



Universidad de Deusto
University of Deusto

Deusto

**Advanced PDR/GNSS Fusion for Smart Devices Using
Factor Graph Optimization**

Doctoral thesis by
Amjad Hussain Magsi

A dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy within the Ph.D. Program in
Engineering for the Information Society and Sustainable Development

Supervisor
PhD. Luis Enrique Díez Blanco



Universidad de Deusto
University of Deusto

Deusto

Advanced PDR/GNSS Fusion for Smart Devices Using Factor Graph Optimization

Doctoral thesis by
Amjad Hussain Magsi

A dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy within the Ph.D. Program in
Engineering for the Information Society and Sustainable Development

Supervisor:
PhD. Luis Enrique Díez Blanco

Doctoral Student

HUSSAIN
AMJAD -
Y9273990J

Firmado
digitalmente por
HUSSAIN AMJAD -
Y9273990J
Fecha: 2026.01.19
13:18:18 +01'00'

Supervisor

DIEZ
BLANCO
LUIS
ENRIQUE -
72737939W

Firmado digitalmente por
DIEZ BLANCO LUIS
ENRIQUE - 72737939W
Nombre de reconocimiento
(DN): c=ES,
serialNumber=IDCES-72737
939W, givenName=LUIS
ENRIQUE, sn=DIEZ BLANCO,
cn=DIEZ BLANCO LUIS
ENRIQUE - 72737939W
Fecha: 2026.01.19 13:23:27
+01'00'

Bilbao, January 2026

Dedication

This doctoral thesis is dedicated with profound gratitude, love, and appreciation to my beloved parents, Engr. Abdul Razzaque Magsi and Mrs. Gulshad, whose unwavering love, sacrifices, and strength have been my guiding light and foundation.

I also dedicate this work to my devoted, caring, and beautiful wife, Sanjha Khan Pathan, whose love and support have been a constant source of encouragement, and to my beloved son, Rayyan Hussain Magsi, my little prince, my joy, and my inspiration. With heartfelt appreciation, I extend my gratitude to my brothers, Asif Ali Magsi, Iftikhar Haider Magsi, and to all my sisters, whose steadfast support and encouragement have been invaluable throughout this journey.

Declaration

I hereby declare that, with the exception of explicit references to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification at this or any other university. Except as mentioned in the text and Acknowledgements, this dissertation is entirely my own work.

Amjad Hussain Magsi
January 2026

Acknowledgments

The completion of this thesis would not have been possible without the invaluable support, encouragement, and guidance of many individuals and institutions. First and foremost, I extend my deepest gratitude to my supervisor, Dr. Luis Enrique Díez Blanco, whose exceptional mentorship, unwavering support, and inspiring intellect have profoundly shaped my PhD journey. To describe him merely as a supervisor would be an understatement. He has been a mentor in the truest sense, offering not only guidance through complex research challenges but also providing steady encouragement and wisdom during personal and professional crossroads. His ability to strike the perfect balance between challenge and support helped me grow as an independent researcher, critical thinker, and confident scholar. His thoughtful feedback, patience, and tireless belief in my potential were constant sources of motivation. In moments of uncertainty, his trust in my capabilities gave me the courage to persevere.

Beyond the academic realm, His kindness, humility, and integrity have left an indelible mark on me. I am honored to have worked under his guidance, and I will carry the lessons I have learned from him well beyond the confines of this thesis. For all the opportunities he has provided, for learning, for growth, and for becoming the researcher I am today I remain eternally grateful. Thank you, Dr. Luis Enrique, for being a mentor, a guide, and an inspiration.

I am also deeply grateful to DEIKER and the FPI Doctoral Training Programme at the University of Deusto for the opportunity and support that have been vital to my academic journey. I sincerely thank the Deusto Smart Mobility research group and DeustoTech for their support and resources throughout this work. I am also grateful to my colleagues and friends for their encouragement and companionship during this journey.

Finally, I am forever grateful to my parents, family members, and friends for their unconditional love, support, and belief in my abilities. Their constant encouragement has been my guiding force, providing me the strength to overcome every challenge I faced. I also dedicate all my accomplishments to them, and I will always strive to make them proud.

Amjad Hussain Magsi
Bilbao, January 2026

Abstract

Pedestrian navigation on smartphones mainly depends on Global Navigation Satellite Systems (GNSS) due to their accessibility and widespread adoption. However, GNSS performance deteriorates in challenging environments such as urban canyons, tunnels, under trees, and covered or partially obstructed environments, where multipath, non-line-of-sight (NLOS) effects, and atmospheric disturbances, combined with limitations of low-cost smartphone GNSS receivers, lead to significant accuracy degradation.

To overcome these limitations, this thesis first presents a systematic review of inertial and GNSS fusion methods implemented on smartphones, providing contextual background for pedestrian positioning and identifying the strengths and shortcomings of existing approaches. Building on these insights, the thesis investigates the fusion of GNSS with Pedestrian Dead Reckoning (PDR) using Factor Graph Optimization (FGO) as a robust alternative to classical Kalman Filter (KF)-based fusion. Unlike strapdown inertial navigation with low-cost MEMS, PDR offers slower drift growth and better resilience to smartphone carrying-mode variations, making it well-suited for infrastructure-free pedestrian navigation. Moreover, while PDR+GNSS integration has been studied using traditional methods, its fusion through FGO remains relatively underexplored in the literature, highlighting a research gap this thesis aims to address.

The first case study evaluates FGO against KF in real-world walking and running trials, demonstrating that FGO consistently improves horizontal accuracy by 30.6% in walking and 10% in running while offering greater robustness in GNSS-degraded conditions. In addition, this case study investigates FGO's resilience to two typical PDR error types, Short and High Errors (SHE) and Continuous and Low Errors (CLE) caused by gait variability and carrying-mode changes. Tests with ten diverse participants show that FGO outperforms both KF and a Smoothed KF (SKF), reducing mean error by about 25% compared to KF (24% compared to SKF) and enabling faster recovery after transient disturbances.

The second case study addresses FGO's computational cost by introducing two sliding-window strategies: Naïve Windowing FGO (NW-FGO) and Marginalized Windowing FGO (MW-FGO). Evaluations under GNSS multipath and PDR error conditions identify optimal window lengths (50 s for GNSS and 30 s for PDR) that preserve Batch FGO accuracy while reducing processing time from 2500 s to 38 s using MW-FGO. Marginalization consistently outperforms naïve windowing, improving mean positional accuracy by 2.18 % across tested window sizes.

The third case study examines the impact of GNSS quality on fusion accuracy by combining PDR with GNSS Single-Point Positioning (SPP) and Post-Processed

Kinematic (PPK) solutions through both FGO and KF. Experiments with a Google Pixel 4 along a 1.8 km semi-obstructed path show that integrating PPK within FGO reduces the 2D mean error by 22.5% compared to FGO with SPP (3.29 m vs. 4.25 m). FGO consistently outperforms KF under both SPP and PPK fusion strategies, although smartphone-based PPK remains at the meter level due to low ambiguity resolution. Finally, the results highlight the high computational cost of batch FGO, suggesting the need for more efficient windowed formulations for real-time applications.

In summary, this thesis advances smartphone-based pedestrian localization by (1) systematically reviewing inertial+GNSS fusion methods, (2) introducing and validating FGO as a high-accuracy alternative to KF-based fusion, and (3) developing computationally efficient windowing strategies for real-time deployment and (4) assessing the impact of higher-quality GNSS processing (PPK) when fused with PDR through FGO. The results demonstrate that FGO-based fusion delivers robust, accurate, and efficient pedestrian positioning in GNSS-challenged environments, offering practical pathways to reliable, infrastructure-free navigation on mobile and wearable platforms.

Contents

Abbreviations	xi
1 Introduction	1
1.1 Introduction	1
1.2 Research Problem	7
1.3 Thesis Objectives	8
1.4 Scope	8
1.5 Research Methodology	8
1.6 Scientific and Societal Impact	11
1.7 Scientific Contributions	11
1.8 Dissemination and Publications	11
1.9 Thesis Outline	12
2 Smartphone-Based Inertial+GNSS Fusion: State of the Art	14
2.1 Introduction	14
2.1.1 Organization of the chapter	16
2.2 Background on Inertial and GNSS Positioning in Smartphone	16
2.2.1 Inertial Positioning in Smartphones	16
2.2.1.1 Dead Reckoning (DR)	17
2.2.1.2 Inertial Navigation Systems (INS)	18
2.2.1.3 MEMS-based IMU	19
2.2.1.4 Pedestrian Dead Reckoning (PDR)	20
2.2.2 GNSS Positioning in Smartphones	24
2.2.2.1 GNSS Overview	24
2.2.2.2 Raw GNSS Observations	24
2.2.2.3 GNSS Signal Processing Techniques	25
2.2.2.4 GNSS Errors	30
2.2.3 Data Collection from Smartphones	32
2.3 Existing Survey Papers	33
2.4 Research Methodology	35
2.4.1 Research Questions	36
2.4.2 Keywords	36
2.4.3 Query	36
2.4.4 Choosing a Study	37
2.4.4.1 Stage 1 Identification	37
2.4.4.2 Stage 2 Criteria for Selection and Screening	38
2.4.4.3 Stage 3 Eligibility	38

2.4.4.4	Stage 4 Included	38
2.4.5	PRISMA Processing	39
2.4.6	Data Extraction	39
2.5	Inertial Navigation used in smartphone-based Inertial+GNSS Fusion Architectures	39
2.5.1	Inertial Navigation System (INS)	41
2.5.2	Pedestrian Dead Reckoning (PDR)	43
2.5.3	Data-Driven Inertial Navigation	45
2.5.4	Overall Analysis of Inertial Navigation:	46
2.6	GNSS used in smartphone-based Inertial+GNSS Fusion Architectures	48
2.6.1	GNSS Signals	48
2.6.2	Raw GNSS Measurements and Access	49
2.6.3	GNSS Positioning Techniques	49
2.6.4	Overall Analysis of GNSS	51
2.7	Fusion methods	53
2.7.1	Kalman Filter (KF)	53
2.7.2	Extended Kalman Filter (EKF)	55
2.7.3	Unscented Kalman Filter (UKF)	58
2.7.4	Particle Filter (PF)	60
2.7.5	Factor Graph Optimization (FGO)	62
2.7.5.1	KF vs.FGO: theoretical equivalence and practical advantages in pedestrian PDR+GNSS fusion	66
2.7.6	Artificial Intelligence and Machine Learning (AI/ML)	69
2.7.7	Overall Analysis of Fusion Methods	70
2.8	Inertial+GNSS Fusion Architectures in Smartphones	71
2.8.1	Integration Levels	72
2.8.1.1	Uncoupled (UC)	73
2.8.1.2	Loosely Coupled (LC)	73
2.8.1.3	Tightly Coupled (TC)	74
2.8.1.4	Ultra-Tightly Coupled (UTC)	74
2.8.2	INS+GNSS Fusion Architecture	75
2.8.3	PDR+GNSS Fusion Architecture	80
2.8.4	Data-Driven Inertial+GNSS Fusion Architectures	84
2.8.5	Overall Analysis of Inertial+GNSS Fusion Architectures	84
2.9	Conclusion	86
3	Case Study I: Pedestrian Motions Transient Variations	90
3.1	Introduction	90
3.2	Publication P1: KF vs. FGO under Walking and Running Conditions	90
3.2.1	Overview	90
3.2.2	Methodology	91
3.2.3	Results	91
3.3	Publication P2: KF, SKF, and FGO under Realistic PDR Error Conditions	91
3.3.1	Overview	91
3.3.2	Methodology	91
3.3.3	Results	92

3.4	Summary	92
4	Case Study II: Sliding Window Techniques for Efficient FGO	113
4.1	Introduction	113
4.2	Key Findings	114
4.2.1	GNSS-Driven Fusion Performance	114
4.2.2	PDR-Driven Fusion Performance	114
4.2.3	General Observations	114
5	Case Study III: Impact of GNSS Quality on Fusion Accuracy	137
5.1	Introduction	137
5.2	Key Findings	137
6	Conclusions and Future Works	158
6.1	Main Findings	158
6.2	Limitations	159
6.3	Future Research Directions	159
6.4	Closing Remarks	160
	Bibliography	161

List of Figures

1.1 Smartphone adoption trends from 2016 to 2025. The “smartphone penetration rate” represents the percentage of the global population that owns or actively uses a smartphone.	2
1.2 GNSS uncertainty due to NLOS and multipath	3
1.3 Inertial-GNSS fusion architecture mechanism.	5
1.4 Research methodology applied across all three case studies.	9
2.1 Inertial positioning in smartphones.	17
2.2 Strapdown Navigation Approach.	18
2.3 INS Flow Diagram	19
2.4 PDR Position Update Mechanism	21
2.5 Smartphone-based PDR architecture	21
2.6 GNSS Trilateration for 3D Position Estimation	25
2.7 SPP.	26
2.8 PPP.	27
2.9 SSR.	27
2.10 RTK.	28
2.11 SBAS.	28
2.12 GNSS Geometry.	31
2.13 Survey filtering process for smart device-based inertial+GNSS fusion surveys.	34
2.14 PRISMA Processing.	40
2.15 Distributions of collected papers over the 2016-2024.	41
2.16 EKF vs. UKF Distribution Propagation	60
2.17 Unscented Kalman Filter Loop	60
2.18 Factor graph structure for PDR+GNSS fusion.	63
2.19 Taxonomy of Smartphone-Based Inertial+GNSS Fusion Architecture Derived from Existing Literature.	72
2.20 Classification of Inertial+GNSS Fusion Architectures in Smartphone-Based Systems.	72
2.21 Uncoupled Integration Strategy.	73
2.22 LC integration level	74
2.23 TC integration level	75
2.24 UTC integration level	76
2.25 positional accuracy ranges across fusion architectures.	86

List of Tables

2.1	Comparison of PDR Components, Algorithms, and Techniques in Smartphone Implementations	22
2.2	Comparison of Absolute and Relative GNSS Positioning Techniques .	29
2.3	Android Apps for Logging Inertial and GNSS Sensor Data	33
2.4	Comparison with Existing Survey Papers Addressing Inertial+GNSS Fusion	35
2.5	Summary of Smartphone-based Navigation Methods by Mounting Position and Dimensionality	47
2.6	GNSS Positioning Techniques Used in Smartphone-Based Architectures	51
2.7	Summary and Insights from GNSS Techniques in Smartphone-Based Inertial+GNSS Fusion Architectures	52
2.8	Steps of the Particle Filter algorithm	61
2.9	Comparison of Sensor Fusion Methodologies	71
2.10	Comparison of different fusion system integration levels.	77
2.11	Summary of INS+GNSS Fusion Architectures	80
2.12	Summary of PDR+GNSS Fusion Architectures	83
2.13	Inertial+GNSS Fusion Architectures covered in Literature	85

Abbreviations

AEKF	Adaptive Extended Kalman Filter
ANN	Artificial Neural Network
ASPN	All Source Positioning and Navigation
BFGO	Batch Factor Graph Optimization
CLE	Continuous and Low Errors
DARPA	Defense Advanced Research Projects Agency
DL	Deep Learning
EKF	Extended Kalman Filter
FGO	Factor Graph Optimization
GBAS	Ground-Based Augmentation System
GNSS	Global Navigation Satellite Systems
GTSAM	Georgia Tech Smoothing and Mapping
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
iSAM2	Incremental Smoothing and Mapping (version 2)
IGS	International GNSS Services
KF	Kalman Filter
LBS	Location Based Services
LC	Loosely Coupled
LM	Levenberg–Marquardt
MEMS	Micro Electro Mechanical Systems
MWFGO	Marginalization-based Windowing Factor Graph Optimization
NLOS	Non-Line-of-Sight
NWFGO	Naive-based Windowing Factor Graph Optimization
PDR	Pedestrian Dead Reckoning
PF	Particle Filter
PPP	Precise Point Positioning
RTK	Real-Time Kinematic
RTSS	Rauch–Tung–Striebel Smoother
SBAS	Satellite-Based Augmentation System
SHE	Short and High Errors
SINS	Strapdown Inertial Navigation Systems
SKF	Smoothed Kalman Filter
SLAM	Simultaneous Localization and Mapping
SPP	Single Point Positioning
SSR	State Space Representation

TC	Tightly Coupled
UC	Uncoupled
UKF	Unscented Kalman Filter
UTC	Ultra-Tightly Coupled
WKF	Weighted Kalman Filter
ZUPT	Zero-Velocity Updates

Chapter 1

Introduction

1.1 Introduction

Positioning systems are the foundation for modern navigation services and play an important role in human mobility, urban transportation, and Internet of Things (IoT)-based applications. Accurate and reliable positioning is essential not only for enabling real-time navigation but also for optimizing resource management, reducing operational costs, and saving time in complex systems. In the context of growing urbanization and smart city development, pedestrian mobility has become increasingly important. Enhancing pedestrian movement through precise positioning and communication technologies can greatly contribute to improving the quality of life in today's connected world.

As the world rapidly urbanizes, cities become increasingly dense, characterized by high-rise buildings, complex indoor infrastructures, underground tunnels, and extensive metro systems [1]. At the same time, global population growth is contributing to larger crowds in both public and private spaces, particularly in megacities like Hong Kong, Tokyo, Beijing, and London [2, 3]. In such dynamic and congested environments, accurately determining an individual's location, whether indoors or outdoors, has become critically important. Precise localization supports a wide range of modern applications, including tracking an individual in a crowd [4], indoor navigation [5], augmented reality [6], and accessibility services for individuals with visual impairments [7].

In recent years, the attention towards precise positioning and navigation systems has grown exponentially due to the rapid development and affordability of smart devices and the technological infrastructure. Achieving precise localization has been a major concern in the scientific community due to its direct relationship with the reliability of navigational applications and location-based services.

Among available smart devices, smartphones have emerged as the most convenient and accessible platform for pedestrian navigation [8]. From Fig. 1.1, it can be seen that between 2016 and 2020, the number of smartphone users increased steadily, along with the global smartphone penetration rate, which refers to the proportion of the world's population that owns or actively uses a smartphone. By 2020, the number of smartphone users worldwide was estimated at around 3.5 billion, accounting for approximately 45% of the global population. As of 2025, there

are approximately 5.25 billion smartphone users globally, representing about 64.1% of the world's population [9]. This widespread adoption introduces the significance of smartphone-based positioning.

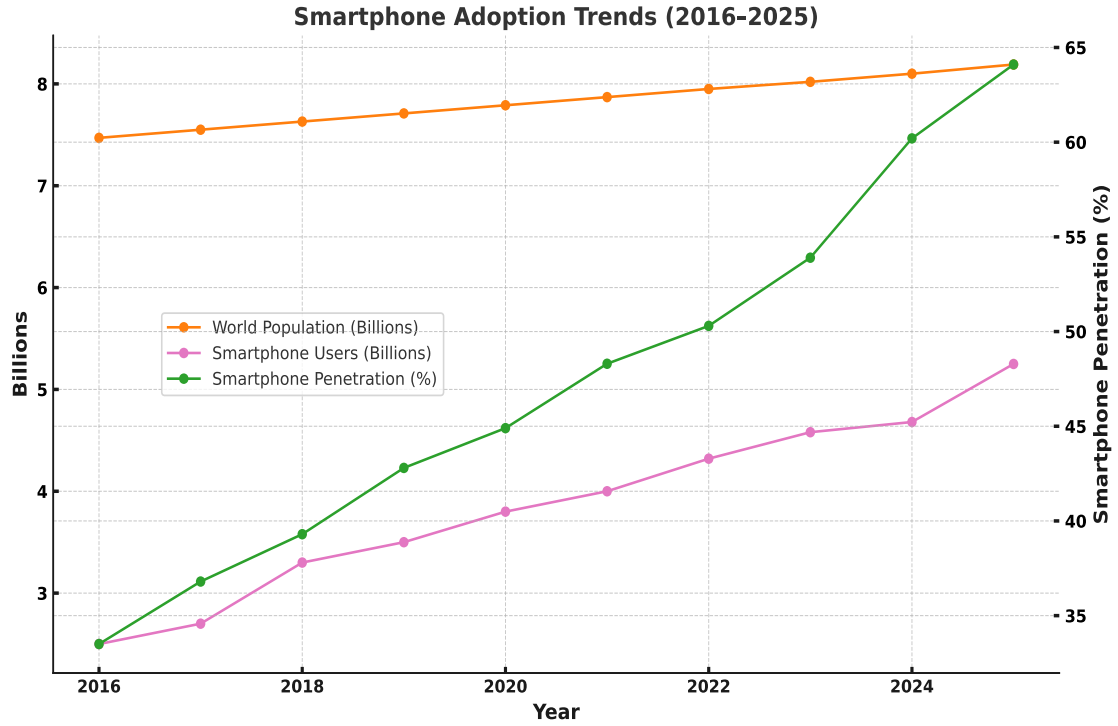


Figure 1.1: Smartphone adoption trends from 2016 to 2025. The “smartphone penetration rate” represents the percentage of the global population that owns or actively uses a smartphone.

While smartphones offer a convenient platform for pedestrian mobility, their use comes with several inherent limitations that significantly affect positioning accuracy, particularly in dense urban environments. Two major factors contribute to this performance degradation are as follows.

- First, smartphone-based positioning relies primarily on Global Navigation Satellite Systems (GNSS), which require signals from at least four satellites to compute a reliable position. However, in urban canyon areas with tall buildings and narrow streets, GNSS signals are often blocked or degraded due to Non-line of sight. As shown in Fig. 1.2, the number of visible satellites may drop to just one or two in such settings, which is insufficient for position calculation. Additionally, GNSS signal quality is further degraded by multipath effects, where signals reflect off buildings before reaching the receiver, and by atmospheric disturbances, such as ionospheric and tropospheric delays, all of which increase positional errors [10].
- Second, smartphones themselves introduce additional limitations. Unlike professional grade GNSS receivers, smartphones are equipped with low-cost GNSS chipsets that are less sensitive and more vulnerable to signal noise. Moreover, hardware related issues such as internal sensor noise, thermal drift, and system instability further impair the accuracy of positioning [11].

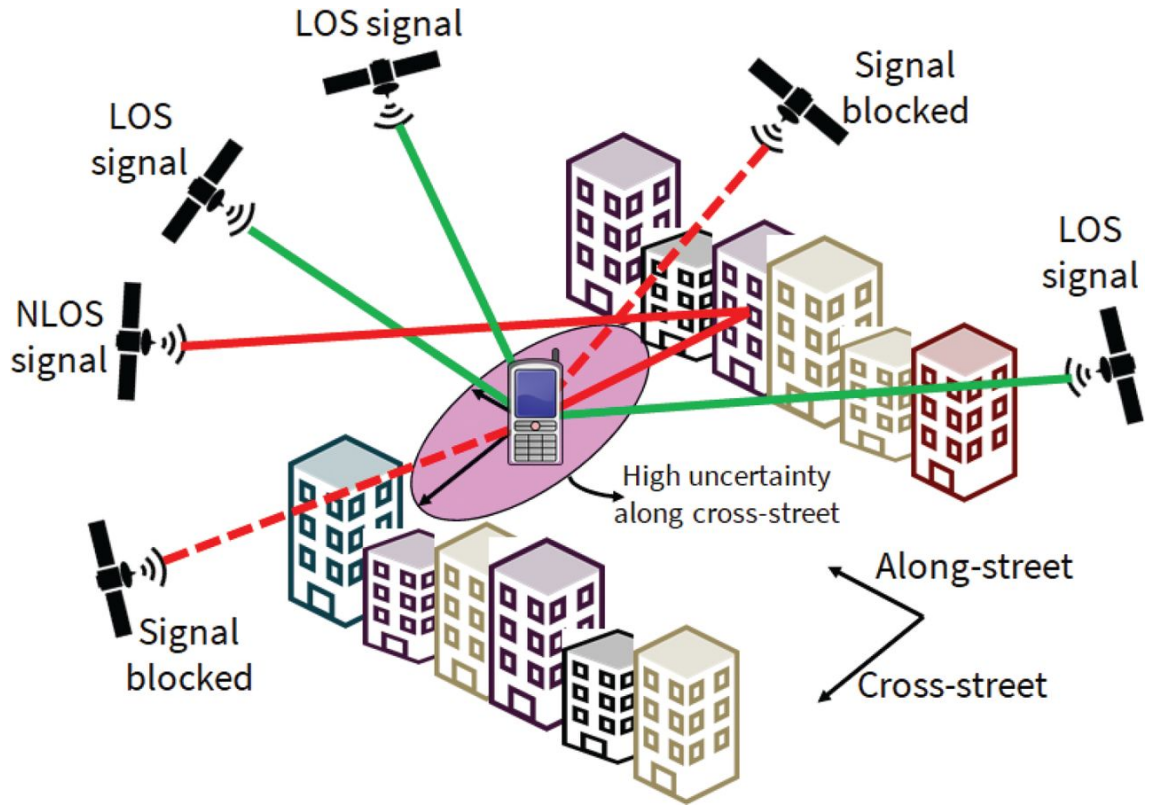


Figure 1.2: NLOS and multipath effects in GNSS satellites lead to high localization uncertainty in urban areas, especially along the cross-street direction. The pink ellipse depicts uncertainty in the receiver position estimate. This figure is adapted from Wang et al. [12].

Together, problems with GNSS signals and smartphone hardware make it difficult to achieve precise smartphone-based positioning in challenging environments.

Despite the wide adoption of GNSS in both professional-grade equipment and consumer smartphones, it remains the core technology behind most modern positioning systems. However, relying solely on GNSS is becoming increasingly unreliable due to a range of factors that limit its accuracy and reliability in many real-world conditions. The performance of GNSS is subject to various environmental and technical limitations as previously discussed and further addressed in Chapter 2. While these limitations affect all GNSS-enabled systems, their impact is especially significant in smartphones, which are constrained by compact hardware, limited antenna performance, and restricted access to high-precision correction services. As a result, achieving reliable and precise positioning on smartphones becomes especially challenging under such conditions.

Recognizing this growing GNSS limitation, the U.S. Defense Advanced Research Projects Agency (DARPA) launched the All Source Positioning and Navigation ASPN program, which explicitly states that relying solely on GNSS is no longer adequate for modern navigation needs. The ASPN initiative advocates for the integration of multiple complementary positioning technologies to create a more robust, accurate, and resilient navigation framework. In other words, instead of depending exclusively on GNSS, future positioning systems should draw on a variety of available signals and data sources to ensure reliability across diverse conditions and use cases [13].

This strategic shift has prompted a surge in global research and development focused on alternative and hybrid navigation technologies that can either support or replace GNSS when needed. Several technologies are currently being explored and implemented, each offering unique strengths. For example, Wi-Fi-based positioning uses the signal strength of nearby wireless access points to estimate a device's location, making it particularly useful in indoor environments where GNSS does not function well [14]. Bluetooth Low Energy (BLE) beacons, which are small and inexpensive transmitters, can also provide accurate indoor positioning when deployed in sufficient density [15]. Ultra-Wideband (UWB) is an emerging technology that enables high-precision short-range positioning by measuring signal travel time with centimeter-level accuracy, making it well-suited for indoor navigation and proximity-based applications [16]. Visual odometry, which involves analyzing images captured by a device's camera to estimate motion and position, is also gaining traction, mainly in robotics and augmented reality applications [17]. Additionally, LiDAR (Light Detection and Ranging) systems can map the surrounding environment in detail using laser pulses, offering strong performance in visually structured indoor or outdoor spaces [18]. Inertial sensors, such as accelerometers and gyroscopes (which are already built into all smartphones), allow for dead reckoning, estimating movement based on prior position and motion, and are especially useful when GNSS signals are temporarily lost.

These technologies are not only being considered as stand-alone alternatives but, more importantly, as complementary sources that can be integrated with GNSS to improve overall positioning performance. Wi-Fi and BLE offer strong benefits in indoor environments, which are otherwise GNSS-denied. UWB, though still emerging, shows promise for high-precision tasks like indoor navigation or device-to-device distance measurements. Meanwhile, visual odometry and LiDAR, though highly accurate, typically require more computing power and specialized sensors, making them better suited to high-end applications such as robotics, autonomous vehicles, or augmented reality rather than general smartphone use. Among these, inertial sensors are especially complementary, as they can "fill in the gaps" during short GNSS outages, providing continuous positioning when GNSS is unavailable. They offer low short-term errors and are not susceptible to signal blockage caused by multipath interference or NLOS conditions. Furthermore, they do not necessitate additional infrastructure.

In the context of smartphones specifically, the integration of these alternative technologies is essential because smartphones face both signal-based limitations and hardware constraints. Therefore, researchers are exploring practical and cost-effective solutions that can be implemented using sensors already found in modern smartphones. Among these alternatives, the fusion of inertial sensors with GNSS has emerged as a particularly effective and widely adopted solution due to the complementary nature of these two technologies. GNSS provides globally referenced absolute positioning but suffers from reduced accuracy or complete outages in NLOS and multipath conditions. In contrast, inertial navigation systems (INS), including Strapdown Inertial Navigation Systems (SINS) and Pedestrian Dead Reckoning (PDR), offer relative positioning based on onboard motion sensors, which are immune to external signal conditions but tend to accumulate error over time. When

integrated, these systems compensate for each other's weaknesses: GNSS corrects long-term drift in inertial data, while INS maintains positioning continuity during GNSS outages. This synergy makes inertial+GNSS fusion a widely recognized and robust architecture for navigation in degraded signal environments [19, 20, 21, 22].

However, the design and implementation of this fusion can vary significantly depending on several factors, as shown in Fig. 1.3. These include the type of inertial navigation approach employed, the processing techniques applied to GNSS signals, the use of multiple satellite constellations and signal frequencies, the choice of fusion algorithms, and the level of integration between the systems.

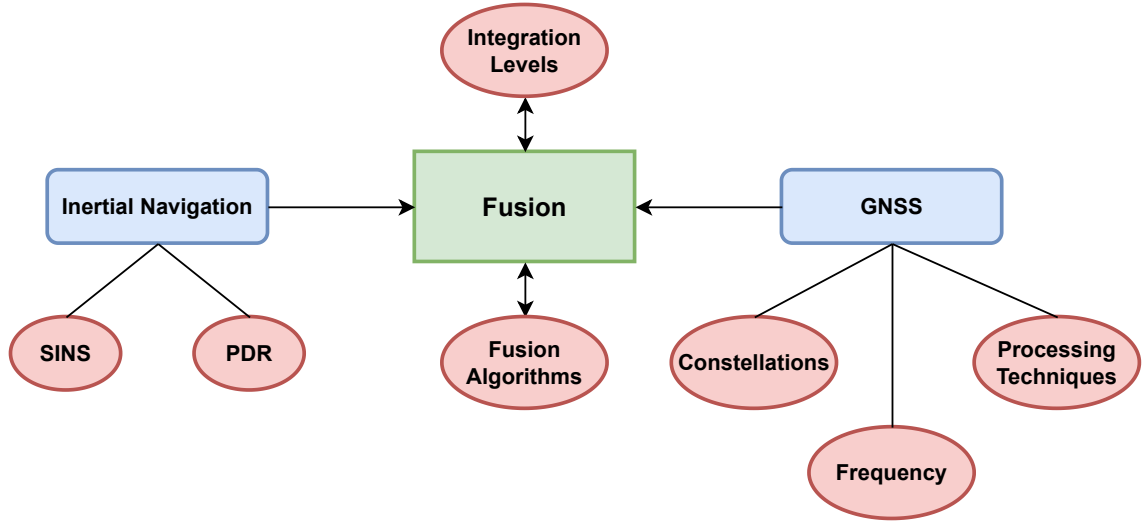


Figure 1.3: Inertial-GNSS fusion architecture mechanism.

Smartphones are commonly equipped with low-cost MEMS (Micro Electro Mechanical Systems) inertial sensors, which form the basis of their internal IMU. While these sensors offer the advantage of being compact and energy-efficient, they are also characterized by high noise levels and relatively low precision. This poses a significant limitation for traditional inertial navigation systems, that is, the SINS, which relies on continuous integration of acceleration and angular velocity to estimate position and orientation over time. In high-grade systems, SINS can deliver reliable short-term navigation. However, in smartphones, the sensor noise causes rapid accumulation of errors, leading to significant drift within seconds or minutes of operation. As a result, implementing a standalone SINS using smartphone-grade sensors is considered impractical for most real-world applications.

In addition to the sensor limitations, the way users carry smartphones introduces further challenges. Smartphones may be held in hand, kept in a pocket, placed in a bag, or mounted on a bicycle or vehicle. These varying carrying modes make it difficult to apply corrective techniques like Zero Velocity Update (ZUPT), which are commonly used in foot-mounted systems to reduce integration error. ZUPT relies on detecting periods when the sensor is stationary typically during the stance phase of a walking step to reset velocity estimates. Since smartphones are rarely fixed to the foot, this opportunity for error correction is largely unavailable in typical use scenarios [23].

To address these limitations, an alternative approach known as PDR has been

developed. PDR estimates position by detecting steps, estimating their length, and determining heading direction. Because it relies on step-wise updates rather than continuous integration, PDR is significantly less sensitive to the rapid error growth associated with SINS. While PDR still accumulates error over time, it does so more slowly and predictably, making it a more suitable choice for smartphone-based pedestrian navigation.

Given its complementary strengths, PDR is often fused with GNSS to improve overall positioning accuracy. GNSS provides absolute position updates, which can be used to correct the drift in PDR estimates, while PDR helps bridge GNSS outages. This fusion of GNSS and PDR has become a widely adopted architecture in smartphone-based pedestrian localization systems. Despite its promise, this approach continues to be an active area of research, with ongoing efforts aimed at improving step detection accuracy, adapting to different user behaviors, and developing robust sensor fusion algorithms that can operate reliably under diverse real-world conditions [24, 25, 26, 27, 28].

Traditionally, these PDR+GNSS fusion systems rely on Bayesian filtering methods, with the KF being the most commonly used algorithm. The reliability of any given positioning system largely depends on the underlying sensor technologies, fusion techniques, and the performance metrics used for evaluation. Among the various approaches, fusion algorithms play an important role and have gained significant attention. Although KF has proven effective in many scenarios, it is based on certain theoretical assumptions, specifically, a linear system model and knowledge of noise statistics. While it is commonly stated that KF assumes Gaussian-distributed noise, this is not strictly required. As clarified in recent in [29], the filter remains valid as long as the noise has finite first and second moments, that is known mean and covariance. The Gaussianity assumption mainly ensures that the KF is statistically optimal and that the posterior can be interpreted as Gaussian. In practice, however, sensor data frequently exhibits nonlinearities and non-Gaussian noise. Under these conditions, the KF still produces estimates, but they may be suboptimal or inconsistent, which can degrade performance in dynamic and complex environments [30, 31].

Moreover, these fusion methods are designed to process sensor data sequentially in a fixed update cycle. While effective in many scenarios, this approach limits the system’s ability to incorporate longer-term temporal correlations between past and present measurements. In addition, such methods often struggle to capture the complex, non-linear motion dynamics such as pedestrian behavior, especially in variable and unconstrained environments [30, 31]. These challenges highlight the need for more flexible and robust fusion methods capable of handling noisy, asynchronous, and non-Gaussian sensor data, leading to growing interest in FGO as an alternative.

As a response to these limitations, FGO has gained traction as a promising solution in the sensor fusion domain. It offers plug-and-play modularity, allowing for the flexible integration of diverse sensor types, and demonstrates strong robustness in handling highly non-linear and complex estimation problems. The first implementation of FGO in navigation was pioneered by the Georgia Institute of Technology, and since then, the method has gained traction in multi-sensor fusion applications. Unlike KF, FGO formulates the state estimation problem as a non-

linear least-squares optimization using historical measurements. It uses iterative solvers such as Gauss-Newton or Levenberg-Marquardt to refine state estimates, enabling greater accuracy and robustness, especially in the presence of GNSS errors or outliers [32].

Several researchers [33, 34, 35, 36, 37, 38] have successfully applied FGO to fuse INS and GNSS for vehicular navigation. Their findings consistently demonstrate FGO’s superior capability to handle GNSS outliers and degraded environments such as urban canyons, where satellite visibility is often restricted.

While FGO provides notable improvements over traditional filtering methods in terms of estimation accuracy and resilience to non-linearities, it introduces its own limitations particularly in the context of smartphone-based positioning. Standard FGO frameworks rely on full-batch optimization, where each estimation iteration processes the entire history of sensor measurements. Although this approach enables global consistency and high-accuracy solutions, it is computationally expensive and memory-intensive. Such demands are not well-suited for smartphones, which operate under tight constraints on processing power, memory usage, and battery life.

The continuous use of full-batch optimization in real-time applications can therefore lead to excessive energy consumption, making direct deployment of conventional FGO architectures impractical for long-term or continuous localization on mobile devices.

While positioning and localization technologies support a wide range of applications, as outlined above, the scientific contributions and experimental investigations presented in this thesis focus specifically on smartphone-based outdoor pedestrian positioning. In particular, the thesis concentrates on PDR+GNSS fusion, which is addressed in depth in the subsequent chapters. Broader references to positioning systems in this introductory section are intended to provide application context and motivation, rather than to define the scope of the thesis contributions.

1.2 Research Problem

Despite significant progress in fusion algorithms and successful applications of FGO in vehicular navigation, a critical research gap remains in applying FGO to PDR+GNSS fusion on smartphones, a configuration increasingly relevant for real-world pedestrian navigation in urban and outdoor environments.

Smartphones rely on low-cost inertial sensors that are highly susceptible to noise, drift, and instability. Additionally, step detection and heading estimation in PDR are highly sensitive to user gait, walking speed, phone placement, and sudden movements. These factors introduce significant uncertainty into the state estimation process.

Meanwhile, traditional FGO frameworks use full-batch optimization, which is computationally expensive and energy-intensive making them unsuitable for real-time localization on resource-constrained mobile devices. As a result, directly applying conventional FGO to smartphone-based PDR systems leads to poor scalability and high power consumption, limiting practical deployment.

This dissertation aims to address this gap by exploring the use of FGO for

positioning and localization on smartphones, with a focus on balancing computational efficiency and estimation robustness for real-world pedestrian mobility.

1.3 Thesis Objectives

In light of the challenges outlined above, the main objective of this thesis is as follows:

To contribute to the understanding and advancement of PDR+GNSS sensor fusion based on FGO for smartphones, with a focus on improving localization accuracy and computational efficiency in real-world conditions.

To achieve this, the thesis focuses on the following specific research objectives:

- **SO1:** To research how FGO deals with PDR errors during the fusion with GNSS
- **SO2:** To analyze and evaluate different strategies for balancing computational efficiency and positioning accuracy in FGO-based PDR+GNSS fusion on smartphones.
- **SO3:** To explore the use of advanced GNSS techniques for PDR+GNSS fusion based on FGO on smartphones

1.4 Scope

This thesis is confined to the domain of smartphone-based pedestrian navigation in outdoor environments. The scope specifically targets challenging conditions such as urban canyons, multipath-prone zones, and NLOS scenarios. The study exclusively considers the fusion of GNSS and PDR using onboard smartphone sensors, without relying on any external infrastructure or dedicated hardware.

The investigation is limited to outdoor pedestrian mobility and does not cover indoor localization or infrastructure-dependent approaches such as Wi-Fi, Bluetooth beacons, or visual odometry. Additionally, the focus remains on evaluating the comparative performance of FGO and KF fusion strategies in smartphone environments, with emphasis on positioning accuracy, system robustness, and computational feasibility under realistic usage conditions.

1.5 Research Methodology

This section presents the methodology followed throughout the research process outlined in this thesis. The study is structured around three case studies, each addressing a specific aspect of smartphone-based GNSS and PDR fusion using FGO. The overall methodological framework is illustrated in Figure 1.4.

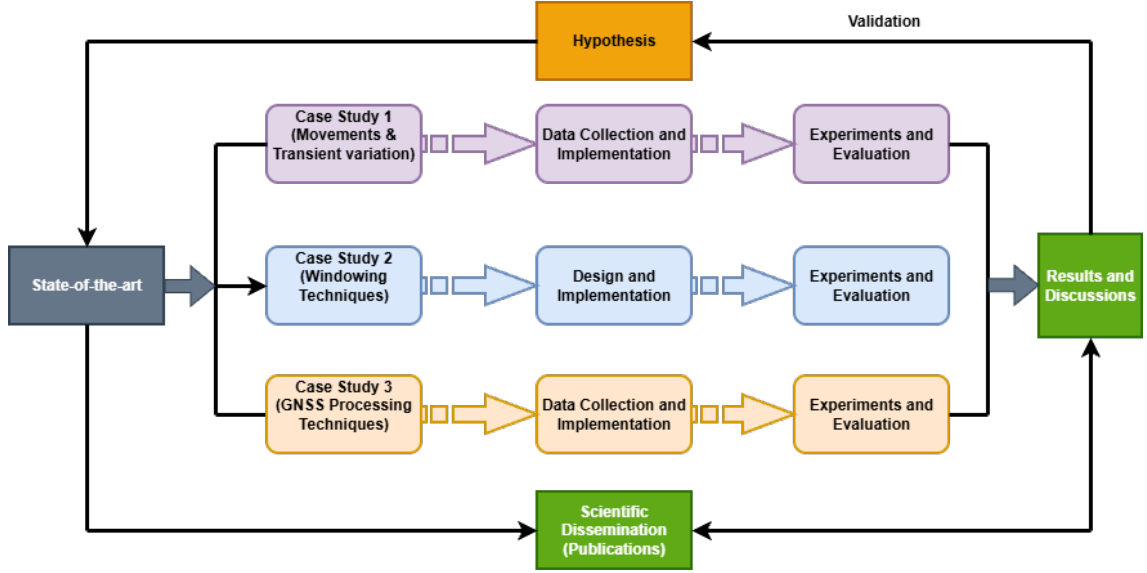


Figure 1.4: Research methodology applied across all three case studies.

State-of-the-art: The methodology begins with an in-depth review of current literature related to inertial+GNSS fusion architectures on smartphones. This includes surveying fusion architectures, algorithmic approaches, error sources, and evaluation metrics. The insights gained helped identify research gaps and guided the formation of the hypothesis and system design.

Case Study 1 – Pedestrian Motions & Transient Variations:

This case study investigates how varying pedestrian behaviors and transient handling of a smartphone affect positioning accuracy in real-world scenarios. Pedestrian data was collected using a smartphone during different motion types, specifically walking and running. In addition to these motion types, the smartphone was deliberately subjected to transient variations such as swinging, shaking, and rotating realistic disturbances that frequently occur during everyday use. These transient changes directly impact the performance of inertial sensors and challenge the stability of PDR systems.

Furthermore, to reflect human diversity, the data collection involved different individuals with varying walking styles, gait patterns, heights, and genders. These physiological and behavioural differences introduce additional variability in inertial measurements and help evaluate how PDR performance is influenced across user profiles.

Given the complexity introduced by these motion and user-induced variations, it becomes essential to examine how different sensor fusion strategies perform under such conditions. Therefore, the fusion of GNSS and PDR data was implemented using FGO and compared against the widely used KF and its smoothed variant, the Rauch–Tung–Striebel (RTS) filter. This comparison aims to evaluate the robustness, accuracy, and adaptability of each method under realistic, dynamic conditions.

Case Study 2 – Sliding Window Techniques for Efficient FGO:

This case study explores the use of sliding window-based FGO methods, which restrict the optimization to a limited, most recent subset of states and measurements, thus significantly reducing computational overhead while still maintaining accept-

able estimation accuracy. Two variants of sliding window strategies are implemented and evaluated:

1. **Naïve Sliding Window:** This method simply discards older states and measurements beyond the window size. It reduces computation but can lead to information loss, especially when handling delayed or out-of-sequence observations.

2. **Marginalization-based Sliding Window:** This is a more advanced strategy that marginalizes out the old variables in a mathematically consistent manner by transforming them into prior factors that retain essential historical information. Although slightly more complex, it preserves important correlations and supports smoother state transitions over time.

Both approaches are implemented and tested on pedestrian datasets affected by typical GNSS errors and PDR-related errors due to user motion dynamics and transient variations in smartphone orientation. The case study analyzes and compares their performance in terms of positioning accuracy, computational load, and suitability for real-time deployment on mobile devices.

Case Study 3 – Impact of GNSS Quality on Fusion Accuracy:

This case study investigates how the quality of GNSS data affects the performance of fusion architectures when combined with PDR on smartphones. It focuses on comparing SPP, which is commonly available on smartphones and represents a lower-accuracy mode, with PPK positioning, which uses correction data from a nearby base station to achieve higher precision. SPP is more prone to various errors such as multipath, atmospheric delays, and signal degradation, particularly in urban environments. In contrast, PPK offers centimeter-level accuracy but is typically used offline due to its higher computational demand. By fusing both SPP and PPK data with PDR using KF and FGO, this case study evaluates how each fusion method responds to changes in GNSS accuracy. The analysis aims to understand not only the positioning accuracy under each condition but also the robustness of each fusion approach in compensating for GNSS degradation. This comparison provides insight into how improved GNSS data impacts overall fusion reliability and highlights the adaptability of FGO relative to traditional filtering methods in practical smartphone-based navigation scenarios.

Experiments and Evaluation: Each case study includes a detailed evaluation phase where fusion performance is assessed using standard error metrics. Comparative analysis is conducted to measure the improvements offered by FGO over classical methods in various scenarios.

Results and Discussion: Findings from each case study are synthesized and compared with existing literature to validate the proposed hypothesis. The results help demonstrate the efficacy, limitations, and future potential of FGO-based fusion systems for smartphone-based pedestrian positioning.

Scientific Dissemination: The outcomes of this research have been documented and submitted for publication in peer-reviewed journals and conferences, contributing to the broader scientific understanding of inertial+GNSS fusion on smartphones.

1.6 Scientific and Societal Impact

The growing reliance on smartphones for real-time pedestrian localization has highlighted critical limitations in existing fusion systems. This research contributes to the broader goal of enabling reliable localization in urban environments where GNSS performance is degraded. Improving smartphone-based positioning systems has potential applications in intelligent transport systems (ITS), pedestrian navigation, urban safety monitoring, and augmented reality (AR). Enhancing localization robustness under diverse user behaviors and signal conditions also contributes to increased safety for vulnerable road users, such as children, elderly pedestrians, and cyclists, particularly in smart city contexts.

1.7 Scientific Contributions

The main scientific contributions of this thesis include:

- Development of a robust smartphone-based PDR+GNSS sensor fusion architecture using FGO.
- Comparative evaluation of FGO with KF and Smoothed KF under diverse pedestrian motions and smartphone handling scenarios.
- Investigation of computation-efficient sliding window techniques to reduce the processing cost of FGO on mobile devices.
- Integration and testing of advanced GNSS configurations (SPP vs. PPK) in FGO-based frameworks.

1.8 Dissemination and Publications

Contribution 1: A. Hussain and L. E. Díez, “Comparison of FGO and KF for PDR–GNSS Fusion on Smartphone for Diverse Pedestrian Movements,” *2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Nuremberg, Germany, 2023, pp. 1–6, doi: [10.1109/IPIN57070.2023.10332524](https://doi.org/10.1109/IPIN57070.2023.10332524).

Author Contributions: A. Hussain was responsible for the conceptualization, software development, experimental design, data collection, validation, result analysis, and writing of the original draft. L. E. Díez contributed through methodological guidance, critical review of the results, manuscript revision, and overall supervision of the research.

Contribution 2: A. H. Magsi and L. E. Díez, “Comparison of FGO and KF for PDR–GNSS Fusion Under Different PDR Errors,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024, Art no. 8506312, doi: [10.1109/TIM.2024.3440373](https://doi.org/10.1109/TIM.2024.3440373).

Author Contributions: A. H. Magsi led the research, including algorithm implementation, simulation and experimental evaluation, data analysis, and preparation

of the original manuscript. L. E. Díez provided scientific supervision, contributed to the interpretation of results, and reviewed and refined the manuscript.

Contribution 3: A. H. Magsi and L. E. D. Blanco, “Performance Analysis of FGO Windowing Strategies for PDR+GNSS Fusion Architecture,” *IEEE Access*, vol. 13, pp. 113461–113481, 2025, doi: [10.1109/ACCESS.2025.3584664](https://doi.org/10.1109/ACCESS.2025.3584664).

Author Contributions: A. H. Magsi conducted the study design, developed the proposed windowing strategies, performed experiments and analysis, and wrote the original draft. L. E. Díez Blanco contributed to the conceptual discussion, result interpretation, manuscript revision, and supervision of the research work.

Contribution 4: Magsi A.H., Díez L.E., Knauth S., “Continuous High-Precision Positioning in Smartphones by FGO-Based Fusion of GNSS–PPK and PDR,” *Micro-machines*, 15(9):1141, 2024, doi: [10.3390/mi15091141](https://doi.org/10.3390/mi15091141).

Author Contributions: A. H. Magsi was responsible for the conceptualization, software implementation, data collection, experimental validation, performance evaluation, and writing of the original manuscript. L. E. Díez contributed through methodology development, data analysis support, manuscript review and editing, and overall supervision. S. Knauth provided technical feedback, contributed to result validation, and participated in manuscript review and refinement.

These contributions collectively advance pedestrian navigation by introducing more accurate, adaptable, and computationally efficient fusion solutions designed for practical deployment on smartphones.

1.9 Thesis Outline

This section provides a structured overview of the content and organization of this dissertation. The research follows a problem-driven, case-study-based methodology, presented in a series of interconnected chapters that collectively support the overarching goal: improving smartphone-based pedestrian positioning through advanced fusion techniques. Each chapter is designed to address specific challenges, progressively building toward a robust and scalable localization solution. The dissertation is organized as follows:

Chapter 1 – Introduction: This chapter introduces the motivation and relevance of pedestrian localization using smartphones, particularly in GNSS-degraded environments such as urban canyons. It outlines the limitations of current GNSS and PDR systems, highlights the need for advanced fusion approaches, and presents FGO as a promising alternative to KF-based techniques. The chapter also defines the research problem, primary objective, scope, and specific objectives addressed throughout the thesis.

Chapter 2 – Literature Review: This chapter presents a systematic review of inertial and GNSS fusion architectures implemented on smartphones. This review provides the methodological and architectural background required to contextualize the pedestrian PDR+GNSS fusion approaches investigated in the subsequent

chapters.

Chapter 3 – Case Study 1: Pedestrian Motions and Transient Variations:

This chapter investigates the effect of pedestrian motion types and transient smartphone handling behaviors (swinging, shaking, rotating) on positioning performance. Using real-world data collected from different users, the study analyzes how variations in gait, speed, and smartphone orientation impact the reliability of PDR. GNSS and PDR data are fused using both FGO and KF (including the smoothed RTS variant), and the performance of each method is compared in dynamic conditions.

Chapter 4 – Case Study 2: Sliding Window Techniques for Efficient FGO:

Given the computational burden of full-batch FGO, this chapter examines two sliding window strategies, naïve and marginalization-based designed to reduce processing cost while preserving accuracy. The chapter evaluates these techniques under GNSS and PDR-related errors, quantifying the trade-offs between computational efficiency and localization performance in smartphone-constrained environments.

Chapter 5 – Case Study 3: Impact of GNSS Quality on Fusion Accuracy:

This chapter explores how the quality of GNSS measurements affects overall fusion performance. Two GNSS processing levels, SPP and PPK, are compared when fused with PDR data using both FGO and KF. The study aims to understand the extent to which improved GNSS corrections can enhance fusion accuracy and mitigate errors introduced by PDR variability.

Chapter 6 – Conclusions and Future Work: The final chapter summarizes the major findings and contributions of the dissertation. It evaluates the performance of the proposed FGO-based fusion system under different scenarios and discusses its practical implications for infrastructure-free pedestrian navigation. Limitations and open challenges are identified, and directions for future research are proposed, particularly in areas such as real-time optimization, context-aware adaptation, and integration with emerging sensor modalities.

Chapter 2

Smartphone-Based Inertial+GNSS Fusion: State of the Art

This chapter presents a systematic review of inertial and GNSS fusion architectures implemented on smartphones. The reviewed approaches include SINS-based, PDR-based, hybrid, and data-driven methods that have been applied to smartphone positioning in a variety of contexts. The purpose of this review is to provide the architectural and methodological background required to support the pedestrian-focused investigations developed in the subsequent chapters. Accordingly, while this chapter adopts a broader perspective on smartphone-based fusion methods, the scientific contributions of this thesis are explicitly limited to pedestrian positioning, with a particular emphasis on Pedestrian Dead Reckoning (PDR)+GNSS fusion.

2.1 Introduction

Accurate and reliable positioning is a critical requirement for a wide range of applications, including pedestrian navigation, health monitoring, activity recognition, urban mobility analysis, and emergency response. With the widespread availability of smartphones and smart wearable devices, pedestrian positioning has become increasingly accessible and cost-effective. These smart devices are equipped with various onboard sensors and technologies, including GNSS, accelerometers, gyroscopes, magnetometers, cameras, Wi-Fi, and Bluetooth, which can be effectively utilized to estimate location, even in complex and constrained environments [39, 8].

In recent decades, Location-Based Services (LBS) have predominantly relied on GNSS, which have become an essential part of everyday life [39]. Major GNSS include the United States' GPS, Russia's GLONASS, the European Union's GALILEO, and China's BeiDou. Additionally, regional systems such as Japan's QZSS and India's NavIC enhance positioning performance in their respective areas. Although GNSS performs well in ideal outdoor conditions, its accuracy can still be affected even in open environments. Factors such as satellite signal quality, receiver sensitivity, ionospheric and tropospheric delays, and varying environmental conditions can all impact GNSS reliability. These limitations become even more evident in challenging scenarios such as urban canyons, tunnels, and indoor environments [40]. In such GNSS-degraded areas, issues like signal blockage, multipath interference,

and NLOS reception further degrade positioning performance, making GNSS alone insufficient for accurate and reliable localization.

