previous correlation epochs. The carrier Doppler shift error (equivalent to pseudo-range rate error in the range domain) is detected by the discriminator and used by the FLL to update the current pseudo-range rate, and pseudo-range acceleration estimates using a gain function to smooth the noise in the Is and Qs. It is worth mentioning that if the loop gain was unity, the noise contained in Is and Qs would not be smoothed and would be propagated to the frequency tracking error output of the discriminator function, which would be used directly to control the NCO in order to maintain alignment between the received and internally generated signals. The updated pseudo-range rate estimate is then used to predict the next epoch pseudo-range rate and feed the NCO with a refined Doppler shift estimate to maintain maximum correlation over the coherent time interval.

False locks may occur in FLLs in cases where the Doppler shift estimate is not consistent with the DLL Doppler shift estimate. Usually a false frequency lock manifests itself as a multiple of a minimum frequency, e.g. 25Hz for GPS C/A code, which is equivalent to approx. -4.75m/s in the range domain, calculated using [3]:

$$\Delta f_{ca} \approx -\frac{f_{ca}}{c}\dot{\rho}_R \Leftrightarrow \dot{\rho}_R \approx -\frac{c}{f_{ca}}\Delta f_{ca} \tag{2.5}$$

where $\Delta f_{ca}$ the Doppler shift (Hz), $f_{ca}$ the carrier frequency (Hz), c the signal propagation speed and $\dot{\rho}_R$ the range rate (m/s). This false lock of the Doppler shift could be prevented by comparing the FLL Doppler estimate with the DLL one [40], although usually the carrier tracking loops aid the DLL by providing the Doppler estimate. A false lock may also cause an incorrect frequency estimate in PLLs.

If carrier frequency tracking is used by the receiver, the data-bit transitions are detected by observing the changes in (a 4-quadrant) arctan(Q/I) as multiples of π rad. This provides noisier data demodulation than phase tracking. In both cases, there is a sign ambiguity in the demodulated data-bit stream. In frequency tracking, this occurs because the sign of the initial bit is unknown, whereas in phase tracking, it occurs because it is unknown whether the tracking loop is locked in-phase or π out-of-phase. The ambiguity is resolved using the parity check information broadcast in the message itself. This must be checked continuously, as phase tracking is vulnerable to cycle slips and frequency tracking to missed detection of the bit transitions [3].

Generally, carrier frequency tracking is more robust than phase tracking and can be maintained under lower $C/N_0$ compared to PLL requirements, but to prevent false lock detection from PLLs and FLLs, it is essential to introduce a metric of their robustness, e.g. by comparing the integral of the Doppler shift with the change in code phase [3].

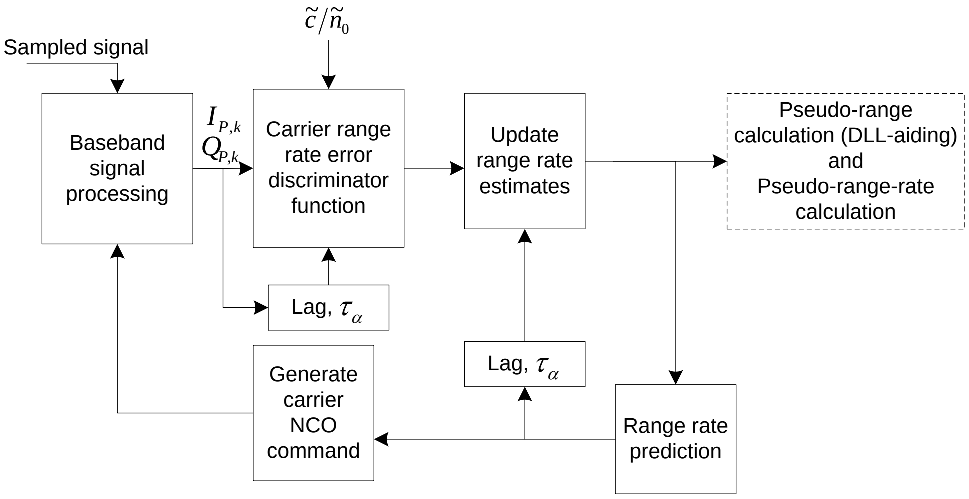
$\tilde{c}/\tilde{n}_0$
Sampled signal
Baseband signal processing
$I_{P,k}$
$Q_{P,k}$
Carrier range rate error discriminator function
Update range rate estimates
Pseudo-range calculation (DLL-aiding) and Pseudo-range-rate calculation
Lag, $\tau_\alpha$
Lag, $\tau_\alpha$
Generate carrier NCO command
Range rate prediction

Figure 2.12. Architecture of a generic FLL using range rate estimates (after [3]).

The signal strength is usually measured in terms of the signal-to-noise ratio (SNR), which represents the (unitless) ratio of the signal power to the noise power over a given bandwidth. Another common way of expressing signal strength is using the carrier power-to-noise density ratio ($C/N_0$), measured in dB-Hz, and represents the ratio of the carrier power to the noise power per unit bandwidth. The SNR, as opposed to $C/N_0$, depends on the bandwidth of observation. For a GNSS receiver, this is usually the effective noise bandwidth, once the signal has passed the last filtering stage in the (analog) RF front-end, which is typically about 4MHz [62].

### 2.2.4 **Integrated Navigation Systems**

An INS provide a self-sufficient navigation solution, typically at an output rate higher than 50Hz, but this solution drifts (accumulates errors) due to the constituent inertial sensors' properties and initialisation errors. On the other hand, GNSS can provide a navigation solution which does not drift, but the output rate is limited compared to INS, typically less than 10Hz. Therefore, by integrating INS and GNSS it is possible to derive a navigation solution with limited drift and at a high output rate. For example, the Xsens MTi-G IMU/GPS, has a GPS-only output rate at 4Hz, while the IMU can output data up to 512Hz [49]. INS also smooths out GNSS errors due to tracking noise, dynamics response lag, multipath, NLOS etc.

The integration is typically realised through a Kalman Filter (KF) algorithm, which efficiently fuses measurements from different sensors. A KF is a well-documented Bayesian state estimation algorithm, e.g. [3][61][75], which is used to fuse measurements from different sources in order to estimate a real-world system modelled as a set of parameters called "state vector" or "state", with a defined mathematical relationship between them. The basic principle underlying a KF is that it is possible to predict the state of a real-world system by knowing the initial (or previous) state vector and its corresponding uncertainty, along with its deterministic and statistical (stochastic) properties. The predicted state is then combined with the current state (coming from the actual sensor measurements, as well as the time evolution of other states) as a weighted average, depending on their uncertainties (standard deviations). The underlying assumptions of a (linear) KF are that the next state can be predicted by combining linearly the previous state with the system model (time evolution of state), while the current state can be calculated from the predicted state, the measurements and the measurement model, as well as that noise properties can be modelled by Gaussian distributions. Different versions of the KF that are not limited by all of these assumptions exist, e.g. the Extended KF [3], but they are out of the scope of this thesis. An example of a KF used to integrate GNSS and inertial measurements is given in [70]. KF may also be implemented as a forward filter, i.e. in order to reach the final state conditions, of backwards, in order to reach a set of initial state conditions from the final state (backward filter). A combination of a forward and backward filter, called a Kalman Smoother, is generally a robust way to smooth motion capture data [76]. Figure 2.13 illustrates a basic integration scheme of INS/GNSS measurements, using a Kalman Filter (KF) integration algorithm.

An INS/GNSS integration architecture may be uncoupled, i.e. the GNSS measurements simply update (or re-initialise) the inertial navigation solution at specific time intervals. In a loosely-coupled integration, GNSS provides position and velocity measurements to update the inertial navigation solution and compensate some inertial sensors biases and orientation errors. The Xsens MTi-G IMU/GPS employs a loosely-coupled KF [49], in order to correct the inertial position, velocity and orientation solutions using GPS updates. The inertial and GNSS navigation solutions (position and velocity) are fed into the integration algorithm which calculates the error states, e.g. accelerometers, gyros and orientation (attitude and heading) biases, and not the original quantities themselves (e.g. specific force or angular rate). These errors are then used in a correction stage to update the inertial navigation solution and provide the integrated navigation solution (position, velocity and orientation), noting that the actual KF implementation for the Xsens MTi-G is proprietary so the end user cannot amend it, neither feed the GPS-only and intertial solutions separately to their own integration algorithm.

The KF implementation in the Xsens MTi-G can calibrate the accelerometers' biases, provided that a GPS fix is available and the IMU is subject to significant horizontal acceleration, in order to allow the accelerometers to sense specific force and create linear velocity and quadratic position errors which can be observed by the KF. Note that in the vertical direction the accelerometers sense significant specific force anyway due to the internal proof mass's reaction to gravity (due to the chassis's walls that do not let the proof mass exhibit free fall), therefore the KF will compensate for these biases. The KF can then correct the accelerometers' biases using GPS updates (as external absolute position information is required), and observe the heading error of the IMU, in other words, it integrates the accelerometers' measurements in the correct direction to derive the position and velocity solutions. Once the heading error has been observed, the KF stabilises and the attitude of the IMU can be calculated, considering the gyros' measurements and an initial attitude solution, obtained by the 3D magnetometers. It must be noted that roll and pitch attitude components do not require heading error to be observed and that the IMU heading drift can be corrected by the magnetometers' readings, if they are accurate enough. The KF heading error observability disappears when the IMU stops experiencing horizontal acceleration (and no magnetometer heading measurements are available). For pedestrian applications, this implies regular horizontal motion while conducting an experiment. A calibration procedure, as part of the experimental protocol before starting pedestrian motion capture, ensures that the KF has stabilised and the final state can be stored for future use [49], in order to speed the calibration process for the next experiments. A potential issue that requires attention while using a KF, is discontinuities when the GNSS measurements are fused with the inertial solution, e.g. sudden peaks when the GPS measurements update the inertial position solution. Further details on this observed issue during the thesis' experiments are given in Section 3.2, with an implemented mitigation technique detailed in Appendix B.

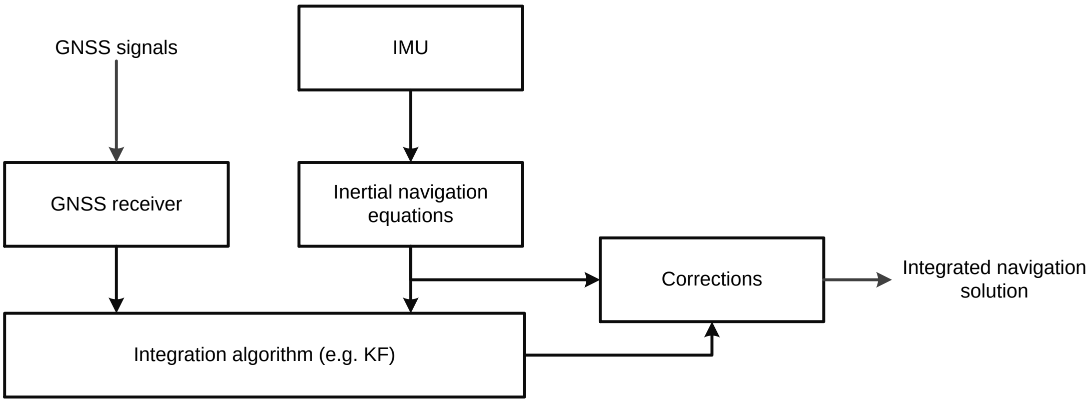
GNSS signals
IMU
GNSS receiver
Inertial navigation equations
Corrections
Integrated navigation solution
Integration algorithm (e.g. KF)

Figure 2.13. GNSS corrections for INS (after [3])

Different levels of integrated INS/GNSS architectures, which are out of the scope of this thesis, encompass *tightly coupled* integration, which fuses GNSS pseudo-range and pseudo-range rate measurements with the inertial navigation solution, and *deeply coupled* integration, which integrates Is and Qs from a GNSS signal processor with inertial sensors' measurements. Inertial sensors' measurements may also be used to aid GNSS tracking loops in tightly or deeply coupled integration, also called *ultra-tight* integration [3].

## 2.3 Pedestrian Motion Capture and Modelling

This section discusses the methods used to capture and model human motion, in order to inform the development of an experimental protocol for capturing human motion, as well as the development of a pedestrian motion model (detailed in Chapter 5), which simulates the performance behaviour of GNSS carrier-tracking loops under the presence of real pedestrian motion.

### 2.3.1 Human motion capture and modelling

Human motion can be captured using a range of different sensors or their combinations. The capture, analysis and modelling of human motion supports the development of a pedestrian motion model (PMM) for the purposes of this thesis. Clinical applications of human gait analysis may employ optoelectronic systems using active or passive markers, which are considered standard techniques [18], inertial measurement units (IMUs), which typically encompass accelerometers and gyros, discussed in Section 2.2.2. In [46], the authors assess how suitable inertial sensors are for the study and analysis of human motion, concluding that they have the potential to be employed in clinical applications involving walking. For example, in [77] a shank-mounted IMU was placed on 8 subjects to estimate their walking speed (with a root mean square error - RMSE 7%) and their travelled distance on the ground (with a RMSE 4%), although the algorithm failed to give accurate results on inclined surfaces due to a wrong estimated initial angle between the shank and the IMU. In [9], a trunk-placed tri-axial accelerometer was used to study the gait spatio-temporal parameters for a group of elderly people. A review of how accelerometry has been used to study human gait patterns is given in [78]. Also, an experiment with a head-mounted GPS receiver, aided by real-time kinematic corrections to achieve better accuracy, was used in [79] to monitor gait spatio-temporal parameters (walking speed, step length and step frequency) of human subjects walking on a level surface for about 30 minutes; however, the results showed that the inter-subject variation for the observed parameters was not consistent to other similar studies in the literature, therefore, further this method is recommended by the authors of this study alternative to standard technologies, e.g. vision-based systems, used in biomechanical studies. Other MoCap techniques may employ magnetic transmitters (human body markers) and a sensor(s) to measure joints' position and attitude [80], noting that due to the free space loss in a magnetic field's strength, this method would not be suitable for this thesis, i.e. in order to capture a walking pedestrian away from the sensor.

A prerequisite of using IMU sensors for human motion capture (MoCap) is the static bias calibration of their constituent inertial sensors. The Xsens MTi-G IMU/GPS used for the human MoCap experiments of this thesis is described in Section 3.3. The GNSS antenna design, e.g. gain pattern, may also affect the precision of MoCap as, typically, the antenna attitude will changes following the human body movement, where it is attached to. Another important element when using an IMU/GNSS device for human MoCap, is the GNSS antenna placement on the human body. Ideally, the selected device locations should allow for good GNSS satellite visibility during a MoCap experiment. The sensor location must therefore be carefully selected depending on the application requirements and considering that the human body may block GNSS signals or induce relative motion dynamics between the GNSS antenna and the IMU, which may render the integrated navigation solution unreliable.

Pedestrian motion capture involves the tracking of the 3D coordinates of human body segments and joints which may involve a combination of inertial sensors and optoelectronic systems. Optoelectronic equipment can be used for MoCap in order to track the 3D motion of passive markers placed on the human body [18], which have a retroreflective coating to reflect the light from sources near the MoCap system sensors, typically high-speed cameras. MoCap systems may also use active markers, i.e. markers which emit light (in white or various colours). Examples of commercial MoCap solutions are: Qualisys [81] using active or passive markers, CODA [82] using active markers and Vicon [83] using passive markers, which can be used for (high-end) professional MoCap applications, e.g. in the film or sports industry. Also, commercial markerless motion capture systems exist, for the lower-end of the market, e.g. [84],[85],[86] and [87]. MoCap systems offer various performance specifications in terms of field-of-view, resolution and accuracy, so extended research is recommended before committing into purchasing/hiring a MoCap system solution.

With respect to the used technologies in this thesis, it must be noted that the original scope of this thesis also included investigating how pedestrian motion affects inertial sensors, so the project budget for equipment purchase was committed to the Xsens MTi-G IMU/GPS solution in order to provide data both from inertial and GPS sensors. However, as the project progressed the scope had to be focused on investigating the pedestrian motion effects on GNSS carrier-tracking loops, so the option to hire or purchase a high-end vision-based system for very accurate human MoCap was not an available option.

Methods employing active or passive markers need to be calibrated or different experimental subjects, as the locations of markers may differ between subjects, while the skin upon which markers are placed introduces additional relative motion artefacts w.r.t. the body segments/joints in question. For both marker-based and markerless systems, calibration of the models w.r.t. the actual anatomical points (e.g. joint centres) in the human body is challenging as these are not physically accessible. A typical implementation of a vision-based system may involve at least body 15 markers, with at least two cameras for body-tracking applications and at least three cameras if body modelling (reconstructions) is sought. A common problem using vision-based systems is markers' occlusion from human body segments the human body moves [88], noting that redundancy cameras, may be used in order to mitigate that effect.

For the modelling of MoCap data using optoelectronic means tracking active or passive markers, the following three steps may be applied [18]:

- The markers are attached to the subject and their motion is tracked, e.g. by video cameras in 3D. The 3D kinematics of the markers are then calculated using techniques of stereophotogrammetry.
- The human body model (segments and joints) are best fitted to the markers trajectories for each frame of their captured motion, using an appropriate algorithm, e.g. one that minimises the RMSE between the human body model segments and the tracked markers position. The human body model may be a skeleton comprising linear segments and joints or a set of cylinders and spheres which also represent the volume of human body segments and joints. Human body constraints may also be used to avoid unrealistic postures when fitting the human body model to the tracked markers. The human body model parameters are obtained by anthropometric studies [88],[89].
- The kinematic quantities of human segments and joints are then calculated, e.g. the position, velocity and acceleration of a segment's or joint's CoM.

In the case that specific motion dynamics need to be recreated by the model, a fourth step needs to be applied using inverse kinematics' methods that "force" a particular segment or joint to follow a specific motion dynamics profile. Due to the many combinations of joint rotations that could result to the same motion profile, this problem is typically overdetermined as more than one solutions exist. Therefore, motion constraints may be applied to reduce the possible solutions to one that is the most "human-like". The results can then be validated by comparing the model output with the captured motion dynamics.

Inertial sensors typically provide a high output rate e.g. above 100Hz, which is required for tracking pedestrian motion, especially when it involves high dynamics, e.g. running. Also, inertial sensors can also be integrated with non-optical systems, e.g. magnetometers [89], barometers, and GNSS [90], ultrasonic [91] or ultra-wide band (UWB) [92] devices. Inertial sensors are typically fixed on specific anatomic locations of the human body [93], e.g. on the head, shoulder upper arm, forearm, hand, above the chest-bone or the tail-bone and on the foot. The inertial sensors cannot be fixed at the exact anatomical point in question, e.g. for COM movement analysis while walking, the CoM lies inside the pelvis. Therefore, a correction of has to be applied, e.g. by smoothing measurements or applying lever arm corrections, in order to compensate for the relative position difference. Other experimental effects, e.g. such as clothes or skin movement, where the sensor is attached to, should also be taken into account when processing/interpreting the results. If the IMU is placed above the tail-bone it can be generally approximated as fixed w.r.t. the CoM for most pedestrian applications, e.g. in [94]. A review of non-optical commercial systems in use for pedestrian motion tracking and gait analysis is presented in [95]. Multiple inertial sensors may also be used as part of a full-body wearable system [96]. This thesis falls within this category as it uses an Xsens MTi-G IMU with GPS, magnetometers and barometers aiding. The candidate sensor locations for modelling in this thesis, i.e. head, arm, hand, foot and inside a bag, are discussed in Section 3.2.

MoCap data interpretation needs to take into account the experimental conditions. The differences between capturing data on a treadmill and on the ground are examined in [97], where the authors found that the subjects were habituated to the use of a treadmill after 10 minutes. In another study, the differences between treadmill and walking on ground were not found to be statistically insignificant although data captured at experiments conducted on the ground implied a slightly wider range of motion compared to those carried out on a treadmill [98]. Therefore, this study suggests that a pedestrian MoCap experimental protocol that involves a treadmill should take into consideration the habituation of the subjects with the equipment.

The human body movement can be modelled using appropriate biomechanical models; examples can be found in [7],[99],[100], [101]. These models contain a simplified version of human body segments, e.g. modelled as cylinders or spheres, and motion constraints between these segments, which represent the mechanical properties and DOF of human joints. Using a human biomechanical model (HBM), it is possible to simulate particular type of activity, e.g. walking, jogging, and/or gestures. Human gesture examples may include moving arms back and forth, answering a phone, turning the head, kicking with a foot etc.

The processing of captured pedestrian motion data can be a quite involved process. A comparison study of 6 smoothing methods for biomechanical data provides evidence that there is no optimum standard technique and that if a method is efficient in smoothing biomechanical data (e.g. in the position domain), this is not necessarily the case for higher derivatives (e.g. velocity and acceleration domains) [102], which are important for testing GNSS equipment, as they can cause disruptions to the nominal carrier-tracking operation, as discussed in Sections 4.2 and 4.3.

In a human animated motion generation study [103], a collection of motion clips was captured (sequences of frames represented as graphs) and then combined using a weighting and smoothing function to produce smooth, human-like synthetic motion, based on some hard or soft spatial and posture constraints, i.e. specific points/postures that the robot should reach or come close to respectively. This method is suggested for animation applications and not for re-creating pedestrian motion using inverse kinematics, due to the high number of constraints that would be required to control the motion and posture of each human segment over the time of the movement duration.

Other methods of human motion modelling may be physics-based and aim to optimise the movement of a HBM in terms of minimising the energy required to move the HBM during the whole movement. Although using this method the HBM segments' movement are constrained using a set of rules [104], various human gestures and movements can be generated automatically without requiring prior motion capture and calibration. However, this method does not generate realistic human motion due to introduced artefacts, e.g. exaggerated hip movement while walking. Therefore, it has not been considered further for analysis in this thesis. Examples of applications using human body modelling encompass ergonomic product design (taking into account human factors) and workers' health risk assessment in industrial settings, e.g. the automobile, aerospace and manufacturing industries; and also, clinical applications, which may focus on specific parts/activities of the human body, e.g. muscular activity modelling for athletes' fatigue/injury risk assessment [105], [106].

### 2.3.2 **Pedestrian routing**

Human locomotion is goal-oriented, therefore encompasses inherently a navigational element along with the biomechanical activity itself. Pedestrian navigation encompasses two basic actions, wayfinding and positioning [3]. Positioning encompasses the process of finding the current pedestrian position and a target position, referenced to and resolved along a known reference frame axes, i.e. an Earth-fixed frame for pedestrian applications, such as a local tangent plane. Pedestrian wayfinding (or routing) implies a target position which the pedestrian aims to reach by moving its body in 3D space, i.e. locomotion. In this thesis, a pedestrian routing model is meant to be an algorithm used to plan a route (or path) between two pre-defined geographic points in 3D space.

Generally, pedestrian models can be viewed from two perspectives [107]: *macroscopic* pedestrian models, which are used to represent groups of pedestrians (or pedestrian flows) using principles of fluid dynamics, and *microscopic* pedestrian models, which focus on the level of individual pedestrians and are more relevant to the scope of this thesis. A pedestrian model also, may be *deterministic*, *stochastic* or a combination of both, e.g. [108],[109]. Deterministic pedestrian models provide the same results, e.g. path-finding trajectories, on every simulation run at given initial conditions, while stochastic models provide a range of possible outcomes, e.g. by adding random noise when the pedestrian model state is updated.

In the case of a continuous space-time domain, a pedestrian model agent may be represented as a particle whose 3D movement in space is driven by attractive/repulsive factors modelled as Newtonian forces, also called 'social forces' in [110]. Social forces can be attractive, e.g. towards a target or points of interest, or repulsive, e.g. against other pedestrians or obstacles like walls etc., noting that this could also be represented using a magnetic analogy and the respective EM field equations. An example application of the social force model is to determine irregular individual pedestrian behaviour in crowds [111]. Although social force modelling may provide valuable statistical results when a large number of pedestrian particles are employed, this may not be the case in simulating motion at individual pedestrian level or when a small number of pedestrian agents is required for a given application [109]. Last but not least, social force models implementation can be computationally demanding as they require multiple iterations to calculate interactions between modelled pedestrian particles.

In the discrete space-time domain, there are mainly two approaches of modelling pedestrian behaviour: *cellular automata* and *agent-based* models, both using a matrix of cells to represent space, as well as discrete time steps to advance between current and future pedestrian states [112]. Cellular automata models are considered efficient in macroscopic modelling of pedestrian flows [113], while agent-based models allow agents to follow individual behavioural rules, which is more relevant to the scope of this thesis. Pedestrian agents may also be given a specific target to navigate to, or may move around using a stochastic process, e.g. performing a random walk, or may select a target based on a pre-defined probability distribution [114],[115].

Pedestrian routing models have been employed in robotics applications, both for trajectory planning and in order to make robotic trajectories more 'human-like' (or 'anthropomorphic') [116]. Other application where pedestrian routing models have been employed encompass prediction of pedestrian flows for transport and retail facilities, evacuation planning and path optimisation for emergency situations, pedestrian tracking [117] and subgroup characteristics simulation, e.g. pedestrians holding hands [118].

For the purposes of driving a pedestrian agent between two known geographic points, which is more relevant to the scope of this thesis, path-finding algorithms can also be employed. A path-finding algorithm aims at optimising the path between two locations (or any two network nodes), e.g. in terms of time, distance, cost or any other appropriate criteria [119]. Generally, path-finding algorithms can be applied using deterministic or stochastic rules, e.g. in order find a route within a transport network. Typically, the domain of a path-finding algorithm is represented by a graph or, equivalently, by a grid (lattice) of cells.

A commonly used method of finding the shortest path between two nodes, either in a graph or a grid of cells, is Dijkstra's algorithm [120]. Dijkstra's algorithm starts on a specific cell and searches the adjacent cells to find which one provides the lowest cost. a specific node and expanding gradually the search around that node, until the target node is reached, provided that at least one connecting path between the start and end nodes exists. A special case of Dijkstra's algorithm is the A* (pronounced "A-Star") algorithm, which aims at making the search process for the shortest path more efficient computationally by using heuristic rules, e.g. constraining the search on cells within a buffer along a straight path between the start and finish points [121]. Examples of A* implementations are reported in [122][123][124] and [125]. It is worth noting that the A* algorithm search is optimal (compared to the original Dijkstra's algorithm), only if the applicable heuristic rules are optimal.

A drawback on using path-finding algorithms on grid space is that the final path is not smooth, as it depends on the density of the underlying grid [126]. One possible solution is to increase the density of the grid, increasing the computational load, as the number of cells in the search space will increase quadratically. Another possible solution is to apply a smoothing function to the final path produced by a path-finding algorithm. In pedestrian motion simulation applications, which are relevant to this thesis, this smoothing is required in order to avoid artefacts, e.g. sudden position "jumps" of an animated character, or potential artificial effects on simulated GNSS carrier-tracking loops (SGCTLs), e.g. artificial cycle-slips on carrier phase tracking. Smooth trajectories generation is an issue that concerns the animation and gaming industry [127] and robotics [128]. Also, a review of spline (curve-fitting) methods that can be used as smoothing functions is provided in [129].

To summarise, this chapter reviewed the relevant literature to provide the background knowledge required to address the aims of this thesis, stated in Section 1.2. The chapter introduced to the reader some fundamental concepts of pedestrian motion analysis, e.g. the gait cycle, and methods used to capture human motion, e.g. using optoelectronic systems and/or inertial or other positioning and navigation sensors. In addition, this chapter introduced basic concept and principles in inertial and GNSS positioning and navigation. Finally, this chapter discussed different ways to model human body motion, e.g. with systems employing active/passive markers or markerless systems, and also introduced Dijsktra's path-finding algorithm between two locations. This discussion in this chapter informs the design and execution of human motion capture experiments (see Chapter 3), the development of the pedestrian motion model (PMM), described in Chapter 5, and the simulation of GNSS carrier tracking loops (see Chapter 4) as a test platform for analysing and comparing the real human MoCap and PMM synthetic motion data. The thesis continues with the description of the human motion capture experiments.

# Chapter 3. Field Data Collection and Pre-processing

This chapter describes the experiments of human motion capture (MoCap) of this thesis. The human MoCap experiments provide the ground truth data for assessing the impact of human motion on the performance of GNSS equipment (see Chapter 6) and for validating the synthetic output of the pedestrian motion model (PMM), detailed in Chapter 5. This chapter comprises three sections, the first one describes the experimental equipment used to capture human motion data, the second one (Section 3.2) describes the pilot experiments and the third (Section 3.3) describes the main experiment of human MoCap. Alongside the discussion of the pilot and main experiments, this chapter discusses the quality issues identified during pre-processing the data in order to facilitate further analysis (discussed in Chapter 6), by using the simulated GNSS carrier-tracking loops (SGCTLs), which are detailed in Chapter 4.

The motion capture process described in Chapter 3 was essential in order to analyse the effects of real human motion on the performance of SGCTLs (see results in Section 6.2); to validate the synthetic motion output of the PMM (see results in Section 6.3); and also to make recommendations on GNSS carrier tacking loops design (see results in Section 6.4). It must be noted that human MoCap data was not used for human biomechanical modelling, as this was based on other studies in the human biomechanical literature (see Section 5.3).

## 3.1 Experimental Equipment Description

Xsens MTi-G sensor, illustrated in Figure 3.1, provides an integrated KF navigation solution for position, velocity and orientation, w.r.t the local navigation frame (see Section 2.2.1). It comprises a triad of accelerometers, a triad of gyroscopes (gyros), a triad of magnetometers, a thermometer for temperature-dependent calibration, a barometer and a GPS receiver operating in the L1 frequency band and tracking coarse/acquisition (C/A) ranging code. The device provides the options to output calibrated or uncalibrated (raw) accelerometers' measurements. In the case of calibrated accelerometers' measurements, the device performs (at firmware-level) temperature-dependent calibration of the inertial sensors' measurements and correction of the accelerometers' static and run-in biases. The accelerometer's bias calibration is achieved by updating the inertial navigation solution (position, velocity and attitude) using GPS updates at a 4Hz rate, through a Kalman Filter (KF) which has to be selected (from a list of 6 proprietary pre-defined KF scenarios) before the motion capture starts and cannot be changed afterwards.

The KF scenarios provide the option to fuse accelerometers, gyros and GPS measurements (as a minimum), with additional options to include measurements from the triad of magnetometers and the barometer. The two KF scenarios considered in this thesis are the "General" which uses the

IMU (accelerometers and gyros), GPS and barometric measurements, and the "Aerospace", which uses IMU (accelerometers, gyros and magnetometers), GPS and barometric measurements. It follows that the IMU outputs of the general and aerospace scenarios are identical, but the integrated navigation solution (position in latitude, longitude and altitude) and velocity, differ. Other KF scenario options include "General/Aerospace_nobaro" which are similar to the general and aerospace scenarios, but do not fuse the barometric measurements. The latter KF scenarios were not used as the motion capture experiments were conducted outside, so the barometers should be able to measure atmospheric pressure without constraints occurring, e.g. inside a sealed lab. Also, two additional KF options intended for automotive applications (applying constraints on the heading) and marine applications (assuming constant significant velocity), are considered out of the scope of this thesis. Xsens MTi-G also provides the option to store the current Kalman Filter state (inertial sensor biases) and re-use it in the future for similar MoCap scenarios. In this way, the Kalman Filter settles faster next time that new MoCap data is to be captured [49].

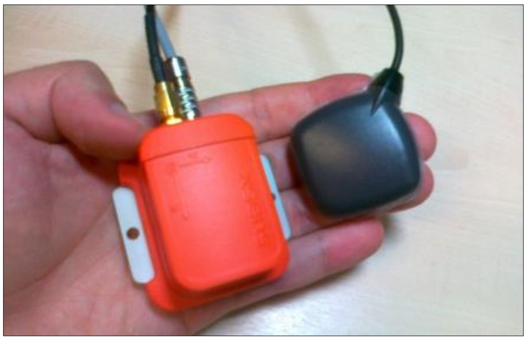

Figure 3.1. The Xsens MTi-G IMU/GPS device with a GPS antenna

As far as heading observability is concerned (see Section 2.2.4 for an explanation), if the Xsens MTi-G is static or moving at constant speed >10s, then the heading observability will degrade and the KF status will be set as "invalid". The aerospace KF scenario however, uses magnetometers so the heading is always observable. For KF scenarios fusing magnetometer measurements, it is recommended to perform a magnetic calibration, in order to calibrate any hard/soft iron effects at fixed position/attitude w.r.t the sensors frame, before commencing the data collection. The process of magnetic calibration of the Xsens MTi-G is explained in [130]. Xsens MTi-G updates the local magnetic declination when there is GPS fix, based on the WMM-2010. The X-axis of the body frame points to the magnetic north when no GPS fix/set magnetic declination are available, otherwise X-axis will be aligned with the true North. The magnetic calibration process assumes that there is no horizontal acceleration, in order to calculate accurately the tilt during rotating the sensor. Also, the magnetometers output is normalised to the local geomagnetic field strength (i.e. after calibration), as the body frame attitude can be determined by the relative change of the measurements between the magnetometers' axes, without having the absolute measurements.

The raw data provide the inertial sensors' measurements at the analog-to-digital conversion level, i.e. uncalibrated for temperature (or accelerometers' bias via GPS updates). The raw data output may also include the GPS-only position, velocity and timing (PVT) solution, if the user selects so. The option to capture raw data instead of calibrated, enables the user to export calibrated data at software-level from the logged raw data binary file (i.e. without having the IMU/GPS device connected to the user's computer) using any of these 6 proprietary KF scenarios. In terThe software package that provides this functionality is the Xsens proprietary "MT Manager", which is a graphical user interface for visualising the logged data during motion capture and export the logged binary files into text format, using an appropriate KF scenario, noting that Xsens encrypts and records the necessary calibration parameters logged into the binary file, in order to allow the user to export calibrated data during post-processing. The technical specifications for the Xsens MTi-G device are mentioned in Appendix C, based on [49].

The chapter continues with the description of the pilot MoCap experiments.

## 3.2 Pilot Experiments

This section describes the pilot experiments of this thesis. The pilot experiments supported the design and implementation of the main experiment, by enabling:

- Familiarisation with the experimental equipment, described in Section 3.1.
- Design and implementation of an appropriate calibration protocol for Xsens MTi-G INS/GPS,
- Design and implementation of an appropriate method to conduct the main experiment, e.g. keeping logs, video recordings, timings etc.,
- Development of appropriate tools for pre-processing human MoCap, e.g. Matlab tools to identify and/or filter out potential artefacts due to the data collection process or equipment design.

Table 3.1 lists the candidate sensor locations, applicable gestures and activities considered for modelling on the top of nominal walking movement in this thesis. The sensor candidate locations were selected as they comprise the most typical places to hold a sensor, depending on pedestrian activity, e.g. foot for measuring fitness activity or informing a pedestrian navigation system with zero-velocity updates (ZVU), noting that, as explained in Section 1.1, the purpose of this thesis' developed pedestrian model (see Chapter 5) is to drive the SGCTLs, not to form a complete pedestrian navigation system, which could be used e.g. as a testbed for pedestrian dead-reckoning algorithms. The foot as a sensor candidate location was explored during pilot experiments but was found to cause the inertial sensors exceeding their dynamic output range (a similar example from MoCap experiments is illustrated in Figure 3.3). The hand is a common place of placing/holding PNE such as IMU/GPS-enabled wrist watches or smartphones, respectively, in pedestrian applications. The head may also be a candidate location to model as e.g. a helmet could be equipped with a GNSS receiver; however, for practical reasons and time constraints this location was not prioritised in the MoCap experiments. Also, an armband or a bag, e.g. back-pack or hand-bag, are typical places where PNE is attached to or held, respectively. Finally, candidate locations in the waist (using e.g. a waist-band) and the back (above the tail-bone) were not included in the main MoCap experiment, as during the pilot experiments they were found to impose practical issues on the MoCap process, i.e. it was not practical to fix the sensor with the antenna in those locations, ensuring that the relative motion between the sensor and the human body is minimised. The main MoCap experiment is described in Section 3.3.

The most relevant gestures to the scope of this thesis are those involving hand movement, as hand movement can subject GPS-enabled devices to significant jerk dynamics along the line of sight (LOS) between the receiver and GNSS satellites' antennae. This could potentially lead to performance degradation by introducing phase cycle slips and false frequency locks in the receiver's carrier tracking loops, as explained in Section 2.2.3.4. A practical problem observed while captured human motion was that the gyroscopes cannot in Xsens MTi-G could not capture the whole motion range of the "answer the phone" gesture, as illustrated in Figure 3.3. Therefore, adjustments had to be made during the main experiment in order to manually avoid exceeding the (limited) dynamic output range of the gyros, otherwise the inertial navigation solution would not be updated accurately, as described in Section 2.2.2.

It is also worth noting that in this thesis, MoCap refers to the captured trajectory of the Xsens IMU/GPS sensor attached on or held by a pedestrian at typical body locations, such as the arm, during the thesis' pilot and main field experiments. The MoCap in this thesis employed one sensor only and was used to validate the final trajectory of that sensor, not to capture the motion of individual human body segments for biomechanical modelling purposes, or isolate different movements, e.g. walking from gestures. The human biomechanical model used in this thesis was based on biomechanical studies (see Section 5.3 for further details).

In terms of relevant pedestrian activities, such as walking, jogging and running (all encompassed by the human gait definition in Section 2.1.3), the pilot experiments showed that jogging and especially running could drive the accelerometers and gyros outside their specified output dynamic range. Therefore, the main experiment MoCap protocol was adjusted in order to accommodate this limitation, by excluding running MoCap scenarios and jog at a lower (than originally chosen) speed in order to induce less motion dynamics stress to these inertial sensors. It is also worth noting that in all pilot and the main human MoCap experiments, the right leg is the reference leg of the gait cycle (stride), i.e. a gait cycle is defined between two consecutive heel strikes of the right leg, and that all MoCap scenarios started by moving the right leg first.

| *Candidate sensor location* | *Examples of applicable gestures* | *Examples of applicable activities* | *Experimental Motion Capture* |
|---|---|---|---|
| (1) Foot | (A) Kick a ball<br>(B) Stamp foot on the ground | (i) Walking<br>(ii) Jogging<br>(iii) Running | **No** (as gyros and accelerometers exceeded their specified dynamic range) |
| (2) Hand | (A) Answer a phone<br>(B) Put a phone down (after answering)<br>(C) Send a text or email/looking at the phone screen while holding it<br>(D) Put/Remove the phone in/from the pocket | | **Yes** (for walking and adjusted jogging activities only, in order to avoid exceeding the inertial sensors' specified dynamic range) |
| (3) Head | (A) Turn head (up/down/right/left) | | **No** |
| (4) Bag (e.g. back-pack or handbag) | n/a | | **Yes** |
| (5) Waist (e.g. waist-band | n/a | | **No** (experimental capture practicalities) |
| (6) Back (above tail-bone) | n/a | | **No** (experimental capture practicalities) |

Table 3.1. Candidate HBM sensor locations and gestures.

The first pilot experiment was conducted in St. James's Park and Hyde Park in May 2012, using an Xsens MTi-G (IMU/GPS) and a digital video camera. The first stage of the experiment was the calibration of the Xsens IMU/GPS unit by holding it over the user's head and moved horizontally in a figure of 8 pattern, with the Z axis of the device (body frame) always vertical. The purpose of an IMU calibration was explained in section 2.2.4. The video-camera was used in order to have a visual record of the pedestrian movement with a time reference and identify the timings of specific movements during post-processing, e.g. in which timeframe the pedestrian user stopped the calibration and started performing the test scenario or a particular gesture (e.g. pick up the phone). The first walking scenario involved placing the IMU on the centre of the thorax (underneath the chest bone), with the GPS antenna on the head and walking in a 30m straight line, reversing and walking up to the starting point. The second walking scenario involved another calibration (same as in the first walking scenario) with the GPS antenna attached to the top of the right hand (for better satellite visibility) and the IMU on the other side of the hand. During walking, the user performed an "answering the phone" gesture three times. Figure 3.2 shows an example of MoCap in Hyde Park, using the Xsens MTi-G IMU/GPS device.

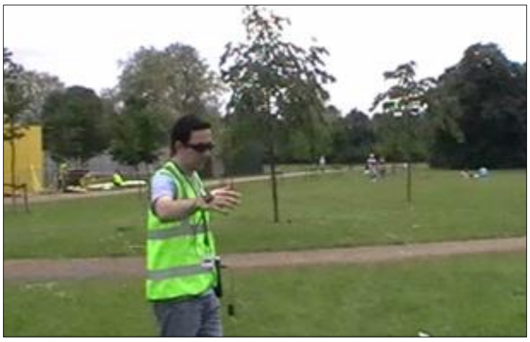

Figure 3.2. An example of using the Xsens MTi-G in a human MoCap experiment

The results of this experiment showed that there was a need to improve the calibration process, noting that the equipment user manual does not describe what movements are needed in order to calibrate the IMU [49]. Another observation from that experiment was that GPS position, velocity and orientation corrections to the integrated (KF) solution cause sharp peaks or transients due to inefficiency of the Xsens KF to smooth their effect. These transients are artefacts introduced by the Xsens KF, thus they need to be smoothed, because they could potentially cause artificial cycle slips when fed to the GNSS tracking loop model. As part of the data collection using the Xsens MTi-G IMU/GPS, an experimental protocol to calibrate the device was developed in order to calibrate the accelerometers' bias using the regular GPS updates.

Another observation from these experiments was that capturing pedestrian motion by attaching (or holding) the IMU and GPS antenna into a fixed relative position w.r.t. the human body, while trying to have a clear view of the sky for the GPS antenna for the duration of the movement, often impose conflicting requirements which cannot be addressed simultaneously. A potential solution would be to perform the calibration in a good signal environment and then move the equipment elsewhere to capture human motion; however, it was found experimentally the Xsens KF algorithm was degrading during the equipment setup stage, i.e. between start recording human MoCap and placing the laptop securely inside a back-pack. However, the location of the main MoCap experiment (see Section 3.3) was selected to be in a field with good satellite visibility, in order to enable the calibration to be performed on site.

The last pilot experiments took place in Regents Park (18/6/2013) and Kensington Gardens (25/6/2013). In both experiments, the Xsens MTi-G was used to capture pedestrian motion data at a 100Hz output rate. In response to the experimental MoCap issues identified in the previous pilot experiments, this last pilot experiment involved a new method of calibrating the IMU (using GPS updates), by walking in a cycle of 5m radius while moving the hand-held IMU/GPS device along the left-to-right direction. Also, in this experiment the data were collected in raw mode (instead of calibrated mode in the previous pilot experiments), to allow selecting different KF scenarios in the Xsens user interface (MT Manager) during post-processing, as well as potential future calibration of the IMU measurements by GPS ones through a custom KF (noting that this was not realised due to the thesis' time constraints).

The last pilot MoCap experiment pedestrian motion scenarios were:

- Walking 20m in a straight line, having the IMU in front of the body and going back.
- Walking 20m in a straight line, holding the IMU in the right hand, placing the IMU by the right ear (like answering the phone) on the 5th step, then removing the IMU beside the ear and keep holding it on the 10th step and going back.
- Walking 20m in a straight line, having the IMU in the back pocket, placing the IMU by the right ear (like answering the phone) on the 10th step, then removing the IMU beside the ear and keep holding it on the 20th step and going back.
- Walking 20m in a straight line, with the IMU placed above the tail bone and going back.
- Jogging 20m in a straight line, with the IMU placed above the tail bone and going back.
- Running 20m in a straight line, with the IMU placed above the tail bone and going back.
- Walking 20m in a straight line, with the IMU placed on the right foot and going back.
- Jogging 20m in a straight line, with the IMU placed on the right foot and going back.
- Running 20m in a straight line, with the IMU placed on the right foot and going back.

The selection of these motion capture scenarios aimed at capturing the differences between walking, jogging and running, as well as some typical phone locations on the human body and gestures while carrying the phone. The running scenarios showed clearly that the IMU was exceeding its nominal specifications, so these scenarios could not be repeated for this thesis, at least using the same experimental equipment.

An important issue observed while analysing the captured human motion data from the pilot experiments was that during holding or answering the phone the z-axis (of the body frame) gyro, exceeded the manufacturer's specified nominal dynamic range (300deg/s, [49]). This results in uncompensated non-linear scale factor errors (as scale factor errors can be compensated effectively within the specified sensor output range only), leading to an inaccurate navigation solution frame calculation, e.g. an erroneous local navigation frame. As a result, the acceleration is integrated in an erroneous navigation solution frame, causing the integrated position and velocity solutions firstly to drift, and secondly, not to change smoothly as the regular GPS updates will attempt to correct the inertial navigation solution errors, also depending on which KF scenario has been selected. An example of the z-axis gyro output is illustrated in Figure 3.3, noting that this figure describes the IMU movement in terms of gestures applicable to a phone, e.g. "answer the phone". A potential solution to this issue would be to rotate the hand such as minimising the angular rate about the z-axis of the body frame when performing the "answering the phone" gesture in the rest of the experiments.

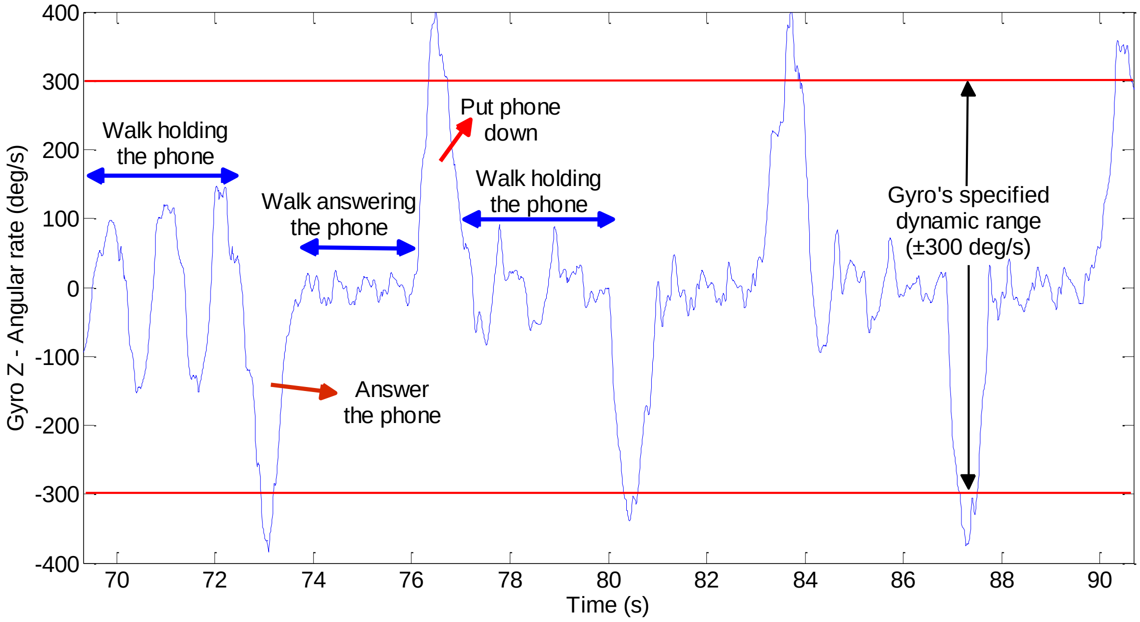

Figure 3.3. Gyro Z exceeding the specified (as per manufacturer's datasheet [49]) nominal dynamic range (red line) while using the IMU.

Probably the most significant issue observed during the pilot experiments which, as a result contributed to the design of the main experiment and analytical method of this thesis was the existence of transients in the integrated KF position solution, as illustrated in

Figure 3.4, but are not propagated to the accelerometers' (calibrated by GPS) outputs, illustrated in Figure 3.5. The snapshot of captured data in

Figure 3.4, corresponds to the IMU/GPS sensor being held by the ear while walking for 6 steps, so the true height solution should have been smoother (sinusoidal) without transients ("jumps"), highlighted with orange arrows. These transient artefacts appear every 250ms, which is the GPS update time interval (equivalent to a 4Hz update rate in the frequency domain) of the KF fusion algorithm. This is a result of pedestrian motion-induced dynamics to the KF fusion algorithm, in other words, their magnitude would be less if the measurement update interval was smoother and the IMU specifications were better (assuming that motion does not increase the IMU noise). The transients' magnitude also depends on the KF design, e.g. which states are estimated, although this information, as well as further details on the tuning of the internal KF of the IMU/GPS are proprietary. A solution to this issue would be independent motion capture (MoCap) using an opto-electronical system, as discussed in Section 2.3.1. with regards to human motion capture methods.

Further analysis (detailed in Section 6.1.1) showed that these transients can introduce significant artefacts on the simulated GNSS carrier-tracking loops. Therefore, it was decided that for further analysis, the inertial position solution had to be calculated, which does not introduce artificial tracking slips (i.e. cycle slips or false frequency locks) to the simulated GNSS carrier-tracking loops, as explained in Section 6.1.1. In addition, a proposed method of detecting and removing these transients is presented in Appendix B, noting that another option of removing the transients would be the development of a custom Kalman Smoother which would fuse the inertial and GPS measurements of the Xsens MTi-G; however, it was not feasible to implement a custom Kalman Smoother, as the Xsens MTi-G could not readily provide the GPS pseudorange measurements which are required for this method to work.