To overcome these limitations of GNSS-only positioning, the integration of alternative navigation and positioning technologies is increasingly preferred. This direction is also emphasized by the ASPN initiative, which highlights that relying solely on GNSS is insufficient, especially for important or safety-related applications [41, 42]. A variety of available sensors and technologies, including Wi-Fi, IMUs, BLE, cameras, and LiDAR, are capable of providing independent or complementary navigation solutions. Sensor fusion combines different sensors to overcome each other's weaknesses, improving accuracy and reliability, especially in environments where GNSS alone is unreliable. Over the years, many researchers have explored fusion techniques to enhance localization in such challenging conditions [43].

Among the alternatives available, inertial sensors are a key complementary technology to GNSS. These provide high-frequency motion data and are immune to external signal disruptions, enabling uninterrupted navigation. Their self-contained nature allows them to operate independently of any external infrastructure [44]. However, inertial sensors suffer from cumulative drift over time, leading to poor long-term accuracy. In contrast, GNSS typically offers high long-term accuracy but it is prone to short-term errors due to signal blockage, multipath effects, or atmospheric disturbances. This complementary relationship where inertial sensors excel in short-term stability and GNSS ensures long-term accuracy, forms the basis for their frequent integration in sensor fusion systems to achieve more robust and continuous positioning [45]. Due to this, inertial+GNSS fusion architectures are well-known and remain an active area of research for several decades. However, many of the proposed solutions utilize professional-grade GNSS receivers and inertial sensors, which increases the cost, size and power consumption of the system and makes it unsuitable for mass market LBS.

Although smartphones have for many years been equipped with miniaturized, cost-effective, and power-efficient GNSS receivers, MEMS-based IMUs, and various other sensors, thanks to recent technological advances, it is now possible, though still with some limitations, to use advanced GNSS and sensor fusion techniques in mass-market devices, which were previously only available in professional equipment. In particular, the possibility of accessing raw GNSS measurements on Android devices, as well as the availability of multi-constellation and multi-frequency GNSS receivers, has attracted the attention of the research community to improve positioning on smartphones. In addition, the use of machine learning techniques in the field of localization for fusion as well as for inertial navigation, GNSS outliers detection, or motion types classification has grown a lot in the last years. Also, recently, new inertial+GNSS fusion methods are being proposed in front of the classical ones, such as FGO. In summary, a large number of proposals and techniques aimed at improving the positioning on smartphones have appeared in recent years.

A review of the existing literature reveals no specific surveys focused on inertial+GNSS fusion techniques for smartphones (see Table.2.4). Existing surveys either review smartphone positioning in a general way, focus on indoor positioning, or concentrate on the positioning of robotics, unmanned aerial vehicles, and autonomous vehicle navigation with professional-grade GNSS receivers and IMUs

(Table.2.4). Therefore, it would be advisable to carry out a deep and systematic review of the inertial+GNSS fusion on smartphones to give a complete view of the state of the art of this field (architectures, GNSS techniques, fusion methods, integration levels...), identifying the unsolved problems and outlining future trends. This is both the aim and the contribution of this work.

This dissertation focuses on outdoor localization through the fusion of inertial sensors and GNSS using smart devices, such as smartphones. Therefore, the scope of the literature reviewed in this chapter is intentionally limited to works relevant to this context. The selection of papers was guided by the following baseline criteria.

- Only studies that primarily focus on outdoor scenarios are considered. Papers dedicated solely to indoor positioning or where outdoor use is secondary are excluded.
- Only papers that utilize mass-market smart devices, such as smartphones, are included in the literature.
- Only works in which inertial+GNSS fusion importance is not reduced by other information sources were included.

2.1.1 Organization of the chapter

The remainder of this chapter is organized as follows: Section 2.2 provides a general overview of inertial and GNSS positioning on smartphones. It also reviews existing survey papers on sensor fusion in different positioning scenarios and highlights how our work is unique. Section 2.4 details the research methodology used to collect and analyze the literature in line with the objectives of this survey. Section 2.5 focuses on inertial navigation techniques used in inertial+GNSS fusion architectures on smartphones, while Section 2.6 presents GNSS positioning and its associated processing methods used in inertial+GNSS fusion architectures on smartphones. Section 2.7 discusses various sensor fusion methods used. Section 2.8 presents commonly used inertial+GNSS fusion architectures from the literature. Section 2.9 provides an open discussion on current challenges and concludes the chapter with a summary of key findings and insights.

2.2 Background on Inertial and GNSS Positioning in Smartphone

In this section, some background on smartphone-based inertial and GNSS positioning and its limitations is provided.

2.2.1 Inertial Positioning in Smartphones

The term inertial positioning refers to a navigation method that tracks a device's movement and orientation using onboard inertial sensors such as accelerometers and gyroscopes. This technology enables smartphones to estimate their position,

velocity, and direction of movement without relying on an external signal. Inertial positioning is useful in GNSS-denied scenarios. It offers fast, continuous tracking and is resilient to signal loss. Notably, inertial navigation exhibits complementary behaviour to GNSS: while inertial sensors provide low short-term error but suffer from long-term drift due to error accumulation, GNSS offers consistent long-term accuracy but may exhibit high short-term fluctuations due to multipath and signal blockage. When combined, these systems complement each other, enabling more robust and accurate positioning across a wider range of scenarios [45]. Inertial positioning in smartphones can be divided into two categories, as shown in Fig. 2.1. Below is a brief discussion of inertial positioning concepts.

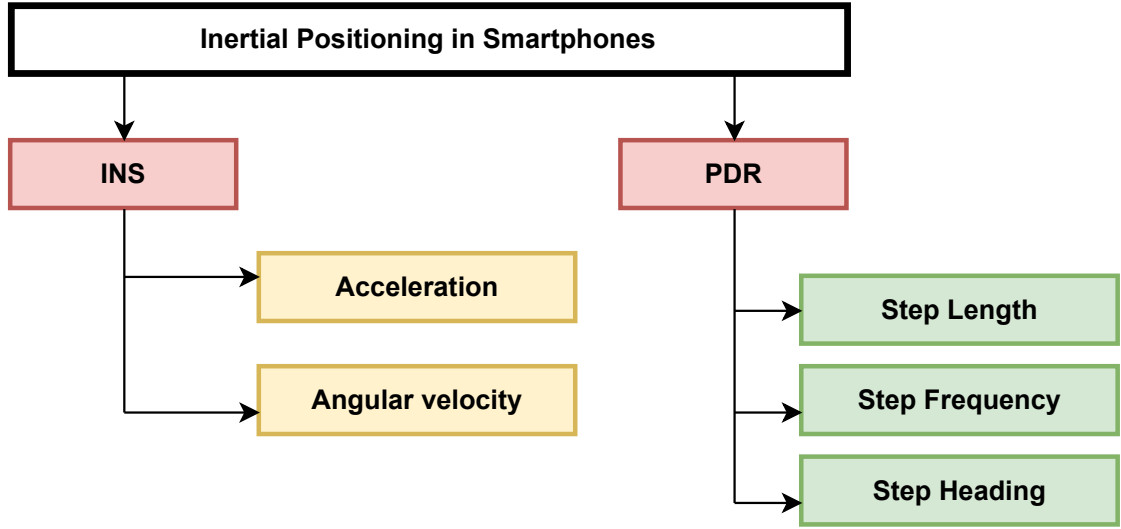


Figure 2.1: Inertial positioning in smartphones.

2.2.1.1 Dead Reckoning (DR)

Dead Reckoning (DR) is a classical navigation technique that estimates the current position based on previously known positions, heading direction, and distance travelled [43]. Historically used by sailors and nomadic travellers before the invention of GNSS, DR remains relevant in modern localization systems as a simple, low-cost method. In this chapter, dead reckoning is formulated in two dimensions, which is consistent with pedestrian smartphone positioning scenarios where horizontal motion is of primary interest.

The method uses the heading angle and displacement between previous locations to compute the current position. The heading angle ϕ_k is derived from the direction of motion, and the displacement D_k is calculated using the Euclidean distance.

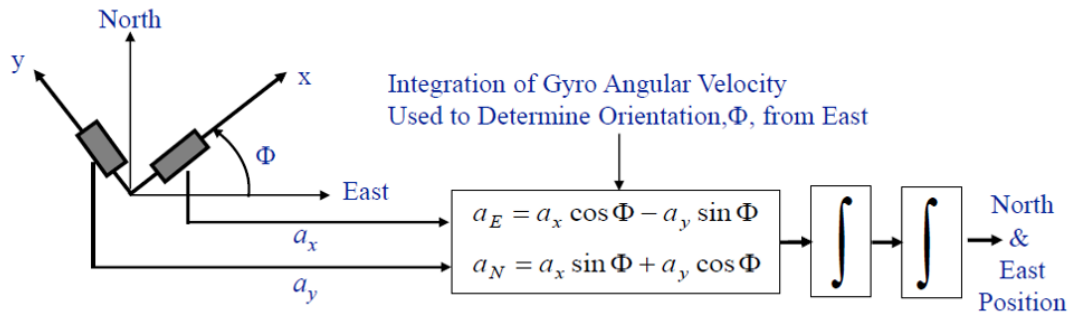
In smartphone-based applications, dead reckoning suffers from rapid error accumulation due to low-cost MEMS inertial sensors and unconstrained user motion. As a result, standalone DR is generally unsuitable for long-term pedestrian positioning and is typically employed as a component within fusion architectures.

2.2.1.2 Inertial Navigation Systems (INS)

One way of implementing the DR technique with inertial sensors is the well-known INS. INS is a self-sufficient navigation approach that tracks the location and attitude of an item in relation to a defined starting point, orientation, and velocity. In a Strapdown Navigation System (SNS), as shown in Fig. 2.2, the gyroscope and accelerometer sensors are fixed directly to the device. Because of this, the data they provide is in its own body frame, not in the global frame. To get the orientation, the gyroscope readings are integrated over time. This orientation is then used to convert the accelerometer readings into global coordinates. After that, a stable platform algorithm combines the global acceleration values to estimate the position.

However, a problem with this method is that errors grow over time. In an INS, small errors in sensor readings and integration cause the position error to increase quickly. As a result, the position error can increase rapidly over time, growing roughly with the cube of time, making long-term accuracy difficult without additional sensors or correction techniques [23].

- “Flat-Earth” Navigation



- Major functions:

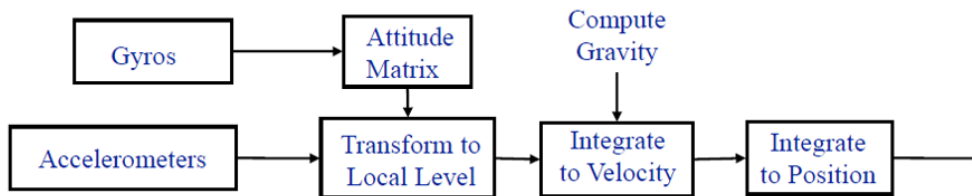


Figure 2.2: Strapdown Navigation Approach.

In smartphones, INS implementation is challenged by the limitations of MEMS-based IMUs, which, although low-cost and compact, suffer from higher noise, bias instability, and drift compared to tactical-grade sensors. Nonetheless, understanding the theoretical foundation and mechanization principles of INS remains essential to contextualize its use in consumer devices.

Mechanization Approaches: INS can be realized using two main mechanization models

1. **Platform INS:** Inertial sensors are mounted on a mechanically stabilized platform that maintains a fixed orientation relative to the Earth. This traditional configuration is less relevant for smartphones due to size and complexity.

2. **Strapdown INS:** More relevant to smartphone applications, this approach involves rigidly fixing the sensors to the body of the device. Orientation changes are tracked computationally using gyroscope outputs. Strapdown systems are computationally efficient but highly sensitive to sensor errors.

Figure 2.3 outlines the inertial navigation pipeline, including sensor data acquisition, transformation, integration, and resulting state estimation:

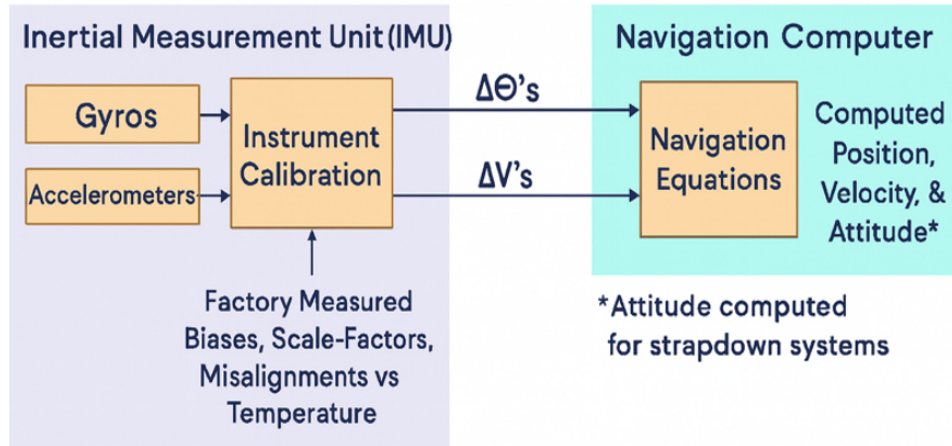


Figure 2.3: INS flow diagram: computation of attitude, velocity, and position from raw inertial sensor measurements [23].

While strapdown INS provides accurate navigation in vehicular and aerospace applications, its direct use on smartphones is severely limited by sensor noise, frequent device reorientation, and unconstrained pedestrian motion. These limitations motivate alternative inertial strategies, such as pedestrian dead reckoning, which better exploit motion regularities in human walking.

2.2.1.3 MEMS-based IMU

IMUs combine inertial sensors, typically accelerometers and gyroscopes to measure motion. In smartphones, these sensors are implemented using MEMS technology, making them compact, lightweight, low-cost, and ideal for widespread use in commercial smart devices.

MEMS-based IMUs provide acceleration and angular velocity data, which are used to estimate displacement and orientation through integration, commonly referred to as dead reckoning. Although IMUs do not give absolute position, they allow relative positioning when initialized from a known location. Over time, however, integration errors accumulate due to sensor drift and bias, especially in low-cost MEMS devices.

Compared to professional-grade IMUs, smartphone-based MEMS IMUs offer both advantages and limitations:

- **Cost:** Smartphone IMUs cost less than \$15, while professional-grade systems range from \$1,000–\$5,000. High-end alternatives like Fiber Optic Gyroscopes (FOGs) can exceed \$10,000

- **Size and Weight:** MEMS IMUs are extremely small (often $<1 \text{ cm}^3$ and $<1 \text{ g}$), ideal for portable use, whereas professional-grade IMUs are bulkier and more rugged.
- **Power Efficiency:** Designed for mobile devices, MEMS IMUs consume minimal power compared to their high-end counterparts.
- **Accuracy and Drift:** MEMS IMUs have higher noise and bias drift, resulting in greater long-term error. Professional-grade IMUs maintain better accuracy through environmental compensation and good calibration.
- **Accessibility:** MEMS-based IMUs are found in nearly all smartphones, enabling large-scale deployment and practical research.

Because of their cost-effectiveness, low power usage, and availability, MEMS IMUs are widely used in real-world pedestrian navigation.

2.2.1.4 Pedestrian Dead Reckoning (PDR)

PDR is a navigation method specifically adapted to estimate a pedestrian's position by incrementally computing displacements from a known starting point. Unlike vehicle-based dead reckoning, which typically integrates continuous velocity and direction data, PDR relies on modeling human motion step-by-step.

PDR operates based on three components: (i) step detection, (ii) step length estimation, and (iii) heading estimation. These measurements are primarily obtained from inertial sensors, which can range from MEMS-based IMUs (commonly found in smartphones) to high-grade professional units, depending on the application [46].

PDR Mechanism and Equations: The basic PDR estimation model updates the user's position based on detected steps and estimated step vectors. As shown in Fig. 2.4, the incremental displacement between positions at steps k and $k + 1$ can be modeled as:

Let $\mathbf{POS}_k = [x_k, y_k]^\top$ denote the two-dimensional pedestrian position vector at step k , and let $\Delta\mathbf{POS}_{k+1}$ represent the incremental displacement between steps k and $k + 1$.

$$\mathbf{POS}_{k+1} = \mathbf{POS}_k + \Delta\mathbf{POS}_{k+1} \quad (2.1)$$

$$\Delta\mathbf{POS}_{k+1} = S_{k+1} \begin{bmatrix} \cos(\theta_{k+1}) \\ \sin(\theta_{k+1}) \end{bmatrix} \quad (2.2)$$

where S_{k+1} is the estimated step length and θ_{k+1} is the step heading. This approach shifts the burden from continuous integration of raw acceleration to discrete updates per step, which are more robust to sensor drift.

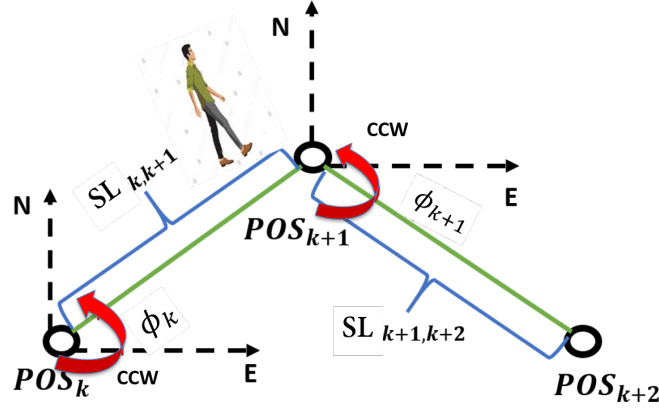


Figure 2.4: PDR position update based on step length and heading estimation [24].

MEMS IMU-Based PDR Implementation: The actual implementation of PDR on smartphones builds upon the principles above but introduces practical challenges and adaptations. A general pipeline is shown in Fig. 2.5, where PDR modules are constructed on top of smartphone APIs and sensor streams. Table 2.1 shows a comparison of PDR components, algorithms, and techniques in smartphone implementations.

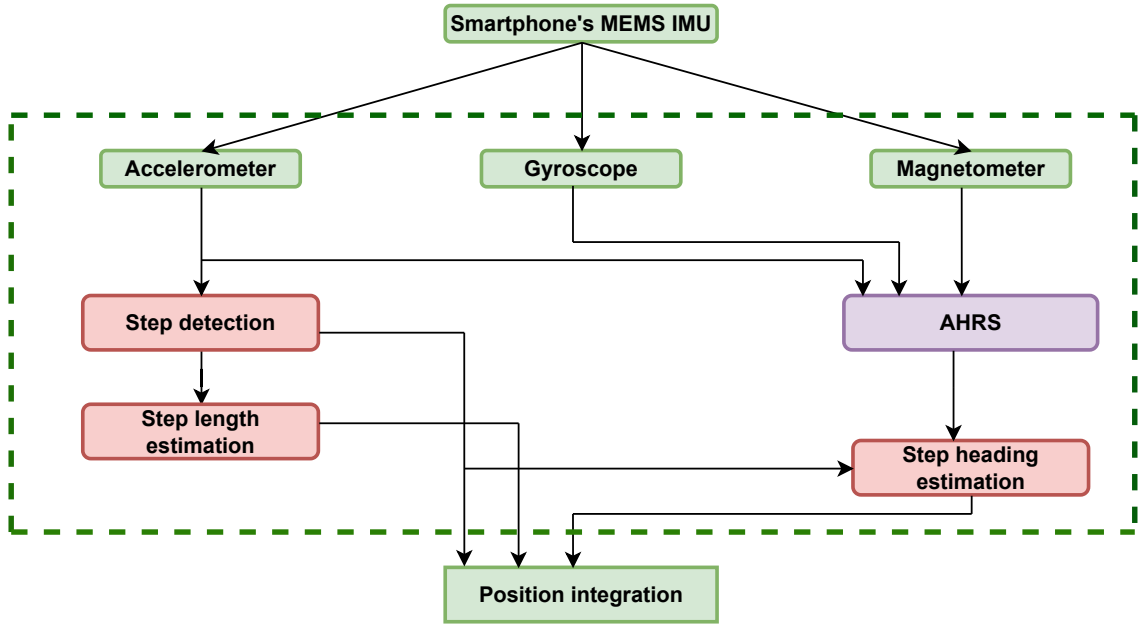


Figure 2.5: Smartphone-based PDR architecture: IMU data is processed to detect steps, estimate heading and step length, and update position [47].

A variety of PDR techniques have been proposed for step detection, step length estimation, heading determination, and height/elevation changes, each offering different trade-offs in complexity, robustness, and accuracy. Table 2.1 summarizes the most widely used PDR components, algorithms, and smartphone-based implementation techniques, highlighting their advantages, limitations, and overall reliability.

As shown in Table 2.1, different PDR processing stages rely on algorithms of varying complexity and robustness. For example, peak-based step detection and zero-crossing approaches are computationally efficient and suitable for smartphones,

Table 2.1: Comparison of PDR Components, Algorithms, and Techniques in Smartphone Implementations

PDR Step	Technique / Algorithm	Description / Equation	Accuracy / Reliability Notes
Step Detection	• Peak-Valley Detection • Zero-Crossing • FSM (Finite State Machine)	Peak-Valley: detects local maxima/minima in $a_{mag} = \sqrt{a_x^2 + a_y^2 + a_z^2}$. Zero-Crossing: uses sign changes in vertical acceleration. FSM: models step cycle via states.	Peak-Valley: Accurate on flat surfaces, sensitive to noise. FSM: More robust across varied gait patterns but needs calibration. Zero-Crossing: Prone to false positives in irregular terrain.
Step Length Estimation	• Weinberg’s Model • Regression Models • Machine Learning (NN, SVM) • Constant Step Size	Weinberg: $S_k = k \cdot \sqrt{a_{peak} - a_{min}}$ Regression: fits features like frequency, peak mag, duration. ML: uses labeled training data for step prediction.	ML: High accuracy if well-trained, but device/user dependent. Weinberg: Simple, moderate accuracy. Constant: Low accuracy, error grows rapidly.
Heading Estimation	• Madgwick Filter • Mahony Filter • Android Rotation Vector API	Fuses accelerometer, gyroscope, magnetometer. AHRS outputs device orientation θ_k , projected to ENU.	Madgwick: Good real-time accuracy, low computation. Mahony: More stable in dynamic motion. API: Fast but less transparent and harder to tune.
Height / Altitude Estimation	• Barometric (2.5D) • Vertical Acceleration (3D)	Barometric: $\Delta h = \frac{T}{C} \cdot \ln\left(\frac{P_0}{P}\right)$ 3D: integrates vertical a_z to height.	Barometric: Good for stair/floor detection, sensitive to weather. 3D: High drift, rarely used alone.
Smartphone Constraints	• Carrying Position • Sampling Jitter • Power Management	Pocket, hand, waist, bag = different motion profiles. Android OS introduces variable sampling and sensor fusion issues.	Performance drops significantly with handheld use. Sensor dropout, latency, and placement cause inconsistencies.

while machine-learning-based methods may provide improved robustness at the cost of higher computational load. Step-length estimation methods also vary widely: empirical and regression-based models are simple but device- and user-dependent, whereas biomechanical or adaptive filtering approaches can provide higher accuracy under diverse walking conditions. Heading estimation remains one of the most challenging components, as magnetometer-based heading is sensitive to environmental disturbances; therefore, many smartphone systems fuse gyroscope and accelerometer signals to improve stability.

Overall, the table highlights that smartphone-based PDR systems must balance computational cost, sensor noise, and environmental variability, leading to a diverse set of algorithmic choices depending on the target application, accuracy requirements, and device constraints

Comparison with INS:

While both PDR and INS rely on inertial sensors, their mechanisms, assumptions, and limitations differ significantly, especially when implemented on smartphones:

- **Error Accumulation:** *INS* suffers from exponential error growth due to continuous integration of acceleration and angular velocity. In contrast, *PDR* uses discrete updates per step, which makes it more robust against drift over time.
- **Sensor Requirements:** *INS* requires high sensor accuracy and precise calibration (bias, scale, misalignment), which is often not feasible with consumer-grade MEMS. *PDR* can tolerate lower-quality sensors by relying on repeatable patterns like walking.
- **Motion Constraints:** *INS* assumes arbitrary motion in 3D space, while *PDR* assumes bipedal step-based movement. Therefore, *INS* can be used in any scenario (including vehicles), while *PDR* is specifically tailored to pedestrian navigation.
- **Complexity:** *INS* involves more complex mechanization equations, orientation estimation, and sensor fusion. *PDR* models are simpler and more interpretable, especially when relying on handcrafted step detection and length estimation.
- **Drift and Stability:** *INS* systems without correction fail within seconds due to drift, especially in smartphones. *PDR*, when well-calibrated, can achieve acceptable accuracy over longer durations, particularly in indoor or GNSS-degraded conditions.
- **Implementation Feasibility on Smartphones:** *PDR* is more practical for smartphones due to computational efficiency, robustness to noise, and independence from absolute positioning. *INS*, on the other hand, requires advanced filtering, synchronization, and often fusion with GNSS or other sensors to remain viable.
- **Use Cases:** *INS* is preferable for vehicle-based scenarios with stable mounting and consistent dynamics. *PDR* is preferred for human-centric applications such as indoor navigation, activity recognition, or pedestrian mobility analysis.

PDR offers more flexibility, as inertial sensors can be positioned in various locations on the body such as handheld, in a pocket, on the wrist, or in a backpack making it ideal for widespread use. Compared to traditional *INS*, which continuously integrates acceleration and angular velocity, *PDR* has a more controlled error growth pattern. Specifically, *PDR*'s error tends to grow linearly with the number of steps, which is comparable to foot-mounted *INS* systems employing zero-velocity updates (ZUPT). In contrast, traditional strapdown *INS* without such correction mechanisms typically exhibits cubic error growth due to compounding integration drift.

However, as with all dead reckoning techniques, *PDR* is not without limitations. Its independence from external signals also means it cannot correct accumulated errors. For example, sensor bias, miscalibration, or incorrect step estimation can lead to position drift that cannot self-correct without external updates (e.g., from GNSS or map constraints). Additionally, *PDR* can struggle with non-walking activities such as elevator use, cycling, or rapid directional changes. Despite these challenges,

PDR remains a practical solution for short-to-medium-range localization and is often fused with GNSS or other technologies to enhance long-term accuracy.

2.2.2 GNSS Positioning in Smartphones

Modern smartphones utilize GNSS as a core technology for navigation and location-based services, enabling faster signal acquisition and enhanced reliability. This section provides an overview of GNSS technology as used in smartphones, covering signal structure, satellite systems, data processing techniques, and common limitations in real-world conditions.

2.2.2.1 GNSS Overview

GNSS are composed of orbiting satellites that transmit precise timing and positioning signals to Earth. By receiving signals from at least four satellites, a GNSS receiver estimates its 3D position (latitude, longitude, altitude) and clock bias through techniques such as trilateration based on pseudorange measurements. Each pseudorange reflects the distance between the satellite and the receiver, computed by comparing the satellite's transmission time with the receiver's clock. The solution involves solving a system of nonlinear equations using known satellite positions and estimated time of flight [39].

$$\rho_i = c(t_r - t_{si}) = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + b_u \quad (2.3)$$

Where:

- ρ_i is the pseudorange to satellite i
- c is the speed of light
- t_r and t_{si} are the receiver and satellite transmission times
- (x_i, y_i, z_i) are satellite coordinates
- (x_u, y_u, z_u) are user coordinates (unknown)
- b_u is the receiver's clock bias

Solving four such equations allows the receiver to compute its 3D position and clock offset. Figure 2.6 visually illustrates this process.

2.2.2.2 Raw GNSS Observations

Most traditional GNSS applications rely on processed outputs such as position, velocity, and time (PVT). These values are computed internally by the GNSS chipset and offer limited flexibility for advanced research or high-precision applications [49].

However, raw GNSS observations provide direct access to the unprocessed measurements received from satellites. These include:

- **Pseudorange:** Estimated distance from the satellite to the receiver based on the time-of-flight of the signal.

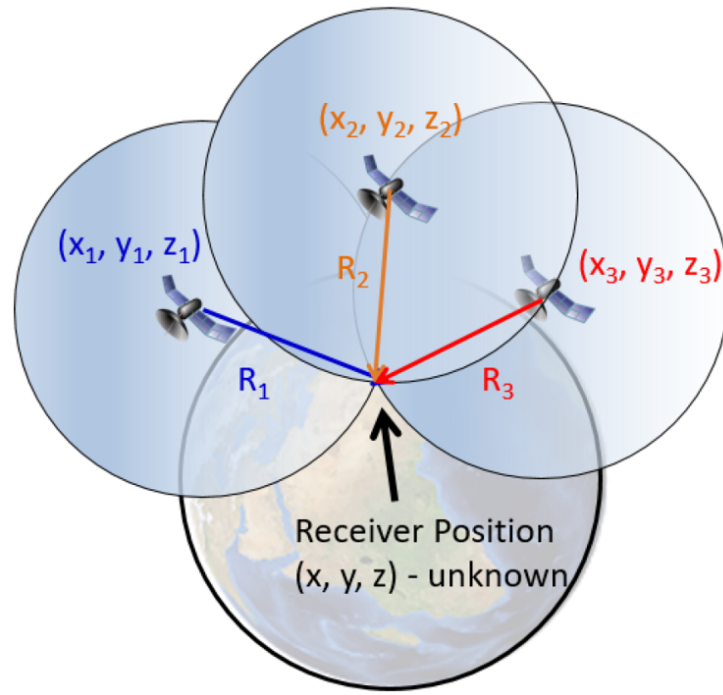


Figure 2.6: 3D position estimation via GNSS trilateration [48].

- **Carrier-phase:** The phase of the received carrier wave, used for highly accurate positioning techniques like RTK and PPP.
- **Doppler shift:** Frequency changes due to the relative motion between satellite and receiver, useful for velocity estimation.
- **Carrier-to-Noise Density Ratio (C/N_0):** Indicates signal quality and strength.
- **Satellite Metadata:** Includes satellite ID, transmission time, and orbital parameters.

Raw GNSS measurements enable detailed control over positioning computations and make advanced algorithms feasible on modern devices. Its introduction represents a key enabler for smartphone-based high-precision navigation research.

2.2.2.3 GNSS Signal Processing Techniques

GNSS positioning techniques are generally classified into two broad categories:

- **Absolute Positioning:** These methods determine the user's location directly from satellite signals without requiring a local reference station. Examples include SPP, PPP, Satellite-Based Augmentation System (SBAS), and State Space Representation (SSR).
- **Relative Positioning:** These techniques rely on at least one reference station at a known location. The receiver compares its observations with those from the reference station to correct errors. Examples include Real Time Kinematic (RTK), Differential Global Positioning System (DGPS), and Ground-Based Augmentation System (GBAS).

A brief explanation of these techniques is provided below, and Table 2.2 summarizes the key characteristics of each method.

Single Point Positioning (SPP) is a fundamental GNSS technique used to determine a user's position by processing signals from multiple satellite constellations such as GPS, GLONASS, Galileo, and BeiDou. SPP operates with a single GNSS receiver and utilizes the broadcast ephemeris and clock data transmitted by satellites to compute the receiver's position. It relies on code-based measurements (pseudoranges) and does not incorporate correction data from external sources, making it suitable for applications where real-time, stand-alone positioning with meter-level accuracy is sufficient. Despite its simplicity and broad accessibility, SPP is limited in precision compared to advanced techniques such as differential GNSS or Precise Point Positioning (PPP). Figure 2.7 illustrates the typical flow of the SPP process, from satellite signal acquisition to the generation of the final position output.

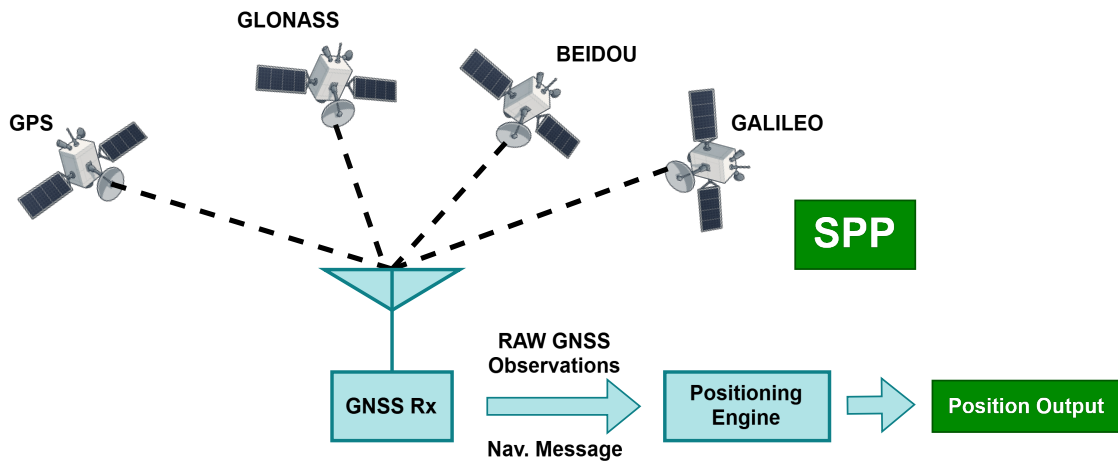


Figure 2.7: SPP.

Precise Point Positioning (PPP) is an advanced GNSS positioning technique that provides high-accuracy location estimates using a single receiver. Unlike SPP, PPP incorporates both code and carrier-phase GNSS measurements, along with precise satellite orbit and clock correction data, typically obtained from global or regional correction services. This enables PPP to achieve sub-meter to centimeter-level accuracy without requiring a local base station. The positioning engine processes both raw GNSS observations and the external corrections to produce a highly accurate position output. Due to its reliance on precise corrections and complex processing, PPP is commonly used in geodetic, surveying, and scientific applications where high precision is essential. Figure 2.8 illustrates the PPP architecture, highlighting the integration of satellite data and correction inputs to generate precise positioning results.

State Space Representation (SSR) is an advanced correction method used in precise GNSS positioning, including applications such as PPP. Unlike traditional PPP, which uses pre-computed correction files, SSR delivers real-time, modular corrections for satellite orbits, clocks, and atmospheric delays. These corrections are transmitted to the GNSS receiver, which then integrates them with raw satellite observations to compute a highly accurate position. SSR improves convergence time

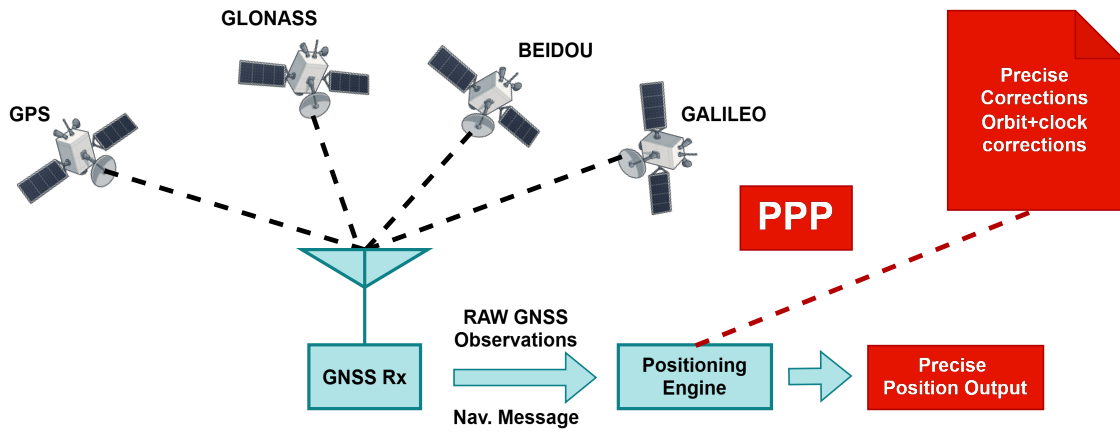


Figure 2.8: PPP.

and allows for scalable, real-time positioning services. Figure 2.9 illustrates the SSR-based positioning architecture, showing the integration of both precise and real-time state-space corrections into the positioning engine to produce accurate output.

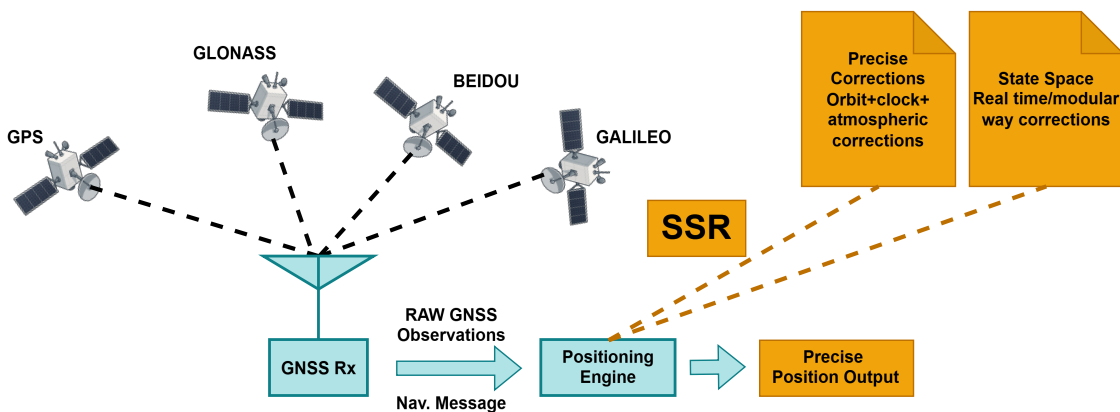


Figure 2.9: SSR.

Real Time Kinematics (RTK) is a high-precision GNSS positioning technique that relies on carrier-phase measurements and real-time corrections from a nearby base station. The base station transmits precise correction data to the rover (user) receiver, enabling it to eliminate or greatly reduce common positioning errors such as atmospheric delays and satellite clock inaccuracies. RTK is capable of delivering centimeter-level accuracy within seconds, making it ideal for applications requiring real-time precision, such as surveying, autonomous vehicles, and precision agriculture. Figure 2.10 illustrates the RTK workflow, where the GNSS receiver uses satellite signals and real-time corrections from a base station to compute an accurate position output.

SBAS enhances GNSS positioning accuracy, integrity, and reliability by providing real-time correction data through geostationary satellites. SBAS relies on a network of ground reference stations that monitor GNSS signals and compute corrections for errors such as satellite clock drift, orbit inaccuracies, and ionospheric delays. These corrections are sent to SBAS geostationary satellites, which then

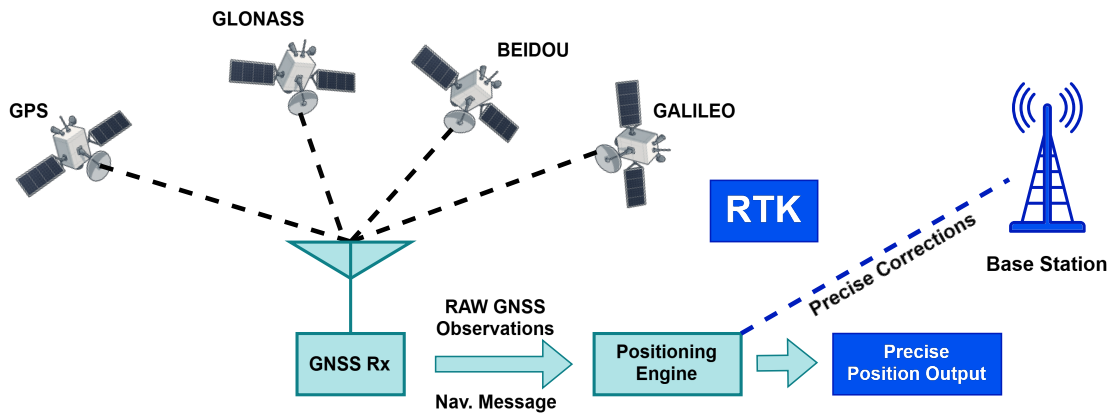


Figure 2.10: RTK.

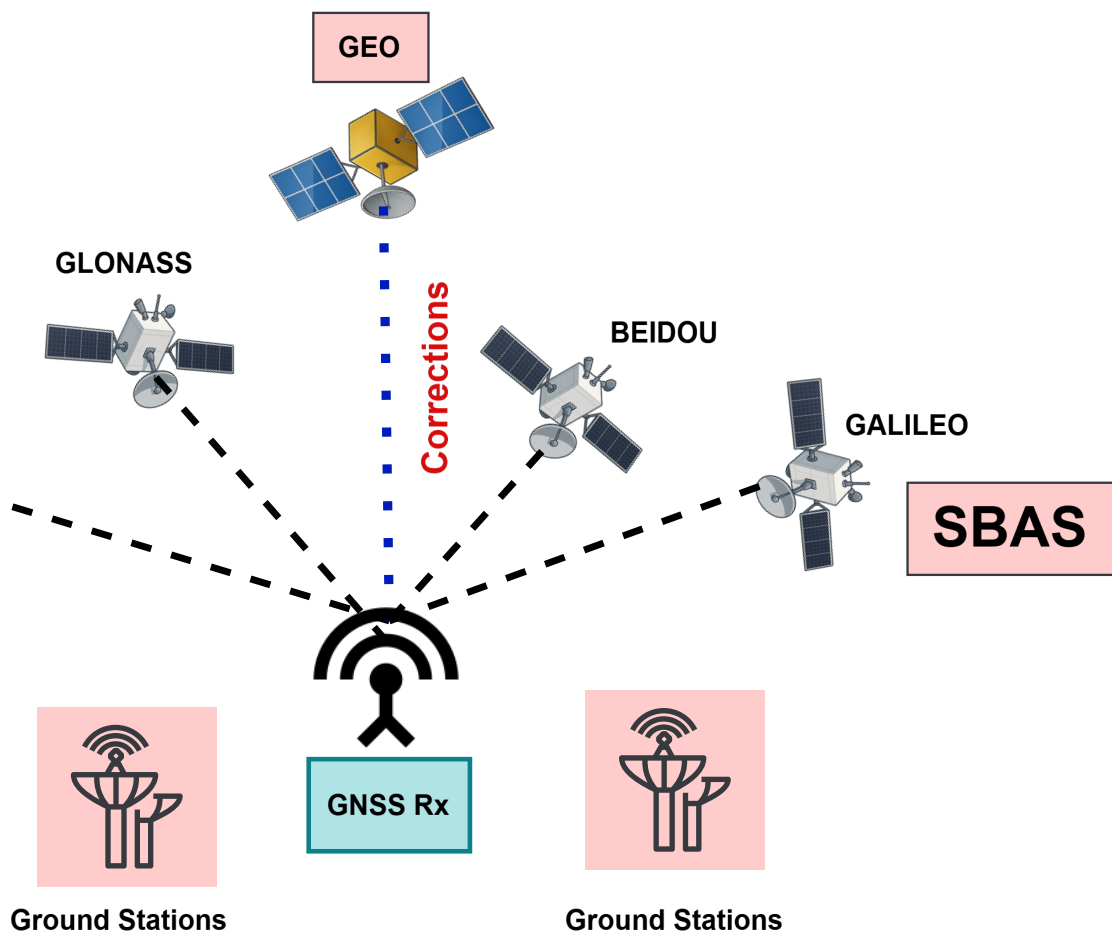


Figure 2.11: SBAS.

broadcast the data to GNSS receivers over wide areas. SBAS is particularly useful for aviation and safety-critical applications where integrity and real-time correction are essential. Figure 2.11 illustrates the SBAS architecture, showing how ground stations, geostationary satellites, and GNSS receivers interact to improve positioning performance.

Table 2.2: Comparison of Absolute and Relative GNSS Positioning Techniques

Technique	Positioning Type	Corrections Used	Infrastructure Requirement	Typical Accuracy	Use Cases
SPP (Standard Point Positioning)	Absolute	Broadcast ephemeris and satellite clock	None (standalone)	3–10 meters	Consumer GNSS, smartphones, navigation apps
SBAS (Satellite-Based Augmentation System)	Absolute	GNSS corrections via GEO satellites (e.g., clock, iono)	SBAS satellites and ground segment	1–3 meters	Aviation, safety-critical navigation
PPP (Precise Point Positioning)	Absolute	Precise clock and orbit (from SSR)	Global correction service (e.g., IGS, RTX)	Decimeter to centimeter (after convergence)	Geodesy, offshore navigation, precision farming
SSR (State Space Representation)	Absolute	Orbit, clock, ionosphere, troposphere corrections (broadcast as state models)	Correction provider or satellite broadcast	Enables PPP or similar solutions	Correction model distribution for high-accuracy GNSS
DGPS (Differential GPS)	Relative	Code-based differential corrections from nearby reference station	Local base station	1–3 meters	Vehicle tracking, mapping
GBAS (Ground-Based Augmentation System)	Relative	High-integrity corrections near airports (via VHF)	Airport-based ground stations	<1 meter (with integrity monitoring)	Precision aircraft landing systems
RTK (Real-Time Kinematic)	Relative	Carrier-phase differential corrections	Local or network RTK base station (with data link)	1–5 cm (real-time)	Land surveying, precision agriculture, autonomous vehicles

2.2.2.4 GNSS Errors

Despite the widespread use of GNSS in consumer and professional applications, its positioning accuracy and reliability can be significantly affected by several factors, particularly in challenging environments such as urban canyons or dense forests. These limitations include:

- **Multipath Effects:** Multipath occurs when GNSS signals reflect, diffract, or scatter off nearby surfaces (e.g., buildings, trees, or the ground) before reaching the receiver. These NLOS signals follow longer paths, resulting in various signal distortions:
 - *Signal Delay:* Reflected signals arrive later than direct signals, leading to pseudorange errors.
 - *Signal Attenuation:* Multipath can weaken the received signal strength.
 - *Polarization Shift:* Right-hand circularly polarized (RHCP) signals may change to left-hand circularly polarized (LHCP), affecting reception quality.
 - *Environmental Dependence:*
 - * In **urban environments**, strong reflections may introduce errors up to 10–15 meters [50].
 - * In **open sky conditions**, ground reflections still cause multipath, but typically below 1 meter [50].
 - *Time-Correlated Noise:* Multipath is not purely random and may persist over short durations, especially in static scenarios.
- **Atmospheric Delays:** As GNSS signals travel through the Earth's atmosphere, they are subject to delays that impact positioning accuracy. These delays primarily arise from two atmospheric regions:
 - *Ionospheric Delay:* The ionosphere is a region of the upper atmosphere that begins around 50 km and extends up to approximately 1000 km in altitude. It contains ionized sublayers specifically the ionized D-layer, E-layer, and F-layer where the density of free electrons varies. The most significant concentration of electrons typically occurs between 300 km and 400 km. GNSS signals experience a group delay and carrier phase advancement when passing through this region. The delay increases with Total Electron Content (TEC), which depends on factors like the solar cycle, time of day, geomagnetic latitude, and solar flux. Daytime delays can exceed 30 meters, while nighttime delays are generally in the range of 1–2 meters. The delay effect also decreases with the square of the signal frequency [51].
 - *Tropospheric Delay:* The troposphere is the lower part of the atmosphere, reaching up to about 50 km in altitude. It contains both hydrostatic (dry gases) and wet (water vapor) components with dry gases contributing 90% and water vapor 10% to the total delay. The troposphere introduces

signal delay due to refraction, particularly at lower satellite elevation angles. Unlike the ionosphere, this delay is not frequency dependent. A typical zenith delay is around 2.4 meters and can increase to 25–30 meters at low elevation angles [52].

These delays are difficult to model accurately in real-time and can introduce meter-level errors.

- **Satellite Geometry (Dilution of Precision - DOP):** The relative positioning of satellites affects the geometric strength of the position solution. Poor satellite geometry (e.g., satellites clustered in one part of the sky) leads to a high DOP value, which amplifies positioning errors even if signal quality is good, as shown in Fig. 2.12 [53].

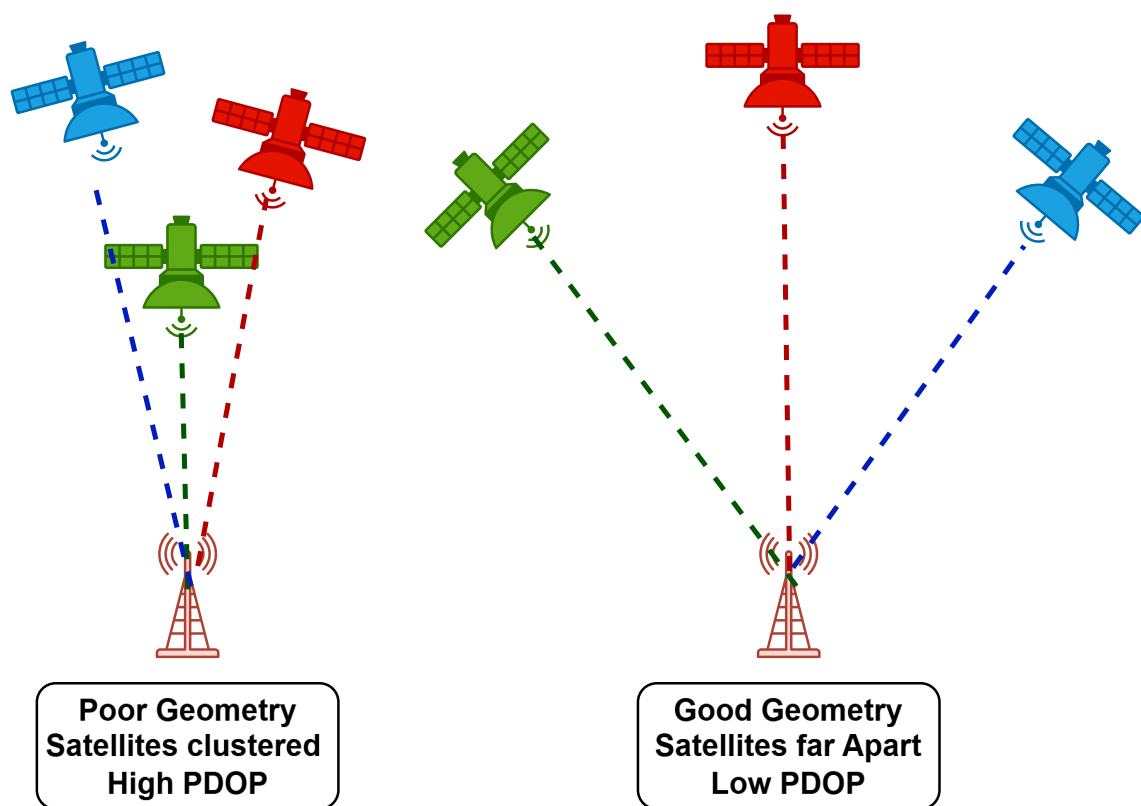


Figure 2.12: GNSS Geometry.

- **Receiver Errors:** Inexpensive GNSS receivers in smartphones suffer from limitations such as low-quality antennas, poor clock stability, and suboptimal signal processing algorithms. These constraints lead to noisier measurements and higher position uncertainty compared to professional-grade receivers.
- **Satellite Clock and Ephemeris Errors:** Inaccuracies in the timing information or satellite orbital data (ephemeris) can result in position deviations. Although these are generally corrected by satellite control segments, residual errors still affect the final solution [54].
- **Spoofing and Jamming Attacks:**

- *Spoofing* involves transmitting fake GNSS-like signals to mislead the receiver into calculating an incorrect position [55].
- *Jamming* refers to deliberate or unintentional interference that overwhelms GNSS signal reception, potentially causing loss of signal or degraded accuracy [55].

Both threats are increasingly relevant, especially in safety-critical and military applications.

GNSS in Smartphones: Initially, smartphones provided only coarse GNSS outputs such as PVT. These outputs were derived from internal processing performed by the GNSS chipset, with no access to the underlying signal measurements.

The introduction of raw GNSS measurements marked a major shift. Smartphones began exposing low-level data such as pseudorange, carrier-phase, Doppler shift, and carrier-to-noise density ratio (C/N_0). This advancement opened up the possibility for custom algorithms to be applied on-device or offline, supporting high-precision methods such as RTK, PPP, and spoofing detection.

A further enhancement came with support for dual-frequency GNSS. By simultaneously receiving signals on two frequencies (L1 and L5), smartphones can mitigate ionospheric delay errors and improve measurement precision, particularly in multipath-prone environments like urban canyons.

In terms of practical performance, standard GNSS positioning on smartphones typically achieves 3–5 meter accuracy in open-sky conditions. However, signal degradation due to urban obstructions, reflections, and poor satellite geometry often leads to larger errors in real-world scenarios.

Despite these advances, smartphone GNSS still faces inherent limitations. Compact antennas, limited processing capabilities, power constraints, and restricted access to correction services reduce the achievable accuracy and reliability compared to professional-grade receivers. Nevertheless, the combination of raw data access, multi-constellation support, and dual-frequency reception has significantly improved the research and application potential of GNSS on smartphones.

2.2.3 Data Collection from Smartphones

An essential step in smartphone-based positioning is the collection of raw sensor data. A common question is: How can this data be reliably accessed and recorded for processing? Fortunately, several Android applications have been developed to log raw inertial and GNSS data from smartphone sensors.

Table 2.3 provides a list of widely used mobile applications for logging data from accelerometers, gyroscopes, magnetometers, and GNSS receivers. These tools are commonly used in research and development for pedestrian dead reckoning, sensor fusion, and precise GNSS positioning studies.

Table 2.3: Android Apps for Logging Inertial and GNSS Sensor Data

App	Developer	Output Format
GetSensorData	Loopsi	Multi-sensor: accelerometer, gyroscope, magnetometer, GNSS, etc. [56]
MATLAB Sensor Logging	MathWorks	–
Sensor Kinetics	Innoventions, Inc	Real-time inertial logs (multi-IMU)
Sensor Data Recorder / Pro	Elan Rozner	–
Sensors Log	EXALAB	–
SensorsData	Alexander Balberdin	–
Sensor Stream	Innoventions, Inc	–
GNSSLogger App [57]	Google	CSV, NMEA, RINEX
Geo++ RINEX Logger [58]	Geo++ GmbH	RINEX
rinexON App [59]	FLAMINGO	NMEA, RINEX
GalileoPVT App [60]	ESA	CSV, NMEA
G-RitZ Logger	Ritsumeikan University	RINEX, NMEA
GNSS/IMU Logger [61]	Bundeswehr München	IMU, CSV, RINEX

2.3 Existing Survey Papers

To assess the landscape of prior surveys and position this work within the current body of literature, a focused search was conducted for existing survey and review papers on inertial and GNSS fusion, particularly in the context of smartphone-based localization. Searches were conducted using Google Scholar and Web of Science (WoS), spanning the years 2017 to 2024. The following query was used in both databases, with the publication year filter set from 2017 to 2024:

("GNSS" OR "GPS") AND ("inertial sensors" OR "IMU" OR "PDR") AND ("fusion" OR "integration") AND ("survey" OR "review") AND (year:2017..2024)

Results from both databases were combined, and duplicates were removed. This yielded 48 unique publications. After screening based on relevance to smart device-based localization, 23 were excluded for not involving smartphones or wearable systems. From the remaining 25, an additional 17 were removed due to their exclusive focus on indoor-only scenarios. The remaining 8 survey papers were retained for detailed comparison. Fig. 2.13 summarizes the filtering process.

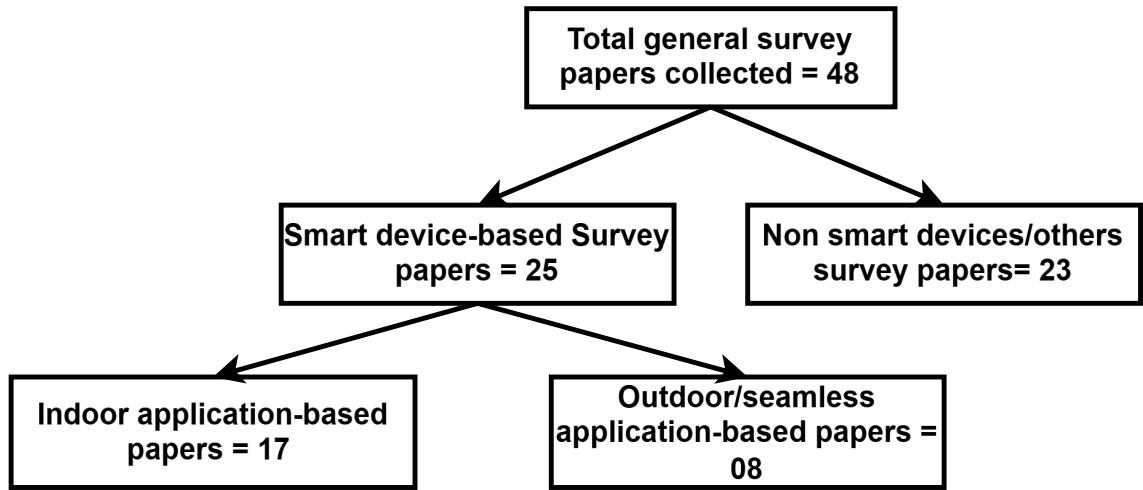


Figure 2.13: Survey filtering process for smart device-based inertial+GNSS fusion surveys.

A review of eight existing survey papers was conducted to assess the current state of research and identify remaining gaps relevant to this thesis. A summary of their contributions and limitations is presented below, followed by a structured comparison in Table 2.4.

[62] offers a general overview of localization technologies but does not examine GNSS fusion architectures or smartphone-specific applications. [63] considers smartphone platforms and filter-based fusion methods but omits machine learning strategies. [64] emphasizes GNSS vulnerabilities rather than fusion architectures or processing methods. [65] is an earlier work that lacks coverage of recent techniques or smartphone relevance. [66] focuses on AI-based localization but does not integrate inertial and GNSS fusion within smart device contexts. [67] presents optimization-based approaches yet omits a broader discussion of fusion architecture types. [68] addresses machine learning-based positioning using smartphones but does not include traditional fusion structures. Lastly, [43] presents a valuable survey on multi-sensor integration, including inertial-GNSS fusion, but lacks a focus on smartphones and their particular constraints.

These limitations underscore the need for a dedicated and comprehensive survey focused on smartphone-based inertial and GNSS fusion. This thesis fills this gap by systematically covering all major integration architectures (loosely, tightly, and ultra-tightly coupled), evaluating a range of fusion strategies (filter-based, optimization based, and learning-based), and addressing emerging research opportunities specific to smartphones and wearable devices.

Table 2.4 presents a detailed comparison between the eight selected survey papers and our work, based on several key evaluation dimensions. These include whether the surveys explicitly address smartphones and wearable devices, the extent to which different inertial+GNSS fusion architectures are covered (whether only a few or all known architectures are discussed), the inclusion of various fusion strategies such as filter-based, optimization-based, and machine learning (ML)-based approaches, and whether future directions or alternative fusion opportunities are explored.

The analysis reveals that none of the existing surveys comprehensively address inertial and GNSS fusion from a smartphone-centric perspective across all architectural and processing dimensions. In contrast, this thesis presents a survey that targets smartphone and wearable devices specifically, thoroughly covers all principal integration architectures (including loosely, tightly, and ultra-tightly coupled designs), reviews a broad spectrum of fusion strategies (ranging from filtering and optimization to ML-based methods), and emphasizes unexplored opportunities and future research challenges. This positions the presented survey as a unique and timely contribution, offering valuable insight for both academic researchers and practitioners working in the area of smartphone-based localization, particularly in GNSS-challenged or denied environments.

Table 2.4: Comparison with Existing Survey Papers Addressing Inertial+GNSS Fusion

Reference	Publication Year	Smartphone-Based	Few Architectures Covered	All Architectures Covered	Filter-Based	Optimization-Based	ML-Based / Future Opportunities
Asaad et al. [62]	2022	✓	✗	✗	✗	✗	✗
Elsanhoury et al. [63]	2021	✓	✓	✗	✓	✗	✗
Zangenehnejad et al. [64]	2021	✓	✓	✗	✗	✗	✗
Gobana et al. [65]	2018	✓	✗	✗	✓	✗	✗
Al bitar et al. [66]	2020	✗	✓	✗	✓	✗	✓
Nagula et al. [67]	2022	✗	✓	✗	✗	✓	✗
Cohen et al. [68]	2024	✗	✓	✗	✗	✗	✓
Zhuang et al. [43]	2023	✗	✓	✗	✓	✓	✗
Our Survey	2024	✓	✓	✓	✓	✓	✓

Different databases were used at different stages of the review process: Google Scholar and Web of Science were employed to identify existing survey papers, while Web of Science and Scopus were used for systematic searches addressing the defined research questions due to their structured indexing and filtering capabilities.

2.4 Research Methodology

This section describes the process used to find and select studies for this systematic review. A clear and straightforward approach was adopted to ensure transparency and replicability for future researchers. This method enables a thorough review of existing literature, the identification of common trends, and the recognition of areas requiring further investigation.

The PRISMA guidelines (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) were followed to guide the work [69]. PRISMA includes a 27-point checklist and a flowchart that breaks the review process into four key steps: identifying studies, screening them, checking if they meet the criteria, and finally, including them in the review.

2.4.1 Research Questions

A key part of any systematic review is defining the main research question, as it helps focus the review and guides the overall analysis. This review begins by addressing the following main research question (MRQ):

MRQ: What are the alternative methods, potential, and research gaps regarding the fusion of inertial sensors and GNSS in smartphones for location and navigation?

Since this main question is quite general, it is divided into multiple research questions (RQs) below:

1. **RQ1:** Which INS methods are used with smartphone sensor data in existing inertial and GNSS fusion systems?
2. **RQ2:** What GNSS processing techniques have been implemented and evaluated in smartphones?
3. **RQ3:** What sensor fusion methods/algorithms have been used to combine inertial sensors and GNSS in smartphones?
4. **RQ4:** What fusion architectures for inertial sensors and GNSS have been proposed and evaluated for smartphone positioning?
5. **RQ5:** What are the current challenges and future opportunities in inertial+GNSS fusion for smartphones?

2.4.2 Keywords

In recent years, there has been a rapid rise in the number of studies about the fusion of inertial+GNSS architectures in mass-market devices. As a result, it is essential to establish specific search criteria and tactics to find the publications that are most pertinent to the subject of this systematic review. Consequently, the process continues with selecting keywords related to the specific topic and goals of the study. The chosen keywords include "inertial sensors fusion," "inertial measurement unit fusion," "INS/GNSS," "IMU/GNSS," "PDR/GNSS," "data-driven INS/GNSS," "position*," "location," and "navigation localization. Any number of characters is indicated by the wildcard pattern (* in the query). These keywords were used in queries to identify related terms with similar prefixes (such as position, positioning, and positions).

2.4.3 Query

Once the keywords were defined, the search queries were structured to retrieve relevant literature. These queries were applied using the advanced search features of two well-known academic databases: *Web of Science* and *Scopus*. The following search strings were used:

Web of Science Query:

```
TS((((('Inertial sensors' OR 'Inertial Measurement Unit'
OR IMU OR PDR OR INS OR Accelerometer OR Gyroscope
OR Magnetometer OR 'data-driven ins'))
AND (gnss OR gps OR PPP OR RTK))
AND (fusion OR integration OR merge))
AND (position* OR location OR localisation OR navigation))
Timespan: 2016-2024
```

Scopus Query:

```
TITLE-ABS-KEY((((('Inertial sensors' OR 'Inertial Measurement Unit'
OR imu OR PDR OR ins OR accelerometer OR gyroscope
OR magnetometer OR 'data-driven ins'))
AND (gnss OR gps OR RTK OR PPP))
AND (fusion OR integration OR merge))
AND (position* OR location OR localisation OR navigation))
AND (LIMIT-TO (PUBYEAR , 2024) OR LIMIT-TO (PUBYEAR , 2023)
OR LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021)
OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019)
OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017)
OR LIMIT-TO (PUBYEAR , 2016))
```

Both queries aimed to retrieve peer-reviewed papers relevant to the scope of this review, including conference papers, journal articles, and other scholarly sources. However, not all returned results met the inclusion criteria.

2.4.4 Choosing a Study

The results of the preceding search must go through a process outlined in the PRISMA diagram and checklist document. In this step, all related studies are found, duplicate records are eliminated, and inclusion and exclusion criteria are defined. These criteria serve as the foundation for the final choice of works to be included in the qualitative and quantitative synthesis.

2.4.4.1 Stage 1 Identification

SCOPUS and Web of Science are both prominent databases that index content from a wide range of sources, including IEEE Xplore, SpringerLink, ArXiv, Wiley Online Library, and others. When the results from the two data sets are combined, duplicate records are created that need to be deleted. Thus, utilizing reference management software, the obtained records are stored along with their metadata and bibliography. This software is used to categorize studies found through search engines and delete duplicate information.

2.4.4.2 Stage 2 Criteria for Selection and Screening

After removing duplicate records, a set of distinct registries remains, which must be filtered to retain only the articles relevant to this review. To achieve this, specific inclusion and exclusion criteria were established for selecting articles relevant to the systematic survey. Therefore, two main criteria are established: Inclusion Criteria (IC) and Exclusion Criteria (EC) with the following tactics.

- IC1 It must be written in the English language.
- IC2 Research works that must fuse or combine inertial sensors and GNSS fusion architectures.
- IC3 Articles must deal with smartphones or other mass-market smart devices (e.g. smartwatches).
- EC1 Works only dealing with indoor scenarios.
- EC2 Works not published in peer-reviewed international journals or conference proceedings.
- EC3 Works not using commercial devices, i.e. using custom-made devices.

Only works published in the past eight years were included in this review. The decision to use the material from 2016 was influenced by the surge in research on positioning systems based on the fusion of inertial sensors and GNSS in smartphones. Thanks to the Google Android team for making it possible to access raw GNSS measurements from Android smartphones [70].

A manual revision of titles and abstracts was conducted, followed by tagging as ACCEPTED for selected articles and REJECTED + EC for excluded records, to retain only those works that met all criteria stated in the inclusion (IC) and exclusion (EC) conditions.

2.4.4.3 Stage 3 Eligibility

Each paper was carefully examined with consideration of the main objective of the review and the defined inclusion and exclusion criteria. The reviewed article is not included in our work if it fails to meet the standards set out earlier (see stage 2).

2.4.4.4 Stage 4 Included

In this stage, the articles are reviewed based on their conclusion and contribution related to smartphone positioning from the fusion of inertial sensors and GNSS. This is the final step to ensure that only pertinent publications are chosen for this evaluation.

2.4.5 PRISMA Processing

After executing the search queries, a total of 3,874 records were retrieved: 1,993 from SCOPUS and 1,881 from Web of Science. As expected, there was some overlap between the two databases, with many studies indexed in both. To avoid double-counting, the results from both sources were merged, and inter-database duplicates were identified and removed. This deduplication step was applied only across the combined dataset, not within each database, resulting in 1,595 unique records for screening.

The selection process then proceeded through the PRISMA framework. These 1,595 records were screened based on titles and abstracts. At this stage, 1,531 studies were excluded, the majority of which did not meet the inclusion criteria. A significant portion of these fell under EC3, which excludes studies that rely on custom-made devices or non-commercial hardware. In most cases, the type of hardware could be identified from the abstract, as many papers explicitly mentioned using platforms such as Raspberry Pi, Arduino, or specialized IMU modules. While full verification would require reading the entire paper, the abstracts generally provided enough detail to make a confident decision. A smaller number of exclusions were due to EC1 (indoor-only studies).

After the screening phase, 64 articles remained and were evaluated at the full-text level. Among them, 25 were excluded for not meeting the inclusion criteria, and 3 full-texts were unavailable. As a result, 36 articles were successfully included in the final analysis, and the distribution of obtained papers is shown in Fig. 2.15. The entire selection process is illustrated in the PRISMA flow diagram shown in Fig. 2.14.

2.4.6 Data Extraction

For each of the final selected studies, a structured data extraction process was applied to identify which research question (as introduced earlier) each work contributed to. The extracted data were organized and analyzed to address the predefined research questions.

Each of the following sections in this chapter is structured to address one of the research questions individually. Section 2.5 discusses the inertial navigation methods (RQ1), followed by GNSS processing techniques in section 2.6 (RQ2). Section 2.7 is dedicated to sensor fusion methods/algorithms (RQ3), while Section 2.8 presents the architectural frameworks proposed in the literature (RQ4). Lastly, Section 2.9 outlines open challenges and future opportunities in smartphone-based inertial+GNSS fusion (RQ5). This structure ensures that all findings are presented in direct alignment with the objectives of the review.

2.5 Inertial Navigation used in smartphone-based Inertial+GNSS Fusion Architectures

Inertial navigation on smartphones can be approached through different strategies, primarily including classical INS, PDR, and data-driven INS/PDR variants. Each

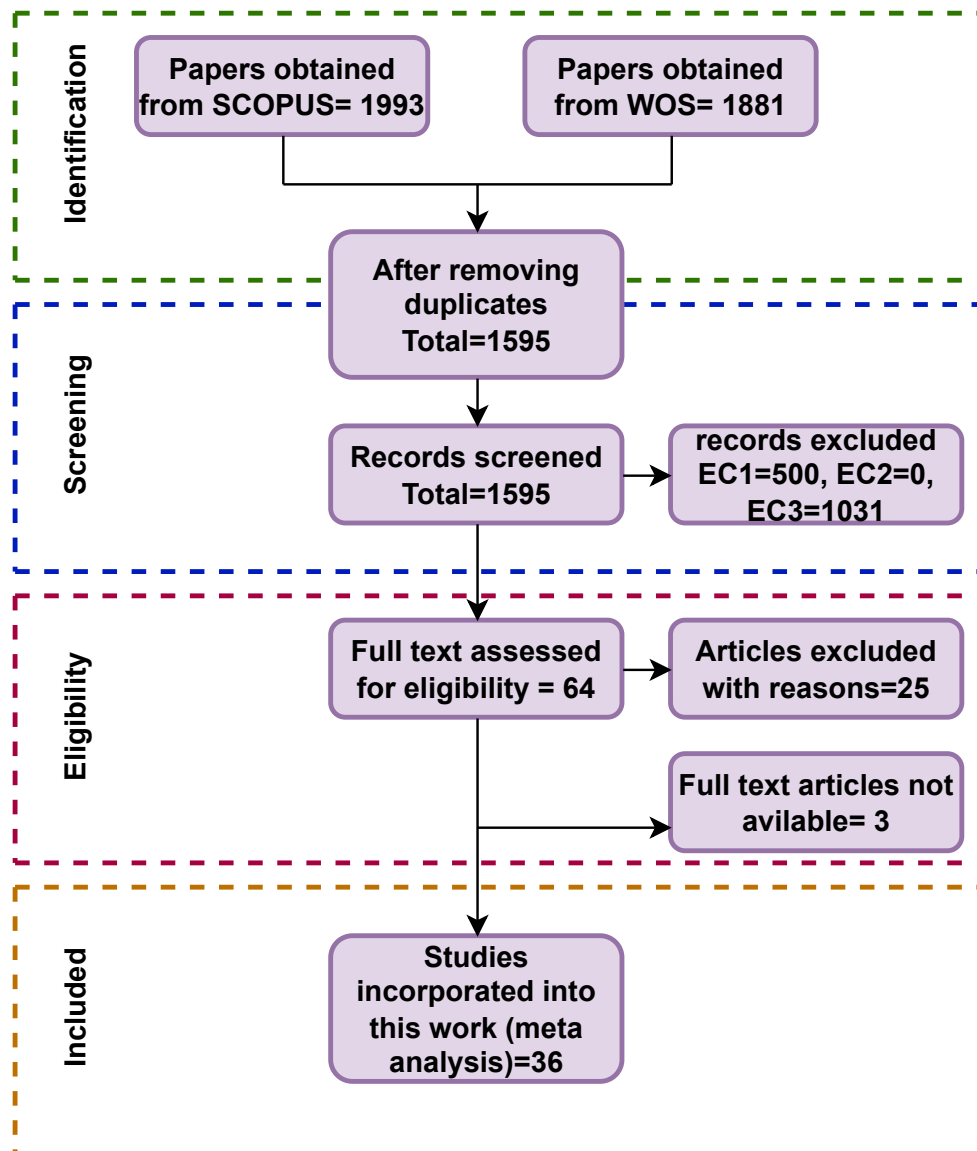


Figure 2.14: PRISMA Processing.

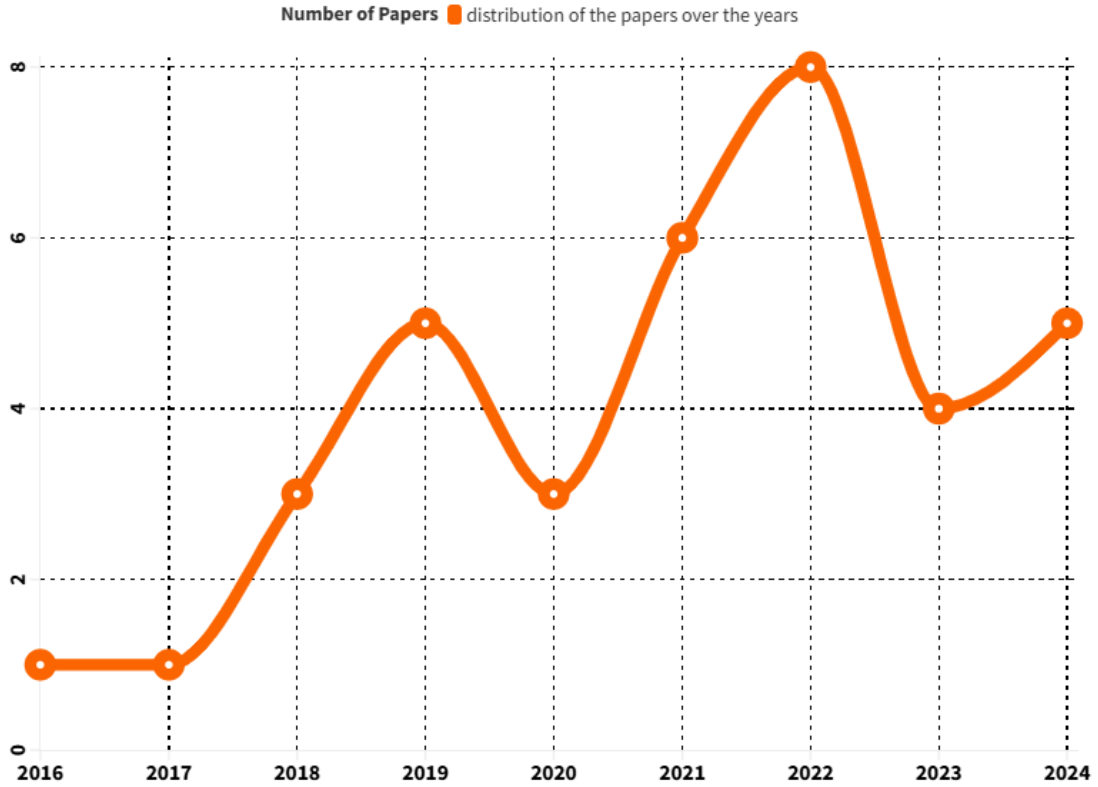


Figure 2.15: Distributions of collected papers over the 2016-2024.

of these methods relies on the inertial data collected by built-in MEMS sensors in smartphones. The following subsections first outline the fundamental principles behind each technique, followed by a critical examination of how they have been implemented, evaluated, and adapted in recent research. Where relevant, practical challenges and insights emerging from the literature are also highlighted.

2.5.1 Inertial Navigation System (INS)

Background: Recent literature highlights a growing interest in deploying INS on smartphones for pedestrian navigation, driven by the ubiquity of embedded MEMS IMUs. While traditional INS implementations have seen success in structured environments (e.g., vehicles or robotics), extending these techniques to handheld smartphone scenarios remains non-trivial. Studies have consistently reported degraded performance in pedestrian use cases due to unpredictable motion patterns, limited sensor fidelity, and inconsistent device handling. Several works attempt to mitigate these limitations through model simplification, external calibration, or signal corrections; however, no unified solution has emerged. The following analysis synthesizes recurring limitations reported across multiple studies to reveal common implementation challenges specific to smartphone-based INS systems.

In the literature: Several studies have identified key implementation challenges when deploying INS on smartphones, particularly in pedestrian applications. These challenges often stem from irregular user motion, sensor noise, limited computational resources, and the constraints of consumer-grade MEMS hardware.

Use Case Dependence and Motion Complexity

Unlike vehicular applications where motion is smooth and predictable, pedestrian navigation involves irregular gait, frequent orientation changes, and varied smartphone placements (e.g., handheld, pocket, or bag). These factors significantly increase the complexity of inertial tracking. While studies such as [71, 72] have shown improved performance using smartphones in vehicle-based scenarios, robust standalone pedestrian INS remains an open problem due to higher motion variability and noise sensitivity.

Model Simplification and Orientation Constraints

To reduce system complexity, several approaches simplify the navigation model by assuming planar motion, using reduced sensor sets (e.g., one gyro and two accelerometers), or limiting orientation degrees of freedom. For example, [73, 74, 75] apply Reduced Inertial Sensor Systems (RISS) in vehicle scenarios, while [76] develops a custom Android app to aid orientation estimation in pedestrian settings. Although such simplifications reduce computational load, they limit adaptability to diverse and dynamic use cases.

Sensor Sampling and Timing Irregularities

IMU data collected from smartphones often suffers from non-uniform sampling, asynchronous clocks, and inconsistent timestamps. These inconsistencies complicate accurate integration and sensor fusion. As discussed in [77], resampling techniques using MATLAB were employed to enforce a consistent 100 Hz rate, while other works [78, 79] used interpolation and filtering. Still, hardware-dependent variability, especially on Android devices, remains unresolved.

Sensor Calibration and Error Compensation

Consumer-grade MEMS sensors are subject to bias, scale factor errors, and drift. Existing solutions apply static six-position calibrations [73, 19] or dynamic methods such as turntable-based gyro correction and motion modeling [80, 72]. Despite these efforts, calibration methods often lack generalizability across devices and conditions, limiting scalability.

Orientation Estimation and Gravity Compensation

Accurate orientation is essential to separate gravitational and motion-induced accelerations when transforming measurements from the device to world frame. Some studies [72, 81] leverage Android’s rotation vector API for real-time estimation, while others [76] rely on sensor fusion via complementary or Kalman filters. However, dynamic handheld usage leads to orientation drift, which propagates into position errors.

Drift Accumulation Without External Corrections

Pure inertial navigation suffers from unbounded error growth without external references. To mitigate drift, many studies incorporate corrections via GNSS or Zero-Velocity Updates (ZUPT) [74, 82]. Yet, without tight integration or periodic corrections, standalone INS becomes unsuitable for long-term pedestrian tracking [83].

These challenges collectively highlight the need for robust fusion architectures tailored to the constraints and dynamics of smartphone-based pedestrian navigation.

2.5.2 Pedestrian Dead Reckoning (PDR)

Background: PDR is a modular inertial navigation method that estimates a pedestrian's position by detecting steps, estimating their length, and updating direction based on heading. When implemented on smartphones, PDR leverages data from built-in MEMS IMUs but must adapt to varying user behavior, device placement, and environmental conditions. Unlike traditional PDR systems with dedicated foot-mounted sensors and fixed alignment, smartphone-based PDR must handle unpredictable sensor orientations, inconsistent gait cycles, and device-specific limitations. Each stage of the PDR pipeline, step detection, step length estimation, and heading computation, is influenced by factors such as gait irregularity, terrain variation, sensor noise, and magnetic interference. Understanding the performance and limitations of these components in the literature provides critical insight into the current capabilities and constraints of smartphone-based pedestrian tracking systems.

In the Literature: Smartphone-based PDR systems typically follow a modular pipeline involving step detection, step length estimation, and heading computation, all of which are influenced by how the device is carried and the surrounding environment. The following review categorizes the literature according to these core PDR components, providing a structured analysis of techniques and limitations reported across studies.

Step Detection: Most smartphone PDR systems begin by detecting footfall events through variation in the acceleration magnitude:

$$a_{\text{mag}} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2.4)$$

Classical peak–valley detection is employed by Jiang et al. [24, 84, 25, 85, 86] and Basso et al. [87, 88]. It performs well for regular gait cycles but degrades under varying pace or terrain. More robust systems, such as Zhang et al. [28], apply a finite-state machine (FSM) to model heel strike and toe-off phases, minimizing false detections during turns or phone shifts. However, adaptive threshold tuning and per-user calibration remain necessary.

Step Length Estimation: Once steps are detected, estimating their length becomes key. The popular Weinberg model is:

$$S_k = k \cdot \sqrt{a_{\text{peak}} - a_{\text{min}}} \quad (2.5)$$

where k is an empirical constant. This model is used in studies by Jiang et al. [86] but it tends to lose accuracy at slow speeds or irregular gait. Regression-based alternatives are proposed in [28, 89, 26], incorporating parameters like step frequency, duration, and acceleration features. Others [90, 91] apply machine learning models (e.g., SVMs, neural networks) trained on smartphone data for more adaptive stride estimation, though these depend on large datasets and face issues with generalization and drift.

Heading Estimation (Orientation): Estimating the user's heading, θ_k , is essential for displacement updates:

$$\Delta \vec{p}_k = S_k \begin{bmatrix} \cos \theta_k \\ \sin \theta_k \end{bmatrix} \quad (2.6)$$

Heading is commonly derived from sensor fusion using AHRS algorithms. Madgwick’s filter is used in [87, 92]. These methods integrate gyroscope, accelerometer, and magnetometer data. Jiang et al. [93] proposed a binary linear regression model for heading estimation. Android’s built-in rotation-vector sensor is also utilized in some cases [27]. However, magnetic interference, especially indoors, frequently disrupts heading estimates.

Device Carrying Mode and Alignment. PDR performance is strongly affected by how the smartphone is carried. Fixed placements, such as in the waist or pocket, tend to offer better inertial stability, as shown in studies by Jiang et al. [24, 84, 25, 85, 86], who evaluated both handheld and body-mounted configurations. Zhu et al. [94] leveraged consistent carrying positions using the “Walker” app to gather high-frequency (100 Hz) IMU data from hardware-aligned mounts. Li et al. [95] proposed adaptive filtering techniques to mitigate the impact of orientation shifts during dynamic motion. Despite such efforts, accurate real-time alignment between the phone’s axes and walking direction remains a significant challenge for PDR systems.

Height and Altitude Tracking. Barometric sensors are occasionally integrated to support vertical positioning. Zhu et al. [94] and Li et al. [95] used barometer data for detecting floor changes, often referred to as “2.5D” PDR, *where vertical motion is modeled discretely as floor-to-floor transitions rather than as continuous 3D height estimation*. Some works, [96], attempted full 3D tracking by integrating vertical acceleration, though cumulative drift limits reliability. Most studies agree that standalone smartphone altitude tracking remains inaccurate without additional correction or sensor fusion.

Challenges in Smartphone-Based PDR Based on the reviewed literature [24, 84, 25, 85, 86, 93, 94, 87, 88, 95, 27, 89, 90, 96, 26, 97, 28, 92, 91] The following recurring challenges hinder robust and accurate PDR deployment on smartphones:

- **Variability in Walking Patterns and Terrain:** Changes in gait, speed, and surface type introduce inconsistencies that degrade the reliability of heuristic step detection methods such as peak-valley or zero-crossing techniques.
- **Device Orientation Changes:** Smartphones carried in the hand, moved between different positions, or subject to shaking, rotation, or swinging experience frequent orientation changes. These variations make heading estimation more challenging and require flexible, adaptive correction methods.
- **Sensor Errors and Environmental Disturbances:** Magnetometers are strongly affected by environmental magnetic disturbances in indoor and urban environments, which are spatially correlated and not sensor-intrinsic, while barometers are susceptible to both sensor noise and environmental pressure variations caused by weather conditions and indoor airflow. These effects

cannot be accurately modelled as constant sensor biases or white measurement noise and therefore pose challenges for reliable heading and height estimation.

- **Model Generalizability:** Step length estimators trained on individual users (e.g., regression or ML-based models) often fail to generalize across different users, gaits, or environments without extensive recalibration.
- **Smartphone Hardware Constraints:** Limitations such as non-uniform IMU sampling, sensor latency, occasional data dropout, and aggressive power-saving modes on Android systems impact the continuity and accuracy of real-time PDR computation.

Despite these challenges, smartphone-based PDR remains the most practical inertial-only positioning method for indoor pedestrian navigation, especially when models are carefully calibrated and validated in context specific scenarios.

2.5.3 Data-Driven Inertial Navigation

Background: Conventional INS and PDR techniques suffer from cumulative drift, especially when implemented using low-cost MEMS inertial sensors on smartphones. These systems require either continuous integration of acceleration or step-by-step updates based on heuristic detection methods, both of which are susceptible to noise, sensor bias, and user-specific gait variations.

However, AI/ML-based data-driven inertial navigation systems offer an alternative approach, where displacement is learned directly from raw inertial measurements using machine learning and deep learning models. These models can predict displacement vectors from temporal windows of sensor data, eliminating the need for traditional components such as step detection, orientation estimation, or handcrafted heuristics.

Implementation: To build an AI/ML-based PDR system, raw data from smartphone inertial sensors, specifically, 3-axis accelerometer and gyroscope readings, is used. At each time step t , the acceleration $\vec{a}(t) \in \mathbb{R}^3$ and angular velocity $\vec{\omega}(t) \in \mathbb{R}^3$ are collected. Instead of analyzing a single reading, a short sequence (or window) of data over the past T time steps is used.

$$\mathcal{X}_t = \{[\vec{a}(t - \tau), \vec{\omega}(t - \tau)] \mid \tau = 0, \dots, T - 1\} \quad (2.7)$$

This sequence represents the recent motion history of the device. A machine learning model $\mathcal{F}_\theta$, with parameters θ , learns to map this input window to the estimated 2D displacement (movement in the ground plane):

$$\Delta \vec{p}_t = \mathcal{F}_\theta(\mathcal{X}_t) \quad (2.8)$$

Then reconstruct the full trajectory by adding each predicted displacement step-by-step:

$$\vec{p}_{t+1} = \vec{p}_t + \Delta \vec{p}_t \quad (2.9)$$

The model is trained using supervised learning, where the goal is to minimize the difference between predicted and actual displacements (from ground truth), usually measured by Mean Squared Error (MSE):

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{t=1}^N \|\Delta \vec{p}_t^{\text{true}} - \mathcal{F}_\theta(\mathcal{X}_t)\|^2 \quad (2.10)$$

This framework enables the model to learn motion dynamics directly from data, eliminating the need for traditional steps such as step detection or heading estimation.

In the Literature: To date, there is very limited work on deep learning-based fusion frameworks that directly integrate inertial sensors and GNSS using smartphone data. Existing learning-based approaches are primarily applied to inertial-only components, while the fusion with GNSS measurements are, still performed using classical filtering techniques. An example of this hybrid strategy is presented in [91], where a data-driven inertial model is combined with GNSS updates through a conventional filtering framework to improve robustness under degraded GNSS conditions.

2.5.4 Overall Analysis of Inertial Navigation:

Table 2.5 offers a comprehensive overview of smartphone-based inertial navigation systems by organizing the literature across three main categories INS, PDR, and data-driven INS/PDR, while accounting for both device mounting positions and dimensionality (2D, 2.5D, and 3D). The analysis shows that handheld configurations are the most commonly addressed across all navigation types, reflecting realistic user behavior and widespread practical relevance. Pocket and waist-mounted positions are also frequently studied, particularly in PDR systems, which tend to rely on more consistent motion profiles for improved performance. In contrast, fixed or static surface configurations are typically observed only in classical INS systems, often in controlled settings.

In terms of dimensionality, most existing works, whether INS or PDR, focus on 2D and 2.5D implementations, with only a limited subset extending to full 3D positioning by incorporating vertical measurements from barometers or vertical acceleration data. Notably, PDR systems demonstrate broader coverage across all three-dimensional categories, due to their modular architecture and adaptability. Data-driven inertial navigation, in contrast, remains minimally explored in the literature, with only one known study [91] that employs deep learning to predict 2D displacements directly from raw inertial measurements.

This breakdown reveals several key trends: while PDR continues to dominate due to its robustness and reduced reliance on drift-prone integration, classical INS approaches remain valuable in scenarios requiring higher precision or fusion with external sensors. The scarcity of data-driven methods, especially those capable of 2.5D or 3D tracking, underscores a significant research gap. Overall, the table illustrates that despite the growing attention to inertial navigation, challenges such as dimensional limitations, variable mounting conditions, and the lack of

Table 2.5: Summary of Smartphone-based Navigation Methods by Mounting Position and Dimensionality

Mounting Position	INS			PDR		
	2D	2.5D	3D	2D	2.5D	3D
Handheld	[73, 98, 80, 78]	[75]	[80]	[24, 84, 87, 90]	[95, 97]	[94, 95]
Pocket	[72, 81, 74]	[82]	[76]	[86]	[94, 95]	[95]
Waist / Hip	[77, 71]	[79]	[76]	[89]	[94, 95]	[95]
Bag / Backpack	[75, 99]	—	[83]	[28, 88]	[95]	[95]
Not Specified or Mixed Modes	[83]	[79, 82]	[76]	[25, 85, 27, 96, 26, 28, 92]	[94, 95]	[95]
Mounting Position	Data-Driven ([91])					
	2D	2.5D	3D	2D	2.5D	3D
Handheld	[91]	—	—	—	—	—

generalizable machine learning models still constrain the broader applicability of these systems in real-world smartphone-based localization tasks.

Although these inertial navigation approaches have been applied across different smartphone positioning scenarios, their implications for pedestrian motion and Smartphone sensor constraints are of particular relevance for the investigations presented in the remainder of this thesis.

2.6 GNSS used in smartphone-based Inertial+GNSS Fusion Architectures

In smartphone-based Inertial+GNSS positioning systems, GNSS serves a crucial role in constraining inertial drift, initializing state estimators, and providing absolute positioning updates. While its foundational function remains consistent, offering globally referenced location information, its implementation varies significantly across studies. These variations include the type of signals used (e.g., frequency bands and constellations), the level of access to raw data, and the processing techniques applied. This section synthesizes key patterns and advancements found in the literature, structured around three core elements: (i) GNSS signal diversity, (ii) access to and usage of raw GNSS measurements, and (iii) the positioning techniques deployed in smartphone-based fusion architectures.

2.6.1 GNSS Signals

The evolution from single-frequency (SF) to dual-frequency (DF) GNSS reception represents a major advancement in smartphone positioning capabilities. While early devices were limited to L1-band signals highly susceptible to ionospheric delays and multipath errors modern chipsets increasingly support L5-band reception, allowing for improved vertical accuracy and ionospheric error mitigation. Despite this hardware progress, a critical insight from the literature is that most smartphones in practical use remain SF-only, and even those supporting DF often do not expose full raw access to both bands. As a result, the vast majority of studies still rely primarily on L1 data for algorithm development and evaluation.

Nevertheless, a few works have begun to explore the benefits of DF reception. For instance, [80] combined L1 and L5 observations in a fusion framework, demonstrating measurable accuracy gains from frequency diversity. Parallel to this, multi-constellation support has also become mainstream, with modern smartphones capable of receiving signals from GPS, Galileo, BeiDou, and GLONASS. This improves satellite geometry and availability, especially in degraded environments such as urban canyons. Studies such as [26, 82] incorporated multi-GNSS measurements to improve positioning continuity and resilience, though others (e.g., [47, 74]) remain largely GPS-dependent.

Despite these developments, few studies systematically evaluate the combined impact of dual-frequency and multi-constellation integration under real-world smartphone conditions. The effects of different constellation pairings, frequency combinations, and signal quality under dynamic scenarios remain largely unquantified.

Furthermore, challenges such as inconsistent DF support across device models, power consumption constraints, and increased processing complexity must be addressed for practical deployment. These gaps highlight an opportunity for future research to move beyond theoretical capability and toward end-to-end solutions optimized for mobile platforms.

2.6.2 Raw GNSS Measurements and Access

The introduction of raw GNSS access in Android 7 significantly expanded research opportunities in mobile positioning. While earlier studies were constrained to using processed Position, Velocity, and Time (PVT) outputs, recent works have increasingly leveraged raw observables such as pseudoranges, carrier-phase, and Doppler to achieve higher precision and robustness.

Notably, [94] applied carrier-phase smoothing to refine pseudorange measurements, improving stability in single-receiver GNSS setups. Environmental error mitigation has also seen progress; for example, [78] incorporated ionospheric and tropospheric models to correct raw data before positioning. More advanced pipelines have adopted PPP, as in [81, 80, 77], where global correction products, that is, precise ephemerides and clock biases, were used to reduce long-term drift and improve absolute accuracy. Meanwhile, [74] demonstrated RTK integration on smartphones using base-station corrections, achieving centimeter-level positioning.

These efforts illustrate a transition from basic raw data access to fully customized processing chains. However, challenges remain in handling carrier-phase ambiguity, maintaining fix continuity in urban environments, and real-time deployment on power-constrained mobile platforms. Overall, while raw GNSS access has unlocked substantial potential, realizing its full benefits requires careful algorithmic design and often, external correction sources.

2.6.3 GNSS Positioning Techniques

Within smartphone-based GNSS positioning research, various GNSS techniques have been selected based on specific application requirements, hardware limitations, and environmental constraints. Single Point Positioning (SPP) has been extensively adopted primarily due to its straightforward implementation, universal chipset compatibility, and independence from external correction data. Numerous studies, including [98, 72, 75, 99, 100, 47, 87, 95, 89, 90, 96, 28, 92], have employed SPP for its low computational overhead and suitability for real-time operation, particularly in resource-constrained smartphone environments. The literature emphasizes that despite inherent accuracy limitations due to multipath effects and atmospheric delays, SPP remains the preferred choice when ease of deployment, minimal processing resources, and robust real-time functionality are prioritized.

In contrast, researchers have explored advanced positioning techniques like Post-Processed Kinematic (PPK), Precise Point Positioning (PPP), and Real-Time Kinematic (RTK) to overcome the accuracy limitations of SPP. For example, studies such as [101] utilized PPK to achieve improved accuracy through precise carrier-phase processing and post-processing of GNSS data, highlighting its significant

benefit in pedestrian navigation scenarios. Similarly, Precise Point Positioning (PPP), demonstrated in studies [81, 80, 77], employs precise orbit and clock correction data to enhance positioning accuracy, particularly in environments where traditional single-frequency GNSS is severely degraded by multipath or atmospheric disturbances.

RTK positioning, as illustrated by [74], further underscores the feasibility of achieving centimeter-level accuracy using smartphone GNSS by integrating corrections from local base stations. However, its broader application remains limited due to dependency on infrastructure availability, higher computational complexity, and power consumption.

Additional literature contributions have focused on refining the robustness and reliability of GNSS measurements themselves without fully adopting infrastructure-dependent corrections. Techniques such as pseudorange smoothing using carrier-phase observations [94] and applying environmental error models (ionospheric and tropospheric corrections) [78] offer intermediate accuracy improvements without sacrificing practical smartphone deployment feasibility.

Overall, the selection of positioning techniques in smartphone GNSS literature reveals clear trade-offs: simplicity and real-time practicality (SPP), higher accuracy at the cost of complexity and resource demands (PPP, PPK, RTK), and intermediate strategies that balance accuracy and feasibility through measurement enhancement. These insights help clarify the rationale behind current research choices and suggest pathways for future development. Furthermore, the summary of these positioning techniques can be seen in the Table. 2.6.

Table 2.6: GNSS Positioning Techniques Used in Smartphone-Based Architectures

Technique	SPP	PPP	RTK	PPP-AR	Insights and Key Messages
Advantages	Simple implementation; minimal computational load; no external corrections required.	Sub-meter accuracy achievable; infrastructure-independent but requires precise orbit and clock products.	Centimeter-level accuracy in real-time; fast initialization; best suited for open-sky conditions.	High accuracy without real-time corrections; robust in dual-frequency and multi-GNSS setups.	Technique selection involves trade-offs between accuracy, complexity, and resource requirements. Practical choice heavily dependent on application and environmental constraints.
Representative Studies	[98, 72, 78, 75, 99, 84, 25, 85, 86, 100, 47, 93, 94, 87, 95, 89, 90, 96, 26, 28, 92, 91]	[80, 77, 81, 93]	[74]	[101]	Most studies rely on SPP for its ease of integration and real-time capabilities despite known accuracy limitations. Advanced methods (PPP, RTK, PPP-AR) remain underutilized but demonstrate significant potential for high-precision tasks.
Implementation Summary	Predominantly chipset-based or raw pseudorange; basic smoothing ([94]), environmental corrections ([78]).	Carrier-phase measurements with precise satellite corrections; primarily post-processing; dual-frequency often required.	Uses real-time differential corrections from base stations; high-precision and low latency.	Combines dual-frequency, multi-GNSS raw observations; ambiguity resolution enhances accuracy.	While advanced methods offer superior accuracy, they demand significantly more computational resources and power, limiting smartphone deployment feasibility. Future research must balance these aspects.
Practical Limitations	High sensitivity to multipath and ionospheric errors; limited accuracy in urban canyons.	Long convergence time; reliance on precise satellite products; not ideal for real-time mobile use.	Infrastructure dependency (base station); resource-intensive in computation and power.	Requires dual-frequency support; processing complexity remains high.	Understanding these limitations is critical for matching GNSS positioning methods with realistic use-case scenarios.

2.6.4 Overall Analysis of GNSS

The reviewed literature on smartphone-based GNSS positioning clearly demonstrates a prevalent preference for SPP, primarily due to its simplicity, minimal computational requirements, and ease of real-time implementation. As indicated in Table 2.7, SPP remains widely favored, especially in scenarios demanding low computational overhead or where infrastructure and dual-frequency (DF) support are unavailable. However, its widespread adoption comes with inherent limitations most notably susceptibility to multipath errors, atmospheric disturbances, and resultant accuracy constraints in urban environments.

In contrast, advanced positioning techniques such as PPP, PPP-AR, and RTK methods have shown substantial potential to overcome these accuracy limitations. Nonetheless, their usage remains comparatively rare, primarily due to practical barriers including dependency on external corrections, dual-frequency GNSS hardware availability, prolonged convergence periods, and higher computational and power demands unsuitable for many smartphone applications.

A notable gap identified in the literature is the limited integration of intermediate GNSS enhancement strategies, such as carrier phase smoothing and environmental error modeling, which could strike a balance between accuracy, complexity, and resource constraints. Techniques like these represent viable pathways for improving the accuracy of smartphone GNSS positioning without significant increases in computational complexity or external infrastructure reliance.

Overall, while advanced GNSS techniques promise substantial accuracy improvements, achieving practical and widespread adoption in smartphones demands further research into computational efficiency, device hardware optimization, and solutions tailored specifically to the constraints and use cases characteristic of mobile platforms.

Table 2.7: Summary and Insights from GNSS Techniques in Smartphone-Based Inertial+GNSS Fusion Architectures

GNSS Method	Freq.	Constellations	Studies	Remarks
Single Point Positioning (SPP)	SF	Primarily GPS, limited multi-GNSS	[73, 98, 72, 78, 75, 79, 76, 99, 83, 24, 84, 25, 85, 86, 100, 47, 94, 87, 95, 27, 89, 90, 96, 26, 28, 92, 91]	Predominantly chosen for simplicity, real-time feasibility, and low computational overhead. Easily deployable with minimal resource use. However, vulnerable to multipath and atmospheric interference, limiting performance in urban scenarios. Remains the practical standard for real-time smartphone GNSS integration.
Precise Point Positioning (PPP)	DF (L1/L5)	GPS, Galileo	[80, 77, 81, 93]	Achieves significantly higher accuracy compared to SPP by using precise orbit and clock corrections. However, typically requires dual-frequency GNSS data, external correction products, and longer convergence times, limiting real-time applicability. Ideal for high-precision applications accepting delayed results.
PPP with Ambiguity Resolution (PPP-AR)	DF	GPS, Galileo	[101]	Offers enhanced PPP accuracy through resolving carrier-phase ambiguities, achieving near-centimeter precision without real-time corrections. Still, it is computationally demanding, requires dual-frequency and multi-constellation support, and is challenging for real-time smartphone implementation.
Real-Time Kinematic (RTK)	DF	GPS	[74]	Enables real-time centimeter-level accuracy using differential corrections from a nearby base station. Despite its exceptional accuracy, RTK's dependency on external infrastructure and computational complexity restrict its broad practical use in typical smartphone scenarios.
Intermediate Techniques (Carrier Smoothing, Error Modeling)	SF	GPS	[78, 94]	Provide accuracy improvements without extensive external corrections or dual-frequency requirements. Carrier-phase smoothing ([94]) enhances pseudorange accuracy; error modeling ([78]) mitigates atmospheric errors. These techniques represent a practical balance between accuracy enhancement and complexity, suitable for constrained smartphone platforms.

2.7 Fusion methods

Fusion algorithms are the main block in any fusion system. They play an important role in combining the strengths and weaknesses of each input source. In the literature, three main categories of fusion algorithms have been identified for inertial+GNSS fusion on smartphones.

- Bayesian filtering-based fusion algorithms
- Graph optimization-based fusion algorithm
- AI/ML-based fusion algorithm

2.7.1 Kalman Filter (KF)

Background: The KF introduced by Kalman in 1960 is a recursive estimator that predicts the state of a dynamic system using linear state-space models [102]. It is optimal under linear dynamics and typically, but not strictly, Gaussian noise assumptions. As justified in [29], Gaussianity is not a strict requirement. The KF recursive structure mainly relies on first- and second-order statistics (mean and covariance), which makes it broadly applicable under non-Gaussian conditions as long as the moments exist.

A standard linear state-space model used in KF is given by:

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{v}_{k-1} \quad (2.11)$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k \quad (2.12)$$

where:

- $\mathbf{x}_k$ is the state vector,
- $\mathbf{u}_{k-1}$ is the control input,
- $\mathbf{F}_{k-1}$ is the state transition matrix,
- $\mathbf{B}_{k-1}$ is the control input matrix,
- $\mathbf{z}_k$ is the measurement vector,
- $\mathbf{H}_k$ is the observation matrix,
- $\mathbf{v}_{k-1} \sim \mathcal{N}(0, \mathbf{Q}_{k-1})$ and $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{R}_k)$ are process and measurement noise vectors, respectively.

The KF operates in two steps:

Prediction step:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} \quad (2.13)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (2.14)$$

Update step:

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \quad (2.15)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (2.16)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (2.17)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k \quad (2.18)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (2.19)$$

These recursive equations allow real-time state estimation under uncertain and noisy conditions [103, 102].

In the literature: The KF has been widely adopted in smartphone-based inertial+GNSS fusion, primarily because it provides a computationally efficient, real-time recursive solution for integrating noisy sensor data. Several studies have demonstrated the flexibility and effectiveness of KF by addressing practical implementation challenges such as noise modeling, parameter tuning, asynchronous sensor measurements, and computational resource constraints.

A common motivation behind using KF is its simplicity and low computational overhead, making it suitable for real-time smartphone implementation. For instance, studies such as [72, 79, 83] have emphasized KF's capability to fuse GPS and IMU data efficiently, even on resource-constrained microcontrollers, under the assumption that device orientation is either known or externally estimated. This highlights that careful selection of the KF state-space and noise parameters significantly improves navigation performance. Since accurate transformation of inertial measurements requires orientation information, several works extend the KF formulation to explicitly estimate heading or orientation states. Specifically, [83] leveraged inertial predictions corrected by GPS updates, effectively managing real-time computation demands with minimal drift.

Parameter tuning for the process and measurement noise covariance matrices is important for KF performance. Therefore, adaptive covariance approaches, such as in [78], demonstrated that reinforcement learning-based adjustments of these noise parameters greatly enhanced filter performance under dynamic pedestrian movements. Similarly, studies like [93] adopted a joint heading estimation approach, explicitly reducing KF parameter sensitivity by combining low-noise PDR heading estimates with high-noise GNSS heading, thus improving robustness to parameter uncertainties.

Managing asynchronous sensor data with different update rates between GNSS and inertial sensors remains a practical challenge. Studies [87, 88] specifically addressed this by employing covariance-varying, multi-rate KFs. These variants dynamically adapted state updates based on the availability and timing of GNSS measurements, optimizing both accuracy and power consumption on smartphones. This insight underscores the importance of careful handling of asynchronous measurements in real-world KF implementations.

Noise and error modeling have also been critical considerations. For instance, studies [78, 94, 96] employed advanced error modeling techniques within KF architectures, such as carrier-phase smoothing, multipath mitigation, and environmental error corrections (ionospheric and tropospheric models). Particularly, Zhu et al. [94]

used posterior residual analysis to detect and exclude faulty satellite signals, a technique that enhanced accuracy and reduced sensitivity to measurement anomalies.

Despite these strengths, practical KF implementations encounter several challenges. A common misconception is that KF strictly requires Gaussian noise; however, as clarified by Uhlmann and Julier [29], KF optimality does not depend on Gaussian noise assumptions but rather on minimizing the mean squared error given linear dynamics and known noise covariance. Nonetheless, real-world pedestrian navigation scenarios often involve nonlinear dynamics such as varying device orientation and irregular walking patterns that traditional KF struggles to handle effectively. While Extended Kalman Filters (EKF) or Unscented Kalman Filters (UKF) have been employed to address these nonlinearities, careful system modeling, noise covariance tuning, and robust handling of sensor data irregularities remain essential for reliable performance.

In summary, while the KF offers a robust and computationally efficient fusion solution, achieving optimal performance in smartphone-based inertial+GNSS architectures requires meticulous attention to noise modeling, parameter tuning, and asynchronous data management. Recognizing these insights from the literature provides essential guidance for future system design and implementation.

2.7.2 Extended Kalman Filter (EKF)

Background: The EKF is a nonlinear extension of the classical KF, developed to handle systems where either the state transition model or the measurement model, or both, are nonlinear [103]. The standard Kalman filter offers a recursive estimation framework that is optimal in the minimum mean squared error (MMSE) sense for linear systems, assuming zero-mean noise with known covariance. Although the KF is frequently associated with Gaussian noise, Gaussianity is not a strict requirement. As clarified by [29], the KF remains valid and optimal in the MMSE sense as long as the noise has finite first and second moments, that is, a well-defined mean and covariance. This distinction is particularly important in navigation systems, where sensor noise may be non-Gaussian but still suitable for Kalman-based filtering.

Many real-world systems, such as those encountered in inertial+GNSS integration, do not satisfy the linearity assumption. EKF addresses this limitation by linearizing the nonlinear models around the current state estimate at each time step, allowing the filter to approximate the behaviour of a nonlinear system using a locally linear representation [104].

In the standard KF, the system is modeled using linear stochastic difference equations, as described in the KF section. For nonlinear systems, these equations are replaced by: In nonlinear systems, these equations are replaced by:

Nonlinear State Transition:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \quad (2.20)$$

Nonlinear Measurement Model:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (2.21)$$

Here, $f(\cdot)$ and $h(\cdot)$ are nonlinear functions describing system dynamics and observations, respectively. EKF applies a first-order Taylor Series expansion [105] to linearize these functions around the current state estimate.

To use Kalman filtering with nonlinear functions, EKF computes the Jacobians matrices of first-order partial derivatives of the functions $f(\cdot)$ and $h(\cdot)$

State Transition Jacobian:

$$\mathbf{F}_{k-1} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}} \quad (2.22)$$

Measurement Model Jacobian:

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}} \quad (2.23)$$

These Jacobians approximate the nonlinear system with a linear model valid in a neighborhood of the current estimate. This approach is effective when the state estimate is sufficiently close to the true state, allowing the linearized model to remain valid over small deviations.