In summary, the knowledge gained from the pilot experiments in terms of experimental protocol design and data processing, supported the main experiment of the thesis and the analysis of the collected data in terms of:

- Developing an appropriate IMU/GPS calibration protocol, as discussed in this section.
- Identifying quality issues with the MoCap profiles and respond by adjusting the motion profiles, e.g. smoother motion dynamics by excluding running scenarios.
- Identifying quality issues with the MoCap data processing method and develop mitigation strategies, i.e. collect data in raw mode to enable different KF selection during post-processing; detection/smoothing of GPS transients (see Appendix B); and finally, the decision to use the inertial solution which is transient-free (since no sensor fusion is involved).

The Chapter continues with the description of the thesis' main experiment.

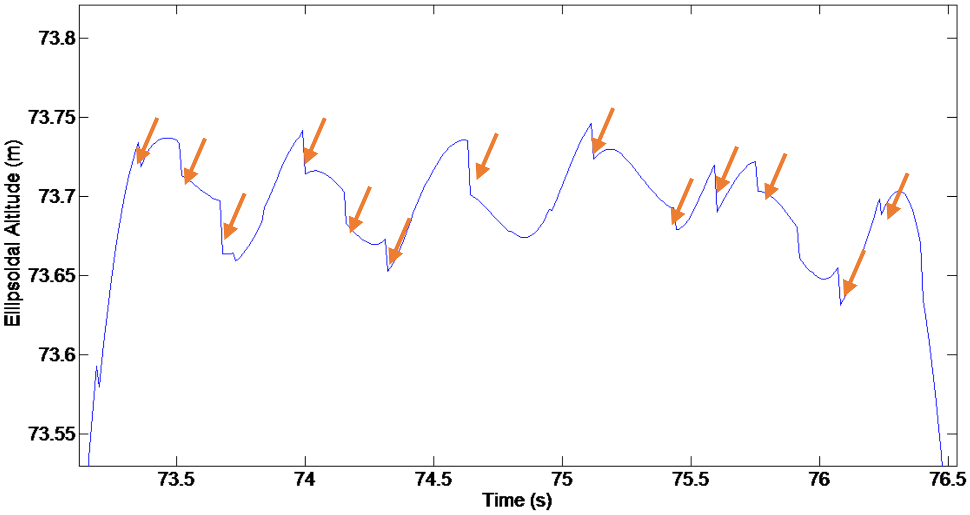

Figure 3.4. Examples of GPS-induced transients (pointed by orange arrows) to KF Altitude solution (blue line).

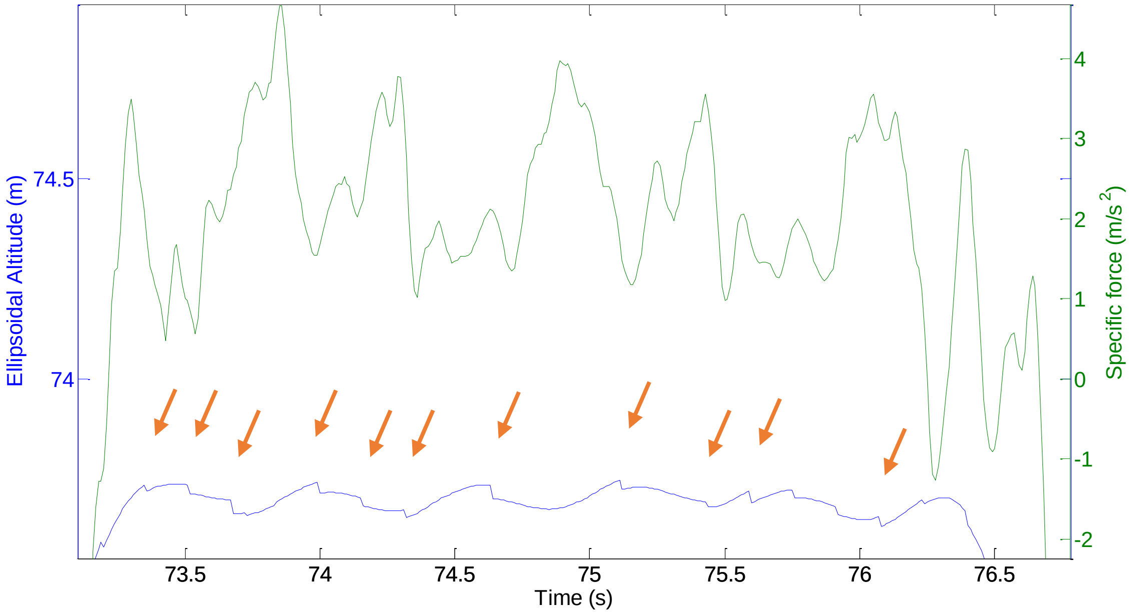

Figure 3.5. Transients (pointed by orange arrows) can be observed in the integrated KF solution (blue line), but not in the inertial specific force output (green line) of Xsens MTi-G

## 3.3 Main Experiment

The purpose of the main experiment was to provide the real human motion data in order to test the effects of human dynamics on GNSS carrier tracking, to determine which movements most affect the GNSS carrier tracking performance and also, to validate, or not, the pedestrian motion model (see Chapter 6), according the aims of this thesis (see Section 1.2). The human MoCap data collected during the main experiment are aimed at driving the simulated GNSS carrier tracking loops (SGCTLs, see Chapter 4); the requirements of the human MoCap the can be summarised as:

- Be representative of underlying motion, e.g. follow a trend when the user moves the IMU/GPS up or down.
- Not include any artefacts due to the applicable MoCap experimental protocol, e.g. insufficient calibration.
- Not include artefacts due to the Xsens internal IMU/GPS measurements' fusion algorithm implementation, e.g. transients when updating the integrated solution using measurements from different sensors.

The main experiment of the thesis took place in a field with a good sky view enabling GPS fix availability throughout the duration of the experiment, in order to calibrate the IMU prior to MoCap, while during MoCap it is essential for heading observability and accelerometers' bias correction. The main experiment was conducted in the last week of June 2013. The main experiment comprised human MoCap profiles with the pedestrian replicating the usage of a smartphone, i.e. walk holding, answering and/or texting, and scenarios where the smartphone is carried at different locations on the human body, e.g. in a pocket, in a bag, or on an arm band, as summarised in Table 3.2 (noting that the term "IMU" used in this table refers to the Xsens MTi-G IMU/GPS device, for brevity).

To conduct the MoCap experiments, a specific experimental protocol was developed and followed, whose workflow is illustrated in Figure 3.6. The first stage is to set-up all the equipment, including the magnetic calibration of the IMU, explained in [130]. The magnetic calibration compensates for the hard iron effects, distorting permanently the geomagnetic field, and involves rotating the sensor in all different directions (at least 30 samples are recommended by the manufacturer) in order to map the magnetic field distortions in 3D (with a 2D also available), compared to the internally stored values during factory calibration. Then the next two stages involve start recording video (using a digital camera), covering visually the area where the MoCap takes place and start recording the output data from the IMU/GPS device through its proprietary user interface "MT Manager".

The IMU/GPS device is calibrated for the accelerometer biases by comparing (and correcting) the inertial solution, fusing the regular GPS updates, which are not subject to drift. This requires that the IMU/GPS device is subject to significant horizontal acceleration prior to the actual motion capture phase in order to calibrate the acceleration biases (expressed) in the horizontal plane, as the calibration along the vertical direction takes place anyway due to the combined effect of the gravity and the accelerometers' chassis causing the proof mass to exhibit a reaction force to gravity (of equal magnitude but opposite sign). It is worth noting that the GPS antenna was in fixed relative position w.r.t the IMU chassis, with the lever arm at (0, 0, 0.05), units in m, in body frame axes. This lever arm effect is compensated by the Xsens "MT Manager" processing software.

| ID | **MoCap Scenario Name:** Description |
|---|---|
| **1** | **L_H_W30m_S**: Walk 30m holding the IMU (swinging the arm), then stop. |
| **2** | **L_P_W30m_S**: Walk 30m with the IMU in the pocket, then stop. |
| **3** | **L_B_W30m_S**: Walk 30m with the IMU in a back-pack, then stop. |
| **4** | **L_A_W30m_S**: Walk 30m with the IMU in an arm-band, then stop. |
| **5** | **L_H_W10m_A_W10m_D_W10m**: Hold the IMU swinging the arm, walk 10m, perform an "answer the phone" gesture with the IMU, walk 10m, perform a "put the phone down" gesture, walk 10m, stop walking and perform a "stop answering the phone" gesture. |
| **6** | **L_P_W10m_A_W10m_D_W10m**: Walk 10m with the IMU in the right pocket, perform an "answer the phone" gesture with the IMU, walk 10m, perform a "put the phone down" gesture, walk 10m, stop walking and perform a "stop answering the phone" gesture. |
| **7** | **L_H_W15m_A_W15mS_D**: Hold the IMU (arm swing), walk 15m, perform an "answer the phone" gesture, walk 15m, stop walking and perform a "put the phone down" gesture. |
| **8** | **L_P_W15m_A_W15mS_D**: IMU in the pocket, walk 15m, perform an "answer the phone" gesture with the IMU, walk 15m, stop walking and perform a "put the phone down" gesture. |
| **9** | **L_H_W15m_T_W15mS_D**: Hold the IMU, walk 15m, perform a "send a text" gesture with the IMU, walk 15m, stop walking and perform a "put the phone down" gesture. |
| **10** | **L_P_W15m_T_W15mS_Dh**: IMU in the right pocket, walk 15m, perform a "send a text" gesture with the IMU, walk 15m, stop walking and perform a "put the phone down" gesture with the IMU and keep holding it (i.e. don't put it back in the pocket). |
| **11** | **L_H_J30mS**: Jog 30m holding the IMU, stop. |
| **12** | **L_P_J30mS**: Jog 30m, with the IMU in the pocket, stop. |
| **13** | **L_H_W30m_U_W30m**: Walk 30m holding the IMU (swinging the arm), turn around (U-turn) and walk another 30m holding the phone, stop. |
| **14** | **L_P_W30m_U_W30m**: Walk 30m having the IMU in the right pocket, turn around (U-turn) and walk another 30m (having the IMU in the right pocket), stop. |
| **15** | **L_H_W15m_A_W15mUW15m_Dh_W15mS**: Hold the IMU (swinging the arm), walk 15m, perform an "answer the phone" gesture with the IMU, walk 15m, turn around (U-turn) and walk another 15m, perform a "put the phone down" gesture, walk another 15m and stop. |
| **16** | **L_H_W15m_T_W15mUW15m_Dh_W15mS**: Hold the IMU (swinging the arm), walk 15m, perform an "send a text/email" gesture with the IMU, walk 15m, turn around (U-turn) and walk another 15m, then perform a "put the phone down" gesture, walk 15m and stop. |
| **17** | **L_P_W15m_T_W15mUW15m_Dp_W15mS**: Walk 15m having the IMU in the right pocket, perform an "send a text/email" gesture, walk 15m, turn around (U-turn) and walk another 15m, perform a "put the phone in the pocket" gesture, walk 15m and stop. |
| **18** | **L_P_W15m_A_W15mUW15m_Dp_W15mS**: Walk 15m having the IMU in the right pocket, perform an "answer the phone" gesture, walk 15m, turn around (U-turn) and walk another 15m, perform a "put the phone in the pocket" gesture, walk 15m and stop. |

Table 3.2. Descriptions of MoCap scenarios of the main experiment

During the next stage of the workflow, the user goes to the initial position and performs event-based synchronisation (EBS) between the IMU/GPS device and the video camera; this involves moving the IMU/GPS device up and down for five times, which is a distinct movement that makes easier to recognise the start and end of the actual movement during data post-processing. The user performs the motion scenario in question and, when finished, performs EBS again. The EBS process is particularly important in order to identify the time instants when MoCap starts and stops during the post-processing of the captured data, distinguishing between MoCap and the calibration phase.

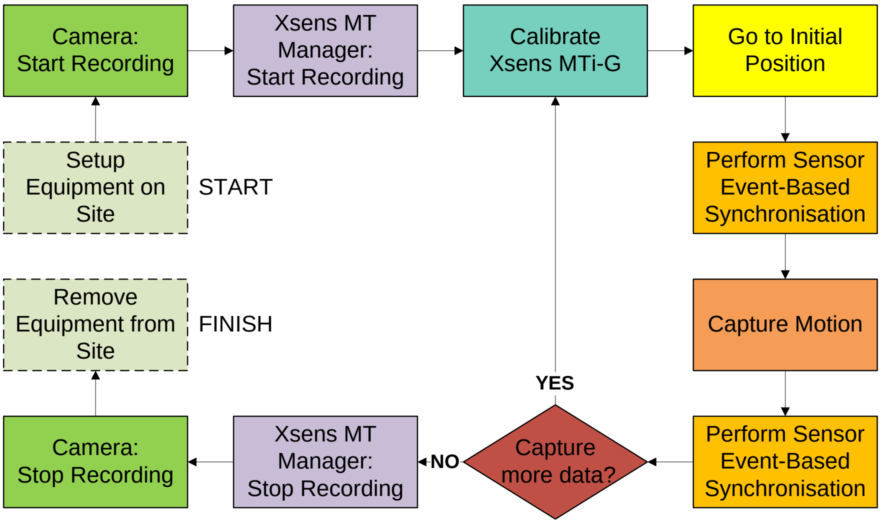
Camera: Start Recording
Xsens MT Manager: Start Recording
Calibrate Xsens MTi-G
Go to Initial Position
Setup Equipment on Site
START
Perform Sensor Event-Based Synchronisation
Remove Equipment from Site
FINISH
Capture Motion
YES
Camera: Stop Recording
Xsens MT Manager: Stop Recording
NO
Capture more data?
Perform Sensor Event-Based Synchronisation

Figure 3.6. Motion capture workflow

The selected calibration manoeuvres involve walking in a circle of 3 metres radius, as shown in Figure 3.7, and rotating the IMU describing a 3D sinusoidal motion, as illustrated in Figure 3.8. During this movement of the IMU it is important not to exceed the gyros' maximum output range (300 degrees/second). Also, the GPS antenna should be in a fixed position w.r.t the IMU. The IMU is rotated three times on each direction of the 3m-radius circle following the "figure of 8" pattern. This movement pattern was selected in order to force the triad of gyros of the IMU to sense angular rate about all 3D directions, as well as the triad of accelerometers to sense horizontal specific force, noting that in the vertical dimension, they sense specific force anyway due to reaction of the chassis walls to the gravity exerted on the proof mass. This important in order to observe the heading of the IMU, using accelerometer outputs combined with regular GPS corrections, as only horizontal acceleration can raise horizontal velocity errors. In other words, the heading of a static IMU could not be observed using this method. The relative accuracy of the pre-calibrated data was found by visual inspection in excess of 20m, with the sensor trajectory not following the actual motion trend. Visually inspected post-calibration data do not exhibit these errors.

Also, following the discussion from Section 2.2.2 about the IMU calibration, the integrated KF would calibrate the IMU errors along the vertical axis of the navigation solution as a result of the accelerometers' proof masses reaction to gravity, causing the inertial velocity error to grow (linearly) and the position (quadratically), which can then be corrected by the height (and vertical velocity) solution from GPS.

The thesis continues with Chapter 4, which details the methods of implementing (in Matlab) and validating the simulated GNSS carrier-tracking loops (SGCTLs), used to assess the impact of the MoCap profiles explained in this chapter, as well as the synthetic motion profile (output of the pedestrian motion model, described in Chapter 5).

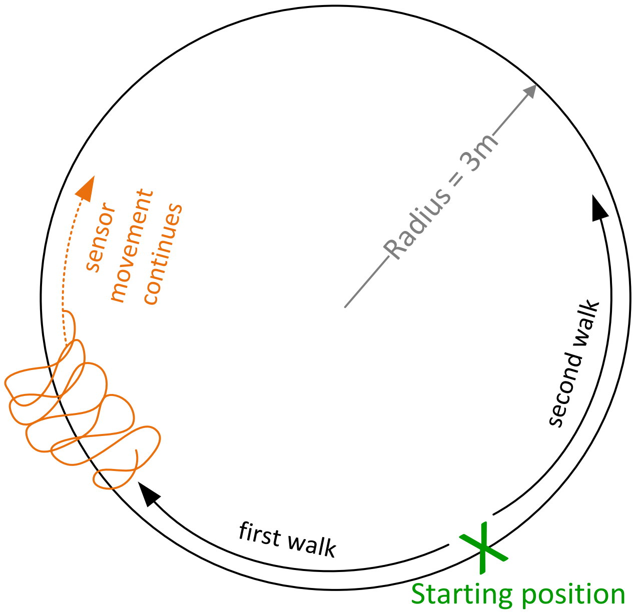

Figure 3.7. IMU/GPS sensor calibration – ground trajectory

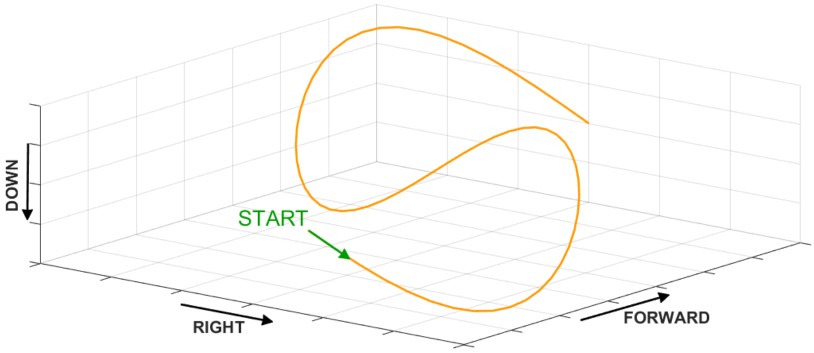

Figure 3.8. IMU/GPS hand-held sensor calibration Forward, Right and Down trajectory (example covers only one period of movement)

# Chapter 4. GNSS Carrier-Tracking Loops: Simulation and Validation

This chapter explains the method of simulating GNSS carrier-tracking loops in Matlab, based on the literature review presented in Chapter 2. This chapter is divided into four sections, the first (Section 4.1) details the assumptions made in order to simulate GNSS carrier-tracking loops (SGCTLs) in Matlab, the second, Section 4.2, describes the implementation of a simulated GNSS phase lock loop (SGPLL) in Matlab, the third, Section 4.3, describes the implementation of a simulated GNSS frequency lock loop (SGFLL) in Matlab, and finally, Section 4.4. concludes the chapter describing the validation process and results of the SGCTLs, under static antenna conditions in the presence of white (Gaussian) noise, generated in Matlab. The SGCTLs presented in this chapter provide the testbed in order to assess the impact of human captured motion (MoCap, discussed in Chapter 3) on the carrier-tracking function of GNSS receivers, and validate the pedestrian motion model (PMM) by comparing the effect of the PMM synthetic output (3D trajectory of the GNSS receiver antenna) on the SGPLL to the real MoCap profiles' effect. The validation process of the PMM is described in Chapter 5, while the results of analysing the impact of human MoCap and PMM synthetic data on the SGCTLs are detailed in Chapter 6.

The work presented in Chapter 4 was essential in order to address the research questions of the thesis (see Section 1.2) as it provides the testbed for testing the effects of real (MoCap) human motion (whose capture process is described in Chapter 3) and the effects of synthetic motion output from the pedestrian motion model (described in Chapter 5) on the simulated GNSS carrier-tracking loops (SGCTLs) developed for this thesis.

## 4.1 Simulation of GNSS Carrier-tracking Loops

This section presents a method of simulating GNSS carrier-tracking loops in Matlab. The development of the simulated GNSS carrier-tracking loops (SGCTLs) provides the platform for:

- Analysing the effects of real pedestrian motion on the performance of GNSS carrier-tracking loops.
- Validating the pedestrian motion model (PMM) – discussed in Chapter 5, in order to ensure that the synthetic output, i.e. 3D trajectory of the PNT equipment attached or held near to the human body, has a similar impact on the performance of SGCTLs as real motion, whose capture was discussed in Chapter 3.

The operation principles of GNSS carrier-tracking loops were discussed in Section 2.2.3. For the simulation of carrier-tracking loop models in Matlab, the following assumptions have been made:

- The code-tracking error is negligible, i.e. the incoming PRN code and the locally generated replica PRN code in the receiver are fully aligned and remain so during the simulation experiments. As a result, the autocorrelation function output of the authentic and replica PRN codes is assumed to be unity and code-tracking loops are not explicitly simulated. This is because the maximum position amplitude for pedestrian motion (e.g. 2m for moving a sensor from the floor to a shelf), is much less than the length of a GPS L1 C/A code chip, about 293m.
- The carrier-tracking loops have a fixed gain (response), independent of $C/N_0$.
- The carrier frequency and phase errors (difference between incoming and locally generated carrier signals) can be modelled by the standard Gaussian distribution, and are initially within 3σ (where sigma σ equals 1 standard deviation) from their zero mean.
- The signal noise standard deviation after the AGC stage is constant and equal for both I and Q channels.
- Factors such as band-limiting, quantisation and clock noise have not been modelled. It is worth noting that band-limiting and quantisation noise typically reduce the $C/N_0$ of a real GNSS receiver by 1 dB [62].
- The tracking noise in the I and Q channels is assumed to be standard Gaussian.
- The observed frequency and phase errors by the SGCTLs are only due to relative motion between the GNSS antenna and the reference satellite along the LOS.
- This analysis assumes a GPS L1 C/A signal, without loss of generalisation for other GNSS signals.
- Any factors not mentioned as part of the simulation, are out of scope for the present analysis, or they are considered negligible, e.g. rounding errors in Matlab.

The SGCTLs comprise a simulated GNSS phase lock loop (SGPLL) and a simulated GNSS frequency lock loop (SGFLL), which are explained below.

## 4.2 Carrier Phase Lock Loop Simulation

Figure 4.1 illustrates a typical functional diagram of a (digital) phase lock loop (PLL), divided into 7 functional blocks (stages). The variables in this diagram refer to the range domain (m) for simplicity purposes, as true (incoming) range $\rho_{ca}$ and its derivative quantities (range rate $\dot{\rho}_{ca}$, range acceleration $\ddot{\rho}_{ca}$) are not cyclic variables, like phase. Starting from the lower-left part of this figure, the range and range rate (always refer to the LOS between the receiver and satellite APCs), enter the PLL (Stage 1). In the first epoch (k=1), the range error $\delta\rho_{ca}$ (difference between incoming and locally generated carrier phase/range) is initialised using a random number taken from a continuous uniform distribution *U*, which spans [-3σ(Φ),+3σ(Φ)] around the zero mean of the range (equivalent to carrier phase) tracking noise, Φ (units in m), whose standard deviation is denoted with *σ(Φ)*. Assuming a Costas two-quadrant arctan range error discriminator, this range error standard deviation is given by [3]:

$$\sigma(\Phi) \approx \frac{c}{2\pi f_{ca}} \sqrt{\frac{B_{L\_CA}}{(c/n_0)} \left[ 1 + \frac{1}{2(c/n_0)\tau_a} \right]} \quad , \tag{4.1}$$

where *c* is the speed of light in vacuum (m/s), $f_{ca}$ is the carrier central frequency (Hz), $B_{L\_CA}$ is the effective PLL bandwidth (Hz), $\tau_a$ is the accumulation time interval between two PLL iterations, and $c/n_0$ is the carrier-power-to-noise-density ratio. Note that $c/n_0$ is equivalent to the carrier-power-to-noise-density ratio in logarithmic scale (dB) which is denoted as $C/N_0$, with capital letters for clarity, following the equation:

$$C/N_0 = 10\log_{10}(c/n_0) \Leftrightarrow c/n_0 = 10^{\frac{C/N_0}{10}} \quad . \tag{4.2}$$

The range rate error $\delta\dot{\rho}_{ca}$, which is equivalent to Doppler shift error in the frequency domain, is also initialised using a random number taken from a continuous uniform distribution *U*, between [-3σ,+3σ] around a zero mean of the range rate (carrier frequency) tracking noise, which for a PLL with a Costas two-quadrant Arctan discriminator is [3]:

$$\sigma(\delta\dot{\rho}_{ca}) \approx \sqrt{\frac{0.72 B_{L\_CA}}{\tau_a}} \sigma(\Phi) \quad . \tag{4.3}$$

The initialised range and range rate errors of Stage 1, are then converted to phase and frequency errors, respectively, in Stage 2, solving for phase and frequency errors in Eqs (4.4) and (4.5):

$$\delta\rho_{ca} = -\frac{\lambda_{ca}}{2\pi}\delta\varphi_{ca} \quad , \tag{4.4}$$

$$\delta\dot{\rho}_{ca} = -\frac{c}{f_{ca}}\delta f_{ca} \quad , \tag{4.5}$$

where $\lambda_{ca}$ the carrier wavelength (m) and considering any relativistic effects negligible for pedestrian applications in this thesis.

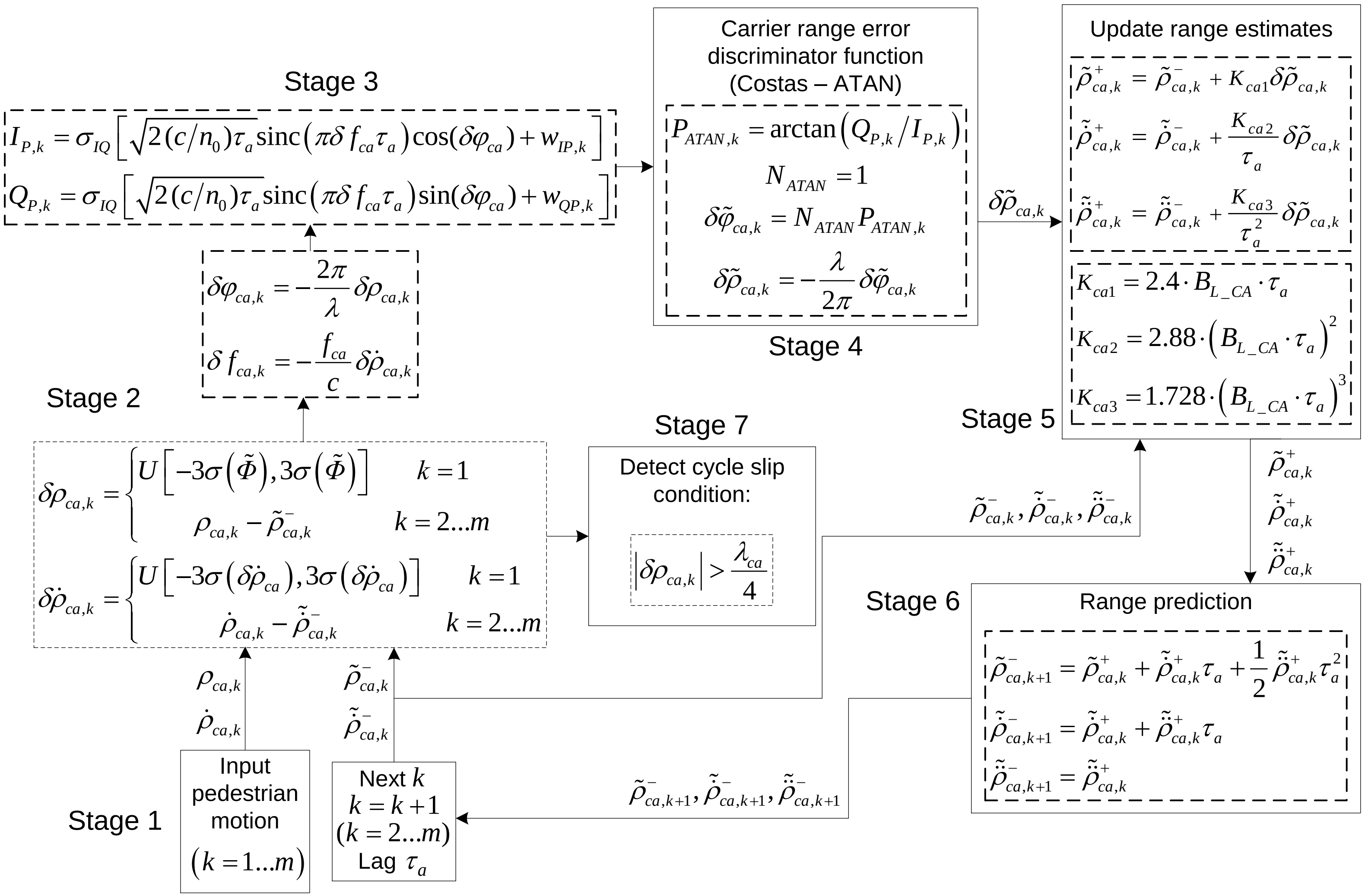
Stage 3
$I_{P,k} = \sigma_{IQ}\left[\sqrt{2(c/n_0)\tau_a}\,\mathrm{sinc}\left(\pi\delta f_{ca}\tau_a\right)\cos(\delta\varphi_{ca}) + w_{IP,k}\right]$
$Q_{P,k} = \sigma_{IQ}\left[\sqrt{2(c/n_0)\tau_a}\,\mathrm{sinc}\left(\pi\delta f_{ca}\tau_a\right)\sin(\delta\varphi_{ca}) + w_{QP,k}\right]$
Carrier range error discriminator function (Costas – ATAN)
$P_{ATAN,k} = \arctan\left(Q_{P,k}/I_{P,k}\right)$
$N_{ATAN} = 1$
$\delta\tilde{\varphi}_{ca,k} = N_{ATAN}P_{ATAN,k}$
$\delta\tilde{\rho}_{ca,k} = -\frac{\lambda}{2\pi}\delta\tilde{\varphi}_{ca,k}$
Stage 4
$\delta\tilde{\rho}_{ca,k}$
Update range estimates
$\tilde{\rho}^{+}_{ca,k} = \tilde{\rho}^{-}_{ca,k} + K_{ca1}\delta\tilde{\rho}_{ca,k}$
$\dot{\tilde{\rho}}^{+}_{ca,k} = \dot{\tilde{\rho}}^{-}_{ca,k} + \frac{K_{ca2}}{\tau_a}\delta\tilde{\rho}_{ca,k}$
$\ddot{\tilde{\rho}}^{+}_{ca,k} = \ddot{\tilde{\rho}}^{-}_{ca,k} + \frac{K_{ca3}}{\tau_a^2}\delta\tilde{\rho}_{ca,k}$
$K_{ca1} = 2.4\cdot B_{L\_CA}\cdot\tau_a$
$K_{ca2} = 2.88\cdot\left(B_{L\_CA}\cdot\tau_a\right)^2$
$K_{ca3} = 1.728\cdot\left(B_{L\_CA}\cdot\tau_a\right)^3$
Stage 5
$\delta\varphi_{ca,k} = -\frac{2\pi}{\lambda}\delta\rho_{ca,k}$
$\delta f_{ca,k} = -\frac{f_{ca}}{c}\delta\dot{\rho}_{ca,k}$
Stage 2
$\delta\rho_{ca,k} = \begin{cases} U\left[-3\sigma\left(\tilde{\Phi}\right), 3\sigma\left(\tilde{\Phi}\right)\right] & k=1 \\ \rho_{ca,k} - \tilde{\rho}^{-}_{ca,k} & k = 2...m \end{cases}$
$\delta\dot{\rho}_{ca,k} = \begin{cases} U\left[-3\sigma\left(\delta\dot{\rho}_{ca}\right), 3\sigma\left(\delta\dot{\rho}_{ca}\right)\right] & k=1 \\ \dot{\rho}_{ca,k} - \dot{\tilde{\rho}}^{-}_{ca,k} & k = 2...m \end{cases}$
Stage 7
Detect cycle slip condition:
$\left|\delta\rho_{ca,k}\right| > \frac{\lambda_{ca}}{4}$
$\tilde{\rho}^{-}_{ca,k}, \dot{\tilde{\rho}}^{-}_{ca,k}, \ddot{\tilde{\rho}}^{-}_{ca,k}$
$\tilde{\rho}^{+}_{ca,k}$
$\dot{\tilde{\rho}}^{+}_{ca,k}$
$\ddot{\tilde{\rho}}^{+}_{ca,k}$
Stage 6
Range prediction
$\tilde{\rho}^{-}_{ca,k+1} = \tilde{\rho}^{+}_{ca,k} + \dot{\tilde{\rho}}^{+}_{ca,k}\tau_a + \frac{1}{2}\ddot{\tilde{\rho}}^{+}_{ca,k}\tau_a^2$
$\dot{\tilde{\rho}}^{-}_{ca,k+1} = \dot{\tilde{\rho}}^{+}_{ca,k} + \ddot{\tilde{\rho}}^{+}_{ca,k}\tau_a$
$\ddot{\tilde{\rho}}^{-}_{ca,k+1} = \ddot{\tilde{\rho}}^{+}_{ca,k}$
$\rho_{ca,k}$
$\dot{\rho}_{ca,k}$
$\tilde{\rho}^{-}_{ca,k}$
$\dot{\tilde{\rho}}^{-}_{ca,k}$
Stage 1
Input pedestrian motion
$\left(k = 1...m\right)$
Next $k$
$k = k+1$
$(k = 2...m)$
Lag $\tau_a$
$\tilde{\rho}^{-}_{ca,k+1}, \dot{\tilde{\rho}}^{-}_{ca,k+1}, \ddot{\tilde{\rho}}^{-}_{ca,k+1}$

Figure 4.1. Custom PLL implementation for pedestrian motion.

This is essential in order to generate the simulated post-correlated prompt I and Q signals (entering the carrier tracking loops) in Stage 3, using the equations [3]:

$$\begin{aligned} I_P(t_{sa}) &= \sigma_{IQ}\left[\sqrt{2\,(c/n_0)\tau_a}\,R(x)D(t_{st})\mathrm{sinc}(\pi\delta f_{ca}\tau_a)\cos(\delta\phi_{ca}) + w_{IP}(t_{sa})\right] \\ Q_P(t_{sa}) &= \sigma_{IQ}\left[\sqrt{2\,(c/n_0)\tau_a}\,R(x)D(t_{st})\mathrm{sinc}(\pi\delta f_{ca}\tau_a)\sin(\delta\phi_{ca}) + w_{QP}(t_{sa})\right] \end{aligned}, \qquad (4.6)$$

where $I_P$ and $Q_P$ are the prompt I and Q signals, respectively, $t_{sa}$ is the time of signal arrival at the receiver APC, $\sigma_{IQ}$ is the standard deviation of the normalised amplitude as it leaves the receiver AGC (and has been omitted in the Matlab code implementation as by dividing Q over I in the arctan discriminator in Stage 4, they both cancel out). $R(x)$ is the code autocorrelation function output (assumed as unity for fully aligned received and internally generated PRN codes, therefore omitted from the Matlab code implementation for optimisation purposes). $D(t_{st})$ is the navigation data bit (-1 or 1) which is referenced to the time of signal transmission from the satellite $t_{st}$; this is also omitted in the Matlab code implementation as the Costas phase discriminator used in the present analysis is not sensitive to navigation bit transitions over the accumulation time interval $\tau_a$. Finally, $w_{IP}$ and $w_{QP}$ denote the noise on the accumulated correlator outputs (Is and Qs) and are modelled by standard Gaussian distributions, noting that this is generated as pseudo-random number sequence in Matlab which is different for I and Q channels and for repeatability purposes the seed is not randomised between simulation runs, but fixed to the (hard-coded) default value set in Matlab software. As seen in Eq. (4.6), the higher the $c/n_0$ the lower the impact that the simulated tracking noise has in the Is and Qs generation. It is worth underlining that the code tracking noise is significant at low $C/N_0$, so the simulation investigates $C/N_0$ values between 20dB-Hz and 50dB-Hz at 1dB-Hz steps, as code tracking noise is not part of the simulated parameters.

In Stage 4, the Arctan (Costas) discriminator function determines the range (phase) error based on the I and Q signals, generated in the previous stage. The selection of a Costas discriminator was based on the fact that it is not sensitive to navigation bit transitions, since its pull-in range is ±π/2 rad (see Section 2.2.3), so the input-output curve repeats itself every π rad, as illustrated in Figure 4.2. It is worth noting that the input to the discriminator is the observed phase error $\delta\tilde{\varphi}_{ca}$ (derived by the Is and Qs) and also, that within the pull-in range of the discriminator, the input-output curve is linear, as shown in Figure 4.2, which is why it is commonly used in GNSS receivers [131]. A Costas Arctan discriminator ensures a linear input to the tracking loop so that the effective gain is constant, within the defined (-π/2, π/2) pull-in range, as Figure 4.2 illustrates. Cycle slips occur when the absolute tracking error exceeds π/2 rad, i.e. they can tolerate a sudden range (phase) jump up to a quarter of the carrier wavelength between two subsequent PLL epochs. In this stage, the observed range tracking error $\delta\tilde{\rho}_{ca}$ is converted from phase (rad) back to the range (m), noting that although this is not the case in a real PLL, for simulation purposes it is easier to work with quantities which are measured in meters (range) and m/s (range rate).

In Stage 5, the observed range error $\delta\tilde{\rho}_{ca}$ (m) from the discriminator is used to update the current range $\rho_{ca}^{+}$, range rate $\dot{\rho}_{ca}^{+}$ and range acceleration $\ddot{\rho}_{ca}^{+}$ estimates via the loop filter (gain function) of the PLL, which controls the magnitude of the loop filter response (compensation) w.r.t. the input range error. This compensation stage is essential in order to provide a stable input into the NCO, which will then provide a stable response in order to reduce the range error in the next PLL epoch. The gain values $K_{ca1}$, $K_{ca2}$ and $K_{ca3}$, are calculated as a function of the PLL effective (3dB) bandwidth, denoted as $B_{L\_CA}$ between 5Hz and 20Hz, at 5Hz intervals. For simulation optimisation purposes, the intermediate $B_{L\_CA}$ values' results are not calculated, noting that in a real receiver the NCO noise (not simulated) is the dominant type of noise for bandwidths below 5Hz therefore simulation excludes these values. The coherent integration time interval $\tau_a$ (typically it is 0.01s), has been selected to match the time interval of the input data from Xsens MTi-G (captured at 100Hz output rate, as explained in Section 3.3), so that all captured data could be used in the GNSS tracking simulation, without any oversampling or under-sampling issues.

In Stage 6, the estimated range, range rate and range acceleration are used to update the respective predicted quantities for the next PLL iteration, denoted with a "k+1" subscript for the epoch and a minus superscript for their status, i.e. predicted but not yet updated. In the present analysis, these predicted quantities are used to determine the phase $\delta\varphi_{ca}$ and frequency $\delta f_{ca}$ tracking errors, used to calculate the Is and Qs. This stage is equivalent to the generation of NCO commands to drive the carrier NCO. The loop then starts again from Stage 1 to produce the new range/ range rate errors as the difference between the predicted range values (Stage 6 of the previous epoch) and new input range values (next epoch Stage 1) from the incoming GNSS signal. The estimated range error shows whether a cycle slip has occurred (Stage 7), depending on whether the absolute range error is larger than the range error discriminator pull-in range (a quarter of the carrier wavelength – equivalent to the $\pm\pi/2$ rad).

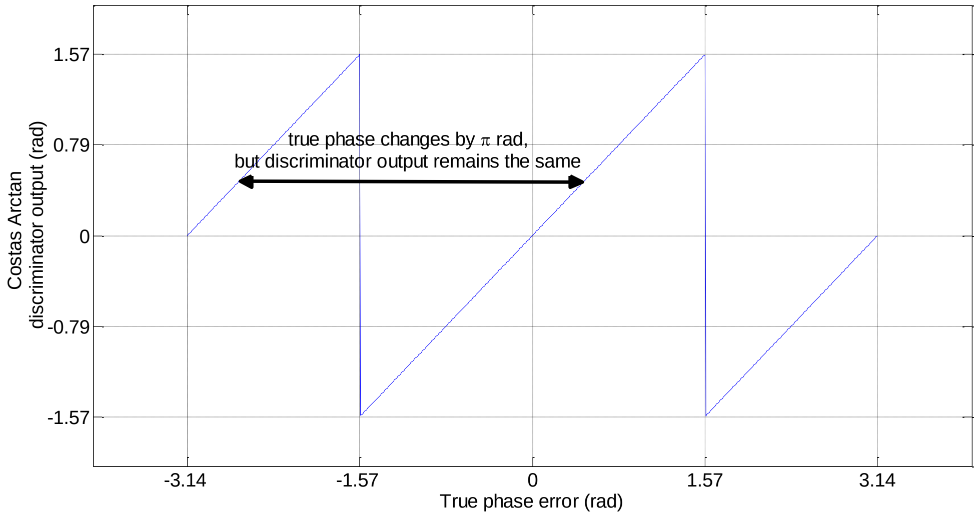

Figure 4.2. Costas Arctan (2 quadrant) discriminator is not sensitive to navigation data-bit transitions over the accumulation time interval.

It is important to mention at this point that the synthetic or captured pedestrian motion (3D trajectory), will be input into the PLL by projecting the 3D motion dynamics along the LOS between the receiver and a (hypothetically static for the duration of motion) satellite, e.g. for a simulated static satellite at zenith (above the pedestrian subject), any sudden vertical jump of the GNSS antenna (height change) between two epochs is equivalent to the initial range/ range rate errors input into the SGCTLs. Similarly, assuming a static satellite on the local horizon, e.g. at a Northbound direction, any horizontal movement along that direction would introduce range/range rate errors in the PLL simulation. Defining static satellites (which do not introduce additional tracking errors) is considered acceptable for simulation purposes in this thesis, assuming that the real satellite geometry does not change significantly over the course of a typical one-minute pedestrian movement. In addition to that, in a real receiver tracking satellite signals, the motion of the satellites is known (from the almanac) and smooth, so it would not significantly affect tracking errors.

## 4.3 Carrier Frequency Lock Loop Simulation

A custom FLL simulated in Matlab is illustrated in Figure 4.3, noting that the main differences between typical PLLs and FLLs are the discriminator function (FLL discriminators cannot estimate phase error, which PLL discriminators can do) and the loop filter response which tries to compensate for the frequency error in FLLs instead of the phase error (in PLLs), as described in this section. Figure 4.3 illustrates a typical functional diagram of an FLL, adjusted to the purposes of this thesis, i.e. converting the frequency error to range rate error in order to have easier to handle units (m/s) instead of Hz, when feeding the pedestrian motion into the FLL, similar to what was previously described for the simulated custom PLL.

In Stage 1, the GNSS equipment antenna velocity (range rate $\dot{\rho}_{ca}$ – induced by pedestrian motion dynamics) along the LOS is input into the FLL. Then in Stage 2, for the first epoch (initialisation of the FLL), the range rate error $\delta\dot{\rho}_{ca}$ is a random number taken from a continuous uniform distribution (denoted as *U*), with an absolute value less than three standard deviations of the carrier frequency tracking noise (units in m/s), which is given by [3]:

$$\sigma\left(\delta\dot{\rho}_{cf}\right) \approx \frac{c}{\pi f_{ca}\tau_a}\sqrt{\frac{B_{L\_CF}}{\left(c/n_0\right)}\left[1+\frac{1}{\left(c/n_0\right)\tau_a}\right]} \tag{4.7}$$

For subsequent epochs in Stage 2, the observed range rate error $\delta\tilde{\dot{\rho}}_{ca}$ is calculated as the difference of the input velocity (range rate) with the forward prediction of the range rate from the previous epoch. The range rate error of Stage 2 is then converted to Doppler error $\delta\tilde{f}_{ca}$ and is used to generate the Is and Qs signals in Stage 3, using the same formulae from Eq. (4.6). The generation of the Is and Qs is similar to the one for the simulated PLL, with the only difference that only actual input from the captured pedestrian motion (projected along the LOS) is the range rate.

In Stage 4, the frequency error discriminator function uses the generated Is and Qs in Stage 3 (both from the current and previous epoch) in order to determine the frequency error, which is then converted to range rate error. A Costas arctan (two-quadrant) frequency error discriminator was selected because it is not sensitive to navigation data-bit transitions, which invert the incoming signal phase (π rad) over the correlator accumulation time interval $\tau_\alpha$, as well as because the specific discriminator provides a linear input-output curve within the pull-in range $\left(-\frac{1}{4\tau_\alpha},\frac{1}{4\tau_\alpha}\right)$. For the current implementation ($\tau_a$ = 0.01s), therefore the pull-in range will be (-25Hz, 25Hz).

Stage 5 comprises the loop filter, whose gain $K_{cf1}$ and $K_{cf2}$ (response magnitude to compensate the frequency error) is a function of the FLL effective bandwidth, as illustrated in Figure 4.3. During Stage 5, the current range rate $\tilde{\dot{\rho}}^{+}_{ca,k}$ and range acceleration $\tilde{\ddot{\rho}}^{+}_{ca,k}$ are estimated by updating the respective ones from the previous epoch. In real FLLs, this stage is the compensation of the frequency error in order to produce a stable signal that will feed the NCO. The current range rate and range acceleration estimates are then used to predict forward the respective quantities in the next FLL iteration in Stage 6, denoted with a "k+1" subscript for the epoch and a minus superscript for their status (i.e. predicted but not updated), which represents the NCO function in real FLLs. Finally, the loop starts over again with the predicted range rate of the previous iteration becoming the current (observed) range rate of the next epoch. The new range rate error then can be calculated as the difference between the current (observed) range rate and the range rate input from the pedestrian velocity projected along the LOS. This range rate error is then compared with the tolerance of the frequency error discriminator (converted in range rate units as well), in order to determine whether a false lock occurs in the current epoch or not. A false lock is detected when the absolute range rate error exceeds the pull-in range of the frequency error discriminator, i.e. on-quarter of a carrier wavelength $\lambda_{ca}$ divided by the coherent integration time interval for $\tau_a = 0.01$s, which is equivalent to (-12.5Hz, 12.5Hz) in the frequency domain.

This chapter continues with the validation of the SGCTLs in Matlab.