In practice, the Jacobians can be computed either analytically by differentiating the nonlinear functions manually or numerically, using finite differences. The correctness of these Jacobians is critical, as inaccurate derivatives can lead to divergence [106].

While the EKF provides a tractable solution for nonlinear estimation, it also has some limitations, particularly under dynamic noise conditions, model mismatch, or severe nonlinearity. Two important extensions of EKF are taken into consideration to tackle these: Adaptive EKF (AEKF) and Iterative EKF (IEKF).

Adaptive EKF (AEKF)

The EKF assumes that the process and measurement noise covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ are known, though they are not necessarily constant over time. However, in many real-world scenarios such as pedestrian or smartphone navigation, these noise levels may vary unpredictably due to changes in motion dynamics, signal blockage, or sensor behaviour [107]. The AEKF addresses this issue by updating the noise covariances online, using real-time statistics derived from the filter innovation (residuals). Common adaptation techniques include:

- Maximum likelihood or covariance matching using innovation sequences [108].
- Adaptive tuning based on innovation consistency or statistical thresholds [108].

This adaptive strategy allows the EKF to remain robust under non-stationary noise conditions, improving both stability and accuracy.

Iterative EKF (IEKF)

The EKF applies a single update based on a first-order linearization at each step. However, this can be insufficient when the measurement function is highly nonlinear.

In such cases, the update step may be iterated multiple times, re-evaluating the Jacobian and the innovation at each iteration until convergence is achieved [109].

The IEKF refines the posterior estimate by performing multiple linearizations within the update step. This leads to improved accuracy, particularly when the initial predicted state is far from the true value or when measurements have strong nonlinear characteristics.

Mathematically, IEKF performs:

$$\hat{\mathbf{x}}_k^{(i+1)} = \hat{\mathbf{x}}_k^{(i)} + \mathbf{K}_k^{(i)} \left(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^{(i)}) \right) \quad (2.24)$$

for $i = 1, \dots, N$ iterations until convergence.

The EKF framework extends linear Kalman filtering to nonlinear systems via local linearization. While powerful and widely used, it has limitations under strong nonlinearities or dynamic noise conditions. Adaptive and Iterative variants offer enhanced robustness and accuracy in such cases, making EKF and its extensions central tools in fusion systems and estimating location.

In the literature: In smartphone navigation [95] explored the fusion of heading and velocity information from inertial sensors with sparse and noisy GNSS updates to reduce drift while ensuring real-time responsiveness. The nonlinearities arose from orientation updates and the relationship between GNSS positions and the body frame. EKF's design, especially its use of Jacobian-based linearization was leveraged to preserve consistency despite these nonlinearities. A tightly-coupled fusion of low-cost sensors would have otherwise led to filter divergence or poor accuracy with simpler techniques.

In [80, 77], the central challenge was not just sensor fusion, but sensor synchronization. GNSS and IMUs operate at different frequencies, and GNSS updates may arrive irregularly. The EKF's prediction step allowed for state propagation between asynchronous updates, essentially serving as an interpolation mechanism. This predictive ability enabled consistent integration of IMU data with delayed GNSS observations, mitigating time misalignment. The authors' insight was that, beyond state estimation, EKF also serves as a temporal harmonizer, especially important when dealing with low-cost, loosely synchronized mobile sensors.

The study in [75] targeted degraded GNSS environments such as urban canyons and indoor transitions. Here, EKF was used to selectively incorporate Zero Velocity Updates (ZUPTs) when the system detected static periods. The key problem was navigation during GNSS signal outages, where inertial sensors alone would drift rapidly. EKF's structure allowed the dynamic integration of new measurement sources (e.g., ZUPTs) without requiring full reinitialization. This flexibility is particularly important in modular navigation pipelines where sensor availability may change at runtime.

In [74], EKF was applied to RTK systems that fused inertial data with GNSS carrier-phase and Doppler measurements. These observations are inherently nonlinear and require precise modeling of state-dependent uncertainty. The use of EKF was motivated by its ability to handle the nonlinearity in measurement models while updating uncertainty dynamically through its covariance propagation mechanism. The insight here is that EKF does not just manage mean state estimation, it also

serves as a real-time uncertainty engine, which is critical in RTK scenarios where estimation error bounds directly influence ambiguity resolution.

A particularly insightful application is found in [28], where EKF was used for pedestrian navigation in GNSS-denied environments. The authors implemented a step-based EKF that fused stride-length and heading information with GNSS when available. The core challenge was to maintain localization in buildings or underground areas. The filter's design allowed it to gracefully switch between GNSS-dominant and motion-dominant regimes, depending on signal availability. This illustrates EKF's ability to support mode aware estimation, where different measurement models are activated depending on context.

Finally, [98] introduced an AEKF in a dynamic pedestrian navigation system. Here, motion modes (walking, running, stopping) caused significant variation in the sensor noise characteristics. AEKF was employed to dynamically tune the process and measurement noise covariances based on real-time innovation statistics. This adaptation helped maintain consistency and avoided over- or under trusting the sensors under changing dynamics. The insight is that AEKF is not merely a robustness upgrade it enables the filter to be self-aware, responding to shifts in system behaviour or environmental quality.

2.7.3 Unscented Kalman Filter (UKF)

Background: The UKF emerged as a solution to the limitations posed by the EKF, particularly in highly nonlinear systems where the EKF's first-order linearization results in significant approximation errors. Rather than relying on Jacobians, the UKF employs the Unscented Transform (UT), which deterministically selects a set of carefully chosen sigma points that represent the mean and covariance of a Gaussian distribution. These sigma points are then propagated through the actual nonlinear functions to capture the system's true nonlinear behavior [108].

The key idea behind the UKF is that it is easier to approximate a probability distribution than it is to approximate a nonlinear transformation. As visualized in Fig. 2.16, unlike EKF, which uses a single linearized estimate, UKF transforms multiple sigma points through the nonlinear function, thus capturing the true mean and covariance more accurately. This allows UKF to handle stronger nonlinearities and reduce the numerical errors introduced by Jacobian approximations. The UKF continues to follow the familiar two-step predict-and-update structure but enhances accuracy, stability, and consistency.

UKF typically uses $2n + 1$ sigma points for an n -dimensional state space, and each sigma point is weighted to reconstruct the posterior distribution. The covariance and mean update steps involve computing a weighted average of the transformed sigma points, leading to improved higher-order moment approximations. The resulting filter is not only more accurate but also avoids the complex and sometimes intractable computation of Jacobians, which is especially valuable in real-time embedded systems or applications with black-box models.

In the literature: The UKF has been successfully used in GNSS/INS fusion where the system exhibits strong nonlinearity. For instance, in [71], the authors implemented a UKF variant to better represent nonlinear inertial dynamics while

simplifying parts of the measurement model. UKF was found particularly effective when model linearization (required by EKF) introduced errors larger than the sensor noise itself. A similar motivation appears in [82], where the authors used UKF alongside a backward smoother to improve trajectory consistency in smartphone-based pedestrian navigation, dealing with high levels of motion uncertainty and sparse GNSS updates.

Another major insight comes from comparing the propagation structure of EKF and UKF. While EKF relies on a first-order Taylor series (linearization) and propagates only the state estimate, UKF propagates the distribution through nonlinearities by transforming multiple sigma points (Figure 2.17). These points are chosen to capture the spread and shape of the prior Gaussian distribution, and they are transformed through the nonlinear functions to produce a more accurate estimation of the posterior. This makes UKF not only more robust to modeling errors but also numerically more consistent [110].

UKF is especially recommended in cases where Jacobians are difficult to derive or when the system dynamics change rapidly. For example, in systems involving orientation, quaternions, or pseudorange-based GNSS updates, where closed form Jacobians either do not exist or vary with system state, UKF avoids this burden entirely. This flexibility is also evident in aided navigation applications (e.g., strap-down systems), where UKF can operate in the whole state space rather than using error states only [110].

Despite its benefits, UKF does incur a slightly higher computational cost compared to EKF due to the multiple sigma point propagations and covariance reconstructions. However, for mid- to high-fidelity applications such as precision GNSS/INS integration, autonomous navigation, and robotics, this tradeoff is usually acceptable and even necessary to ensure estimator reliability.

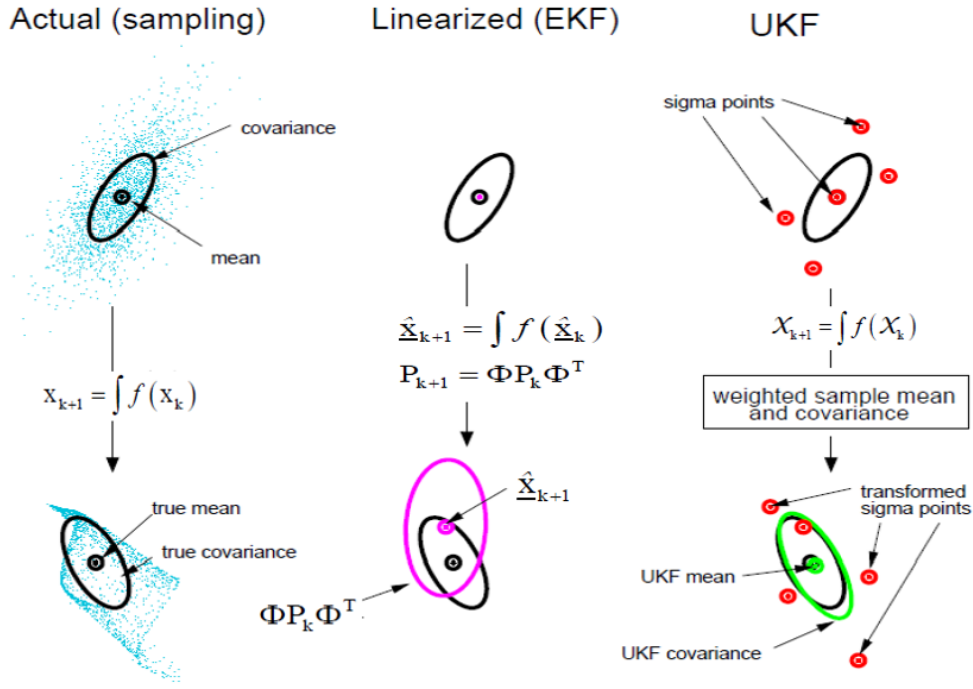


Figure 2.16: Comparison of actual distribution propagation, EKF, and UKF through a nonlinear transformation. UKF captures the transformed distribution more accurately [110].

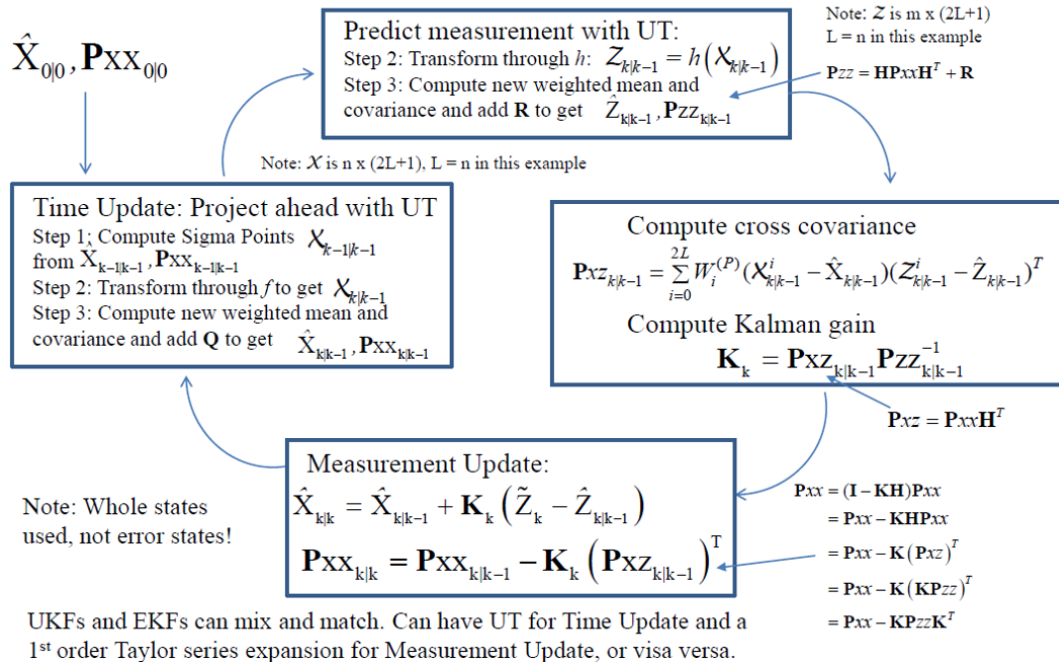


Figure 2.17: Unscented Kalman Filter loop showing the sigma point-based predict and update structure [110].

2.7.4 Particle Filter (PF)

The PF is a Monte Carlo-based Bayesian filtering technique designed to overcome the limitations of Kalman Filter variants (KF, EKF, UKF), especially in situations

involving strong nonlinearity and non-Gaussian noise [111]. Unlike KF, which requires linear models, or EKF and UKF, which approximate nonlinearity via Jacobian linearization or sigma point sampling, PF makes no such assumptions. Instead, it represents the posterior distribution of the state by a set of weighted particles, each denoting a possible state hypothesis. This allows PF to model arbitrary distributions, including multi-modal and skewed ones, which are often encountered in real-world navigation scenarios.

In PF, the state estimation process unfolds through four main steps as shown in the Table. 2.8.

Table 2.8: Steps of the Particle Filter algorithm

Step	Description
1. Initialization	A set of N particles $\{x_k^i\}_{i=1}^N$ is sampled from the initial probability distribution $p(x_0)$. These particles represent different hypotheses about the initial state of the system [111].
2. Prediction	Each particle is propagated forward through the nonlinear motion model to predict the next state based on system dynamics and process noise [111].
3. Weighting (Update)	The importance weight w_k^i of each particle is updated based on the observation likelihood $p(z_k x_k^i)$, which measures how well the particle's predicted state explains the observed measurement. Weights are normalized so that $\sum w_k^i = 1$ [111].
4. Resampling	When the number of effective particles N_{eff} falls below a threshold, resampling is triggered to discard low-weight particles and duplicate high-weight ones. This addresses sample degeneracy and maintains filter diversity [111].

Compared to EKF and UKF, PF's strength lies in its ability to deal with non-Gaussian posteriors and nonlinear transformations without approximations. However, this flexibility comes with a price: the need for a large number of particles, especially in high-dimensional state spaces, can lead to significant computational burdens. To mitigate this, adaptive resampling strategies, parallel computing, and hybrid techniques have been developed [112].

In terms of practical usage, PF is particularly attractive for applications with abrupt dynamics or ambiguous measurements, for instance, indoor pedestrian tracking, where GNSS signals may degrade or disappear. PF also shows value in robotic navigation where sudden jumps (e.g., kidnapped robot problem) occur. Unlike KF-based methods, PF can maintain multiple hypotheses and recover from unexpected shifts if particles cover sufficient state space [113].

In the literature: researchers have proposed enhancements to address PF's

computational demands. [84] proposed a Krill Herd Algorithm (KHA) [114] assisted PF, which introduced bio-inspired exploration behavior to guide particle diversity and optimize convergence during resampling. This allowed the filter to better cope with signal degradation in GNSS-denied environments by adapting the search space dynamically. Such hybrid methods highlight the growing trend of integrating PF with optimization techniques to improve robustness without relying on strict assumptions of process or measurement models.

Despite these improvements, PF remains challenged by scalability and sensitivity to initial distribution and resampling criteria. Particularly in scenarios with high measurement noise but limited computational resources, performance may degrade due to particle impoverishment. However, with continued algorithmic and hardware advancements, PF stands as a flexible and powerful estimator capable of modeling real-world complexity where Gaussian-based filters may fail [112].

2.7.5 Factor Graph Optimization (FGO)

Background: FGO originated in the robotics domain, specifically to solve Simultaneous Localization and Mapping (SLAM) problems [115]. Traditional SLAM systems often relied on recursive filters such as the EKF-SLAM [116], which became error-prone in large-scale nonlinear environments. In response, researchers proposed a smoothing-based alternative using factor graphs, FGO, which has emerged as a powerful and flexible framework for solving nonlinear state estimation problems. Unlike traditional Bayesian filtering techniques, which rely on sequential Bayesian updates, FGO takes a global optimization approach. It models the estimation problem as a probabilistic graphical model, specifically, a bipartite factor graph where nodes represent unknown states (variables) and edges (factors) represent probabilistic constraints arising from sensor measurements or process models [117].

Bayesian filters operate recursively often, updating only the current state based on the prior state and new measurement. While this is computationally efficient, it limits the ability to correct past estimates when new information becomes available. Additionally, EKF suffers from inaccuracies due to Jacobian linearizations and cannot handle multimodal distributions well. FGO, on the other hand, optimizes over a full history of states, thus enabling global consistency, better handling of nonlinearities, and the fusion of heterogeneous sensor data. FGO was first tested and applied in robotics and SLAM applications by researchers at Georgia Tech around 2012. The development of this method eventually led to the creation of the GTSAM library, which provided a practical and efficient implementation of FGO [117].

FGO gradually gained popularity in other areas beyond robotics, including multipath mitigation in GNSS and multi-sensor fusion [118, 119]. Furthermore, researchers have applied the FGO method to a range of challenging problems, showcasing its flexibility and robustness in complex navigation environments. For instance, it has been used to mitigate satellite signal degradation effects through robust optimization techniques [119], enabled scalable multi-sensor fusion for inertial and visual odometry [120], improved GNSS accuracy in dynamic environments [34, 121], and supported modular plug-and-play sensor integration frameworks that adapt to varying system configurations [118].

Over the last decade, FGO has expanded into broader applications such as inertial+GNSS fusion [34, 121, 120, 122], visual odometry [123], and smartphone-based localization [24, 84, 25, 85, 86, 100, 124]. In these domains, FGO has demonstrated superior accuracy over traditional filtering techniques, especially under nonlinear dynamics, asynchronous sensor measurements, and complex environmental conditions. This is exemplified in the work by Suzuki [124], who achieved decimeter-level positioning accuracy and won first place in the Smartphone Decimeter Challenge organized by the Institute of Navigation (ION) in 2021.

A factor graph is a bipartite graphical model that represents the structure of a probabilistic estimation problem, which represents how different pieces of information, like sensor readings or motion models, relate to the unknown quantities we want to estimate, such as position, velocity, or orientation, as shown in Fig. 2.18. It consists of two kinds of nodes:

- **Variable Nodes:** These represent the unknown values we aim to estimate, such as pose, velocity, and bias.
- **Factor Nodes:** These represent constraints or measurements that relate to one or more variables, such as GNSS, IMU, or motion models.

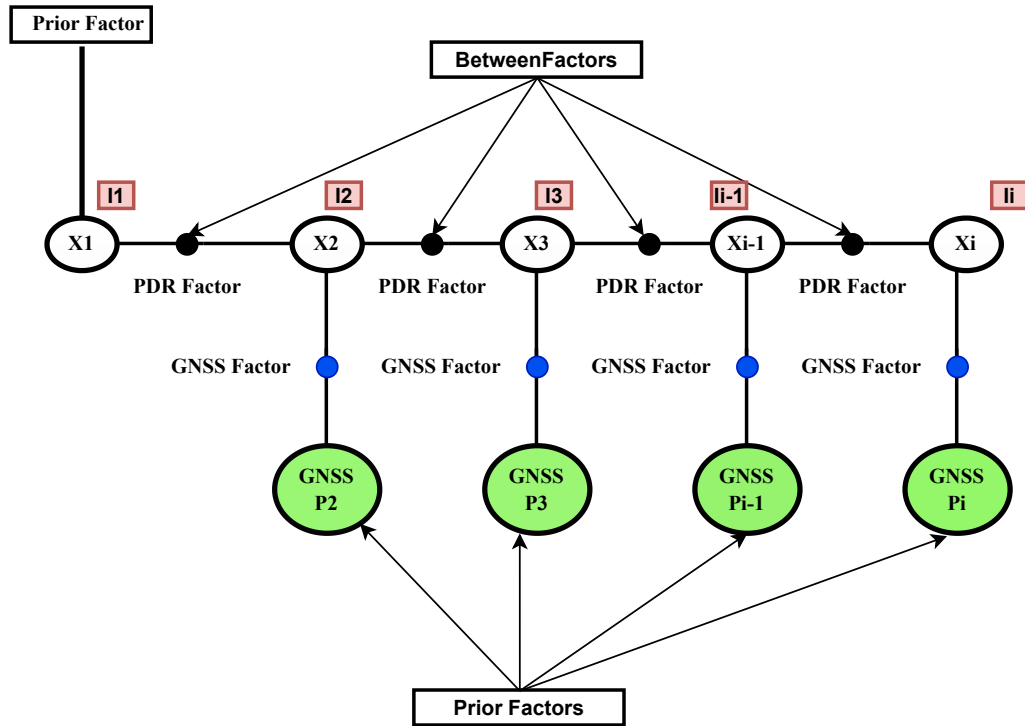


Figure 2.18: Factor graph structure for PDR+GNSS fusion.

Each factor node encodes a mathematical error term (residual) based on the difference between what the model expects and what is observed. The entire estimation problem becomes one of minimizing the total error across all factors, which can be expressed as a nonlinear least-squares problem:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_i \|r_i(\mathbf{x}_i)\|_{\Sigma_i^{-1}}^2 \quad (2.25)$$

where $r_i(\mathbf{x}_i)$ is the residual of the i -th factor, and Σ_i is the associated uncertainty (covariance). This process yields the most likely estimate of all unknowns, given all measurements and constraints.

FGO mainly consists of two steps:

1. **Nonlinear Graph Construction:** All variable nodes and factor nodes are added to the graph. This may include such as
 - IMU pre-integration factors between consecutive poses.
 - GNSS position measurements.
 - PDR position.
 - Heading, barometer, magnetometer, or ZUPT constraints.
 - Prior factors for initial conditions.
2. **Graph Optimization:** Once the graph is constructed, a nonlinear optimization technique such as Gauss-Newton or Levenberg-Marquardt [125] is used to minimize the total residual error. Jacobians are computed analytically or via automatic differentiation.

In the literature: In smartphone-based navigation, several studies have investigated the application of FGO for PDR+GNSS fusion [24, 84, 25, 85, 86]. Most of these adopt a batch optimization approach, where all historical states and measurement factors are optimized together in a single graph. While this method provides high accuracy due to full-horizon smoothing, it is computationally intensive and unsuitable for real-time implementation on smartphones.

To mitigate this issue, windowed FGO approaches have been proposed. A commonly used technique is the Naïve Sliding-Window FGO (NW-FGO), where older variables and factors beyond a fixed time horizon are discarded, and the latest posterior estimate is used to reinitialize the graph as a new prior. Although simple and lightweight, this strategy suffers from information loss, as previously optimized constraints are not retained.

To address this, Marginalization-Based Windowed FGO (MW-FGO) has been developed, where outdated variables are marginalized rather than discarded. This preserves their informational contribution in compressed form by introducing a prior factor using the Schur complement. Mathematically, if the full system is partitioned into retained variables $\mathbf{x}_r$ and marginalized variables $\mathbf{x}_m$, and the system information matrix is partitioned as:

$$\begin{bmatrix} \mathbf{H}_{mm} & \mathbf{H}_{mr} \\ \mathbf{H}_{rm} & \mathbf{H}_{rr} \end{bmatrix} \begin{bmatrix} \delta \mathbf{x}_m \\ \delta \mathbf{x}_r \end{bmatrix} = \begin{bmatrix} \mathbf{b}_m \\ \mathbf{b}_r \end{bmatrix} \quad (2.26)$$

then marginalization results in a prior for $\mathbf{x}_r$ with reduced information:

$$\begin{aligned}\mathbf{H}'_{rr} &= \mathbf{H}_{rr} - \mathbf{H}_{rm} \mathbf{H}_{mm}^{-1} \mathbf{H}_{mr}, \\ \mathbf{b}'_r &= \mathbf{b}_r - \mathbf{H}_{rm} \mathbf{H}_{mm}^{-1} \mathbf{b}_m\end{aligned}\tag{2.27}$$

This marginalized prior is then injected back into the factor graph to maintain a consistent and informative estimate over time.

The windowing strategies discussed earlier, such as Naïve and Marginalization-Based FGO, were primarily implemented using the Levenberg–Marquardt (LM) optimizer [125]. While LM is well-suited for batch nonlinear optimization, more general and scalable alternatives exist for incremental smoothing, particularly when marginalization is involved.

One of the most widely adopted frameworks for this purpose is iSAM2 (incremental smoothing and mapping) [126], a key optimizer provided in the Georgia Tech Smoothing and Mapping (GTSAM) library [117]. iSAM2 enables efficient real-time optimization by incrementally updating the factor graph and relinearizing only affected variables. This makes it highly suitable for systems requiring constant time updates while maintaining the benefits of global smoothing.

The GTSAM library also provides flexible tools to define nonlinear factors, handle marginalization through the Schur complement, and implement a wide range of fusion architectures. These capabilities significantly reduce the engineering effort needed to prototype and deploy FGO pipelines.

Although such incremental and marginalized optimizers have not yet been fully explored in smartphone-based PDR+GNSS systems, they have been successfully applied to GNSS/INS integration problems. For example [127] leveraged factor graph techniques to enhance urban navigation performance by tightly coupling inertial measurements with GNSS pseudorange constraints using sliding window and smoothing-based FGO.

Given the increasing computational power of smartphones and the modularity of tools like GTSAM, integrating iSAM2-based marginalization strategies into real-time mobile localization remains a promising direction for future research.

Several open-source libraries support the implementation of FGO, enabling both batch and incremental solutions suitable for GNSS/INS and multi-sensor fusion problems:

- **GTSAM** [117]: A widely-used C++ library with Python bindings developed at Georgia Tech. It provides tools for building factor graphs and includes solvers like Gauss-Newton, Levenberg-Marquardt, and the incremental iSAM2 optimizer for real-time applications.
- **Ceres Solver** [128]: A C++ library by Google for solving large-scale nonlinear least squares problems. While not a full factor graph library, it can be used to construct FGO backends manually and supports robust loss functions and automatic differentiation.
- **g2o** [129]: A general framework for optimizing graph-based nonlinear error functions, originally developed for SLAM. It allows flexible graph construction and supports different solvers and linear algebra backends.

These characteristics make FGO particularly suitable for pedestrian PDR+GNSS fusion in the literature, where nonlinear motion, intermittent measurements, and model imperfections are prevalent.

2.7.5.1 KF vs.FGO: theoretical equivalence and practical advantages in pedestrian PDR+GNSS fusion

The analysis in this thesis shows that FGO-based PDR+GNSS fusion can outperform Kalman-based solutions in the considered pedestrian scenarios. However, under ideal conditions, Kalman filtering and smoothing are optimal estimators and solve the same underlying least-squares problem as FGO.

Recent studies have clarified that the practical differences observed between these fusion approaches are primarily driven by modelling assumptions and implementation choices, including the treatment of temporal correlations, the use of historical measurements, and the adopted optimization structure, rather than by fundamentally different probabilistic objectives [130, 131, 132]. These aspects are further discussed in the following point

Equivalence under linear-Gaussian assumptions: From an estimation-theoretic perspective, both KF-based methods and FGO aim to solve the same Bayesian inference problem. When the system dynamics and measurement models are linear, and when all uncertainties can be accurately described as independent Gaussian noise, the resulting posterior estimation problem reduces to a weighted least-squares formulation. Under these restrictive conditions, recursive Kalman filtering, fixed-interval smoothing, and batch optimization via factor graphs are mathematically equivalent in the sense that they converge to the same optimal solution.

Recent studies have emphasized that this equivalence is not accidental but a direct consequence of the underlying assumptions. In particular, Song et al. [131] demonstrate that, when a factor graph is constrained to a single-state window, employs a quadratic (L2) cost, and enforces Markovian state transitions, the resulting optimization becomes equivalent to EKF-family estimators. Similarly, Taylor and Gross [130] show that Kalman filtering can be interpreted as a specific instance of probabilistic inference on a factorized graph, where the apparent differences between KF and FGO stem primarily from how the problem is structured and solved, rather than from fundamentally different probabilistic objectives.

This perspective highlights that FGO does not derive its practical advantages from violating Kalman filtering theory but rather from relaxing or extending the assumptions under which that theory is strictly optimal. As discussed in the following sections, these assumptions are routinely violated in pedestrian PDR+GNSS scenarios, which explains why performance differences emerge in practice despite theoretical equivalence under idealized conditions.

Why pedestrian PDR+GNSS violates ideal assumptions: The smartphone-based pedestrian fusion problem considered in this thesis operates far from the ideal conditions under which KF/KS optimality and theoretical equivalence with FGO strictly hold. Pedestrian motion exhibits inherently nonlinear behavior, with frequent and abrupt changes in heading, velocity, and device orientation caused by

gait variability and unconstrained phone handling. These effects lead to dynamics that are poorly captured by simple linear motion models and challenge single-pass linearization strategies.

At the measurement level, GNSS observations in pedestrian environments, particularly in urban and semi-urban settings, are often affected by multipath and NLOS reception. Such effects introduce biased, heavy-tailed, and temporally correlated errors that deviate significantly from the Gaussian noise assumptions commonly adopted in Kalman-based formulations. In parallel, PDR estimates can produce intermittent outliers due to missed or spurious step detections, heading disturbances, and step-length misestimation, further violating the assumptions of independent and well-modeled measurement noise.

In addition, both process and measurement noise statistics vary substantially with environmental context, satellite visibility, and carrying mode, resulting in persistent model mismatch and covariance uncertainty. These combined effects undermine the conditions required for KF/KS optimality and have been identified in recent comparative studies as precisely the regimes in which sliding-window FGO can offer practical advantages over EKF-style single-pass estimation. The impact of these non-idealities, and the corresponding behavior of FGO relative to Kalman-based solutions, is analyzed in detail in Chapters 3, 4, and 5 through the experimental results reported in the associated publications.

Practical advantages of FGO in this thesis context: Given the above non-idealities, FGO provides practical advantages that explain the improvements observed in pedestrian PDR+GNSS fusion:

- **Explicit treatment of historical measurements (smoothing by construction):** Unlike recursive Kalman-based estimators, which marginalize past information at each update step, FGO explicitly estimates a window of states jointly. This formulation allows newly available measurements to influence not only the current state but also previous states within the optimization window, effectively redistributing corrections over time. Such behavior is particularly advantageous in pedestrian PDR+GNSS fusion, where GNSS updates may be intermittent, delayed, or temporarily unavailable, and where PDR constraints must maintain trajectory stability during outages. As discussed by Taylor and Gross, factor graphs naturally support smoothing over a history of states, and real-time implementations can retain these benefits through windowed optimization and marginalization strategies [130].
- **Iterative optimization and relinearization of past states:** A key limitation of EKF-based estimators in nonlinear systems is that model linearization is performed only once, around the current state estimate, at the time each measurement is processed. If this linearization is inaccurate due to strong nonlinear motion, abrupt heading changes, or poor initial estimates the resulting errors can propagate forward and are difficult to correct retrospectively.

In contrast, FGO formulates the estimation problem as a nonlinear optimization over a window of states and solves it iteratively. As the solution improves, the underlying linearizations can be updated, allowing both current and past

states within the window to be reinterpreted based on improved estimates. This iterative relinearization is particularly important in pedestrian PDR+GNSS fusion, where motion dynamics and measurement relationships are highly nonlinear. Recomputing Jacobians for affected states can significantly improve estimation accuracy and consistency, especially when compared to single-pass EKF updates [130]. Song et al. further explain that such behavior represents a key mechanism by which sliding-window FGO departs from EKF equivalence in strongly nonlinear regimes [131].

- **Robustness to real-world effects and non-Gaussian PDR errors:** In pedestrian PDR+GNSS fusion, many dominant error sources arise from human motion variability rather than from sensor noise. Step-length estimation errors, heading disturbances, missed or false step detections, and abrupt changes in smartphone orientation caused by different carrying modes or gait patterns produce error distributions that are strongly non-Gaussian and often contain outliers. These effects violate the assumptions of Gaussian residuals underlying Kalman-based estimators, making them particularly sensitive to transient PDR failures.

Within the FGO framework, such non-ideal behaviors can be handled more naturally through robust cost functions and factor formulations that reduce the influence of outliers without destabilizing the overall estimate. This allows sporadic PDR errors caused by irregular walking behavior or device handling to be absorbed locally, rather than propagating through the entire trajectory. The factor-graph formulation provides a flexible mechanism for incorporating robust losses within the same optimization structure [130]. Song et al. further highlight that robustness to non-Gaussian errors constitutes a key regime in which sliding-window FGO departs from EKF equivalence and yields practical advantages in real-world navigation scenarios [131].

Time-correlated errors and non-Markov behavior: A fundamental source of performance difference between Kalman-based estimators and FGO in smartphone pedestrian positioning arises from the presence of time-correlated errors. In pedestrian PDR, dominant error sources such as step-length estimation biases, heading drift, gait-dependent dynamics, and smartphone carrying-mode variations evolve gradually over time rather than appearing as independent, white-noise disturbances. These effects introduce temporal correlations that violate the Markov and white-noise assumptions commonly adopted in KF/EKF formulations.

In Kalman-based approaches, such correlations must be approximated indirectly through covariance inflation, heuristic tuning, or explicit state augmentation, which increases model complexity and can lead to inconsistency or instability when the assumed error structure does not match reality. In contrast, FGO naturally accommodates temporal correlations by explicitly linking consecutive states through motion and PDR factors and by jointly optimizing a window of states. This allows correlated errors to be distributed across time rather than being absorbed instantaneously at a single update.

Recent studies have shown that the ability of FGO to exploit time-correlated measurements and states is a key reason for its superior robustness compared to EKF in real-world navigation problems. Wen et al. demonstrate that FGO can explicitly model and leverage temporal correlations to improve robustness against persistent measurement errors, while EKF-based estimators remain sensitive to such effects [132]. Similarly, Song et al. and Taylor and Gross identify time correlation and windowed smoothing as critical regimes in which sliding-window FGO departs from KF equivalence and yields measurable practical advantages under non-ideal conditions [131, 130].

- **Tolerance to model mismatch and heterogeneous constraints:** Pedestrian PDR+GNSS fusion requires the integration of heterogeneous constraints, including step-wise PDR increments, heading estimates, GNSS position updates, and prior information, whose reliability and error characteristics vary over time and across environments. In practice, accurately modeling the relative confidence of these constraints is challenging and often leads to persistent model mismatch in Kalman-based formulations. FGO provides a more flexible representation in which diverse constraint types can be incorporated explicitly and adapted without restructuring the entire estimation framework. As emphasized by Taylor and Gross, this structural flexibility allows factor graphs to accommodate changing measurement conditions and modeling assumptions more naturally than recursive Kalman filters, contributing to improved robustness in real-world pedestrian navigation scenarios [130].

In summary, under linear-Gaussian assumptions and matched configurations, KF and FGO should provide comparable solutions. The practical superiority of FGO observed in this thesis arises because smartphone-based PDR+GNSS fusion is dominated by nonlinearity, non-Gaussian measurement corruption, intermittent updates, and model mismatch. In this case, windowed/iterative FGO, together with robust cost functions and explicit smoothing over history, provides advantages that are consistent with the literature and with the experimental findings presented in this thesis.

2.7.6 Artificial Intelligence and Machine Learning (AI/ML)

Background: Although ML / AI has seen widespread application in fields such as computer vision, natural language processing, and autonomous driving, their use in navigation and localization, particularly in inertial + GNSS fusion architectures, remains limited [43]. Despite the clear potential, most existing smartphone-based fusion systems continue to rely on traditional filtering or optimization-based methods such as Kalman Filters or FGO.

However, the fundamental capabilities of ML/AI make them highly suitable for this domain [133]. These models can learn complex, nonlinear relationships between noisy sensor inputs and accurate position estimates, which are difficult to model analytically. For instance, a deep neural network can be trained to directly estimate position displacement from raw IMU and GNSS sequences, effectively learning the underlying motion model and noise behavior. Similarly, recurrent neural networks

(RNNs) or transformers can model temporal dependencies to smooth trajectories or detect anomalies in GNSS signals [134, 135].

A hybrid system might use a neural network to predict position increments from IMU data and correct them with GNSS updates when available. Alternatively, ML models can act as learned motion priors within an FGO framework to improve estimation in degraded environments. Despite these opportunities, literature applying such approaches is sparse, highlighting a valuable direction for future research and development.

In principle, ML could be used for full sensor fusion, but in practice, most existing works do not apply ML/DL to perform direct inertial+GNSS fusion. Instead, ML is more commonly used for aiding tasks such as PDR enhancement, velocity estimation, context recognition, or detecting degraded GNSS conditions (e.g., NLOS detection). While these techniques are often described in the context of sensor fusion, their role is typically supportive rather than acting as the core fusion engine.

In the literature: Current literature shows limited use of ML/DL for direct fusion of GNSS and inertial data. For instance, [91] used a feedforward ANN for improving PDR, but not for fusing it with GNSS. The model was trained over 100 epochs with ELU activation and Adam optimization, and was evaluated across various data configurations, including refined and augmented inputs.

[91] also evaluated the performance of multiple MLP architectures on inertial data using both simple and complex models. These models were assessed primarily on their ability to improve dead reckoning performance.

Although some studies refer to ML/DL in the context of fusion, closer inspection often reveals their use in preprocessing, aiding, or correcting rather than direct fusion. This insight is aligned with broader reviews, such as [43], which noted that ML is generally used for signal classification, motion context awareness, velocity correction, or data augmentation in GNSS/INS systems.

2.7.7 Overall Analysis of Fusion Methods

A concise analysis of the surveyed fusion methods highlights the evolution of strategies to address the inherent challenges of inertial+GNSS fusion systems. Bayesian filtering methods (KF, EKF, UKF, PF) remain the backbone of most real-time systems due to their recursive nature and computational efficiency. FGO, on the other hand, introduces a powerful global optimization framework that enables robust estimation even under degraded GNSS conditions. It supports flexible factor modeling and benefits from optimization toolkits like GTSAM and Ceres solver.

Moreover, it is evident that most existing ML/DL applications in navigation focus on auxiliary tasks such as NLOS detection, context awareness, or velocity estimation, rather than directly performing GNSS and inertial sensor fusion. This observation highlights a gap in the current literature and suggests a promising direction for future research: deeper integration of ML techniques within core fusion architectures. Moreover, the comparison of the existing fusion methods can be seen in the Table. 2.9. Among the reviewed fusion architectures, optimization-based approaches provide structural flexibility that is later exploited in this thesis for pedestrian PDR+GNSS fusion using FGO.

Table 2.9: Comparison of Sensor Fusion Methodologies

Fusion Method	Handles Nonlinearity	Supports Non-Gaussianity	Real-Time Feasibility	Covariance/Mean Handling	Computational Demand
KF	✗	✓ [†]	✓	Closed-form mean/covariance update (linear model)	Low
EKF	✓	✗	✓	First-order linearization (Jacobian) of mean/covariance	Moderate
UKF	✓	✗	Moderate	Sigma point approximation (unscented transform)	High in high dimensions
PF	✓	✓	✗	Non-parametric sampling (particles)	Very High
FGO	✓	✓	Moderate	Optimization-based weighting; no recursion; global consistency	High
AI/ML	✓	✓	✓	Learned latent representation; implicit uncertainty handling	Varies by architecture

[†]While KF is optimal under Gaussian assumptions, it remains applicable with non-Gaussian noise. It still yields the Best Linear Unbiased Estimate (BLUE), as discussed in [29].

2.8 Inertial+GNSS Fusion Architectures in Smartphones

To systematically present the landscape of Inertial+GNSS fusion architectures, a taxonomy was developed based on a comprehensive analysis of the literature, as shown in Fig. 2.19. It is worth noting that each type of inertial navigation system can be integrated with various GNSS processing techniques, resulting in distinct fusion architectures that are further characterized by integration levels and fusion methods. For the sake of simplicity, this taxonomy is illustrated in Fig. 2.20.

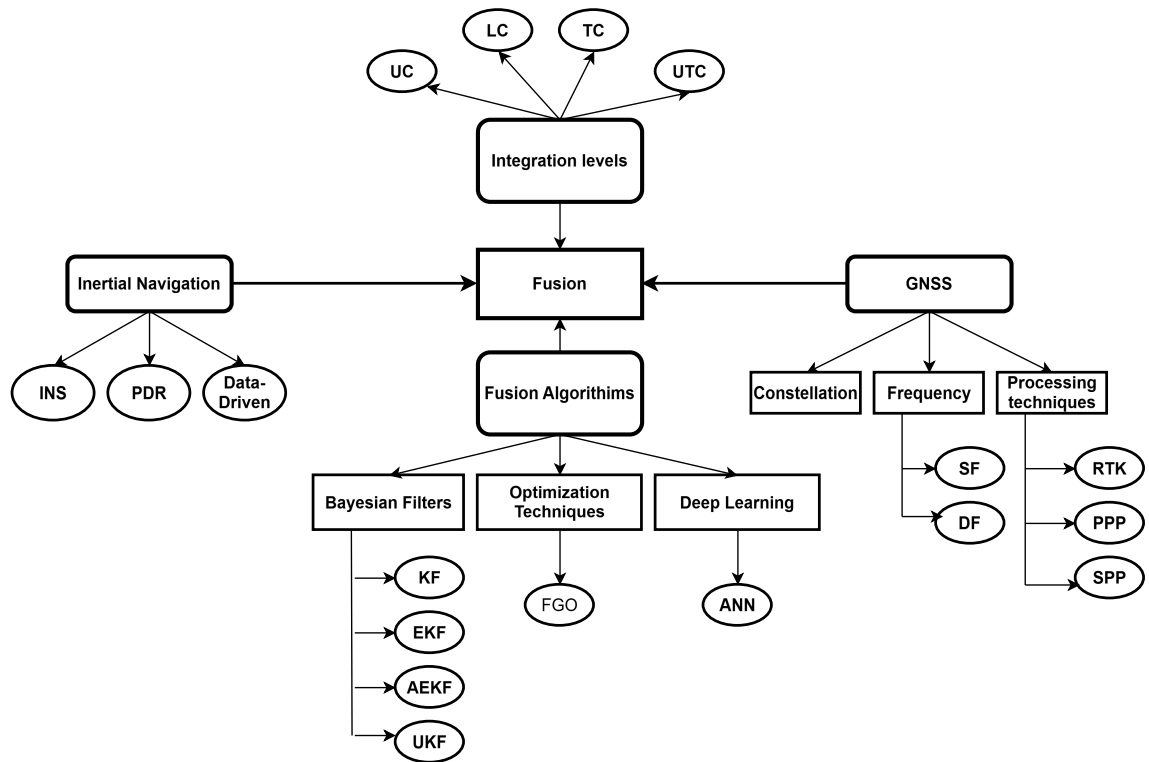


Figure 2.19: Taxonomy of Smartphone-Based Inertial+GNSS Fusion Architecture Derived from Existing Literature.

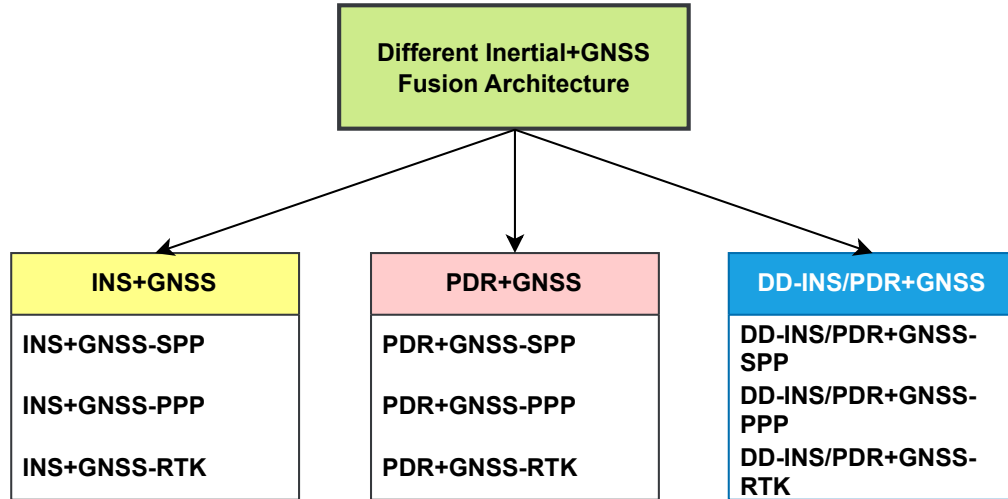


Figure 2.20: Classification of Inertial+GNSS Fusion Architectures in Smartphone-Based Systems.

2.8.1 Integration Levels

Fusion architectures are commonly integrated by different levels of integration, each representing a unique strategy for integrating sensor data. Although these architectures often utilize similar dynamic models, they differ significantly in how GNSS observations are used, affecting both the system's complexity and overall

positioning accuracy. These integration levels are briefly described below and are summarized in Table 2.10.

2.8.1.1 Uncoupled (UC)

The uncoupled integration strategy represents the most basic form of combining GNSS and INS data. In this architecture, both systems operate independently, and there is no direct fusion at the algorithmic level.

As shown in Fig. 2.21, the GNSS module produces standalone position and velocity outputs. These outputs are then periodically used to reset or correct the drift in the INS solution. The INS, on the other hand, continuously propagates position, velocity, and orientation estimates using inertial measurements (accelerometer and gyroscope data) but without incorporating GNSS measurements in real time.

This approach is sometimes referred to as "loosely reset-based integration" or "feedback correction". Since the GNSS and INS operate as separate blocks, the correction from GNSS is only applied at discrete intervals, typically when GNSS data is available and deemed reliable [23, 99].

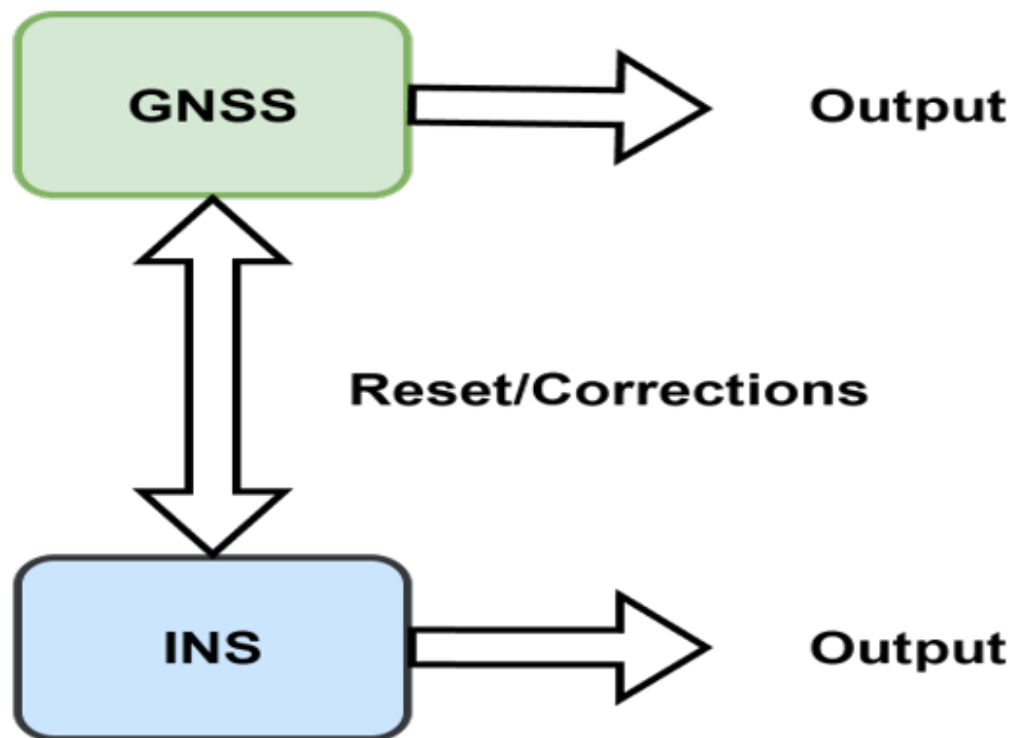


Figure 2.21: Uncoupled Integration Strategy.

2.8.1.2 Loosely Coupled (LC)

Loosely coupled (LC) integration, also referred to as decentralized or cascaded filtering, is one of the most commonly adopted approaches in Inertial+GNSS fusion. In this architecture, the fusion algorithm utilizes the processed outputs from the GNSS receiver, such as position, velocity, and time (PVT), rather than raw satellite measure-

ments. These GNSS-derived states are then used to update the inertial navigation solution. The general structure of LC integration is illustrated in Fig. 2.22 [136, 21].

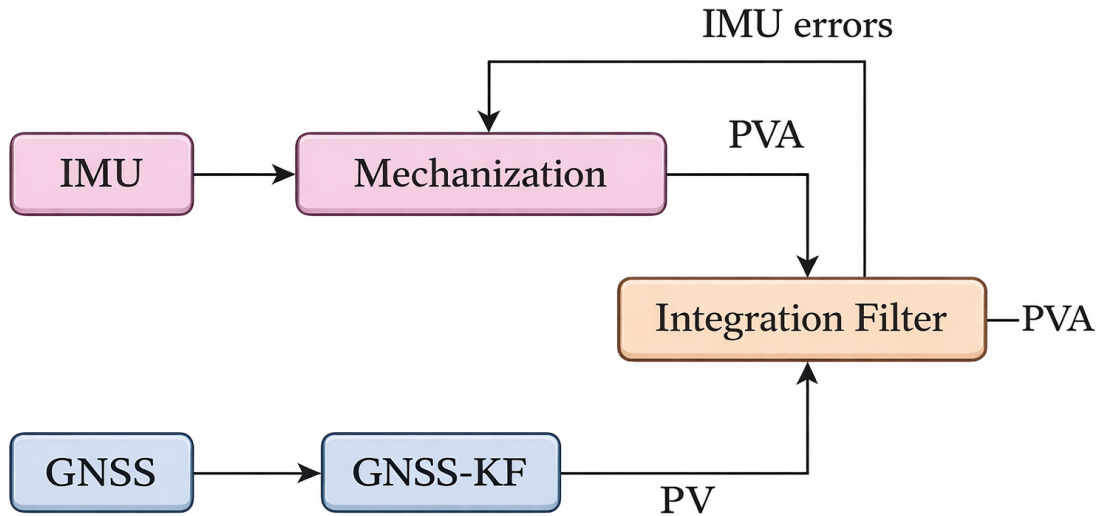


Figure 2.22: LC integration level [136].

2.8.1.3 Tightly Coupled (TC)

The raw GNSS measurements are combined directly with the IMU data in an appropriate fusion algorithm during the TC integration. This is sometimes referred to as a centralized system, and it is accomplished by processing INS navigation parameters and GNSS raw signals in a central algorithm, as illustrated in Fig 2.23.[21]

2.8.1.4 Ultra-Tightly Coupled (UTC)

UTC integration is the most sophisticated level of fusion architecture, in which the GNSS and INS subsystems are no longer treated as loosely/tightly connected but instead form a deeply integrated architecture. Unlike LC or TC systems that operate at the data level, UTC integration works at the signal processing level, requiring access to the receiver's baseband correlator outputs, such as the in-phase (I) and quadrature (Q) components. Accessing low-level data usually needs changes to the receiver's firmware, so this approach is mostly practical for GNSS receiver manufacturers or systems that use software-defined GNSS receivers [137].

In a UTC system, the INS outputs are directly fed into the GNSS receiver's tracking loops. These measurements are used to dynamically assist the receiver's code and carrier tracking by predicting Doppler shifts and code phase changes caused by user motion. Simultaneously, GNSS observables contribute to estimating and correcting INS drift errors through an integration Kalman filter. This bidirectional information

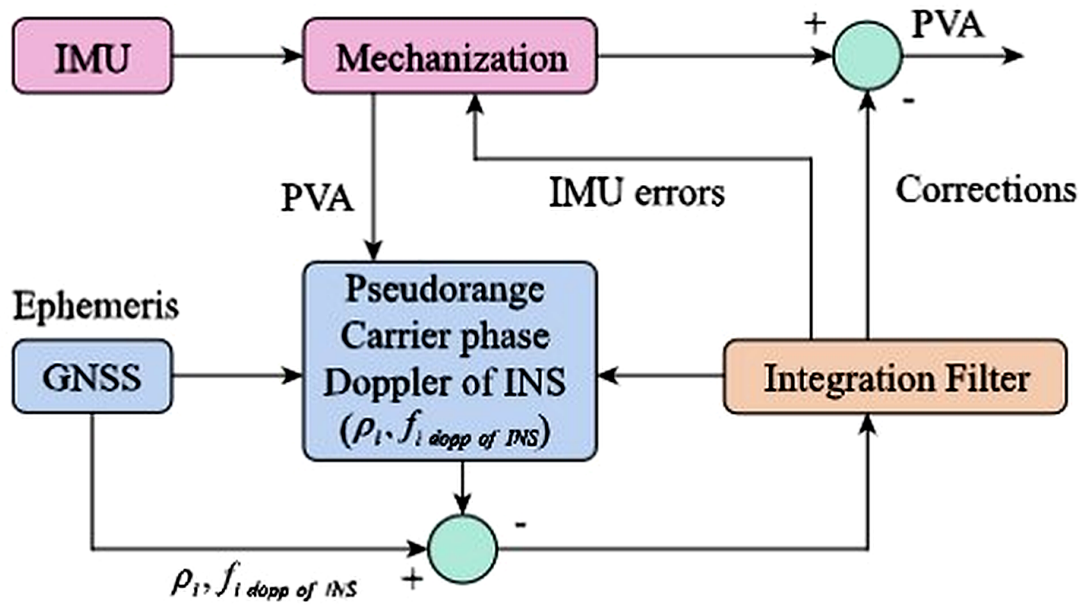


Figure 2.23: TC integration level [136].

flow tightly synchronizes both systems and significantly enhances robustness against jamming, multipath, and signal outages.

The integration Kalman filter within the UTC framework includes signal tracking error estimators that replace conventional code and carrier loop filters. As highlighted by Hwang et al. [137]. This unified approach facilitates multiple implementations, including discriminator-based, pre-filter-based, and Extended Range Correlator (ERC)-based designs. These variations differ in how tracking errors are estimated, but all aim to improve the performance of GNSS receivers under dynamic and degraded signal conditions.

Simulation results confirm that UTC systems outperform LC and TC configurations in both navigation accuracy and resilience to RF interference. For example, the ERC-based UTC system demonstrated the lowest vertical RMS error and superior jamming resistance in scenarios where continuous wave (CW) interference was applied at varying J/S ratios. Moreover, it can be seen in the Fig. 2.24 [137].

By integrating at the physical layer and synchronizing feedback between INS and GNSS signal processing, the ultra-tightly coupled architecture represents a powerful solution for high-precision and interference-resistant navigation applications.

2.8.2 INS+GNSS Fusion Architecture

INS+GNSS fusion in smartphone-based positioning systems is shaped by the need to mitigate the limitations of each sensor modality while accommodating the hardware constraints of the smartphone. The main goal of these architectures is to maintain continuous, accurate positioning in real-world environments where GNSS alone fails and where low-cost IMUs onboard smartphones are prone to drift and noise. The integration of INS and GNSS is not just a theoretical solution, but a practical necessity driven by the non-ideal conditions of smartphone use. A review of the literature shows that INS fuses mainly with SPP, PPP, and RTK GNSS position-

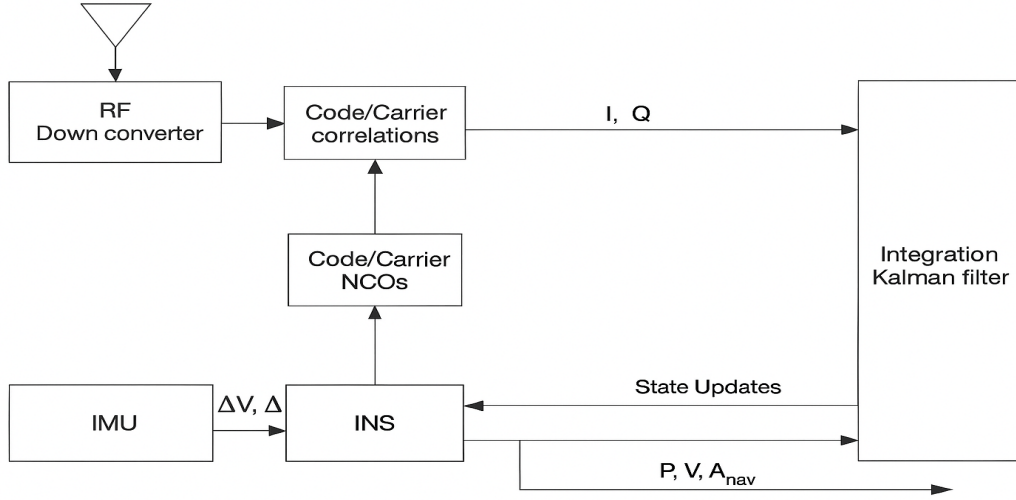


Figure 2.24: UTC integration level [137].

ing techniques (see Fig. 2.20). Each technique is selected based on the accuracy requirement, computational budget, and environmental constraints of the target application.

INS+SPP is widely adopted in smartphone-based studies due to its low complexity, minimal infrastructure dependency, and compatibility with standard Android APIs. It is well-suited for pedestrian navigation, basic localization services, and consumer-level applications where the accuracy demands are moderate and consistent GNSS visibility is expected [73, 98, 72, 79].

INS+PPP is used when high positioning accuracy is needed (often below one meter), but without depending on real-time corrections such as RTK because PPP uses precise satellite orbit and clock corrections, which can be obtained from global correction services or downloaded from official sources such as the International GNSS Service (IGS), Natural Resources Canada (NRCAN), or other regional agencies [138, 139, 140, 141]. This method is computationally heavier and may suffer from longer convergence times, but it provides a valuable trade-off for applications like geo-tagging, fitness tracking, and mobile mapping that require improved accuracy without needing dense infrastructure support [80, 77].

INS+RTK, while less common in consumer-grade applications, offers centimeter-level precision by incorporating real-time corrections from a base station. This integration is highly accurate but relies on an external communication link and close proximity to reference stations. It is primarily used in infrastructure-supported environments such as autonomous driving, UAV operations, and professional surveying, where accuracy and latency are critical performance metrics [74]. The integration of RTK data with inertial systems helps reduce ambiguity resolution time and enhances reliability during GNSS signal interruptions.

In INS+GNSS fusion systems, loosely coupled integration levels are commonly used not because of technical limitations, but also because they are easier to implement and more practical for real-world use, especially on smartphones. Although raw

Table 2.10: Comparison of different fusion system integration levels.

Integration Level	Method	Benefits	Drawbacks
Loosely Coupled (LC)	Integration occurs at the PVT solution level using decentralized or cascaded filtering.	- Simple and flexible implementation. - Reduced filter size compared to tightly coupled approaches.	- Each subsystem (GNSS/INS) must produce independent solutions. - Lower positional accuracy, particularly in degraded GNSS conditions.
Tightly Coupled (TC)	Integration occurs at the observation level using centralized filtering methods.	- Requires only one system to produce independent measurements. - Capable of identifying and rejecting erroneous measurements.	- More complex than LC integration. - Requires prior compensation for systematic sensor errors in smartphones.
Ultra-Tightly Coupled (UTC)	Integration is performed at the physical signal tracking layer using centralized architecture.	- Offers the highest accuracy among all integration strategies. - Enhances GNSS signal tracking under challenging conditions.	- Rigid architecture—typically fixed during hardware manufacturing. - Most complex among all integration levels.

GNSS measurements are now accessible on many smartphones, LC-based architectures remain popular due to their simplicity and ease of integration, as they rely on readily available position and velocity output from GNSS modules [78, 75, 79, 136]. Additionally, their modularity simplifies integration with the Android GNSS APIs and standard IMU interfaces. This design choice supports real-time operation, reduces development complexity, and ensures broader device compatibility. It also permits easier adaptation to varying phone orientations and sensor quality levels, which are commonly encountered in user-generated data.

Among the various fusion methods used in INS+GNSS architectures, the EKF is the most widely chosen due to its potential to handle effectively the non-linear nature of sensor measurements and coordinate transformations while maintaining computational efficiency suitable for real-time applications. Since INS and GNSS data operate in different reference frames and update at different rates, accurate state estimation depends on careful handling of timing, orientation, and velocity alignment. The EKF is well-suited to this task due to its recursive structure and relatively low computational cost, typically scaling with the cube of the state dimension. This efficiency makes EKF a practical choice for real-time applications on smartphones, where processor resources and power consumption are constrained. In contrast, the UKF requires propagation of multiple sigma points, resulting in processing loads several times higher than EKF, while the PF demands even greater computational resources, as they rely on a large number of particles to represent state distributions. As a result, UKF and PF, although better at capturing nonlinearities, are rarely adopted in smartphone-based implementations [71, 99, 83].

Two main KF structures are implemented for smartphone-based inertial+GNSS fusion architectures, that is, the full-state Kalman filter (FSKF) and the error-state Kalman filter (ESKF). In inertial+GNSS fusion systems, the size of the Kalman filter’s state vector is determined by the required positioning accuracy, the quality of the inertial sensors, and the specific variables the system intends to estimate. As

clarified in [142], different systems choose the size and content of the state vector based on the trade-off between accuracy and computational load. Moreover, Extra elements such as sensor bias or lever-arm corrections are also added when higher precision is required. At the fundamental level, an LC-based fusion system may use a minimal state vector consisting of only position and velocity, resulting in a total of six states. These include three position components (x, y, z) and three velocity components (v_x, v_y, v_z) . However, this minimal configuration is generally insufficient for inertial systems, particularly those using low-cost MEMS sensors in smartphones, due to unmodeled drift and bias errors.

To address this, most practical systems extend the state vector to include additional variables such as device orientation (represented via Euler angles or quaternions) and biases of the accelerometer and gyroscope. This extended formulation improves robustness and accuracy, especially over longer durations or during GNSS signal outages.

The implementation presented in [76] uses a 15-dimensional full-state Kalman filter. This state vector is defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} & \mathbf{v} & \boldsymbol{\phi} & \mathbf{b}_a & \mathbf{b}_g \end{bmatrix}^T \quad (2.28)$$

where:

- $\mathbf{p} = [x, y, z]$ is the 3D position,
- $\mathbf{v} = [v_x, v_y, v_z]$ is the 3D velocity,
- $\boldsymbol{\phi} = [\phi_x, \phi_y, \phi_z]$ represents the orientation in roll, pitch, and yaw,
- $\mathbf{b}_a = [b_{a_x}, b_{a_y}, b_{a_z}]$ are the accelerometer biases, and
- $\mathbf{b}_g = [b_{g_x}, b_{g_y}, b_{g_z}]$ are the gyroscope biases.

This gives a total of 15 states: 3 for position, 3 for velocity, 3 for orientation, 3 for accelerometer biases, and 3 for gyroscope biases. The 15-state KF has become a widely used configuration for applications using MEMS-grade inertial sensors. This full-state formulation allows for direct estimation of key navigation parameters. In [76], this structure was successfully implemented and tested in a TC GNSS/INS system running on Android smartphones. The results in [76] confirm that this structure supports lane-level accuracy, though it requires careful calibration of the low-cost MEMS sensors and consistent GNSS updates to maintain reliability.

In contrast to full-state Kalman filters, which estimate the actual position, velocity, and orientation directly, the ESKF focuses on estimating small deviations from a separately propagated nominal state. The nominal state represents the system's predicted trajectory, computed using strapdown integration of raw IMU data, and serves as a reference for error estimation [143].

Rather than continuously re-estimating the full navigation state, the ESKF tracks how far the current prediction deviates from the nominal state. This approach improves numerical stability and is particularly robust when applied to low-cost, noisy IMUs such as those embedded in smartphones.

The error-state vector used in [72, 79] is defined as:

$$\delta \mathbf{x} = \begin{bmatrix} \delta \mathbf{p} & \delta \mathbf{v} & \delta \boldsymbol{\theta} & \delta \mathbf{b}_a & \delta \mathbf{b}_g \end{bmatrix}^T \quad (2.29)$$

where $\delta \mathbf{p}$ represents the position error, $\delta \mathbf{v}$ is the velocity error, $\delta \boldsymbol{\theta}$ is the orientation error (typically expressed as small-angle approximations), and $\delta \mathbf{b}_a$ and $\delta \mathbf{b}_g$ are the estimated biases of the accelerometer and gyroscope, respectively.

The primary advantage of the ESKF structure is that it simplifies orientation updates and improves long-term filter stability.

As demonstrated in [72], the ESKF was deployed in a real-time smartphone-based vehicular navigation system, maintaining stable positioning performance in challenging driving environments. Similarly, [79] presents a robust ESKF-based fusion framework specifically optimized for MEMS sensors, emphasizing low-latency correction and effective bias estimation.

Together, these implementations demonstrate that while full-state filters can offer direct and interpretable state estimation, the error-state formulation is better aligned with the limitations of smartphone-grade sensors and computational constraints. The ESKF also facilitates more accurate and stable integration in practical scenarios where GNSS data may be intermittent or delayed.

Synchronizing IMU and GNSS data is a key challenge in INS+GNSS fusion architectures. IMUs typically sample at high rates (up to 200 Hz), while GNSS updates are much slower, usually between 1 and 10 Hz. To keep the fusion accurate, the data from both sensors must be properly aligned in time. Most systems handle this by using interpolation or buffering techniques to synchronize the data streams and apply GNSS corrections at the correct moment [144]. These methods ensure that GNSS measurements are aligned to the correct IMU timestamp before being passed to the fusion process. Once synchronized, the KF uses high-rate IMU data for continuous state prediction and integrates GNSS updates as they become available [98, 78, 83]. This approach maintains reliable estimates even with delayed or infrequent GNSS corrections, but only if the input data streams are properly time-aligned beforehand.

Smartphone-based INS+GNSS fusion systems must operate under several practical constraints, including limited processing power, battery life, and variable sensor orientation due to the way smartphones are carried or used. To manage these limitations, researchers often simplify the motion and sensor models to make real-time processing feasible. For instance, some implementations assume a constant velocity/acceleration model between GNSS updates, which reduces the computational complexity of the fusion method. Similarly, when it comes to estimating orientation, simpler systems may assume the smartphone is held in a fixed position. In contrast, more advanced approaches employ quaternion-based orientation estimation to account for arbitrary rotations of the device during use [81, 78].

Further, recent developments have introduced hybrid approaches that combine traditional filtering techniques with data-driven models, such as artificial neural networks (ANNs). [78] presents a KF enhanced with machine learning, where the network learns to improve positioning accuracy in challenging urban environments affected by multipath and signal occlusion. These methods show potential for

improving robustness, particularly in areas where GNSS signals are unreliable. However, they are still in the early stages of research and development, and questions remain regarding their generalizability, computational demands, and long-term stability across diverse conditions.

The summarized outcomes of various smartphone-based INS+GNSS fusion architectures are presented in Table 2.11, highlighting the diversity of integration strategies, sensor models, and fusion methods used across different implementations. In summary, the design choices in INS+GNSS smartphone fusion systems reflect a layered set of trade-offs between GNSS processing precision, filter complexity, real-time feasibility, and hardware variability. SPP-based fusions dominate for their ease of implementation, while PPP and RTK architectures offer greater precision for specialized applications. KF especially EKF, remain the core fusion mechanism due to their flexibility and efficiency. Though more advanced techniques like UKF, PF, FGO, or ANN-based models offer incremental benefits, their adoption is limited by computational cost, real-time constraints, and implementation complexity on smartphones. These constraints and gaps in performance set the stage for the next line of architectures such as PDR+GNSS fusion which aim to overcome many of the limitations inherent in INS-based systems by leveraging human motion constraints for improved drift control and localization reliability.

Table 2.11: Summary of INS+GNSS Fusion Architectures

Fusion Architecture	Accuracy Range (m)	Best Reported Accuracy	Re- Fusion Type(s)	Fusion Algorithm(s) Used	References
INS+GNSS-SPP	4.2 – 29 m	0.8 m (motorway)	LC, UC	KF, EKF, UKF	[72, 81], [78, 75, 79, 76, 82, 99, 83], [73, 98]
INS+GNSS-PPP	1.9 – 11.9 m	1.2 m	TC	EKF	[80, 77], [81]
INS+GNSS-RTK	0.24 – 3.23 m	0.24 m	TC	EKF	[74]

2.8.3 PDR+GNSS Fusion Architecture

PDR+GNSS fusion architectures are popular for pedestrian mobility because they utilize human walking behavior to estimate the pedestrian’s position more accurately, especially in environments where GNSS signals are distorted or unavailable. Unlike INS, which integrates acceleration to estimate position and suffers from rapid drift, PDR relies on step detection, heading estimation, and step length modeling, resulting in slower error accumulation. This makes PDR a suitable counterpart to GNSS, particularly when the application domain is constrained to human movement, as in indoor navigation, urban pedestrian tracking, or first responder localization [24, 25, 85, 93].

While PDR is a useful complement to GNSS in pedestrian navigation, its performance relies heavily on accurate step detection and heading estimation. A key limitation is the accumulation of heading errors over time, which can significantly

degrade accuracy. PDR is also highly sensitive to rapid changes in the way the smartphone is carried, such as shaking, swinging, rotation, or repositioning of the device (for example, moving it from hand to pocket), which introduces unpredictable noise into the motion model. Such disturbances can cause both sudden, large deviations in position estimates and gradual, persistent drifts, often influenced by natural variability between users. This topic is discussed in more detail in Chapter 3. These factors make PDR performance inconsistent across individuals and highlight the need for robust, adaptive fusion methods. Fusion with GNSS provides the necessary corrections to these heading and position drifts, enabling better localization. Many PDR+GNSS architectures perform data fusion at each detected step rather than at regular time intervals. This event-driven approach allows the system to adapt dynamically to the user’s movement, improving responsiveness and alignment with real pedestrian motion [24, 87, 88, 95, 90].

Like INS+SPP, PDR+SPP is the most commonly implemented architecture due to its lightweight design and compatibility with standard GNSS APIs. This integration is appropriate for general-purpose pedestrian navigation, where real-time feedback is more critical than high precision. The simplicity of SPP allows it to be fused with PDR using either basic correction mechanisms or probabilistic frameworks, enabling widespread deployment on commercial smartphones [87, 93, 94, 96, 88, 95, 90, 27, 28, 92, 89, 24, 25, 26, 85, 86, 84]. By contrast, PDR+PPP has received limited attention in the literature. While a few studies have explored PDR+PPP configurations, PDR+RTK integration remains largely unexamined, likely due to its reliance on real-time correction infrastructure and increased system complexity (see Table 2.13). These architectures target applications requiring high accuracy and robustness in GNSS-degraded environments, such as urban canyons and disaster zones.

Among the various fusion methods, EKF and FGO are the main methods used in the integration process for this type of architecture. In particular, FGO has emerged as a leading and widely used method in the literature. This growing trend is driven by FGO’s ability to model the fusion process as a global optimization problem, where various states and sensor constraints are jointly optimized over a graphical structure. Unlike KF, which operates sequentially and estimates the next state based on the current estimate and recent measurements under predefined noise models, FGO performs iterative, nonlinear optimization by incrementally incorporating new measurements into a factor graph. Instead of relying solely on the immediate past state, FGO estimates the current state by optimizing the entire graph. This makes FGO a more suitable option for various applications that require high accuracy. Although it has been applied in several domains, the literature has predominantly explored its use in pedestrian navigation systems, where challenges such as sudden motion changes, transient variations in smartphone carrying modes, sharp turns, diverse movement patterns, and various GNSS-related errors are common. These issues have been addressed in several studies [24, 25, 26, 85, 86] have demonstrated that FGO consistently improves trajectory stability and positioning accuracy in urban and obstructed environments.

Loosely coupled integration is commonly preferred for fusing GNSS and PDR. The fusion process begins by constructing a nonlinear factor graph, commonly using

the GTSAM library, where GNSS observations are introduced as unary prior factors connected to individual pose nodes. At the same time, PDR updates are modeled as binary odometry factors linking consecutive poses due to the nature of dead reckoning as done in this study [24]. This architectural design allows GNSS and PDR data to be processed independently before fusion, which makes it easier to handle real-world conditions such as irregular GNSS availability or variable walking patterns. Most existing architectures implement a 2D representation, estimating only horizontal positions (x, y), as vertical measurements from smartphone-grade sensors are generally considered too noisy and unreliable to be used effectively in the optimization process. The GTSAM offers the flexibility to apply incremental or batch solvers. Gaussian noise models with diagonal covariance matrices are used to model both GNSS and PDR uncertainties, with parameters fixed and can be changed manually.

Several studies [24, 85, 25] implemented FGO using a batch optimization strategy, which, while effective in terms of accuracy, significantly increases computational load and is not well-suited for smartphone-based applications. To overcome this limitation, a sliding window-based FGO approach is proposed by [145], aiming to limit graph size while preserving essential historical information (see chapter 4). This design offers a practical balance between computational efficiency and positioning accuracy, making it more suitable for real-time pedestrian navigation on resource-constrained mobile devices. This architectural design has proven effective in the literature under different GNSS and PDR errors, such as intermittent GNSS availability, variations in how the smartphone is carried, and differences in user motion patterns.

While FGO provides high positioning accuracy through global optimization, its computational demands make it less suitable for real-time smartphone-based applications. As a result, EKF remains the more practical choice in the literature for PDR+GNSS fusion, particularly in systems requiring step-wise, low-latency updates. In [145], sliding window strategies for FGO were introduced for the first time in smartphone-based PDR+GNSS architectures, with a minimum window size of one, effectively resembling EKF behavior. Although this study did not directly compare FGO with EKF, further evidence from this study [127] shows that even with a one-second window, FGO maintained higher accuracy than EKF while incurring lower computational cost. These findings highlight that while FGO can be made more efficient through architectural adaptation, EKF continues to be favored in mobile pedestrian systems due to its real-time feasibility and compatibility with incremental step-based updates [95, 90, 28, 92].

The modeling of sensor error and user dynamics is important to PDR+GNSS fusion. Many systems account for heading drift, variable step length, and orientation changes as part of the state estimation process, recognizing that PDR performance is heavily influenced by how the smartphone is carried and by individual factors such as walking speed, body height, and movement patterns. Some studies [85, 86] incorporate motion-related cues such as step events and velocity updates to improve PDR robustness, particularly in dense urban environments.

PDR estimates are typically generated at each step, even irregularly and user-dependent, while GNSS updates arrive at fixed intervals at 1 Hz. To address this mismatch, most implementations in the literature align the GNSS position

fix to the nearest PDR step or interpolate positions when necessary. This step-based synchronization ensures that each fusion epoch corresponds to a physically meaningful pedestrian motion update, improving the consistency of the integrated trajectory. In FGO-based systems, this alignment is handled naturally through the graph structure, where GNSS factors are attached only to nodes corresponding to available timestamps. This design avoids forcing synchronization at uniform time intervals and instead fuses data when it is most reliable and contextually relevant, improving both accuracy and computational efficiency in real-world deployments.

From a system design perspective, PDR+GNSS fusion architectures are generally more computationally efficient than INS+GNSS approaches because PDR avoids continuous double integration of noisy inertial signals, which is both computation heavy and drift prone. However, this efficiency comes with trade-offs that is PDR systems must handle human-induced variability such as inconsistent step lengths, stop-and-go motion, and heading errors influenced by phone orientation or environmental factors. To address these challenges, some systems use PDR not only for motion estimation but also for integrity monitoring [84]. [84] used PDR derived step length to probabilistically validate GNSS updates, excluding faulty position fixes in real time and thereby improving robustness in degraded signal conditions.

In conclusion, PDR+GNSS fusion architectures are used for pedestrian navigation and exploit human motion patterns to enhance localization stability where GNSS alone fails. The dominance of SPP-based fusion reflects the practical constraints of smartphones, while PPP and RTK configurations demonstrate the feasibility of achieving higher accuracy with more infrastructure or post-processing support. EKF remains the standard filtering method due to its simplicity and effectiveness in step-wise fusion, but FGO is gaining ground for its ability to perform trajectory smoothing and handle non-Gaussian uncertainty. Unlike INS+GNSS systems, PDR+GNSS fusion requires careful handling of user behavior, device placement, and motion variability. These systems achieve robustness not by heavy computation, but by smartly modeling how people move and using GNSS corrections only when needed an approach that suits the practical limits of smartphone-based positioning.

The comparative analysis of reported PDR+GNSS fusion systems, including fusion types, algorithms, and positional accuracy, is presented in Table 2.12.

Table 2.12: Summary of PDR+GNSS Fusion Architectures

Fusion Architecture	Accuracy Range (m)	Accuracy		Fusion Type	Algorithms Used	References
PDR+GNSS-SPP	1.45 – 2.26 m	1.45 (FGO)	m	LC, TC	FGO, EKF, KF	[24], [25, 85, 86], [89], [26]
PDR+GNSS-PPP	< -	< -	-	-	-	-
PDR+GNSS-RTK	-	-	-	-	-	-

2.8.4 Data-Driven Inertial+GNSS Fusion Architectures

The emergence of data-driven techniques in pedestrian navigation has been largely driven by the limitations of classical model-based PDR and INS approaches. Traditional PDR systems depend heavily on manually crafted models for step detection, stride length estimation, and heading updates, often tailored to fixed walking conditions or device orientations. Similarly, classical inertial navigation systems rely on linear motion assumptions and struggle with cumulative drift due to sensor biases and noise. These handcrafted methods, while computationally efficient, are not sufficiently robust for the variability encountered in real-world smartphone usage where users may walk irregularly, change device placement, or move through complex environments such as staircases, elevators, and crowded urban areas. Moreover, sensor characteristics vary significantly across commercial devices, reducing the generalizability of fixed-parameter models.

In contrast, data-driven approaches offer a compelling alternative by learning motion patterns directly from raw sensor data. Techniques involving convolutional and recurrent neural networks have demonstrated improved performance in step estimation, trajectory prediction, and inertial signal correction. These models inherently adapt to user behavior and environmental changes without the need for extensive manual calibration. As a result, research on data-driven methods for standalone PDR or INS has rapidly expanded, showing measurable improvements over classical baselines. However, despite this growing interest, the application of data-driven techniques in the context of PDR/INS fusion with GNSS remains extremely limited. While machine learning has been successfully applied to refine PDR or INS independently, the number of studies that implement a fully data-driven fusion architecture where the learning model integrates both inertial and GNSS measurements is minimal. A thorough review of the literature revealed no existing studies that apply machine learning specifically for smartphone-based inertial+GNSS fusion.

2.8.5 Overall Analysis of Inertial+GNSS Fusion Architectures

The literature reveals three main categories of inertial+GNSS fusion architectures in smartphone localization: INS+GNSS, PDR+GNSS, and data-driven PDR/INS methods each reflecting distinct trade-offs in design complexity, accuracy, and practical deployment.

INS+GNSS Architectures: Among the reviewed approaches, INS+GNSS-SPP-LC was the most commonly studied, offering a practical trade-off between signal availability and reasonable accuracy. These systems mainly used KF and its nonlinear extensions because of their lower computational demands. INS+GNSS-RTK configurations showed better positioning precision but required external corrections and supporting infrastructure. PPP-based methods appeared less frequently and showed more variation in performance.

PDR+GNSS Architectures: PDR-based fusion is well-suited for pedestrian use, especially in indoor or GNSS-challenged environments. Most studies combined PDR with GNSS-SPP using FGO or EKF in loosely coupled setups, making PDR+GNSS-

SPP-FGO/EKF-LC a popular choice, balancing simplicity and real-time feasibility. Although less accurate than RTK-based systems, these architectures are efficient and practical for handheld applications. In contrast, PDR+GNSS-RTK and PDR+GNSS-PPP are rarely explored due to the challenges of real-time correction.

Data-Driven PDR/INS+GNSS Architectures: While data-driven methods, particularly machine learning, are increasingly applied to standalone PDR or INS, their integration with GNSS remains rare. A thorough review of the literature revealed no existing studies that apply machine learning specifically for smartphone-based inertial+GNSS fusion. This highlights a clear research gap and underscores the potential for future work aimed at developing more adaptive, personalized, and robust fusion architectures, especially for challenging GNSS limited environments.

The Table 2.13 shows the overall analysis of inertial+GNSS fusion architectures covered in the literature.

Table 2.13: Inertial+GNSS Fusion Architectures covered in Literature

Fusion Algorithms	INS			PDR			Data-Driven INS		
	SPP	PPP	RTK	SPP	PPP	RTK	SPP	PPP	RTK
Kalman Filter (KF)	[78, 72, 79, 76, 99, 83]	X	X	[87, 93, 94, 96, 88]	X	X	X	X	X
Extended Kalman Filter (EKF)	[75, 73, 98]	[80, 77, 81]	[74]	[95, 90, 27, 28, 92]	X	X	X	X	X
Adaptive EKF (AEKF)	X	X	X	[89]	X	X	X	X	X
Unscented Kalman Filter (UKF)	[71, 82]	X	X	X	X	X	X	X	X
Factor Graph Optimization (FGO)	X	X	X	[24, 25, 26, 85, 86]	X	X	X	X	X
Particle Filter	X	X	X	[84]	X	X	X	X	X

The analysis highlights three main trends:

- **Accuracy Complexity Tradeoff:** Higher positioning accuracy is generally associated with tightly coupled RTK systems and advanced filtering methods, but these require greater computation and infrastructure.
- **Context-Driven Design:** PDR-based systems dominate pedestrian use-cases, while INS architectures are preferred in vehicles and unconstrained navigation.
- **Gap in Data-Driven Fusion:** While ML/AI enhances individual subsystems, integrated learning-based fusion remains underexplored, representing a valuable direction for future research.

The selection of an inertial+GNSS fusion architecture depends on application-specific needs, such as required positioning accuracy, computational constraints, and environmental conditions. Based on the literature, the three primary fusion categories, INS+GNSS, PDR+GNSS, and data-driven approaches, exhibit varying trade-offs across these dimensions. A summary of the average positioning accuracy

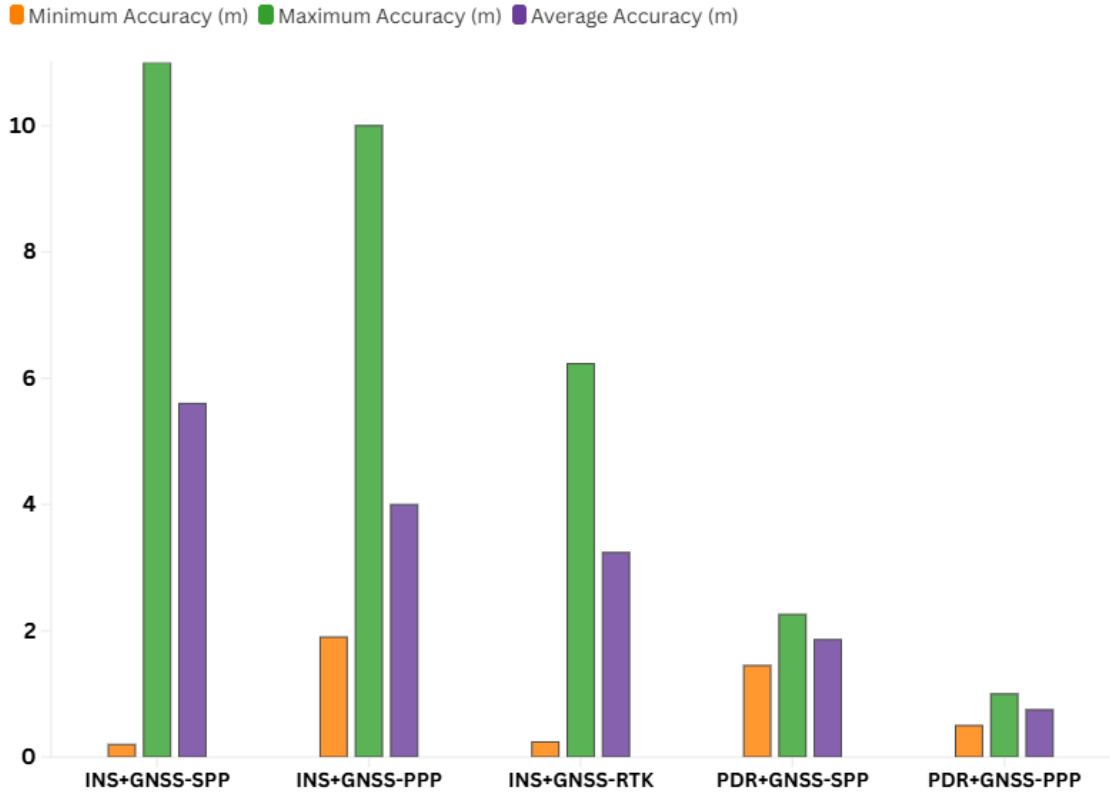


Figure 2.25: positional accuracy ranges across fusion architectures.

reported for each architecture type is presented in Fig. 2.25, providing a comparative overview to guide design choices.

Fig. 2.25 summarizes the positional accuracy reported across different fusion architectures by aggregating values from existing literature. Here, minimum accuracy (orange) refers to the highest observed positional error that is worst-case performance while maximum accuracy (green) represents the lowest positional error, indicating best case performance. The average accuracy (purple) denotes the typical accuracy reported across studies. These values are based on reported positioning errors (in meters) from literature. This figure provides a comparative overview to illustrate the performance variability and trade-offs among different INS+GNSS and PDR+GNSS integration strategies.

2.9 Conclusion

The integration of inertial and GNSS technologies plays an important role in smartphone based localization, with different fusion architectures designed to meet various needs for accuracy, efficiency, and application context. A review of the literature shows common patterns and key insights that go beyond individual studies.

The review of existing literature on inertial and GNSS fusion has demonstrated how architecture selection is not arbitrary but is shaped by a combination of environmental constraints, target applications, required accuracy, available infrastructure, and system limitations such as processing power and energy budget. Each fusion architecture reflects an effort to balance these competing demands.

INS+GNSS fusion remains the most common approach for vehicular and infrastructure based use cases, particularly where smoother motion dynamics allow more accurate inertial propagation. Within this space, SPP is widely used due to its compatibility with off-the-shelf smartphone GNSS chips, though its accuracy is limited. PPP and RTK offer significantly improved precision, but their dependency on correction services, higher initialization time, and greater computational demand limit their adoption in real-time mobile applications. Consequently, loosely coupled architectures dominate, particularly for INS+GNSS-SPP, where simple Kalman-based filters are sufficient and easy to implement on mobile devices. However, this restricts robustness in degraded environments and limits the achievable accuracy.

In contrast, PDR+GNSS fusion architectures are applied to pedestrian applications where gait, step detection, and heading estimation offer a self-contained motion model. In this domain, FGO has emerged as a preferred fusion method due to its capability to handle asynchronous updates, delayed GNSS corrections, and sparse measurements. Most of the FGO implementations in this context focus on 2D trajectory estimation. While this is reasonable for many outdoor navigation tasks, it overlooks the growing demand for 3D localization in urban multi-level settings. In domains outside smartphones, including robotics and autonomous systems, FGO has been successfully extended to full 3D estimation. This suggests a promising, yet underexplored, opportunity to extend FGO-based PDR+GNSS systems to support full 3D positioning on smartphones.

Most existing smartphone-based FGO solutions use GTSAM, which, while popular, has notable limitations in real-time smartphone deployments. The library does not support a complete, native sliding window implementation out of the box, requiring additional customization to support fixed-lag smoothing efficiently. Moreover, some key components required for robust optimization such as marginalization, window maintenance, and real-time pose pruning require significant effort to implement manually. Alternatives such as Ceres Solver offer more flexible optimization strategies and better native support for sliding window techniques, including support for iSAM2-style incremental smoothing, but they are rarely used in the context of smartphone localization. Very few studies investigate these options, and fewer still compare their computational footprint, convergence behavior, or estimation accuracy under realistic smartphone conditions.

In terms of comparative analysis, FGO has not been consistently compared against other smoothing methods. While KF and EKF are the most common baselines, their smoothing variants, such as RTS or fixed interval EKF smoothers, offer a more appropriate basis for comparison. These smoothing filters can also be adapted for hybrid fusion, providing a better balance between computational cost and accuracy. Currently, this class of algorithms is rarely evaluated alongside FGO in a fair, unified framework. As a result, the understanding of the real trade-offs between filtering and smoothing approaches remains limited, especially for smartphone-based systems.

Architecture selection is strongly influenced by the application domain. In automotive scenarios, tightly coupled INS+RTK systems are desirable for achieving centimeter-level accuracy. In contrast, low-power fitness trackers or pedestrian mobility systems prioritize energy efficiency and drift correction over high absolute

accuracy, making loosely coupled PDR+SPP architectures more suitable. PPP is attractive for infrastructure-free, high-accuracy use cases like urban AR or drone navigation, but its slow convergence and satellite dependency pose significant implementation hurdles. Thus, each integration is a product of a trade-off, dictated by the interplay of hardware access, algorithmic complexity, application requirements, and environmental constraints.

Another recurring pattern is the lack of diversity in fusion algorithms. While KF and EKF remain dominant in INS+GNSS-SPP systems, their limitations in handling strong non-linearities, ambiguous measurement models, or missing data are well known. The use of PF and UKF remains marginal, and adaptive filters like AEKF are largely absent from smartphone platforms, despite their proven ability to adjust to changing noise characteristics and dynamic system behavior. Similarly, deep learning and hybrid ML-physics models, which have gained traction in robotics, are mostly absent from inertial+GNSS fusion in smartphones. These approaches could offer new ways to learn sensor error models, detect motion anomalies, or adaptively weigh sensor inputs, particularly under harsh conditions such as GNSS outages, indoor transitions, or erratic user motion.

Many systems continue to evaluate performance under idealized conditions uniform user motion, strong GNSS visibility, and consistent device orientation. Very few consider dynamic smartphone placement, multipath interference, varying pedestrian gait, or motion transitions from walking to running or climbing. Vertical accuracy, essential for multi-level buildings and indoor navigation, is rarely analyzed, despite being critical for next-generation applications like indoor mapping, emergency response, and AR guidance. Energy efficiency, too, remains underreported, though it plays a decisive role in real-world deployment viability.

The taxonomy presented in this work shows that while INS has been fused with SPP, PPP, and RTK using a variety of filters, many combinations remain unexplored, particularly in the context of smartphones. For example, INS+GNSS-SPP is consistently implemented with KF-based methods, while fusion with AEKF, PF, or FGO is absent. The same is true for INS+GNSS-PPP and RTK, where EKF dominates. Yet each fusion algorithm brings different strengths in terms of adaptability, resilience to non-linearities, and computational complexity, all of which could be leveraged in smartphone applications. The same observation applies to PDR-based systems. While PDR has been successfully fused with GNSS-SPP using FGO, fusion with PPP and RTK is extremely rare, despite the increasing availability of smartphones with dual-frequency GNSS support.

There is also very limited exploration of hybrid systems that combine INS and PDR with GNSS, even though combining these approaches could offer complementary advantages. INS for high-speed motion, PDR for pedestrian tracking, and GNSS for absolute correction. Hybrid filter architectures, where multiple motion models are fused through multi-rate or hierarchical filters, represent another underdeveloped research direction.

Although data-driven approaches are included as a conceptual fusion category due to their growing relevance in PDR and inertial navigation research, the literature review revealed very limited work on data-driven INS+GNSS fusion using smartphone data; therefore, such approaches do not yet constitute a mature or influ-

ential direction in current smartphone-based fusion systems. Overall, this review highlights a clear opportunity to move beyond current implementation patterns and explore the full space of fusion architectures and algorithms. Future research should focus on algorithmic diversity, better adaptation to smartphone-specific conditions, and rigorous comparative studies that evaluate accuracy, energy efficiency, and robustness across environments. Moving forward, the fusion landscape must evolve from fixed, handcrafted pipelines to more adaptive, modular systems capable of adjusting to user context, motion state, and environmental uncertainty. Only then can smartphone-based localization systems reach the level of accuracy, robustness, and scalability demanded by emerging applications in navigation, mobility, and augmented reality.

Chapter 3

Case Study I: Pedestrian Motions Transient Variations

3.1 Introduction

This chapter presents the main findings from two peer-reviewed publications that examine the comparative effectiveness of KF and FGO for smartphone-based pedestrian navigation in real-world scenarios. KF offers efficient, real-time recursive updates but is limited by its reliance on only the most recent measurement and previous state for estimating the next state. In contrast, FGO incorporates all available past measurements into a global optimization problem, enabling better error correction and trajectory consistency over time.

The first publication evaluates KF and FGO for walking and running motions in partial urban environments, with a primary focus on handling motion-induced PDR errors rather than GNSS inaccuracies. The second publication extends the investigation to more challenging conditions, introducing controlled pedestrian-induced errors to assess the robustness of KF, its smoothed variant (SKF), and FGO. Together, these studies provide a comprehensive assessment of FGO's advantages over traditional KF-based methods for practical pedestrian tracking applications.

3.2 Publication P1: KF vs. FGO under Walking and Running Conditions

3.2.1 Overview

The first study compared FGO and KF in partial urban environments where GNSS signal quality was degraded by surrounding buildings. PDR, while effective for certain motion types, struggles to adapt to all human motion patterns. Walking generally exhibits stable and predictable dynamics, making it easier to model, whereas running introduces rapid and complex movements that lead to significant drift.

3.2.2 Methodology

A dual-frequency smartphone (Huawei P9) collected GNSS and inertial sensor data, with high-precision RTK GNSS from a Ublox ZED-F9P receiver providing ground truth. Both KF and FGO were implemented with identical input parameters and noise models to ensure fair comparison.

3.2.3 Results

FGO performed better than KF in both walking and running. Running, despite introducing larger drift errors, still showed improved accuracy under FGO due to its use of historical data for drift mitigation.

Key Findings

- Walking: FGO reduced mean horizontal error by 30.6% and median error by 24.7% compared to KF.
- Running: FGO reduced mean error by 10% and median error by 6.15% compared to KF.
- KF was more responsive to rapid movement changes but suffered from drift accumulation.
- FGO's historical optimization better handled transient GNSS degradation and PDR drift.

3.3 Publication P2: KF, SKF, and FGO under Realistic PDR Error Conditions

3.3.1 Overview

The second study expanded the evaluation by comparing KF, SKF (KF with Rauch–Tung–Striebel (RTS) smoothing), and FGO under realistic pedestrian motion disturbances. In addition to transient changes in smartphone carrying modes, such as shaking, swinging, and rotating. This study also broadened the evaluation to include more pedestrians, capturing variations in gender, height, walking speed, and gait patterns.

3.3.2 Methodology

Ten pedestrians performed both walking and running, introducing two categories of PDR errors:

- **Short and High Errors (SHE):** Brief but substantial disruptions, such as sudden shakes or rotations.
- **Continuous and Low Errors (CLE):** Gradual variations from natural differences in gait and user characteristics.

Smartphone data were fused with GNSS measurements using KF, SKF, and FGO.

3.3.3 Results

FGO performed better. Here are the main findings of this study.

Key Findings

- FGO reduced mean horizontal positional errors by $\approx 25\%$ compared to KF and 24% compared to SKF.
- Median errors were reduced by 30% vs. KF and 32% vs. SKF.
- FGO converged faster after high-intensity disruptions.
- FGO demonstrated better resilience to diverse pedestrian-induced errors and carrying-mode disturbances.

3.4 Summary

Collectively, these publications affirm the considerable potential of FGO for enhancing the accuracy and reliability of pedestrian localization systems in challenging and realistic usage environments. The results establish FGO as a promising and robust alternative to KF-based methods for smartphone-based pedestrian navigation.

Publication P1

Published in: International Conference on Indoor
Positioning and Indoor Navigation (IPIN)

Comparison of FGO and KF for PDR-GNSS Fusion on Smartphone for Diverse Pedestrian Movements

Amjad Hussain and Luis Enrique Díez

Faculty of Engineering

University of Deusto

Avda. Universidades 24, 48007 Bilbao, Spain

{amjad.hussain, luis.enrique.diez}@deusto.es

Abstract—Smartphone-based positioning has gained significant popularity due to its widespread availability. However, its reliance on GNSS as the primary positioning component poses limitations in challenging environments such as urban canyons, tunnels, and indoor areas, where GNSS signals are distorted or missed due to multi-path interference and non-line-of-sight (NLOS). To address this, sensor fusion techniques have been employed, with the integration of MEMS IMU being a common approach due to its complementary features with GNSS. For pedestrian localization, the fusion of PDR and GNSS is a classic architecture that commonly involves the use of the Kalman Filter (KF). Recently, researchers have started to consider the Factor Graph Optimization (FGO) algorithm for the fusion, due to its better accuracy. However, it is still necessary to assess its performance compared to the KF under different situations, such as various pedestrian movements. In this study, experimental evaluations during walking and running activities demonstrate that FGO consistently outperforms KF, resulting in a significant reduction of mean horizontal errors by 30.6% and 10% in walking and running, respectively. These findings highlight that between walking and running modes, the accuracy of FGO is influenced by the magnitude of PDR errors and the level of correlation in GNSS measurements, resulting in varying performance. Overall, FGO demonstrated superior efficacy compared to KF in enhancing the accuracy and reliability of smartphone-based positioning systems, especially in challenging pedestrian scenarios.

Index Terms—Factor Graph Optimization, PDR, GNSS, Kalman Filter, Smartphone

I. INTRODUCTION

Smartphone-based positioning has gained increasing popularity in the realm of pedestrian positioning and navigation, owing to its mass availability and easy accessibility to individuals [1]. In contemporary mobile technology, the Global Navigation Satellite System (GNSS) chip integrated into smartphones constitutes a pivotal component for location-based services (LBS). Nonetheless, the effectiveness of the GNSS chip is dependent on the availability of an unobstructed path to satellites for receiving location information. However, its accuracy suffers greatly in indoor areas, urban canyons, and tunnels where satellite signals are distorted, diverted, weak, and obstructed, resulting in either no data or noisy data with outliers at the GNSS chip size receiver [2].

To enhance the positioning system, researchers have implemented the concept of multi-sensor fusion to improve

smartphone-based positioning in such GNSS-denied scenarios. As smartphones come with multiple sensors, that can be utilized as an alternative solution to provide location parameters along with GNSS. These sensors include a camera, inertial sensors, light sensors, barometer, BLE, and WIFI interfaces, and even LiDAR. Among available sensors, the MEMS IMU has greater significance because of its complementing features to GNSS (low errors in the short term, high update rate) and also it does not require additional infrastructure in the environment [3].

The fusion of inertial sensors and GNSS in positioning depends on factors such as the type of inertial navigation system used, GNSS processing techniques, integration levels, and fusion algorithms. Smartphone MEMS IMUs frequently face limitations by their high noise output which precludes the design of Strapdown Inertial Navigation Systems (SINS) and their fusion with chip-scale GNSS. SINS accumulate error that is proportional to the elapsed time, while Pedestrian Dead-Reckoning (PDR) systems accumulate error based on the number of steps taken, as it relies on pedestrian walking gaits [4]. Therefore, in order to attain more precision, researchers have focused more on integrating PDR with GNSS in smartphones. MEMS IMU-based PDR depends on detection and length estimation of the number of steps, and heading estimation in order to estimate pedestrian movements [5], [6]. The accuracy of PDR is contingent upon various pedestrian movements making it challenging for PDR+GNSS fusion architecture. The Kalman Filter (KF) has traditionally been the main fusion algorithm for PDR+GNSS fusion architecture in smartphones [7]–[12]. However, achieving high accuracy using KF and its variant models has been challenging due to the linear first-order Markov dynamic model and Gaussian noise assumptions, and the fact of using only the latest available measurement.

Instead, the newly proposed Factor Graph Optimization (FGO) [13] is a well-liked approach for the multi-sensor fusion concept. Factor graphs are graphical models used to effectively address complex estimation problems like Simultaneous Localization and Mapping (SLAM) [14]. FGO utilizes iterative solutions and graphical models to optimize state estimates. It transforms state estimation into a nonlinear least squares problem by establishing a link between the state and collected data. FGO constructs a cost function using factor

This work has been supported by the Research Training Grants Programme of the University of Deusto.

constraints and employs iterative solvers like Gauss-Newton or Levenberg-Marquardt to minimize the cost function and obtain optimal estimations. Notably, FGO considers historical measurements and system updates to optimize the full state set, distinguishing it from KF-based integration. It uses the non-linear optimization technique to fuse several sensor outputs for better estimation. Consequently, there is a growing trend to utilize FGO in various GNSS+INS fusion scenarios [15]–[17]. Researchers [18], [19] have applied graph optimization techniques as fusion algorithms for PDR+GNSS fusion, such as FGO, and claimed greater accuracy for pedestrian walking movement than the conventional KF.

Based on our current understanding, no previous study has investigated the performance of FGO in various pedestrian movements. Therefore, we utilized FGO to assess position errors in different pedestrian movements (walking and running modes) for PDR+GNSS fusion in smartphones and evaluate its performance in both modes. A comparative analysis was carried out to assess the effectiveness of the FGO-based PDR+GNSS fusion algorithm in contrast to the conventional KF-based PDR+GNSS algorithm under diverse pedestrian mobility scenarios.

The rest of the paper explains the methodology, including the PDR mechanism, KF, and FGO-based PDR+GNSS fusion architecture are described. In the experimental evaluation section, the experiments conducted for both walking and running modes are presented, accompanied by meticulous findings and in-depth analysis.

II. METHODOLOGY

This section of the paper outlines the methodology used in the study, which involved the development of PDR, KF-based PDR+GNSS fusion, and FGO-based PDR+GNSS fusion.

A. Development of PDR

PDR estimates pedestrian movements based on the detection of steps, step length, and heading estimation. The PDR working mechanism can be mathematically written as follows

$$\mathbf{Pos}_{k+1} = \mathbf{Pos}_k + \Delta\mathbf{Pos}_{k|k+1} \quad (1)$$

where $\mathbf{Pos}_{k+1}$ and $\mathbf{Pos}_k$ represent the position at steps $k+1$ and k respectively, and $\Delta\mathbf{Pos}_{k|k+1}$ represents the position increment from step k to $k+1$ that, expressed in ENU reference system, can be written as:

$$\Delta\mathbf{Pos}_{k|k+1} = \begin{bmatrix} \Delta\mathbf{Pos}_{k|k+1}^E \\ \Delta\mathbf{Pos}_{k|k+1}^N \end{bmatrix} = \begin{bmatrix} \mathbf{SL}_{k|k+1} \cdot \cos(\phi) \\ \mathbf{SL}_{k|k+1} \cdot \sin(\phi) \end{bmatrix} \quad (2)$$

where the heading angle is indicated by ϕ , the step length from step k to step $k+1$ is shown by $\mathbf{SL}_{k|k+1}$.

For step detection, we utilized the magnitude of the triaxial accelerometer (see eq.3) as an input signal. To detect steps, we applied a Butterworth filter with a 2 Hz cutoff frequency to analyze the peak of acc_{magk} . For walking, we set a lower acceleration threshold of 0.3 m/s^2 and a higher threshold of

1.5 m/s^2 . For running, the lower threshold was 1.0 m/s^2 and the higher threshold was 4.5 m/s^2 .

$$\text{Acc}_{magk} = \sqrt{\text{Acc}_{xk}^2 + \text{Acc}_{yk}^2 + \text{Acc}_{zk}^2} \quad (3)$$

For step length estimation, we used Weinberg's algorithm [20] that assumes that each step taken by a pedestrian corresponds to a local minimum in the acceleration signal. By detecting these local minima and measuring the time between them, the step length can be estimated.

$$\mathbf{SL}_{k|k+1} = C \cdot [\max(\text{Acc}_{magk}) - \min(\text{Acc}_{magk})]^{1/4} \quad (4)$$

in this $\mathbf{SL}_{k|k+1}$ represents the estimated step length between two consecutive acceleration peaks and C is a constant factor used to scale the estimated step length.

For step heading estimation, we employed the heading angle obtained from the attitude and heading reference system (AHRS) integrated into the smartphone at each detected step epoch.

B. KF Based PDR+GNSS Fusion Architecture in Smartphones

The KF operates through two steps: the prediction step and the correction step. In the prediction step the state of the system at the next time step is estimated by using a dynamic model and the current state of the system. The correction step, on the other hand, involves refining the estimated state by incorporating measurements from sensors. This allows for an update of the estimated state, resulting in a more accurate position estimation. The state vector, represented as $\mathbf{X}$, serves as the fundamental element for the state transition in PDR+GNSS integration which is the position errors of PDR in x and y coordinates as shown.

$$\mathbf{X} = \begin{pmatrix} \Delta\text{PDR}_x \\ \Delta\text{PDR}_y \end{pmatrix}$$

The formulation of this transformation is expressed as follows:

$$\mathbf{X}_{k+1} = \mathbf{F}_{k|k+1} \cdot \mathbf{X}_k + \mathbf{W}_{k+1} \quad (5)$$

where $\mathbf{X}_{k+1}$ and $\mathbf{X}_k$ indicate the state vector at $k+1$ and k epochs, respectively; $\mathbf{W}_{k+1}$ is process noise vector which is assumed to be subject to a standard Gaussian distribution, expressed as $\mathbf{W}_{k+1} \sim \mathcal{N}(0, Q_k)$, where Q_k represents the covariance matrix of the process noise; and $\mathbf{F}_{k|k+1}$ refers to the transformation matrix at epochs $k+1$ and k :

$$\mathbf{F}_{k|k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6)$$

The measurement model is

$$\mathbf{Z}_{k+1} = \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1} + \mathbf{V}_{k+1} \quad (7)$$

where $\mathbf{Z}_{k+1}$ represent the measurement vector; $\mathbf{V}_{k+1}$ is the measurement noise vector, which is also assumed to be subject to a Gaussian distribution, expressed as $\mathbf{V}_{k+1} \sim \mathcal{N}(0, R)$, where

R represents the covariance matrix of the measurement noise; and $\mathbf{H}_{k+1}$ represents the observation matrix:

$$\mathbf{H}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (8)$$

The actual observations will be the difference of PDR and GNSS positions:

$$\mathbf{Z}_{k+1} = \mathbf{Pos}_{k+1}^{GNSS} - \mathbf{Pos}_{k+1}^{PDR} \quad (9)$$

The prediction step of KF can be mathematically written as follow.

$$\hat{\mathbf{X}}_{k+1} = \mathbf{F}_{k|k+1} \cdot \mathbf{X}_k \quad (10)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_{k|k+1} \cdot \mathbf{P}_k \cdot (\mathbf{F}_{k|k+1})^T + \mathbf{Q}_k \quad (11)$$

Here, the state prediction $\hat{\mathbf{X}}_{k+1}$ is obtained using the transformation model. The matrix $\mathbf{P}_k$ represents the predicted state covariance matrix at epoch k , while $\mathbf{P}_{k+1}^-$ refers to the prediction made at epoch $k+1$. Then, the correction step of KF is updating the state estimation as follows:

$$\hat{\mathbf{X}}_{k+1} = \mathbf{X}_{k+1}^- + \mathbf{K}_{k+1} \cdot (\mathbf{Z}_{k+1} - \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1}^-) \quad (12)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \cdot (\mathbf{H}_{k+1})^T \cdot [\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1}^- \cdot (\mathbf{H}_{k+1})^T + \mathbf{R}_{k+1}]^{-1} \quad (13)$$

$$\mathbf{P}_{k+1} = (\mathbf{I}_{2 \times 2} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{P}_{k+1}^- \quad (14)$$

so where $\hat{\mathbf{X}}_{k+1}$ is the updated state estimation at $k+1$, $\mathbf{K}_{k+1}$ represents the Kalman gain matrix at $k+1$ and $\mathbf{I}_{2 \times 2}$ represents the identity matrix of size 2×2 .