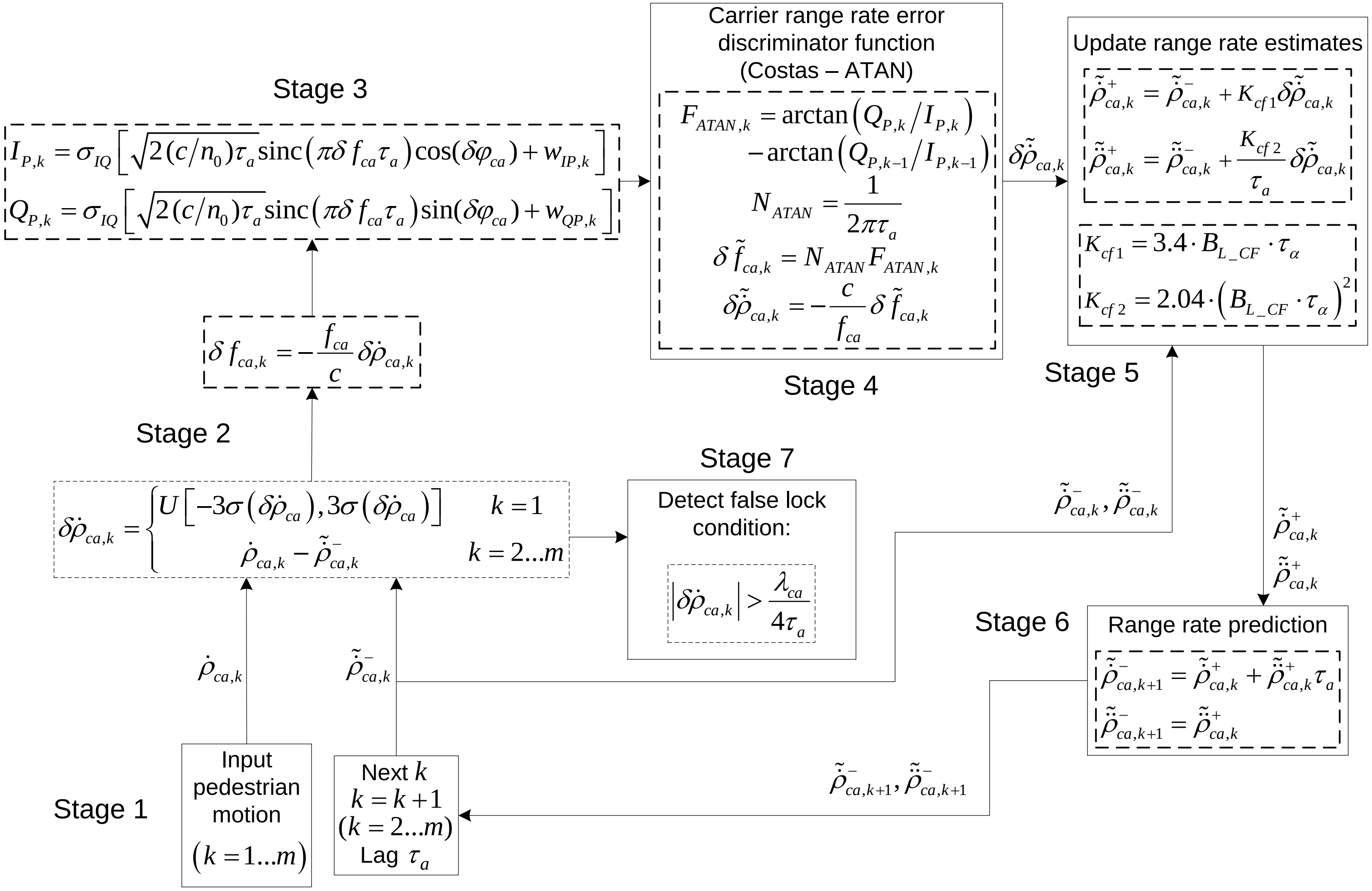
Stage 3
$I_{P,k} = \sigma_{IQ}\left[\sqrt{2(c/n_0)\tau_a}\,\mathrm{sinc}\left(\pi\delta f_{ca}\tau_a\right)\cos(\delta\varphi_{ca}) + w_{IP,k}\right]$
$Q_{P,k} = \sigma_{IQ}\left[\sqrt{2(c/n_0)\tau_a}\,\mathrm{sinc}\left(\pi\delta f_{ca}\tau_a\right)\sin(\delta\varphi_{ca}) + w_{QP,k}\right]$
Carrier range rate error discriminator function (Costas – ATAN)
$F_{ATAN,k} = \arctan\left(Q_{P,k}/I_{P,k}\right) - \arctan\left(Q_{P,k-1}/I_{P,k-1}\right)$
$N_{ATAN} = \frac{1}{2\pi\tau_a}$
$\delta\tilde{f}_{ca,k} = N_{ATAN}F_{ATAN,k}$
$\delta\tilde{\dot{\rho}}_{ca,k} = -\frac{c}{f_{ca}}\delta\tilde{f}_{ca,k}$
Stage 4
$\delta\tilde{\dot{\rho}}_{ca,k}$
Update range rate estimates
$\tilde{\dot{\rho}}^{+}_{ca,k} = \tilde{\dot{\rho}}^{-}_{ca,k} + K_{cf1}\delta\tilde{\dot{\rho}}_{ca,k}$
$\tilde{\ddot{\rho}}^{+}_{ca,k} = \tilde{\ddot{\rho}}^{-}_{ca,k} + \frac{K_{cf2}}{\tau_a}\delta\tilde{\dot{\rho}}_{ca,k}$
$K_{cf1} = 3.4\cdot B_{L\_CF}\cdot\tau_\alpha$
$K_{cf2} = 2.04\cdot\left(B_{L\_CF}\cdot\tau_\alpha\right)^2$
Stage 5
$\delta f_{ca,k} = -\frac{f_{ca}}{c}\delta\dot{\rho}_{ca,k}$
Stage 2
$\delta\dot{\rho}_{ca,k} = \begin{cases} U\left[-3\sigma\left(\delta\dot{\rho}_{ca}\right), 3\sigma\left(\delta\dot{\rho}_{ca}\right)\right] & k=1 \\ \dot{\rho}_{ca,k} - \tilde{\dot{\rho}}^{-}_{ca,k} & k=2...m \end{cases}$
Stage 7
Detect false lock condition:
$\left|\delta\dot{\rho}_{ca,k}\right| > \frac{\lambda_{ca}}{4\tau_a}$
$\tilde{\dot{\rho}}^{-}_{ca,k}, \tilde{\ddot{\rho}}^{-}_{ca,k}$
$\tilde{\dot{\rho}}^{+}_{ca,k}$
$\tilde{\ddot{\rho}}^{+}_{ca,k}$
Stage 6
Range rate prediction
$\tilde{\dot{\rho}}^{-}_{ca,k+1} = \tilde{\dot{\rho}}^{+}_{ca,k} + \tilde{\ddot{\rho}}^{+}_{ca,k}\tau_a$
$\tilde{\ddot{\rho}}^{-}_{ca,k+1} = \tilde{\ddot{\rho}}^{+}_{ca,k}$
$\dot{\rho}_{ca,k}$
$\tilde{\dot{\rho}}^{-}_{ca,k}$
$\tilde{\dot{\rho}}^{-}_{ca,k+1}, \tilde{\ddot{\rho}}^{-}_{ca,k+1}$
Stage 1
Input pedestrian motion $\left(k=1...m\right)$
Next $k$
$k=k+1$
$(k=2...m)$
Lag $\tau_a$

Figure 4.3. Custom FLL implementation for pedestrian motion.

## 4.4 Validation of SGCTLs

The validation of simulated GNSS carrier-tracking loops (SGCTLs) was conducted using a static test case scenario. This is because the static scenario can only induce cycle slips and/or false frequency locks to the SGCTLs due to the tracking noise contained in the I and Q signals entering the SGCTLs (simulated via Matlab pseudo-random sequences, as explained in Sections 4.2 and 4.3). Therefore, when the real motion scenarios are tested via the SGCTLs (using the same set of pseudo random sequences as in the static scenario testing), any *additional* observed cycle slips and/or false frequency locks can only be attributed to the presence of LOS dynamics, which is a useful distinction in order to understand and analyse the effect of the pedestrian motion induced to the sensor (projected along the LOS).

In this static scenario, (pseudo) random noise sequences, generated by standard Matlab tools, were injected in the SGCTLs in order to simulate the RF noise received by the GNSS antenna and is passing through the signal processing chain onto the tracking loops. The SGCTLs' response was analysed in terms of phase noise (for the simulated PLL only) and frequency noise (for both SGCTLs), for a range of $C/N_0$ and effective carrier tracking loop bandwidth sets of parameters through a Monte Carlo simulation in Matlab. The phase and frequency noise is assumed to be Gaussian. The results were plotted as tracking noise (vertical axis) versus $C/N_0$ (horizontal axis) for different tracking loop bandwidths (depicted as coloured solid lines) and compared against the theoretical tracking noise (1$\sigma$) curves (depicted as coloured dotted lines) as, as per Eqs. (4.1), (4.3) and (4.7). The baseline Matlab code used for the present analysis is reported in Appendix F on the CD. The parameters used for the simulation are:

- The number of runs per Monte Carlo simulation (MCS) is 10,000. Each MCS corresponds to one pair of $C/N_0$ and PLL/FLL effective bandwidth value, $B_{L\_CA}$ and $B_{L\_CF}$, respectively. Each run is initialised with a different tracking error (taken from the uniform distributions explained in Sections 4.2 and 4.3. Each MCS contains the same pseudo-random sequence, generated in Matlab using the same noise seed, in order for each pair of $C/N_0$ and effective bandwidth values to be subjected under the same noise profile. Assuming statistical independence of the results from each pseudo-random sequence run, the standard error (showing how close is the sample to the population standard deviation) is inversely proportional to the square root of the number of samples. In this case, 10,000 runs per MCS, mean that the results are 1% accurate w.r.t. their true values, i.e. the values they would have obtained if the simulation could run an infinite number of times. This level of accuracy is considered acceptable for this thesis.
- The number of simulation iterations is 5,000. This is defined by creating input (static) range and range rate datasets whose number of epochs is 5,000, noting that the time interval between

epochs is defined as 0.01s, in order to match the time interval that human motion data was captured (see Section 3.3). The number of iterations was chosen after a pilot MCS showing that the SGCTLs were able to track the input phase/frequency error (as appropriate), when the lowest effective bandwidth was used (5Hz for the SGPLL and 1Hz for the SGFLL), noting that the lower the bandwidth, the slower the response time to dynamic stress along the LOS.

The *validation criteria* considered in the current analysis are:

- The ratio of the simulated tracking noise standard deviation (SD), over the theoretical tracking noise SD (see Sections 4.2 and 4.3), should be within a factor of two, since the static tracking noise is bandlimited as a result of the input motion entering the SGCTLs (through the Is and Qs) at a rate of 100Hz, equivalent to samples at 0.01s time intervals, while the theoretical tracking noise assumes an infinite (pre-correlation) bandwidth. The factor of two maximum difference means that the ratio of the simulated static test case tracking noise SD over the theoretical tracking noise should not exceed two, or be less than half.
- The simulated PLL and FLL are expected to exhibit relatively more tracking slips (i.e. phase cycle slips and/or false frequency locks, as appropriate) due to tracking noise at a lower $C/N_0$ with higher effective bandwidths, e.g. a simulated PLL would exhibit more tracking noise-induced cycle slips below a $C/N_0$ of 30dB-Hz for a $B_{L\_CA}$ of 20Hz than a $B_{L\_CA}$ of 5Hz, noting that the SGCTLs would be more resistant to dynamic stress-induced tracking slips at higher effective bandwidths (as they would be able to respond faster to LOS dynamics).

Meeting the above-mentioned assumption means that no significant tracking slips (cycle slips and/or false frequency locks, as appropriate) are introduced by the SGCTLs when processing the real and synthetic (pedestrian motion model output) motion profiles in Chapter 6.

### 4.4.1 Custom PLL validation

Figure 4.4 illustrates the SGPLL simulation results for a static test scenario. The only contributing factor to the SGPLL tracking noise SD $\sigma\left(\delta\varphi_{PLL,S}\right)$, given the assumption that rounding errors are negligible, is the Matlab-generated pseudo-random noise sequences for the I and Q signals entering the SGPLL and the random initialisation of the range and range rate prediction errors, shown in stage 2 of Figure 4.1. The vertical axis in Figure 4.4 shows the ratio between the simulated (static test case) SGPLL phase tracking noise SD (rad) over the theoretical phase tracking noise, and the horizontal axis shows the simulated $C/N_0$ (dB-Hz). The coloured solid lines depict the results at different PLL effective bandwidths, with the black dotted lines representing the thresholds where the simulated values exceed the theoretical ones by a factor of 2, i.e. either more than two times the theoretical values, or less than half of the theoretical values, noting that the theoretical values can be calculated using Eq. (4.1). The results show that the SGPLL breaks, i.e. simulated tracking errors due to cycle slips exceed more than twice of the theoretical values, at various $C/N_0$ levels, depending on the effective SGPLL bandwidth. The static phase-tracking noise SD exceeds the theoretical phase-tracking noise SD, above the factor-of-two threshold, at 25dB-Hz $C/N_0$ with a 5Hz bandwidth, at 28dB-Hz $C/N_0$ with a 10Hz bandwidth, at 29dB-Hz $C/N_0$ with a 15Hz bandwidth, and at 31dB-Hz $C/N_0$ with a 20Hz bandwidth. This shows that higher PLL bandwidths allow more tracking noise to enter the SGPLL. It is also worth noting that the theoretical values do not account for any observed cycle slips, even at a $C/N_0$ below 25dB-Hz with high PLL bandwidths (e.g. 20Hz). Also, for $C/N_0$ where no cycle slips are observed, the ratio of the simulated tracking noise over the theoretical tracking noise is constant, so the validation criteria set in the introduction of Section 4.4 are met.

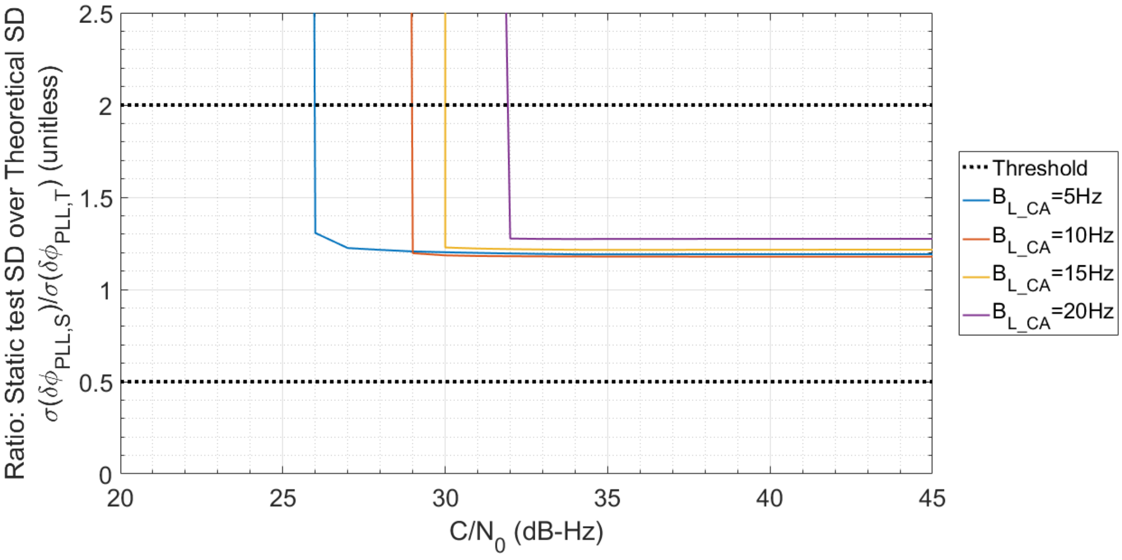

Figure 4.4. Simulated over Theoretical PLL carrier phase tracking noise SD ratio (unitless)

This section continues with the validation of the simulated FLL results.

### 4.4.2 Custom FLL validation

Figure 4.5 illustrates the ratio of the simulated FLL frequency tracking noise SD $\sigma(\delta f_{FLL,S})$ over theoretical FLL frequency tracking noise SD $\sigma(\delta f_{FLL,S})$. It is important to note at this point, that the theory predicting the FLL tracking noise, as expressed in Eq. (4.7), assumes that the tracking noise exiting the discriminator is Gaussian, although this is not the case, as the discriminator (see Section 4.3) estimates the frequency error by differencing the Is and Qs of the current and previous epochs, so the Gaussian tracking noise contained in them becomes anticorrelated. To address this issue, the theoretical tracking noise prediction $\sigma\left(\delta\dot{\rho}_{cf}\right)$ in Eq. (4.7) has been adjusted, giving a new tracking noise prediction $\sigma'\left(\delta\dot{\rho}_{cf}\right)$, for a first-order FLL, following (P. Groves, personal communication, 24 January, 2017):

$$\begin{gathered}\sigma'\left(\delta\dot{\rho}_{cf}\right)=\sigma\left(\delta\dot{\rho}_{cf}\right)\sqrt{\kappa}, \\ \kappa=4\cdot B_{L\_CF}\cdot\tau_a\end{gathered}, \tag{4.8}$$

The coloured lines in Figure 4.5 represent different effective bandwidths, as shown in the legend. The results show that the static frequency-tracking noise SD exceeds the theoretical one, above the factor-of-two threshold, at 35dB-Hz $C/N_0$ with a 1Hz bandwidth, at 29dB-Hz $C/N_0$ with a 2Hz bandwidth, noting that the theoretical tracking noise SD does not account for any observed frequency-locks and that the oscillator noise in a real receiver (not simulated in this thesis) would introduce additional false frequency locks at these low bandwidths. The results also show that the static frequency-tracking noise SD exceeds the theoretical one, above the factor-of-two threshold, at 24dB-Hz $C/N_0$ with a 5Hz bandwidth, and at 26dB-Hz $C/N_0$ with a 10Hz bandwidth, which shows that higher FLL bandwidths allow more tracking noise to enter the SGFLL. Also, for higher $C/N_0$ values, where no false frequency locks are observed, the ratio of the simulated frequency-tracking noise over the theoretical one, is contained within the factor-of-two threshold limits, and (at higher effective bandwidths) the ratio remains relatively constant, therefore it is considered that the validation criteria set in the introduction of Section 4.4 are met.

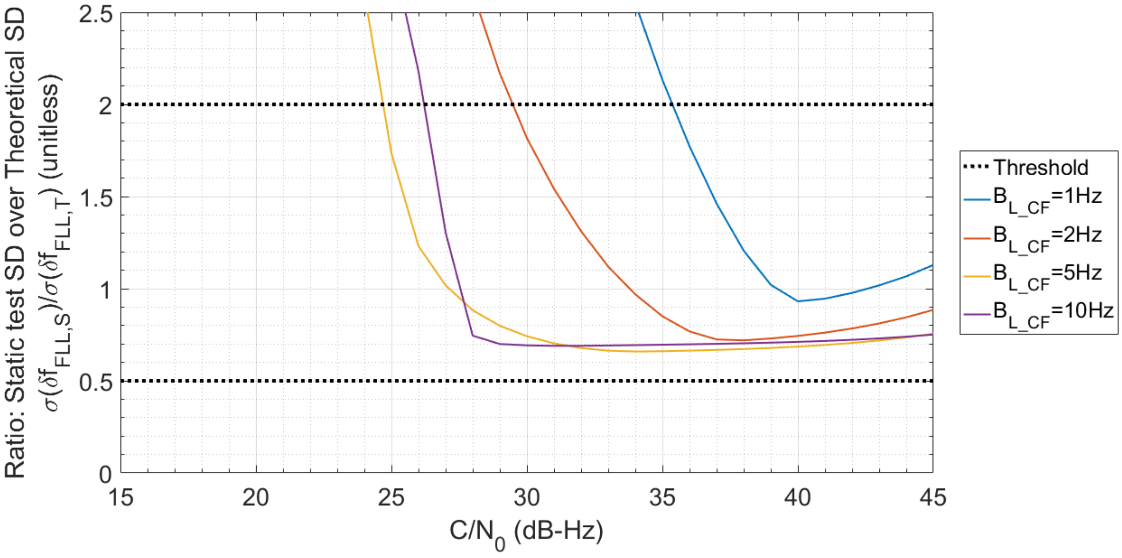

Figure 4.5. Theoretical and Simulated FLL carrier frequency tracking noise SD (Hz).

The $C/N_0$ minimum level before the factor-of-two threshold of the simulated and theoretical tracking noise SDs is exceeded, both for the SGPLL and the SGFLL are summarised in Table 4.1. The thesis continues with the discussion of the pedestrian motion model development in the next chapter.

| *SGPLL*<br>*Phase-tracking* | | *SGFLL*<br>*Frequency-tracking* | |
|---|---|---|---|
| *$B_{L\_CA}$ (Hz)* | *Minimum $C/N_0$* | *$B_{L\_CF}$ (Hz)* | *Minimum $C/N_0$* |
| 5 | 26 | 1 | 36 |
| 10 | 29 | 2 | 30 |
| 15 | 30 | 5 | 25 |
| 20 | 32 | 10 | 27 |

Table 4.1. Minimum $C/N_0$ level before exceeding the factor-of-two simulated tracking noise SDs over theoretical tracking noise threshold.

# Chapter 5. Proposed Pedestrian Motion Model

This chapter explains the development method of a custom pedestrian motion model (PMM) which can simulate the 3D motion of positioning and navigation equipment held by or attached on a pedestrian. The discussion in this chapter is based on the literature review of human motion analysis, capture and modelling introduced in Sections 2.1 and 2.3. This chapter is divided into four sections. The first Section 5.1, describes the candidate pedestrian motion methods in order to justify the currently selected one. The second, Section 5.2, details the implementation of the pedestrian routing model (PRM), and the third, Section 5.3, explains the development of a human biomechanical model (HBM), which is used to simulate segments motion (gestures). The final Section 5.4, explains the synthesis of the HBM and the PRM into the pedestrian motion model (PMM), i.e. the HBM following the PRM output path. The output of the PMM is synthetic 3D motion data of the positioning and navigation equipment (PNE), when attached on or held by a pedestrian. To assess the impact of the synthetic motion on GNSS receivers, the synthetic data output is fed into the simulated GNSS carrier-tracking loops (SGCTLs, as described in Sections 4.2 and 4.3), with the results of this analysis presented in Chapter 6.

The work presented in Chapter 5 is important for the purposes of the thesis, as a custom-made PMM can be manipulated (by means of rotating constituent body segments) to represent human motion, without the need to conduct new MoCap experiments when a new motion profile is required. Although that work was a major time commitment for the thesis, it also facilitated the future productisation of the thesis' outcomes. This is firstly in terms of licensing, i.e. avoiding issues which may arise when using third-party functionality, and secondly, future support and/or tailored enhancements of the PMM, since the PMM code is fully available and understood.

Furthermore, Chapter 5 explains how a tailored human biomechanical model can be controlled using MoCap data in order to produce realistic human motion. This was considered appropriate for the thesis as although modelling human motion using MoCap techniques produces realistic, human-like results, the process can be involved in terms of conducting the experiments end processing the results, e.g. by weighting and smoothing measurements. In addition, combining MoCap segments to synthesise a complete trajectory may involve additional smoothing in order to avoid discontinuities (especially when higher-order quantities, e.g. velocity or acceleration are of interest). Other methods involve imposing physical constrains on human movement, and generate motion without requiring prior MoCap and equipment calibration. However, the result may be unrealistic, due to the high dimensionality of human movement, i.e. there are several ways for human body segments to move and rotate in order to reach from a specific posture to another one. Therefore, these methods have not been considered suitable for this thesis.

## 5.1 Pedestrian Motion Modelling Approaches

Human motion is subject to great variability depending on individual characteristics, e.g. gender, height, health status. Therefore, appropriate modelling of human motion is critical for testing its effects on GNSS receivers. This study proposes a method of modelling human motion, using a human biomechanical model (HBM), which will be driven by a pedestrian routing model (PRM) in order to simulate the 3D trajectory of PNT equipment, held by or attached to a pedestrian. The output of the proposed PMM is validated in comparison with human motion capture (MoCap) data to ensure that both have the same effect on GNSS carrier tracking loops. Three methods have been identified as candidate approaches in order to generate a pedestrian movement model for this thesis.

The first is the *trials-based* approach, which involves capturing of human motion data (MoCap), discussed in Section 2.3.1, and then the identification of pedestrian movement elementary components within the captured data, e.g. individual steps, or specific gestures. These basic components will then be combined to create a representative sequence for the pedestrian movement in question. Due to the repeatable manner (i.e. periodic events) of human gait, this option may be practical in simple cases of human movement, e.g. series of forward steps without gestures. However, this method involves MoCap in order to record specific human movement variations, e.g. step length (discussed in Section 2.1) within strides of the same subject, or between different subjects, although limited scalability in terms of in terms of varying the motion speed or the height of the subject could be applied. Also, since the combined sequence of movement signatures has to follow tightly the output path from the pedestrian routing model, the sequence will have to accommodate turns, (and gestures as well), so it is essential to ensure that the transitions between different movement segments are smooth, in order to avoid artefacts such as position/velocity/orientation discontinuities which may degrade the performance of the SGCTLs by introducing artificial cycle-slips. Transitioning smoothly between subsequent postures and gestures can be a tedious task involving smoothing algorithms development and manual editing of human body coordinates. This approach would be proven challenging within the timescales of this thesis, especially when alternative approaches exist. Another issue with employing this method is that all different pedestrian movement scenarios (involving different sensors locations) would need to be captured separately. In other words, trials-based motion generation can only apply only within the scope of the specific captured motion (MoCap) scenarios, with very limited options to expand this scope.

The second is the *analytical* approach, e.g. [100], [101], which encompasses an analytical model of the human body describing movement of human body segments. Any pedestrian size, gait (walking or running) speed, gender motion variations (discussed in Section 2.1) and gestures can also be modelled, e.g. using lever arms starting from a reference point, whose motion is also defined. The humanoid motion is well-defined and repeatable, noting that an analytical model of pedestrian movement may not be entirely deterministic, but could encompass a stochastic element which would model pedestrian-induced noise, e.g. due to the muscles physiological tremor [132][133]. The advantage of this method is that the scope of the analytical model can be extended in order to represent pedestrians with different anatomical characteristics (e.g. gender, step length, cadence). It is worth noting that although the model described in [100] was available to be reused for research purposes, the licence agreement prohibited any future commercial use, which is a prospective outcome from this thesis, as mentioned in Section 1.1.

Finally, the third is the *hybrid* approach, which can employ a combination of the trials-based and analytical approaches e.g. by modelling a specific gesture analytically and over-imposing this onto the trials-based motion. This method provides some additional scalability for particular cases of motion, compared to the limited scalability of the trials-based approach, e.g. in terms of adjusting the overlaid gesture type and speed. Overall though, the hybrid approach is bound to the same scope limitations as the trials-based approach. Also, the hybrid approach could add some stochastic element on the trials-based motion, e.g. white noise with a variable mean depending on the exhibited acceleration.

For the purposes of thesis, the PMM should allow the overlay of gestures, in order to extend the range of the motion profiles that can be simulated, noting that in real life, gestures may occur at any time point over the gait cycle. Therefore, the trials-based and hybrid approaches are practically limiting the scope of this requirement, as any new gesture occurring at a different time instant (during the gait cycle) would require a new motion capture (MoCap) experiment. Furthermore, the analytical approach enables the PMM scope to expand in future in order to follow any given trajectory generated by the PRM. This option would not be available in the case of the trials-based and hyrid approaches, as a new path might mean an experimental re-capture of pedestrian motion. Considering these points, the analytical approach has been selected as the most suitable approach for the purposes of this thesis.

## 5.2 Pedestrian Routing Model

Simulation of human motion requires a PRM, combined with an HBM. This section describes the implementation of the pedestrian routing model (PRM) which, along with the human biomechanical model (HBM) described in Section 5.3.3, comprise the integrated pedestrian motion model (PMM) described in Section 5.4. The (integrated) PMM outputs synthetic human motion data, representing the 3D trajectory of positioning and navigation equipment (PNE) while held by or attached on the human body. The simulated GNSS carrier tracking loops (SGCTLs, described in Section 4.1) provide the validation test platform for the PMM, as both synthetic and real captured human motion (MoCap) data should exert similar behaviour on the SGCTLs, in terms of phase cycle slips and frequency false locks, for the simulated carrier phase and frequency tracking loops, respectively.

The current PRM implements the standard Dijkstra's pathfinding algorithm [120], based on the code from [134], and a standard smoothing function, explained later in this section, in order to make the output trajectory more realistic. The operation of Dijkstra's algorithm was introduced in Section 2.3.2. The current implementation runs the algorithm between two points, a pre-defined start point and a user-defined finish point, both located on a pre-defined grid, as illustrated in Figure 5.1. The underlying map (grid) is a binary image with white pixels which the moving agent (representing the pedestrian moving on the grid) is permitted to move onto and black squares which the agent cannot move onto. The current algorithm implementation could also account for intermediate "grey" pixels, i.e. representing a cost function, but this is out of the thesis' scope. The output path of the Dijkstra algorithm, e.g. as illustrated in Figure 5.1, cannot be considered realistic enough as firstly, the motion direction can only change in discrete angles of $\pi/4$ rad, i.e. a maximum of 8 permitted directions around the current pixel, and secondly, the path nodes are on the centre of the pixels, i.e. the size of the pixel controls how smooth the output path will be, noting that the more pixels (for the same size of grid, i.e. a denser grid) the higher the output path spatial resolution, but also, the higher processing load in order to run Dijkstra's algorithm. It follows that there are two step sizes on the grid, i.e. unity size when the agent moves between adjacent right, left, up and down cells, and $\sqrt{2}$ size when the agent moves diagonally.

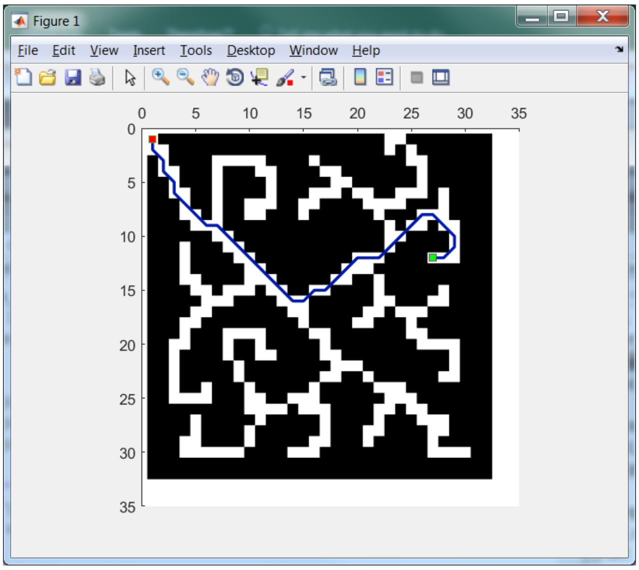

Figure 5.1. Dijkstra's algorithm Matlab implementation

The output of Dijkstra's algorithm can be made more realistic by equalising the distance between epochs and smooth out the original Dijkstra's algorithms output trajectory, which is based on a grid with a specific spacing (pixel size). Also, the PRM path is processed by applying a smoothing function averaging the positions of the data-points, using the *central difference* standard method, explained in [135]. The only two points not smoothed by applying this method are the start/end ones. The method averages two position vectors with an intermediate position vector between them:

$$\mathbf{r}_i = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2}, \qquad 1 < i < m-1, \tag{5.1}$$

where vector $\mathbf{r}_i$ is the position at epoch *i*, and *m* is the total number of epochs (data-points) of the output path, noting that the first epoch index is zero (i=0). The new intermediate point is located half-way w.r.t. the previous and next points. Applying the central difference method more than once, results to a smoother position for the intermediate vector. It follows that applying the method a large number of times on a curved path, it will result to a straight line connecting the start/end points of that path. Using this method, Eq. (5.1), on time-stamped position vectors, such as the output of the PRM, it is possible to maintain the same time validity of the intermediate position vectors. For example, assume that the old intermediate point is valid at time *t*. The new intermediate position vector is also valid at time *t*, as it is a function of two (equally-weighted) position vectors, previous and next, valid at *t-τ* and *t+τ*, respectively, where *τ* the time interval between position vectors and it is fixed for the duration of the output path.

Also, it is worth noting that in order to simulate different walking speeds, this time interval can be increased or decreased accordingly. The underlying assumption is that, for non-straight paths, moderate variations of the walking speed, e.g. up to 10%-15%, would produce the same positions of output path data-points, in other words, the result of the changing speed on the gait characteristics, e.g. step length or human body tilt while turning, is negligible. These small variations in walking speed can be simulated by altering proportionally the time interval between the output path data-points' positions.

The smoothed output of the PRM is the path which the HBM needs to follow in order to simulate the complete PNE 3D trajectory, encompassing human walking and any gestures. The current implementation integrates the PRM and HBM at walking speed in order to be comparable with the capture motion (MoCap) scenarios described in Chapter 3. The PNE dynamics are then calculated from this 3D position trajectory and projected along the lines-of-sight between the sensor and three simulated satellites (positioned in forward, left and overhead directions) to produce the synthetic motion data. The synthetic motion is then fed into the simulated GNSS tracking loops (SGCTLs), described in Chapter 4, in order to study the impact on the SGCTLs' performance and compare this impact with the real-world MoCap data for validation purposes. The input routing trajectories used for PMM validation (see Section 6.3) were simpler than the one depicted in Figure 5.1. (30m straight line and a U-turn). However, this PRM is an essential part for the thesis as this functionality can be used in the future to automatically calculate the route given a map, e.g. like the one depicted in Figure 5.1, which is faster than defining the same path manually as pairs of 2D coordinates.

This chapter continues with the description of the HBM in Section 5.3 and concludes with how the HBM and PRM were integrated to form the PMM in Section 5.4.

## 5.3 Human Biomechanical Model

This section describes how the design, and implementation of the HBM. The HBM moves along the PRM output path (forming the integrated PMM, described in Section 5.4), in order to provide a synthetic positioning and navigation equipment (PNE) 3D trajectory which reflects the motion dynamics exhibited by the PNE in the presence of underlying human motion. The impact of this synthetic trajectory on the SGCTLs (which were detailed in Section 4.1), can then be assessed and compared to the real-world human MoCap for PMM validation in Chapter 6.

### 5.3.1 Human biomechanical model definition

The human body can be modelled as a system of segments and connecting joints/nodes, as discussed in Section 2.3.1 of the literature review. The HBM proposed in this thesis, is shown in Figure 5.2 and is referenced to the pedestrian *p* frame, illustrated in the same figure. The design of the HBM is based on an anthropometric study [135] in order to represent the average proportions of the human body segments. The segments and nodes of the current HBM implementation are summarised in Table 5.1, with the coordinates in the last column being normalised in body height, in order to be able to represent bodies of different heights. For the purposes of this thesis the segments are all considered rigid (inelastic) bodies, whose length cannot change. The term “node” encompasses points which could be considered as “joints” of the human body from a physio-anatomical perspective, and other HBM points which are used to connect the human body segments with the PRM (node no. 1, “Op”), the modelled position of the PNE on the human body (node no. 23, “Sr”), and the calculated human body centre of mass (node no. 24, “CoM”). Each HBM segment can be defined as lying in-between a proximal and a distal end. The proximal end is closer to the human body CoM, e.g. for the head and neck segment the proximal end is the head and neck joint (no. 3, “Hn”), while for the right upper arm segment, the proximal end is the right shoulder joint (no. 5, “Rsh”). The distal end is the segment end that lies further away from the CoM.

The HBM moves by segments’ rotations and translations. In terms of rotations, it is useful to define segment ends in terms of parent frames, denoted with *a*, and child frames, denoted with *β*, which are rotated by the parent frames. As parent frames *a* can be considered all the HBM proximal end points, plus the *p* frame origin (node no. 1 “Op” in Figure 5.2) which as part of the HBM can (only) be used to rotate its child frame (node no. 2, “Hc”). Node “Op” is where the HBM midline (illustrated with a dotted red line in Figure 5.2) intersects with the horizontal level, noting that the midline is the vertical line between the CoM and the horizontal level when the HBM is at the fundamental position, described in Section 2.1.1.

The set of child frames encompasses the distal end points, plus the sensor/PNE frame (node no. 23, "Sr") which can only be rotated by its parent frame as appropriate for the user-defined sensor location on the HBM. The only HBM node which cannot be considered as a parent or child frame, is node no. 24 "CoM" (illustrated as a solid black point in Figure 5.2). This node represents the calculated position of the HBM's CoM at an epoch *k*, following the method given in Appendix A.

The attitude of a segment *j* at epoch *k* can be updated by rotating the distal end w.r.t. its proximal node, i.e. from parent frame *α* to child frame *β*, in order to represent human motion. The blue arrows in Figure 5.2 show the direction of the rotation from a parent frame *α* to a child frame *β*. All frames in-between two segments, can be interchangeably parent or child frames, except for:

- The *p* frame origin: node no. 1 "Op". This can only be a parent frame (illustrated as a solid blue point in Figure 5.2). The translation of this point to local tangent plane (LTP) coordinates enables the PMM to also have LTP coordinates.
- The distal ends of: the head (no. 4, "Ht"), right hand (no. 8, "Rfi"), left hand (no. 17, "Lfi"), right foot (no. 13, "Rfo"), left foot (no. 22, "Lfo"), and the sensor/PNE frame (node no. 23, "Sr"). These can only be child frames (illustrated as solid orange points in Figure 5.2). The sensor/PNE frame (node no. 23, "Sr") can be a child frame to any parent frame of the HBM, representing different sensor locations on the human body that the PNE can be attached to.

Table 5.2 defines the frame origins contained in the HBM from a functional anatomical perspective, noting that the HBM joints are assumed to rotate about their nominal centre of rotation, although in reality the joint centre is translated during the rotation, as explained in Section 2.1.2.

Table 5.1 summarises the parent and child frames of the HBM. The child-joint frame coordinates (normalised in height) are referenced and resolved in parent frame axes, and not the *p* frame, e.g. the coordinates of head top "Ht" node in Table 5.1 are (0, 0, 0.182) w.r.t to its parent frame (head and neck "Hn" node), while the head top normalised coordinates referenced and resolved in *p* frame axes are (0, 0, 1). It follows that preceding rotations of a parent frame, propagate to subsequent rotations of any child frames.

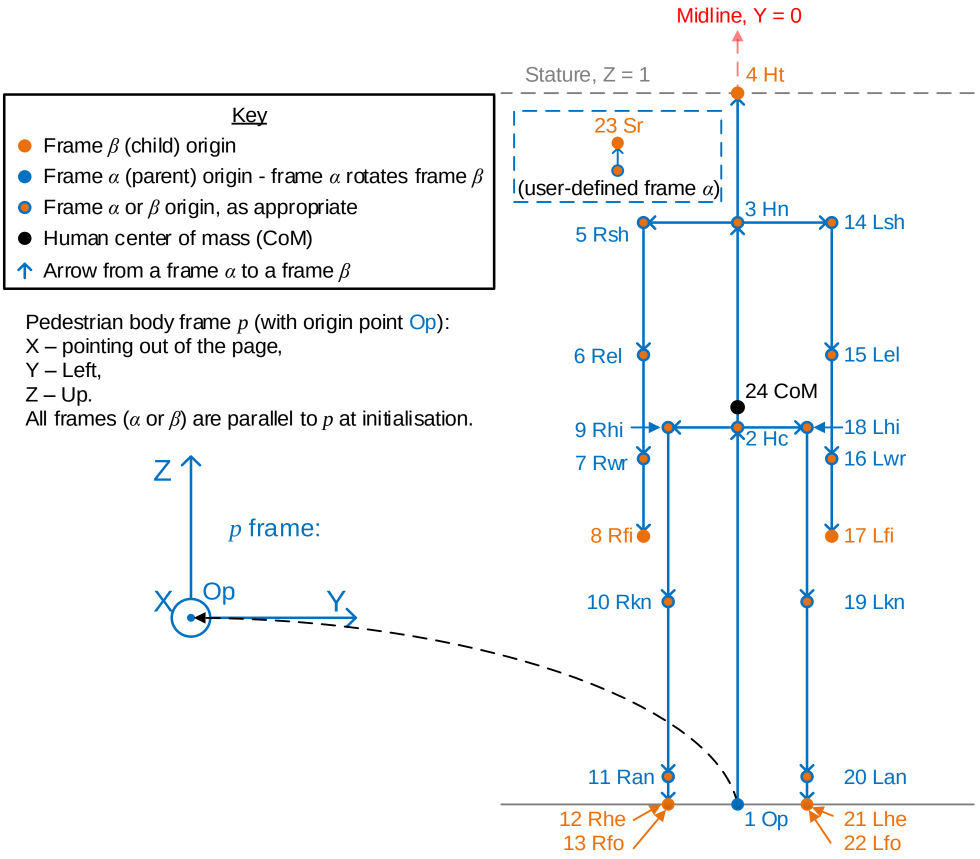
Midline, Y = 0
Stature, Z = 1
4 Ht
Key
Frame β (child) origin
Frame α (parent) origin - frame α rotates frame β
Frame α or β origin, as appropriate
Human center of mass (CoM)
Arrow from a frame α to a frame β
23 Sr
(user-defined frame α)
3 Hn
5 Rsh
14 Lsh
Pedestrian body frame p (with origin point Op):
X – pointing out of the page,
Y – Left,
Z – Up.
All frames (α or β) are parallel to p at initialisation.
6 Rel
15 Lel
24 CoM
9 Rhi
2 Hc
18 Lhi
7 Rwr
16 Lwr
Z
p frame:
8 Rfi
17 Lfi
X
Op
Y
10 Rkn
19 Lkn
11 Ran
20 Lan
12 Rhe
13 Rfo
1 Op
21 Lhe
22 Lfo

Figure 5.2. Proposed Human Biomechanical Model (front-facing), based on [135].

The joints and sensor frame coordinates in Table 5.1, denoted with **d**, are normalised in body height and referenced to their respective parent frame, e.g. the "Ht" (head top) point has (0, 0, 0.182) coordinates w.r.t. its parent frame "Hn" (head & neck joint), but if it was referenced to "Op" (HBM frame) the coordinates would be (0, 0, 1) as it marks the body height along HBM's Z-axis. To obtain these normalised coordinates in meters, denoted with **r**, a scale factor $s$ corresponding to pedestrian stature (height) must be applied:

$$\mathbf{r}_{\alpha\beta}^{\gamma} = s \cdot \mathbf{d}_{\alpha\beta}^{\gamma} \quad . \tag{5.2}$$

Also, the sensor "Sr" frame can be modelled w.r.t. any parent frame on the HBM. This provides flexibility in order to accommodate different pedestrian applications, e.g. requiring sensor placement on the shoulder or the head (via a helmet). An applicable gesture is meaningful when it affects the trajectory of the sensor frame, either directly, by being applied on the parent segment of the sensor frame, or indirectly, by rotating a preceding parent frame, which will result in moving the sensor. Therefore, the list of candidate relevant gestures applied by the HBM depends on the candidate positions of the sensor on the HBM.

| *Node index* | *Frame β origin* | *Frame β description* | *Segment (between frames β and α) description* | *Frame α origin* | *Frame β's 3D normalised coordinates, referenced and resolved in parent frame α axes, at first epoch (k=1)* $\mathbf{d}_{\alpha\beta}^{\alpha}\big|_{k=1}$ |
|---|---|---|---|---|---|
| 1 | Op | Pedestrian frame *p* | n/a | Origin of *l* (LTP), not shown in Figure 5.2. | n/a (used to resolve PRM in LTP axes) |
| 2 | Hc | Hip centre | n/a | Op (*p* frame) | (0, 0, 0.530) |
| 3 | Hn | Head & neck | Torso | Hc | (0, 0, 0.288) |
| 4 | Ht | Head top | Head & neck | Hn | (0, 0, 0.182) |
| 5 | Rsh | Right shoulder | Right shoulder | Hn | (0, -0.1295, 0) |
| 6 | Rel | Right elbow | Right upper arm | Rsh | (0, 0, -0.186) |
| 7 | Rwr | Right wrist | Right forearm | Rel | (0, 0, -0.146) |
| 8 | Rfi | Right fingertip | Right hand | Rwr | (0, 0, -0.108) |
| 9 | Rhi | Right hip | Right part of pelvis | Hc | (0, -0.0955, 0) |
| 10 | Rkn | Right knee | Right thigh | Rhi | (0, 0, -0.245) |
| 11 | Ran | Right ankle | Right shank | Rkn | (0, 0, -0.246) |
| 12 | Rhe | Right heel | Right heel | Ran | (0, 0, -0.039) |
| 13 | Rfo | Right foot tip | Right foot | Ran | (0.152, 0, -0.039) |
| 14 | Lsh | Left shoulder | Left shoulder | Hn | (0, 0.1295, 0) |
| 15 | Lel | Left elbow | Left upper arm | Lsh | (0, 0, -0.186) |
| 16 | Lwr | Left wrist | Left forearm | Lel | (0, 0, -0.146) |
| 17 | Lfi | Left fingertip | Left hand | Lwr | (0, 0, -0.108) |
| 18 | Lhi | Left hip | Left part of pelvis | Hc | (0, 0.0955, 0) |
| 19 | Lkn | Left knee | Left thigh | Lhi | (0, 0, -0.245) |
| 20 | Lan | Left ankle | Left shank | Lkn | (0, 0, -0.246) |
| 21 | Lhe | Left heel | Left heel | Lan | (0, 0, -0.039) |
| 22 | Lfo | Left foot tip | Left foot | Lan | (0.152, 0, -0.039) |
| 23 | Sr | Sensor location | n/a | User-defined as any parent frame *α*, from no. 3 to no. 22 above. | User-defined as relative 3D position between "Sr" and its parent frame *α*. |
| 24 | CoM | Centre of mass | n/a | Op (*p* frame) | (0, 0, 0.5584) Calculated as per Appendix A |

Table 5.1. HBM segments and joints with initial normalised coordinates in height, based on [135]. Frame *α* encompasses segments' proximal ends, and frame *β* the distal ends

| *Frame name (abbreviation)* | *Definition (proximal/ distal end)* |
|---|---|
| Head Top (Ht) | Distal end of skull |
| Head & Neck (Hn) | T1 (1st thoracic vertebra)/ C7 (7th cervical vertebra) |
| Shoulder (Rsh, Lsh) | Glenoid cavity of scapula/ head of humerus. |
| Elbow (Rel, Lel) | Distal end of humerus/ proximal ends of ulna and radius. |
| Wrist (Rwr, Lwr) | Proximal end of wrist (lunate, pisiform, scaphoid, triquetral bones)/ distal ends of ulna and radius. |
| Finger-tip (Rfi, Lfi) | Distal phalanx of middle-finger |
| Hip (Rhi, Lhi) | Acetabulum/ Head of femur |
| Hip centre (Hc) | Mid-point between hip joints |
| Knee (Rkn, Lkn) | Distal end of femur/ Proximal end of tibia |
| Ankle (Ran, Lan) | Distal end of tibia |
| Heel (Rhe, Lhe) | Calcaneus bone end (in the opposite direction from the foot-tip) |
| Foot-tip (Rfo, Lfo) | Distal phalanx of big toe |

Table 5.2. HBM frames' anatomical definitions

A convenient way to express the rotation between two frames is by using a coordinate transformation matrix (CTM), which is also called a "rotation matrix" or "direction cosine matrix". The coordinate transformation matrix representing the rotation from a frame *β* to a frame *α* is denoted with $\mathbf{C}_{\beta}^{\alpha}$, i.e. "from" frame in subscript and "to" frame in superscript. The transpose matrix, which represents the rotation from frame *α* to frame *β,* is represented by:

$$\mathbf{C}_{\beta}^{\alpha} = \left(\mathbf{C}_{\alpha}^{\beta}\right)^{\mathrm{T}} \quad , \tag{5.3}$$

noting that $\mathbf{C}_{\alpha}^{\alpha} = \mathbf{C}_{\beta}^{\beta} = \mathbf{I}_3$, as the attitude of a frame w.r.t. itself is the 3x3 identity matrix. Also, a CTM and its transpose matrix are orthogonal:

$$\mathbf{C}_{\beta}^{\alpha}\mathbf{C}_{\alpha}^{\beta} = \mathbf{I}_3 \quad . \tag{5.4}$$

The CTM chain rule shows how successive rotations from a frame *β* to a frame *γ*, through an intermediate frame *a*, can be performed:

$$\mathbf{C}_{\beta}^{\gamma} = \mathbf{C}_{\alpha}^{\gamma}\mathbf{C}_{\beta}^{\alpha} \quad , \tag{5.5}$$

noting that the fact that physical rotations are non-commutative is mathematically represented by the CTM multiplication non-commutativity. To transform the resolving frame of a vector $\mathbf{x}$ from a frame *γ* to a frame *δ*:

$$\mathbf{x}_{\beta\alpha}^{\delta} = \mathbf{C}_{\gamma}^{\delta}\mathbf{x}_{\beta\alpha}^{\gamma} \quad . \tag{5.6}$$

### 5.3.2 **Gestures simulation**

For pedestrian modelling purposes in this thesis, a *human gesture* is defined as the movement that a human body segment (on which the sensor is located) performs in addition to the nominal human body movement during the gait cycle. Therefore, additional movements that may be performed during the gait cycle are considered human gestures which can be overlaid on the pedestrian movement model. Human gestures examples for the purposes of this thesis can be considered as those movements which can alter the position of the positioning and navigation equipment (PNE) in addition to the underlying gait movement, e.g. walking. Examples of gestures falling into this definition are answering a smartphone (with PNE), using a smartphone to text/email, or putting it in a pocket or bag.

To control the HBM, gestures are applied through a series of segment rotations. Namely, a gesture comprises of rotations of individual segments. Each individual rotation, rotates a frame *β* (called "child-frame" herein) w.r.t. a frame *a* (called "parent-frame" herein), noting that a specific joint, e.g. the right shoulder can be defined as frame *β* when rotated by the head and neck joint (which in that case is frame *a*), but it is frame *a* when it rotates the right elbow joint (frame *β*). It follows that frames *a* and *β* can be considered as elements of a set A which comprises the following pairs of rotating frames:

$$(\alpha,\beta)\in \mathrm{A} = \left\{ \begin{array}{l} (Hc,Hn),(Hn,Ht), \\ (Hn,Rsh),(Rsh,Rel),(Rel,Rwr),(Rwr,Rfi), \\ (Hc,Rhi),(Rhi,Rkn),(Rkn,Ran),(Ran,Rhe),(Ran,Rfo), \\ (Hn,Lsh),(Lsh,Lel),(Lel,Lwr),(Lwr,Lfi), \\ (Hc,Lhi),(Lhi,Lkn),(Lkn,Lan),(Lan,Lhe),(Lan,Lfo). \end{array} \right\} \quad , \tag{5.7}$$

with their abbreviations, listed in Table 5.1. The pairs of frames in set A correspond to specific human body segments, also listed in Table 5.1, noting that the pairs *(l,p)* and *(p,Hc)*, have been added for completeness purposes, although they do not correspond to a human body segment; however, the rotation of the *(l,p)* lever arm orientates the pedestrian frame (HBM) w.r.t the local tangent plane, and the rotation of the hip centre "Hc" w.r.t. the pedestrian frame is the first rotation among a chain of applicable rotations on the HBM defining its posture at any epoch *k*.

The 3D attitude update of frame *β* w.r.t. frame *a*, with $(\alpha,\beta)\in \mathrm{A}$, i.e. from the previous attitude *β*- to the updated *β*+, or in other words from *β*(t-τ) to *β*(t) after a time interval *τ*, occurs around a rotational axis in 3D. The attitude increment from *β*- to *β*+ can be expressed as a coordinate transformation matrix (CTM), where the CTM and attitude increment relationship is defined for a specific segment *j* at a specific epoch *k*, e.g. $\mathbf{C}_{\beta-}^{\beta+}\Big|_{k}^{j}$ ; however, for readability purposes the segment and epoch upper and lower indexes have been omitted in the following analysis. Also, although in the case of the HBM the Rodrigues' formula is applied for a rotation from a frame *β*- to *β*+, it can be applied to any frame rotation in 3D, without loss of generality, as any 3D rotation can be represented as a rotation about an appropriate axis.