C. FGO Based PDR+GNSS Fusion Architecture in Smart-phone

A factor graph is a probability model that utilizes a graph to represent the probability dependence of different variables. Specifically, a factor graph is a bipartite graph composed of two types of nodes: the unknown state variables and the factor nodes, which represent the conditional probabilities of some set of state variables. The undirected edges connect the state variable nodes and corresponding factor nodes. There is an edge between the two nodes only when the variable node and the factor node have a functional relationship. A factor graph allows to decompose a global probability function into the product of multiple local probability functions, reducing the complexity of solving the global function.

In sensor integration, we are usually interested in estimating a set of state variables $\mathbf{X}$ from a set of given measurements $\mathbf{Z}$. Assuming a Markov model, that conditional probability density function can be formulated:

$$P(\mathbf{X}|\mathbf{Z}) = \prod_{i=1}^k \frac{P(z_i|x_i)P(x_i|x_{i-1}, u_i)}{P(z_i)} P(x_0) \quad (15)$$

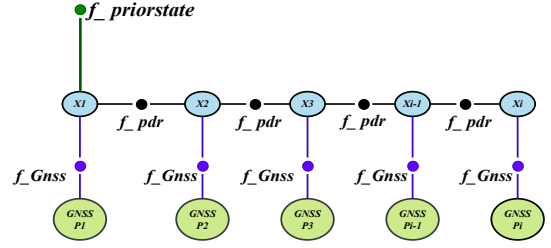


Fig. 1. Formation of PDR and GNSS factors within FGO mechanism

where z_i represents the measurements observed at epoch i (e.g. GNSS measurements), x_i represents the system state at i , and u_i denotes the control input (e.g., PDR measurements).

The most often used estimator for these unknown state variables $\mathbf{X}$ is the maximum a posteriori or MAP estimate:

$$\hat{\mathbf{X}} = \arg \max P(\mathbf{X}|\mathbf{Z}) = \arg \max \prod_{i=1}^k P(z_i|x_i)P(x_i|x_{i-1}, u_i) \quad (16)$$

In FGO-based integration, all these likelihood and transition probabilities are treated as the factorization of the global probability:

$$\hat{\mathbf{X}} = \arg \max \prod_{j=1}^n f(x_j) \quad (17)$$

Each factor models a constraint and must include a measure of uncertainty. The most common model is a Gaussian noise:

$$P(z_i|x_i) \propto \exp \left(-\frac{1}{2} \|h_i(x_i) - z_i\|^2 \nabla_i \right) \quad (18)$$

$$P(x_i|x_{i-1}) \propto \exp \left(-\frac{1}{2} \|\Phi_i(x_{i-1} - x_i)\|^2 \Omega_i \right) \quad (19)$$

where the function $\Phi_i(\cdot)$ describes the relationship between the preceding states x_{i-1} and x_i , the function $h_i(\cdot)$ represents the relationship between the state x_i and the measurement z_i , and the covariance matrices are denoted by ∇_i and Ω_i .

Taking the negative log and dropping the factor $1/2$ transforms the problem into the minimization of an error function, that is, a nonlinear least-squares problem:

$$\hat{\mathbf{X}} = \arg \min \left(\sum_{i=1}^k \|\Phi_i(x_{i-1}) - x_i\|_{\Omega_i}^2 + \sum_{i=1}^k \|h_i(x_i) - z_i\|_{\nabla_i}^2 \right) \quad (20)$$

The major difference in comparison to KF is that FGO includes all historical observation measurements. Recent studies demonstrate that FGO outperforms traditional KF, and researchers are actively exploring the benefits of FGO in different navigation systems [18], [19], [21]–[23].

In our case, as shown in Fig. 1, the factors represent the GNSS measurements and the transition between states due to the motion model and PDR measurements:

PDR Factor This factor represents the constraint imposed by the motion model derived from the PDR system:

$$\begin{aligned} \|e_{k+1}^{PDR}\|_{\sum_{k+1}^{PDR}}^2 &= \|X_{k+1} - f(X_k)\|_{\sum_{k+1}^{PDR}}^2 \\ &= \|x_{k+1} - (x_k + [\mathbf{SL}_{t,t+1} \cdot \cos(\phi) \\ &\quad \mathbf{SL}_{t,t+1} \cdot \sin(\phi)])\|_{\sum_{k+1}^{PDR}}^2 \end{aligned} \quad (21)$$

GNSS Factor This factor represents the constraint imposed by the GNSS measurements:

$$\|e_{k+1}^{GNSS}\|_{\sum_{k+1}^{GNSS}}^2 = \|X_{k+1} - GNSS_{k+1}\|_{\sum_{k+1}^{GNSS}}^2 \quad (22)$$

Therefore:

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \left[\|e_{i+1}^{PDR}\|_{\sum_{i+1}^{PDR}}^2 + \|e_{i+1}^{GNSS}\|_{\sum_{i+1}^{GNSS}}^2 \right] \right) \quad (23)$$

Optimization methods play a crucial role in determining optimal values. In our strategy, we opted for the Levenberg-Marquardt (LM) method [24], known for its effectiveness, using the specialized library GTSAM [25].

III. EXPERIMENTAL EVALUATION

A. Experimental Description

Experiments were conducted at an outdoor football ground at the University of Deusto in Bilbao, Spain, where we picked a small section of a sharp square route extending over 200 meters and bordered by a few towering buildings. This limited the clear view of the sky and led to some noisy data.

Volunteers were first asked to walk around the football ground on the outside marked lines and then to repeat the same path while running. In both cases, the volunteers had the freedom to choose their own normal pace. Data collection for pedestrian movement while walking took about 5.3 minutes, whereas data collection while running took 2.5 minutes.

B. Data Collection and processing

Smartphone's Data: The Xiaomi Mi 8, the first dual-frequency GNSS smartphone, was utilized to collect IMU and GNSS chip scale receiver data. The characteristics of the smartphone and its GNSS constellations can be found in the datasheet. We used the GetsensorsData app [26] to collect data from the smartphone sensors. Regarding the GNSS positions, this app only collects position coordinates provided by the smartphone's GNSS receiver, not the raw GNSS measurements. During data collection, the individual held the smartphone in portrait mode, ensuring it was aligned with their walking and running heading direction. The pedestrian made two full laps around the predetermined route while walking and running.

Ground Truth Data: For ground truth, we used Ublox C099 F9P application board, which includes the F9P-01B-00 L1/L2 GNSS module with built-in RTK technology, delivering centimeter-level accuracy. We leveraged the GPS, GLONASS, Galileo, and BeiDou constellations.

Separate data collection was conducted for obtaining walking and running reference datasets. Utilizing Ublox's U-center 22.07 software, we performed post-processing to refine the acquired data in the table view. The combined use of these constellations resulted in horizontal position accuracy surpassing 2.5 cm. This precise ground truth data served as the foundation for our comprehensive analysis.

Coordinates transformation: The GNSS data obtained from both the smartphone and Ublox devices were initially in the longitude, latitude, and altitude (LLA) format. To transform these coordinates into a more suitable representation, we utilized the built-in LLA to NED (North, East, Down) function in MATLAB. Subsequently, we applied a rotation matrix to further convert the coordinates into the East, North, and Up (ENU) format, which facilitated a more convenient and consistent analysis of the data.

Timely Synchronization of Ground Truth and Smartphone data: The GNSS data from Xiaomi Mi 8 was accurately synchronized with the Ublox data with precise synchronization at the seconds-level precision. The distance between the Ublox GNSS receiver and the smartphone was kept at zero meters.

Fusion Methods Parametrization In order to perform the experiments under the same conditions as in references [18], [19], both KF and FGO were parameterized with the same values as in those works. In the case of the KF, its parametrization in our case, we set the KF parameters as follows:

$$P = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \quad Q_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

where P is the initial state covariance matrix, Q_k is the process noise covariance matrix, and R is the measurement noise covariance matrix.

For the FGO, the GTSAM (Georgia Tech Smoothing and Mapping) library facilitated the import and utilization of two-dimensional Pose2 representations within a NonLinearFactorGraph for our work. And its parametrization is as follows:

odometryNoise = noiseModel.Diagonal.Sigmas([0.01; 0.01; 0.1]);

priorNoise = noiseModel.Diagonal.Sigmas([2; 2; 0.01]);

Those GTSAM library instructions define the covariance matrices (x, y and theta components) of the PDR and GNSS measurement factors, respectively. Subsequently, the optimization was performed using the Levenberg-Marquardt optimizer.

C. Results

1) **Walking Mode:** The developed PDR and GNSS data of the smartphone obtained from a walking mode were fused via KF and FGO algorithms. Subsequently, we conducted an in-depth analysis, and the resulting findings are presented in the accompanying figures. Figure 2 illustrates the trajectories obtained from the fusion of KF and FGO, as well as the GNSS alone, along with a comparison to the ground truth. Figure 3 illustrates the analysis of the Cumulative Distribution Function (CDF) in relation to Horizontal Position Errors (HPE). The graphs clearly demonstrate that FGO exhibits significantly

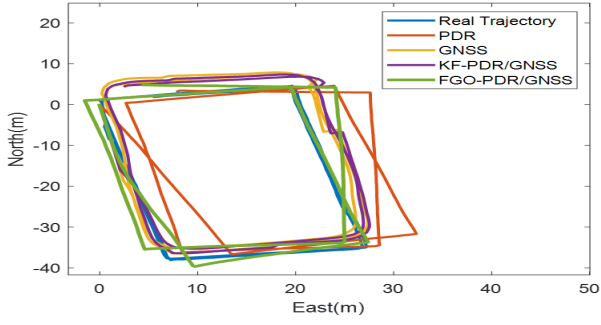


Fig. 2. Comparison of trajectories in walking mode

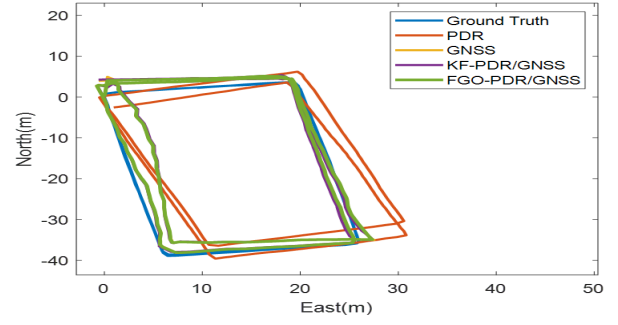


Fig. 5. Comparison of trajectories in running mode

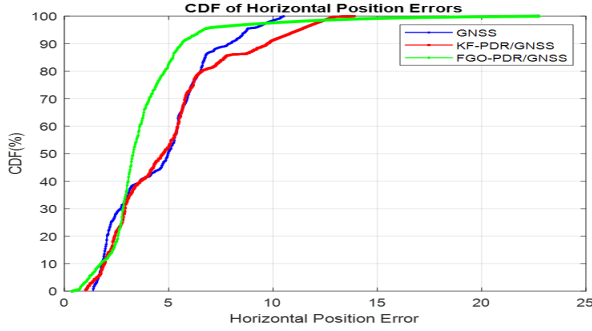


Fig. 3. Cumulative distribution function of horizontal position errors in walking mode

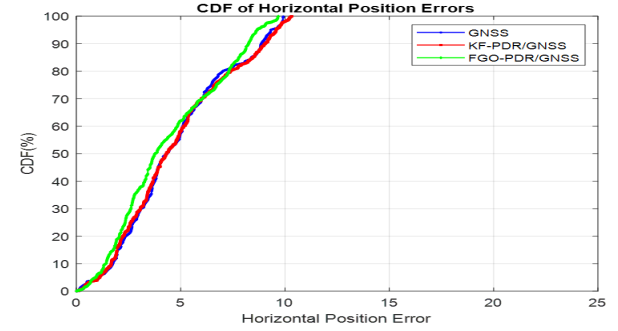


Fig. 6. Cumulative distribution function of horizontal position errors in running mode

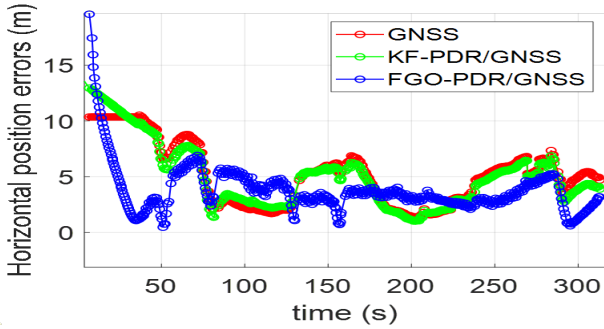


Fig. 4. Horizontal position errors over time in walking mode

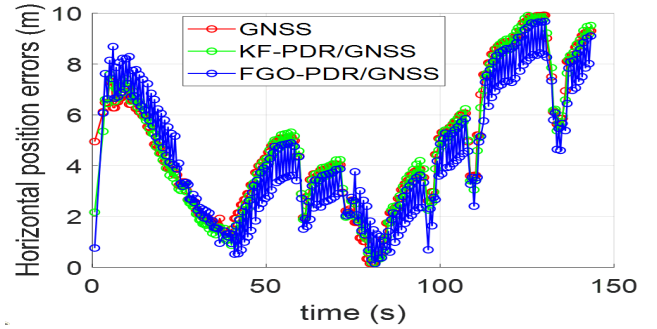


Fig. 7. Horizontal position errors over time in running mode

lower positional errors compared to KF. Moreover, the error curves against time can be seen in figure 4 and reveal that FGO consistently exhibits lower errors compared to KF and GNSS alone.

2) **Running Mode:** In the running experiment, we replicated the setup as described earlier. The PDR and GNSS data collected in the running mode were combined using KF and FGO. Comparing the results, FGO outperformed KF in terms of estimation accuracy. The trajectory plots in Figure 5 clearly illustrate the superior performance of FGO over KF. Additionally, examining the CDF of HPE in Figure 6, FGO exhibited lower positional errors compared to KF. This observation is further supported by Figure 7, demonstrating

the improved estimation quality of FGO over time.

3) **Discussion:** The statistical measures provided in Table I offer valuable insights into the performance of the FGO-based PDR+GNSS fusion algorithm compared to the KF algorithm. It is worth noting that the FGO-based PDR+GNSS fusion showcases a substantial decrease in horizontal position errors, surpassing the results obtained from the KF. Specifically, the mean values exhibit an impressive reduction of 30.6% in walking and 10 % in running mode highlighting the enhanced accuracy achieved by the FGO-based PDR+GNSS fusion. Furthermore, the median values demonstrate a significant decrease of 24.7% in walking and 6.15 % in running mode further solidifying the superiority of the FGO approach in

mitigating errors. These findings underscore the effectiveness and efficacy of the FGO fusion method in achieving more precise and reliable positioning results. Remarkably, the disparity in accuracy between the walking and running modes became evident in the FGO results. Notably, the running mode exhibited a discernible decrease in accuracy, attributed to the inherent challenges of running movements, including heightened PDR errors, coupled with less correlated GNSS measurements. These findings underscore the critical importance of accounting for such factors when assessing the performance of FGO in diverse pedestrian scenarios.

TABLE I

2D ERROR ANALYSIS: MEAN AND MEDIAN VALUES FOR WALKING AND RUNNING EXPERIMENTS

	<i>Walking Experiment</i>		<i>Running Experiment</i>	
Fusion Algorithm	Median	Mean	Median	Mean
FGO	3.78m	3.29m	4.44m	3.80m
KF	5.02m	4.74m	4.73m	4.22m
GNSS (Alone)	5.17m	5.28m	4.80m	4.37m

IV. CONCLUSION

In summary, our study compared FGO and KF in running and walking modes. FGO outperformed KF, showing promise for pedestrian positioning advancements. However, FGO's accuracy was lower in running mode due to larger PDR errors and less correlated GNSS measurements. In the future, it would be great to see how well FGO performs when combined with PDR and GNSS in different scenarios. This can be done by using various datasets that capture different types of pedestrian movements. It will help us understand and analyze the effectiveness of FGO in this fusion architecture. Continued refinement of FGO will enhance localization solutions in real-world scenarios.

REFERENCES

- [1] Jacek Paziewski. Recent advances and perspectives for positioning and applications with smartphone gnss observations. *Measurement Science and Technology*, 31(9):091001, 2020.
- [2] Cezary Specht, Tomasz Szot, Paweł Dąbrowski, and Mariusz Specht. Testing gnss receiver accuracy in samsung galaxy series mobile phones at a sports stadium. *Measurement Science and Technology*, 31(6):064006, 2020.
- [3] Abdelsatar Elmezayen, Mohamed Elsheikh, Umar Iqbal, and Aboelmagd Noureldin. Examining the capabilities of smartphone-based high precision positioning in challenging suburban environments. In *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, pages 2329–2338, 2022.
- [4] Jian Kuang, Xiaoji Niu, and Xingeng Chen. Robust pedestrian dead reckoning based on mems-imu for smartphones. *Sensors*, 18(5):1391, 2018.
- [5] Alwin Poullose, Benaoumeur Senouci, and Dong Seog Han. Performance analysis of sensor fusion techniques for heading estimation using smartphone sensors. *IEEE Sensors Journal*, 19(24):12369–12380, 2019.
- [6] Dayu Yan, Chuang Shi, and Tuan Li. An improved pdr system with accurate heading and step length estimation using handheld smartphone. *The Journal of Navigation*, 75(1):141–159, 2022.
- [7] Michele Basso, Alessio Martinelli, Simone Morosi, and Fabrizio Sera. A real-time gnss/pdr navigation system for mobile devices. *Remote Sensing*, 13(8):1567, 2021.
- [8] Feng Zhu, Xianlu Tao, Wanke Liu, Xiang Shi, Fuhong Wang, and Xiaohong Zhang. Walker: Continuous and precise navigation by fusing gnss and mems in smartphone chipsets for pedestrians. *Remote sensing*, 11(2):139, 2019.
- [9] Xianghong Li, Dongyan Wei, Wenchao Zhang, Ying Xu, and Hong Yuan. Integrated pdr/gnss at different times for pedestrian localization in urban canyon. *IEEE Access*, 9:22698–22710, 2021.
- [10] A Rehman, H Shahid, MA Afzal, and H MA Bhatti. Accurate and direct gnss/pdr integration using extended kalman filter for pedestrian smartphone navigation. *Gyroscope and Navigation*, 11:124–137, 2020.
- [11] Rui Zhang, Jing Mi, Jing Li, and Qing Wang. A continuous pdr and gnss fusing algorithm for smartphone positioning. *Remote Sensing*, 14(20):5171, 2022.
- [12] Antonio Angrisano, Mario Vultaggio, Salvatore Gaglione, and Nicola Crocetto. Pedestrian localization with pdr supplemented by gnss. In *2019 European Navigation Conference (ENC)*, pages 1–6. IEEE, 2019.
- [13] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.
- [14] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [15] Zheng Gong, Peilin Liu, Qiang Liu, Ruihang Miao, and Rendong Ying. Tightly coupled gnss with stereo camera navigation using graph optimization. In *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*, pages 3070–3077, 2018.
- [16] Seyed Fakoorian, Kyohei Otsu, Shehryar Khattak, Matteo Palieri, and Ali-akbar Agha-mohammadi. Rose: Robust state estimation via online covariance adaption. In *Robotics Research*, pages 452–467. Springer, 2023.
- [17] Weisong Wen, Xiwei Bai, Yin Chiu Kan, and Li-Ta Hsu. Tightly coupled gnss/ins integration via factor graph and aided by fish-eye camera. *IEEE Transactions on Vehicular Technology*, 68(11):10651–10662, 2019.
- [18] Changhui Jiang, Yuwei Chen, Chen Chen, Jianxin Jia, Haibin Sun, Tinghuai Wang, and Juha Hyypä. Implementation and performance analysis of the pdr/gnss integration on a smartphone. *GPS Solutions*, 26(3):81, 2022.
- [19] Changhui Jiang, Yuwei Chen, Chen Chen, Jianxin Jia, Haibin Sun, Tinghuai Wang, and Juha Hyypä. Smartphone pdr/gnss integration via factor graph optimization for pedestrian navigation. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.
- [20] Harvey Weinberg. Using the adxl202 in pedometer and personal navigation applications. *Analog Devices AN-602 application note*, 2(2):1–6, 2002.
- [21] Weisong Wen, Tim Pfeifer, Xiwei Bai, and Li-Ta Hsu. Factor graph optimization for gnss/ins integration: A comparison with the extended kalman filter. *NAVIGATION: Journal of the Institute of Navigation*, 68(2):315–331, 2021.
- [22] Ryan M Watson and Jason N Gross. Robust navigation in gnss degraded environment using graph optimization. In *Proceedings of the 30th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2017)*, pages 2906–2918, 2017.
- [23] Guangyi Guo, Ruizhi Chen, Xiaoguang Niu, Ke Yan, Shihao Xu, and Liang Chen. Factor graph framework for smartphone indoor localization: Integrating data-driven pdr and wi-fi rtt/rss ranging. *IEEE Sensors Journal*, 2023.
- [24] Henri P Gavin. The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, 19, 2019.
- [25] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.
- [26] Antonio R Jiménez, Fernando Seco, and Joaquín Torres-Sospedra. Tools for smartphone multi-sensor data registration and gt mapping for positioning applications. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2019.

Publication P2

Published in: IEEE Transactions on Instrumentation and
Measurement (TIM)

Comparison of FGO and KF for PDR-GNSS Fusion Under Different PDR Errors

Amjad Hussain Magsi¹ and Luis Enrique Díez²

Abstract—Smartphone-based positioning, heralded for its widespread accessibility, encounters challenges due to its reliance on global navigation satellite systems (GNSSs) in unfavorable conditions such as urban canyons, tunnels, and indoor areas. Even in clear-sky conditions, signal distortions, interruptions, and the limitations of cost-effective smartphone GNSS prompt researchers to explore alternative positioning methods. This has led to the adoption of sensor fusion techniques, often integrating the inertial measurement unit (IMU) for its complementary features with GNSS. In pedestrian localization, the fusion of pedestrian dead reckoning (PDR) and GNSS, traditionally employing the Kalman filter (KF) as the main fusion algorithm, has been standard practice. The emerging factor graph optimization (FGO) algorithm has recently gained attention for its better accuracy for inertial navigation system (INS)-GNSS fusion architectures, especially under GNSS outliers. However, the FGO implementation for PDR-GNSS fusion architecture has been less investigated, and little is known about its performance under different PDR outliers. As the different gait dynamics of humans and transient variations in the way the smartphone is carried, the PDR system can generate short and high errors (SHEs) or continuous and low errors (CLEs). We analyze the improvement of FGO over KF in mitigating these PDR errors in the PDR-GNSS fusion architectures for smartphone-based positioning. Since FGO is a smoothing technique and KF is a filtering method, for a fairer comparison, we also implemented a smoothed KF (SKF) using the Rauch-Tung-Striebel smoother (RTSS) technique. Our investigation, involving ten individuals with diverse heights, genders, and gait patterns in walking and running motions, underscores FGO's superior performance in the presence of PDR errors and across various pedestrian and motion scenarios, achieving a stable 25% improvement for the mean position error and 30% for the median position error in comparison to KF, 24% mean improvement, and 32% median improvement in comparison to SKF. Furthermore, the convergence time for FGO after the SHE PDR errors is comparably shorter than SKF and KF.

Index Terms—Factor graph optimization (FGO), fusion, global navigation satellite system (GNSS), Kalman filter (KF), outliers, pedestrian, pedestrian dead reckoning (PDR), smartphone.

I. INTRODUCTION

WITH the prevalence and accessibility of smartphones in today's digital age, pedestrian positioning has

Manuscript received 1 January 2024; revised 14 June 2024; accepted 2 July 2024. Date of publication 12 August 2024; date of current version 26 August 2024. This work was supported in part by the Research Training Grants Program of the University of Deusto and in part by REPIN++ under Grant RED2022-134355-T. The Associate Editor coordinating the review process was Dr. Martti Kirkko-Jaakkola. (Corresponding author: Amjad Hussain Magsi.)

The authors are with the Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain (e-mail: amjad.hussain@deusto.es; luis.enrique.diez@deusto.es).

Digital Object Identifier 10.1109/TIM.2024.3440373

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <https://creativecommons.org/licenses/by/4.0/>

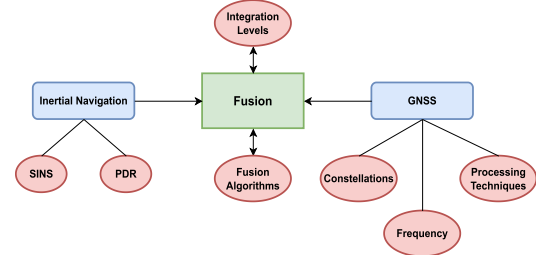


Fig. 1. Inertial-GNSS fusion architecture mechanism.

witnessed a substantial increase. Smartphones have become the predominant tools for individuals relying on positioning and localization services, with the global navigation satellite system (GNSS) playing a pivotal role in smartphone-based positioning, especially optimized for outdoor environments characterized by unobstructed views of the sky and direct satellite reception [1]. However, the performance of GNSS deteriorates in indoor environments such as tunnels and buildings, as well as in densely urban outdoor areas. In these scenarios, GNSS signals become weaker and susceptible to multipath interference and non-line-of-sight (NLOS) conditions, leading to noise issues and occasional data gaps in smartphone GNSS chips [2], [3], [4], [5]. To address this issue, researchers have focused more on alternative positioning solutions and have come up with the multisensor fusion concept in order to boost the performance of GNSS-based positioning in smartphones. Modern smartphones are equipped with a range of sensors that can serve as alternative positioning. These sensors include the camera, inertial sensors, barometer, light sensors, BLE, Wi-Fi, and even LiDAR. Among them, inertial sensors hold greater importance due to their complementary features with GNSS. They offer low short-term errors and are not susceptible to signal blockage caused by multipath interference or NLOS conditions. Furthermore, they do not necessitate additional infrastructure. This makes inertial + GNSS a well-known and popular fusion architecture [6], [7], [8]. However, this fusion can be implemented in several ways, depending on the different types of inertial navigation, GNSS processing techniques, the utilization of multiple constellations and frequencies, fusion algorithms, and integration levels, as depicted in the accompanying Fig. 1.

The high noise of the microelectromechanical systems (MEMSs) inertial measurement unit (IMU) present in smartphones makes the strapdown inertial navigation system (SINS) impractical due to the fast error accumulation over time. In addition, smartphone carrying modes do not allow

for the implementation of techniques such as zero velocity update (ZUPT) for reducing the error rate [9]. Pedestrian dead reckoning (PDR) is an alternative navigation method based on integrating each step displacement, reducing the error rate this way compared to pure inertial navigation system (INS). Consequently, the fusion of PDR and GNSS is a common architecture for pedestrian positioning using smartphones, which is still an open research topic [10], [11], [12], [13], [14]. In order to alleviate the errors of low-cost IMUs, PDR is based on detecting steps, estimating their length and direction, and then integrating the associated displacements to update the user's position. We can divide the errors associated with such detection and estimation processes into two types.

- 1) *Short and High Errors (SHEs)*: Situations such as transient variations in the way the smartphone is carried can generate brief but large errors (outliers).
- 2) *Continuous and Low Errors (CLEs)*: Human gait dynamics vary due to changes in walking pattern, gender, and height, and these factors influence the performance of individual PDR blocks, making it difficult to develop methods that generalize PDR performance for any user. This leads to CLEs in the PDR block.

To address this challenge, fusion algorithms play a crucial role in mitigating the impact of PDR errors for such fusion architectures. Historically, the Kalman filter (KF) has served as the primary fusion algorithm for the PDR-GNSS fusion architecture. However, achieving high accuracy using KF and its variant models has been challenging due to the linear first-order Markov dynamic model, Gaussian noise assumptions, and the fact that it considers only the latest available measurement. Instead, the newly proposed factor graph optimization (FGO) [15] is a well-liked approach for the multisensor fusion concept. Factor graphs are graphical models used to effectively address complex estimation problems such as simultaneous localization and mapping (SLAM) [16]. FGO optimizes state estimates through iterative solutions, treating state estimation as a nonlinear least-squares problem. It employs historical measurements and iterative solvers such as Gauss–Newton or Levenberg–Marquardt to minimize the cost function, achieving optimal estimations. Consequently, there is a growing trend to utilize FGO as an alternative to KF for both GNSS processing and inertial-GNSS fusion. Researchers [17], [18], [19], [20], [21], [22], [23] fused INS and GNSS using FGO and compared its performance against the traditional KF. Their findings highlighted FGO's notable capability to mitigate GNSS outliers, particularly in scenarios characterized by urban canyons or limited visibility of satellites during vehicular testing.

Up until now, there has been limited research on PDR-GNSS using FGO on smartphones. Past studies mainly focused on such fusion architectures for walking with smartphones and found that FGO outperformed traditional KF [10], [11], [12], [24], [25]. In the same way that some studies have explored the impact of GNSS outliers on FGO in INS-GNSS fusion, we believe that it is equally important to examine FGO's effect on PDR outliers, which occur

more frequently than INS outliers. To the best of our knowledge, no one has yet investigated the impact of PDR errors on FGO performance. We compared FGO and KF in PDR-GNSS fusion for walking and running motions using a single dataset. Results showed that FGO performed better overall though its effectiveness slightly decreased during running compared to walking. In the running motion, the PDR's accuracy was affected due to misleading step detections and step length estimation compared to walking. This highlights a significant research gap in evaluating FGO against KF for various PDR errors stemming from diverse smartphone disturbances, multiple pedestrians with differing heights, genders, walking patterns, and distinct motions. In this extended version of our previous article [26], we highlight a deeper understanding of FGO's capabilities and limitations.

It is usually hard to generalize PDR performance to different users and motion modes. We aim to provide valuable insights into the effectiveness of fusion algorithms within the PDR-GNSS architecture, with a specific focus on their performance in PDR errors, i.e., various pedestrians, motions (CLE), and outliers (SHE). Since FGO is a smoothing technique and KF is a filtering method, for a fairer comparison, we also implemented a smoothed version of KF (SKF) using the Rauch–Tung–Striebel smoother (RTSS) technique [27]. We hypothesize that FGO deals with that better than KF and SKF.

The main contributions of this research are given as follows.

- 1) *Exploration of PDR Outliers*: We examined how FGO, SKF, and KF performed under various PDR outliers (SHEs), which are caused by different disturbances to smartphone carrying, which produce a negative influence on the PDR accuracy. FGO's effectiveness in handling these PDR outliers is analyzed, as it has been done similarly in the literature to assess the impact of GNSS outliers for INS-GNSS.
- 2) *Analysis of Different Pedestrians and Motions With FGO*: The performance of the PDR block is impacted by pedestrians with varying heights, walking patterns, and genders. To scrutinize the associated PDR errors (continuous and long errors) stemming from diverse pedestrian characteristics and movements, we examined multiple individuals of varying heights and genders engaged in both running and walking activities. Our analysis aimed to evaluate the effectiveness of FGO compared to KF in mitigating these PDR errors associated with the lack of generalization ability.

The remaining article is structured as follows. Section II provides an overview of the related work. Section III presents the proposed methodology, which includes the development of the PDR mechanism and the development of KF- and FGO-based PDR-GNSS fusion architectures. Section IV describes the experimental evaluation and setup. Section V details the results. Finally, Section VI concludes this article, summarizing the findings and future work.

II. RELATED WORK

A. Traditional PDR-GNSS Fusion Approaches

PDR is a widely recognized inertial navigation algorithm designed for pedestrians that works on three fundamental aspects: step detection, step length estimation, and step heading estimation. These displacement-related parameters are usually obtained from data provided by inertial sensors such as accelerometers and gyroscopes. PDR, on its own, tends to accumulate position errors over time, mainly due to variations in pedestrian walking patterns. Therefore, enhancing localization accuracy necessitates integration with GNSS. Many researchers have already explored PDR-GNSS fusion architectures [13], [28], [29], [30]. [28] proposed the PDR-GNSS fusion method by jointly estimating the heading angle and step length. However, they did not account for the negative influence of GNSS MP and NLOS signal reception. Therefore, later on, Rehman et al. [13] focused on the MP and NLOS conditions, and with this, they addressed the issue of cumulative errors stemming from the estimation of heading and step size. They mitigate these errors by implementing metrics for mean cumulative heading error and cumulative step length error, respectively. Following this error reduction process, they integrate PDR with GNSS to enhance the overall performance. Another study [29] introduced a precise displacement estimation method by integrating PDR and GNSS using a time-differenced carrier phase (TDCP). The TDCP-PDR-GNSS integration approach demonstrated a significant reduction in position errors, thereby improving the accuracy of the estimated displacements. The optimization-based PDR (OBPDR) was proposed [30] through an improved finite state machine (IFSM) gait detection method. Then, they fused this OBPDR with GNSS for better results.

B. FGO-Based PDR-GNSS Fusion Architecture

The traditional fusion approaches have several issues that have been highlighted by [31]. These issues are summarized as follows.

- 1) Traditional single-source navigation methods lack efficiency, while integrated systems, which emphasize specific sensors and measurements, limit adaptability and precision.
- 2) System miniaturization and clustering introduce non-linear and non-Gaussian challenges in navigation, exceeding traditional filtering methods' precision capabilities.
- 3) The navigation system faces sensor data synchronization challenges, held back by sensor heterogeneity and varying working conditions, making traditional alignment methods inadequate.

To address these issues, the USA-based Advanced Defense Research Project Agency (DARPA) introduced the concept of all-source positioning and navigation (ASPN) in November 2010. ASPN proposes to merge data sources for versatile navigation, featuring quick integration and adaptability. However, practical implementation remains challenging. Georgia Institute of Technology came up with the concept of factor graph-based methods and made early attempts with this for ASPN. Factor graph-based methods, known for plug-and-play

functionality and the ability to handle complex, asynchronous data, have gained recent attention [32].

The introduction of the FGO formulation by Indelman et al. [33] has paved the way for a fresh perspective on multisensor integration, as further explored by [34] and [35]. The factor graph improves state estimation by considering past measurements and system updates. Unlike the KF, FGO iteratively adjusts the measurement model, enhancing accuracy. Factor graphs efficiently use time correlations between past and present data through batch-style processing. In this mode, a factor graph acts as both a forward filter and a backward smoother after each measurement update. Thus, FGO is applied in demanding GNSS situations, as highlighted in works by [36], [37], [38], [39], and [40]. Furthermore, Pfeifer and Protzel [41] illustrated FGO's significant promise in sensor fusion, even when dealing with non-Gaussian noise models. Researchers have recently worked on the fusion of INS and GNSS for vehicular navigation systems, in which they fused the professional inertial and GNSS devices [18], [19], [42], [43], [44].

In the context of pedestrian navigation using a smartphone, a relatively limited body of research exists. Only a handful of researchers have explored the fusion of PDR and GNSS for smartphones. Jiang et al.'s [10], [11] research team was the first one proposing FGO-based PDR-GNSS fusion: In [10], they delved into the performance and implementation of FGO compared to KF for single-user and walking motion data only, in which they showed that FGO is better than KF. Later, in [11], they proposed two modifications by introducing two constraints (position and distance) in the factor graph and evaluated the performance against KF using multiple datasets collected in open sky and partially open sky conditions by placing the smartphone handheld and in a backpack and pocket for only walking motion. They also proposed a cooperative positioning system based on FGO PDR-GNSS fusion [24]. More recently, they proposed a tight-coupled FGO PDR-GNSS fusion-based [25]. They employed fuzzy adaptive FGO to further enhance accuracy and suppress pseudorange outliers. In a similar work, Zhong et al. [12] investigated the tightly coupled FGO-based PDR-GNSS fusion for urban canyons in which they created the factor graph constraints from raw GNSS pseudorange, Doppler measurements, and raw accelerometer-based PDR algorithm. They also proposed smoothness-driven motion model-based pedestrian navigation with a small acceleration for trajectory smoothing and increased resilience to potential GNSS outliers in standalone positioning.

In the above studies, the authors have primarily focused on the quality of the fused solution and the impact of the GNSS outliers, considering only walking motion. However, to the best of our knowledge, no one has yet examined the impact of PDR errors caused by various human motions, different gait patterns, and transient variations in smartphone positioning. In our previous article [26], we highlighted the performance of FGO for walking and running motion using a single user. Each pedestrian has a unique walking style, influenced by factors such as height, gender, body posture, and dynamics. The results revealed FGO's superiority in both motions, but, still, there is a need to investigate FGO's performance with different

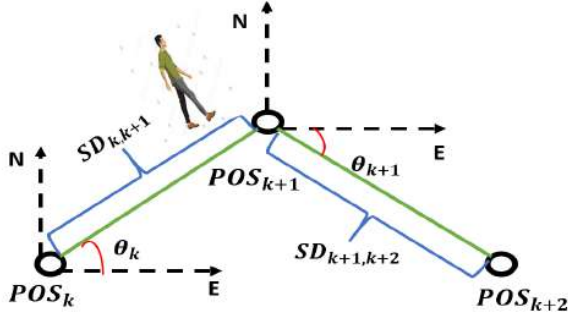


Fig. 2. PDR mechanism.

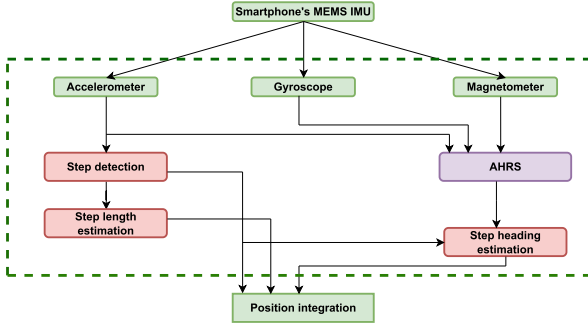


Fig. 3. Smartphone-based PDR.

aspects of PDR errors. Second, due to transient variations in smartphone position from normal placement to sudden, rapid changes that can lead to misdetected steps and heading errors, this article aims to analyze the FGO performance under PDR outliers (SHEs), in the same fashion as other works studied the impact of GNSS outliers for both PDR-GNSS and INS-GNSS [19].

III. METHODOLOGY

In this section of this article, we elaborate on the methodological approach adopted in this study, which includes the development of PDR and its fusion with GNSS through both the KF and FGO techniques.

A. Development of PDR

PDR estimates pedestrian movements based on the detection of steps, step length, and heading estimation. The PDR working mechanism shown in Fig. 2 can be mathematically written as follows:

$$\mathbf{Pos}_{k+1} = \mathbf{Pos}_k + \Delta\mathbf{Pos}_{k|k+1} \quad (1)$$

where $\mathbf{Pos}_{k+1}$ and $\mathbf{Pos}_k$ represent the position at steps $k+1$ and k , respectively, and $\Delta\mathbf{Pos}_{k|k+1}$ represents the position increment from step k to $k+1$ that, expressed in the east north up (ENU) reference system [9], can be written as

$$\Delta\mathbf{Pos}_{k|k+1} = \begin{bmatrix} \Delta\mathbf{Pos}_{k|k+1}^E \\ \Delta\mathbf{Pos}_{k|k+1}^N \end{bmatrix} = \begin{bmatrix} \mathbf{SL}_{k|k+1} \cdot \cos(\phi) \\ \mathbf{SL}_{k|k+1} \cdot \sin(\phi) \end{bmatrix} \quad (2)$$

where the heading angle is indicated by ϕ and the step length from step k to step $k+1$ is shown by $\mathbf{SL}_{k|k+1}$.

A schematic illustration of the PDR system from the smartphone is presented in Fig. 3. For step detection,

we utilized the magnitude of the triaxial accelerometer [see (3)] as an input signal. To detect steps, we used the code provided by [10], and we modified some threshold values to adapt the step detector to different motion modes

$$|\text{Acc}_k| = \sqrt{\text{Acc}_{xk}^2 + \text{Acc}_{yk}^2 + \text{Acc}_{zk}^2}. \quad (3)$$

For step length estimation, we employed Weinberg's algorithm [45]; within each detected step, the step length is determined from the range of the acceleration magnitude values and a scaling constant factor

$$\mathbf{SL}_{k|k+1} = C \cdot [\max(|\text{Acc}_k|) - \min(|\text{Acc}_k|)]^{1/4} \quad (4)$$

where $\mathbf{SL}_{k|k+1}$ represents the estimated step length between two consecutive acceleration peaks and C is a constant factor used to scale the estimated step length.

For step heading estimation, we employed the heading angle obtained from the attitude and heading reference system (AHRS) integrated into the smartphone at each detected step epoch.

B. KF-Based PDR-GNSS Fusion Architecture in Smartphone

The KF operates through two steps: the prediction step and the correction step. In the prediction step, the state of the system at the next time step is estimated by using a dynamic model and the current state of the system. The correction step, on the other hand, involves refining the estimated state by incorporating measurements from sensors. This allows for an update of the estimated state, resulting in a more accurate position estimation. The state vector, represented as $\mathbf{X}$, serves as the fundamental element for the state transformation in PDR-GNSS integration, which is the position errors of PDR in x - and y -coordinates. Therefore, we have implemented a KF-based fusion using an error state approach instead of a position state, which is based on a linear dynamic model as outlined by [10] and [11]

$$\mathbf{X} = \begin{pmatrix} \Delta\text{PDR}_x \\ \Delta\text{PDR}_y \end{pmatrix}.$$

The formulation of this transformation is expressed as follows:

$$\mathbf{X}_{k+1} = \mathbf{F}_{k|k+1} \cdot \mathbf{X}_k + \mathbf{W}_{k+1} \quad (5)$$

where $\mathbf{X}_{k+1}$ and $\mathbf{X}_k$ indicate the state vector at $k+1$ and k epochs, respectively; $\mathbf{W}_{k+1}$ is a process noise vector, which is assumed to be subject to a standard Gaussian distribution, expressed as $\mathbf{W}_{k+1} \sim \mathcal{N}(0, \mathbf{Q}_k)$, where $\mathbf{Q}_k$ represents the covariance matrix of the process noise and $\mathbf{F}_{k|k+1}$ refers to the transformation matrix at epochs $k+1$ and k

$$\mathbf{F}_{k|k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (6)$$

The measurement model is

$$\mathbf{Z}_{k+1} = \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1} + \mathbf{V}_{k+1}. \quad (7)$$

Equation (7) represents the general measurement model, combining the state vector $\mathbf{X}_{k+1}$, observation matrix $\mathbf{H}_{k+1}$, and measurement noise vector $\mathbf{V}_{k+1}$ to calculate the predicted measurement $\mathbf{Z}_{k+1}$. The actual observations are calculated

differently in the following equation, specifically tailored for our PDR-GNSS fusion approach:

$$\mathbf{Z}_{k+1} = \mathbf{Pos}_{k+1}^{\text{GNSS}} - \mathbf{Pos}_{k+1}^{\text{PDR}}. \quad (8)$$

In this equation, the actual observations $\mathbf{Z}_{k+1}$ are derived as the difference between the GNSS position estimate $\mathbf{Pos}_{k+1}^{\text{GNSS}}$ and the PDR position estimate $\mathbf{Pos}_{k+1}^{\text{PDR}}$ at time $k+1$. Also, the $\mathbf{H}_{k+1}$ observation matrix can be written as follows:

$$\mathbf{H}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (9)$$

The prediction step of KF can be mathematically written as follows:

$$\mathbf{X}_{k+1}^- = \mathbf{F}_{k|k+1} \cdot \mathbf{X}_k \quad (10)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_{k|k+1} \cdot \mathbf{P}_k \cdot (\mathbf{F}_{k|k+1})^T + \mathbf{Q}_k. \quad (11)$$

In these equations, $\mathbf{X}_{k+1}^-$ and $\mathbf{P}_{k+1}^-$ are the a priori estimated state vector and covariance matrix, respectively, obtained by applying the dynamic model to the previous estimation. Then, the correction step for KF updates the state estimation as follows:

$$\hat{\mathbf{X}}_{k+1} = \mathbf{X}_{k+1}^- + \mathbf{K}_{k+1} \cdot (\mathbf{Z}_{k+1} - \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1}^-) \quad (12)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \cdot (\mathbf{H}_{k+1})^T \cdot [\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1}^- \cdot (\mathbf{H}_{k+1})^T + \mathbf{R}_{k+1}]^{-1} \quad (13)$$

$$\mathbf{P}_{k+1} = (\mathbf{I}_{2 \times 2} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{P}_{k+1}^- \quad (14)$$

where $\hat{\mathbf{X}}_{k+1}$ is the updated state estimation at $k+1$, $\mathbf{K}_{k+1}$ represents the Kalman gain matrix at $k+1$, and $\mathbf{I}_{2 \times 2}$ represents the identity matrix of size 2×2 .

C. Implementation of the SKF Using RTSS

As the KF is a forward filtering method that does not consider historical measurements, we employed the RTSS technique to implement the SKF [27]. This allows for a more equitable comparison with FGO, which is inherently a smoothing technique. The SKF implementation involved the following steps.

1) *Forward Pass (KF)*: Estimation of the state variables using the standard KF equations, as described above.

2) *Backward Pass (RTSS)*: Smoothing the state estimates using the RTSS equations. The RTSS algorithm involves a backward recursion that refines the forward pass estimates by considering future measurements, leading to more accurate state estimates.

The RTSS equations are given as follows.

Smoothing Gain:

$$\mathbf{G}_k = \mathbf{P}_{k|k} \mathbf{A}^T \mathbf{P}_{k+1|k}^{-1}. \quad (15)$$

Smoothed State:

$$\hat{\mathbf{x}}_{k|N} = \hat{\mathbf{x}}_{k|k} + \mathbf{G}_k (\hat{\mathbf{x}}_{k+1|N} - \hat{\mathbf{x}}_{k+1|k}). \quad (16)$$

Smoothed Covariance:

$$\mathbf{P}_{k|N} = \mathbf{P}_{k|k} + \mathbf{G}_k (\mathbf{P}_{k+1|N} - \mathbf{P}_{k+1|k}) \mathbf{G}_k^T. \quad (17)$$

In these equations, $\hat{\mathbf{x}}_{k|N}$ represents the smoothed state estimate at time k given all measurements up to time N , $\mathbf{P}_{k|N}$ is the smoothed covariance matrix, and $\mathbf{G}_k$ is the smoothing gain that adjusts the state estimate based on future observations.

3) *Smoothing Factor*: The term “smoothing factor” refers to the parameters influencing the RTSS process. In our implementation, these include the following.

- 1) *Process Noise Covariance ($\mathbf{Q}_k$)*: Tuned to reflect the variability and dynamics of pedestrian movement.
- 2) *Measurement Noise Covariance ($\mathbf{R}_k$)*: Determined based on the expected accuracy of GNSS and PDR measurements.
- 3) *State Initialization*: Initial state estimates and their covariances were carefully chosen to ensure stability and convergence of the smoother. These factors collectively determine the effectiveness of the smoothing process. For pedestrian navigation, the process and measurement of noise covariances were specifically tuned through empirical analysis to handle the variations in human gait and smartphone handling. Moreover, these noise covariances are listed under fusion method parameterization in Section IV.

D. FGO-Based PDR-GNSS Fusion Architecture in Smartphone

A factor graph is a probability model that utilizes a graph to represent the probability dependence of different variables. Specifically, a factor graph is a bipartite graph composed of two types of nodes: the unknown state variables and the factor nodes, which represent the conditional probabilities of some set of state variables. The undirected edges connect the state variable nodes and corresponding factor nodes. There is an edge between the two nodes only when the variable node and the factor node have a functional relationship. A factor graph allows the decomposition of a global probability function into the product of multiple local probability functions, reducing the complexity of solving the global function.

In sensor integration, we are usually interested in estimating a set of state variables $\mathbf{X}$ from a set of given measurements $\mathbf{Z}$. Assuming a Markov model, that conditional probability density function can be formulated

$$P(\mathbf{X}|\mathbf{Z}) = \prod_{i=1}^k \frac{P(z_i|x_i)P(x_i|x_{i-1}, u_i)}{P(z_i)} P(x_0) \quad (18)$$

where z_i represents the measurements observed at epoch i (e.g., GNSS measurements), x_i represents the system state at i , and u_i denotes the control input (e.g., PDR measurements).

The most often used estimator for these unknown state variables $\mathbf{X}$ is the maximum a posteriori or MAP estimate

$$\hat{\mathbf{X}} = \arg \max P(\mathbf{X}|\mathbf{Z}) = \arg \max \prod_{i=1}^k P(z_i|x_i)P(x_i|x_{i-1}, u_i). \quad (19)$$

In FGO-based integration, all these likelihood and transition probabilities are treated as the factorization of the global probability

$$\hat{\mathbf{X}} = \arg \max \prod_{j=1}^n f(x_j). \quad (20)$$

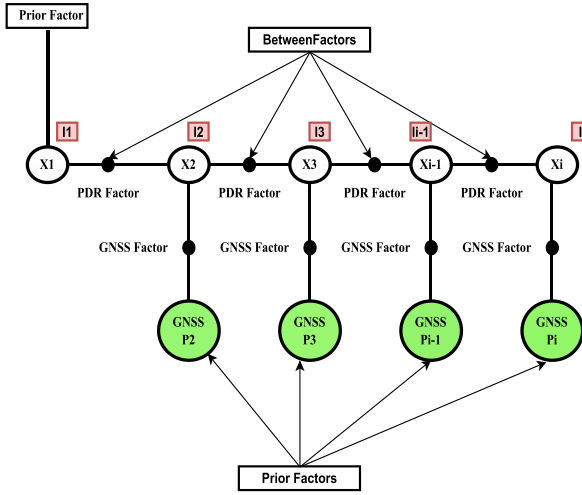


Fig. 4. Formation of PDR and GNSS factors within FGO mechanism.

Each factor models a constraint and must include a measure of uncertainty. The most common model is Gaussian noise

$$P(z_i|x_i) \propto \exp\left(-\frac{1}{2}\|h_i(x_i - z_i)\|^2 \nabla_i\right) \quad (21)$$

$$P(x_i|x_{i-1}) \propto \exp\left(-\frac{1}{2}\|\Phi_i(x_{i-1} - x_i)\|^2 \Omega_i\right) \quad (22)$$

where the function $\Phi_i(\cdot)$ describes the relationship between the preceding states x_{i-1} and x_i , the function $h_i(\cdot)$ represents the relationship between the state x_i and the measurement z_i , and the covariance matrices are denoted by ∇_i and Ω_i .

Taking the negative log and dropping the factor 1/2 transform the problem into the minimization of an error function, that is, a nonlinear least-squares problem

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \|\Phi_i(x_{i-1}) - x_i\|_{\Omega_i}^2 + \sum_{i=1}^k \|h_i(x_i) - z_i\|_{\nabla_i}^2 \right). \quad (23)$$

In our case, as shown in Fig. 4, the factors represent the GNSS measurements and the transition between states due to the motion model and PDR measurements.

1) *PDR Factor*: This factor signifies the constraint enforced by the motion model derived from the PDR system. As can be seen, the PDR factor is modeled using the true-state nonlinear dynamic model, in contrast to the error-state linear dynamic model used in the KF implementation. This way, the potential advantage of the FGO dealing with nonlinear models is counteracted. This strategy was also followed by [10], [11], and [12]

$$\begin{aligned} \|e_{k+1}^{\text{PDR}}\|_{\sum_{k+1}^{\text{PDR}}}^2 &= \|X_{k+1} - f(X_k)\|_{\sum_{k+1}^{\text{PDR}}}^2 \\ &= \left\| X_{k+1} - \left(x_k + \begin{bmatrix} \mathbf{S}_{L,t,t+1} \cdot \cos(\phi) \\ \mathbf{S}_{L,t,t+1} \cdot \sin(\phi) \end{bmatrix} \right) \right\|_{\sum_{k+1}^{\text{PDR}}}^2. \end{aligned} \quad (24)$$

2) *GNSS Factor*: Similarly, this factor represents the constraint imposed by the GNSS measurements

$$\|e_{k+1}^{\text{GNSS}}\|_{\sum_{k+1}^{\text{GNSS}}}^2 = \|X_{k+1} - \text{GNSS}_{k+1}\|_{\sum_{k+1}^{\text{GNSS}}}^2. \quad (25)$$

Therefore,

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \left[\|e_{i+1}^{\text{PDR}}\|_{\sum_{i+1}^{\text{PDR}}}^2 + \|e_{i+1}^{\text{GNSS}}\|_{\sum_{i+1}^{\text{GNSS}}}^2 \right] \right). \quad (26)$$

We utilized the Georgia Tech Smoothing and Mapping (GTSAM) library, an open-source tool developed by researchers at the Georgia Institute of Technology, to implement FGO [32]. This library is well-known for its effectiveness in solving problems related to SLAM, factor graphs, and pose graph-based optimization, making it user-friendly for FGO implementation in various applications. As mentioned before, the graph consists of PDR and GNSS factors. Whenever a new GNSS measurement is available, new PDR and GNSS factors are appended to the graph. After that, the Levenberg–Marquardt optimizer [46] is used to optimize the completely new graph, and this process is repeated after each new GNSS measurement. It is important to mention that other optimizers such as Newton–Gaussian, iSAM, and iSAM2 are also viable alternatives.

IV. EXPERIMENTAL EVALUATION

A. Experimental Description

It is challenging to develop a flexible PDR for different pedestrian walking patterns and various carrying modes of smartphones. We conducted two different experiments for SHE and CLE analyses.