As an example, assume that gesture *i* "answer the phone" comprises rotations of three segments *j*=1,2,3: 1. upper arm rotation, where the shoulder (parent-frame *a*) rotates the elbow (child-frame *β*), 2. forearm rotation, where the elbow (now parent-frame *a*) rotates the wrist (child-frame *β*), and 3. hand rotation, where the wrist (now parent-frame *a*) rotates the fingertip (child-frame *β*), noting that the segment between the wrist and the fingertip models the hand as a solid segment, i.e. the fingers' rotation has not been currently modelled. Also, only one gesture can be applied on a specific segment at a given epoch in order to avoid conflicting applied gestures on that segment, noting that this is in addition to the underlying walking motion (encompassing arm swinging). For an epoch *k*, the attitude of a frame *β* w.r.t. frame *a* can be expressed using the notation $\mathbf{C}_{\beta}^{\alpha}\Big|_{k}^{j}$ , while a vector will be written in the general form $\mathbf{x}_{\alpha\beta}^{\gamma}\Big|_{k}^{j}$ . Following these notation conventions the attitude, denoted with a plus sign at a given epoch *k* and with a minus sign at the previous epoch *k-1*, can be updated using Rodrigues' formula [3]:

$$\mathbf{C}_{\beta+}^{\alpha} = \mathbf{C}_{\beta-}^{\alpha}\mathbf{C}_{\beta+}^{\beta-} \quad , \tag{5.8}$$

$$\mathbf{C}_{\beta(t)}^{\beta(t-\tau)} = \mathbf{C}_{\beta+}^{\beta-} = \mathbf{I}_3 + \frac{\sin\left|\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right|}{\left|\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right|}\left[\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\wedge\right] + \frac{1-\cos\left|\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right|}{\left|\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right|^2}\left[\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\wedge\right]^2 \quad , \tag{5.9}$$

where $\mathbf{I}_3$ is the 3x3 identity matrix, $\left|\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right|$ is the magnitude of the attitude increment vector, which is defined as the integral of the angular rate vector, noting that it accounts for rotations non-commutativity:

$$\boldsymbol{\alpha}_{\alpha\beta}^{\gamma} = \int_{t-\tau}^{t} \boldsymbol{\omega}_{\alpha\beta}^{\gamma} dt \quad , \tag{5.10}$$

and $\left[\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\wedge\right]$ is the corresponding skew-symmetric matrix:

$$\boldsymbol{\alpha}_{\alpha\beta}^{\beta} = \begin{pmatrix} \alpha_{\alpha\beta,x}^{\beta} \\ \alpha_{\alpha\beta,y}^{\beta} \\ \alpha_{\alpha\beta,z}^{\beta} \end{pmatrix} \quad , \quad \left[\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\wedge\right] = \begin{bmatrix} 0 & -\alpha_{\alpha\beta,z}^{\beta} & \alpha_{\alpha\beta,y}^{\beta} \\ \alpha_{\alpha\beta,z}^{\beta-} & 0 & -\alpha_{\alpha\beta,x}^{\beta} \\ -\alpha_{\alpha\beta,y}^{\beta-} & \alpha_{\alpha\beta,x}^{\beta} & 0 \end{bmatrix} \quad . \tag{5.11}$$

To avoid division by zero, or numerical instability issues due to dividing with a small number, in Eq. (5.9), or when $\left|\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right|$ is small, e.g. lower than 10$^{-8}$ rad, the fourth-order approximation can be used [3]:

$$\mathbf{C}_{\beta+}^{\beta-} \approx \mathbf{I}_3 + \left(1 - \frac{\left|\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right|^2}{6}\right)\left[\boldsymbol{\alpha}_{\alpha\beta}^{\beta} \wedge\right] + \left(\frac{1}{2} - \frac{\left|\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right|^2}{24}\right)\left[\boldsymbol{\alpha}_{\alpha\beta}^{\beta}\right]^2 \quad . \tag{5.12}$$

A gesture is defined as the final attitude position of each parent frame that exhibits rotation as part of this gesture, e.g. for an "answer the phone gesture" the relevant parent frames are the shoulder, elbow and wrist parent frames. The final attitude position is defined to be reached at epoch *k+n*, where *k* is the previous epoch, so in order for parent frame *α* to reach the desired final attitude position w.r.t. frame *p*, using Eq. (5.8), there are *n-1* intermediate rotations involved:

$$\mathbf{C}_{\alpha(n)}^{p} = \mathbf{C}_{\alpha(1)}^{p} \prod_{i=2}^{n} \mathbf{C}_{\alpha(i)}^{\alpha(i-1)} \quad , \tag{5.13}$$

where $\mathbf{C}_{\alpha(i)}^{\alpha(i-1)}$ can be calculated using Eq. (5.9) or Eq. (5.12), as appropriate. In terms of simulation time-into-run, the gesture applies at time $t_{start} = (k-1)\tau$, where *τ* is the uniform time interval between subsequent epochs, and $t_{end} = t_{start} + (n-1)\tau = (k+n-2)\tau$. The HBM allows more than one gestures to be applied at the same epoch(s), provided that they encompass different parent frames each), so a parent frame's rotation can only be controlled by one gesture at any epoch. The advantage of this method is that a gesture can be applied from any starting epoch *k*. Another advantage of this method is that if the number of intermediate epochs between $t_{start}$ and $t_{end}$ changes, noting that these time instants remain the same, the gesture can still be applied in the same way, with the only difference being the intermediate number of rotations, which is calculated by Eq. (5.13). The time interval *τ* may change e.g. in case that the underlying data are interpolated with a subsequent decrease of *τ* and increase of *n*, or smoothed with a subsequent increase of *τ* and decrease of *n*.

The parent-frames coordinates are referenced and resolved along their respective parent frame *a* axes, are assumed to remain constant (forming a lever arm) for all epochs:

$$\left.\mathbf{r}_{\alpha\beta}^{\alpha}\right|_{k} = \left.\mathbf{r}_{\alpha\beta}^{\alpha}\right|_{1} \quad , \quad k = 1 \ldots m, \tag{5.14}$$

where *m* is the total number of epochs.

It is important to note that the HBM's child and parent-frames' attitude positions are defined as parallel to the pedestrian frame (w.r.t. the local tangent plane) at the first epoch, when the HBM is defined to be in fundamental position (explained in Section 2.1.1 and illustrated in Figure 5.2):

$$\mathbf{C}_p^l\big|_1 = \mathbf{C}_\beta^\alpha\big|_1 = \mathbf{I}_3 \quad , \quad (\alpha,\beta)\in \mathrm{A} \quad . \tag{5.15}$$

It follows that if $\mathbf{C}_\beta^\alpha\big|_k = \mathbf{I}_3$ at any epoch *k* (during movement), then the HBM is in fundamental position at epoch *k*, as the HBM posture depends on the applicable chain of rotations of its nodes for any epoch *k*.

From Eqs. (5.6) and (5.9), or (5.12) as appropriate, the rotation of a child-frame w.r.t. its parent-frame can be applied:

$$\mathbf{r}_{\beta\alpha}^{\delta} = \mathbf{C}_\gamma^\delta \mathbf{r}_{\beta\alpha}^{\gamma} \quad , \tag{5.16}$$

The updated coordinates of a child-frame *β* w.r.t., and resolved in, local tangent plane *l* axes at the end of epoch *k* will be:

$$\mathbf{r}_{l\beta}^{l}(+) = \mathbf{r}_{l\alpha}^{l}(+) + \mathbf{C}_\alpha^l(+)\mathbf{r}_{\alpha\beta}^{\alpha}(+) \quad , \quad (\alpha,\beta)\in \mathrm{A}-\{(l,p)\} \quad , \tag{5.17}$$

where

$$\mathbf{C}_\alpha^l = \mathbf{C}_p^l \cdot \mathbf{C}_{Hp}^p \cdot \mathbf{C}_{\alpha(n-1)}^{Hp} \cdot \ldots \cdot \mathbf{C}_{\alpha(2)}^{\alpha(3)} \cdot \mathbf{C}_{\alpha(1)}^{\alpha(2)} \quad , \tag{5.18}$$

is the chain of rotations from the current *a* frame to the *l* frame, with *n* intermediate frame rotations, i.e. from parent frame *α*(1), of the child-frame β whose updated coordinates $\mathbf{r}_{l\beta}^{l}(+)$ are sought, up to (and including) parent frame *Hp* which the last frame in the chain of rotations before the pedestrian frame *p*, as illustrated in Figure 5.2. From Eqs. (5.15), (5.16), (5.17) and (5.18) the coordinates of any frame *β* can be calculated, referenced and resolved in LTP axes. The calculation of (linear) velocity, can be given by [3]:

$$\mathbf{v}_{l\beta}^{l} = \mathbf{v}_{l\alpha}^{l} + \mathbf{v}_{\alpha\beta}^{l} + \dot{\mathbf{C}}_\alpha^l \mathbf{r}_{\alpha\beta}^{\alpha} = \mathbf{v}_{l\alpha}^{l} + \mathbf{v}_{\alpha\beta}^{l} + \mathbf{C}_\alpha^l \mathbf{\Omega}_{l\alpha}^{\alpha} \mathbf{r}_{\alpha\beta}^{\alpha} = \mathbf{v}_{l\alpha}^{l} + \mathbf{C}_\alpha^l \mathbf{\Omega}_{l\alpha}^{\alpha} \mathbf{r}_{\alpha\beta}^{\alpha} \quad , \tag{5.19}$$

since $\mathbf{v}_{\alpha\beta}^{l} = [0,0,0]^{\mathrm{T}}$ as the position between the alpha and beta frames is fixed. In Eq. (5.19):

$$\begin{aligned} \mathbf{v}_{l\alpha}^{l} &= \dot{\mathbf{r}}_{l\alpha}^{l} \quad , \\ \mathbf{v}_{\alpha\beta}^{l} &= \mathbf{C}_\alpha^l \mathbf{v}_{\alpha\beta}^{\alpha} = \mathbf{C}_\alpha^l \dot{\mathbf{r}}_{\alpha\beta}^{\alpha} \quad , \end{aligned} \tag{5.20}$$

and

$$\dot{\mathbf{C}}_\alpha^l = \mathbf{C}_\alpha^l \mathbf{\Omega}_{l\alpha}^{\alpha} \quad , \tag{5.21}$$

noting that $\mathbf{C}_{\alpha}^{l}$ is given by Eq. (5.18). The position vector derivative $\dot{\mathbf{r}}_{l\alpha}^{l}$ and $\dot{\mathbf{r}}_{\alpha\beta}^{\alpha}$ in Eq. (5.20) can be calculated using the custom numerical differentiation method, described in Appendix E. Also, the skew-symmetric matrix $\mathbf{\Omega}_{l\alpha}^{\alpha}$ of the angular rate $\boldsymbol{\omega}_{l\alpha}^{\alpha}$ can be calculated using:

$$\mathbf{\Omega}_{l\alpha}^{\alpha} = \left[\boldsymbol{\omega}_{l\alpha}^{\alpha} \wedge\right] = \frac{1}{\tau}\left[\boldsymbol{\alpha}_{l\alpha}^{\alpha} \wedge\right] \quad , \tag{5.22}$$

where $\tau$ the time interval between epochs (τ=0.01s) and $\boldsymbol{\alpha}_{l\alpha}^{\alpha}$ the attitude increment vector, between epoch *k* and *k+1*, which is a function of the respective coordinate transformation matrix (CTM):

$$\boldsymbol{\alpha}_{l\alpha}^{\alpha} = f\left(\mathbf{C}_{\alpha-}^{\alpha+}\right) = f\left(\mathbf{C}_{\alpha(t)}^{\alpha(t+\tau)}\right) = f\left(\mathbf{C}_{\alpha(k)}^{\alpha(k+1)}\right) \quad , \quad 1 \leq k \leq m, \tag{5.23}$$

as implied by Rodrigues' formula in Eq. (5.9). The CTM can then be given by Eq. (5.8).

The implementation of the HBM can be found in the Matlab scripts, provided in Appendix F on the thesis' accompanying CD.

### 5.3.3 **Gait cycle simulation**

This section describes the process of defining the PMM segment rotations, in order to simulate free-walking for one gait cycle, from published biomechanical studies, e.g. [19], [98], by extracting the human body segments' and joints' rotation angles over one gait cycle, e.g. knee rotation in 3D axes, and feeding them to the PMM. The following describes the process of extracting the 3D rotation angles from these studies, corresponding to one gait cycle, by using a custom Matlab toolkit "ExportGraphData", developed specifically to extract the relevant data from these studies and import them to the HBM in order to drive it by rotating its constituent joints and segments. To simulate a motion profile involving several steps, e.g. for the whole length of a PRM output path, it is necessary to combine a number of these extracted datasets. The underlying assumption is that due to the gait periodicity (discussed in Section 2.1.3), the motion profile corresponding to one gait cycle is repeated as many times needed in order to cover the applicable path length.

As discussed in Section 5.2, applicable HBM segments can rotate in order to simulate human walking, including arms swinging during walk. The custom Matlab toolkit "ExportGraphData" was were developed in order to export to the PMM profiles of joint angles during walking, from graphs of published biomechanical studies [19], [98], which illustrate the 3D rotations exhibited by human body joints and segments during a free-walking (nominal) gait cycle, e.g. ankle rotation in 3D. The custom Matlab toolkit "ExportGraphData" is included in Appendix F on the CD. The output of this process is a Matlab structure containing the 3D attitude positions (in Euler angles measured in degrees) of the HBM segments over one gait cycle. This custom process, which was developed specifically for this thesis, involves four stages which are described below and illustrated in Figure 5.3.

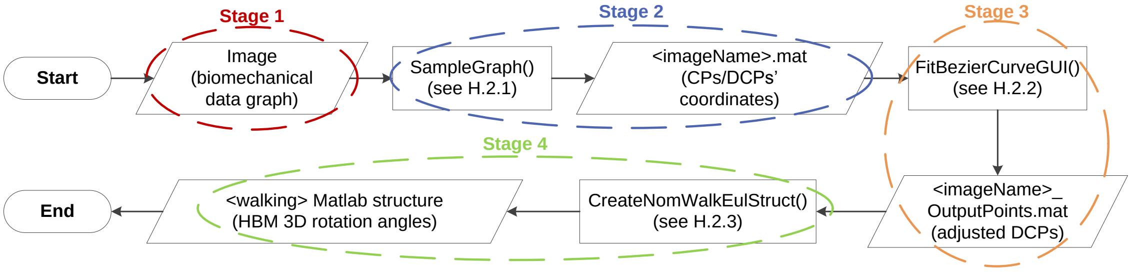
Stage 1
Stage 2
Stage 3
Stage 4
Start
Image (biomechanical data graph)
SampleGraph() (see H.2.1)
<imageName>.mat (CPs/DCPs' coordinates)
FitBezierCurveGUI() (see H.2.2)
<imageName>_ OutputPoints.mat (adjusted DCPs)
CreateNomWalkEulStruct() (see H.2.3)
<walking> Matlab structure (HBM 3D rotation angles)
End

Figure 5.3. Beziér curve-fitting workflow diagram (four stages)

The first stage of the process implemented to import join rotations from biomechanical studies to the HBM, is to identify appropriate biomechanical studies and save the graphs into an image format readable by Matlab. The second stage involves opening the image through the "ExportGraphData" Matlab toolkit in order to convert the image native coordinates (units in pixels) into the underlying graphs' units. Typically, the underlying graph horizontal shows normalised time for one gait cycle, and the vertical axis shows the rotation (in degrees) of a specific segment about a specific axis, e.g. ankle rotation about the Y-axis of the HBM, discussed in Section 5.2. Three points on the graph must be defined in order to transform the image coordinates to the actual units in the graph (degrees along the vertical axis and normalised time for one gait cycle along the horizontal axis). These points are selected as a. the intersection of the graph axes (0,0), b. the maximum horizontal axis value set to 100, i.e. (100,0), and c. the maximum vertical axis value *y* (which depends on the magnitude of rotation that a particular segment exhibits during the gait cycle), i.e. (0,*y*). This process is similar to geo-registering an aerial/satellite image to an Earth-referenced coordinate system.

The user can then define control points (CPs) along the underlying curve depicting the angle profile which need to be extracted to the HBM. The CPs should be chosen at local extrema and inflection points for a better cubic Beziér curve fit in the next stage. The tool will automatically create two derivative control points (DCPs) for each CP and initialise their positions with the same vertical axis value as the CP, with one DCP on the left and the other on the right of the CP, at an equal (arbitrary) distance of 3 horizontal axis units. These DCPs can be moved manually (by dragging them with the mouse). Their coordinates on the graph control the gradient of the fitted curve around the CP, i.e. the left DCP will control the gradient (first derivative) of the curve approaching to the CP, and the right DCP the gradient of the fitted curve when leaving the CP. Due to the gait cycle periodicity, the first and the last CPs are essentially the same point (with one gait period difference), so any action in one of the DCPs corresponding to the first/last CPs, will be automatically applied to the other as well, i.e. dragging the last CP's previous DCP up in the graph, will result to the first

CP's previous DCP being dragged up at the same amount. The tool saves the CPs and DCPs coordinates in tabular format (native Matlab *.mat file). If the user wishes to add/remove CPs, the same tool provides the option to read the saved tabular file and allow the user to add/remove new CPs (and save them to a new or overwriting the existing file).

It must be noted at this point that fitting cubic Beziér curves is a standard method used to interpolate human body segments' trajectories between two known positions [136], noting that higher order embodiments are out of this thesis' scope. In this thesis, cubic Beziér curves were used to interpolate between two attitude positions in 2D, i.e. the horizontal axis is the time axis and the vertical axis the angle of rotation, for each segment of the HBM. The interpolation by cubic Beziér curves was decided because a. this method was found to a. it does not require many parameters to be defined, and b. it can be used in a piece-wise manner, fitting more effectively the underlying angle profiles on the graphs provided by the biomechanical studies [19], [98], used to drive the HBM. To use a cubic Beziér fitting curve the coordinates of 4 points must be defined, which is two control points (CPs) for the initial and final attitude positions which are to be interpolated and two derivative control points (DCPs), each one controlling the first derivative of the fitted curve at the respective CP, i.e. the gradient of the fitted curve around that CP, for better fitting results. To represent the rotation of a human body joint/segment three defined (single-axis) attitude interpolations must be performed, noting that a gesture in the current PM embodiment can encompass multiple joint/segment rotations.

In the third stage of the curve-fitting process, the user opens the saved CPs and DCPs table in order to modify the DCPs positions (the CPs' coordinates cannot be modified at this stage), in order to fine-tune the curve-fitting. The custom toolkit "ExportGraphData" will automatically fit an initial cubic Beziér curve (explained later in this section), based on the CPs and DCPs coordinates in the coordinates table. Any change on the DCPs' position causes and automatic update of the fitted cubic spline on the graph. It is worth noting that in order to achieve a smooth transition of the fitted curve approaching and leaving the CPs, the left and right DCPs around a given CP are constrained to rotate in opposite directions at equal angles. The result is that the fitted curve is linear close to the CP (approaching and leaving from it). This implies that the first-order derivative at the specific CP is constant, so the second-order derivative at that specific CP is set to zero. By manual manipulation of the DCPs coordinates it is possible to fine-tune the cubic spline (thin orange line in Figure 5.4) to the underlying (blue) curve in question in the graph, depicting the (right) hip adduction angle about the X-axis of the pedestrian body frame, i.e. when the right leg moves towards the body. The user can save the curve-fitting results by clicking "Save Graph". The custom toolkit "ExportGraphData" will then sample at user-defined fixed time intervals (along the x-axis), typically at 101 samples, noting that the number of samples over one gait cycle can be increased (at multiples of 10), as appropriate. The first and last sample points show the same attitude position with one gait period difference. This allows multiple exported datasets to be joined together without discontinuities (by omitting the last sampled point). The number of gait cycles (datasets) joined together is controlled by the PMM, based on the length of the PRM trajectory (and the fixed stride length which depends on the stature), e.g. if the path is 15m long and the stride length 1.5m, then 10 datasets (representing segments' rotations over 10 gait cycles) are joined together to drive the HBM from the PRM path's start to finish. The fourth (and final) processing stage involves creating a Matlab structure containing the 3D rotation (Euler) angles of the HBM's joints/segments.

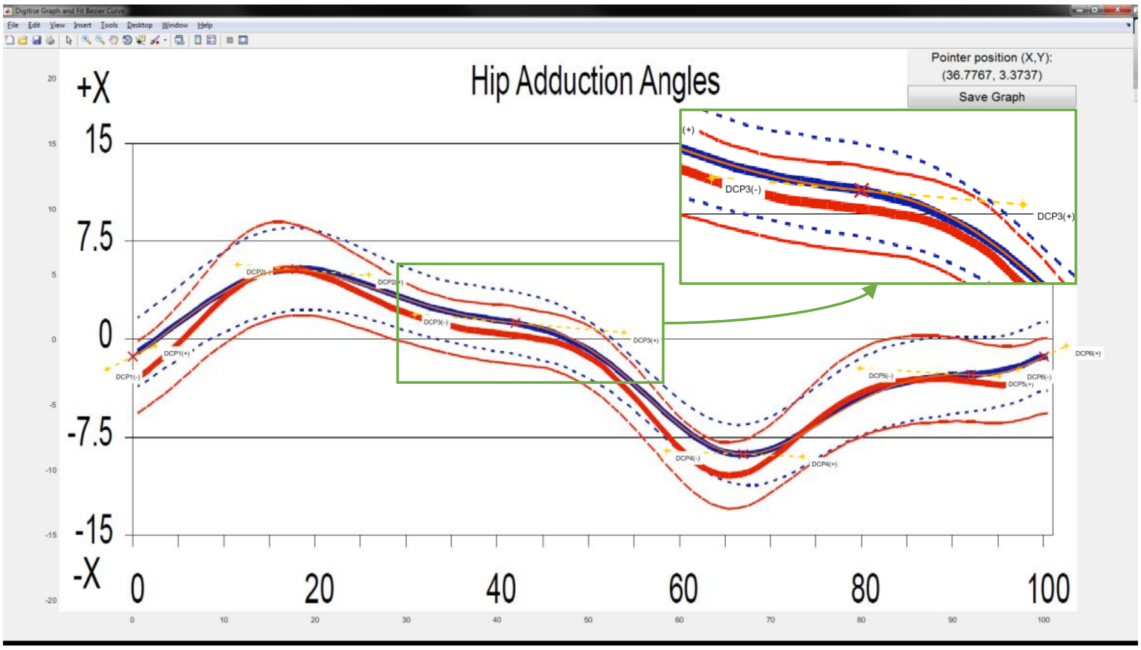

Figure 5.4. Custom Matlab toolkit "ExportGraphData" screenshot fitting a cubic Bezier curve (orange solid line) on biomechanical data (blue solid line)

## 5.4 Integrated Pedestrian Motion Model

This section describes the development method of the PMM, comprising the PRM (discussed in Section 5.2) and HBM (described in Section 5.3), whose purpose is to feed the SGCTLs (presented in Section 4.1) with synthetic human motion data. The constituent parts of the PMM are listed below:

- Human Biomechanical Model (HBM)
    - PNE/GNSS antenna location
    - Type of activity, e.g. walking
    - Gestures, e.g. answer phone
- Pedestrian Routing Model (PRM)
    - Dijkstra’s algorithm
    - Interpolation function

In the context of this thesis, the scope of the PMM is to simulate the motion dynamics’ environment of the PNE, which is attached on or held by a pedestrian, in the presence of pedestrian motion.

It is important to underline that for validation purposes, the PMM simulations have to be consistent with the human motion capture (MoCap) data, presented in Chapter 3. Pedestrian motion modelling encompasses human gait, e.g. walking or jogging, and overlaid gestures, e.g. “pick-up a phone” or “send a text”, assuming that the PNE is embedded in a smartphone. The range of simulated pedestrian motion, based on the human MoCap data are discussed in Section 6.1 of the thesis results. The PMM simulates the motion dynamics environment of a PNE, depending on:

- The selected route by the pedestrian, modelled by the PRM (which can also include turns).
- The type of activity (e.g. slow/fast walking).
- The PNE location on the human body (e.g. head-mounted or hand-held)
- Specific gestures of a human body segment, which can affect the PNE motion, in addition to the underlying nominal walking, e.g. “answer the phone” or “send a text”, assuming a smartphone with embedded PNE.
- The height of the HBM and the anthropometric modelling embedded in the HBM, i.e. the length of human body segments (normalised in human body height).

It is also worth noting that in the current PMM implementation, the step length is proportional to the user-defined height of the HBM, and is assumed that any step length variations during the MoCap experiments, e.g. due to a different (free) walking speed, are negligible. In addition to the currently supported features by the PMM, a (future) production version could support (see also future work in Chapter 8):

- A user interface for defining gestures, e.g. defining the final attitude of a human body segment, by rotating (in 3D) segments of a displayed human body model.

- Jogging and running activities, i.e. adding a “flight phase” to the modelled gait cycle (see Section 2.1.3).
- A pedestrian routing model with inclined trajectories, e.g. staircases. This presupposes modelling the appropriate gait motion ascending/descending stairs.
- Input of the stride (or step) length directly in the model as a parameter, noting that this would involve an inverse kinematics correction, similar to the one described in [137], in order to constrain the heel to walk on the ground level.

An underlying assumption for the development of the PMM, is that a specific PRM output path is independent from the sensor location on the human body, the particular activity and gestures that the pedestrian may perform. Also, it is assumed that the PRM route does not have an effect on the choice of particular types gestures and the time of their applicability to the HBM, e.g. whether/when a pedestrian answers a smartphone (with embedded PNE).

The human gait cycle forms the uniform basis of characterising the coordinated (and periodical) movement of human body segments during gait. All human body segments’ and joints’ movement can be referenced to the gait cycle, in terms of position and attitude (6 DOF). Assuming that human movement is identical across gait cycles, the HBM segments’ and joints’ motion is coherent, i.e. the relative phase between them w.r.t. the gait cycle remains constant. It follows from the gait cycle periodicity assumption, discussed in Section 2.1.3, that at the start/end of each gait cycle, the human body and its constituent segments and joints exhibit the same posture (attitude). In practice though, this is not the case as the variation in external conditions, e.g. ground inclination, will change slightly the human body posture between subsequent gait cycles; however, for the purposes of pedestrian motion modelling in this thesis, these factors are considered negligible and are discussed in the future work Chapter 8. Also, for modelling purposes, the PMM should be able to transition from/to the fundamental posture (described in Section 2.1.1) at the start/end of movement scenarios, respectively, as this is the posture at the beginning and end of MoCap, as described in Section 3.3. In the current PMM implementation, the step length is proportional to the user-defined height of the HBM, i.e. the step length variation has not been modelled, see future work in Chapter 8.

At initialisation, all HBM $\beta$ frames are parallel with the pedestrian $p$ frame (illustrated in Figure 5.2 of Section 5.3.1 which describes the HBM), therefore:

$$\left.\begin{aligned}\mathbf{r}^{p}_{p\beta}\big|_{k} &= \mathbf{C}^{p}_{\alpha}\big|_{k}\,\mathbf{r}^{\alpha}_{p\beta}\big|_{k}\\ \mathbf{C}^{p}_{\alpha}\big|_{1} &= \mathbf{I}_{3}\end{aligned}\right\}\Rightarrow \mathbf{r}^{\alpha}_{p\beta}\big|_{1} = \mathbf{r}^{p}_{p\beta}\big|_{1} \tag{5.24}$$

For a straight PRM output path, as in the current PMM implementation, the pedestrian $p$ frame is defined as parallel to the LTP $l$ frame at any epoch $k$ of movement:

$$\mathbf{C}_p^l\big|_k = \mathbf{I}_3 \quad , \quad 1 \le k \le m, \tag{5.25}$$

where *m* is the total number of epochs. The method of rotating the *p* frame w.r.t. *l* frame, i.e. by updating $\mathbf{C}_p^l\big|_k$, in order to follow a PRM output path involving turns.

The coordinates of the HBM child-frames *β*, e.g. the sensor "Sr" frame – listed in Table 5.1, referenced and resolved in *l* frame axes, are calculated using:

$$\mathbf{r}_{l\beta}^l\big|_k = \mathbf{r}_{l\alpha}^l\big|_k + \mathbf{C}_\alpha^l\big|_k\,\mathbf{r}_{\alpha\beta}^\alpha\big|_k \quad , \tag{5.26}$$

noting that $\mathbf{r}_{l\beta}^l\big|_k$ is the synthetic output of the PMM which can be fed to the SGCTLs, in order to assess the effect of the synthetic motion. An example of a synthetic motion 3D trajectory is illustrated in Figure 5.5. The PMM implementation also provides the option of a vertical correction, in order to ensure that at any instant during movement, at least one HBM node will be in touch with the ground level (for walking scenarios), i.e. the right/left leg heel ("Rhe", "Lhe") or toe ("Rfo", "Lfo"). In the current pedestrian motion model (PMM) results, presented in, the human biomechanical mode (HBM) follows either a straight path or a path with a U-turn, as discussed in the results presentation in Chapter 6.

All relevant gestures are applied on the HBM on the top of the underlying walking movement (and arms swing). To calculate the PMM all HBM points have to be referenced and resolved along local tangent frame (LTP) *l* axes. To do this, an additional rotation of the pedestrian frame *p* w.r.t. the *l* frames at each epoch *k* has to be used. The rotation matrix may change from epoch to epoch.

For epochs *k*>1, the rotation matrix needs to update the attitude of the pedestrian frame, in order to account for the turns (heading changes) of the followed PRM output path, assuming no tilt (pitch and roll) between *p* and *l* frames. The heading $\psi_{lp}$ of the *p* frame w.r.t. the *l* frame can be calculated by:

$$\psi_{lp} = \begin{cases} ATAN2\left(\dfrac{y_k - y_{k-1}}{x_k - x_{k-1}}\right) & , \quad x_k \ne x_{k-1} \\ \dfrac{\pi}{2} & , \quad x_k = x_{k-1},\ y_k > y_{k-1} \\ -\dfrac{\pi}{2} & , \quad x_k = x_{k-1},\ y_k < y_{k-1} \end{cases} \quad , \tag{5.27}$$

where $(x_k, y_k)$ the PRM output path coordinates (expressed in units of meters) at epoch *k*, noting that the position vector is $\mathbf{r}_{lp}^l\big|_k = \left[x_k, y_k, z_k\right]^T$. The change in attitude due to turns in the route, expressed as an attitude increment vector, is:

$$\boldsymbol{\alpha}_{lp}^{l} = \begin{pmatrix} 0 \\ 0 \\ \psi_{lp} \end{pmatrix} , \qquad (5.28)$$

so to re-align the $p$ frame the opposite attitude increment vector $-\boldsymbol{\alpha}_{lp}^{l}$ needs to be used in order to calculate $\mathbf{C}_{p+}^{p-} = \mathbf{C}_{p(t)}^{p(t-\tau)}$ using Eqs.  to (5.12), and update the $\mathbf{C}_{p}^{l}$ attitude matrix using Eq. (5.8).

Figure 5.5 shows an example of the PMM synthetic 3D position output for a sensor held by the right hand. The pedestrian routing model output path is not a straight line but involves slight turns, which the HBM follows.

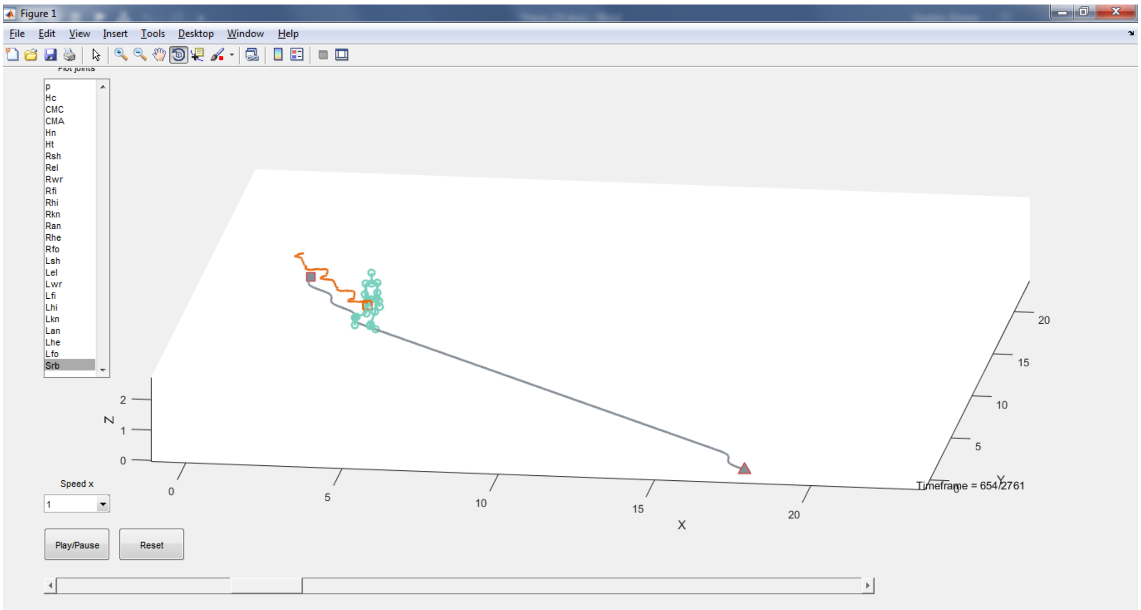

Figure 5.5. PMM synthetic output example for a hand-held sensor.

In summary, this chapter proposed and described a PMM enabling the simulation motion dynamics induced by pedestrian motion and exhibited by PNE equipment. The PMM output is a synthetic 3D position trajectory representing the motion of the PNE equipment. This synthetic output trajectory can then be fed into the SGCTLs in order to compare the effects on the performance of simulated carrier phase/tracking loops w.r.t. the effects of human captured motion (MoCap). The thesis continues with the presentation of the analytical method of the results in Chapter 6, which also encompass the comparative effects analysis for the real human MoCap and synthetic motion on the simulated GNSS phase lock loop (SGPLL).

# Chapter 6. Results

Following the description of the human motion captured (MoCap) data in Chapter 3, the description and validation of the simulated GNSS carrier tracking loops (SGCTLs) in Chapter 4, and the pedestrian motion model (PMM) description in Chapter 5, this chapter presents the thesis' results, which address the two research questions of the thesis discussed in Section 1.2, i.e. which are the types of pedestrian motion that most affect the performance of GNSS equipment and which is the best way to model pedestrian motion for simulation testing GNSS equipment. This chapter is divided into four sections, the first, Section 6.1, describes the method used to pre-process the data, i.e. convert them from the Xsens MTi-G readily available navigation solution to motion projected along the lines-of-sight between the sensor and three simulated static satellites. Also, Section 6.1, describes the analytical method used to assess the impact of real and synthetic output motion (from the pedestrian motion model – PMM). The chapter continues with Section 6.2, which addresses the first research question, by showing the effects of the real MoCap scenarios on the SGCTLs' performance. The third, Section 6.3, addresses the second research question, by analysing the effects of synthetic motion (from the PMM) on the simulated GNSS carrier-tracking loop (SGPLL) and comparing these results to the ones obtained from the real motion in Section 6.2. The chapter concludes with Section 6.4, which summarises the results of this chapter and provides recommendations for GNSS receiver design in the presence of typical pedestrian motion.

The results are based on analysing the impact of real and synthetic human motion using the SGCTLs (see Chapter 4), i.e. in a Matlab simulation environment. Using a real receiver to assess the impact of human motion was beyond the timescales of the thesis, but is part of the productization process of the thesis' outcomes, as explained in Chapter 8 "Future Work".

## 6.1 Analytical Method

### 6.1.1 Real motion data pre-processing

This section discusses the three methods identified in order to process the human motion captured (MoCap) data in the main experiment of the thesis (see Section 3.3) and describes the selected method for pre-processing the MoCap data in order to generate the line-of-sight (LOS) motion profiles required to drive the SGCTLs (see Sections 4.2 and 4.3). Also, this pre-processing stage aims to ensure that the MoCap data represent the underlying motion, without any potential artefacts induced as a result of the MoCap process. After pre-processing, the MoCap data are ready to enter the SGCTLs, in order to assess the impact of real motion on the SGCTLs. The three methods identified to pre-process the MoCap data are listed below (and denoted with "A", "B" and "C"):

A. Convert the Xsens MTI-G-provided integrated inertial/GPS navigation solution, from latitude, longitude, altitude (LLA) coordinates, expressed in North, East and Down (NED) axes, to local tangent plane (LTP) coordinates.

Smooth the integrated inertial/GPS navigation solution (A), i.e. already converted from LLA to LTP coordinates, by detecting and removing ensuing transients from the internal Xsens Kalman Filter, as the examples shown in

B. Figure 3.4 and Figure 3.5.

C. Calculate the inertial navigation solution, using the measurements from the inertial sensors (accelerometers and gyros) following the calculation steps described in Section 2.2.2. and convert it from LLA to LTP (similar to solutions "B" and "C").

Options "A" and "B" were not selected as data pre-processing methods, since the results, discussed in Chapter 6, show they may still introduce artificial false frequency locks on the simulated GNSS frequency lock loop (SGFLL), which was described Section 4.3. The calculation of the inertial navigation solution, method "C", encompasses the following four stages:

- Stage 1: Read and validate for recording errors the integrated navigation solution provided by Xsens MTi-G. The navigation solution is expressed in North East and Down local navigation frame axes. The position solution is given in curvilinear form, i.e. latitude, $\breve{L}_b$ (rad), longitude, $\breve{\lambda}_b$ (rad), and height, $\breve{h}_b$ (m); the velocity solution is denoted with $\breve{\mathbf{v}}_{eb}^{n} = \begin{bmatrix} \breve{v}_{eb,N}^{n} & \breve{v}_{eb,E}^{n} & \breve{v}_{eb,D}^{n} \end{bmatrix}^{\mathrm{T}}$ (units in m/s) and the attitude solution in the form of a coordinate transformation matrix (CTM) is denoted with $\breve{\mathbf{C}}_{b}^{n}$.
- Stage 2: Average the inertial sensor measurements (specific force, angular velocity) from Xsens MTi-G which are time-valid at epochs instead of being valid over time intervals, as required by the inertial navigation equations. The time of validity for inertial sensor measurement required by the applicable inertial navigation equations is discussed in Section 2.2.2 and illustrated in Figure 2.8. The average inertial sensor measurements, specific force $\bar{\mathbf{f}}_{ib}^{b}\big|_{k}$ and angular velocity $\bar{\boldsymbol{\omega}}_{ib}^{b}\big|_{k}$ at epoch *k* are:

$$\left.\begin{aligned} \bar{\mathbf{f}}_{ib}^{b}\big|_{k} &= \frac{\mathbf{f}_{ib}^{b}\big|_{k-1}+\mathbf{f}_{ib}^{b}\big|_{k}}{2} \\ \bar{\boldsymbol{\omega}}_{ib}^{b}\big|_{k} &= \frac{\boldsymbol{\omega}_{ib}^{b}\big|_{k-1}+\boldsymbol{\omega}_{ib}^{b}\big|_{k}}{2} \end{aligned}\right. , \quad 2 \le k \le m, \qquad (6.1)$$

noting that for the first epoch (*k*=1) the average inertial sensor measurements are equal to the non-averaged ones.

- Stage 3: Calculate the inertial navigation solution for position in curvilinear form, i.e. latitude, $L_b$ (rad), longitude, $\lambda_b$ (rad), and height, $h_b$ (m); the velocity solution, denoted with $\mathbf{v}_{eb}^{n} = \begin{bmatrix} v_{eb,N}^{n} & v_{eb,E}^{n} & v_{eb,D}^{n} \end{bmatrix}^{\mathrm{T}}$ (units in m/s) and the attitude solution, denoted with $\mathbf{C}_b^n$. This is done by applying the described method and implemented Matlab function "Nav_equations_NED" from [3], noting that this Matlab function is also provided in the accompanying CD of the thesis.
- Stage 4: Convert the inertial navigation solution to LTP resolving axes, in two (sub)stages:
  a. the first one encompasses converting the inertial navigation solution into Cartesian ECEF frame resolving axes, by applying the described method in , from Eq. (B.2) to Eq. (B.6), based on the implemented Matlab function "NED_to_ECEF" from [3], which is provided in the accompanying CD of the thesis, and
  b. the second one encompasses converting the navigation solution from ECEF to LTP resolving axes, by applying the described method in Appendix B, from Eq. (B.26) to Eq. (B.30), implemented in Matlab function "ECEF_to_LTP", which is provided in the accompanying CD of the thesis. The final output of this stage is the position vector $\tilde{\mathbf{r}}_{lb}^{l}$, velocity vector $\tilde{\mathbf{v}}_{lb}^{l}$ and attitude CTM $\tilde{\mathbf{C}}_b^l$.
- Stage 5: Rotate the position and velocity vectors (fed into the SGCTLs) on the horizontal plane in order to align the motion along the forward direction with the Northings, and motion along the right direction with the Eastings. As forward motion direction is considered the vector from the start to the end positions for a straight walking path (without a U-turn), and the vector from the start to the point with the maximum Northing for a U-turn walking path, as illustrated in Figure 6.1, noting that this analysis uses the Xsens MTi-G integrated position solutions (Option "A") which do not drift, due to the applicable GPS updates. The rotation of the position and velocity solutions on the horizontal plan is applied by rotating the integrated position solution trajectory 0.01° anti-clockwise, until the start and end points (for paths without a U-turn) or the start and maximum Northing points (for paths with a U-turn, as the one illustrated in Figure 6.1) have an Eastings' difference less than 10cm. This process is implemented in Matlab function "CreateRealMotionDataStruct", provided in Appendix F with the accompanying CD of the thesis.

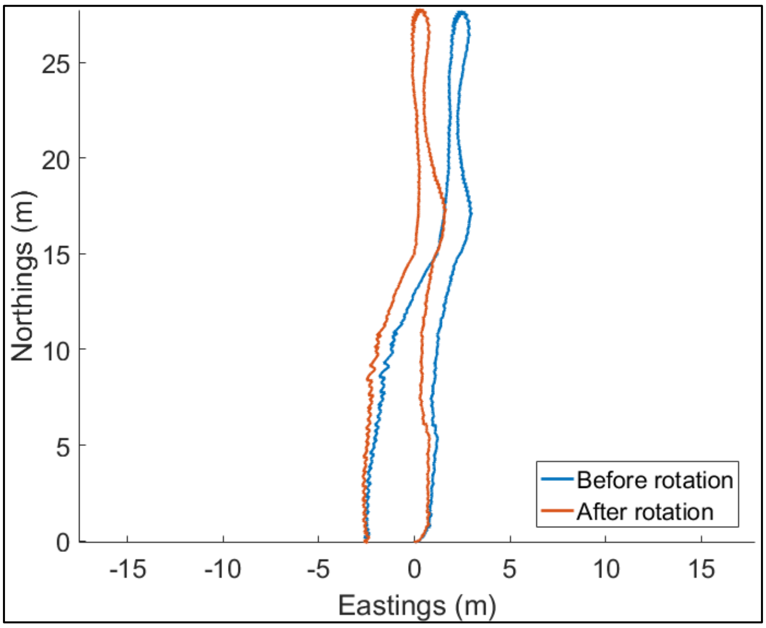

Figure 6.1. Rotation example

The final calculated (inertial) position and velocity solutions are referred as "real motion" in the remainder of this thesis. By aligning the MoCap Forward and Right direction with the Northings' and Eastings' LTP axes, each component of the (final) position vector $\mathbf{r}_{lb}^{l}=\begin{bmatrix} r_{lb,F}^{l} & r_{lb,R}^{l} & r_{lb,D}^{l}\end{bmatrix}^{\mathrm{T}}$ and the velocity vector $\mathbf{v}_{lb}^{l}=\begin{bmatrix} v_{lb,F}^{l} & v_{lb,R}^{l} & v_{lb,D}^{l}\end{bmatrix}^{\mathrm{T}}$ refers to Forward, Right and Down direction ("FRD"), which readily provide the LOS motion w.r.t. three hypothetical static satellites placed along the FRD axes (assumed to be at an appropriate distance from the body-frame, i.e. exceeding the maximum magnitudes of the position components along each FRD direction). It is also important to note that this process is not required for the synthetic motion from the PMM, as they are readily output referenced and resolved to LTP axes which are aligned with the FRD directions.

The calculated inertial position and velocity solutions (Option "C") do not introduce artefacts in terms of false frequency locks (see Figure 6.2) to the simulated GNSS frequency lock loop (SGFLL), as the ones observed for options "A" and "B", which are illustrated in Figure B.1 and Figure B.4, respectively, in Appendix B. Therefore, it is appropriate to use the calculated inertial navigation solution using method "C" for driving the SGCTLs. A disadvantage of the inertial navigation solution is that the position, velocity and attitude solutions drift, due to the accumulated errors from the inertial sensors, as explained in Section 2.2.2. However, for the purposes of this thesis, it is more important to drive the SGCTLs using LOS motion which does not introduce any artefacts, due to the GPS-induced transients (see Section 3.2). A proposed method for combining the advantages of the inertial navigation solution (transients-free) with the integrated navigation solution (drift-free) is presented in Appendix B, noting that it requires further qualification, as discussed in future work Chapter 6.

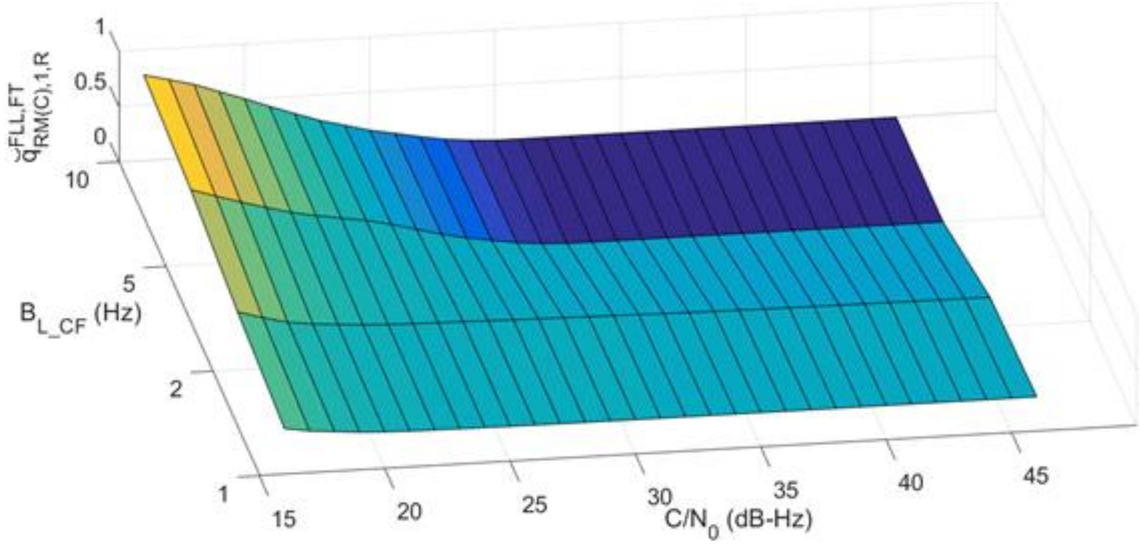

Figure 6.2. Real motion (inertial navigation solution, – subscript "C") results to SGFLL example, without any observed artefacts.

### 6.1.2 Analytical method for real motion impact assessment on the SGCTLs

Section 6.1.1 described the process of processing the MoCap data in order to generate the LOS motion profiles required to drive the SGCTLs (see Sections 4.2 and 4.3). This section describes the analytical method for assessing the effect of the MoCap and synthetic motion profiles SGCTLs, in order to answer which types of real motion affect most the performance of the SGCTLs, and validate the PMM in terms of comparing the effects of the synthetic motion to the real one. The list of analysed MoCap scenarios is presented in Table 3.2. The analysis in this section addresses the first research question of the thesis, discussed in Section 1.2, i.e. which are the key aspects of pedestrian motion that affect the performance of GNSS equipment.

The analysis of this section encompasses feeding the real MoCap data, i.e. the inertial position and the static test scenario (control data) into the simulated GNSS carrier-tracking loops (SGCTLs). The static test case scenario is the same one used for the validation of the SGCTLs, in Section 4.4, noting that the acronym "SGCTLs" encompasses a simulated GNSS phase lock loop (SGPLL) and a simulated GNSS frequency lock loop (SGFLL), in order to investigate the errors induced in terms of cycle slips (for SGPLL only) and frequency false locks (both for SGPLL and SGFLL).

The result from each MCS, i.e. using the SGCTLs for a given effective bandwidth $B_L$ and $C/N_0$, is a scalar $\breve{q}\Big|_{(B_L,C/N_0)}$, which represents the relative frequency of cycle slip/false frequency lock occurrences across the epochs of a given input motion profile, i.e. the ratio of the total number of cycle slips detected for a given motion profile, over the total number of epochs minus one, as the tracking error at the first epoch is initialised (see Sections 4.2 and 4.3) in way that does not cause a cycle slip. Please also note that $\breve{q} \geq 0$, as the $\breve{q}$ values express relative frequency which is a non-negative quantity. The total count is then divided by the number of epochs of a given MoCap profile *i* in order to provide the relative frequency:

$$\left\{\begin{array}{l} \breve{q}_{S,LOS}^{PLL,PT}\Big|_{(B_L,C/N_0)} = \dfrac{N_{PLL,cs}\Big|_{(B_L,C/N_0)}}{(m-1)\cdot n} \\ \breve{q}_{RM,i,LOS}^{PLL,PT}\Big|_{(B_L,C/N_0)} = \dfrac{N_{PLL,cs}\Big|_{(B_L,C/N_0)}}{(m-1)\cdot n} \\ \breve{q}_{PMM,i,LOS}^{PLL,PT}\Big|_{(B_L,C/N_0)} = \dfrac{N_{PLL,cs}\Big|_{(B_L,C/N_0)}}{(m-1)\cdot n} \end{array}\right. , \\ \left\{\begin{array}{l} \breve{q}_{S,LOS}^{FLL,FT}\Big|_{(B_L,C/N_0)} = \dfrac{N_{FLL,ffl}\Big|_{(B_L,C/N_0)}}{(m-1)\cdot n} \\ \breve{q}_{RM,i,LOS}^{FLL,FT}\Big|_{(B_L,C/N_0)} = \dfrac{N_{FLL,ffl}\Big|_{(B_L,C/N_0)}}{(m-1)\cdot n} \\ \breve{q}_{PMM,i,LOS}^{FLL,FT}\Big|_{(B_L,C/N_0)} = \dfrac{N_{FLL,ffl}\Big|_{(B_L,C/N_0)}}{(m-1)\cdot n} \end{array}\right. , \tag{6.2}$$

noting that the $\breve{q}$ superscripts: “*PLL,PT*” and *“FLL,FT”*, denote PLL phase-tracking and FLL frequency-tracking, respectively. The “S” subscript denotes the static case scenario, and “*RM,i*” or “*PMM,i*” subscripts denote a real or synthetic, respectively, motion scenario with index *i*, which follows the indexing of MoCap scenarios from Table 3.2. The “LOS” subscript denotes the line-of-sight towards the simulated static satellites, along the forward “F”, right “R” or down “D” directions. *N* is the number of epochs that exhibited tracking slips (cycle slips or false frequency locks), with “*PLL,cs*” and “*FLL,ffl*” subscripts denoting PLL cycle slips and FLL false frequency locks, respectively. Also, *m* is the number of epochs of the real motion scenario *i* and *n* the number of simulation runs in each Monte-Carlo set of $(B_L, C/N_0)$ parameters. The meaning of $(m\text{-}1)$ in the denominator is that the first epoch is excluded from the total count of epochs as the pre-defined initial phase/frequency tracking errors in the SGCTLs (phase and frequency, see Sections 4.2 and 4.3) do not introduce cycle slips or frequency false locks to the SGCTLs. Unless otherwise specified, the subscript indexes $(B_L, C/N_0)$ have been removed from the notations in the remainder of this chapter, for readability purposes.

To facilitate the analysis in this section and provide results independent of the particular choice of the LOS (forward "F", right "R" or down "D") between the sensor and the simulated (static) satellite, the 3D equivalent of the $q$ elements in Eq. (6.2) was calculated, e.g. for the SGPLL phase tracking for a static test case scenario, without loss of generality, i.e. it applies also to SGPLL and SGFLL frequency-tracking:

$$
\left\{
\begin{array}{l}
q_S^{PLL,PT} = \dfrac{\breve{q}_{S,F}^{PLL,PT} + \breve{q}_{S,R}^{PLL,PT} + \breve{q}_{S,D}^{PLL,PT}}{3} = \dfrac{3 \cdot \breve{q}_{S,LOS}^{PLL,PT}}{3} = \breve{q}_{S,LOS}^{PLL,PT} \\
q_{RM,i}^{PLL,PT} = \dfrac{\breve{q}_{RM,i,F}^{PLL,PT} + \breve{q}_{RM,i,R}^{PLL,PT} + \breve{q}_{RM,i,D}^{PLL,PT}}{3} \\
q_{PMM,i}^{PLL,PT} = \dfrac{\breve{q}_{PMM,i,F}^{PLL,PT} + \breve{q}_{PMM,i,R}^{PLL,PT} + \breve{q}_{PMM,i,D}^{PLL,PT}}{3}
\end{array}
\right. ,
\qquad
\left\{
\begin{array}{l}
q_S^{FLL,FT} = \dfrac{\breve{q}_{S,F}^{FLL,FT} + \breve{q}_{S,R}^{FLL,FT} + \breve{q}_{S,D}^{FLL,FT}}{3} = \dfrac{3 \cdot \breve{q}_{S,LOS}^{FLL,FT}}{3} = \breve{q}_{S,LOS}^{FLL,FT} \\
q_{RM,i}^{FLL,FT} = \dfrac{\breve{q}_{RM,i,F}^{FLL,FT} + \breve{q}_{RM,i,R}^{FLL,FT} + \breve{q}_{RM,i,D}^{FLL,FT}}{3} \\
q_{PMM,i}^{FLL,FT} = \dfrac{\breve{q}_{PMM,i,F}^{FLL,FT} + \breve{q}_{PMM,i,R}^{FLL,FT} + \breve{q}_{PMM,i,D}^{FLL,FT}}{3}
\end{array}
\right. ,
\tag{6.3}
$$

Also, Eq. (6.3) assumes that the $\breve{q}$ values, along each LOS for the static case scenario, are equal. Examples of the $q_{RM,i}^{PLL,PT}\Big|_{(B_L,C/N_0)}$ and $q_{RM,i}^{FLL,FT}\Big|_{(B_L,C/N_0)}$ elements are shown in Figure 6.3 and Figure 6.4, respectively. It is worth noting that the It follows that the standard deviation (SD) of $q$'s (assumed to be statistically independent variables):

$$
\left\{
\begin{array}{l}
\sigma(q_S^{PLL,PT}) = \dfrac{\sigma(\breve{q}_S^{PLL,PT})}{\sqrt{3}} \\
\sigma(q_{RM,i}^{PLL,PT}) = \dfrac{1}{\sqrt{3}}\left[\sigma(\breve{q}_{RM,i,F}^{PLL,PT})^2 + \sigma(\breve{q}_{RM,i,R}^{PLL,PT})^2 + \sigma(\breve{q}_{RM,i,D}^{PLL,PT})^2\right]^{\frac{1}{2}} \\
\sigma(q_{PMM,i}^{PLL,PT}) = \dfrac{1}{\sqrt{3}}\left[\sigma(\breve{q}_{PMM,i,F}^{PLL,PT})^2 + \sigma(\breve{q}_{PMM,i,R}^{PLL,PT})^2 + \sigma(\breve{q}_{PMM,i,D}^{PLL,PT})^2\right]^{\frac{1}{2}}
\end{array}
\right. ,
\qquad
\left\{
\begin{array}{l}
\sigma(q_S^{FLL,FT}) = \dfrac{\sigma(\breve{q}_S^{FLL,FT})}{\sqrt{3}} \\
\sigma(q_{RM,i}^{FLL,FT}) = \dfrac{1}{\sqrt{3}}\left[\sigma(\breve{q}_{RM,i,F}^{FLL,FT})^2 + \sigma(\breve{q}_{RM,i,R}^{FLL,FT})^2 + \sigma(\breve{q}_{RM,i,D}^{FLL,FT})^2\right]^{\frac{1}{2}} \\
\sigma(q_{PMM,i}^{FLL,FT}) = \dfrac{1}{\sqrt{3}}\left[\sigma(\breve{q}_{PMM,i,F}^{FLL,FT})^2 + \sigma(\breve{q}_{PMM,i,R}^{FLL,FT})^2 + \sigma(\breve{q}_{PMM,i,D}^{FLL,FT})^2\right]^{\frac{1}{2}}
\end{array}
\right. .
\tag{6.4}
$$

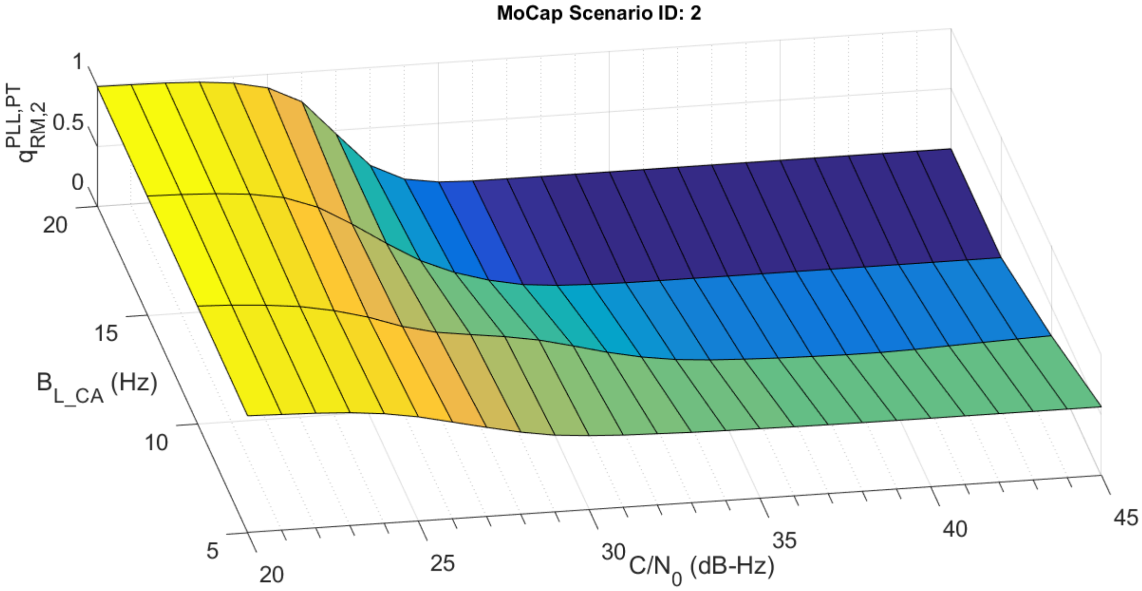

Figure 6.3. Relative frequency of cycle-slip occurrences on SGPLL (for all epochs and MCS runs), due to a real motion scenario (ID: 2), encompassing walking 30m in a straight line having the sensor in the pocket.

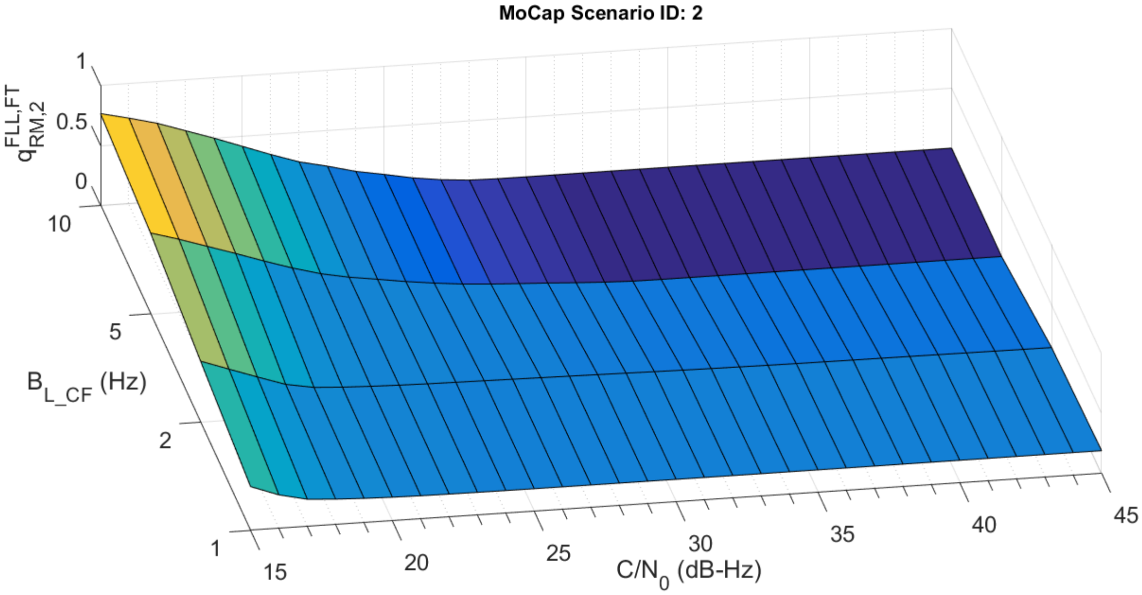

Figure 6.4. Relative frequency of false frequency-lock occurrences on SGFLL (for all epochs and MCS runs), due to a real motion scenario (ID: 2), encompassing walking 30m in a straight line having the sensor in the pocket.

Figure 6.3 and Figure 6.4 show that the SGPLL and SGFLL, respectively, can track pedestrian motion more effectively when both the effective bandwidth and $C/N_0$ are high. However, the SGCTLs' validation results presented in Section 4.4, show that under static conditions the SGCTLs exhibit more tracking noise at higher effective bandwidths (see Figure 4.4 and Figure 4.5.), and as a result will exhibit more errors in terms of cycle slips and false frequency locks (see also Table 4.1 for minimum *C/N₀* levels).

The difference $\Delta q$ is the tracking slip difference (encompassing cycle slips and false frequency locks, as appropriate), between the real motion scenario *i* (or synthetic motion scenario, where appropriate) and the static case scenario *q* values:

$$\begin{cases} \Delta q_{RM,i/S}^{PLL,PT} = q_{RM,i}^{PLL,PT} - q_{S}^{PLL,PT} \\ \Delta q_{PMM,i/S}^{PLL,PT} = q_{PMM,i}^{PLL,PT} - q_{S}^{PLL,PT} \end{cases} ,$$
$$\begin{cases} \Delta q_{RM,i/S}^{FLL,FT} = q_{RM,i}^{FLL,FT} - q_{S}^{FLL,FT} \\ \Delta q_{PMM,i/S}^{FLL,FT} = q_{PMM,i}^{FLL,FT} - q_{S}^{FLL,FT} \end{cases} . \tag{6.5}$$

It follows, assuming that *q*'s are statistically independent variables that the standard deviation (SD) of $\Delta q$ is equal to the root of the squares sum of the SDs of $q_{S}^{PLL,PT}$, and $q_{RM,i}^{PLL,PT}$ (or $q_{PMM,i}^{PLL,PT}$, as appropriate), noting that the SD is calculated separately for each row $\Delta q$, i.e. for each simulated effective bandwidth:

$$\begin{cases} \sigma(\Delta q_{RM,i/S}^{PLL,PT}\big|_{B_L}) = \sqrt{\sigma(q_{S}^{PLL,PT}\big|_{B_L})^2 + \sigma(q_{RM,i}^{PLL,PT}\big|_{B_L})^2} \\ \sigma(\Delta q_{PMM,i/S}^{PLL,PT}\big|_{B_L}) = \sqrt{\sigma(q_{S}^{PLL,PT}\big|_{B_L})^2 + \sigma(q_{PMM,i}^{PLL,PT}\big|_{B_L})^2} \end{cases} ,$$
$$\begin{cases} \sigma(\Delta q_{RM,i/S}^{FLL,FT}\big|_{B_L}) = \sqrt{\sigma(q_{S}^{FLL,FT}\big|_{B_L})^2 + \sigma(q_{RM,i}^{FLL,FT}\big|_{B_L})^2} \\ \sigma(\Delta q_{PMM,i/S}^{FLL,FT}\big|_{B_L}) = \sqrt{\sigma(q_{S}^{FLL,FT}\big|_{B_L})^2 + \sigma(q_{PMM,i}^{FLL,FT}\big|_{B_L})^2} \end{cases} . \tag{6.6}$$

Therefore, if the tracking slip difference $\Delta q$ exceeds three times the SD *σ(Δq)*, the motion scenario is considered to have introduced significant dynamic stress to the SGCTLs, at a confidence level of 99.7%, under the assumption that the $\Delta q$ distribution is Gaussian. It follows that if the tracking slip difference $\Delta q$ exceeds more than 0.6745 times the SD *σ(Δq)*, defined as tolerance $T_{\Delta q}$, then the motion scenario has introduced significant dynamic stress to the SGCTLs, at a confidence level above a 50% level:

$$\begin{cases} \left|\Delta q_{RM,i/S}^{PLL,PT}\big|_{B_L}\right| > T_{\Delta q} \\ \left|\Delta q_{PMM,i/S}^{PLL,PT}\big|_{B_L}\right| > T_{\Delta q} \\ \left|\Delta q_{RM,i/S}^{FLL,FT}\big|_{B_L}\right| > T_{\Delta q} \\ \left|\Delta q_{RM,i/S}^{FLL,FT}\big|_{B_L}\right| > T_{\Delta q} \end{cases} , \quad T_{\Delta q} = 0.6745 \cdot \sigma(\Delta q) \quad . \tag{6.7}$$

The metric *M* used to represent the effect of each MoCap scenario on the SGCTLs, is:

$$\begin{cases} M^{PLL,PT}_{RM,i/S} = \sum_{B_{PLL}} \sum_{C_{PLL}} \frac{\Delta q^{PLL,PT}_{RM,i/S}}{\sigma\left(\Delta q^{PLL,PT}_{RM,i/S}\right)} \\ M^{PLL,PT}_{PMM,i/S} = \sum_{B_{PLL}} \sum_{C_{PLL}} \frac{\Delta q^{PLL,PT}_{PMM,i/S}}{\sigma\left(\Delta q^{PLL,PT}_{PMM,i/S}\right)} \end{cases},$$

$$\begin{cases} M^{FLL,FT}_{RM,i/S} = \sum_{B_{FLL}} \sum_{C_{FLL}} \frac{\Delta q^{FLL,FT}_{RM,i/S}}{\sigma\left(\Delta q^{FLL,FT}_{RM,i/S}\right)} \\ M^{FLL,FT}_{PMM,i/S} = \sum_{B_{FLL}} \sum_{C_{FLL}} \frac{\Delta q^{FLL,FT}_{PMM,i/S}}{\sigma\left(\Delta q^{FLL,FT}_{PMM,i/S}\right)} \end{cases}, \qquad (6.8)$$

where $M^{PLL,PT}_{RM,i/S}$ and $M^{FLL,FT}_{RM,i/S}$, follow the same superscript and subscript notation as for $\Delta q$. The $B_{PLL}$ (SGPLL effective bandwidths') set is equal to {5,10,15,20}Hz and the $B_{FLL}$ (SGFLL effective bandwidths') set is equal to {1,2,5,10}Hz. The $C_{PLL}$ and $C_{FLL}$ are defined, following the discussion in Sections 4.4.1 and 4.4.2, respectively, which provided the $C/N_0$ values where the SGCTLs' tracking noise exceeded the factor-of-two set threshold of the simulated over the theoretical tracking noise SDs' ratio, with the results summarised in Table 4.1.

In addition to the relative frequency of cycle slips $\breve{q}$ elements, calculated from the SGPLL via Eq. (6.2), it is useful to define a metric corresponding to the percentage of MCS runs which exhibit at least one cycle slip:

$$\begin{cases} \breve{p}^{PLL,PT}_{S,LOS}\Big|_{(B_L,C/N_0)} = \dfrac{\sum_{j=1}^{n} f(\delta\varphi_{ca,k}, j)}{n} \\ \breve{p}^{PLL,PT}_{RM,i,LOS}\Big|_{(B_L,C/N_0)} = \dfrac{\sum_{j=1}^{n} f(\delta\varphi_{ca,k}, j)}{n} \\ \breve{p}^{PLL,PT}_{PMM,i,LOS}\Big|_{(B_L,C/N_0)} = \dfrac{\sum_{j=1}^{n} f(\delta\varphi_{ca,k}, j)}{n} \end{cases}, \quad 1 \le k \le m, \qquad (6.9)$$

$$f(\delta\varphi_{ca,k}, j) = \begin{cases} 1 & , \quad \left|\delta\varphi_{ca,k}\right| > \dfrac{\pi}{2} \\ 0 & , \quad otherwise \end{cases}, \quad 1 \le k \le m, \quad 1 \le j \le n,$$

where $\delta\varphi_{ca,k}$ is the phase tracking error (rad) at epoch $k$ (within a specific MCS run $j$) and observing that function $f$ evaluates to unity (when non-zero) only if $\left|\delta\varphi_{ca,k}\right| > \dfrac{\pi}{2}$, which is the cycle-slip detection criterion used in Section 4.2. To help illustrate Eq. (6.9), it is useful to consider the phase tracking error elements of the matrix:

$$\delta\varphi_{ca(k,j)} = \begin{bmatrix} \delta\varphi_{ca(1,1)} & \delta\varphi_{ca(1,2)} & \dots & \delta\varphi_{ca(1,n)} \\ \delta\varphi_{ca(2,1)} & \delta\varphi_{ca(2,2)} & \dots & \delta\varphi_{ca(2,n)} \\ \vdots & \vdots & \ddots & \vdots \\ \delta\varphi_{ca(m,1)} & \delta\varphi_{ca(m,2)} & \dots & \delta\varphi_{ca(m,n)} \end{bmatrix}, \qquad (6.10)$$
$$1 \leq k \leq m, \quad 1 \leq j \leq n,$$

noting that this matrix corresponds to a single MCS for a pair of simulated effective bandwidth $B_{PLL}$ and $C/N_0$ parameters, with the rows (indexed with *k*, total number is *m*) referring to the epochs of the input motion (subscripts "*RM,i*", "*PMM,i*" for real and synthetic motion, respectively) or static (subscript "S") profiles, and with the columns (indexed with *j*, total number is *n*) referring to the number of MCS runs, i.e. one run per each added pseudorandom Matlab sequence, as discussed in Section 4.2.

Similarly, we can define the percentage of MCS runs per ($B_{FLL}$, $C/N_0$), which exhibit at least one false frequency lock:

$$\begin{cases} \breve{p}_S^{FLL,FT}\Big|_{(B_L,C/N_0)} = \dfrac{\sum\limits_{j=1}^{n} g(\delta f_{ca,k}, j)}{n} \\ \breve{p}_{RM,i}^{FLL,FT}\Big|_{(B_L,C/N_0)} = \dfrac{\sum\limits_{j=1}^{n} g(\delta f_{ca,k}, j)}{n} \\ \breve{p}_{PMM,i}^{FLL,FT}\Big|_{(B_L,C/N_0)} = \dfrac{\sum\limits_{j=1}^{n} g(\delta f_{ca,k}, j)}{n} \end{cases}, \quad 1 \leq k \leq m, \qquad (6.11)$$

$$g(\delta f_{ca,k}, j) = \begin{cases} 1 & , \quad |\delta f_{ca,k}| > \dfrac{1}{4\tau_\alpha} \\ 0 & , \quad otherwise \end{cases}, \quad 1 \leq k \leq m, \quad 1 \leq j \leq n,$$

where $\delta f_{ca,k}$ the frequency tracking error (Hz) at epoch *k* (within a specific MCS run *j*), and noting that function *g* evaluates to unity (when non-zero) only if $|\delta f_{ca,k}| > \dfrac{1}{4\tau_\alpha}$, with $\tau_\alpha$ (s) the correlator's accumulation time interval, discussed in Section 4.3.

The average *p* values for PLL phase-tracking and FLL frequency-tracking will then be:

$$\begin{cases} p_{S}^{PLL,PT} = \dfrac{\breve{p}_{S,F}^{PLL,PT} + \breve{p}_{S,R}^{PLL,PT} + \breve{p}_{S,D}^{PLL,PT}}{3} = \dfrac{3 \cdot \breve{p}_{S,LOS}^{PLL,PT}}{3} = \breve{p}_{S,LOS}^{PLL,PT} \\ p_{RM,i}^{PLL,PT} = \dfrac{\breve{p}_{RM,i,F}^{PLL,PT} + \breve{p}_{RM,i,R}^{PLL,PT} + \breve{p}_{RM,i,D}^{PLL,PT}}{3} \\ p_{PMM,i}^{PLL,PT} = \dfrac{\breve{p}_{PMM,i,F}^{PLL,PT} + \breve{p}_{PMM,i,R}^{PLL,PT} + \breve{p}_{PMM,i,D}^{PLL,PT}}{3} \end{cases} ,$$

$$\begin{cases} p_{S}^{FLL,FT} = \dfrac{\breve{p}_{S,F}^{FLL,FT} + \breve{p}_{S,R}^{FLL,FT} + \breve{p}_{S,D}^{FLL,FT}}{3} = \dfrac{3 \cdot \breve{p}_{S,LOS}^{FLL,FT}}{3} = \breve{p}_{S,LOS}^{FLL,FT} \\ p_{RM,i}^{FLL,FT} = \dfrac{\breve{p}_{RM,i,F}^{FLL,FT} + \breve{p}_{RM,i,R}^{FLL,FT} + \breve{p}_{RM,i,D}^{FLL,FT}}{3} \\ p_{PMM,i}^{FLL,FT} = \dfrac{\breve{p}_{PMM,i,F}^{FLL,FT} + \breve{p}_{PMM,i,R}^{FLL,FT} + \breve{p}_{PMM,i,D}^{FLL,FT}}{3} \end{cases} . \qquad (6.12)$$

Figure 6.5 illustrates the calculated $p_{RM,2}^{PLL,PT}$ metric, showing the percentage of the MCS runs (1,000 runs in total), for which at least one cycle slip was observed on the SGPLL, in the presence of a motion scenario (ID:2) encompassing walking 30m in a straight line having the sensor in the pocket. Figure 6.6

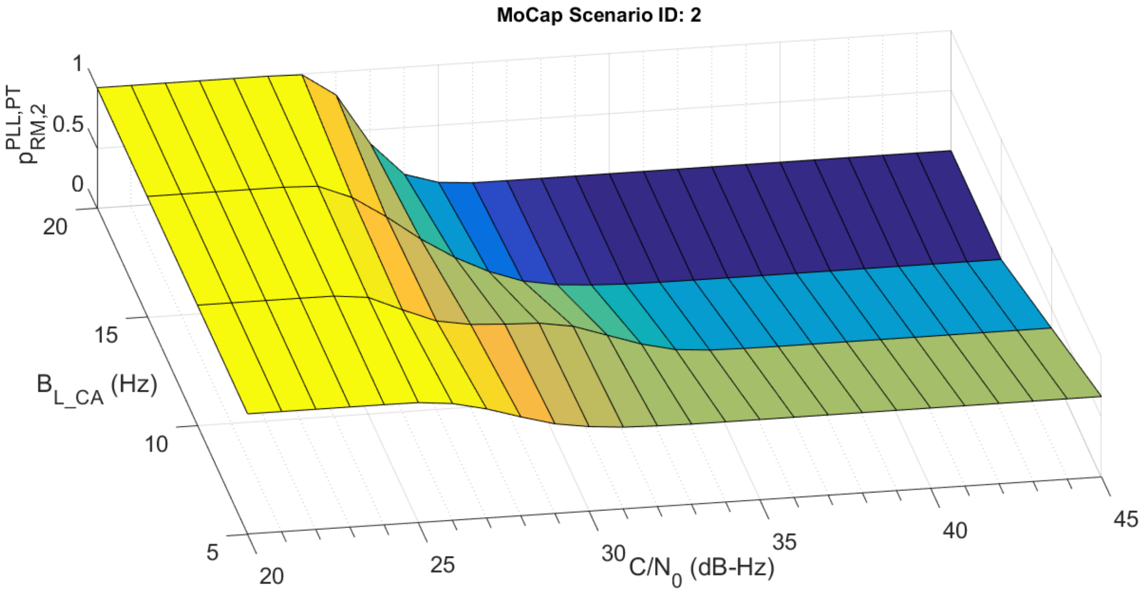

Figure 6.5. Percentage of real motion-induced cycle slips on SGPLL (per MCS run – 1,000 in total), for a real motion profile (ID: 2) encompassing walking 30m in a straight line having the sensor in the pocket.

Figure 6.6 illustrates the calculated $p_{RM,2}^{FLL,FT}$ metric, showing the percentage of the MCS runs (1,000 runs in total), for which at least one false frequency lock was observed on the SGFLL, in the presence of a motion scenario (ID: 2) encompassing walking 30m in a straight line having the sensor in the pocket.

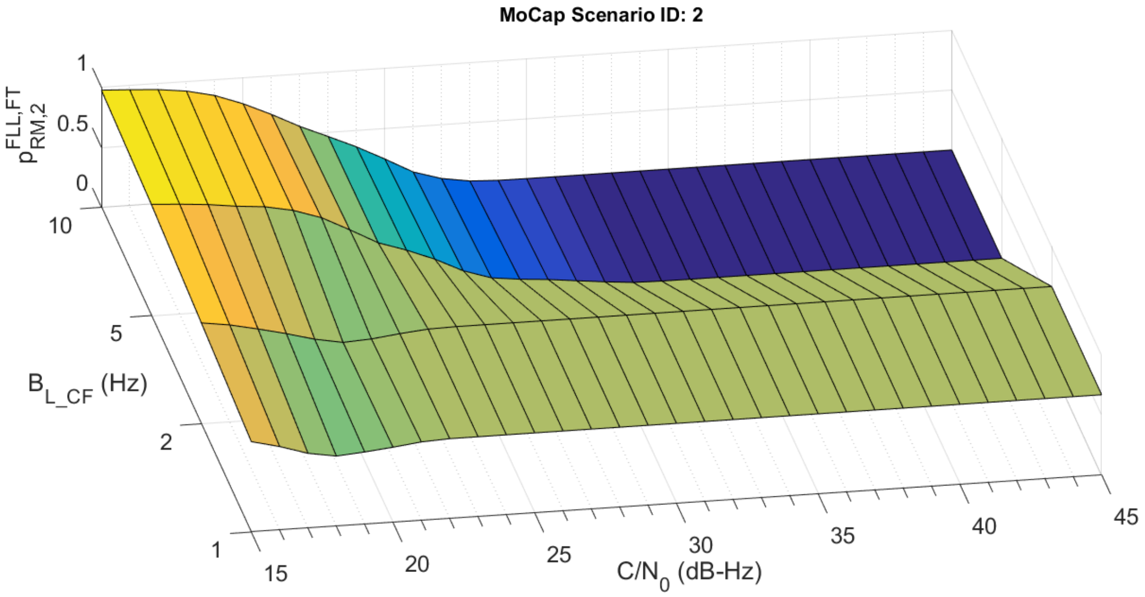

Figure 6.6. Percentage of real motion-induced false frequency locks on SGFLL (per MCS run – 1,000 in total), for a real motion profile (ID: 2) encompassing walking 30m in a straight line having the sensor in the pocket.

The results between *p* and *q* metrics are very similar, as shown in Figure 6.3 and Figure 6.4, as well as Figure 6.5 and Figure 6.6. It is worth noting that the *p* metrics are typically higher than the *q* metrics, as Figure 6.5 and Figure 6.6 illustrate. This is due to the fact that even one tracking slip over the whole motion profile can cause the whole MCS run to be characterised as affected by tracking slips, and cause function $f$ in Eq. (6.9) and $g$ in Eq. (6.11) to evaluate to unity, instead of zero. An extreme example would be to observe one tracking slip on each MCS run (1,000 runs in total), so the *p* metric would be unity (i.e. 100% of the 1,000 MCS runs are affected by at least one tracking slip), but the relative frequency, calculated via Eq. (6.2), would be divided by the number of epochs (minus one), multiplied by the number of runs, so e.g. for a motion profile of 2,001 epochs the *q* metric would be 1,000/(1,000x2000) = 0.05%. However, such extreme cases have not been observed in the analysed MoCap data.

The difference of *p* metrics between the real (or synthetic) motion and the static test case, will then be:

$$\begin{cases} \Delta p_{RM,i/S}^{PLL,PT} = p_{RM,i}^{PLL,PT} - p_{S}^{PLL,PT} \\ \Delta p_{PMM,i/S}^{PLL,PT} = p_{PMM,i}^{PLL,PT} - p_{S}^{PLL,PT} \end{cases},$$
$$\begin{cases} \Delta p_{RM,i/S}^{FLL,FT} = p_{RM,i}^{FLL,FT} - p_{S}^{FLL,FT} \\ \Delta p_{PMM,i/S}^{FLL,FT} = p_{PMM,i}^{FLL,FT} - p_{S}^{FLL,FT} \end{cases}. \tag{6.13}$$

The chapter continues with the presentation of the effects of the real motion on the SGCTLs, employing the methods and metrics discussed in this section.

## 6.2 Real Motion Effect on GNSS Carrier-tracking

### 6.2.1 Real motion effect on SGPLL phase-tracking

The SGPLL phase-tracking $M_{RM,i/S}^{PLL,PT}$ metrics for all MoCap scenarios, are presented in Table 6.1, between the third and the 6th columns, for each of the simulated effective bandwidths (5Hz, 10Hz, 15Hz and 20Hz, respectively). These represent the significance of the real motion-induced dynamic stress on the SGPLL. The first column provides a brief description of the specific MoCap profile and the second column the MoCap scenario ID (representing the *i* subscript of the metrics' notation). This indexing is the same as the one used in Table 3.2, which describes in more detail the encompassed motion in the MoCap scenarios. The "Sum" (in 7th) column shows the sum of the metrics for individual SGPLL effective bandwidths, in descending order, i.e. the motion scenarios inducing the most significant dynamics stress on the SGPLL, appear on the top of the table. Each individual metric per bandwidth value, is the area between the respective (solid coloured) lines and the threshold (dotted black) line in Figure 6.7. The "Sum" column is the total sum of these metrics/areas showing the performance of the SGPLL across different bandwidths for that specific input MoCap scenario. The number in the brackets (in the "Sum" column) shows the percentile rank of the sum of metrics. The asterisk in the MoCap scenario ID column denotes which MoCap scenarios have been simulated by the PMM, with their results presented in Section 6.3.

The results in Table 6.1 show that motion profiles encompassing walking with U-turns and gestures ("answer the phone" and "send a text/email") are above the 75th percentile, with the top two motion profiles (above the 98th percentile) encompassing a "send a text/email" gesture. Also, above the 75th percentile is the MoCap scenario representing an "answer the phone" gesture for walking without U-turns. The motion profiles with the least significant impact on the SGPLL (lower 45th percentile) include walking with the sensor held by hand, or having the sensor inside a back-pack or attached on the upper arm with an arm-band, with the latter two scenarios being on the lower 11th percentile.

Figure 6.7 illustrates the results summarised in Table 6.1. The horizontal axis shows the simulated $C/N_0$ (dB-Hz) values, and the coloured lines the different SGPLL effective bandwidths. The coloured lines are truncated for lower $C/N_0$ values, following the analysis in Section 4.4.1 and the minimum $C/N_0$ thresholds from Table 4.1, which shows the level at which the tracking noise would induce significant stress on the SGPLL, in other words, the real motion-induced cycle slips would be shadowed by the dominant tracking noise below those minimum $C/N_0$ thresholds. The vertical axis represents the significance of the real motion-induced dynamic stress on the SGPLL, in terms of the ratio between the $\Delta q_{RM,i/S}^{PLL,PT}$ metrics and the respective standard deviation (SD), i.e. $\sigma\left(\Delta q_{RM,i/S}^{PLL,PT}\right)$. The black dotted line in the subplots of Figure 6.7 represents the 50% confidence level (CL), i.e. when the $\Delta q_{RM,i/S}^{PLL,PT}$ is higher than at 0.6745 SDs, following Eq. (6.7).

The results in Figure 6.7 show that all MoCap scenarios had a significant impact on the SGPLL (above 3SDs, i.e. 99.7% CL) for a 5Hz effective bandwidth, except for walking with the sensor held by hand (MoCap ID:1) above a $C/N_0$ of 38dB-Hz and walking with the sensor inside a back-pack (MoCap ID: 3) or attached on the upper arm with an arm-band (MoCap ID: 4). For a 10Hz effective bandwidth, the real motion profiles did not introduce significant dynamic stress (above the 50% CL threshold – denoted with the black dotted line), except for walking with the sensor inside the pocket (MoCap ID: 2), jogging with the sensor held by hand (MoCap ID: 11) or inside a pocket (MoCap ID: 12). In addition, real motion profiles which introduced significant dynamic stress on the SGPLL (above a 50% CL), encompass walking with the sensor inside a pocket and perform an "answer the phone gesture" (MoCap ID: 8) below a $C/N_0$ of 32dB-Hz, or a "send a text/email" gesture (MoCap ID: 10), at a 31dB-Hz $C/N_0$; and also, walking and performing a U-turn with the sensor in the pocket, without gestures (MoCap ID: 14) for a $C/N_0$ below 35dB-Hz, or with a gesture, "send a text/email" (MoCap ID:17) or "answer the phone" (MoCap ID: 18), both for a $C/N_0$ below 30dB-Hz.

For higher effective bandwidths, i.e. 15Hz and 20Hz, the results in Figure 6.7 show that the real motion did not have a significant impact on the SGPLL, as the ratio in the vertical axis was below the 50% CL line for all MoCap scenarios for all applicable $C/N_0$ values, with the 20Hz bandwidth results below a 1% CL threshold (not illustrated as a line in Figure 6.7). The maximum real motion induced dynamic stress significance levels for a 15Hz effective bandwidth (below the 50% CL), are observed at 30dB-Hz $C/N_0$, for MoCap scenarios encompassing walking with the sensor inside the pocket and performing a U-turn (MoCap ID: 14), with a ratio of 0.35 (27.3% CL), the same motion profile without a U-turn (MoCap ID: 2) with a ratio of 0.27 (24.4% CL), and also, jogging with the sensor held by hand (MoCap ID: 11) or inside the pocket (MoCap ID: 12), with ratios 0.32 (25.1% CL) and 0.345 (27% CL), respectively.

| *Brief MoCap Scenario Description* | *MoCap Scenario ID (i)* | *SGPLL Phase-tracking* $M^{PLL,PT}_{RM,i/S}$ | | | | *Sum of metrics and (percentile rank)* |
|---|---|---|---|---|---|---|
| | | *B*<sub></sub>$_{PLL}$ *(Hz)* | | | | |
| | | *5* | *10* | *15* | *20* | |
| Walk 15m; sensor in pocket, “send a text/email”; U-turn | 17 | 145.3 | 3.4 | 0.1 | 0 | 148.7 (100) |
| Walk 15m; sensor in hand, “send a text/email”; U-turn | 16* | 146.2 | 0.2 | 0 | 0 | 146.4 (98) |
| Walk 15m; sensor in pocket, “answer the phone”; U-turn | 18 | 137.2 | 6.2 | 0.1 | 0 | 143.5 (97) |
| Walk 15m; sensor in pocket, “answer the phone” | 8 | 136.1 | 5.5 | 0.1 | 0 | 141.6 (95) |
| Walk 15m; sensor in hand, “send a text/email” | 9* | 116.9 | 0.1 | 0 | 0 | 117 (79) |
| Walk 15m; sensor in hand, “answer the phone”; U-turn | 15* | 111 | 0.1 | 0 | 0 | 111.1 (75) |
| Jog; sensor in pocket | 12 | 88.7 | 18.4 | 0.6 | 0 | 107.7 (72) |
| Walk 10m; sensor in hand, “answer the phone” | 5* | 106.8 | 0.2 | 0 | 0 | 107 (72) |
| Walk; sensor in pocket; U-turn | 14 | 86.9 | 14.2 | 0.8 | 0 | 101.8 (68) |
| Walk 10m; sensor in pocket, “answer the phone” | 6 | 93 | 7.3 | 0.1 | 0 | 100.4 (68) |
| Walk; sensor in pocket | 2 | 79.6 | 18 | 0.5 | 0 | 98.1 (66) |
| Walk 15m; sensor in pocket, “send a text/email” | 10 | 87.8 | 6.5 | 0.1 | 0 | 94.4 (63) |
| Walk; sensor in hand; U-turn | 13* | 92.3 | 0 | 0 | 0 | 92.3 (62) |
| Walk 15m; sensor in hand, “answer the phone” | 7* | 89.7 | 0.4 | 0 | 0 | 90.2 (61) |
| Jog; sensor in hand | 11 | 72.7 | 16.5 | 0.6 | 0 | 89.7 (60) |
| Walk; sensor in hand | 1* | 66.6 | 0 | 0 | 0 | 66.6 (45) |
| Walk; sensor in back-pack | 3 | 16.2 | 0.5 | 0 | 0 | 16.7 (11) |
| Walk; sensor in arm-band | 4 | 12.6 | 0 | 0 | 0 | 12.6 (8) |

Table 6.1. Metrics for real motion scenarios’ effect on the SGPLL

In addition to the real motion-induced dynamic stress significance ratios’ results, Figure 6.8 shows the percentage of real motion-induced cycle slips (per MCS) on the SGPLL, i.e. the $\Delta p$ metrics defined in Eq. (6.9). The black dotted line shows the $\Delta p$ level where 50% of the MCS runs (1,000 in total) exhibit at least one cycle slip. The results show that for an effective bandwidth of 20Hz, the real motion scenarios did not introduce any significant dynamic stress on the SGPLL, with $\Delta p$<1%, for all $C/N_0$ values, noting that the same effect applies for a 15Hz effective bandwidth, for all $C/N_0$ values above 34dB-Hz. The maximum $\Delta p$ metrics for a 15Hz effective bandwidth (with $\Delta p$<50%, i.e. below the 50% threshold), are observed at 30dB-Hz $C/N_0$, for MoCap scenarios encompassing walking with the sensor inside the pocket and performing a U-turn (MoCap ID: 14), with $\Delta p$<31.5%, the same motion profile without a U-turn (MoCap ID: 2) with $\Delta p$=18.7%, and also, jogging with the sensor held by hand (MoCap ID: 11) or inside the pocket (MoCap ID: 12), with $\Delta p$ metrics at 25.2% and 25.4%, respectively.

For an effective bandwidth of 10Hz, the results in Figure 6.8 show that the $\Delta p$ metric is inversely proportional to the $C/N_0$, with most MoCap profiles not introducing dynamic stress on the SGPLL above $\Delta p$=60%, at a $C/N_0$ level of 29dB-Hz, except for motion profiles encompassing walking with the sensor inside the pocket (MoCap ID: 2) where $\Delta p$=67.1%; jogging with the sensor held by hand (MoCap ID: 11) or inside the pocket (MoCap ID: 12), with $\Delta p$ metrics at 67% and 63.1%, respectively; and also, walking with the sensor inside the pocket and performing a U-turn (MoCap ID: 14), with $\Delta p$=68.9%. Finally, for an effective bandwidth of 5Hz, the results in Figure 6.8 show that real motion introduced significant dynamic stress on the SGPLL, with $\Delta p$>*97*% for MoCap scenarios encompassing walking (without making a U-turn) holding the sensor and performing an "answer the phone" gesture (MoCap ID: 5); walking with the sensor in the pocket and performing an "answer the phone" gesture (MoCap ID: 8); walking holding the sensor and performing an "send a text/email" gesture (MoCap ID: 9); walking (making a U-turn) holding the sensor and performing an "send a text/email" gesture (MoCap ID: 16); walking (making a U-turn) having the sensor inside the pocket, and performing an "send a text/email" gesture (MoCap ID: 17), or an "answer the phone" gesture (MoCap ID: 18).

This section (6.2) continues with the presentation of the real motion effect on the simulated GNSS frequency lock loop (SGFLL). A summary of the results presented in this (sub)section, as well as recommendations deriving from these results, are discussed in Section 6.4.

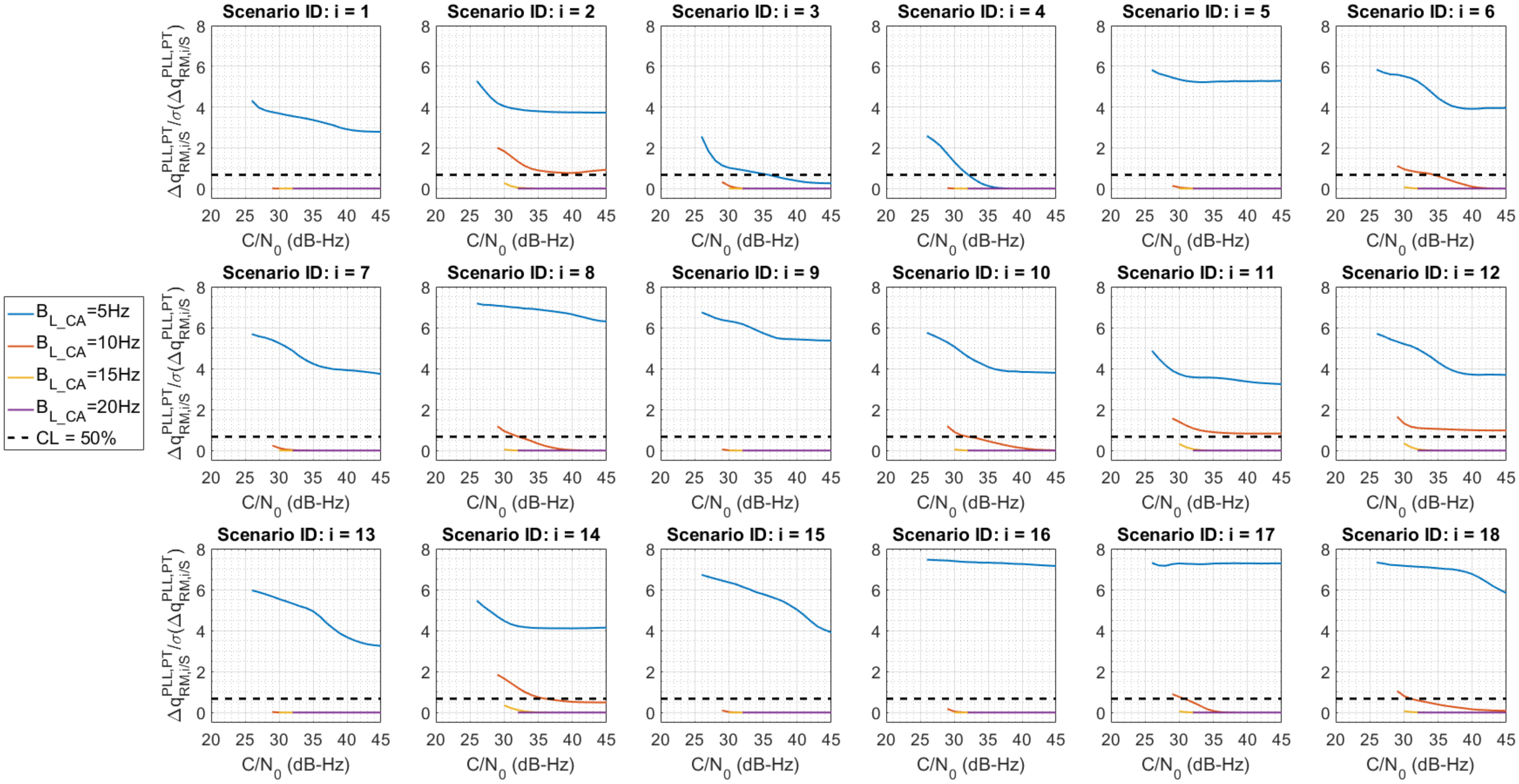

Figure 6.7. Real motion-induced dynamic stress significance on the SGPLL

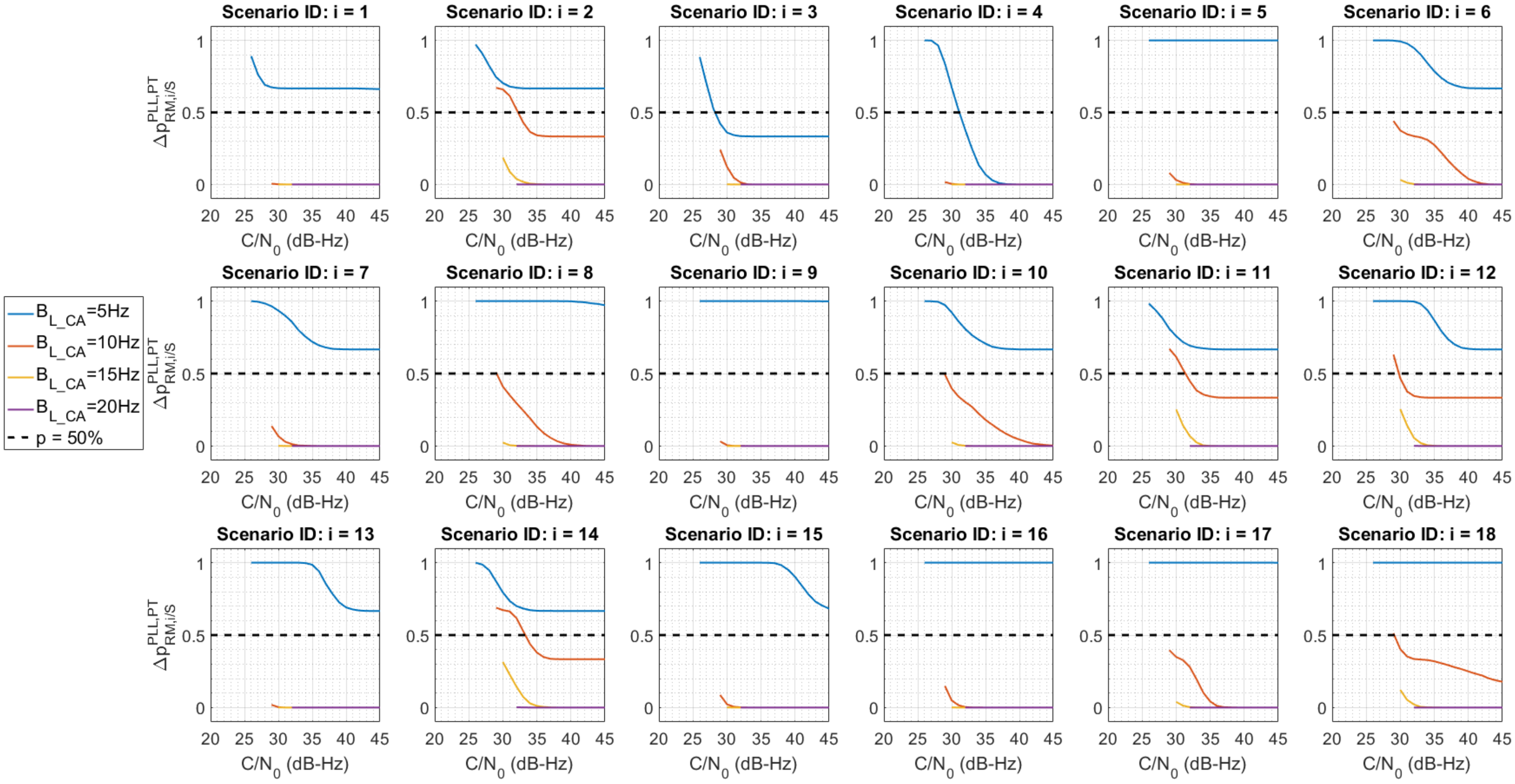

Figure 6.8. Percentage of real motion-induced cycle slips (per MCS) on the SGPLL

### 6.2.2 Real motion effect on SGFLL frequency-tracking

The SGFLL frequency-tracking $M_{RM,i/S}^{PLL,PT}$ metrics for all MoCap scenarios, are presented in Table 6.2, between the third and the 6th columns, for each of the simulated effective bandwidths (1Hz, 2Hz, 5Hz and 10Hz, respectively). These represent the significance of the real motion-induced dynamic stress on the SGFLL. The first column provides a brief description of the specific MoCap profile and the second column the MoCap scenario ID (representing the *i* subscript of the metrics' notation). This indexing is the same as the one used in Table 3.2, which describes in more detail the encompassed motion in the MoCap scenarios. The "Sum" (in 7th) column shows the sum of the metrics for individual SGFLL effective bandwidths, in descending order, i.e. the motion scenarios inducing the most significant dynamics stress on the SGFLL, appear on the top of the table. Each individual metric per bandwidth value, is the area between the respective (solid coloured) lines and the threshold (dotted black) line in Figure 6.9. The "Sum" column is the total sum of these metrics/areas showing the performance of the SGFLL across different bandwidths for that specific input MoCap scenario. The number in the brackets (in the "Sum" column) shows the percentile rank of the sum of metrics.