For the SHE analysis, the experiment was conducted in an outdoor area with enough satellite visibility. The pedestrian held the smartphone initially aligned with its body and just walked at a normal pace for about 52 m on a predefined straight path, but while walking, the pedestrian moved his smartphone to generate short unexpected dynamics (outliers) as follows.

- 1) *Swinging of a Smartphone (Case I)*: The smartphone briefly entered a swinging mode for a few seconds before returning to its original position.
- 2) *Changing the Heading of a Smartphone (Case II)*: When changing the smartphone's heading, it was rotated counterclockwise for a few seconds twice along the entire path to observe outliers more clearly.
- 3) *Shaking of Smartphone (Case III)*: Similarly, the smartphone was shaken in an upright position for a brief duration.

In another experiment aimed at observing CLE analysis across various pedestrians and motions (walking and running), participants completed a full lap along a predefined trajectory outdoors. Throughout the entire duration, each pedestrian held the smartphone in front of their chest. Participants were instructed to walk and run at their own natural pace while maintaining the same position of the smartphone.

B. Data Collection and Processing

The Huawei Y8P smartphone was used for inertial sensors and GNSS data. For reference, the Ublox ANN-MB-00-00



Fig. 5. Data collection and experimental setup.

TABLE I
DIFFERENT PEDESTRIANS FOR DATA COLLECTION

Pedestrian	Height (cm)	Gender	Pace
A	155 cm	Female	Own Will
B	160 cm	Male	
C	157 cm	Male	
D	150 cm	Male	
E	147 cm	Female	
F	140 cm	Female	
G	180 cm	Male	
H	185 cm	Male	
I	188 cm	Male	
J	168 cm	Male	

L1/L2 multiband GNSS antenna, which has a C099 F9P application processing board, has been used. Data collection took place outdoors in the open sky at the football field of the University of Deusto, Bilbao, Spain. We selected a small section of a straight-line path, spanning over 52 m. Refer to Fig. 5 for a visual representation of the data collection and experimental setup.

1) *Smartphone Data*: The GetSensorsData app [47] was used to collect inertial sensors and GNSS data from the smartphone.

The SHE analysis experiment involved three datasets, including smartphone swinging, heading change, and shaking. For the CLE analysis experiment, 40 datasets were collected from ten pedestrians with diverse heights, walking patterns, and genders, as detailed in Table I, encompassing both walking and running motions. Pedestrians had the flexibility to choose their own walking pace during the experiments. However, we set the upper threshold values for walking and running motions in the step detector.

2) *Reference Data*: The data were collected using U-Center 23.08 software provided by Ublox. The GPS, GLONASS, Galileo, and BeiDou satellite constellations were used to

ensure the accuracy of the reference. The combined use of these satellite constellations resulted in a horizontal position accuracy of 2.5 cm.

3) *Timely Synchronization of Reference and Smartphone Data*: Every pedestrian, motion, and PDR outliers experiment has its own reference data, which has been synchronized with the smartphone via GPS time. We placed the Huawei Y8P smartphone's GNSS receiver right next to the reference data source (Ublox GNSS receiver), keeping the distance between them at 0 m. This proximity minimized any potential delays in the signals' travel, ensuring that our data were as precise as possible.

4) *Fusion Methods' Parameterization*: The efficacy of the FGO and KF fusion algorithms hinges on their adept utilization of parameter covariance settings. Throughout this study, we preserved uniform parameter settings for all pedestrian motions across the comprehensive dataset comprising 40 instances. Regarding the KF, we set the KF parameters as follows when we parameterize it in our case:

$$P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad Q_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

where P is the initial state covariance matrix, Q_k is the process noise covariance matrix, and R is the measurement noise covariance matrix.

In the domain of FGO, our approach involves two factors: PDR and GNSS. It is important to recognize that both of these factors come with uncertainties and noise. To handle these uncertainties, we have used GTSAM's odometry and prior noise model classes. We set the same values for both KF and FGO

odometryNoise

= noiseModel.Diagonal.Sigmas([2; 2; 0.1])

priorNoise = noiseModel.Diagonal.Sigmas([1.4; 1.4; 0.1]).

V. RESULTS

A. Analysis of Improvement in FGO Versus PDR Outliers

To characterize FGO performance over KF and SKF with transient variations in the way the smartphone is carried while walking, we fluctuated the smartphone during carrying in three different ways: swinging, changing the heading, and shaking the smartphone. From Figs. 6–8, it can be seen that with little disturbance, smartphone placement produces short-term high errors in the PDR models. We fused these PDR's displacements with GNSS using FGO and KF. In addition, we used SKF to make the comparison with FGO fair. The results indicated that FGO performed better than KF and SKF in all three cases.

Case 1: In the case of smartphone swinging, Fig. 6(a) illustrates the noticeable impact on PDR output, evident in red as the PDR trajectory. The swinging of the smartphone not only produced general positional error but also the heading of the smartphone was changed, and the misdetection of the steps generated outliers in the PDR output, as visualized in Fig. 6(a). As a result, the PDR trajectory lost its path during swinging, leading to heightened positional errors over time, as portrayed in Fig. 6(c) and highlighted in purple. In comparison, the smartphone's GNSS remained stable, offering a contrast to

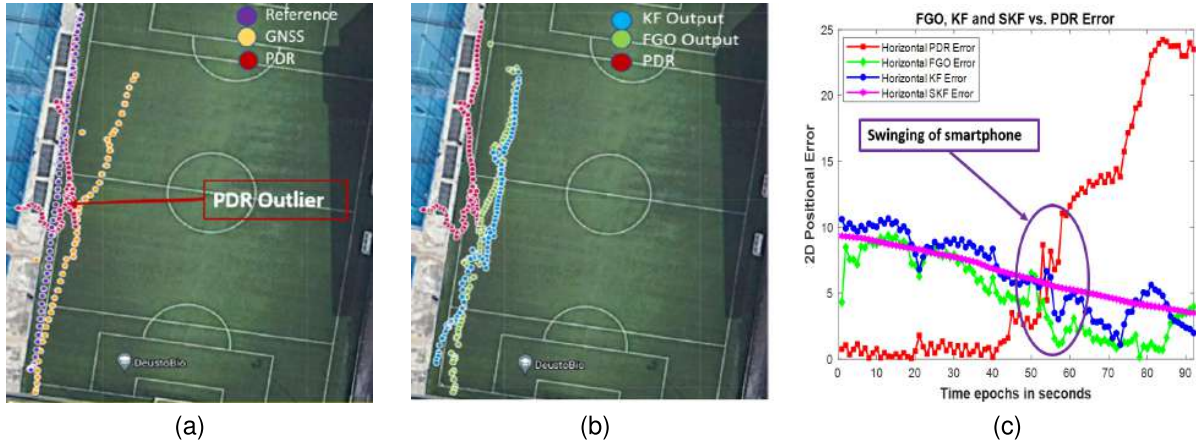


Fig. 6. (a) Reference, PDR output and GNSS trajectories are shown along with the effect of smartphone swinging on PDR's output highlighted in red. (b) Fused trajectories from KF and FGO, showing FGO's superior performance during smartphone swinging, outperforming KF. (c) Visualizes the horizontal position errors over time and reveals FGO's ability to mitigate PDR errors during smartphone swinging.

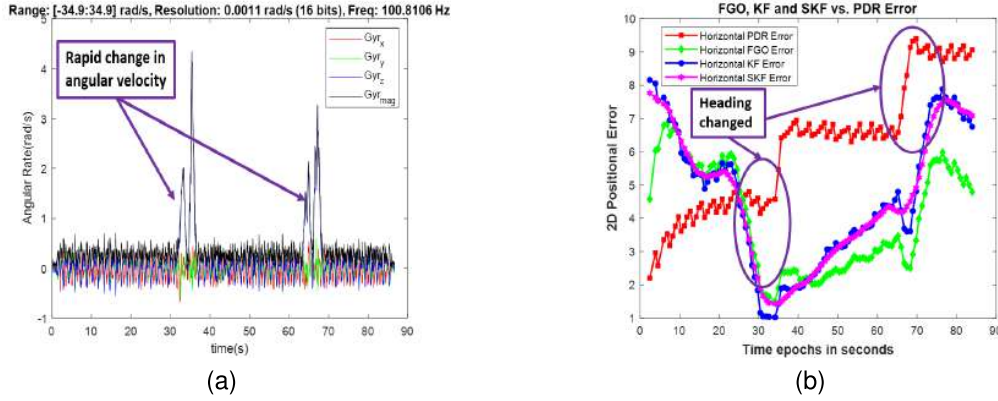


Fig. 7. (a) Rapid changes in the angular velocity when the smartphone was rotated. (b) Comparison of horizontal position errors from PDR, KF, SKF, and FGO with smartphone heading change (highlighted in purple).

the PDR's fluctuations. The fused trajectories from KF and FGO are shown in Fig. 6(b). It can be also seen that FGO is more robust to such high PDR error and produces better output especially when the PDR errors become so high. We computed the stable convergence time for these fusion techniques by setting the PDR error threshold at 10 m, which is when the errors peaked. We then observed how FGO, KF, and SKF responded to these outliers. The convergence time for FGO was 1.45 s compared to 3.78 s for KF and 5.36 s for SKF. This is due to the nature of FGO using historical measurements in an iterative process that helps it mitigate the impact of PDR errors in PDR-GNSS fusion architecture. Besides this, we set a fair comparison and implemented the SKF for fusion and compared it to FGO, which still shows the FGO superior to SKF, as well as emphasizing its enhanced performance in challenging conditions that can be seen in the comparison of SKF, KF, and FGO horizontal position errors over time [see Fig. 6(c)].

Case II: Similarly, we checked the FGO improvement when the smartphone heading changes during the walking motion. In Fig. 7, we present the horizontal position errors over time for PDR, KF, SKF, and FGO, as well as the angular rate over time. From Fig. 7(a), it can be seen that the angular rate was rapidly disturbed twice, and as a result, the effect in the PDR model can be seen in Fig. 7(b) highlighted in purple color.

Moreover, the improvement in FGO performance is greater compared to the first outlier, as FGO considers the historical measurements in the graph and then optimizes the whole graph. After the first PDR outlier, we set the PDR error threshold at 5 m, the point of peak error, to evaluate the convergence time for FGO, KF, and SKF. In this window, FGO took 1.90 s, KF 1.68 s, and SKF 1.70 s to converge post the first outlier. In the second outlier, FGO's convergence was slightly faster than SKF and KF. With a 7-m PDR error threshold, FGO converged in 3.14 s, KF in 4.20 s, and SKF in 4.24 s. It becomes more stable when it utilizes more historical measurements and optimizes the trajectory compared to KF and SKF, which works only on the last measurement and whose performance has degraded in the presence of second PDR outliers.

Case III: Similarly, in Fig. 8, when the smartphone experienced shaking, both step detection and heading were adversely affected, leading to inaccurate readings and high PDR horizontal errors over time. Following the smartphone shake, the PDR lost its trajectory and produced elevated errors that can be easily seen in Fig. 8(b). Despite encountering these challenges in the PDR block, the fused output generated by FGO demonstrated significantly improved results, effectively reducing errors and mitigating the impact of PDR errors. As highlighted in Fig. 8 with a purple color, it is noteworthy

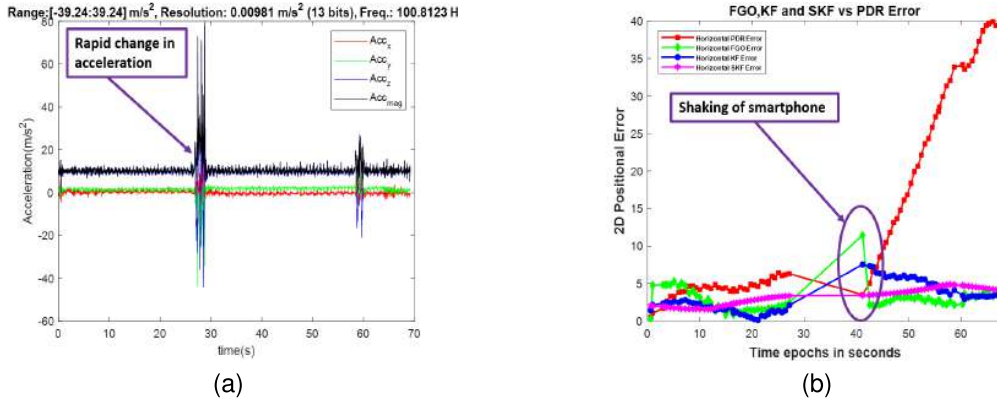


Fig. 8. (a) Rapid change in acceleration when the smartphone was shaken. (b) Comparison of horizontal position errors from PDR, KF, SKF, and FGO with smartphone shaking (highlighted in purple).

that during the shaking event, FGO's errors initially increased. However, immediately after the shaking, FGO swiftly reduced the errors, surpassing KF in error reduction. To determine the stable convergence time for FGO, KF, and SKF following the PDR outlier, we set the PDR error threshold at 11 m, which is when the errors peaked. Within this time window, the convergence time for FGO to address the PDR effect was 2.57 s, while, for KF, it was 5.58 s, and for SKF, it was 3.63 s. This observation underscores FGO's superiority over KF and SKF, even in the face of PDR errors.

The general implementation of the PDR model does not fit well with different carrying modes of smartphones. From the results, it can be seen that a disturbance of short duration in the smartphone's carrying style while walking produces a short-term and higher error in the PDR output, and eventually, the fusion methods produce lower errors while mitigating the bad performance of PDR models. However, the FGO seems like a better option compared to KF because it has fewer errors. Moreover, we have seen that FGO is more robust in mitigating the effect of PDR outliers in the same way as it has been analyzed that it mitigates the impact of GNSS outliers in the INS-GNSS fusion by [19]. Since FGO utilizes both current and previous measurements iteratively, each time a new measurement becomes available, FGO optimizes the entire factor graph. Therefore, it produces better output than SKF and KF. On the other hand, SKF also incorporates historical measurements. However, it is less robust compared to FGO due to its noniterative optimization approach, whereas KF only estimates based on the current and last measurements. To check the overall performance of FGO under the SHE PDR errors, the cumulative distribution function (cdf) and box plots of horizontal positioning errors obtained from three methods have been computed, and based on these analyses, FGO seems to be a better option than SKF and KF, which can be seen in Figs. 9 and 10. Moreover, from all three cases, it can be noted that FGO has a quicker convergence time to stability compared to KF and SKF.

B. Analysis of Improvement in FGO Versus Different Pedestrians and Motions

1) *FGO Improvement Over Motions*: It becomes necessary to assess the impact of these types of motions on the performance of FGO. The positioning performance of FGO

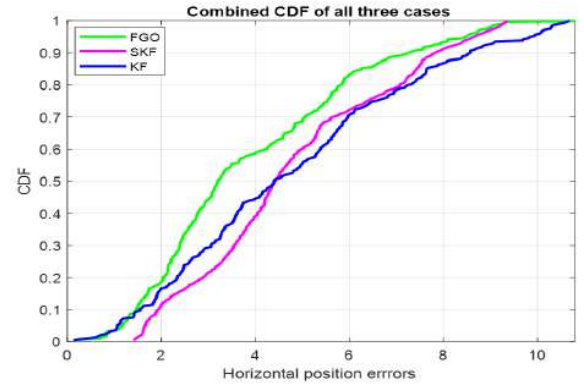


Fig. 9. Overall cdf of horizontal positioning errors from all three cases.

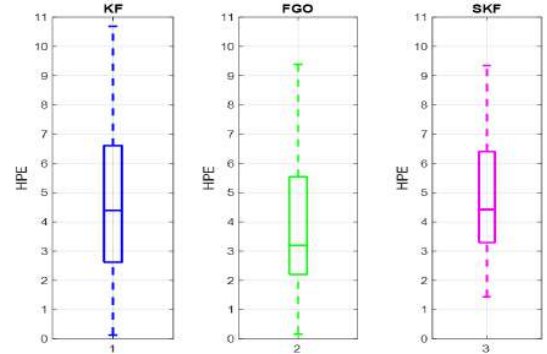


Fig. 10. Overall box plots from three cases.

over these two motions of the different pedestrians is detailed in Table II and visually represented in the box-and-whisker plot in Fig. 11. From the results, it can be noted that, overall, the performance of FGO is better than that of KF. In both motions, FGO gets lower errors compared to KF. We can also see that the results of walking motion are better compared to running motion, and this is because the PDR model is better adapted to that type of motion. The improvement of FGO is quite consistent over the two motions and different statistics Table II: around 25% for the mean error, around 30% for the median error, and around 25% for the 90th percentiles. However, there is an improvement in the standard deviation for walking motion of only around 6%, but we can see a negative improvement in running motion, which means that

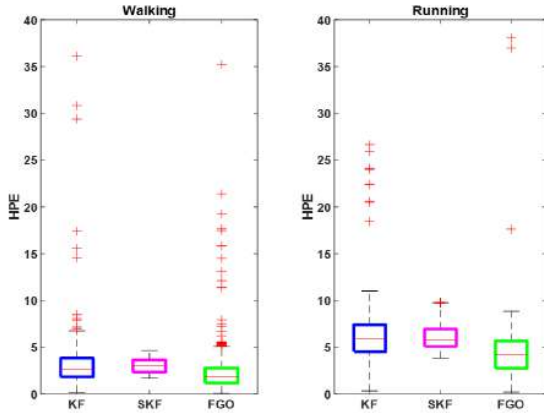


Fig. 11. Box plots illustrating the performance distribution of FGO (green box), SKF (magenta box), and KF (blue box) in PDR-GNSS fusion for walking and running motions.

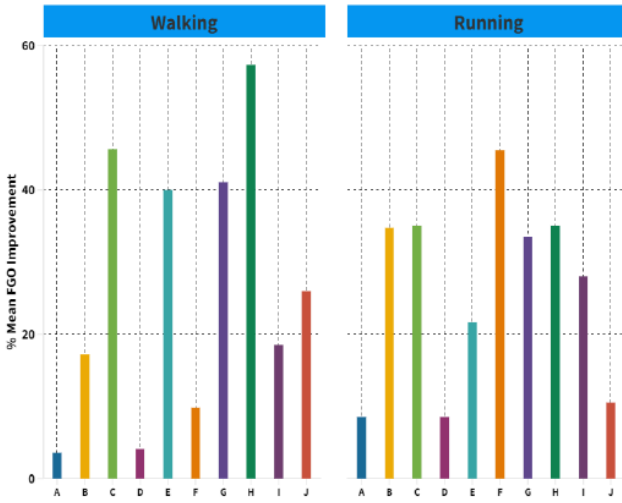


Fig. 12. Comparative analysis of 2-D mean position error via FGO versus KF for walking and running for different pedestrians (A-J).

the dispersion of the errors seems similar for both fusion methods. This is due to the diverse involvement of pedestrians, variations in gait cycles, and differing heights, all of which contribute to the variability observed in the FGO standard deviation for running motion.

2) *FGO Improvement Over Users*: Human gait dynamics differ due to differences in walking pattern, gender, and height, and the performance of individual PDR blocks is influenced by these aspects. As mentioned earlier, developing a flexible step detector, step length estimator, and heading estimator for diverse pedestrians poses a significant challenge. Consequently, no matter how diverse users' walking patterns and attributes align with the implemented PDR models, the standard deviation in the preceding boxplot did not improve for running motion. We can see in Fig. 12 that the actual improvement of FGO varies a lot for different users.

3) *General FGO Improvement*: To characterize the overall improvement of FGO over KF, we compare their 2-D mean position errors while taking into consideration the performance of PDR for each individual and motion type. However, the PDR's performance is determined by the individual performance of its main components: the step detector, the step length estimator, and the step heading estimator. Since we

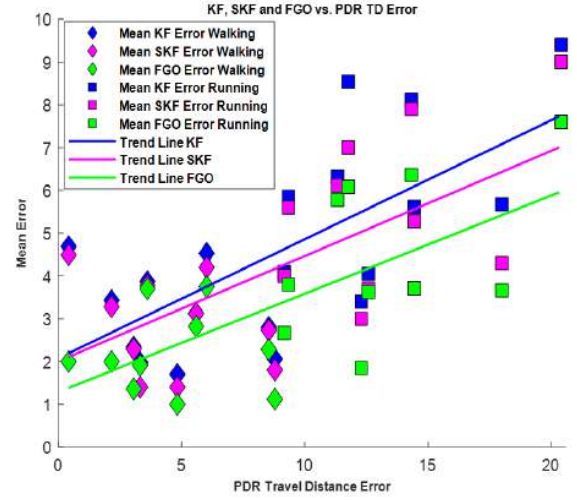


Fig. 13. Comparison of 2-D mean positioning errors between KF, SKF, and FGO with PDR travel distance error. The KF performance is depicted by the blue trend line and SKF by the magenta line, while the FGO performance is represented by the green trend line.

do not have the ground truth of these individual components, we considered the PDR travel distance as a proxy of it. It can be seen the individual mean error points and two regression lines from both fusion algorithms in Fig. 13. These mean error points are categorized by the type of motion and fusion algorithm.

From Fig. 13, it can be observed that FGO consistently shows lower mean error values compared to KF for each pedestrian and both motion types, regardless of the PDR performance value. Moreover, it can also be seen that the FGO regression line is more flatter than KF. However, the difference between the two regression lines increases as the PDR performance gets worse. We can measure that with the slope of the lines by $y = mx + b$, which gives us the slope for the KF regression line, which is 0.27823, and the FGO trend line is 0.22964. These results highlight that FGO gets lower errors and is somewhat more robust against bad PDR performances.

It is difficult to compare the obtained results to the ones reported by prior studies because of the different types of experiments. The study in [10] evaluated FGO for PDR-GNSS fusion with single walking motion data in the open sky environment. Meanwhile, a study [11] examined nine different datasets in urban environments with different smartphone positionings, such as box, pocket, and handheld. Our experiments are different from those presented in these studies as we analyzed the performance of FGO against KF and SKF under different PDR errors arising from transient variations in the way a smartphone is carried and due to different motions, heights, and gender (walking and running, and ten different pedestrians). Nevertheless, the trends and behaviors of FGO are coherent in all articles.

In FGO, all historical information is considered during the optimization, and the overall performances are improved gradually, exhibiting the capability to effectively mitigate errors associated with PDR data. As a result, despite the bad performance of the PDR block, FGO consistently outperformed KF. Implementing real PDR systems that

TABLE II
FGO IMPROVEMENT AGAINST KF AND SKF FOR PDR-GNSS FUSION ARCHITECTURE

Motion	Mean				Median			
	KF	FGO	SKF	% (KF and SKF)	KF	FGO	SKF	% (KF and SKF)
Walking	3.0325	2.2434	3.0120	26.06 and 25.00	2.6483	1.8479	3.0710	30.25 and 39.82
Running	6.1233	4.6483	6.0011	24.04 and 22.30	5.9661	4.1786	5.5001	30.00 and 25.40
Motion	STD				90th Percentile			
	KF	FGO	SKF	% (KF and SKF)	KF	FGO	SKF	% (KF and SKF)
Walking	2.3984	2.2655	0.674	5.56 and - 70.00	5.0441	3.9103	3.878	22.47 and - 0.01
Running	2.9973	3.1459	1.350	-4.78 and -57.00	9.2437	6.8397	8.4200	25.99 and 18.00

generalize well over different pedestrians, FGO appears to be a superior choice compared to KF.

VI. CONCLUSION AND FUTURE WORK

The performance of FGO has been investigated for the GNSS outliers in INS-GNSS fusion systems. However, its performance in the PDR + GNSS fusion architecture in a pedestrian navigation context was not investigated under the different PDR outliers. Therefore, this article provides a thorough comparison of FGO over KF for PDR-GNSS fusion in smartphones, taking into account the impact of PDR performance in the fusion using a real dataset collected at the football field. In this research, PDR errors are comprised of two main sources: on the one hand, the transients in the way the smartphone is carried; on the other hand, the different gait dynamics of pedestrians. Since FGO is well known for mitigating the impact of GNSS outliers in INS-GNSS fusion architectures, similarly, it has been observed that FGO mitigates the impact of the PDR errors formed by SHEs and CLEs in the PDR-GNSS fusion architectures. Based on the analysis findings, the enhanced performance of FGO compared to KF can be attributed to multiple iterations and the utilization of historical measurements. Opting for FGO seems more favorable when deploying practical PDR systems designed to work effectively across diverse pedestrians.

Despite FGO's effective performance in mitigating the impact of PDR errors in smartphone-based PDR-GNSS fusion, its batch-style fusion is impractical for real-time applications. Incorporating historical measurements raises computation costs, and considering that pedestrians have a limited pace, there is no necessity to involve all measurements in the fusion process. Instead, it can be implemented with a window of a few previous measurements. Future work could focus on the following aspects:

- 1) impact of different window sizes on the improvements of FGO over traditional fusion algorithms;
- 2) the tradeoff between the accuracy and computational load of FGO.

Overall, our assessment leads us to conclude that FGO-based sensor fusion holds significant promise as a prospective alternative to KF-based approaches in the forthcoming decades.

REFERENCES

- [1] F. Zangenehnejad and Y. Gao, "GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives," *Satell. Navigat.*, vol. 2, no. 1, pp. 1–23, Nov. 2021.
- [2] P. D. Groves, "Principles of GNSS, inertial, and multisensor integrated navigation systems, 2nd edition [book review]," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 30, no. 2, pp. 26–27, Feb. 2015.
- [3] W. Yan, L. Bastos, A. Magalhães, Y. Zhang, and A. Wang, "Assessing Android smartphone based GNSS positioning accuracy," in *Proc. China Satell. Navigat. Conf. (CSNC)*. Singapore: Springer, 2020, pp. 144–153.
- [4] J. Hu, D. Yi, and S. Bisnath, "A comprehensive analysis of smartphone GNSS range errors in realistic environments," *Sensors*, vol. 23, no. 3, p. 1631, Feb. 2023.
- [5] U. Robustelli, V. Baiocchi, and G. Pugliano, "Assessment of dual frequency GNSS observations from a Xiaomi Mi 8 Android smartphone and positioning performance analysis," *Electronics*, vol. 8, no. 1, p. 91, Jan. 2019.
- [6] G. Falco, M. Pini, and G. Marucco, "Loose and tight GNSS/INS integrations: Comparison of performance assessed in real urban scenarios," *Sensors*, vol. 17, no. 2, p. 255, Jan. 2017.
- [7] A. Angrisano et al., "GNSS/INS integration methods," Ph.D. thesis, Dipartimento di Scienze Applicate, Università degli Studi di Napoli PARTHENOPE, Naples, Italy, 2010, vol. 21.
- [8] J. A. Farrell and J. Wendel, "GNSS/INS integration," in *Springer Handbook of Global Navigation Satellite Systems*. Cham, Switzerland: Springer, 2017, pp. 811–840.
- [9] D. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, vol. 17. Edison, NJ, USA: IET, 2004.
- [10] C. Jiang et al., "Implementation and performance analysis of the PDR/GNSS integration on a smartphone," *GPS Solutions*, vol. 26, no. 3, p. 81, Jul. 2022.
- [11] C. Jiang et al., "Smartphone PDR/GNSS integration via factor graph optimization for pedestrian navigation," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 8504112, doi: [10.1109/TIM.2022.3186082](https://doi.org/10.1109/TIM.2022.3186082).
- [12] Y. Zhong, W. Wen, and L.-T. Hsu, "Trajectory smoothing using GNSS/PDR integration via factor graph optimization in urban canyons," 2022, *arXiv:2212.14264*.
- [13] A. Rehman, H. Shahid, M. A. Afzal, and H. M. A. Bhatti, "Accurate and direct GNSS/PDR integration using extended Kalman filter for pedestrian smartphone navigation," *Gyroscope Navigat.*, vol. 11, no. 2, pp. 124–137, Apr. 2020.
- [14] H. Lan, C. Yu, and N. El-Sheimy, "An integrated PDR/GNSS pedestrian navigation system," in *Proc. China Satell. Navigat. Conf. (CSNC)*, vol. 3. Berlin, Germany: Springer, 2015, pp. 677–690.
- [15] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Found. Trends Robot.*, vol. 6, nos. 1–2, pp. 1–139, 2017.
- [16] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.
- [17] W. Li, X. Cui, and M. Lu, "A robust graph optimization realization of tightly coupled GNSS/INS integrated navigation system for urban vehicles," *Tsinghua Sci. Technol.*, vol. 23, no. 6, pp. 724–732, Dec. 2018.
- [18] W. Wen, X. Bai, Y. C. Kan, and L.-T. Hsu, "Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10651–10662, Nov. 2019.
- [19] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "Comparison of extended Kalman filter and factor graph optimization for GNSS/INS integrated navigation system," 2020.
- [20] W. Wen, Y. C. Kan, and L.-T. Hsu, "Performance comparison of GNSS/INS integrations based on EKF and factor graph optimization," in *Proc. 32nd Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, Oct. 2019, pp. 3019–3032.

- [21] Q. Li, L. Zhang, and X. Wang, "Loosely coupled GNSS/INS integration based on factor graph and aided by ARIMA model," *IEEE Sensors J.*, vol. 21, no. 21, pp. 24379–24387, Nov. 2021.
- [22] S. Xin, X. Wang, J. Zhang, K. Zhou, and Y. Chen, "A comparative study of factor graph optimization-based and extended Kalman filter-based PPP-B2b/INS integrated navigation," *Remote Sens.*, vol. 15, no. 21, p. 5144, Oct. 2023.
- [23] N. Boguspayev, D. Akhmedov, A. Raskaliyev, A. Kim, and A. Sukhenko, "A comprehensive review of GNSS/INS integration techniques for land and air vehicle applications," *Appl. Sci.*, vol. 13, no. 8, p. 4819, Apr. 2023.
- [24] C. Jiang et al., "Cooperative smartphone GNSS/PDR for pedestrian navigation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 6, pp. 2301–2305, Jun. 2023.
- [25] C. Jiang, Y. Chen, C. Chen, and J. Hyppä, "Walking gaits aided mobile GNSS for pedestrian navigation in urban areas," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8499–8510, Mar. 2024.
- [26] A. Hussain and L. E. Díez, "Comparison of FGO and KF for PDR-GNSS fusion on smartphone for diverse pedestrian movements," in *Proc. 13th Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2023, pp. 1–6.
- [27] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA J.*, vol. 3, no. 8, pp. 1445–1450, Aug. 1965.
- [28] A. Rehman et al., "PDR/GNSS fusion algorithm based on joint heading estimation," in *Proc. China Satell. Navigat. Conf. (CSNC)*, vol. 1. Singapore: Springer, 2019, pp. 326–339.
- [29] X. Tao, X. Zhang, F. Zhu, F. Wang, and W. Teng, "Precise displacement estimation from time-differenced carrier phase to improve PDR performance," *IEEE Sensors J.*, vol. 18, no. 20, pp. 8238–8246, Oct. 2018.
- [30] R. Zhang, J. Mi, J. Li, and Q. Wang, "A continuous PDR and GNSS fusing algorithm for smartphone positioning," *Remote Sens.*, vol. 14, no. 20, p. 5171, Oct. 2022.
- [31] X. Wu, B. Xiao, C. Wu, Y. Guo, and L. Li, "Factor graph based navigation and positioning for control system design: A review," *Chin. J. Aeronaut.*, vol. 35, no. 5, pp. 25–39, May 2022.
- [32] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. GT-RIM-CP&R-2012-002, 2012.
- [33] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robot. Auto. Syst.*, vol. 61, no. 8, pp. 721–738, Aug. 2013.
- [34] D. Chen and G. X. Gao, "Probabilistic graphical fusion of LiDAR, GPS, and 3D building maps for urban UAV navigation," *NAVIGATION*, vol. 66, no. 1, pp. 151–168, Jan. 2019.
- [35] T. Pfeifer and P. Protzel, "Expectation-maximization for adaptive mixture models in graph optimization," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3151–3157.
- [36] S. Bhamidipati and G. X. Gao, "Multiple GPS fault detection and isolation using a graph-SLAM framework," in *Proc. Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2018, pp. 2672–2681.
- [37] J. Huang and F. Wu, "A study on the use of graph signal processing techniques for satellite-based navigation systems," in *Proc. Int. Tech. Meeting Inst. Navigat.*, 2016, pp. 448–455.
- [38] R. M. Watson and J. N. Gross, "Evaluation of kinematic precise point positioning convergence with an incremental graph optimizer," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2018, pp. 589–596.
- [39] G. Guo, R. Chen, X. Niu, K. Yan, S. Xu, and L. Chen, "Factor graph framework for smartphone indoor localization: Integrating data-driven PDR and Wi-Fi RTT/RSS ranging," *IEEE Sensors J.*, vol. 23, no. 11, pp. 12346–12354, Jun. 2023.
- [40] R. M. Watson and J. N. Gross, "Robust navigation in GNSS degraded environment using graph optimization," in *Proc. 30th Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, 2017, pp. 2906–2918.
- [41] T. Pfeifer and P. Protzel, "Robust sensor fusion with self-tuning mixture models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3678–3685.
- [42] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "It is time for factor graph optimization for GNSS/INS integration: Comparison between FGO and EKF," 2020, *arXiv:2004.10572*.
- [43] W. Wen, L.-T. Hsu, and G. Zhang, "Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong," *Sensors*, vol. 18, no. 11, p. 3928, Nov. 2018.
- [44] X. Li, H. Yu, X. Wang, S. Li, Y. Zhou, and H. Chang, "FGO-GIL: Factor graph optimization-based GNSS RTK/INS/LiDAR tightly coupled integration for precise and continuous navigation," *IEEE Sensors J.*, vol. 23, no. 13, pp. 14534–14548, Jul. 2023.
- [45] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," Analog Devices, Wilmington, MA, USA, Appl. Note AN-602, 2002, vol. 2, no. 2, pp. 1–6.
- [46] H. P. Gavin, "The Levenberg–Marquardt algorithm for nonlinear least squares curve-fitting problems," Dept. Civil Environ. Eng., Duke Univ., Durham, NC, USA, 2019, vol. 19.
- [47] A. R. Jiménez, F. Seco, and J. Torres-Sospedra, "Tools for smartphone multi-sensor data registration and GT mapping for positioning applications," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, 2019, pp. 1–8.



Amjad Hussain Magsi received the B.E. degree in electronic engineering and the master's degree in electronic systems engineering from the Mehran University of Engineering Technology, Jamshoro, Pakistan, in 2017 and 2022, respectively. He is currently pursuing the Ph.D. degree in engineering for the information society and sustainable development under the supervision of Luis Enrique Díez Blanco with the University of Deusto, Bilbao, Spain.

During his bachelor's, he was an Erasmus Scholar at the University of Limerick, Limerick, Ireland.

From 2019 to 2022, he worked as a Lab Engineer at Dawood University of Engineering and Technology, Karachi, Pakistan. He is also working at the Deusto Smart Mobility Research Team, University of Deusto. His research interests include data fusion for developing location-based services, smartphone positioning, factor graph optimization, and machine learning theory and algorithms for localization and intelligent environments.



Luis Enrique Díez received the bachelor's degree in telecommunications engineering from the University of Deusto, Bilbao, Spain, in 2005, the master's degree in communications technologies and systems from the Telecommunications Engineering School, Polytechnic University of Madrid, Madrid, Spain, in 2012, and the Ph.D. degree in engineering from the University of Deusto in 2019.

From 2005 to 2011, he worked at Everis, Madrid, as a Senior IT Consultant. From 2013 to 2014, he was with the SOFTLAB Research Group, Carlos

III University of Madrid (UC3M), Madrid, as a Research Support Technician. From 2014 to 2022 he worked as a Researcher at DeustoTech, the Research Institute of the Engineering Faculty, University of Deusto. He currently works as an Assistant Professor at the Faculty of Engineering, University of Deusto, and he is a member of the research group "Deusto Smart Mobility." He has worked (leading some of them) on more than ten regional, national, and international research projects and contracts. He has co-authored 15 research manuscripts published in international journals and more than ten communications at international congresses. His research interests include signal processing and data fusion for the development of location-based services, ambient assisted living, and intelligent environments.

Chapter 4

Case Study II: Sliding Window Techniques for Efficient FGO

4.1 Introduction

Building on the insights from the first two publications, the third study investigates the optimization of FGO performance through advanced windowing strategies in PDR–GNSS fusion systems. While previous work demonstrated the accuracy benefits of FGO over Kalman-based methods, its computational load and scalability remain challenges for real-time applications, particularly on smartphones.

This study evaluates and compares two distinct windowing strategies for improving the computational feasibility of FGO while preserving localization accuracy:

1. **Naïve Windowed FGO (NW-FGO):** A basic sliding window approach that discards old states without retaining their information.
2. **Marginalized Windowed FGO (MW-FGO):** An enhanced sliding window approach that marginalizes out older states while preserving their information through prior factors.

Both strategies were compared against a full batch FGO to assess trade-offs between performance and efficiency.

Two main experimental cases were considered:

- **GNSS-driven fusion performance:** Assessing algorithm robustness in environments with multipath effects and degraded satellite visibility.
- **PDR-driven fusion performance:** Assessing robustness under intentional PDR errors, with two error types:
 - **Short and High Errors (SHE):** Brief, intense disruptions such as shaking, swinging, or rotating the device.
 - **Continuous and Low Errors (CLE):** Low-intensity but persistent drift caused by gait variability and user differences (e.g., gender, height, walking style).

Experiments were conducted using a dual-frequency smartphone in realistic pedestrian motion scenarios. Each method was assessed in terms of horizontal positioning accuracy, drift behavior, convergence time, and computational cost.

4.2 Key Findings

4.2.1 GNSS-Driven Fusion Performance

- MW-FGO consistently outperformed NW-FGO in multipath-affected GNSS conditions, maintaining accuracy close to full batch FGO.
- NW-FGO reduced computational load but exhibited larger drift due to the loss of historical information.
- MW-FGO offered a balanced trade-off between computational efficiency and trajectory stability.

4.2.2 PDR-Driven Fusion Performance

Short and High Errors (SHE)

- MW-FGO recovered more rapidly from sudden large PDR disturbances, minimizing trajectory deviations.
- NW-FGO suffered from greater performance degradation under SHE due to lack of informative priors.

Continuous and Low Errors (CLE)

- MW-FGO effectively mitigated slow drift caused by persistent gait-related biases, producing more stable long-term trajectories.
- NW-FGO showed increasing error growth over time in CLE conditions.

4.2.3 General Observations

- Marginalized windowing preserved essential factor graph structure, enabling stable performance across GNSS and PDR cases.
- Naïve windowing improved efficiency but at the cost of accuracy in both GNSS and PDR conditions.
- Intelligent window management is critical for enabling real-time FGO deployment on resource-constrained platforms such as smartphones.

Publication P3

Published in: IEEE ACCESS

Received 5 June 2025, accepted 26 June 2025, date of publication 30 June 2025, date of current version 8 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3584664

RESEARCH ARTICLE

Performance Analysis of FGO Windowing Strategies for PDR+GNSS Fusion Architecture

AMJAD HUSSAIN MAGSI¹ AND LUIS ENRIQUE DÍEZ BLANCO²

Faculty of Engineering, University of Deusto, 48007 Bilbao, Spain

Corresponding author: Amjad Hussain Magsi (amjad.hussain@deusto.es)

This work was supported in part by the Research Training Grants Programme of the University of Deusto; and in part by the Spanish Ministry of Science, Innovation and Universities through the Projects REPIN++ and AGINPLACE under Grant RED2022-134355-T and Grant PID2023-146254OA-C44.

ABSTRACT Smartphone-based pedestrian navigation has become increasingly popular due to the widespread availability of low-cost Android devices. However, these smartphones often suffer from low positional accuracy caused by inexpensive GNSS antennas and receivers. To address these limitations, fusion mechanisms are employed, with Factor Graph Optimization (FGO) emerging as a prominent method in navigation solutions, particularly for INS+GNSS integration. While FGO has also gained popularity in Simultaneous Localization and Mapping (SLAM) applications, its reliance on batch data processing, referred to as Batch Factor Graph Optimization (BFGO) and which is used as the baseline for this study, results in significant computational overhead, making it impractical for real-time or resource-constrained scenarios. To address these challenges, two windowing-based FGO approaches are proposed: the Naive Windowing FGO (NW-FGO) and the Marginalized Windowing FGO (MW-FGO). By incorporating marginalization, MWFGO retains critical information while reducing computational costs. Both methods are evaluated under two error conditions GNSS multipath and PDR errors to assess their impact on positional accuracy and efficiency. Our findings reveal that optimal window sizes, such as 50s for GNSS errors and 30s for PDR errors, achieve accuracy comparable to BFGO while significantly reducing computational time from 2500 seconds to 38 seconds with MWFGO. Furthermore, MWFGO outperforms NW-FGO, improving mean positional accuracy by 2.18% across varying window sizes, underscoring the effectiveness of marginalization. This study demonstrates the feasibility of windowing-based FGO, particularly MWFGO, in balancing computational efficiency and accuracy. By optimizing window sizes and leveraging marginalization, we provide a robust alternative to FGO, making real-time pedestrian navigation viable in resource-constrained environments.

INDEX TERMS FGO, fusion, GNSS, marginalization, Levenberg Marquardt optimiser, PDR, outliers, pedestrian movements, sliding window, smartphone.

I. INTRODUCTION

Smartphone-based positioning has become a primary resource for pedestrians in urban environments. Smartphones are widely used for navigation and mobility because of their affordability and accessibility. However, they continue to face significant challenges in delivering accurate positioning. These limitations are primarily due to the use of low-cost,

built-in GNSS antennas and receivers, which are susceptible to various errors such as hardware biases, clock inaccuracies, and frequent cycle slips. These issues are particularly prominent in urban environments where GNSS signals are obstructed, leading to challenges like multipath effects and Non-Line-of-Sight (NLOS) conditions [1], [2].

Relying solely on smartphone GNSS may not be suitable for all navigation applications, particularly those requiring high positional accuracy. To address this limitation, alternative positioning methods and multi-sensor fusion techniques

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei¹.

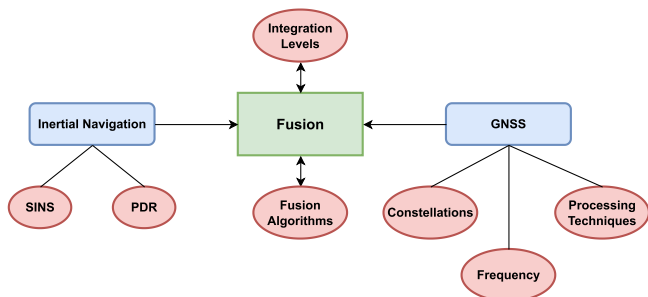


FIGURE 1. Inertial+GNSS fusion architecture mechanism.

are being explored to enhance GNSS performance [3]. Among these, the fusion of inertial sensors with GNSS has become widely recognized due to their complementary characteristics [4]. They offer low short-term errors and are not susceptible to signal blockage caused by multipath interference or NLOS conditions. Furthermore, they do not necessitate additional infrastructure. This makes inertial + GNSS a well-known and popular fusion architecture [5], [6], [7]. This fusion architecture involves several steps and can be implemented in many ways, as shown in Fig. 1. Moreover, the high noise of the MEMS (Micro-Electro-Mechanical Systems)-based IMU (Inertial Measurement Unit) built into smartphones makes the SINS (Strapdown Inertial Navigation System) impractical due to the fast error accumulation over time. In addition, smartphones carrying modes do not allow the implementation of techniques such as ZUPT to reduce the error rate [8]. Consequently, researchers have explored the adoption of a Pedestrian Dead Reckoning (PDR)+GNSS fusion architecture [9], [10], [11], [12], [13]. This approach addresses the issue of error accumulation by utilizing PDR, which takes into account the number of steps taken to estimate position and compensate for errors accordingly [14], [15]. PDR is usually based on step detection, step length, and heading estimation methods that somehow alleviate IMU errors.

Historically, the Kalman Filter (KF) has served as the primary fusion algorithm for the PDR-GNSS fusion architecture. However, achieving high accuracy using KF and its variant models has been challenging due to the linear first-order Markov dynamic model, Gaussian noise assumptions, and the fact that it considers only the latest available measurement. Instead, the newly proposed Factor Graph Optimization (FGO) [16] is a well-liked approach for the multi-sensor fusion concept. Factor graphs are graphical models used to effectively address complex estimation problems like simultaneous localization and mapping (SLAM) [17]. BFGO optimizes state estimates through iterative solutions, treating state estimation as a nonlinear least squares problem. It employs historical measurements and iterative solvers like Gauss-Newton or Levenberg-Marquardt (LM) to minimize the cost function, achieving optimal estimations. Consequently, there is a growing trend to utilize FGO as an alternative to KF for both GNSS processing and

inertial-GNSS fusion. Researchers [18], [19], [20], [21], [22], [23], [24], [25] fused INS and GNSS using FGO and compared its performance against the traditional KF. Their findings highlighted FGO's notable capability to mitigate GNSS outliers, particularly in scenarios characterized by urban canyons or limited visibility of satellites during vehicular testing.

BFGO has been widely employed in pedestrian navigation due to its accuracy and robustness compared to the Kalman Filter (KF). Jiang et al. [9] demonstrated FGO's advantages over KF in single-user walking-motion scenarios, while Jiang et al. [10] expanded this work by incorporating position and distance constraints to improve performance across different smartphone placements (backpack, handheld, and box-mounted). Further extending batch FGO, Jiang et al. [26] proposed a cooperative navigation framework integrating range-based constraints, enhancing GNSS-PDR fusion in multi-agent systems. In challenging environments, Zhong et al. [11] introduced a tightly coupled FGO approach utilizing GNSS pseudorange, Doppler, and PDR constraints, along with a smoothness-driven motion model to improve trajectory accuracy under GNSS multipath. Similarly, Jiang et al. [27] refined FGO integration by incorporating pedestrian step size as a constraint and applying fuzzy adaptive techniques to suppress GNSS pseudorange errors. Our previous studies [12], [28] further evaluated FGO under various motion conditions and PDR error scenarios, demonstrating its superior performance over KF. However, we observed a decline in accuracy during running motions due to step detection and step length estimation errors. Additionally, we analyzed specific PDR error types (SHE and CLE) and showed that FGO effectively mitigated these errors, aligning with its success in handling GNSS outliers [20]. These studies collectively highlight the effectiveness of batch FGO in pedestrian navigation but also underscore its computational burden, motivating the need for more efficient alternatives.

Unlike vehicular platforms, pedestrian navigation presents unique motion dynamics, such as slower, irregular movements and varying device placements (e.g., hand-held, pocket, or bag). This makes BFGO unnecessary for pedestrian navigation, further highlighting its limitations in such scenarios. Although batch optimization may outperform other fusion algorithms in terms of accuracy, its high computational cost makes it impractical for real-time pedestrian positioning. To address this, sliding-window techniques are preferred to limit the number of factors and poses in the factor graph, reducing computational complexity. This is demonstrated by Wen et al. [21] that sliding-window-based FGO using the Incremental Smoothing and Mapping (iSAM2) optimizer for INS+GNSS fusion for vehicle navigation. Similarly, Dong et al. [29] applied windowing FGO with the Levenberg-Marquardt (LM) optimizer for underwater anomaly detection.

In our previous study [28], we conducted a performance comparison of FGO with both KF and Smoothed Kalman

Filter (SKF) under various PDR error conditions. Therefore, the focus of this work is specifically on evaluating the performance variations within the FGO framework itself, particularly with respect to different window sizes in a smartphone-based PDR+GNSS fusion context. However, to the best of our knowledge, no study has yet analyzed different sliding window techniques for PDR+GNSS fusion on smartphones for pedestrian navigation, especially under PDR and GNSS error conditions. This gap presents an opportunity to explore and evaluate the effectiveness of such techniques in smartphone-based pedestrian navigation.

To address these challenges, this study introduces two different windowing-based approaches as mentioned below:

- **Naive-based windowing FGO (NWFGO):** This approach processes a fixed-size window of recent position and GNSS factors, optimizing the graph independently at each step. Older data outside the window is discarded, and the graph is reinitialized, ensuring computational efficiency but potentially losing long-term consistency.
- **Marginalized-based windowing FGO (MWFGO):** This approach maintains a sliding optimization window defined by a fixed temporal lag. While older poses beyond the lag are marginalized to preserve efficiency, their last optimized values remain fixed, ensuring a balance between computational cost and maintaining global consistency.

Our analysis emphasizes positional accuracy across different window sizes exploring the trade-off between window size, FGO accuracy, and computational load. To the best of our knowledge, no prior research has investigated different sliding window-based FGO techniques under PDR errors within PDR+GNSS fusion architectures. Existing studies primarily focus on INS+GNSS integration and address only GNSS errors, whereas PDR+GNSS fusion introduces distinct types of PDR errors that require dedicated analysis.

The main contributions of this research are as follows.

- **Evaluation of NWFGO and MWFGO Sliding Window Approaches:** We implemented and analyzed the NWFGO and MWFGO-based sliding window approaches to compare their performance for PDR+GNSS fusion under varying window sizes.
- **Trade-Off Analysis Between Positional Accuracy and Computational Load:** We carried out further analysis of the trade-off between positional accuracy and computational load for these two windowing approaches. This paper highlights how window size selection affects FGO efficiency.
- **Performance Assessment of Windowing-Based FGO under PDR Errors:** We evaluated the NWFGO and MWFGO approaches in PDR Error scenarios:
 - PDR-specific scenarios, examining the impact of Short and High Errors (SHE) and Continuous and Low errors (CLE). Comparative analysis was performed for both cases.

The remainder of this paper is organized as follows. Section II outlines the methodology used to develop the PDR

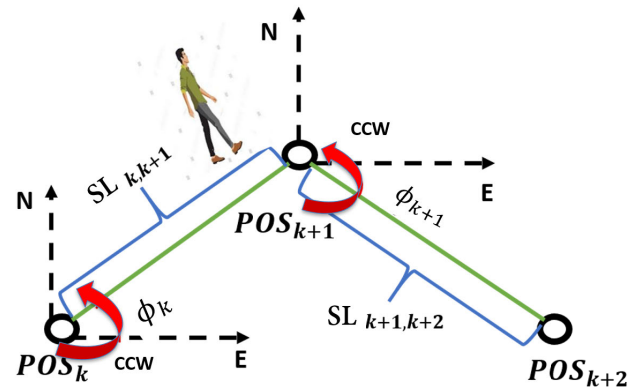


FIGURE 2. PDR mechanism illustrating heading estimation and step-wise position updates in the ENU frame. The heading angles ϕ_k and ϕ_{k+1} are measured counterclockwise from the East axis at step epochs k and $k+1$, respectively. POS_k , POS_{k+1} , and POS_{k+2} represent the pedestrian's estimated positions, while $SL_{k,k+1}$ and $SL_{k+1,k+2}$ denote the corresponding step lengths.

mechanism and the implementation of BFGO, NWFGO, and MWFGO-based sliding window FGO. Section III describes the experimental setup and evaluation process. Section IV presents and discusses the results. Lastly, Section V concludes the paper by summarizing the key findings and suggesting directions for future research.

II. METHODOLOGY

This section outlines the methodological approach employed in the development of PDR and its fusion with GNSS using BFGO, along with two sliding window techniques: NWFGO and MWFGO.

A. PDR DEVELOPMENT

PDR determines pedestrian movements by analyzing step detection, step length, and heading estimation. Its working mechanism, shown in Fig. 2 can be expressed mathematically as follows:

$$Pos_{k+1} = Pos_k + \Delta Pos_{k|k+1} \quad (1)$$

In this equation, Pos_{k+1} and Pos_k are the positions of steps at $k+1$ and k epochs, respectively. $\Delta Pos_{k|k+1}$ shows the positional change from k to $k+1$, which can be expressed as follows in the ENU reference system:

$$\Delta Pos_{k|k+1} = \begin{bmatrix} \Delta Pos_{k|k+1}^E \\ \Delta Pos_{k|k+1}^N \end{bmatrix} = \begin{bmatrix} SL_{k|k+1} \cdot \cos(\phi) \\ SL_{k|k+1} \cdot \sin(\phi) \end{bmatrix} \quad (2)$$

Here, ϕ is representing the heading angle, whereas $SL_{k|k+1}$ is the step length from step k to step $k+1$. The mechanism of the obtained PDR system from the smartphone's inertial sensors is presented in Fig. 3.

Step detection was performed based on acceleration magnitude thresholding, following the method described by Jimenez et al. [30]. The magnitude of the triaxial accelerometer serves as the input signal for identifying peaks

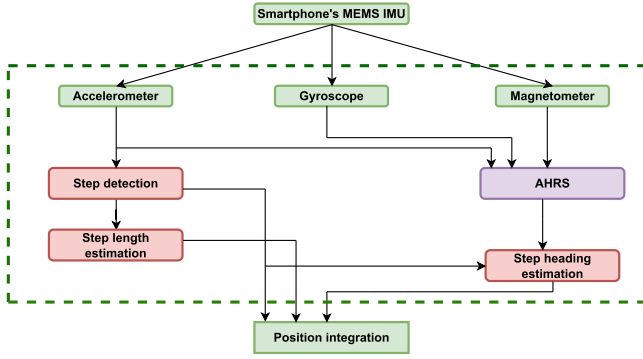


FIGURE 3. Smartphone-based PDR.

corresponding to steps, as shown in equation 3.

$$Acc_{mag_k} = \sqrt{Acc_{x_k}^2 + Acc_{y_k}^2 + Acc_{z_k}^2} \quad (3)$$

To estimate step length, Weinberg's algorithm [31] has been used, which posits that each pedestrian step corresponds to a local minimum in the acceleration signal. By identifying these local minima and measuring the time intervals between them, we could estimate the step length.

$$SL_{k|k+1} = C \cdot [\max(Acc_{mag_k}) - \min(Acc_{mag_k})]^{1/4} \quad (4)$$

Here, $SL_{k|k+1}$ denotes the estimated step length between two consecutive acceleration peaks, and C is a constant scaling factor

For step heading estimation, the attitude and heading reference system (AHRS) integrated into the smartphone has been used at each detected step epoch. The AHRS fuses data from the gyroscope, accelerometer, and magnetometer to compute the smartphone's orientation. The yaw angle extracted from this orientation is directly used as the pedestrian's heading angle in the ENU reference frame more details about the AHRS can be found here [32].