The results in Table 6.2 show that motion profiles which cause the most significant disruption due to dynamics stress on the SGFLL, above the 69th percentile, encompass walking with the sensor in the pocket and making a U-turn, without performing any gestures (100th percentile), or performing a "send a text/email" gesture (77th percentile) or an "answer the phone" gesture (69th percentile), as well as walking with the sensor held by hand and making a U-turn, without performing any gestures (95th percentile), or performing an "answer the phone" gesture (92nd percentile). The motion scenarios causing the least significant disruption due to dynamics stress on the SGFLL, below the 25th percentile, encompass walking holding the sensor by hand without performing any gestures (25th percentile) or performing an "answer the phone" gesture (17th percentile); walking having the sensor attached on the upper arm with an arm-band (16th percentile); walking with the sensor inside a pocket and performing an "answer the phone" gesture (13th percentile); walking having the sensor inside a back-pack (11th percentile); and jogging having the sensor inside the pocket (6th percentile) or held by hand (first percentile).

Figure 6.9 illustrates the results summarised in in Table 6.2. The horizontal axis shows the simulated $C/N_0$ (dB-Hz) values, and the coloured lines the different SGFLL effective bandwidths. The coloured lines are truncated for lower $C/N_0$ values, following the analysis in Section 4.4.1 and the minimum $C/N_0$ thresholds from Table 4.1, which shows the level at which the tracking noise would induce significant stress on the SGFLL, in other words, the real motion-induced false frequency locks would be shadowed by the dominant tracking noise below those minimum $C/N_0$

thresholds. The vertical axis represents the significance of the real motion-induced dynamic stress on the SGFLL, in terms of the ratio between the $\Delta q_{RM,i/S}^{FLL,FT}$ metrics and the respective standard deviation (SD), i.e. $\sigma\left(\Delta q_{RM,i/S}^{FLL,FT}\right)$. The black dotted line in the subplots of Figure 6.9 represents the 50% confidence level (CL), i.e. when the $\sigma\left(\Delta q_{RM,i/S}^{FLL,FT}\right)$ is higher than at 0.6745 SDs, following Eq. (6.7).

| *Brief MoCap Scenario Description* | *MoCap Scenario ID (i)* | *SGFLL Frequency-tracking* $M_{RM,i/S}^{FLL,FT}$ | | | | *Sum of metrics and (percentile rank)* |
|---|---|---|---|---|---|---|
| | | *B*<sub></sub>*FLL (Hz)* | | | | |
| | | *1* | *2* | *5* | *10* | |
| Walk; sensor in pocket; U-turn | 14 | 129.2 | 92.9 | 4.6 | 0.1 | 226.8 (100) |
| Walk; sensor in hand; U-turn | 13 | 126.7 | 86 | 1.7 | 0 | 214.5 (95) |
| Walk 15m; sensor in hand, “answer the phone”; U-turn | 15 | 121.5 | 84 | 2.9 | 0.1 | 208.4 (92) |
| Walk 15m; sensor in pocket, “send a text/email”; U-turn | 17 | 114.3 | 59 | 1.4 | 0 | 174.8 (77) |
| Walk 15m; sensor in pocket, “answer the phone”; U-turn | 18 | 93.3 | 60 | 2.9 | 0.1 | 156.3 (69) |
| Walk; sensor in pocket | 2 | 60.1 | 36.1 | 2.5 | 0.1 | 98.8 (44) |
| Walk 15m; sensor in hand, “send a text/email”; U-turn | 16 | 54.7 | 31.8 | 1.1 | 0.1 | 87.7 (39) |
| Walk 15m; sensor in hand, “send a text/email” | 9 | 49.8 | 35.2 | 2.4 | 0.1 | 87.4 (39) |
| Walk 10m; sensor in hand, “answer the phone” | 5 | 47.7 | 22.2 | 1.1 | 0 | 71 (31) |
| Walk 10m; sensor in pocket, “answer the phone” | 6 | 38.1 | 26.1 | 1.5 | 0.1 | 65.8 (29) |
| Walk 15m; sensor in pocket, “send a text/email” | 10 | 38.2 | 24.8 | 0.5 | 0.1 | 63.6 (28) |
| Walk; sensor in hand | 1 | 33.3 | 21.4 | 0.9 | 0.1 | 55.7 (25) |
| Walk 15m; sensor in hand, “answer the phone” | 7 | 23.9 | 15.3 | 0.4 | 0 | 39.5 (17) |
| Walk; sensor in arm-band | 4 | 22.3 | 13.6 | 0.9 | 0.1 | 36.9 (16) |
| Walk 15m; sensor in pocket, “answer the phone” | 8 | 19.6 | 10 | 0.8 | 0.1 | 30.4 (13) |
| Walk; sensor in back-pack | 3 | 19.9 | 4.3 | 0.2 | 0 | 24.4 (11) |
| Jog; sensor in pocket | 12 | 7 | 4.4 | 1.4 | 0.2 | 13 (6) |
| Jog; sensor in hand | 11 | 0 | 0.2 | 0.9 | 0.2 | 1.3 (1) |

Table 6.2. Metrics for real motion scenarios’ effect on the SGFLL

The results in Figure 6.9 show that when a 1Hz SGFLL bandwidth is employed, real motion cause significant disruption due to dynamic stress on the SGFLL, above a $\Delta q_{RM,i/S}^{FLL,FT} \big/ \sigma\left(\Delta q_{RM,i/S}^{FLL,FT}\right)$ ratio of 2 (i.e. 95% CL) for all $C/N_0$ values and MoCap scenarios, except for jogging with the sensor held by hand (MoCap ID: 11) or inside the pocket (MoCap ID: 12). More specifically the ratio is above 3 except for real motion scenarios encompassing walking with the sensor inside a back-pack (MoCap ID: 3) with a ratio of 2; walking with the sensor attached on the upper arm with an arm-band (MoCap ID: 4), with a ratio of 2.2; walking with the sensor held by hand and performing an "answer the phone" gesture (MoCap ID: 7) with a ratio of 2.4; and walking with the sensor inside the pocket and performing an "answer the phone" gesture (MoCap ID: 8) with a ratio of 2. For a 2Hz SGFLL effective bandwidth, all real motion profiles cause significant disruption due to dynamic stress on the SGFLL, above a $\Delta q_{RM,i/S}^{FLL,FT} \big/ \sigma\left(\Delta q_{RM,i/S}^{FLL,FT}\right)$ratio of 0.6745 (i.e. 50% CL), except for scenarios which include walking with the sensor inside a back-pack (MoCap ID: 3); jogging with the sensor held by hand (MoCap ID: 11) or inside the pocket (MoCap ID: 12); and walking with the sensor inside the pocket and performing an "answer the phone" gesture (MoCap ID: 8), for $C/N_0$ values up to 28dB-Hz.

For a 5Hz SGFLL bandwidth, the results show that the significance of the real motion disruption (due to dynamics stress) on the SGFLL, never exceeds the 0.6745 ratio (i.e. 50% CL, denoted with the black dotted line in in Figure 6.9), for all MoCap scenarios, except for walking with the sensor inside the pocket (MoCap ID: 2) at a 25dB-Hz $C/N_0$; walking holding the sensor and performing an "send a text/email" gesture (MoCap ID: 9) at a 25dB-Hz $C/N_0$; walking with the sensor held by hand and performing a U-turn (MoCap ID: 13) at a 25dB-Hz $C/N_0$; walking with the sensor inside the pocket and performing a U-turn (MoCap ID: 14) between 25dB-Hz and 27dB-Hz $C/N_0$; walking (making a U-turn) with the sensor held by hand, performing an "answer the phone" gesture (MoCap ID: 15) at a 25dB-Hz and a 26dB-Hz $C/N_0$; and walking (making a U-turn) having the sensor inside the pocket, and performing an "send a text/email" gesture (MoCap ID: 17) at a 25dB-Hz and a 26dB-Hz $C/N_0$. The significance ratio $\Delta q_{RM,i/S}^{FLL,FT} \big/ \sigma\left(\Delta q_{RM,i/S}^{FLL,FT}\right)$ of real motion disruption on the SGFLL for a 10Hz effective bandwidth, is below 0.6745 (i.e. 50% CL) for all MoCap scenarios tested with the SGFLL. The maximum value, below the 50% CL level, is observed for the motion profile encompassing jogging with the sensor inside the pocket (MoCap ID: 12), with a ratio of 0.1493 (i.e. 11.9% CL) at a 25dB-Hz $C/N_0$.

Further to the real motion-induced dynamic stress significance ratios' results, Figure 6.10 shows the percentage of real motion-induced false frequency locks (per MCS) on the SGFLL, i.e. the $\Delta p$ metrics defined in Eq. (6.9). The black dotted line shows the $\Delta p$ level where 50% of the MCS runs (1,000 in total) exhibit at least one false frequency lock. It must be noted that, when the (blue) line representing the 1Hz effective bandwidth is not visible in Figure 6.10, it is hidden behind the (red) line representing the 2Hz bandwidth. The results show that the MoCap scenarios do not cause disruption to the SGFLL due to dynamic stress, at a ratio $\Delta q_{RM,i/S}^{FLL,FT} \big/ \sigma\left(\Delta q_{RM,i/S}^{FLL,FT}\right)$ above 0.05 (i.e. 4% CL), above a 31dB-Hz $C/N_0$ for an effective bandwidth of 5Hz, and for all $C/N_0$ values when a 10Hz SGFLL bandwidth is simulated.

The chapter continues with detailing the synthetic motion effect on the simulated GNSS phase lock loop (SGPLL). A summary of the results presented in this (sub)section, as well as recommendations deriving from these results, are discussed in Section 6.4.

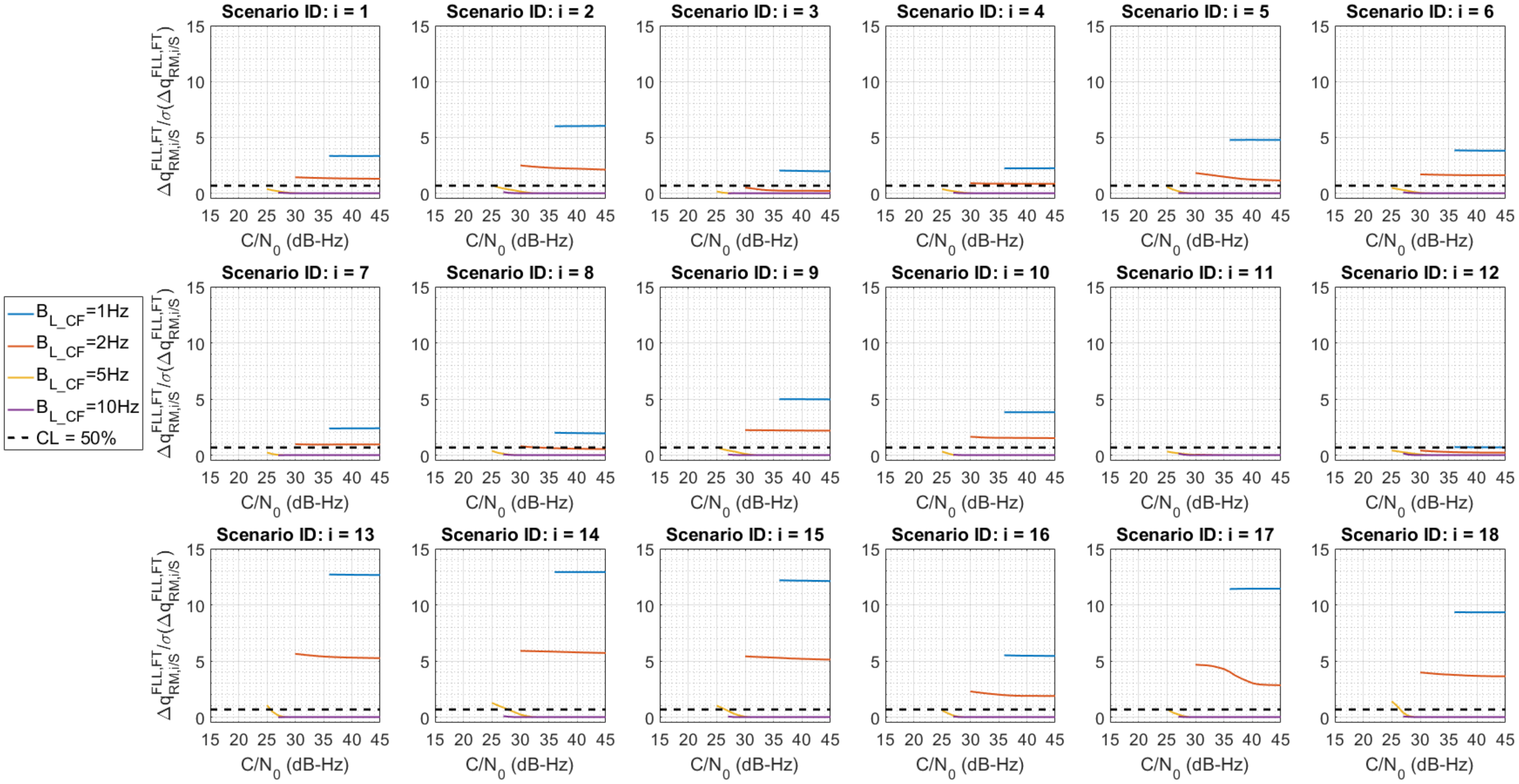

Figure 6.9. Real motion-induced dynamic stress significance on the SGFLL

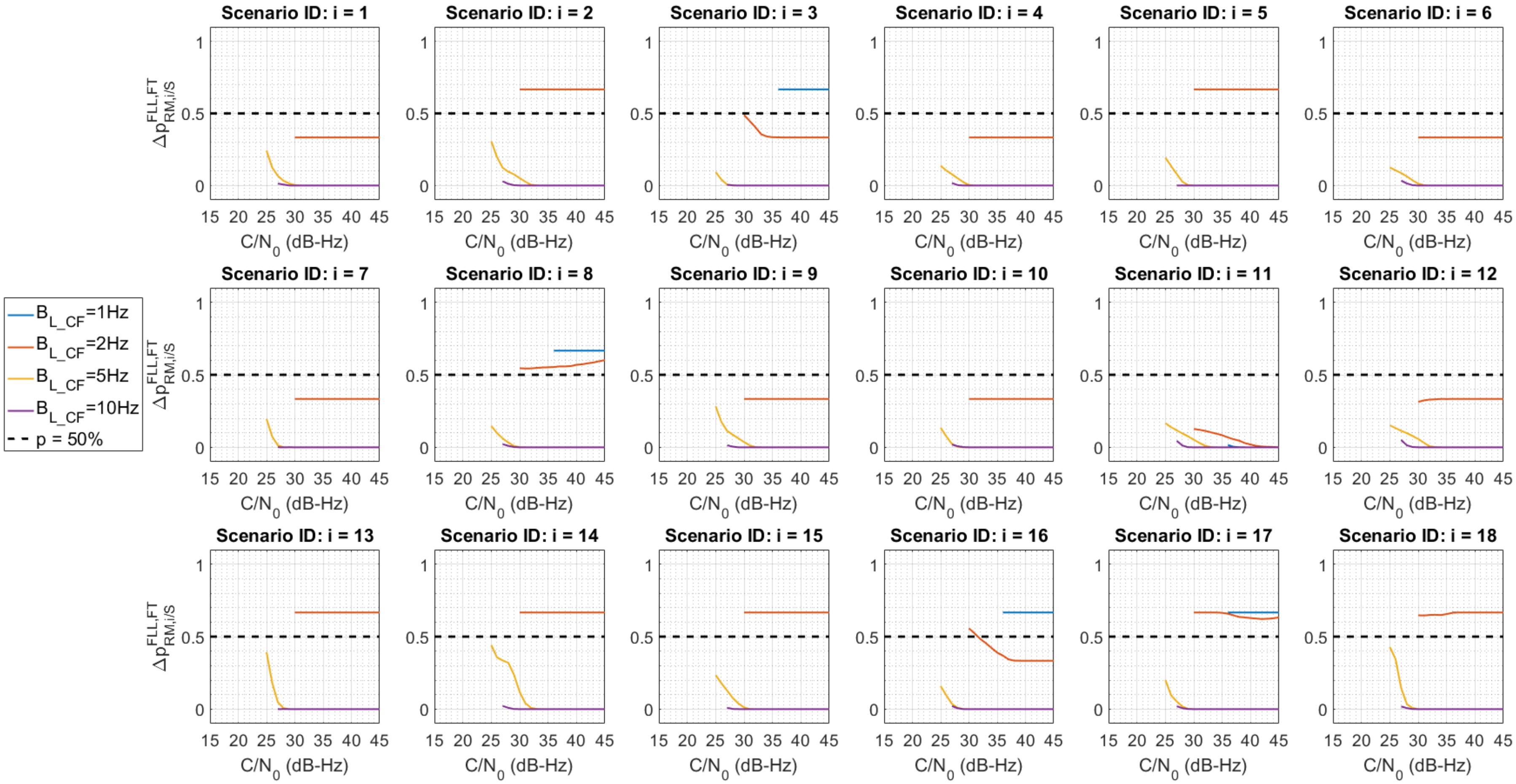

Figure 6.10. Percentage of real motion-induced false frequency locks (per MCS) on the SGFLL

## 6.3 Synthetic Motion Effect on GNSS Carrier Phase-tracking

This section presents the synthetic motion effect on the simulated GNSS carrier-tracking loop (SGPLL), and compares this to the real motion in order to validate, or not, the pedestrian motion model (described in Chapter 5). The analysis of this section addresses the second research question of the thesis, discussed in Section 1.2, i.e. which is the best way to generate a pedestrian motion model for simulation testing of GNSS equipment.

Table 6.3 shows the simulated MoCap scenarios by the PMM, with a brief description of the encompassed motion and their ID (i notation), in the first and second columns, respectively. The third up to the 6th columns show the metrics of the synthetic motion $M_{PMM,i/S}^{PLL,PT}$ (for each SGPLL bandwidth), calculated using Eq. (6.8) in Section 6.1.2, noting that the real motion metrics are shown in Table 6.1 of Section 6.2.1. The last column shows the ratio of the synthetic motion metrics' sum over the real motion metrics' sum (noting that for the real motion the metrics' sum is shown in the last column of Table 6.1 in Section 6.2.1). A sum's ratio of one means that the synthetic motion produced the same metrics' sum as the real motion. A ratio below one means that the synthetic motion metrics' sum is less than the real motion one, i.e. the synthetic motion underrepresents the effect of real motion on the SGPLL (for all effective bandwidths combined under this single metric). It follows that if the sum's ratio is greater than one, the synthetic motion exaggerates the effect of the synthetic motion on the SGPLL, compared to the real motion effect. The results in Table 6.3 show that the ratio is between 0.81 and 1.54, i.e. the synthetic motion output neither underrepresents nor exaggerates the real motion effect on the SGFLL, beyond these limits (which are less than a factor of two), for the simulated MoCap scenarios.

To produce the results in Table 6.3, the PMM output, i.e. the position and velocity of the sensor body-frame projected along the LOS between three simulated static satellites, has been smoothed using the central difference method, following Eq. (5.1) explained in Section 5.2. The smoothing effect is illustrated in Figure 6.11 for the height solution of the MoCap scenario encompassing walking for 15m and performing an "answer the phone" gesture (MoCap ID: 15). The blue and orange lines in Figure 6.11 illustrate the height solution before and after applying the central difference method 500 times. The height of the simulated pedestrian is set to 1.8m, following the human biomechanical model definition, described in Section 5.3.

| *Brief MoCap Scenario Description* | *MoCap Scenario ID (i)* | *SGPLL Phase-tracking* $M_{PMM,i/S}^{PLL,PT}$ | | | | *Synthetic motion metrics' Sum over the Real motion metrics' Sum ratio* |
|---|---|---|---|---|---|---|
| | | *B*<sub></sub>$_{PLL}$ *(Hz)* | | | | |
| | | *5* | *10* | *15* | *20* | |
| Walk; sensor in hand | 1 | 69.5 | 23.1 | 1 | 0 | 1.41 |
| Walk 10m; sensor in hand, "answer the phone" | 5 | 140.2 | 22.8 | 1.6 | 0 | 1.54 |
| Walk 15m; sensor in hand, "answer the phone" | 7 | 130.2 | 23 | 1.4 | 0 | 1.71 |
| Walk 15m; sensor in hand, "send a text/email" | 9 | 84.1 | 23.2 | 1.2 | 0 | 0.93 |
| Walk; sensor in hand; U-turn | 13 | 78.9 | 22.2 | 1 | 0 | 1.11 |
| Walk 15m; sensor in hand, "answer the phone"; U-turn | 15 | 143 | 21.9 | 1.2 | 0 | 1.50 |
| Walk 15m; sensor in hand, "send a text/email"; U-turn | 16 | 95.3 | 22.2 | 1.2 | 0 | 0.81 |

Table 6.3. Difference and ratio of the real and synthetic (smoothed) motion scenarios' metrics, representing their effect on the SGPLL

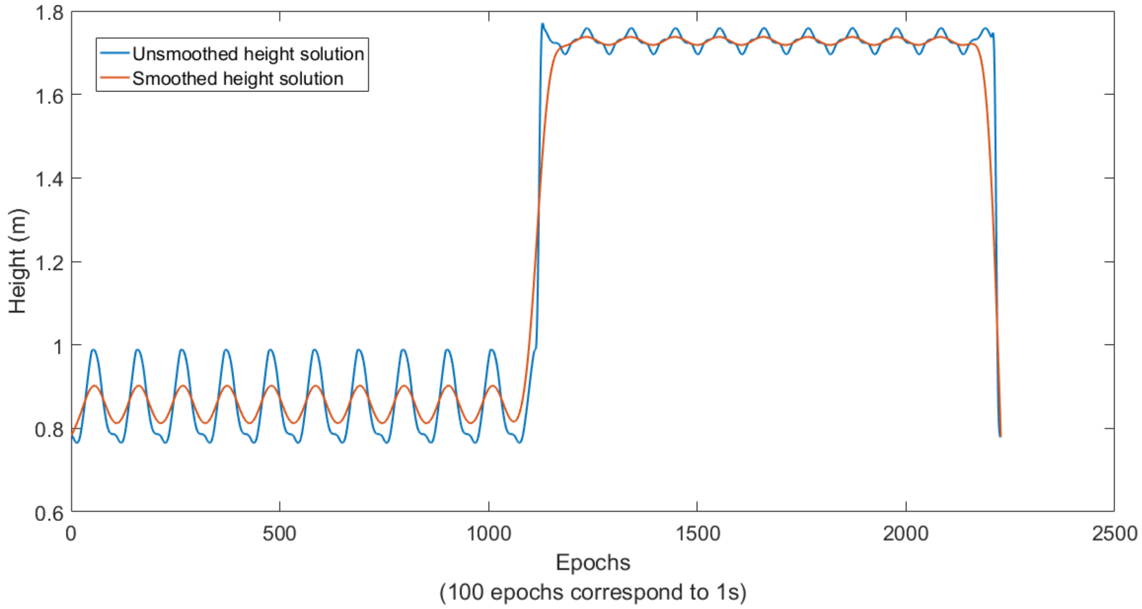

Figure 6.11. Effect of smoothing on PMM position solution output, using the central difference method (applied 500 times).

Figure 6.12 illustrates the results summarised in Table 6.3. The horizontal axis shows the simulated $C/N_0$ (dB-Hz) values, and the coloured lines the different SGPLL effective bandwidths. The coloured lines are truncated for lower $C/N_0$ values, following the analysis in Section 4.4.1 and the minimum $C/N_0$ thresholds from Table 4.1, which shows the level at which the tracking noise would induce significant stress on the SGPLL, in other words, the synthetic motion-induced cycle slips might be shadowed by the dominant tracking noise below those minimum $C/N_0$ thresholds. The vertical axis represents the significance of the synthetic motion-induced dynamic stress on the SGPLL, in terms of the ratio between the $\Delta q_{PMM,i/S}^{PLL,PT}$ metrics and the respective standard deviation (SD), i.e. $\sigma\left(\Delta q_{PMM,i/S}^{PLL,PT}\right)$. The black dotted line in the subplots of Figure 6.12, represents the 50% confidence level (CL), i.e. when the $\Delta q_{PMM,i/S}^{PLL,PT}$ is higher than at 0.6745 SDs, following Eq. (6.7). Also, Figure 6.13 (following the same illustration conventions as Figure 6.12), shows the $\Delta q_{RM,i/S}^{PLL,PT} \big/ \sigma\left(\Delta q_{RM,i/S}^{PLL,PT}\right)$ ratio for the real MoCap scenarios (i.e. only those simulated by the PMM) to facilitate the comparison between the synthetic and real motion results.

The comparison between the synthetic (Figure 6.12) and real motion (Figure 6.13) significance metric $\Delta q/\sigma\left(\Delta q\right)$, shows that the PMM output creates a similar effect to the real motion on the SGPLL when a 5Hz bandwidth is employed, by inducing significant dynamic stress to the SGPLL for all simulated scenarios and $C/N_0$ values (similar to the real motion effect), above a significance ratio of 2.75 (i.e. 99.4% CL). For a 10Hz effective SGPLL bandwidth, the synthetic motion exaggerated the real motion profiles' effect on the SGPLL, by increasing the significance ratio up to 1.4 (i.e. 83.4% CL) compared to the real motion, for the applicable simulated scenarios and $C/N_0$ values.

The results also show that for a 15Hz SGPLL bandwidth, the synthetic motion induces the same effect on the SGPLL as the real motion, i.e. it only increases the significance ratio by 0.05, i.e. a 4% CL, above a 37dB-Hz $C/N_0$, which means that there is a 4% confidence level that the synthetic motion induced a significant dynamic stress (in addition to the one induced by the real motion). Finally, the comparison between the effect of the real and synthetic motion on the SGPLL for a 20Hz effective bandwidth, shows that the synthetic motion induces an additional dynamic stress on the SGPLL of a level (ratio) below 0.02, i.e. there is 1.6% confidence that the synthetic motion produced a significant effect on the SGPLL, in addition to the real motion effect, for all simulated $C/N_0$ values.

In addition to the real motion-induced dynamic stress significance ratios' results, Figure 6.14 shows the percentage of real motion-induced cycle slips (per MCS) on the SGPLL, i.e. the $\Delta p$ metrics defined in Eq. (6.9), noting that to facilitate comparison with the real motion $\Delta p$ values, the latter are illustrated in Figure 6.15. The black dotted line shows the $\Delta p$ level where 50% of the MCS runs (1,000 in total) exhibit at least one cycle slip.

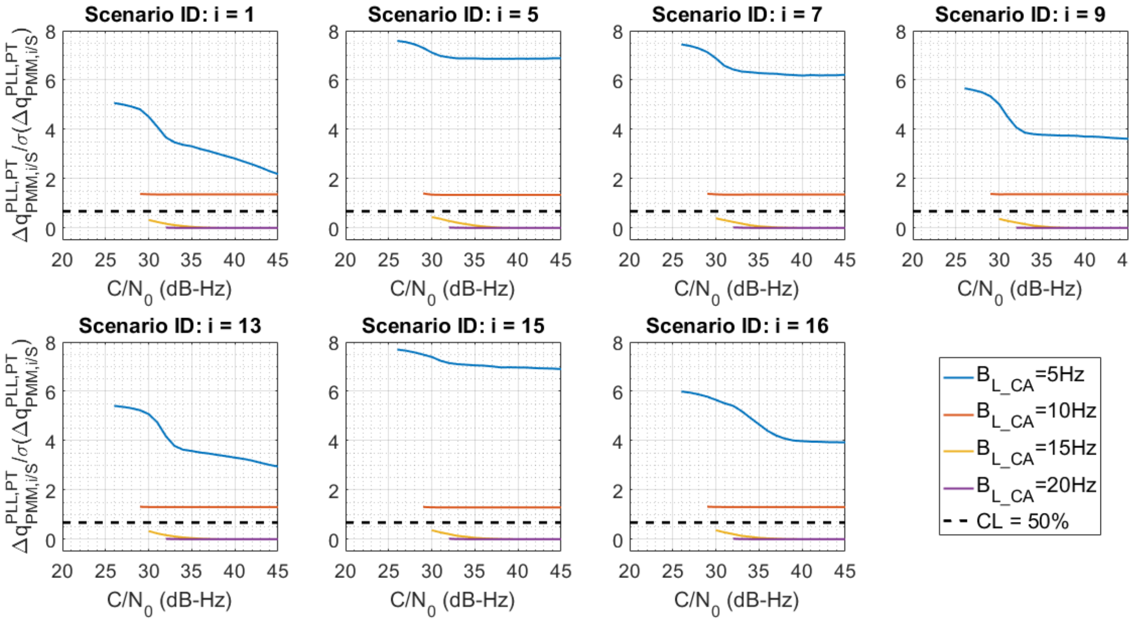

Figure 6.12. Synthetic motion-induced dynamics stress significance on SGPLL (for LOS motion smoothed 500 times, using the central difference method)

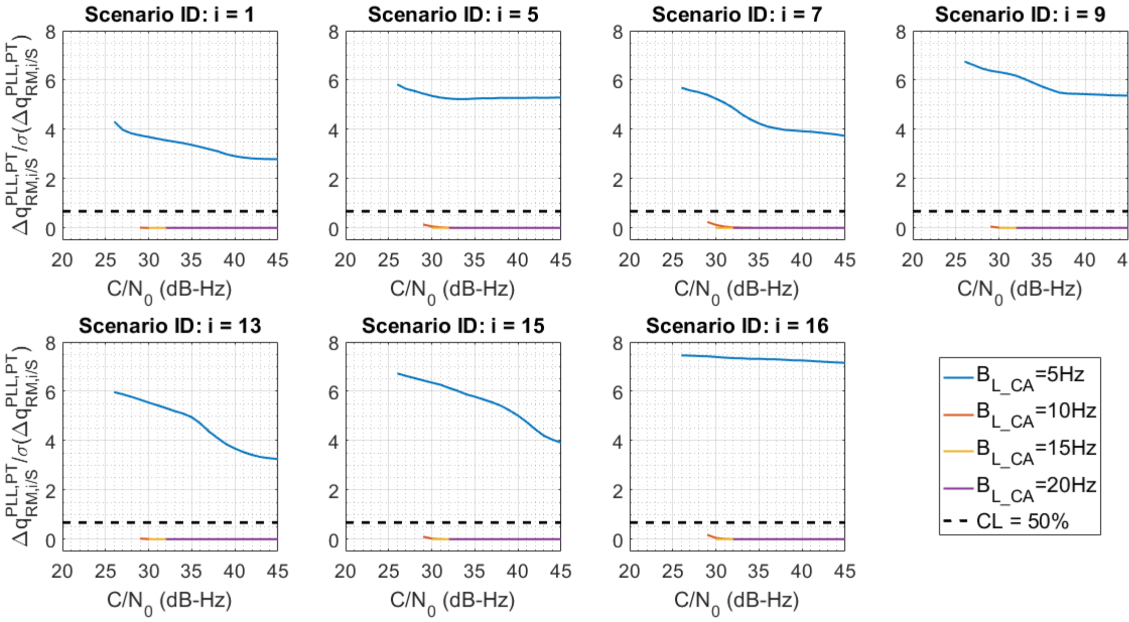

Figure 6.13. Real motion-induced dynamics stress significance on SGPLL (shown for comparison against the synthetic motion results)

The results show that for an effective bandwidth of 5Hz, the (smoothed) synthetic motion has a similar effect to the real motion on the SGPLL, for the simulated motion profile encompassing walking with the sensor held by hand and performing an "answer the phone" gesture, then performing a "put the phone down" gesture and stop (MoCap ID: 5). For a simulated motion profile encompassing walking holding the sensor by hand, the synthetic motion effect ($\Delta p$ value) is 0.11

higher (i.e. 8.8% CL) than the real motion $\Delta p$ value at a 29dB-Hz $C/N_0$ and -0.23 (i.e. 18.5% CL) at a 45dB-Hz $C/N_0$, i.e. there is a 18.5% confidence that the synthetic motion will underrepresent the real motion effect on the SGPLL at a 45dB-Hz $C/N_0$. A simulated motion profile encompassing walking holding the sensor by hand and performing an "answer the phone gesture" (MoCap ID: 7), the synthetic motion $\Delta p$ value is the same as the real motion one at a 29dB-Hz $C/N_0$, but after this point, it exaggerates the effect of the real motion on the SGPLL, up to a $\Delta p$ difference of 0.33 (25.9% CL) at a 45dB-Hz $C/N_0$. The simulated motion profile which includes walking holding the sensor by hand and then performing a "send a text/email" gesture, shows small difference (less than 0.01 in $\Delta p$ values) to the real motion $\Delta p$ value up to a 29dB-Hz $C/N_0$, but after that point it underrepresents the real motion effect up to a 0.37 difference in $\Delta p$ values (28.6% CL) at a 45dB-Hz $C/N_0$.

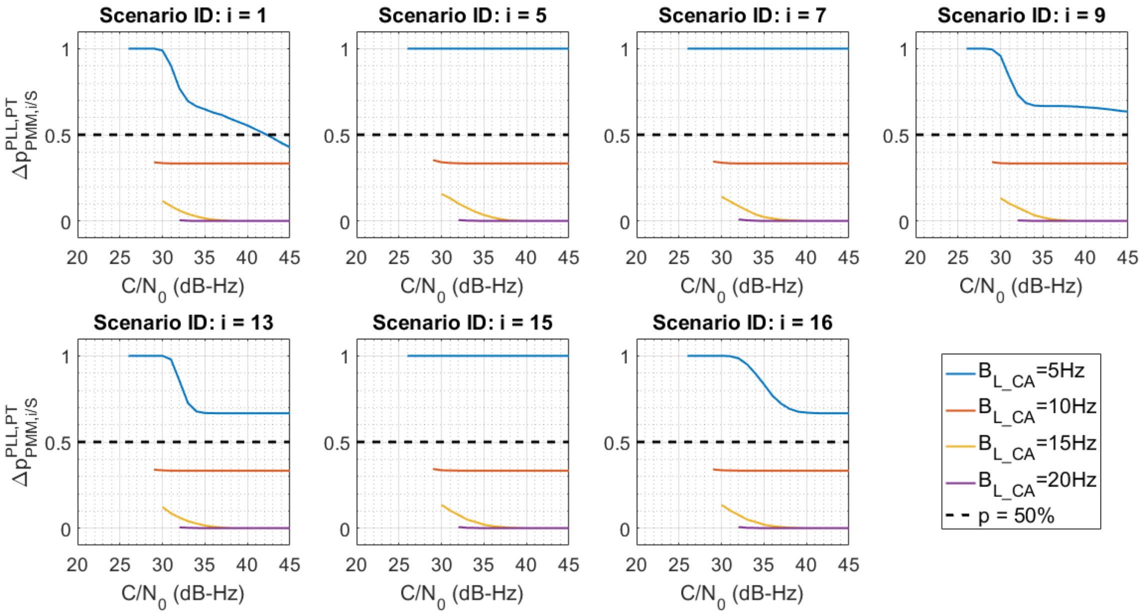

Figure 6.14. Percentage of synthetic motion-induced cycle slips (per MCS) on the SGPLL

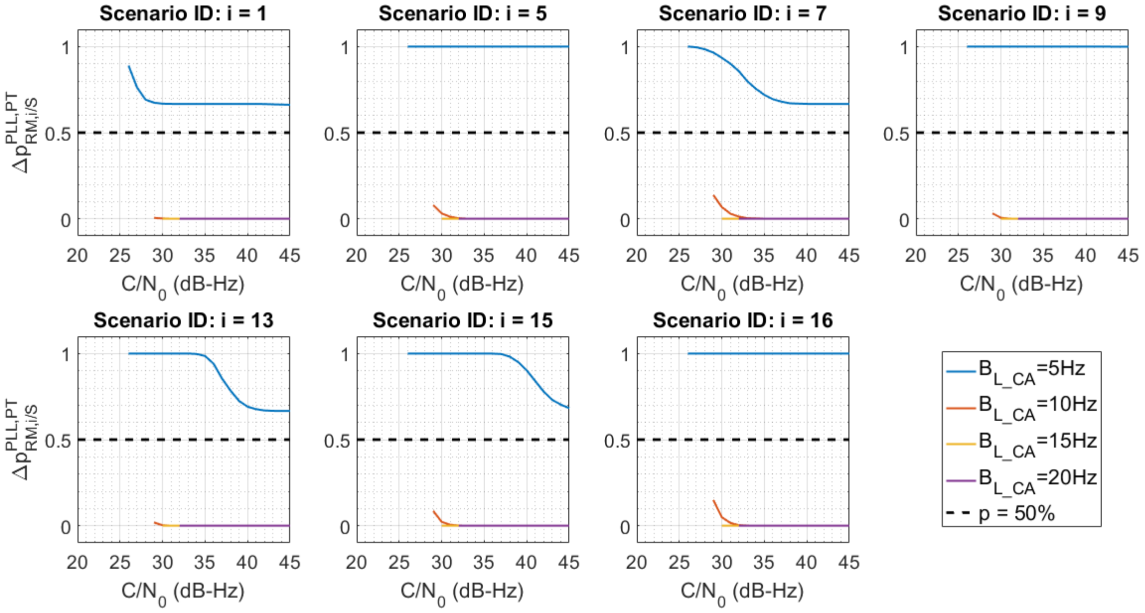

Figure 6.15. Percentage of real motion-induced cycle slips (per MCS) on the SGPLL (shown for comparison against the synthetic motion results)

Furthermore, the simulated motion profile encompassing walking holding the sensor by hand and making a U-turn (MoCap ID: 13), represents accurately (with a *Δp* difference less than 0.02 – i.e. a 1.7% CL) the real motion effect on the SGPLL, up to a 31dB-Hz C/N0, as well as between a 40dB-Hz and a 45dB-Hz C/N0, but for the in-between C/N0 values it underrepresents the real motion effect up to -0.32 (25.9% CL) at a 34dB-Hz C/N0. The simulated MoCap scenario including walking (making a U-turn) with the sensor held by hand, performing an "answer the phone" and a "put the phone down" gesture (MoCap ID: 15), shows that the synthetic motion represents accurately (within a 0.02 *Δp* difference compared to the real motion, i.e. a 1.7% CL), up to a 38dB-Hz C/N0, but above that C/N0 level, it exaggerates the real motion effect on the SGPLL, up to a 0.318 *Δp* difference (i.e. 24.9% CL), for a 34dB-Hz C/N0. Finally, the MoCap scenario representing walking (making a U-turn) holding the sensor, performing an "send a text/email" gesture and a "put the phone down" gesture (MoCap ID: 16), shows that the synthetic motion creates the same effect on the SGPLL as the real motion, i.e. with a less than 0.02 *Δp* difference, up to a 324dB-Hz C/N0, but above that point, it underrepresents the real motion effect on the SGPLL, up to a 0.33 *Δp* difference (i.e. 25.9% CL) at a 38dB-Hz C/N0.

For a 10Hz SGPLL effective bandwidth, the results of the synthetic motion in Figure 6.14 (compared to the real motion shown in Figure 6.15), show that the PMM output exaggerates the effect of the real motion on the SGPLL, between a 0.2 (15.5% CL) and a 0.3 (23.6% CL) *Δp* difference, for all simulated $C/N_0$ levels. The 15Hz SGPLL bandwidth results show that the synthetic motion exaggerates the effect on the SGPLL, with a maximum 0.157 *Δp* difference (i.e. a 12.5% CL) up to a 35dB-Hz $C/N_0$. Above that $C/N_0$ level, the synthetic motion represents accurately the effect of real motion on the SGPLL, with a maximum $\Delta p$ difference of 0.034 (i.e. 2.7% CL). Finally, the 20Hz employed SGPLL bandwidth, shows a good matching between the synthetic and real motion effect on the SGPLL, within a $\Delta p$ difference less than 0.01 (i.e. less than 1% CL), for all simulated $C/N_0$ values.

The chapter continues with Section 6.4 which discusses the presented results in Section 6.2 (real motion effect on the SGCTLs) and Section 6.3 (synthetic motion effect on the SGPLL and comparison with the real motion effect), and concludes with some recommendations for GNSS manufacturers, based on the evidence presented in these sections.

## 6.4 Summary of Results and Recommendations

This section starts with a summary of the results presented in Section 6.2 and Section 6.3, and continues with some recommendations for GNSS receiver design, based on the presented evidence. The results show the performance of the simulated GNSS tracking loops (SGCTLs) in the presence of real and synthetic human motion, as a real receiver testing was beyond the timescales of this thesis. However, investigating the performance of a real receiver against these motion profiles is an essential part of future work which will precede the productisation of the thesis' outcomes, as explained in Chapter 8. This future work will also add more confidence on the recommendations for GNSS design (see Section 6.4.2), noting that the testing will have to focus on GNSS receivers which can output relevant carrier tracking measurements and cannot be exhaustive considering all the different GNSS receivers/designs which are commercially available.

### 6.4.1 Summary of results

The results presented in Section 6.1 show that in terms of real motion effect on the simulated GNSS phase lock loop (SGPLL), motion profiles encompassing walking with U-turns and gestures ("answer the phone" and "send a text/email") are above the 75th percentile (see Table 6.1), with the top two motion profiles (above the 98th percentile) encompassing a "send a text/email" gesture. Also, above the 75th percentile is the MoCap scenario representing an "answer the phone" gesture for walking without U-turns. The motion profiles with the least significant impact on the SGPLL (lower 45th percentile) include walking with the sensor held by hand, or having the sensor inside a back-pack or attached on the upper arm with an arm-band, with the latter two scenarios being on the lower 11th percentile.

The results in Figure 6.7, which illustrate the significance of the dynamic stress induced by the real motion profiles on the SGPLL, show that for a 5Hz effective SGPLL bandwidth, all MoCap profiles cause a disruption due to dynamic stress on the SGPLL above a ratio of 3 (i.e. with a 99.7% confidence level – CL), except for when the motion profile encompasses walking with the sensor held by hand (MoCap ID: 1), above a 38dB-Hz $C/N_0$; walking with the sensor inside a back-pack (MoCap ID: 3) or attached by an arm-band on the upper arm (MoCap ID: 4), for all $C/N_0$ values. For a 10Hz effective bandwidth, the real motion profiles did not introduce significant dynamic stress (above a 50% CL), except for walking with the sensor inside the pocket (MoCap ID: 2), jogging with the sensor held by hand (MoCap ID: 11) or inside a pocket (MoCap ID: 12); walking with the sensor inside a pocket and perform an "answer the phone gesture" (MoCap ID: 8) below a $C/N_0$ of 32dB-Hz, or a "send a text/email" gesture (MoCap ID: 10), at a 31dB-Hz $C/N_0$; and also, walking and performing a U-turn with the sensor in the pocket, without gestures (MoCap ID: 14) for a $C/N_0$ below 35dB-Hz, or with a gesture, "send a text/email" (MoCap ID:17) or "answer the phone"

(MoCap ID: 18), both for a $C/N_0$ below 30dB-Hz. Higher SGPLL bandwidths, show that the real motion profiles did not cause significant disruption on the SGPLL by inducing cycle slips due to dynamic stress, with the significance ratio below the 0.6745 threshold (i.e. 50% CL) for the 15Hz bandwidth and below 0.01 (less than 1% CL) for the 20Hz effective bandwidth.

As far as the effect of real motion on the simulated GNSS frequency lock loop (SGFLL) is concerned, the results in Table 6.2 show that motion profiles which cause the most significant disruption due to dynamics stress on the SGFLL, above the 69th percentile, encompass walking with the sensor in the pocket and making a U-turn, without performing any gestures (100th percentile), or performing a "send a text/email" gesture (77th percentile) or an "answer the phone" gesture (69th percentile), as well as walking with the sensor held by hand and making a U-turn, without performing any gestures (95th percentile), or performing an "answer the phone" gesture (92nd percentile). The motion scenarios causing the least significant disruption due to dynamics stress on the SGFLL, below the 25th percentile, encompass walking holding the sensor by hand without performing any gestures (25th percentile) or performing an "answer the phone" gesture (17th percentile); walking having the sensor attached on the upper arm with an arm-band (16th percentile); walking with the sensor inside a pocket and performing an "answer the phone" gesture (13th percentile); walking having the sensor inside a back-pack (11th percentile); and jogging having the sensor inside the pocket (6th percentile) or held by hand (first percentile).

The results in Figure 6.9, which illustrate the significance of the dynamic stress induced by the real motion profiles on the SGFLL, show that when a 1Hz SGFLL bandwidth is employed, real motion cause significant disruption due to dynamic stress on the SGFLL, above a $\Delta q_{RM,i/S}^{FLL,FT} \Big/ \sigma\left(\Delta q_{RM,i/S}^{FLL,FT}\right)$ ratio of 2 (i.e. 95% CL) for all $C/N_0$ values and MoCap scenarios, except for jogging with the sensor held by hand (MoCap ID: 11) or inside the pocket (MoCap ID: 12). For a 2Hz SGFLL effective bandwidth, all real motion profiles cause significant disruption due to dynamic stress on the SGFLL, above a $\Delta q_{RM,i/S}^{FLL,FT} \Big/ \sigma\left(\Delta q_{RM,i/S}^{FLL,FT}\right)$ ratio of 0.6745 (i.e. 50% CL), except for scenarios which include walking with the sensor inside a back-pack (MoCap ID: 3); jogging with the sensor held by hand (MoCap ID: 11) or inside the pocket (MoCap ID: 12); and walking with the sensor inside the pocket and performing an "answer the phone" gesture (MoCap ID: 8), for $C/N_0$ values up to 28dB-Hz.

For a 5Hz SGFLL bandwidth, the results in Figure 6.9 show that the significance of the real motion disruption (due to dynamics stress) on the SGFLL, never exceeds the 0.6745 ratio (i.e. 50% CL, denoted with the black dotted line in in Figure 6.9), except for lower $C/N_0$ values (below 27dB-Hz) when walking with the sensor inside the pocket, without performing any gestures or U-turns (MoCap ID: 2), or performing a U-turn (MoCap ID: 14), or performing a "send a text/email" gesture (MoCap ID: 17); walking holding the sensor by hand and performing an "send a text/email" gesture

(MoCap ID: 9), or performing a U-turn (MoCap ID: 13); or performing a U-turn) and an “answer the phone” gesture (MoCap ID: 15). The significance ratio of real motion disruption on the SGFLL for a 10Hz effective bandwidth, is below 0.6745 (i.e. 50% CL) for all MoCap scenarios.

Further to the results showing the effect of real motion on the SGCTLs, this chapter assessed the impact of synthetic motion, for the motion profiles simulated by the pedestrian motion model (PMM) which was described in Chapter 5. The difference between the synthetic and real motion effect significance on the SGPLL is given in Table 6.3, with the synthetic motion significance ratios $\Delta q/\sigma(\Delta q)$ illustrated in Figure 6.12 and the real motion ones (for comparison) in Figure 6.13, in Section 6.3. This comparison shows that in some cases the PMM output motion dynamics underrepresent (or exaggerate) the effect of the real motion on the SGPLL, although less than a factor of two (for all effective bandwidths combined in a single metric). More specifically, the PMM output creates a similar effect to the real motion on the SGPLL when a 5Hz bandwidth is employed, by inducing significant dynamic stress to the SGPLL for all simulated scenarios and $C/N_0$ values (similar to the real motion effect), above a significance ratio of 2.75 (i.e. 99.4% CL).

For a 10Hz effective SGPLL bandwidth, the synthetic motion exaggerated the real motion profiles’ effect on the SGPLL, by increasing the significance ratio up to 1.4 (i.e. 83.4% CL) compared to the real motion, for the applicable simulated scenarios and $C/N_0$ values. These results also show that for a 15Hz SGPLL bandwidth, the synthetic motion induces the same effect on the SGPLL as the real motion, i.e. it only increases the significance ratio by 0.05, i.e. a 4% CL, above a 37dB-Hz $C/N_0$, which means that there is a 4% confidence level that the synthetic motion induced a significant dynamic stress (in addition to the one induced by the real motion). Finally, the comparison between the effect of the real and synthetic motion on the SGPLL for a 20Hz effective bandwidth, shows that the synthetic motion induces an additional dynamic stress on the SGPLL of a level (ratio) below 0.02, i.e. there is 1.6% confidence that the synthetic motion produced a significant effect on the SGPLL, in addition to the real motion effect, for all simulated $C/N_0$ values.

### 6.4.2 **Recommendations for GNSS receiver design**

Following the summary of the real motion effect results on the SGCTLs, Table 6.4 shows the minimum bandwidth requirements recommended for a GNSS receiver design operating in a (typical) pedestrian-induced dynamics’ environment, based on the results of the real motion effect significance on the SGPLL and SGFLL, discussed in Section 6.2 and illustrated in Figure 6.7 and Figure 6.9, respectively. These requirements are shown for three $C/N_0$ levels, at a 30dB-Hz $C/N_0$, in the third column, for the phase lock loop (PLL) design and in the 6th column for a frequency lock loop (FLL) design; at a 35dB-Hz $C/N_0$ (in the fourth column for PLL design and 7th column for the FLL design); and at a 40dB-Hz $C/N_0$ (in the fifth column for PLL design and 8th column for FLL design).

Based on the evidence summarised in Table 6.4, the minimum bandwidth recommendation for a GNSS PLL design, in the presence of typical pedestrian motion, is generally 10Hz, while for a GNSS FLL design is 5Hz, with a few exceptions highlighted in bold lettering, i.e. when the pedestrian motion encompasses walking with the sensor in the pocket (MoCap ID: 2) then it is recommended to use an FLL (with a minimum 10Hz effective bandwidth) for carrier-tracking at a 30dB-Hz $C/N_0$ and at least a 15Hz PLL effective bandwidth, for a $C/N_0$ above 35dB-Hz. Also, when the sensor is attached on the upper arm with an arm-band (MoCap ID: 4) and the $C/N_0$ is above 40dB-Hz, then the PLL bandwidth could be 5Hz (or higher). For pedestrian motion profiles encompassing walking with the sensor in the pocket and performing an "answer the phone" or "send a text/email" gesture (MoCap IDs: 6, 8 and 10), the bandwidth recommendation for a PLL design is 15Hz for a 30dB-Hz $C/N_0$, noting that using a higher bandwidth at this $C/N_0$ level would introduce additional cycle slips due to tracking noise (causing a potential degradation in PLL performance instead of improvement); and a minimum 15Hz PLL bandwidth a 35dB-Hz $C/N_0$.

In addition, for a pedestrian motion profile encompassing jogging holding the sensor by hand (MoCap ID: 11), the recommended GNSS receiver design is to switch to FLL (with a minimum 2Hz bandwidth) for carrier tracking at a 30dB-Hz $C/N_0$; use a PLL bandwidth of 15Hz (at least) for a 35dB-Hz or 40dB-Hz $C/N_0$, or an FLL with a minimum 2Hz bandwidth at a 35dB-Hz $C/N_0$, or a minimum 1Hz bandwidth at a 40dB-Hz $C/N_0$. For a motion profile including jogging with the sensor inside a pocket (MoCap ID: 12), it is recommended to use FLL for carrier tracking at a 30dB-Hz $C/N_0$, and a minimum 15Hz PLL bandwidth when the $C/N_0$ is 35dB-Hz or 40dB-Hz. Furthermore, when a pedestrian is walking having the sensor inside the pocket and making a U-turn, the recommended method for carrier tracking at a 30dB-Hz $C/N_0$ is to use an FLL with a 10Hz bandwidth. For the same motion profile, a PLL bandwidth of at least 15Hz might be employed when the $C/N_0$ is 35dB-Hz or 40dB-Hz. Finally, a PLL bandwidth of 15Hz is recommended at a 30dB-Hz $C/N_0$ when the pedestrian motion profile includes walking with the sensor in the pocket, making a U-turn, and performing a "send a text/email" gesture (MoCap ID: 17), or an "answer the phone" gesture (MoCap ID: 18) in which case, a minimum 15Hz PLL bandwidth is recommended for a 35dB-Hz and a 40dB-Hz $C/N_0$.

It must be noted that information on code/carrier-tracking loop bandwidths is typically commercially confidential, especially for smartphones and wearable devices for pedestrian applications (i.e. lower-grade receivers). For high-end receivers this information can be available as the user can adjust the carrier tracking loops' bandwidths depending on the application requirements. However, high-end receivers are not representative of the GNSS equipment in smartphones and wearable devices used for pedestrian applications, noting that for most pedestrian applications carrier phase cycle-slipping is not an important issue as they tend to provide the user with an integrated navigation solution (fusing measurements between GNSS and other sensors) which smooths out these effects (on the expense of a lower output rate). However, for GNSS deign purposes it is important to account for these errors in order to optimise the navigation algorithms. The thesis continues with Chapter 7, which summarises the work conducted in this thesis, and draws some conclusions based on the evidence from the results presented on this chapter.

| *Brief MoCap Scenario Description* | *MoCap Scenario ID (i)* | *Recommended PLL effective bandwidths (Hz), for $C/N_0$ (dB-Hz):* | | | *Recommended FLL effective bandwidths (Hz), for $C/N_0$ (dB-Hz):* | | |
|---|---|---|---|---|---|---|---|
| | | *30* | *35* | *40* | *30* | *35* | *40* |
| Walk; sensor in hand | 1 | 10, 15 | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk; sensor in pocket | 2 | **n/a** | **≥15** | **≥15** | **10** | ≥5 | ≥5 |
| Walk; sensor in back-pack | 3 | 15 | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk; sensor in arm-band | 4 | 10, 15 | ≥10 | **≥5** | ≥5 | ≥5 | ≥5 |
| Walk 10m; sensor in hand, “answer the phone” | 5 | 10, 15 | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk 10m; sensor in pocket, “answer the phone” | 6 | **15** | **≥15** | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk 15m; sensor in hand, “answer the phone” | 7 | 10, 15 | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk 15m; sensor in pocket, “answer the phone” | 8 | **15** | **≥15** | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk 15m; sensor in hand, “send a text/email” | 9 | 10, 15 | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk 15m; sensor in pocket, “send a text/email” | 10 | **15** | **≥15** | ≥10 | ≥5 | ≥5 | ≥5 |
| Jog; sensor in hand | 11 | **n/a** | **≥15** | **≥15** | **≥2** | **≥2** | **≥1** |
| Jog; sensor in pocket | 12 | **n/a** | **≥15** | **≥15** | ≥5 | ≥5 | ≥5 |
| Walk; sensor in hand; U-turn | 13 | 10, 15 | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk; sensor in pocket; U-turn | 14 | **n/a** | **≥15** | **≥15** | **10** | ≥5 | ≥5 |
| Walk 15m; sensor in hand, “answer the phone”; U-turn | 15 | 10, 15 | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk 15m; sensor in hand, “send a text/email”; U-turn | 16 | 10, 15 | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk 15m; sensor in pocket, “send a text/email”; U-turn | 17 | **15** | ≥10 | ≥10 | ≥5 | ≥5 | ≥5 |
| Walk 15m; sensor in pocket, “answer the phone”; U-turn | 18 | **15** | **≥15** | **≥15** | ≥5 | ≥5 | ≥5 |

Table 6.4. Bandwidth recomendations for GNSS receivers – based on a dynamic stress disruption confidence level (CL) less than ±10% (±0.1256 SDs) on the SGCTLs.

# Chapter 7. Summary and Conclusions

This chapter provides a summary of research work conducted in order to address the aims and objectives of the thesis, presented in Section 1.2, and continues with drawing some conclusions from the results detailed in Chapter 6. These two research questions are, firstly, which types of pedestrian movement most affect the performance of GNSS equipment, and secondly, to describe which is the best way to simulate pedestrian motion in order to simulate effectively the effect of real motion on the GNSS equipment.

To address these research questions, the thesis first reviewed the current state of the art (see Chapter 2) in order to introduce basic concepts and principles in pedestrian motion analysis in Section 2.1; GNSS and inertial navigation systems in Section 2.2; and methods of human motion capture and modelling in Section 2.3. This discussion informed the design, development and implementation of methods in the remainder of the thesis, i.e. the design of an appropriate human motion data capture protocol (including usage of the experimental equipment) in Chapter 3; the development of the simulated GNSS carrier-tracking loops (see Chapter 4) to assess the effect of the real captured motion and the synthetic motion output of the proposed pedestrian motion model (described in Chapter 5). In addition, the discussion in Chapter 2 supported the development of the methods to pre-process the real motion and analyse its effect, alongside the synthetic motion, on the simulated GNSS carrier tracking loops (SGCTLs), as well as the interpretation of the results (in Chapter 6), as well as the identification of research gaps and longer term aspirations for this thesis, discussed in Chapter 8.

The thesis progressed in Chapter 3 with describing the pilot and main experiments conducted in order to capture real human motion data. During the pilot experiments two significant issues were identified with the experimental equipment (Xsens MTi-G), i.e. that the gyros' dynamic range was not sufficient to capture human motion and that the integrated GPS/inertial solution was containing significant artefacts which should be removed. For this reason, the inertial raw data, provided by Xsens MTi-G were used to calculate the inertial navigation solution which, as discussed in Section 6.1.1, does not contain these artefacts. The thesis continued, in Chapter 4, with describing how a GNSS phase lock loop and a frequency lock loop were simulated in Matlab environment, in order to simulate and assess the effect of the real captured human motion and the synthetic output of the proposed pedestrian motion model (in Chapter 5) to real GNSS receivers. The simulated GNSS carrier-tracking loops were validated using relevant theory, with the results showing a better compliance of the simulated phase lock loop with the theory, than the frequency lock loop.

The thesis proceeded with the description of the underlying mathematical model for a pedestrian motion model, able to generate a 3D dynamics profile (in terms of position, velocity and attitude) for a sensor held or attached on to the (walking) human body, given some optional pre-defined gestures, applicable to the carried sensor, e.g. "answer the phone". It is worth noting that the proposed pedestrian motion model in this thesis has the potential to export (linear) acceleration, and jerk dynamics, and angular velocity, acceleration and jerk, which will enable it to feed 3D GNSS antenna dynamics to Spirent's SimGEN software suite for GNSS RF signals' simulation [138].

The thesis also investigated, in Chapter 6, the effect of a range of real motion scenarios (18 in total) and synthetic motion scenarios (7 in total) on the performance of the simulated GNSS carrier-tracking loops (SGCTLs). The results presented in Section 6.1, address the first research question about which are the key aspects of pedestrian motion that affect the performance of GNSS equipment, while the results presented in Section 6.3 address the second aim of this thesis, about what is the best way to generate a pedestrian motion model for simulation testing of GNSS equipment, which is, essentially, the validation result (i.e. pass or fail, and under what conditions) of the pedestrian motion model (PMM) described in Chapter 5.

In terms of which movements most significantly affect the performance of the developed SGCTLs, the analysis presented in Section 6.2 showed that the simulated GNSS phase lock loop was affected mostly by the motion profiles involving placing the sensor in the pocket, especially at a high gait speeds, such as when jogging. The performance of the SGCTLs was also found to be most significantly affected by real motion profiles which encompass U-turns, especially at the 5Hz SGPLL and 1Hz SGFLL bandwidths. As a general rule, higher SGCTLs' effective bandwidths were more robust against real motion-induced dynamic stress than lower ones, noting that below the $C/N_0$ levels given in Table 4.1 of Section 4.4, the GNSS receiver performance would degrade as a result of the higher tracking bandwidths allowing more tracking noise to enter the carrier tracking loops. The real motion effect analysis on the SGPLL shows that the MoCap scenarios which has the least significant effect on the are those with the sensor placed inside a back-pack or attached on the upper arm with an arm-band, and for the SGFLL, the motion profiles which encompass jogging.

With respect to the synthetic unsmoothed motion effects on the SGCTLs, discussed in Section 6.3, the results showed that in some cases the PMM output motion dynamics underrepresent (or exaggerate) the effect of the real motion on the SGPLL, although less than a factor of two (for all effective bandwidths combined in a single metric). More specifically, the PMM output creates a similar effect to the real motion on the SGPLL when a 5Hz bandwidth is employed, by inducing significant dynamic stress to the SGPLL for all simulated scenarios and $C/N_0$ values while for a 10Hz effective SGPLL bandwidth, the synthetic motion generally exaggerated the real motion profiles' effect on the SGPLL. These results also show that for a 15Hz SGPLL bandwidth, the synthetic motion induces the same effect on the SGPLL as the real motion, above a 37dB-Hz $C/N_0$, and for all $C/N_0$ values when a 20Hz SGPLL effective bandwidth is employed.

Chapter 6 concludes with recommendations on GNSS receiver design in the presence of typical pedestrian motion, summarised in Table 6.4 in terms of minimum recommended bandwidths (for specific $C/N_0$ levels) for phase lock loop (PLL) and frequency lock loop (FLL) design. As a general recommendation, a PLL should operate at a tracking bandwidth greater than 10Hz, with a few exceptions for which the recommended bandwidth should increase to 15Hz (or decrease to 5Hz, as appropriate), or carrier tracking is recommended to use an FLL instead, e.g. for cases when the pedestrian motion profile encompasses jogging. Therefore, GNSS receiver designers should exercise caution when adjusting the phase lock loop (PLL) and frequency lock loop (FLL) carrier-tracking bandwidths in the presence of typical pedestrian motion.

The thesis proceeds with Chapter 8, which discusses the future work deriving from this research.

# Chapter 8. Future Work

This chapter discusses research areas and additional tasks which derive from the research work conducted for this thesis, and which are recommended for future work purposes. These include the modelling of crystal oscillator phase noise in GNSS carrier tracking loop simulators, in order to study the behaviour of carrier tracking loops in the presence of pedestrian motion at lower bandwidth regions, e.g. less than 5Hz for a phase lock loop (PLL) or less than 2Hz for a frequency lock loop (FLL), at which the clock noise increases significantly. Also, in terms of validating the accuracy of the results from the simulated carrier tracking loops, a validation testing experimental setup can include a real GNSS receiver and Spirent simulators, which can simulate the line-of-sight motion dynamics between the receiver and satellite antennae. This is achieved through a Spirent-proprietary user motion trajectory (UMT) file, which contains a 6 degrees-of-freedom (6 DOF) GNSS antenna trajectory, used to adjust appropriately the phase and Doppler of the simulated GNSS signals. It is worth noting that the thesis's results are currently validated against theoretical models published in the literature, as explained in Section 4.4.

Furthermore, a future line of enquiry could encompass studying the effect of individual gait variations, e.g. due to age, gender and/or health status, on positioning and navigation sensors' performance. This could also inform the scope extension of the pedestrian motion model (PMM), e.g. in terms of greater variability on gait speed selection, curved paths and object avoidance in the routing model, incorporation of environmental features, such as elevation, stairs, and/or other types of activity, like running, cycling or boarding a train. Also, the PMM could be updated in order to allow the user to input a specific step (stride) length, with the PMM adjusting the gait profile (segments' rotation angles) in order to match the specified step length. Another option would be to take as well take into account the in-path step variations, e.g. smaller steps walking at the beginning of the path (or before stopping) and larger steps in the middle of the path. Also, a random element could be added to the attitude increment when the user defines the final gestures' attitude in order to make the segments' rotation (involved in a specific gesture) more realistic.

With respect to interfacing the PMM with Spirent software packages, a production version of the PMM may include a user interface to define parameters, such as the gait speed, human body height and segment proportions, the step length, a facility to calculate the pedestrian route that is to be followed, and the gestures, e.g. by defining the final attitude by rotating the segments of a human biomechanical model in order to reach the desired final posture and the start/end timings of the gestures' application. This user interface may generate a Spirent-proprietary UMT file which can be fed into the Spirent software packages to control the simulated GNSS RF signals.

Finally, additional lines of enquiry may include:

- Develop a custom smoothing algorithm with adjustable gain, depending on the carrier-tracking loop effective bandwidth for the PMM output.
- Investigate the relative motion of the sensor with respect to the user's body, in order to study the effect of GNSS signal shadowing (or masking) due to human body, to GNSS receiver's performance.
- Consider using equipment of better specifications, e.g. an IMU sensor, with gyros which can sense angular rates up to 750deg/s about their sensitive axes (compared to 300deg/s of the used Xsens MTi-G in the thesis' experiments), and a GPS receiver which can provide pseudo-range measurements in order to perform a custom INS/GNSS integration, as this was beyond the timescales of the current research.

The discussion of the future work in this chapter, concludes the main body of the thesis.

# References

[1] D. Demeritt and L. Lees, "Research relevance, 'knowledge transfer' and the geographies of CASE studentship collaboration," *Area*, vol. 37, no. 2, pp. 127–137, Jun. 2005.

[2] M. Braasch and A. Van Dierendonck, "GPS receiver architectures and measurements," *Proc. IEEE*, vol. 87, no. 1, 1999.

[3] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Artech House, 2013.

[4] R. a Nelson, D. D. McCarthy, S. Malys, J. Levine, B. Guinot, H. F. Fliegel, R. L. Beard, and T. R. Bartholomew, "The leap second: its history and possible future," *Metrologia*, vol. 38, no. 6, pp. 509–529, Dec. 2001.

[5] W. Tranter, R. Thamvichai, and T. Bose, *Basic Simulation Models of Phase Tracking Devices Using MATLAB*, vol. 4, no. 1. Morgan & Claypool, 2010.

[6] J. Hamill and K. Knutzen, *Biomechanical basis of human movement*, Third Ed. Wolters Kluwer Health/Lippincott Williams and Wilkins, 2009.

[7] R. L. Huston, *Principles of biomechanics*, vol. 213. Boca Raton, London, New York: CRC Press, 2009.

[8] M. I. Marpet and M. A. Sapienza, Eds., *Metrology of pedestrian locomotion and slip resistance*, no. 1424. West Conshohocken: ASTM International, 2003.

[9] A. Hartmann, K. Murer, R. a de Bie, and E. D. de Bruin, "Reproducibility of spatio-temporal gait parameters under different conditions in older adults using a trunk tri-axial accelerometer system.," *Gait Posture*, vol. 30, no. 3, pp. 351–5, Oct. 2009.

[10] P. K. Levangie and C. C. Norkin, *Joint structure and function: a comprehensive analysis*, Fourth Ed. FA Davis Philadelphia, PA, 2005.

[11] S. A. Gard and D. S. Childress, "The effect of pelvic list on the vertical displacement of the trunk during normal walking," *Gait Posture*, vol. 5, no. 3, pp. 233–238, 1997.

[12] S. M. Bruijn, O. G. Meijer, J. H. van Dieën, I. Kingma, and C. J. C. Lamoth, "Coordination of leg swing, thorax rotations, and pelvis rotations during gait: The organisation of total body angular momentum," *Gait Posture*, vol. 27, pp. 455–462, 2008.

[13] J. Rose, J. G. Gamble, E. Rose, and G. James, *Human walking*. Lippincott Williams & Wilkins, 2006.

[14] J. Pinchin, G. Smith, C. Hill, T. Moore, and I. Loram, "The Potential of Electromyography to Aid Personal Navigation," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), September 8 - 12, 2014, Tampa, Florida*, 2014, pp. 1609–1615.

[15] W. Chen, R. Chen, X. Chen, X. Zhang, Y. Chen, J. Wang, and Z. Fu, "Comparison of EMG-

based and Accelerometer-based Speed Estimation Methods in Pedestrian Dead Reckoning," *J. Navig.*, vol. 64, no. 2, pp. 265–280, Apr. 2011.

[16] R. Stirling, K. Fyfe, and G. Lachapelle, "Evaluation of a New Method of Heading Estimation for Pedestrian Dead Reckoning Using Shoe Mounted Sensors," *J. Navig.*, vol. 58, no. 1, pp. 31–45, Jan. 2005.

[17] J. B. dec. M. Saunders, V. T. Inman, and H. D. Eberhart, "The major determinants in normal and pathological gait," *J. Bone Jt. Surg.*, vol. 35, no. 3, p. 543, 1953.

[18] V. Racic, A. Pavic, and J. M. W. Brownjohn, "Experimental identification and analytical modelling of human walking forces: Literature review," *J. Sound Vib.*, vol. 326, no. 1–2, pp. 1–49, Sep. 2009.

[19] J. Perry, *Gait analysis: normal and pathological function*, vol. 12, no. 6. Thorofare: SLACK Incorporated, 1992.

[20] M. Whittle, *An Introduction to Gait Analysis*, 4th ed. Butterworth-Heinemann, 2007.

[21] M. Yamasaki, T. Sasaki, and M. Torii, "Sex difference in the pattern of lower limb movement during treadmill walking," *Eur. J. Appl. Physiol. Occup. Physiol.*, vol. 62, no. 2, pp. 99–103, 1991.

[22] A. Pachi and J. Tianjian, "Frequency and velocity of people walking," *Struct. Eng.*, vol. 83, no. 3, pp. 36–40, 2005.

[23] H. Hicheur, Q.-C. Pham, G. Arechavaleta, J.-P. Laumond, and A. Berthoz, "The formation of trajectories during goal-oriented locomotion in humans. I. A stereotyped behaviour," *Eur. J. Neurosci.*, vol. 26, no. 8, pp. 2376–2390, 2007.

[24] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds.," *Proc. Biol. Sci.*, vol. 276, no. 1668, pp. 2755–62, Aug. 2009.

[25] A. Panangadan, M. Matarić, and G. S. Sukhatme, "Tracking and Modeling of Human Activity Using Laser Rangefinders," *Int. J. Soc. Robot.*, vol. 2, no. 1, pp. 95–107, Jan. 2010.

[26] R. A. Mann and J. Hagy, "Biomechanics of walking, running, and sprinting," *Am. J. Sports Med.*, vol. 8, no. 5, p. 345, 1980.

[27] T. Novacheck, "The biomechanics of running.," *Gait Posture*, vol. 7, no. 1, pp. 77–95, Jan. 1998.

[28] M. S. Orendurff, A. D. Segal, G. K. Klute, J. S. Berge, E. S. Rohr, and N. J. Kadel, "The effect of walking speed on center of mass displacement," *J. Rehabil. Res. Dev.*, vol. 41, no. 6, p. 829, 2004.

[29] M. W. Whittle, "Three-dimensional motion of the center of gravity of the body during walking," *Hum. Mov. Sci.*, vol. 16, no. 2–3, pp. 347–355, Apr. 1997.

[30] C. R. Lee and C. T. Farley, "Determinants of the center of mass trajectory in human walking and running.," *J. Exp. Biol.*, vol. 201, no. Pt 21, pp. 2935–44, Nov. 1998.

[31] A. L. Betker, Z. M. K. Moussavi, and T. Szturm, “Center of mass approximation and prediction as a function of body acceleration.,” *IEEE Trans. Biomed. Eng.*, vol. 53, no. 4, pp. 686–93, Apr. 2006.

[32] A. Thorstensson, J. Nilsson, H. Carlson, and M. R. Zomlefer, “Trunk movements in human locomotion,” *Acta Physiol. Scand.*, vol. 121, no. 1, pp. 9–22, May 1984.

[33] M. Srinivasan and A. Ruina, “Computer optimization of a minimal biped model discovers walking and running.,” *Nature*, vol. 439, no. 7072, pp. 72–5, Jan. 2006.

[34] M. Murray and R. Kory, “Comparison of free and fast speed walking patterns of normal men,” *Am. J. Phys. Med.*, vol. 45, no. 1, pp. 8–24, 1966.

[35] S. M. Bruijn, O. G. Meijer, P. J. Beek, and J. H. van Dieën, “The effects of arm swing on human gait stability.,” *J. Exp. Biol.*, vol. 213, no. Pt 23, pp. 3945–52, Dec. 2010.

[36] D. Lenzi, A. Cappello, and L. Chiari, “Influence of body segment parameters and modeling assumptions on the estimate of center of mass trajectory,” *J. Biomech.*, vol. 36, no. 9, pp. 1335–1341, Sep. 2003.

[37] S. A. Gard, S. C. Miff, and A. D. Kuo, “Comparison of kinematic and kinetic methods for computing the vertical motion of the body center of mass during walking.,” *Hum. Mov. Sci.*, vol. 22, no. 6, pp. 597–610, Apr. 2004.

[38] C. Hayot, S. Sakka, and P. Lacouture, “Contribution of the six major gait determinants on the vertical center of mass trajectory and the vertical ground reaction force.,” *Hum. Mov. Sci.*, vol. 32, pp. 279–89, 2013.

[39] A. Van Dierendonck, “GPS receivers,” in *Global Positioning System: Theory and applications.*, B. W. Parkinson and J. J. J. Spiker, Eds. Wasington, DC: American Institute of Aeronautics and Astronautics, 1996, pp. 329–407.

[40] E. D. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications, Second Edition.* Artech House Publishers, 2005.

[41] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*, 2nd ed. Linconln, MA: Ganga-Jamuna Press, 2006.

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

[43] J. A. Farrell, *Aided navigation : GPS with high rate sensors.* New York ; London : McGraw-Hill, 2008.

[44] J. Iliffe and R. Lott, *Datums and Map Projections: for Remote Sensing, GIS and Surveying*, 2nd Ed. Caithness, Scotland: Whittles Publishing, 2008.

[45] Ordnance Survey, “A guide to coordinate systems in Great Britain,” Southampton, UK, 2016.

[46] K. Saber-Sheikh, E. C. Bryant, C. Glazzard, A. Hamel, and R. Y. W. Lee, “Feasibility of using inertial sensors to assess human movement.,” *Man. Ther.*, vol. 15, no. 1, pp. 122–5, Feb. 2010.

[47] T. Bosgiraud, “Two Degrees of Freedom Miniaturized Gyroscope based on Active Magnetic Bearings,” These de doctoral d’etat, Faculté Sciences et Techniques de l’ Ingénieur, École Polytechnique Fedérale de Lausanne, Lausanne, Suisse, 2008.

[48] N. El-Sheimy, H. Hou, and X. Niu, “Analysis and Modeling of Inertial Sensors Using Allan Variance,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1, pp. 140–149, Jan. 2008.

[49] Xsens, “MTi-G User Manual and Technical Documentation (Revision H),” 2010.

[50] M. Grewal, A. Andrews, and C. Bartone, *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. Hoboken, NJ: John Wiley & Sons, Inc, 2013.

[51] F. van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*. Boston: Artech House, 2009.

[52] F. van Diggelen and P. Enge, “The World’s first GPS MOOC and Worldwide Laboratory using Smartphones,” in *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015), Tampa, Florida, September 2015*, pp. 361–369.

[53] H. Kuusniemi, J. Liu, L. Pei, Y. Chen, L. Chen, and R. Chen, “Reliability considerations of multi-sensor multi-network pedestrian navigation,” *IET Radar, Sonar Navig.*, vol. 6, no. 3, p. 157, 2012.

[54] Ahmed El-Rabbany, *Introduction to GPS: The Global Positioning System*. Norwood, MA: Artech House, 2002.

[55] M. Grewal, L. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. New York: John Wiley & Sons, 2001.

[56] John Betz, *Engineering Satellite-Based Navigation and Timing: Global Navigation Satellite Systems, Signals, and Receivers*. Wiley-IEEE Press, 2016.

[57] J. Bancroft, G. Lachapelle, T. Williams, and J. Garrett, “GPS Observability and Availability for Various Antenna Locations on the Human Body,” in *Proceedings of ION GNSS 2010*, 2010, pp. 1–11.

[58] M. U. Rehman, “Characterisation of human body and environmental effects on the performance of mobile terminal antennas,” Ph.D. dissertation, School of Elec. Eng. & Comp. Sc., Queen Mary, University of London, London, UK, 2010.

[59] G. Cariolaro, *Unified Signal Theory*. London: Springer London, 2011.

[60] T. Pany, *Navigation signal processing for GNSS software receivers*. Norwood: Artech House, 2010.

[61] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo & more*. Springer, 2007.

[62] A. Joseph, “Measuring GNSS Signal Strength,” *InsideGNSS*, pp. 20–25, 2010.

[63] W.-L. Mao, H.-W. Tsao, and F.-R. Chang, “A New Fuzzy Bandwidth Carrier Recovery System in GPS for Robust Phase Tracking,” *IEEE Signal Process. Lett.*, vol. 11, no. 4, pp. 431–434, Apr. 2004.

[64] Y. Li, M. Zhao, J. Zhong, X. Lv, and Y. Wang, “An algorithm based on dynamic

compensation for GNSS carrier tracking of weak signal," in *2010 12th IEEE International Conference on Communication Technology (ICCT)*, 2010, no. 1, pp. 1184–1187.

[65] Y. Yang, J. Zhou, and O. Loffeld, "GPS receiver tracking loop design based on a Kalman filtering approach," in *ELMAR, 2012 Proceedings*, 2012, pp. 121–124.

[66] G. Yao, W. Wenqi, and H. Xiaofeng, "High dynamic carrier phase tracking based on adaptive Kalman filtering," in *2011 Chinese Control and Decision Conference (CCDC)*, 2011, pp. 1245–1249.

[67] M. Psiaki and H. Jung, "Extended Kalman filter methods for tracking weak GPS signals," in *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, 2002, pp. 2539–2553.

[68] L.-T. Hsu, S.-S. Jan, P. D. Groves, and N. Kubo, "Multipath mitigation and NLOS detection using vector tracking in urban environments," *GPS Solut.*, Jun. 2014.

[69] K. H. Kim, G. I. Jee, and J. H. Song, "Carrier tracking loop using the adaptive two-stage Kalman filter for high dynamic situations," *Int. J. Control. Autom. Syst.*, vol. 6, no. 6, pp. 948–953, 2008.

[70] V. Gabaglio, "Centralised Kalman filter for augmented gps pedestrian navigation," in *Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001) September 11 - 14, 2001, Salt Lake City, UT*, 2001, pp. 312–318.

[71] A. Razavi, D. Gebre-Egziabher, and D. M. Akos, "Carrier loop architectures for tracking weak GPS signals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 697–710, Apr. 2008.

[72] M. Irsigler and B. Eissfeller, "PLL Tracking Performance in the Presence of Oscillator Phase Noise," *GPS Solut.*, vol. 5, no. 4, pp. 45–57, Apr. 2002.

[73] J. T. Curran, G. Lachapelle, and C. C. Murphy, "Digital GNSS PLL Design Conditioned on Thermal and Oscillator Phase Noise," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 180–196, Jan. 2012.

[74] Septentrio nv/sa, "SSRC4 Command Line Interface Reference Guide: Applicable to version 2.9.6 of the GNSS Firmware," Leuven, Belgium, 2016.

[75] S. Gleason and D. Gebre-Egziabher, *GNSS Applications and Methods*. London: Artech House, 2009.

[76] A. Sandholm, N. Pronost, and D. Thalmann, "MotionLab: A Matlab Toolbox for Extracting and Processing Experimental Motion Capture Data for Neuromuscular Simulations," in *Modelling the Physiological Human. 3D Physiological Human Workshop, 3DPH 2009, Zermatt, Switzerland, November 29 – December 2, 2009. Proceedings*, vol. 5903, N. Magnenat-Thalmann, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 110–124.

[77] Q. Li, M. Young, V. Naing, and J. M. Donelan, "Walking speed estimation using a shank-mounted inertial measurement unit.," *J. Biomech.*, vol. 43, no. 8, pp. 1640–3, May 2010.

[78] J. J. Kavanagh and H. B. Menz, "Accelerometry: a technique for quantifying movement

patterns during walking.," *Gait Posture*, vol. 28, no. 1, pp. 1–15, Jul. 2008.

[79] P. Terrier and Y. Schutz, "How useful is satellite positioning system (GPS) to track gait parameters? A review.," *J. Neuroeng. Rehabil.*, vol. 2, p. 28, Jan. 2005.

[80] J. O'Brien, R. Bodenheimer, G. Brostow, and J. Hodgins, "Automatic joint parameter estimation from magnetic motion capture data," *Graph. Interface*, pp. 53–60, 2000.

[81] "Qualisys Motion Capture Systems," 2017. [Online]. Available: http://www.qualisys.com/. [Accessed: 29-Jan-2017].

[82] "Codamotion Solutions," 2017. [Online]. Available: http://www.codamotion.com/. [Accessed: 29-Jan-2017].

[83] "Vicon Motion Capture Systems," 2017. [Online]. Available: https://www.vicon.com/. [Accessed: 29-Jan-2017].

[84] S. Corazza, L. Mündermann, a M. Chaudhari, T. Demattio, C. Cobelli, and T. P. Andriacchi, "A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach.," *Ann. Biomed. Eng.*, vol. 34, no. 6, pp. 1019–29, Jun. 2006.

[85] S. Knoop, S. Vacek, and R. Dillmann, "Sensor fusion for 3D human body tracking with an articulated 3D body model," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 1686–1691.

[86] S. Obdrzálek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population.," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2012, pp. 1188–93, Jan. 2012.

[87] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller.," *Sensors (Basel).*, vol. 13, no. 5, pp. 6380–93, Jan. 2013.

[88] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Underst.*, vol. 104, no. 2–3, pp. 90–126, Nov. 2006.

[89] X. Yun, E. R. Bachmann, H. Moore, and J. Calusdian, "Self-contained Position Tracking of Human Movement Using Small Inertial/Magnetic Sensor Modules," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 2526–2533.

[90] M. Brodie, A. Walmsley, and W. Page, "Fusion motion capture: a prototype system using inertial measurement units and GPS for the biomechanical analysis of ski racing," *Sport. Technol.*, vol. 1, no. 1, pp. 17–28, Jun. 2008.

[91] C. Fischer, K. Muthukrishnan, M. Hazas, and H. Gellersen, "Ultrasound-aided pedestrian dead reckoning for indoor navigation," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments - MELT '08*, 2008, p. 31.

[92] J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schon, “Tightly coupled UWB/IMU pose estimation,” in *2009 IEEE International Conference on Ultra-Wideband*, 2009, pp. 688–692.

[93] T. Liu, Y. Inoue, and K. Shibata, “Development of a wearable sensor system for quantitative gait analysis,” *Measurement*, vol. 42, no. 7, pp. 978–988, Aug. 2009.

[94] M. J. Floor-Westerdijk, H. M. Schepers, P. H. Veltink, E. H. F. van Asseldonk, and J. H. Buurke, “Use of inertial sensors for ambulatory assessment of center-of-mass displacements during walking.,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 7, pp. 2080–4, Jul. 2012.

[95] D. A. Rodriguez-Silva, F. Gil-Castineira, F. J. Gonzalez-Castano, R. J. Duro, F. Lopez-Pena, and J. Vales-Alonso, “Human motion tracking and gait analysis: a brief review of current sensing systems and integration with intelligent environments,” in *2008 IEEE Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, 2008, pp. 166–171.

[96] T. Cloete and C. Scheffer, “Repeatability of an off-the-shelf, full body inertial motion capture system during clinical gait analysis.,” *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2010, pp. 5125–8, Jan. 2010.

[97] M. de Putte, N. Hagemeister, N. St-Onge, G. Parent, and J. de Guise, “Habituation to treadmill walking,” *Biomed. Mater. Eng.*, vol. 16, no. 1/2006, pp. 43–52, 2006.

[98] P. O. Riley, G. Paolini, U. Della Croce, K. W. Paylo, and D. C. Kerrigan, “A kinematic and kinetic comparison of overground and treadmill walking in healthy subjects.,” *Gait Posture*, vol. 26, no. 1, pp. 17–24, Jun. 2007.

[99] Human Animation Working Group, “H|Anim,” *H|Anim 200x specification (ISO_IEC_FCD_19774).* [Online]. Available: http://h-anim.org/. [Accessed: 28-Feb-2017].

[100] S. Delp, F. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and Darryl G. Thelen, “OpenSim: open-source software to create and analyze dynamic simulations of movement,” *IEEE Trans. Biomed. Eng.*, vol. 54, no. 11, pp. 1940–1950, 2007.

[101] E. Y. L. Gu, *A Journey from Robot to Digital Human*, vol. 1. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[102] G. Giakas and V. Baltzopoulos, “A comparison of automatic filtering techniques applied to biomechanical walking data,” *J. Biomech.*, vol. 30, no. 8, pp. 847–850, Aug. 1997.

[103] O. Arikan and D. a. Forsyth, “Interactive motion generation from examples,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 483–490, Jul. 2002.

[104] M. Al Borno, M. de Lasa, and A. Hertzmann, “Trajectory optimization for full-body movements with complex contacts.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 8, pp. 1405–14, Aug. 2013.

[105] V. G. Duffy, Ed., *Digital Human Modeling and Applications in Health, Safety, Ergonomics, and Risk Management. Human Body Modeling and Ergonomics*, vol. 8026. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[106] G. Salvendy and W. Karwowski, Eds., *Advances in Applied Digital Human Modeling*, vol. 11. Boca Raton: CRC Press, 2012.

[107] D. Helbing, "Models for Pedestrian Behavior," *arXiv Prepr. cond-mat/9805089*, pp. 93–98, May 1998.

[108] A. Keßel, H. Klüpfel, J. Wahle, and M. Schreckenberg, "Microscopic Simulation of Pedestrian Crowd Motion," in *Pedestrian and Evacuation Dynamics*, 2002, pp. 193–202.

[109] T. I. Lakoba, "Modifications of the Helbing-Molnar-Farkas-Vicsek Social Force Model for Pedestrian Evolution," *Simulation*, vol. 81, no. 5, pp. 339–352, May 2005.

[110] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, May 1995.

[111] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 935–942.

[112] V. J. Blue and J. L. Adler, "Cellular automata microsimulation for modeling bi-directional pedestrian walkways," *Transp. Res. Part B Methodol.*, vol. 35, no. 3, pp. 293–312, Mar. 2001.

[113] Jarosław Wąs and Konrad Kułakowski, "Multi-agent Systems in Pedestrian Dynamics Modeling," in *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, vol. 5796, N. T. Nguyen, R. Kowalczyk, and S.-M. Chen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 294–300.

[114] T. Schelhorn, D. O'Sullivan, and M. Haklay, M Thurstain-Goodwin, "STREETS: an agent-based pedestrian model," CASA, UCL, London, UK, Working Paper Series, Paper 9, 1999.

[115] J. Kerridge, J. Hine, and M. Wigan, "Agent-based modelling of pedestrian movements: the questions that need to be asked and answered," *Environ. Plan. B Plan. Des.*, vol. 28, no. 3, pp. 327–341, 2001.

[116] L. Scandolo and T. Fraichard, "An anthropomorphic navigation scheme for dynamic scenarios," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 809–814.

[117] S. Pellegrini, A. Ess, M. Tanaskovic, and L. Van Gool, "Wrong turn - No dead end: A stochastic pedestrian motion model," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 15–22.

[118] P. Pecol, P. Argoul, S. Dal Pont, S. Erlicher, others, P. Pécol, P. Argoul, S. D. Pont, and S. Erlicher, "A new crowd movement modeling for pedestrians who hold hands," *Vib. Shock. Noise*, pp. 1–9, 2012.

[119] A. Chen and Z. Ji, "Path finding under uncertainty," *J. Adv. Transp.*, vol. 39, no. I, pp. 19–37, 2005.

[120] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[121] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of

Minimum Cost Paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[122] S. Koenig and M. Likhachev, "Real-time adaptive A*," in *International Conference on Autonomous Agents: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2006, vol. 8, no. 12, pp. 281–288.

[123] X. Sun, S. Koenig, and W. Yeoh, "Generalized Adaptive A*," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008, pp. 469–476.

[124] J. Yao, B. Zhang, and Q. Zhou, "The Optimization of A* Algorithm in the Practical Path Finding Application," in *2009 WRI World Congress on Software Engineering*, 2009, pp. 514–518.

[125] C. L. Chang and J. R. Slagle, "An admissible and optimal algorithm for searching AND/OR graphs," *Artif. Intell.*, vol. 2, no. 2, pp. 117–128, Sep. 1971.

[126] D. Brogan and N. Johnson, "Realistic human walking paths," in *Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA'03)*, 2003, pp. 1–8.

[127] I. Karamouzas and M. H. Overmars, "Adding variation to path planning," *Comput. Animat. Virtual Worlds*, vol. 19, no. 3–4, pp. 283–293, 2008.

[128] F. Ahmed and K. Deb, "Multi-objective path planning using spline representation," in *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 1047–1052.

[129] C. De Boor, *A practical guide to splines*. New York: Springer-Verlag New York Inc., 1978.

[130] Xsens, "Magnetic Field Mapper Documentation (Revision F)," 2010.

[131] K. Bore, D. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A Software-Defined GPS and Galileo Receiver*. Boston: Birkhäuser, 2007.

[132] J. H. McAuley, J. C. Rothwell, and C. D. Marsden, "Frequency peaks of tremor, muscle vibration and electromyographic activity at 10 Hz, 20 Hz and 40 Hz during human finger muscle contraction may reflect rhythmicities of central neural firing," *Exp. Brain Res.*, vol. 114, no. 3, pp. 525–541, May 1997.

[133] J. H. Allum, V. Dietz, and H. J. Freund, "Neuronal mechanisms underlying physiological tremor.," *J. Neurophysiol.*, vol. 41, no. 3, pp. 557–71, May 1978.

[134] W. Limprasert, "PathFinding." MATLAB File Exchange, 2012.

[135] D. A. Winter, *Biomechanics and motor control of human movement*, Fourth. New Jersey: John Wiley & Sons, Inc., 2009.

[136] J. J. Faraway, M. P. Reed, and J. Wang, "Modelling three-dimensional trajectories by using Bézier curves with application to hand motion," *J. R. Stat. Soc. Ser. C (Applied Stat.*, vol. 56, no. 5, pp. 571–585, Nov. 2007.

[137] R. Boulic, N. M. Thalmann, and D. Thalmann, "A global human walking model with real-time kinematic personification," *Vis. Comput.*, vol. 6, pp. 344–358, 1990.

[138] Spirent, "SimGEN Software Suite Datasheet and Specification," 2017. [Online]. Available:

https://www.spirent.com/-/media/Datasheets/Positioning/SimGEN.pdf. [Accessed: 12-Feb-2017].

[139] MathWorks, "Curve Fitting Toolbox™ User's Guide," *Matlab R2012b*, 2012. [Online]. Available: http://www.mathworks.co.uk/help/curvefit/index.html. [Accessed: 07-Oct-2013].

# Appendix A. Centre of Mass Calculation

This section is based on [1] and explains how to calculate the centre of mass (CoM) of a multi-segment system, such as the human body. Assuming a system of *n* segments (noted with index *i*), with $m_i$ the mass of each individual segment, and that *M* is the whole-body mass of the system:

$$\sum_{i=1}^{n} m_i = M \quad . \tag{A.1}$$

If the mass of each constituent segment relative to the whole body mass is

$$w_i = \frac{m_i}{M} \quad , \quad \sum_{i=1}^{n} w_i = 1 \quad , \tag{A.2}$$

then the coordinates of the CoM of each segment *i*, referenced and resolved in the pedestrian frame *p* axes (Section 2.1.1), will be:

$$\mathbf{r}_{pC_i}^{p} = \mathbf{r}_{pP_i}^{p} + \left\| \mathbf{d}_{P_iC_i}^{P_i} \right\| \left( \mathbf{r}_{pD_i}^{p} - \mathbf{r}_{pP_i}^{p} \right) \quad , \tag{A.3}$$

where $\mathbf{r}_{pC_i}^{p}$ is the coordinates of the *i<sup>th</sup>* segment's CoM, $\mathbf{r}_{pP_i}^{p}$ is the coordinates of its proximal end point, $\mathbf{r}_{pD_i}^{p}$ is the coordinates of its distal end point and $\left\| \mathbf{d}_{P_iC_i}^{P_i} \right\|$ is the distance between its proximal and CoM points, normalised in the segment's length, i.e. the sum of the distances of the segment's CoM between the proximal and distal points is unity, with all segments modelled as linear (with their edges being their proximal/distal points). The proximal/distal end points' normalised coordinates have been defined in Table 5.1.

All segments contribute to the whole-body CoM coordinates (w.r.t. the pedestrian frame axes, defined in Section 2.1.1), depending on their location and relative mass w.r.t. the whole body. The CoM coordinates of the multi-segment system *S* (which may be a subset or equal to the total number of segments of the whole body *n*), is:

$$\mathbf{r}_{pC_S}^{p} = \frac{\sum_{i=1}^{n} w_i \mathbf{r}_{pC_i}^{p}}{\sum_{i=1}^{n} w_i} \quad , \tag{A.4}$$

The values for $w_i$ and $\left\| \mathbf{d}_{P_iC_i}^{P_i} \right\|$ are summarised in Table A.1 [1], noting that [1] defines the hand between the wrist and the second knuckle of the middle finger, while in Table 5.2 the hand is defined between the wrist and the fingertip of the middle finger; however, this difference is assumed to be negligible.

Proximal and distal end coordinates of a segment for a specific timeframe can be are obtained from the human biomechanical model (HBM) described in Section 5.2.

| *Segment index* | *Segment name* | *Segment mass proportion to the whole-body mass* $w_i$ | *Normalised distance of segment's CoM from its proximal end, normalised in segment's length* $\left\|\mathbf{u}^{P_i}_{P_iC_i}\right\|$ |
|---|---|---|---|
| 1 | Head and neck | 0.081 | 1.000 |
| 2 | Thorax and abdomen | 0.355 | 0.630 |
| 3 | Pelvis | 0.142 | 0.105 |
| 4 | Right thigh | 0.100 | 0.433 |
| 5 | Right shank | 0.0465 | 0.433 |
| 6 | Right foot | 0.0145 | 0.500 |
| 7 | Left thigh | 0.100 | 0.433 |
| 8 | Left shank | 0.0465 | 0.433 |
| 9 | Left foot | 0.0145 | 0.500 |
| 10 | Right upper arm | 0.028 | 0.436 |
| 11 | Right forearm | 0.016 | 0.430 |
| 12 | Right hand | 0.006 | 0.506 |
| 13 | Left upper arm | 0.028 | 0.436 |
| 14 | Left forearm | 0.016 | 0.430 |
| 15 | Left hand | 0.006 | 0.506 |

Table A.1. Segment parameters used for HBM CoM calculations (after [1])

Following Eqs. (A.1-4) the coordinates of human body CoM, normalised in stature, are:

$$\mathbf{d}_{pC_{total}}^{p} = \begin{pmatrix} 0 \\ 0.5584 \\ 0 \end{pmatrix} \quad , \tag{A.5}$$

Noting that The derivation of $\mathbf{d}_{pC_{total}}^{p}$ coordinates in (A.5) is detailed in the Excel file "CoM_Calculations.xlsx" in the accompanying CD with this thesis.

## References

[1] D. A. Winter, *Biomechanics and motor control of human movement*, Fourth Ed., New Jersey: John Wiley & Sons, Inc., 2009, p. 370.

# Appendix B. GPS-transients Smoothing Filter

This appendix supports the discussion of Section 6.1.1, which introduced the three identified methods for pre-processing the motion capture (MoCap) data, from the thesis' main experiment (see Section 3.3), i.e.:

A. Convert the Xsens MTI-G-provided integrated inertial/GPS navigation solution, from latitude, longitude, altitude (LLA) coordinates, expressed in North, East and Down (NED) axes, to local tangent plane (LTP) coordinates.

Smooth the integrated inertial/GPS navigation solution (A), i.e. already converted from LLA to LTP coordinates, by detecting and removing ensuing transients from the internal Xsens Kalman Filter, as the examples shown in

B. Figure 3.4 and Figure 3.5.

C. Calculate the inertial navigation solution, using the measurements from the inertial sensors (accelerometers and gsyros) following the calculation steps described in Section 2.2.2. and convert it from LLA to LTP (similar to solutions "B" and "C").

Method "C" is the selected option to pre-process the MoCap data, and was described in Section 6.1.1 This appendix discusses why methods "A" and "B" were not selected, and continues with the description of method "B".

Method "A" can use the readily available LLA coordinates from Xsens MTi-G, converts them to LTP coordinates, and then projects the sensor (body frame) 3D motion along the forward, right and down lines-of-sight (LOS) in order to feed the SGCTLs (see Sections 4.2 and 4.3) with LOS range and range of rate dynamics. It is worth noting that the conversion from NED to LTP coordinates involves Eqs. (B.26) to (B.30), presented later in this appendix.

However, method "A" was found to introduce significant artefacts to the simulated GNSS frequency-tracking loop (SGFLL), as shown in Figure B.1 (highlighted in red), which shows the relative frequency of false frequency locks occurrences (in the vertical-up axis). These results were obtained via a Monte Carlo simulation (MCS) for each set of $C/N_0$ (dB-Hz) and effective bandwidth $B_{L\_CF}$ (Hz) values. The input motion profile, with ID "1" denoted in the subscript of $\breve{q}^{FLL,FT}_{RM(A),1,R}$, encompasses walking 30m in a straight line holding the sensor by hand. The motion in this example is projected along the Right "R" LOS. Each MCS (i.e. one pair of $C/N_0$ and $B_{L\_CF}$ values) was ran 1,000 times adding pseudorandom Matlab sequences to simulate tracking noise in the I and Q equations (see Section 4.3). The horizontal axis shows the simulated $C/N_0$ (dB-Hz) values, ranging from 20dB-Hz to 45dB-Hz, the lateral axis (on the horizontal plane) shows the simulated effective bandwidths $B_{L\_CF}$ (Hz) of 1Hz, 2Hz, 5Hz, and 10Hz.

False frequency locks may occur due to tracking noise at low carrier power-to-noise density ratios $C/N_0$ (dB-Hz) and high effective bandwidths $B_{L\_CF}$ (Hz), as shown in Figure B.1, also due to oscillator noise (not simulated in this thesis, so it is not shown in Figure B.1), and due to LOS dynamics, i.e. average jerk exceeding the SGFLL tolerance over the SGFLL time constant, as discussed in Section 4.3. The latter is the cause of the false frequency locks' artefact observed in Figure B.1, which erroneously shows that for lower effective bandwidths there are no false frequency locks, although the SGPLL should be less tolerant to LOS dynamics at lower $B_{L\_CF}$. The cause of this artefact is explained by showing the input range and range rate data to the SGFLL, in Figure B.2 and Figure B.3, which show that they contain oscillatory elements which cancel out for lower $B_{L\_CF}$.

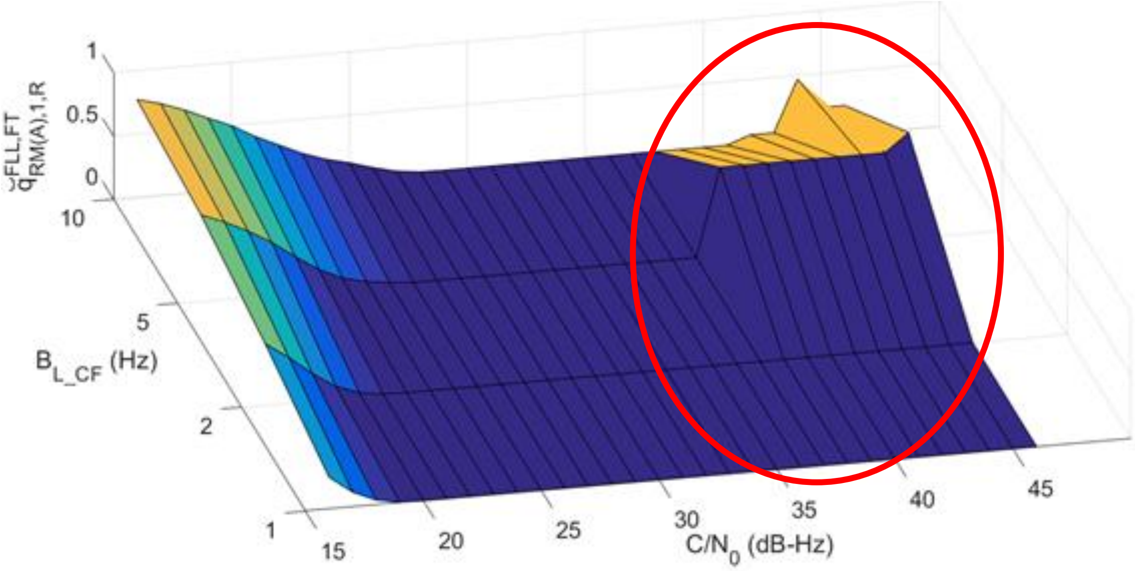

Figure B.1. Real motion (original Xsens integrated navigation solution without smoothing – subscript "A") effects to SGFLL example, with artefacts highlighted in red.

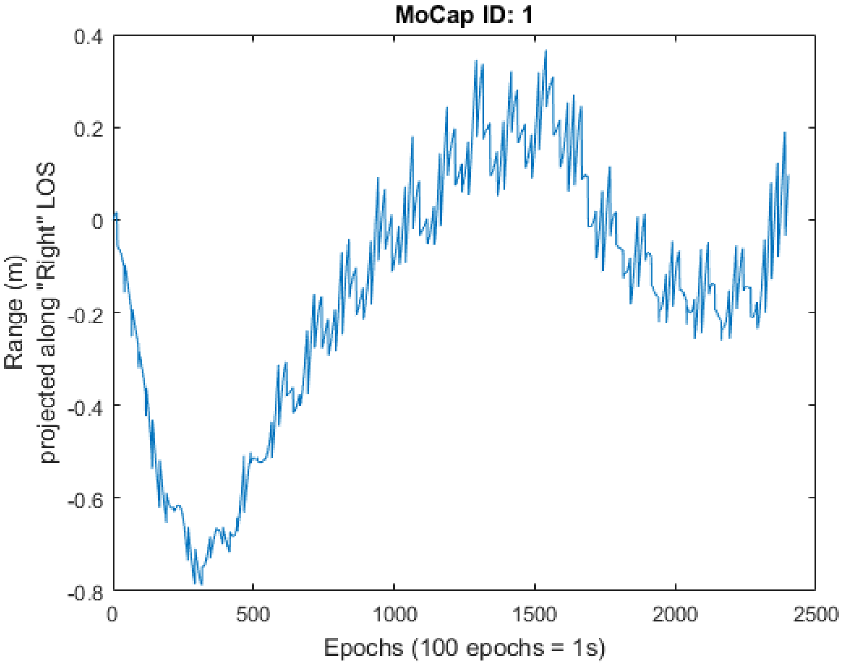

Figure B.2. Range input (m) to the SGFLL, for MoCap ID: 1

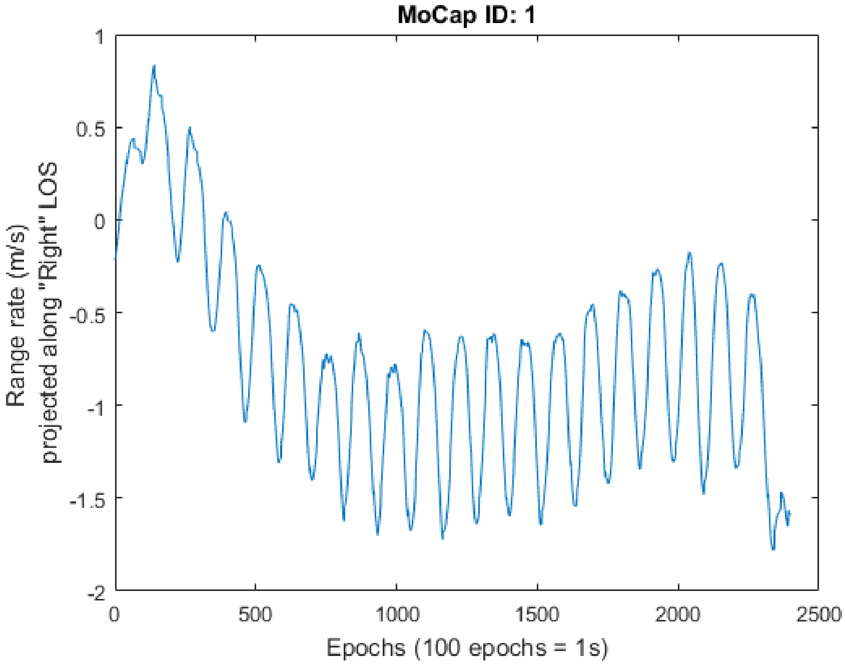

Figure B.3. Range rate (m/s) input to the SGFLL, for MoCap ID: 1

As far as Method "B" is concerned, the Xsens INS/GNSS solution contains position, velocity and attitude transients as a result of corrections applied by the Xsens Kalman filter. To remove the transients introduced by GPS measurements fusion to the integrated KF solution, several methods were explored. The first set of methods used to remove the transients from the data, are internal Matlab functions (part of the "Curve Fitting" toolbox) [139]:

- *Moving average filtering*. This method uses a number of neighbouring points, *n* points before and *n* points after the $i^{\text{th}}$ dataset point that is to be smoothed, according to:

$$y_s(i) = \frac{1}{2n+1}\left[y(i+n) + y(i+n-1) + \ldots + y(i-n)\right] \quad , \tag{B.1}$$

  where $y_s(i)$ is the smoothed value of the $i^{\text{th}}$ dataset point, noting that the total number of points used to smooth *i* (inclusive) is 2*n*+1 points. The problem in using this method is that a span of points cannot be found for the first and last points of the dataset, therefore this method cannot smooth the first and last points of the input dataset; this will result in a discontinuity, which is an artefact not representing real motion and could potentially introduce an artificial tracking slip in the GNSS carrier-tracking simulation.
- *Least-squares smoothing filter* (or curvilinear regression). This is a standard method that involves fitting to the measurements a polynomial of lower degree than the number of measurements. It is essentially interpolation method using polynomials.
- *Local regression smoothing*. This is a standard method similar to the least-squares smoothing filter, but it uses a weight function which focuses where transients occur locally. The weight function is determined by a pre-defined span of points around the central point of interest (which is to be smoothed).

For further information on the above methods, the reader may refer to [139]. These methods were not considered for further analysis in the thesis, as they exhibit a fundamental problem, i.e. they remove the corrections from the underlying real motion (even when different weights for specific points are used), which reduces the effect of the captured real motion dynamics on the simulated GNNS carrier-tracking loops (SGCTLs). In other words, they smooth the captured real motion by removing the transients, which represent corrections necessary for the accurate calculation of the final integrated navigation solution, e.g. if the heading is corrected by 3° at a given epoch, it is necessary to keep this correction in order to maintain an accurate final attitude solution. Therefore, a better approach to this problem is to spread the corrections (transients) among several epochs, instead of one, in order to smooth their effect on the real motion, but including them in the calculation of the final integrated navigation solution.

To implement this approach, a custom transients smoother was developed, which is used to spread the correction to a number of user-defined epochs before and after the epoch where the transient occurs. Therefore, it is necessary to reprocess the integrated navigation solution so that the application of the corrections is spread smoothly over all epochs instead of only occurring four or five times a second. Note that disruptive transients in the position solution arise not only from the position corrections, but also from velocity and attitude corrections integrated through the inertial navigation equations. Therefore, it is not sufficient to smooth the position in isolation.

The selected method is the custom transients smoother, because it smooths the transients without altering the underlying (real) motion, as the GPS correction on the inertial data is spread over a number of epochs (before and after the epoch where the correction is applied to), instead of just one. The custom smoother cannot compensate though cases where the inertial sensors (especially gyros, as discussed in Section 3.2) exceed their dynamic output range. Since transients affect the position, velocity and orientation solutions, a custom smoother should carry out the following 6 processing stages, (in the listed order):

1) Conversion of the Xsens navigation solution to an ECEF-frame referenced and resolved solution.
2) Detection and smoothing of attitude transients using the gyro measurements.
3) Reconstruction of the velocity solution from the accelerometer measurements, smoothed attitude solution and Xsens velocity solution.
4) Reconstruction of the position solution from the reconstructed velocity solution and Xsens position solution.
5) Derivation of a velocity solution that is consistent with the position.
6) Conversion of the position and velocity solution from ECEF-frame (referenced and resolved) to local tangent plane.

**Stage 1**

At all epochs between the user-specified start and stop times, the Cartesian ECEF-referenced and resolved position, velocity and attitude solution is obtained from the Xsens INS/GNSS position, velocity and attitude solution, $\mathbf{C}_b^n$. The position is converted from curvilinear to Cartesian using

$$\begin{aligned} \breve{x}^e_{eb} &= \left(R_E(L_b)+h_b\right)\cos L_b \cos\lambda_b \\ \breve{y}^e_{eb} &= \left(R_E(L_b)+h_b\right)\cos L_b \sin\lambda_b \quad , \\ \breve{z}^e_{eb} &= \left[\left(1-e^2\right)R_E(L_b)+h_b\right]\sin L_b \\ \mathbf{r}^e_{eb} &= \left[\breve{x}^e_{eb} \quad \breve{y}^e_{eb} \quad \breve{z}^e_{eb}\right]^{\mathrm{T}} \quad , \end{aligned} \tag{B.2}$$

where $L_b$ is the latitude, $\lambda_b$ is the longitude, $h_b$ is the height, $e$ is the eccentricity of the ellipsoid (equal to 0.0818191908425 for WGS-84 [3]) and the transverse radius of curvature $R_E$ is given by:

$$R_E(L_b) = \frac{R_0}{\sqrt{1-e^2\sin^2 L_b}} \tag{B.3}$$

where $R_0$ is the equatorial radius of the Earth (equal to 6,378,137m for the WGS-84 ellipsoid [3]).

The velocity is converted using:

$$\breve{\mathbf{v}}^e_{eb} = \mathbf{C}^e_n \mathbf{v}^n_{eb} \tag{B.4}$$

where $\mathbf{C}_n^e$ is given by:

$$\mathbf{C}_n^e = \begin{pmatrix} -\sin L_b \cos \lambda_b & -\sin \lambda_b & -\cos L_b \cos \lambda_b \\ -\sin L_b \sin \lambda_b & \cos \lambda_b & -\cos L_b \sin \lambda_b \\ \cos L_b & 0 & -\sin L_b \end{pmatrix} \tag{B.5}$$

The attitude is converted using:

$$\breve{\mathbf{C}}_b^e = \mathbf{C}_n^e \mathbf{C}_b^n \tag{B.6}$$

The position, velocity and attitude conversion is implemented in the MATLAB function NED_to_ECEF.m, provided in the accompanying CD of this thesis.

**Stage 2**

The smoothed attitude solution is provisionally set equal to the Xsens solution. Thus, for all epochs, $k$,

$$\hat{\mathbf{C}}_{b,k}^e = \breve{\mathbf{C}}_{b,k}^e \tag{B.7}$$

For all epochs, except the starting epoch, the measured attitude increment is compared with the increment in the Xsens attitude solution to identify large transients due to the application of Kalman filter corrections. The measured attitude increment is:

$$\tilde{\boldsymbol{\alpha}}_{ib,k}^b = \tfrac{1}{2}\left(\tilde{\boldsymbol{\omega}}_{ib,k-1}^b + \tilde{\boldsymbol{\omega}}_{ib,k}^b\right)\tau_i \tag{B.8}$$

where $\tau_i$ is the time interval between epochs and $\tilde{\boldsymbol{\omega}}_{ia,k}^a$ is the angular rate at epoch $k$, calculated by the Xsens software from the gyro measurements. The increment in the Xsens attitude solution is given by:

$$\breve{\mathbf{C}}_{b-}^{b+} = \breve{\mathbf{C}}_{e,k}^b \begin{pmatrix} \cos \omega_{ie}\tau_i & \sin \omega_{ie}\tau_i & 0 \\ -\sin \omega_{ie}\tau_i & \cos \omega_{ie}\tau_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \breve{\mathbf{C}}_{b,k-1}^e \tag{B.9}$$

$$\breve{\boldsymbol{\alpha}}_{ib,k}^b = \begin{array}{ll} \dfrac{\mu}{2\sin\mu}\begin{pmatrix} \breve{C}_{b-2,3}^{b+} - \breve{C}_{b-3,2}^{b+} \\ \breve{C}_{b-3,1}^{b+} - \breve{C}_{b-1,3}^{b+} \\ \breve{C}_{b-1,2}^{b+} - \breve{C}_{b-2,1}^{b+} \end{pmatrix} & \mu > 2\times10^{-5} \\ \dfrac{1}{2}\begin{pmatrix} \breve{C}_{b-2,3}^{b+} - \breve{C}_{b-3,2}^{b+} \\ \breve{C}_{b-3,1}^{b+} - \breve{C}_{b-1,3}^{b+} \\ \breve{C}_{b-1,2}^{b+} - \breve{C}_{b-2,1}^{b+} \end{pmatrix} & \mu \le 2\times10^{-5} \end{array} \tag{B.10}$$

where $\omega_{ie}$ is the Earth rotation rate and

$$\mu = \arccos\left[\tfrac{1}{2}\left(\breve{C}^{b+}_{b-1,1} + \breve{C}^{b+}_{b-2,2} + \breve{C}^{b+}_{b-3,3} - 1\right)\right] \tag{B.11}$$

noting that $\breve{\mathbf{C}}^{b}_{e,k} = \breve{\mathbf{C}}^{e\ \mathrm{T}}_{b,k}$.

The attitude discrepancy is then:

$$\delta\boldsymbol{\alpha}^{b}_{ib,k} = \breve{\boldsymbol{\alpha}}^{b}_{ib,k} - \tilde{\boldsymbol{\alpha}}^{b}_{ib,k} \tag{B.12}$$

Smoothing of an attitude transient is required whenever the following condition is met:

$$\left(\delta\alpha^{b\ 2}_{ib,k,x} + \delta\alpha^{b\ 2}_{ib,k,y} + \delta\alpha^{b\ 2}_{ib,k,z}\right) > T_{\alpha}^{2} \tag{B.13}$$

where the threshold, $T_{\alpha}$, can be determined empirically by MCS testing of values ranging e.g. between 0.001 and 0.1 rad. The optimum value should minimise the position drift while maintaining the tracking-slip effects below the applicable method tolerance.

At epochs where smoothing of an attitude transient is required, this can be achieved by spreading that transient over $s$ epochs, where $s$ is an even number less than or equal to the interval between the application of Kalman filter corrections to the Xsens attitude solution. The optimum value of $s$ can be determined empirically by MCS testing of values ranging e.g. between 4 and 30 in steps of 2. The optimum value should minimise the position drift while maintaining the tracking-slip effects below the applicable method tolerance.

The attitude corrections are determined using:

$$\Delta\boldsymbol{\alpha}^{b}_{ib,j} = \begin{array}{ll} \dfrac{j-k+0.5s+1}{s}\,\delta\boldsymbol{\alpha}^{b}_{ib,k} & k-0.5s \le j \le k-1 \\ \dfrac{j-k-0.5s}{s}\,\delta\boldsymbol{\alpha}^{b}_{ib,k} & k \le j \le k+0.5s-1 \\ 0 & \text{otherwise} \end{array} \tag{B.14}$$

$$\Delta\mathbf{C}^{e}_{b,j} = \mathbf{I}_3 + \frac{\sin\left|\Delta\boldsymbol{\alpha}^{b}_{ib,j}\right|}{\left|\Delta\boldsymbol{\alpha}^{b}_{ib,j}\right|}\left[\Delta\boldsymbol{\alpha}^{b}_{ib,j}\wedge\right] + \frac{1-\cos\left|\Delta\boldsymbol{\alpha}^{b}_{ib,j}\right|}{\left|\Delta\boldsymbol{\alpha}^{b}_{ib,j}\right|^{2}}\left[\Delta\boldsymbol{\alpha}^{b}_{ib,j}\wedge\right]^{2} \tag{B.15}$$

and then applied using

$$\hat{\mathbf{C}}^{e}_{b,j}(+) = \hat{\mathbf{C}}^{e}_{b,j}(-)\Delta\mathbf{C}^{e}_{b,j} \quad k-0.5s \le j \le k+0.5s-1 \tag{B.16}$$

where (–) and (+) denote before and after the smoothing, respectively. This enables spreading intervals to overlap where necessary. Note that spreading is applied to the smoothed attitude solution, $\hat{\mathbf{C}}^{e}_{b}$, whereas the Xsens attitude solution, $\breve{\mathbf{C}}^{e}_{b}$, is used for transient detection.

**Stage 3**

A velocity solution unaffected by large transients due to Kalman filter corrections can be determined by integrating the accelerometer measurements, transformed to ECEF-frame resolving axes using the smoothed attitude solution from stage 2. However, this will be subject to inertial drift which must be corrected. Corrections can be applied from the Xsens integrated velocity solution using a first-order loop filter to smooth out the effects of the transients in the Xsens velocity. However, if the filter gain is set too low, the corrected solution will never converge with the Xsens integrated solution. Conversely, if the gain is set too high, there will be insufficient smoothing of the transients due to the Kalman filer velocity corrections. One solution to this problem is to vary the gain. One approach is to use a high default gain, but impose a limit on the size of the corrections that can be applied in order to avoid transients large enough to cause phase tracking slips in the simulated GNSS carrier-tracking loops (explained in Sections 4.2 and 4.3).

The smoothed velocity solution is initialised with the Xsens solution. Thus, $\hat{\mathbf{v}}^e_{eb,1} = \breve{\mathbf{v}}^e_{eb,1}$. For each epoch, the velocity is updated using:

$$\mathbf{v}'^e_{eb,k} = \hat{\mathbf{v}}^e_{eb,k-1} + \left(\tfrac{1}{2}\hat{\mathbf{C}}^e_{b,k-1}\mathbf{f}^b_{ib,k-1} + \tfrac{1}{2}\hat{\mathbf{C}}^e_{b,k}\mathbf{f}^b_{ib,k} + \mathbf{g}^e_b\left(\breve{\mathbf{r}}^e_{eb,k}\right) - 2\boldsymbol{\Omega}^e_{ie}\hat{\mathbf{v}}^e_{eb,k-1}\right)\tau_i \qquad \text{(B.17)}$$

where the acceleration due to gravity, $\mathbf{g}^e_a$, is given by:

$$\mathbf{g}^e_b = -\frac{\mu}{\left|\mathbf{r}^e_{eb}\right|^3}\left\{\mathbf{r}^e_{eb} + \frac{3}{2}J_2\frac{R_0^2}{\left|\mathbf{r}^e_{eb}\right|^2}\left\{\begin{matrix}\left[1-5\left(r^e_{eb,z}/\left|\mathbf{r}^e_{eb}\right|\right)^2\right]r^e_{eb,x}\\ \left[1-5\left(r^e_{eb,z}/\left|\mathbf{r}^e_{eb}\right|\right)^2\right]r^e_{eb,y}\\ \left[3-5\left(r^e_{eb,z}/\left|\mathbf{r}^e_{eb}\right|\right)^2\right]r^e_{eb,z}\end{matrix}\right\}\right\} + \omega^2_{ie}\begin{pmatrix}1&0&0\\0&1&0\\0&0&0\end{pmatrix}\mathbf{r}^e_{eb} \qquad \text{(B.18)}$$

where $\mu$ = 3.986004418x10$^{-14}$ m$^3$s$^{-2}$ and $J_2$ = 1.082627x10$^{-3}$. The full inertial navigation equations are implemented in the MATLAB function “Nav_equations_ECEF.m”, which calls “Skew_symmetric.m” and “Gravity_ECEF.m”, provided in the accompanying CD of the thesis.

Before moving on to the next epoch, the velocity is then corrected using:

$$\delta\mathbf{v}^e_{eb,k} = \breve{\mathbf{v}}^e_{eb,k} - \mathbf{v}'^e_{eb,k} \qquad \text{(B.19)}$$

$$\Delta\mathbf{v}^e_{eb,k} = \begin{matrix} k_v\delta\mathbf{v}^e_{eb,k} & k_v\left|\delta\mathbf{v}^e_{eb,k}\right| \le \Delta v_{\max} \\ \dfrac{\Delta v_{\max}}{\left|\delta\mathbf{v}^e_{eb,k}\right|}\delta\mathbf{v}^e_{eb,k} & k_v\left|\delta\mathbf{v}^e_{eb,k}\right| > \Delta v_{\max} \end{matrix} \qquad \text{(B.20)}$$

$$\hat{\mathbf{v}}^e_{eb,k} = \mathbf{v}'^e_{eb,k} + \Delta\mathbf{v}^e_{eb,k} \qquad \text{(B.21)}$$

The optimum values for the default gain, $k_v$ and the maximum correction, $\Delta v_{max}$, can be determined empirically by MCS testing, e.g. from a range of gain values between 0.001 and 1 for the

$k_v$ gain, and a range between 0.1 and 1ms$^{-1}$ for $\Delta v_{max}$. For each tested set of values, the inertial position solution is calculated (see Stage 4). The optimum values should minimise the position drift while maintaining the tracking-slip effects below the applicable method's tolerance.

**Stage 4**

A position solution unaffected by large transients due to Kalman filter corrections can be determined by integrating the smoothed velocity solution from stage 3. However, this will be subject to inertial drift which must be corrected. Corrections can be applied from the Xsens integrated position solution using a first-order loop filter to smooth out the effects of the transients in the Xsens velocity. However, if the filter gain is set too low, the corrected solution will never converge with the Xsens integrated solution. Conversely, if the gain is set too high, there will be insufficient smoothing of the transients due to the Kalman filer position corrections. Tests have shown that it is difficult to find a suitable value. One solution to this problem is to vary the gain. One approach is to use a high default gain, but impose a limit on the size of the corrections that can be applied in order to avoid transients large enough to cause tracking slips.

The smoothed position solution is initialised with the Xsens solution. Thus, $\hat{\mathbf{r}}^e_{eb,1} = \breve{\mathbf{r}}^e_{eb,1}$. For each epoch $k$, the position is updated using:

$$\mathbf{r}'^e_{eb,k} = \hat{\mathbf{r}}^e_{eb,k-1} + \tfrac{1}{2}\left(\hat{\mathbf{v}}^e_{eb,k-1} + \hat{\mathbf{v}}^e_{eb,k}\right)\tau_i \tag{B.22}$$

Before moving on to the next epoch, the position is then corrected using:

$$\delta\mathbf{r}^e_{eb,k} = \breve{\mathbf{r}}^e_{eb,k} - \mathbf{r}'^e_{eb,k} \tag{B.23}$$

$$\Delta\mathbf{r}^e_{eb,k} = \begin{array}{ll} k_r\delta\mathbf{r}^e_{eb,k} & k_r\left|\delta\mathbf{r}^e_{eb,k}\right| \leq \Delta r_{\max} \\ \dfrac{\Delta r_{\max}}{\left|\delta\mathbf{r}^e_{eb,k}\right|}\delta\mathbf{r}^e_{eb,k} & k_r\left|\delta\mathbf{r}^e_{eb,k}\right| > \Delta r_{\max} \end{array} \tag{B.24}$$

$$\hat{\mathbf{r}}^e_{eb,k} = \mathbf{r}'^e_{eb,k} + \Delta\mathbf{r}^e_{eb,k} \tag{B.25}$$

The optimum values for the default gain, $k_r$, and the maximum correction, $\Delta r_{max}$, can be determined empirically by testing a range of gain values e.g. between 0.001 and 1 for the $k_r$ gain, and between 0.1 and 1m for $\Delta r_{max}$. The optimum values should minimise the position drift while maintaining the tracking-slip effects below the applicable method tolerance.

**Stage 5**

A velocity solution consistent with the position solution is derived using a custom cubic spline method, as explained in Appendix E.

**Stage 6**

The method of converting the position and velocity solutions from ECEF frame (referenced and resolved) to local tangent plane axes (LTP), is described below, and is implemented in Matlab function "ECEF_to_LTP.m" which is provided in the accompanying CD of the thesis.

Similar to Eq. (B.2), the origin of the LTP (originally in curvilinear coordinates in latitude $L_l$ longitude $\lambda_l$, and height $h_l$) w.r.t ECEF frame (Cartesian coordinates) can be calculated by:

$$\begin{aligned} \breve{x}^e_{el} &= \left(R_E(L_l)+h_l\right)\cos L_l \cos\lambda_l \\ \breve{y}^e_{el} &= \left(R_E(L_l)+h_l\right)\cos L_l \sin\lambda_l \\ \breve{z}^e_{el} &= \left[\left(1-e^2\right)R_E(L_l)+h_l\right]\sin L_l \end{aligned} \quad , \tag{B.26}$$

noting that the transverse radius of curvature $R_E$ is given by (B.3). The fixed attitude of the LTP w.r.t ECEF $\mathbf{C}^e_l$ can be calculated, similar to Eq. (B.5), using:

$$\mathbf{C}^e_l = \begin{pmatrix} -\sin L_l \cos\lambda_l & -\sin\lambda_l & -\cos L_l \cos\lambda_l \\ -\sin L_l \sin\lambda_l & \cos\lambda_l & -\cos L_l \sin\lambda_l \\ \cos L_l & 0 & -\sin L_l \end{pmatrix} \quad , \tag{B.27}$$

then the velocity solution of body frame *b*, referenced and resolved in LTP axes, will be:

$$\mathbf{v}^l_{lb} = \mathbf{C}^l_e \mathbf{v}^e_{eb} = \left(\mathbf{C}^e_l\right)^{\mathrm{T}} \mathbf{v}^e_{eb} \quad , \tag{B.28}$$

noting that $\mathbf{v}^e_{eb}$ is calculated from Eq. (B.21).

The position solution of body frame *b*, resolved and referenced in LTP axes will be:

$$\mathbf{r}^l_{lb} = \mathbf{C}^l_e\left(\mathbf{r}^e_{eb} - \mathbf{r}^e_{el}\right) = \left(\mathbf{C}^e_l\right)^{\mathrm{T}}\left(\mathbf{r}^e_{eb} - \mathbf{r}^e_{el}\right) \quad , \tag{B.29}$$

where $\mathbf{r}^e_{el}$ is given by Eq. (B.26) and $\mathbf{r}^e_{eb}$ by Eq. (B.25). The attitude of the body frame w.r.t. the LTP axes is:

$$\mathbf{C}^l_b = \mathbf{C}^l_e \mathbf{C}^e_b = \left(\mathbf{C}^e_l\right)^{\mathrm{T}} \mathbf{C}^e_b \quad , \tag{B.30}$$

noting that $\mathbf{C}^e_b$ is calculated using Eq. (B.16) and $\mathbf{C}^e_l$ from Eq. (B.27).

Preliminary results are shown in Figure B.4, and indicate that using method "B", a smoothed motion profile may still contain artefacts in terms of false frequency locks occurring on the SGFLL, although, compared to the results for the same MoCap profile (and LOS) shown in Figure B.1, the magnitude of the artefact is reduced. The vertical-up axis in Figure B.4, shows the relative frequency $\breve{q}^{FLL,FT}_{RM(B),1,R}$ of false frequency lock occurrences over the total number of samples (number of epochs in the motion multiplied by 1,000 runs for each added pseudorandom sequence in Matlab, as explained in Section 4.3). The axes on the horizontal plane show the simulated carrier power-to-noise density ratio $C/N_0$ measured in dB-Hz (horizontal axis) and the simulated FLL effective bandwidth ($B_{L\_CF}$) measured in Hz (vertical axis). It must be noted that the results in Figure B.4 indicate that a Monte Carlo simulation is required in order to fine-tune the parameters of the custom transients' detection and smoothing algorithm, presented in this appendix, as discussed in future work Chapter 8.

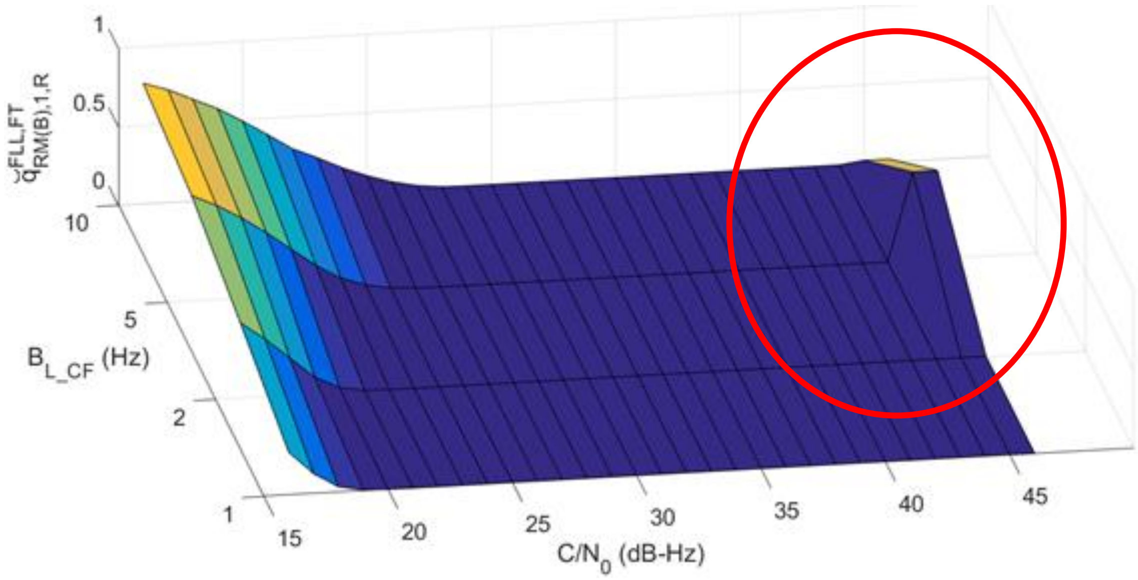

Figure B.4. Example of real motion, walking 30m in a straight line holding the sensor. The smoothed integrated navigation solution (denoted with subscript "B") can cause artificial false frequency locks to the SGFLL, highlighted in red.

## References

[1] P. D. Groves, K. Voutsis, *A Capture of Human Motion Using Xsens IMU - Algorithm design*, Spirent Communications DGP01303AAA, Issue 1-02, February 2014. [Unpublished Technical Report]

# Appendix C. Xsens MTi-G Specifications

The Xsens MTi-G IMU/GPS weights 68g and its size is 58 x 58 x 33 mm (width x length x height). The versions of the Xsens MTi-G IMU/GPS components, used in this project, are shown in Table C.2 [1], noting that Xsens has issued an end-of-life notice for this product [2].

| *Component* | *Version* | *Notes* |
|---|---|---|
| *Firmware* | 2.6.1 | This pre-installed firmware version (not upgradeable) does not support Galileo L1 signal or SBAS. |
| *MT Manager* | 1.7.0 | Associated user interface |
| *MT SDK* | 3.3 | Associated Software Development Kit |

Table C.2. Xsens MTi-G versions of components

Table C.3 provides the performance specification of the Xsens MTi-G IMU constituent parts, i.e. accelerometers, gyros, magnetometers, thermometer and barometer [1]:

| | *Accelerometers* | *Gyros* | *Magnetometers* | *Thermometer* | *Barometer* |
|---|---|---|---|---|---|
| *Units* | $m/s^2$ | deg/s | mGauss | °C | Pa |
| *Full Scale* (Units) | ±50 | ±300 | ±750 | (-55,+125) | (30, 1.2E05) |
| *Bias stability* (Units 1σ) | 0.02 | 1 | 0.1 | 0.5 | 100/year |
| *Scale factor* (% of FS) | 0.2 | 0.1 | 0.2 | <1 | 0.5 |
| *Scale factor stability* (% 1σ) | 0.03 | - | 0.5 | - | - |
| *Noise density* (Units/$Hz^{½}$) | 0.002 | 0.05 | 1.5 (1σ) | - | 4 (~0.3m/$Hz^{½}$) |

Table C.3. Xsens MTi-G versions of components

The GPS receiver update rate is 4Hz and the nominal horizontal position accuracy is 2.5m (CEP – Circular Error Probable) [1], i.e. 50% of position estimates will fall within that circle's radius assuming they follow a normal distribution, which corresponds to about 3.3m (1σ). The vertical (height) accuracy is typically three times higher than the horizontal, i.e. around 8m CEP or 10m (1σ). For this thesis, these specifications are adequate for the initial 3D position, but relative accuracy of subsequent epochs' measurements is more important.

# References

[1] Xsens. (2010). *MTi-G User Manual and Technical Documentation (Revision H).*

[2] Xsens. (2013). *Product End-of-Life Notice: MTi 3rd Generation (Document MT0316P).*

# Appendix D. Xsens Attitude Transformation

By default, Xsens MTi-G outputs the attitude of the body-frame, expressed in FLU (Forward, Left, Up) axes, w.r.t. a North-West-Up (NWU) local navigation frame. The same convention is used for the integrated velocity solution. The user may change the reference frame to a North-East-Down (NED) local navigation frame; however, this action will also change the body-frame of the sensor from FLU, to FRD (Forward, Right, Down) [1]. Therefore, to transform the attitude and velocity solutions, the following calculations need to be applied:

$$\mathbf{C}_{FRD}^{NED} = \left(\mathbf{C}_{NED}^{FRD}\right)^{T} = \left(\mathbf{C}_{FLU}^{FRD}\mathbf{C}_{NED}^{FLU}\right)^{T}, \tag{D.1}$$

where

$$\mathbf{C}_{FLU}^{FRD} = \mathbf{C}_{FRD}^{FLU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \tag{D.2}$$

Also,

$$\mathbf{C}_{NED}^{FLU} = \mathbf{C}_{NWU}^{FLU}\mathbf{C}_{NED}^{NWU} = \left(\mathbf{C}_{FLU}^{NWU}\right)^{T}\mathbf{C}_{NED}^{NWU}, \tag{D.3}$$

where

$$\mathbf{C}_{NWU}^{NED} = \mathbf{C}_{NED}^{NWU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \tag{D.4}$$

From (D.2) and (D.3):

$$\mathbf{C}_{FRD}^{NED} = \left(\mathbf{C}_{FLU}^{FRD}\left(\mathbf{C}_{FLU}^{NWU}\right)^{T}\mathbf{C}_{NED}^{NWU}\right)^{T}. \tag{D.5}$$

Doing the calculations in Eq. (D.5), it is possible to transform the attitude from $\mathbf{C}_{FLU}^{NWU}$ (default) to $\mathbf{C}_{FRD}^{NED}$:

$$\mathbf{C}_{FLU}^{NWU} = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} \Leftrightarrow \mathbf{C}_{FRD}^{NED} = \begin{bmatrix} a & -d & -g \\ -b & e & h \\ -c & f & i \end{bmatrix}. \tag{D.6}$$

Also, using Eqs. (D.2) and (D.4) it can be proven that the default velocity integrated solution of the body-frame (in FLU configuration), referenced to *e* (Earth-fixed Earth-fixed) frame axes and resolved in the default local navigation frame NWU axes, i.e. $\mathbf{v}_{e\,FLU}^{NWU}$, it is the same as $\mathbf{v}_{e\,FRD}^{NED}$:

$$\mathbf{v}_{e\,FRD}^{NED} = \mathbf{C}_{NWU}^{NED}\mathbf{v}_{e\,FLU}^{NWU}\mathbf{C}_{FRD}^{FLU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{v}_{e\,FLU}^{NWU} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \mathbf{v}_{e\,FLU}^{NWU} \quad . \tag{D.7}$$

# References

[1] Xsens. (2010). *MTi-G User Manual and Technical Documentation (Revision H).*

# Appendix E. Custom Numerical Differentiation Method

This appendix explains a custom method of differentiating velocity, constraining the mean velocity and mean acceleration between the input and output datasets to be preserved. This numerical differentiation method was developed by Paul Groves for [1], and may be applied, as appropriate, either for linear or angular velocity datasets. The underlying assumption for this method is that there is no rotation between the resolving and reference frames.

Input dataset (velocity 3D vector with each component measured in m/s):

$$\mathbf{v}_i, \qquad 0 < i < m, \tag{G.1}$$

Output dataset (acceleration 3D vector with each component measured in m/s$^2$):

$$\mathbf{a}_i, \qquad 0 < i < m, \tag{G.2}$$

Given the input and output datasets from Eqs. (G.1) and (G.2), the intermediate epochs of the output dataset can be calculated using:

$$\mathbf{a}_i = \frac{\mathbf{v}_{i+1} - \mathbf{v}_{i\text{-}1}}{2\tau_i}, \qquad 1 < i < m-1. \tag{G.3}$$

where $\tau_i$ (s) is the time interval between epochs. The coordinate frames' superscripts and subscripts have been omitted for simplicity of notation, but it is assumed that there is no rotation between the reference and resolving frames. To determine $a_0$ and $a_m$, two boundary conditions need to be applied:

- Boundary condition 1: Mean acceleration is preserved between input and output datasets.
- Boundary condition 2: Mean velocity is preserved between input and output datasets.

These two boundary conditions are presented as follows.

*Boundary condition 1:*

The mean acceleration from the input dataset is:

$$\bar{\mathbf{a}} = \frac{\mathbf{v}_m - \mathbf{v}_0}{m\tau_i}. \tag{G.4}$$

The mean acceleration from the output dataset is:

$$\bar{\mathbf{a}} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\mathbf{a}_i + \mathbf{a}_{i-1}}{2}\right) = \frac{1}{m}\left(\frac{\mathbf{a}_0 + \mathbf{a}_m}{2} + \sum_{i=1}^{m-1}\mathbf{a}_i\right) = ,$$

$$= \frac{1}{m}\left(\frac{\mathbf{a}_0 + \mathbf{a}_m}{2} + \sum_{i=1}^{m-1}\frac{\mathbf{v}_{i+1} - \mathbf{v}_{i-1}}{2\tau_i}\right) = \frac{1}{m}\left(\frac{\mathbf{a}_0 + \mathbf{a}_m}{2} + \frac{1}{2\tau_i}\left(\mathbf{v}_m + \mathbf{v}_{m-1} - \mathbf{v}_1 - \mathbf{v}_0\right)\right). \tag{G.5}$$

Equating accelerations from Eqs. (G4) and (G.5) we have

$$\mathbf{v}_m - \mathbf{v}_0 = \frac{\left(\mathbf{a}_0 + \mathbf{a}_m\right)\tau_i}{2} + \frac{1}{2}\left(\mathbf{v}_m + \mathbf{v}_{m-1} - \mathbf{v}_1 - \mathbf{v}_0\right),, \tag{G.6}$$

then rearranging Eq. (G.6):

$$\mathbf{a}_0 + \mathbf{a}_m = A = \frac{1}{\tau_i}\left(\mathbf{v}_m - \mathbf{v}_{m-1} + \mathbf{v}_1 - \mathbf{v}_0\right)., \tag{G.7}$$

*Boundary condition 2:*

Mean velocity from the input dataset is:

$$\overline{\mathbf{v}} = \frac{1}{m+1}\sum_{i=0}^{m}\mathbf{v}_i\ , \tag{G.8}$$

Mean velocity from the output dataset is:

$$\begin{aligned}
\overline{\mathbf{v}} &= \mathbf{v}_0 + \frac{1}{m+1}\sum_{i=1}^{m}\left(m+1-i\right)\frac{\left(\mathbf{a}_i + \mathbf{a}_{i-1}\right)\tau_i}{2} = \\
&= \mathbf{v}_0 + \frac{\tau_i}{2\left(m+1\right)}\left[\sum_{i=1}^{m}\left(m+1-i\right)\mathbf{a}_i + \sum_{i=1}^{m}\left(m-1\right)\mathbf{a}_i\right] = \\
&= \mathbf{v}_0 + \frac{\tau_i}{2\left(m+1\right)}\left[m\mathbf{a}_0 + \sum_{i=1}^{m-1}\left(2m+1-2i\right)\mathbf{a}_i + \mathbf{a}_m\right] = \\
&= \mathbf{v}_0 + \frac{\tau_i}{2\left(m+1\right)}\left[m\mathbf{a}_0 + \mathbf{a}_m + \sum_{i=2}^{m}\frac{\left(2m+3-2i\right)\mathbf{v}_i}{2\tau_i} - \sum_{i=0}^{m}\frac{\left(2m-1-2i\right)\mathbf{v}_i}{2\tau_i}\right].
\end{aligned} \tag{G.9}$$

Equating velocities from Eqs. (G.8) and (G.9), we have:

$$\begin{aligned}
\frac{2}{\tau_i}\sum_{i=0}^{m}\mathbf{v}_i &= \frac{2\left(m+1\right)}{\tau_i}\mathbf{v}_0 + m\mathbf{a}_0 + \mathbf{a}_m + \\
&+ \sum_{i=2}^{m}\frac{\left(2m+3-2i\right)\mathbf{v}_i}{2\tau_i} - \sum_{i=0}^{m}\frac{\left(2m-1-2i\right)\mathbf{v}_i}{2\tau_i},
\end{aligned} \tag{G.10}$$

then rearranging Eq. (G.10):

$$\begin{aligned}
& m\mathbf{a}_0 + \mathbf{a}_m = B = \\
& = \frac{1}{\tau_i}\left[2\sum_{i=0}^{m}\mathbf{v}_i - \left(m + \tfrac{5}{2}\right)\mathbf{v}_0 + \left(m + \tfrac{3}{2}\right)\mathbf{v}_1 - 2\sum_{i=2}^{m-2}\mathbf{v}_i - \tfrac{5}{2}\mathbf{v}_{m-1} - \tfrac{3}{2}\mathbf{v}_m\right] =
\end{aligned} \tag{G.11}$$

$$= \frac{1}{\tau_i}\left[2\sum_{i=0}^{m}\mathbf{v}_i - \left(m+{}^5\!/_2\right)\mathbf{v}_0 + \left(m+{}^3\!/_2\right)\mathbf{v}_1 - 2\sum_{i=2}^{m-2}\mathbf{v}_i - {}^5\!/_2\,\mathbf{v}_{m-1} - {}^3\!/_2\,\mathbf{v}_m\right] =$$

$$= \frac{1}{\tau_i}\left[-\left(m+\frac{1}{2}\right)\mathbf{v}_0 + \left(m+\frac{1}{2}\right)\mathbf{v}_1 - \frac{1}{2}\mathbf{v}_{m-1} + \frac{1}{2}\mathbf{v}_m\right].$$

Finally, combining Eqs. (G.7) and (G.11) from the above boundary conditions:

$$\begin{matrix} \mathbf{a}_0 + \mathbf{a}_m = A & (1) \\ m\mathbf{a}_0 + \mathbf{a}_m = B & (2) \end{matrix} \quad , \tag{G.12}$$

$$\begin{aligned} &(2)-(1) \Rightarrow \mathbf{a}_0 = \frac{B-A}{m-1} = \\ &= \frac{1}{(m-1)\tau_i}\left[\left(m-\frac{1}{2}\right)(\mathbf{v}_1 - \mathbf{v}_0) - \frac{1}{2}(\mathbf{v}_m - \mathbf{v}_{m-1})\right] \\ &m(1)-(2) \Rightarrow \mathbf{a}_m = \frac{mA-B}{m-1} = \\ &= \frac{1}{(m-1)\tau_i}\left[\left(m-\frac{1}{2}\right)(\mathbf{v}_m - \mathbf{v}_{m-1}) - \frac{1}{2}(\mathbf{v}_1 - \mathbf{v}_0)\right] \end{aligned} \quad , \tag{G.13}$$

or

$$\begin{aligned} \mathbf{a}_0 &= \frac{(\mathbf{v}_1 - \mathbf{v}_0)}{\tau_i} + \frac{1}{2(m-1)\tau_i}\left[(\mathbf{v}_1 - \mathbf{v}_0) - (\mathbf{v}_m - \mathbf{v}_{m-1})\right] \\ \mathbf{a}_m &= \frac{(\mathbf{v}_m - \mathbf{v}_{m-1})}{\tau_i} - \frac{1}{2(m-1)\tau_i}\left[(\mathbf{v}_1 - \mathbf{v}_0) - (\mathbf{v}_m - \mathbf{v}_{m-1})\right] \end{aligned} \quad , \tag{G.14}$$

noting that intermediate epochs of the output dataset can be calculated using Eq. (G.3).

To reconstruct the velocity and position solutions:

$$\begin{aligned} \mathbf{v}_{eb,k}^{e} &= \mathbf{v}_{eb,k-1}^{e} + \tfrac{1}{2}\left(\mathbf{a}_{eb,k-1}^{e} + \mathbf{a}_{eb,k}^{e}\right)\tau_i \qquad\qquad k \in 2...m \\ \mathbf{r}_{eb,k}^{e} &= \mathbf{r}_{eb,k-1}^{e} + \mathbf{v}_{eb,k-1}^{e}\tau_i + \tfrac{1}{4}\left(\mathbf{a}_{eb,k-1}^{e} + \mathbf{a}_{eb,k}^{e}\right)\tau_i^2 \end{aligned} \tag{G.15}$$

## References

[1] P. D. Groves, K. Voutsis, *A Capture of Human Motion Using Xsens IMU - Algorithm design*, Spirent Communications DGP01303AAA, Issue 1-02, February 2014. [Unpublished Technical Report]

# Appendix F. Matlab Algorithms

The contents of Appendix F are included on the accompanying CD with this thesis.