B. BATCH FGO-BASED PDR+GNSS FUSION ARCHITECTURE

As can be seen in Fig. 4, a factor graph is a type of probability model that uses a graph to represent the probability dependence of various variables. In particular, a factor graph is a bipartite graph made up of two different kinds of nodes: factor nodes, which show the conditional probabilities of a given set of state variables, and unknown state variable nodes. The undirected edges link the state variable nodes with their corresponding factor nodes. An edge exists between two nodes only if there is a functional relationship between the variable node and the factor node. It enables the decomposition of a global probability function into the product of numerous local probability functions, thereby mitigating the complexity associated with solving the global function. Fig.4 illustrates the factor graph, which in our case consists of two main factor nodes, which are PDR and GNSS factors, along with state variable nodes that are going to be estimated with the observation of PDR and GNSS.

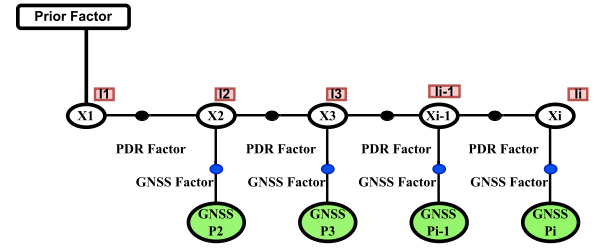


FIGURE 4. Formation of PDR and GNSS factors within BFGO mechanism.

In sensor fusion scenarios, our main goal often involves estimating a group of state variables, denoted as X , from a given set of measurements, Z . Assuming a Markov model, we can express the conditional probability density function in the following way:

$$P(X|Z) = \prod_{i=1}^k \frac{P(z_i|x_i)P(x_i|x_{i-1}, u_i)}{P(z_i)} P(x_0) \quad (5)$$

where z_i represents the measurements observed at epoch i (e.g. GNSS measurements), x_i represents the system state at i , and u_i denotes the control input (e.g., PDR measurements).

The most often used estimator for these unknown state variables X is the maximum a posterior (MAP), or MAP estimate:

$$\hat{X} = \arg \max P(X|Z) = \arg \max \prod_{i=1}^k P(z_i|x_i)P(x_i|x_{i-1}, u_i) \quad (6)$$

Within FGO-based integration, all these probabilities related to likelihood and transition are considered as the factorization of the global probability:

$$\hat{X} = \arg \max \prod_{j=1}^n f(x_j) \quad (7)$$

$$P(z_i|x_i) \propto \exp \left(-\frac{1}{2} \|h_i(x_i) - z_i\|^2 \nabla_i \right) \quad (8)$$

$$P(x_i|x_{i-1}) \propto \exp \left(-\frac{1}{2} \|\Phi_i(x_{i-1}) - x_i\|^2 \Omega_i \right) \quad (9)$$

Here, the function $\Phi_i(\cdot)$ delineates the connection between the preceding states x_{i-1} and x_i , while the function $h_i(\cdot)$ encapsulates the relationship between the state x_i and the measurement z_i . The covariance matrices are denoted by ∇_i and Ω_i .

Taking the negative log and dropping the factor $1/2$ transforms the problem into the minimization of an error function, that is, a nonlinear least-squares problem.

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \|\Phi_i(x_{i-1}) - x_i\|_{\Omega_i}^2 + \sum_{i=1}^k \|h_i(x_i) - z_i\|_{\nabla_i}^2 \right) \quad (10)$$

In our case, as shown in Fig. 4, the factors represent the GNSS measurements and the transition between states due to

the motion model and PDR measurements: Therefore:

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \left[\|e_{i+1}^{PDR}\|_{\Sigma_{i+1}^{PDR}}^2 + \|e_{i+1}^{GNSS}\|_{\Sigma_{i+1}^{GNSS}}^2 \right] \right) \quad (11)$$

PDR Factor: The PDR factor node (black dot) is constructed from the motion model derived from position data, which is obtained from the step length and heading estimation of the smartphone. In our factor graph formulation, the PDR factor plays an important role in capturing the dynamic relationship between consecutive states. This factor, denoted as e_{k+1}^{PDR} , represents the squared Euclidean norm of the PDR error vector at time step $k + 1$. It quantifies the discrepancy between the predicted state X_{k+1} and the estimated state $f(X_k)$ by considering both position and orientation components. This can be mathematically expressed as follows:

$$\begin{aligned} \|e_{k+1}^{PDR}\|_{\Sigma_{k+1}^{PDR}}^2 &= \|X_{k+1} - f(X_k)\|_{\Sigma_{k+1}^{PDR}}^2 \\ &= \left\| x_{k+1} - \left(x_k + \begin{bmatrix} SL_{t,t+1} \cdot \cos(\phi) \\ SL_{t,t+1} \cdot \sin(\phi) \end{bmatrix} \right) \right\|_{\Sigma_{k+1}^{PDR}}^2 \end{aligned} \quad (12)$$

This factor encapsulates the impact of step length ($SL_{t,t+1}$) and heading angle (ϕ) on the positional estimation, providing a comprehensive measure of the PDR-related error at each time step.

GNSS Factor: The GNSS factor node is created based on smartphone GNSS observations. The factor is represented by the squared Euclidean norm of the GNSS factor error vector at time step $k + 1$, denoted as e_{k+1}^{GNSS} . This factor captures the discrepancy between the predicted state X_{k+1} and the actual GNSS measurement $GNSS_{k+1}$. The associated information matrix, Σ_{k+1}^{GNSS} , is employed to weigh the error components, ensuring that the estimated state is adjusted by the uncertainties in GNSS measurements, which can be written as follows:

$$\|e_{k+1}^{GNSS}\|_{\Sigma_{k+1}^{GNSS}}^2 = \|X_{k+1} - GNSS_{k+1}\|_{\Sigma_{k+1}^{GNSS}}^2 \quad (13)$$

Based on the PDR factors (black circle) and GNSS factors (blue circle), the factor graph is constructed as shown in Fig. 4.

C. SLIDING WINDOW-BASED FGO PDR+GNSS FUSION ARCHITECTURE

As previously discussed, BFGO processes all data from the beginning each time optimization is performed. This approach not only requires multiple internal iterations within the LM optimizer but also significantly increases computational demands. Therefore, it is essential to implement a sliding window-based optimization strategy that restricts the number of factors considered during each optimization step, reducing both computational complexity and processing time. In this study, we explore two distinct sliding window approaches in detail to evaluate their impact on optimization efficiency and performance.

1) NAIVE-BASED SLIDING WINDOW FGO (NWFGO)

The naive-based sliding window optimization is a straightforward approach designed to handle state estimation efficiently by limiting computations to a fixed-size window of recent states. At each GNSS measurement, the factor graph is reconstructed, incorporating only the most recent N poses and their associated measurements (factors), while discarding information outside the current window. This ensures computational feasibility but it sacrifices long-term consistency since older states are excluded from optimization once they fall outside the window. In other words, no marginalization is performed in this approach.

Problem Formulation: The optimization problem for the sliding window can be formulated as:

$$\hat{X}_{k-N:k} = \arg \max P(X_{k-N:k} | Z_{k-N:k}), \quad (14)$$

where:

- $\hat{X}_{k-N:k}$ is the estimated trajectory for the most recent N poses within the sliding window,
- $Z_{k-N:k}$ represents the set of measurements (e.g., GNSS and odometry) available in the window.

Using Bayes' rule and assuming a Markovian structure for the motion dynamics, the problem simplifies to:

$$\hat{X}_{k-N:k} = \arg \max \prod_{i=k-N}^k P(z_i | x_i) P(x_i | x_{i-1}, u_i), \quad (15)$$

where:

- $P(z_i | x_i)$: Likelihood of the measurement z_i (e.g., GNSS) given the state x_i ,
- $P(x_i | x_{i-1}, u_i)$: Transition probability between consecutive poses x_{i-1} and x_i , based on odometry u_i .

Factor Graph Construction: At each inclusion of a new GNSS factor, the sliding window optimization involves constructing a factor graph containing:

- 1) **Prior Factor:** A prior factor is applied to anchor the oldest pose in the window (x_{k-N}) to maintain observability:

$$P(z_i | x_i) \sim \exp \left(-\frac{1}{2} (x_i - z_i)^\top \Sigma^{-1} (x_i - z_i) \right), \quad (16)$$

where z_i is the GNSS measurement and Σ is its noise covariance.

- 2) **GNSS Factors:** GNSS measurements are treated as priors for all poses within the window to ensure alignment with global positioning.
- 3) **PDR Position Factors:** The relative motion between consecutive poses is modeled as:

$$P(x_i | x_{i-1}) \propto \exp \left(-\frac{1}{2} \|\Phi_i(x_{i-1}) - x_i\|_{\Omega_i}^2 \right), \quad (17)$$

where $\Phi_i(x_{i-1})$ represents the predicted transformation between x_{i-1} and x_i based on PDR, and Ω_i is the corresponding information matrix.

Optimization Process: At each iteration, a new factor graph is constructed using the N most recent poses and their

associated factors (GNSS and PDR). In both batch FGO and naive windowing, the optimization process follows a similar structure. The constructed factor graph is optimized using the LM algorithm, which iteratively refines the state estimates to obtain the MAP solution. In naive windowing, only the most recent N poses are considered, ensuring a computationally efficient update while still preserving the key constraints. After optimization, the estimated trajectory is updated, with only the most recent pose retained for further evaluation.

2) MARGINALIZED BASED SLIDING WINDOW FGO (MWFGO)

The MWFGO provides a structured approach to address the state estimation problem by maintaining a fixed temporal horizon for optimization. Unlike the naive approach, MWFGO retains prior information of older states that fall outside the active lag window. These states are no longer actively optimized but are used to anchor the trajectory, ensuring both computational efficiency and long-term consistency.

Problem Formulation: The optimization problem in MWFGO can be expressed as:

$$\hat{X}_{k-T:k} = \arg \max P(X_{k-T:k} | Z_{1:k}), \quad (18)$$

where:

- $\hat{X}_{k-T:k}$ is the estimated trajectory for the most recent states within the fixed lag T ,
- $Z_{1:k}$ represents all measurements (GNSS, odometry) up to time k .

Using Bayes' rule and assuming a Markovian structure for motion dynamics, the problem simplifies to:

$$\hat{X}_{k-T:k} = \arg \max \prod_{i=k-T}^k P(z_i | x_i) P(x_i | x_{i-1}, u_i), \quad (19)$$

where:

- $P(z_i | x_i)$: Likelihood of GNSS measurements z_i at state x_i ,
- $P(x_i | x_{i-1}, u_i)$: Transition probability between consecutive states x_{i-1} and x_i , given odometry u_i .

Sliding Window Mechanism: At each new GNSS measurement, MWFGO:

- 1) Maintains a fixed lag time window (T), optimizing only the states within the lag window while marginalizing older states (x_{k-T}).
- 2) Incrementally updates the factor graph by:
 - Adding new measurements and positions,
 - Marginalizing old factors outside the time window.

Factor Graph Construction: The construction of the graph in MWFGO involves:

- 1) **Prior Factor for Oldest Pose:** A prior factor is retained for the marginalized state x_{k-T} to anchor the trajectory:

$$P(z_i | x_i) \sim \exp \left(-\frac{1}{2} (x_i - z_i)^\top \Sigma^{-1} (x_i - z_i) \right), \quad (20)$$

where z_i is the GNSS measurement, and Σ is the covariance of GNSS noise.

- 2) **GNSS Factors:** GNSS measurements for poses within the lag window are added as priors:

$$P(z_i | x_i) = \text{Likelihood of GNSS position at state } x_i. \quad (21)$$

- 3) **PDR Position Factors:** The transition probabilities between consecutive poses are modeled as:

$$P(x_i | x_{i-1}) \propto \exp \left(-\frac{1}{2} \|\Phi_i(x_{i-1}) - x_i\|_{\Omega_i}^2 \right), \quad (22)$$

where $\Phi_i(x_{i-1})$ represents the predicted transformation between x_{i-1} and x_i based on PDR, and Ω_i is the corresponding information matrix.

Optimization Process: At each new GNSS measurement, the factor graph is updated by incorporating new GNSS measurements and PDR position factors while marginalizing factors older than T seconds. The optimization is then performed using the LM algorithm to iteratively refine the state estimates and obtain the MAP solution within the lag time window. After optimization, the estimated trajectory is updated, with optimized states retained for evaluation while older states are replaced by marginalized priors.

Computational Characteristics: MWFGO balances computational efficiency and accuracy by:

- Limiting active optimization to a fixed lag window, keeping computational costs bounded,
- Retaining marginalized priors to ensure trajectory consistency over long-term operations,
- Maintaining scalability for real-time applications by incrementally updating the factor graph.

Marginalization in MWFGO

Marginalization is an essential process MWFGO ensures that the optimization is limited to the poses and factors within the lag window T , while preserving the contributions of older states through priors.

In the fixed-lag smoothing, the state X_k is split into two components:

$$X_k = \{X_{\text{active}}, X_{\text{marginalized}}\},$$

where:

- X_{active} represents the states (poses) within the lag window $[k - T, k]$,
- $X_{\text{marginalized}}$ includes states outside the lag window ($t < k - T$).

The joint probability for all states and measurements can be expressed as:

$$P(X_k | Z_k) \propto \exp \left(-\frac{1}{2} \|AX_k - b\|_{\Sigma}^2 \right),$$

where:

- A is the information matrix,
- b is the measurement vector,
- Σ is the noise covariance.

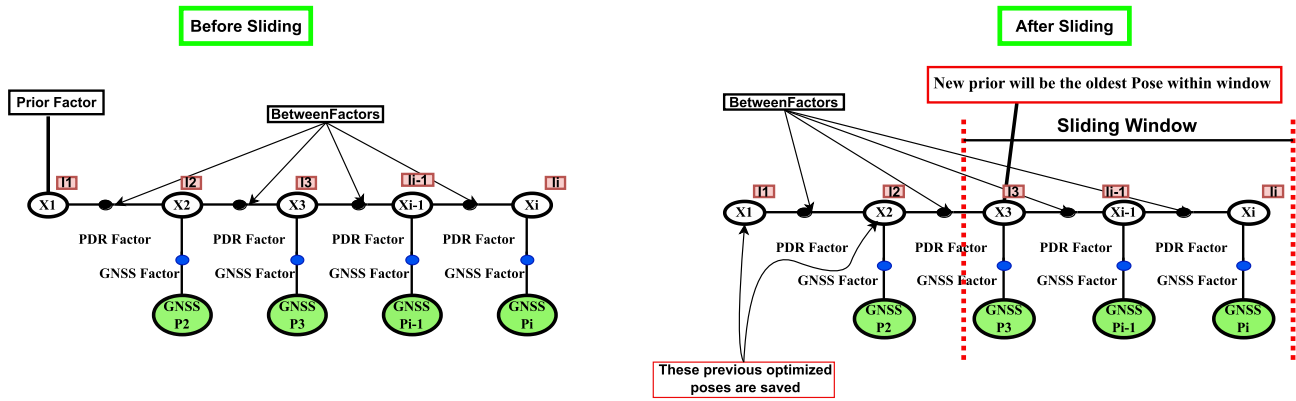


FIGURE 5. Naive-based Sliding Window FGO Mechanism.

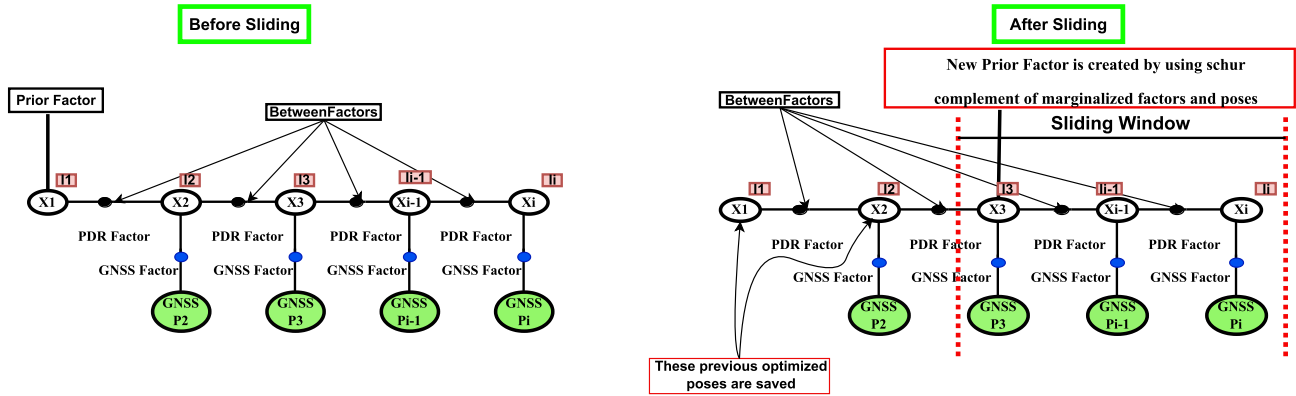


FIGURE 6. Marginalization-based Sliding Window FGO Mechanism.

Marginalization eliminates $X_{\text{marginalized}}$ by integrating it out:

$$P(X_{\text{active}}|Z_k) = \int P(X_k|Z_k) dX_{\text{marginalized}}.$$

Using the Schur complement [29], [33], this marginalization results in:

$$P(X_{\text{active}}|Z_k) \propto \exp\left(-\frac{1}{2}\|A_{\text{new}}X_{\text{active}} - b_{\text{new}}\|^2\right),$$

where:

$$A_{\text{new}} = A_{11} - A_{12}A_{22}^{-1}A_{21}, \quad b_{\text{new}} = b_1 - A_{12}A_{22}^{-1}b_2.$$

This creates a new prior on the active states X_{active} , summarizing the influence of the marginalized states $X_{\text{marginalized}}$.

III. EXPERIMENTAL SETUP

To evaluate the performance of the windowing-based FGO approach, two separate experiments were conducted, each targeting a specific error source in pedestrian navigation. Since sensor fusion aims to compensate for errors, we tested two major sources of uncertainty: GNSS multipath errors and PDR-related errors. While GNSS multipath affected absolute

positioning due to signal reflections, PDR introduced its own challenges, categorized into Short and High Errors (SHE)—brief but large outliers caused by sudden changes in device placement—and Continuous and Low Errors (CLE)—gradual inaccuracies due to variations in walking patterns, gender, and height.

These experiments provide a comprehensive assessment of how the proposed method handles different navigation uncertainties, although the primary focus of this study remains on improving computational efficiency rather than error characterization. The detailed procedures for data collection at two distinct locations have been described comprehensively in subsections A and B. Data analysis and computations were performed offline using a computer equipped with an 11th Generation Intel(R) Core(TM) i5-1135G7 processor running at 2.40 GHz.

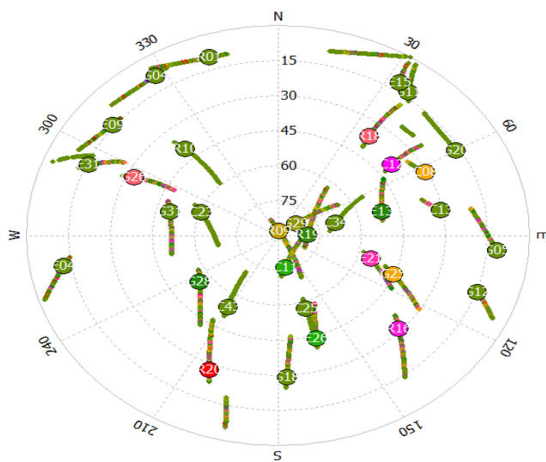
The sliding window sizes utilized in our experiments are explicitly detailed as follows:

For GNSS multipath error analysis, sliding window sizes of 1, 5, 10, 30, and 50 were employed.

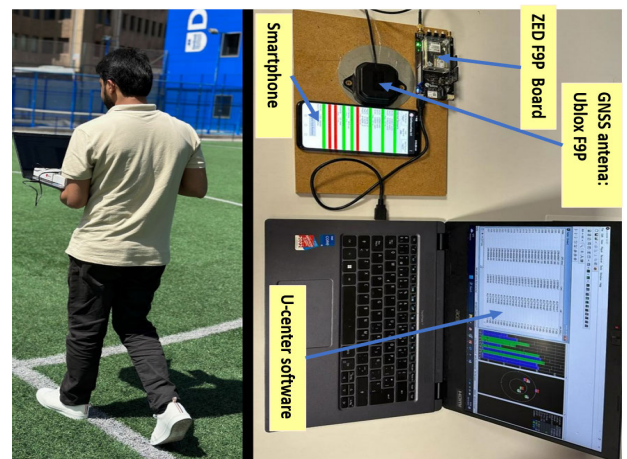
For the analysis of PDR errors (SHE and CLE), window sizes of 1, 5, 10, and 30 were considered.



FIGURE 7. Experimental setup and data collection for GNSS error analysis: ground truth and configuration overview.



(a) Polar skyplot showing the distribution of GNSS satellites at a given epoch. Each marker represents a satellite, with its PRN and constellation indicated (e.g., G01 for GPS satellite PRN 01, R21 for GLONASS satellite PRN 21). Green markers indicate satellites with LOS visibility and good signal quality, while red/pink markers represent NLOS satellites affected by multipath or signal blockage. Orange and yellow colours indicate varying levels of intermediate signal quality or partial obstruction.



(b) Data collection and experimental setup.

FIGURE 8. (a) Satellite visibility for PDR experiment and (b) Data collection setup.

Moreover, comprehensive descriptions of the sliding window FGO techniques and implementation can be found in Section III.

A. UNDER GNSS MULTIPATH CONDITION

The data collection was conducted by a single pedestrian walking along a predefined rectangular path in Korntal Mönchingen, Stuttgart, Germany. The path, approximately 920 meters long, featured partially open-sky conditions with trees and nearby houses. The pedestrian completed two laps of the path, covering a total distance of approximately 1.8 kilometers at a normal walking speed with a duration of 27.5 minutes.

A Google Pixel 4 smartphone was used to record the raw GNSS and IMU measurements. The GNSS data was recorded

using the GNSSLogger app [34], while the smartphone was held in front of the pedestrian's chest during the experiment. Ground truth data was obtained using a u-blox ZED-F9P RTK GNSS receiver, which is capable of providing centimeter-level accuracy. The receiver antenna, an ANN-MB-00, was mounted on a backpack carried by the pedestrian. The 98% RTK fixed solution was achieved via the base station corrections provided by the SAPOS service in Bavaria, Germany. This experimental setup and path are shown in Fig. 7.

B. UNDER PDR SOURCES OF ERRORS

Developing a flexible PDR block for various walking patterns and smartphone-carrying modes is challenging. This study addresses two key sources of PDR errors: SHE,

caused by transient variations in smartphone orientation, and CLE, influenced by pedestrian characteristics and motion dynamics. Data collection was conducted using a Xiaomi Mi 8 smartphone with the GeSensor Data app [35] at the University of Deusto football stadium in Bilbao, Spain. Before starting the data collection process, several conditions were ensured to maintain the quality of the GNSS data, as the focus of the study was primarily on PDR errors. These conditions included adequate satellite visibility, a clear sky environment, and minimal multipath interference. These measures were taken to ensure that the GNSS data were in optimal condition and free from significant errors. Satellite coverage and signal quality were assessed using the skyplot shown in Fig. 8a. The plot illustrates the spatial distribution of GNSS satellites in terms of azimuth and elevation at a specific epoch. This visualization confirms sufficient satellite visibility and highlights the quality of the received signals used in positioning. Ground truth data was recorded simultaneously using a u-blox ZED-F9P RTK GNSS receiver, providing high-precision positioning for comparison.

A 52-meter straight path was used for both experiments under controlled conditions. The brief explanations about the SHE and CLE experiments are detailed below.

1) SHE-PDR OUTLIERS

SHE occurs due to intentional variations in smartphone positioning during movement. For this experiment, the smartphone was subjected to three types of transient disturbances: shaking, swinging, and rotating. These actions were performed to analyze their impact on the PDR block and assess how transient changes in orientation affect positioning accuracy.

2) CLE-PDR ERRORS

CLE experiments investigated the effects of diverse pedestrian characteristics and motion patterns. Ten participants of varying heights, gaits, and gender were selected. Each participant was instructed to hold the smartphone in front of their chest and perform two types of movements: walking and running. These tests were designed to observe how different motion dynamics influence cumulative PDR errors.

For walking and running motions, separate ground truth data was collected and synchronized with GPS time. The smartphone was positioned adjacent to the u-blox GNSS receiver, minimizing the distance between them, as shown in Fig. 8b. This close proximity reduced potential signal delays, ensuring precise and reliable data for analysis.

IV. RESULTS

This section presents the results of the windowing-based FGO approaches, NWFGO (without marginalization) and MWFGO (with marginalization), with a primary focus on the trade-off between computational burden and accuracy. It further analyzes how different windowing strategies and varying window sizes impact FGO's performance, comparing NWFGO, MWFGO, and BFGO in terms of

computational efficiency and positional accuracy. Furthermore, both approaches were evaluated considering that existing literature primarily focuses on INS+GNSS integration, addressing only GNSS errors. However, in PDR+GNSS fusion, PDR errors can vary in nature and require separate consideration.

A. COMPARATIVE TRADE-OFF BETWEEN ACCURACY AND COMPUTATIONAL BURDEN UNDER GNSS MULTIPATH ERRORS

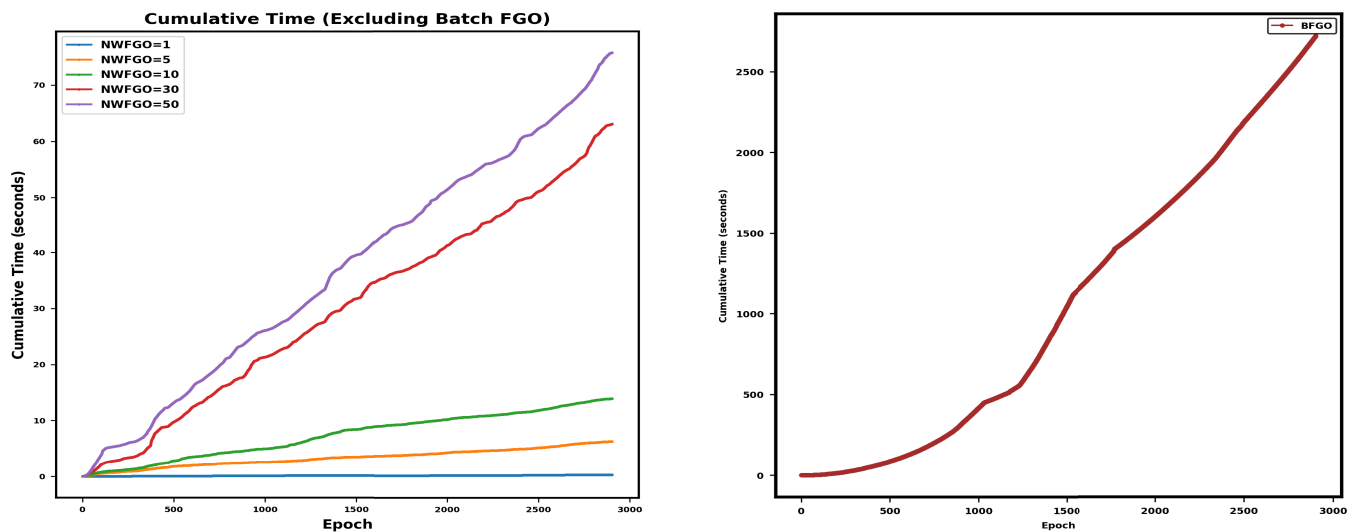
FGO is a promising estimation technique that can potentially replace KF in positioning algorithms. Its key advantage lies in the ability to re-linearize observations multiple times, allowing for better consideration of time correlations in certain states compared to KF. However, a significant drawback of FGO is its high computational burden. For many applications, it is crucial to examine the trade-off between computational cost and accuracy in FGO. Although BFGO provides better accuracy by optimizing all factors, it also incurs a huge computational cost. As shown in Fig. 9, the cumulative time consumed by BFGO reaches 2450s (41min approx.), making it impractical for real-time or resource-constrained applications.

By limiting the number of factors, FGO strikes a balance between computational efficiency and accuracy. However, the choice of the sliding windowing approach significantly influences both accuracy and computational burden. As shown in Fig. 9a, the NWFGO approach achieves a substantial reduction in computational time, reaching 79 seconds for the maximum threshold window size, which translates to a 96.84% reduction in computational power compared to BFGO. When analyzing different window sizes, smaller window sizes result in even greater computational savings. For instance, with a window size of 1, the computational time is reduced to 0.45 seconds, saving 99.98% of the computational power. Moreover, Fig. 10 shows that varying window sizes impact the number of LM optimizer iterations required for convergence. Smaller window sizes result in a lower and more stable number of iterations, while increasing the window size leads to a higher number of iterations.

However, the trade-off becomes evident in accuracy. Smaller window sizes lead to greater accuracy degradation, with a 5.76% reduction in positional accuracy. The optimal threshold for window size is found at 50 factors, beyond which further increasing the window size does not yield additional accuracy improvements. This threshold achieves accuracy comparable to BFGO while maintaining a significantly lower computational burden. A detailed comparison of computational burden and mean error across different window sizes is presented in Table 1.

Comparison of MWFGO against NWFGO and BFGO:

As discussed above, the NWFGO approach is straightforward, optimizing factors within the defined window size while discarding those outside. Although information from discarded factors is not retained, previously optimized poses are stored in a dictionary for trajectory plotting and further



(a) Cumulative computational time consumed by different window sizes with NWFGO technique

(b) Cumulative computational time used by BFGO

FIGURE 9. Comparison of computational time used by different window sizes via NWFGO and BFGO.

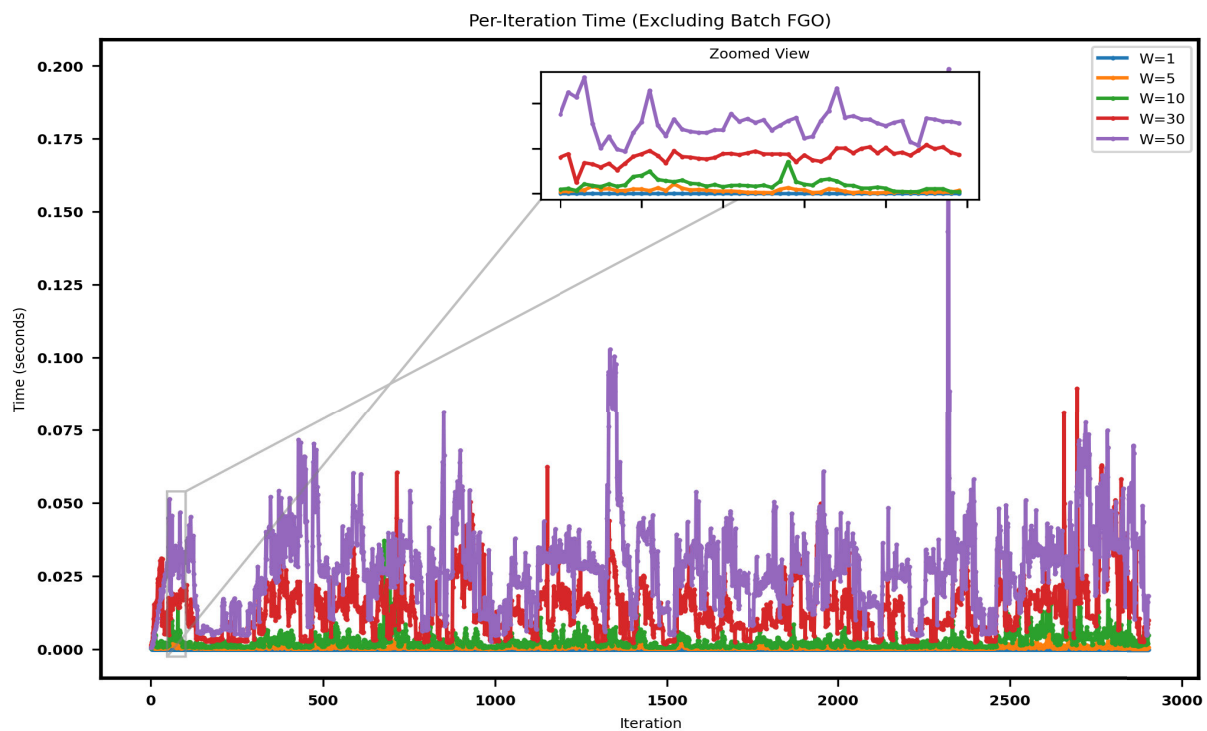
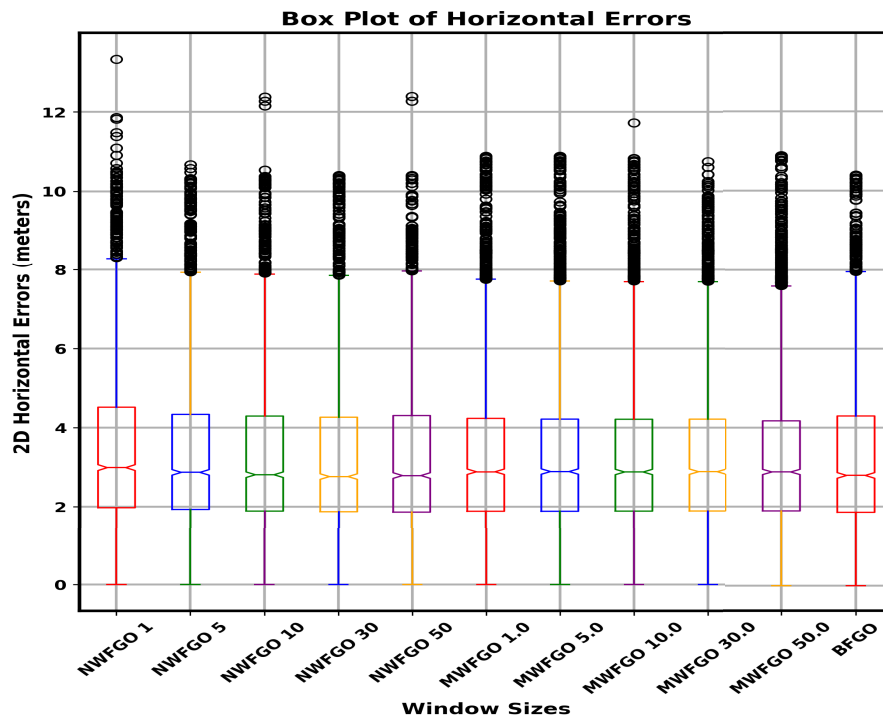


FIGURE 10. Number of LM optimizer iterations taken for convergence from different window sizes via NWFGO.

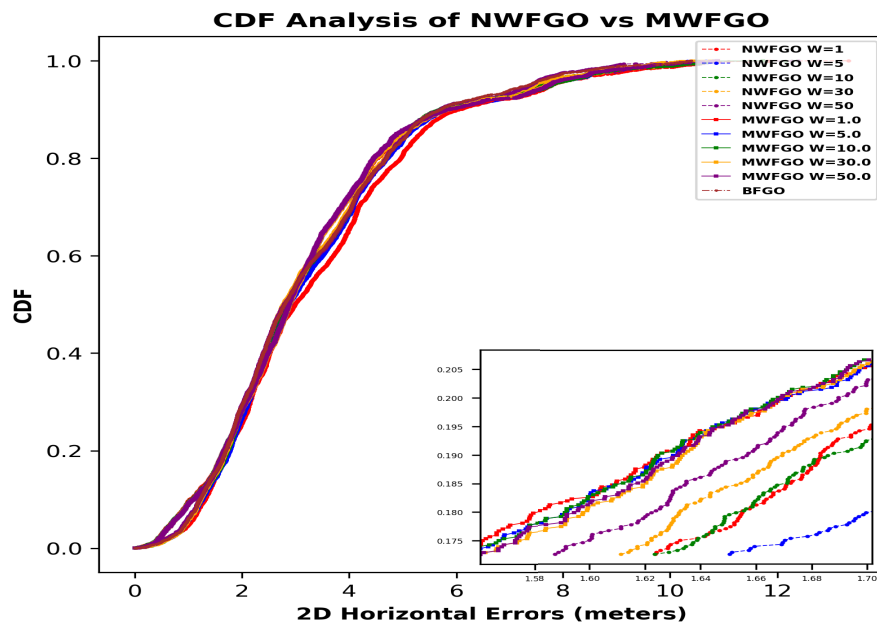
analysis. In contrast, the MWFGO approach retains the information from the removed factors by marginalizing them. This process compresses the removed factors into a new prior, allowing their information to be preserved and utilized as detailed in section II. Below we will highlight the two things effect of marginalization and effect of different window sizes on BFGO via NWFGO and MWFGO.

1) EFFECT OF MARGINALIZATION

From a statistical perspective, MWFGO demonstrates slightly better accuracy across all window sizes compared to the NWFGO approach. This improvement is attributed to the inclusion of compressed, removed factors information in the optimization process. The 2D mean and median errors for MWFGO are marginally better, as observed in Fig. 11a.



(a) Combined box plot for comparative analysis of MWFGO against BFGO and NWFGO



(b) Combined CDF for comparative analysis of MWFGO against NWFGO and BFGO

FIGURE 11. Comparison of results using (a) Box Plots and (b) Cumulative Distribution Function (CDF).

Additionally, the Cumulative Distribution Function (CDF) curves for MWFGO show somehow improved performance over the NWFGO approach which can be seen in Fig. 11b.

Furthermore, the effect of marginalization is reflected in the 2D Horizontal Position Errors (HPEs) shown in Figures 12a and 12b. The GNSS trajectory is intentionally excluded from the comparison plots because it is not part of the performance benchmarking within this specific scope. The MWFGO approach demonstrates slightly better

accuracy, reducing errors compared to NWFGO. Also, from Table 2, the MWFGO approach, which retains information from removed factors through marginalization, achieves an average mean percentage improvement of 0.67% across all window sizes compared to the NWFGO. Moreover, MWFGO exhibits a 0.18% accuracy degradation, whereas NWFGO shows a significantly higher 5.74% accuracy reduction compared to BFGO. This highlights the advantages of marginalization in enhancing positional accuracy over

TABLE 1. HPE and computational times for NWFGO across various window sizes, compared with the BFGO baseline. The “Computational Time Reduction (%)” row shows the percentage decrease in runtime relative to BFGO, while the “Accuracy Reduction (%)” row shows the relative increase in positioning error compared to BFGO. This table illustrates the trade-off between reduced computational cost and positional accuracy as the window size decreases.

Approach	Analysis	1	5	10	30	50	BFGO
NWFGO	Mean Errors (m)	3.454	3.360	3.305	3.288	3.267	3.266
	Time (s)	3	9	12	55	72	2450
Computational Time Reduction (%)		-	99.88	99.64	99.52	97.80	97.12
Accuracy Reduction (%)		-	5.76	2.87	1.19	0.67	0.03

discarding information entirely. The practical implications of these findings are significant. By retaining the compressed information of discarded factors, MWFGO can achieve better accuracy, which is crucial for applications requiring high precision, such as pedestrian navigation in GNSS-challenged environments. Furthermore, the ability to maintain accuracy without relying solely on batch optimization makes MWFGO a more viable option for real-time applications. In summary, the comparison between the NWFGO and MWFGO approaches underscores the importance of marginalization in retaining useful information from discarded factors. MWFGO achieves lower errors and demonstrates its effectiveness in challenging scenarios, making it a more robust and reliable option for FGO.

2) EFFECT OF VARYING WINDOW SIZES

In BFGO commonly, the LM optimizer is used, which re-runs the optimization process from the beginning during each GNSS measurement. This means the optimizer repeatedly processes and optimizes the entire graph in every iteration, regardless of the graph size. For large datasets, such as in our case, this approach becomes extremely time-consuming this can be seen in Fig. 13a. Additionally, the LM optimizer performs multiple internal iterations within each cycle to achieve convergence, minimizing errors and producing more accurate and optimized results. The computational demands of these repeated processes are especially significant for large-scale datasets, as illustrated in Fig. 14.

It can be observed that limiting the number of factors in the factor graph for optimization causes the number of internal iterations to vary with the window size. Specifically, window size 5 exhibits noticeable peaks in the number of LM iterations, followed by window sizes 30s and 10s. In contrast, BFGO maintains a constant number of iterations. However, it is important to note that the time taken for each iteration in BFGO is significantly higher, as shown in the combined Fig. 13b. For NWFGO and MWFGO approaches, it is evident that with larger window sizes, the time per iteration increases, resulting in a substantially higher cumulative computation time. This highlights the trade-off between accuracy and computational efficiency when managing factor graph sizes.

Varying window sizes have a significant impact on the time consumed by FGO, while the improvement in positional accuracy remains marginal. This suggests that using larger window sizes in BFGO may be impractical. From the final pose trajectories shown in Fig. 15a and Fig. 15b, it is

evident that smaller window sizes, such as 1, result in slight outward deviations, followed by window size 5. As the optimization window increases, we can see that the result generally becomes smoother and more stable. however, it is also important to note that In this case, once the window size reaches 50, the 2D mean HPE stabilizes, and beyond this size, the output remains consistent with no further improvement that can be seen in Table 2 and the 2D mean HPE comparison in Fig. 12. Moreover, the computational time in MWFGO is significantly lower than in NWFGO. For a maximum threshold window size of the 30s, MWFGO requires 38 seconds, whereas NWFGO takes 72 seconds that can be seen in Table 3. Based on this analysis, MWFGO proves to be a more suitable option for real-time applications.

B. COMPARATIVE ANALYSIS OF TRADE-OFF BETWEEN ACCURACY AND COMPUTATIONAL BURDEN UNDER PDR SOURCES OF ERRORS

Considering that existing literature primarily focuses on INS+GNSS integration, addressing only GNSS errors. However, in PDR+GNSS fusion, PDR errors can vary in nature and require separate consideration. As previously outlined, two scenarios were designed to introduce PDR errors. These errors arise because pedestrians do not consistently carry smartphones in the same position. Variations in how smartphones are held, combined with differences in gait patterns, height, gender, and motion (walking versus running), significantly impact the performance of the PDR block within the PDR+GNSS fusion architecture.

The first type of error, referred to as Short and High Errors (SHE), is caused by transient variations in smartphone positioning. These errors occur over short durations but have a high magnitude. The second type, known as Continuous and Low Errors (CLE), results from differences in pedestrian characteristics, such as height, gender, gait patterns, and walking speeds. CLE introduces low-magnitude errors that persist continuously over time. The performance of the two sliding window approaches, NWFGO and MWFGO, was evaluated using this dataset under these PDR conditions. The datasets used in this study are from our previous study [12].

1) EFFECT OF MARGINALIZATION UNDER SHE-PDR OUTLIERS

To analyze the impact of marginalization under SHE-PDR outliers, three specific scenarios shaking, swinging, and rotating were chosen and examined. These scenarios

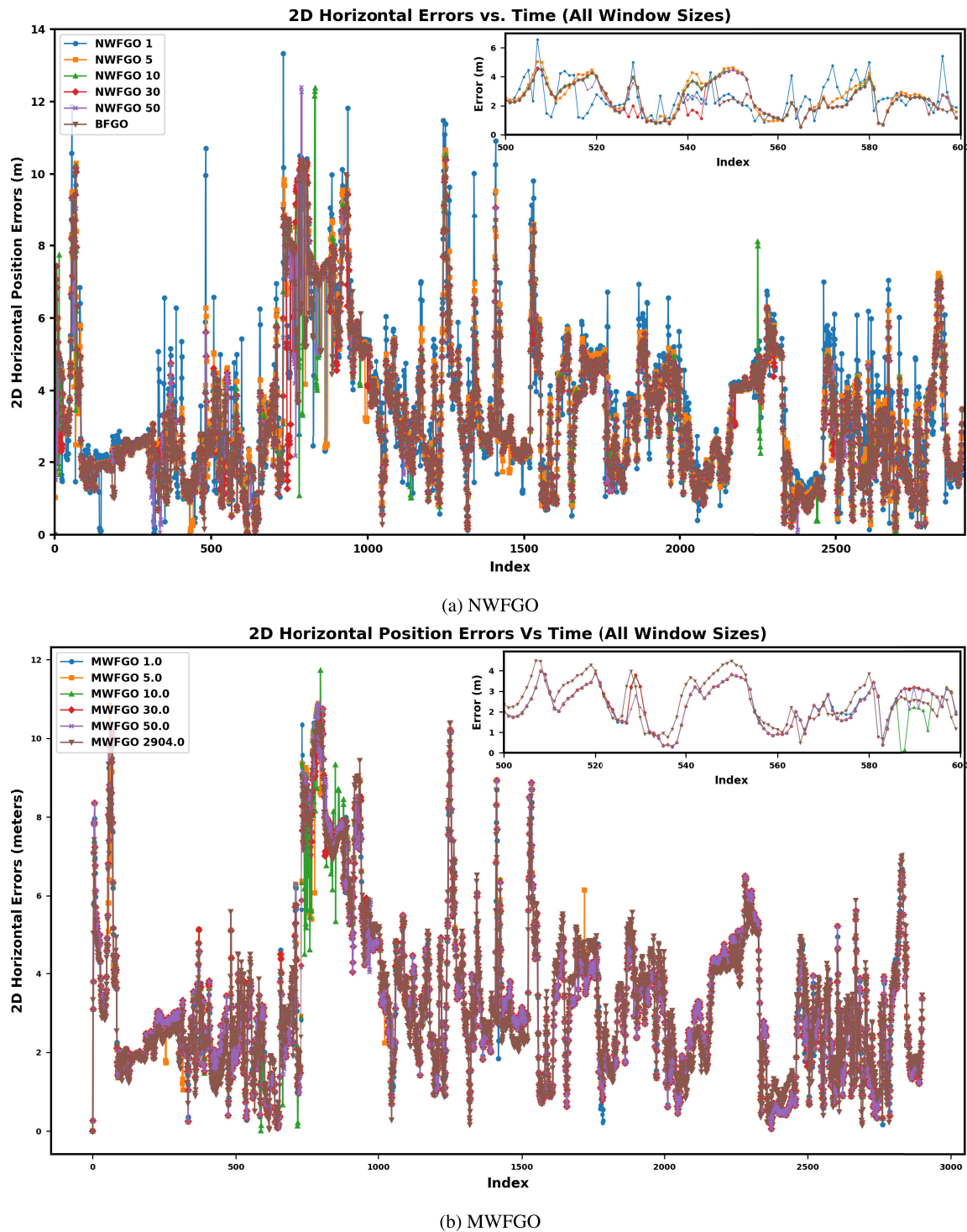


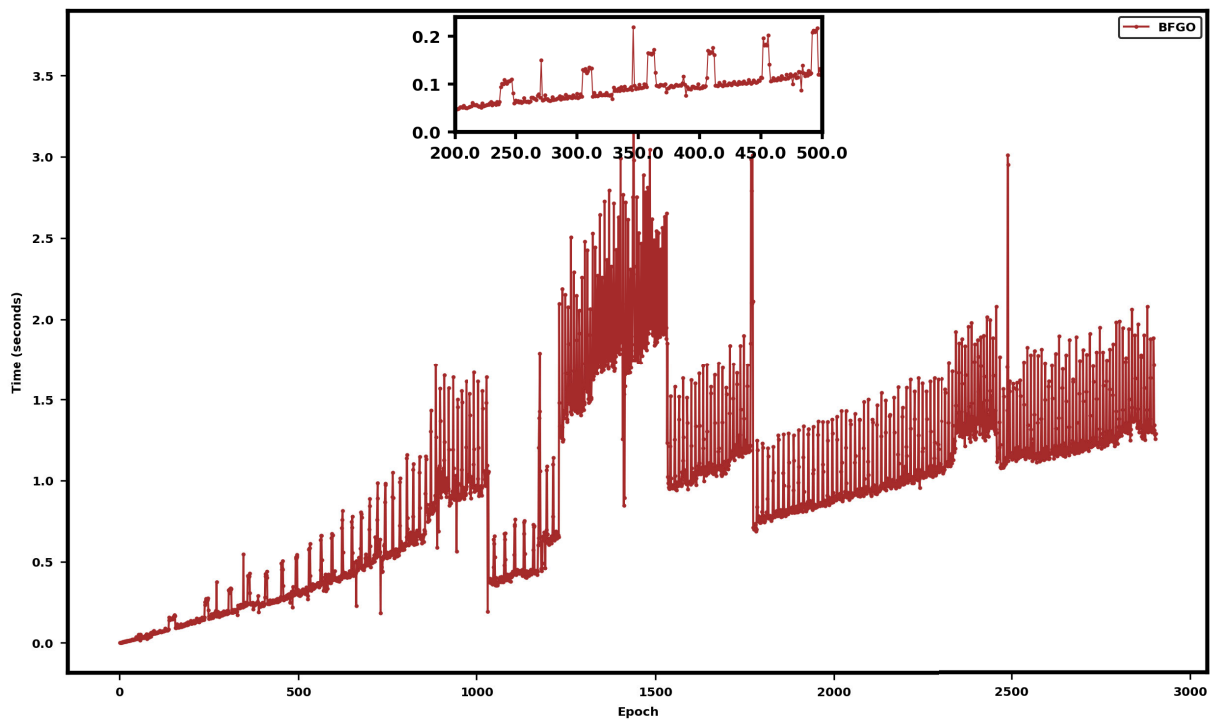
FIGURE 12. Comparison of Horizontal Position Errors (HPE) using (a) NWFGO and (b) MWFGO.

represent common transient variations in how pedestrians carry smartphones, introducing high-magnitude but short-duration errors.

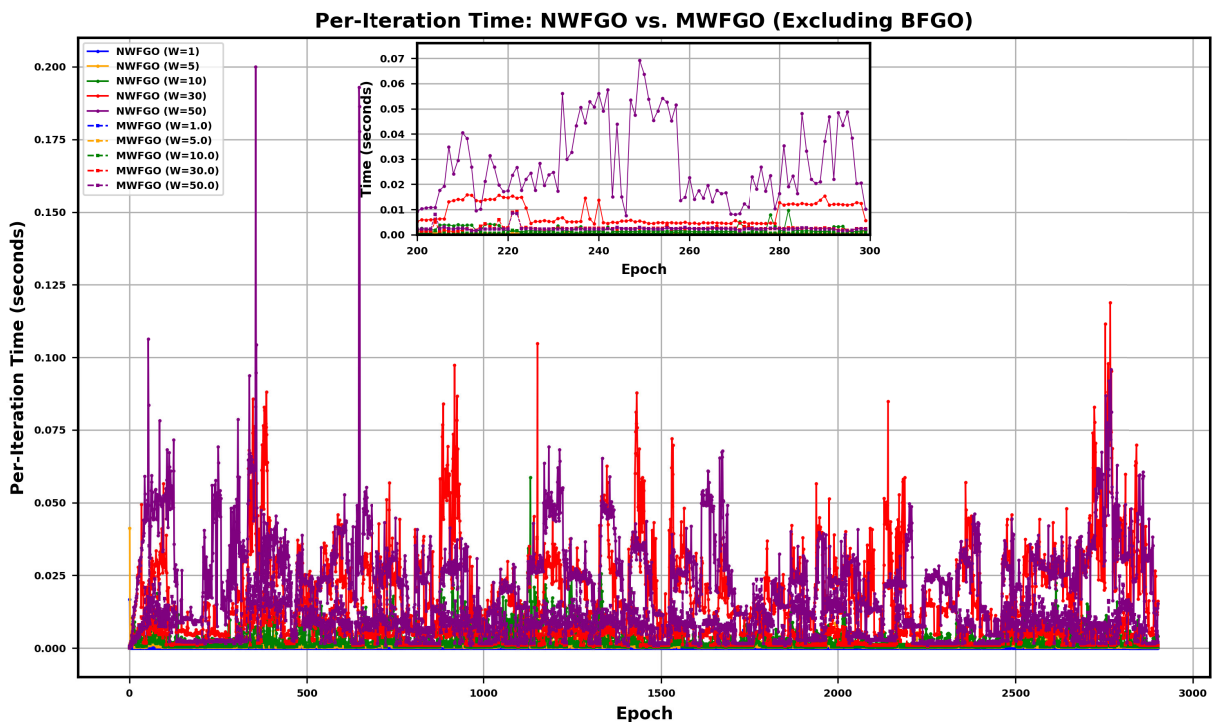
The effects of marginalization were analyzed for each case, focusing on the impact of incorporating compressed information from removed factors as a new prior in the cost function. As shown in Table 4, marginalization resulted in

mean percentage improvements of 5.57% for heading, 2.74% for shaking, and 4.21% for swinging, compared to the Naive approach. Additionally, Figures 16 and 17 illustrate how marginalization enhanced accuracy across different window sizes, further improving the overall output.

Furthermore, Figure 18 shows the 2D horizontal position errors for all three scenarios shaking, swinging, and rotating.



(a) LM iterations for BFGO



(b) LM iterations for NWFGO vs MWFGO

FIGURE 13. Comparison of the number of iterations used by LM: (a) BFGO and (b) NWFGO vs MWFGO, across different window sizes.

The results confirm that MWFGO performs slightly better in these cases, producing smoother and more stable trajectories. This shows that marginalization effectively mitigates the impact of transient variations, offering a more robust performance in PDR+GNSS fusion architectures.

In general, the analysis underscores the importance of marginalization in retaining critical information, which is particularly beneficial in scenarios with SHE-PDR outliers. By compressing and reusing discarded factors and poses, MWFGO achieves better accuracy and stability compared

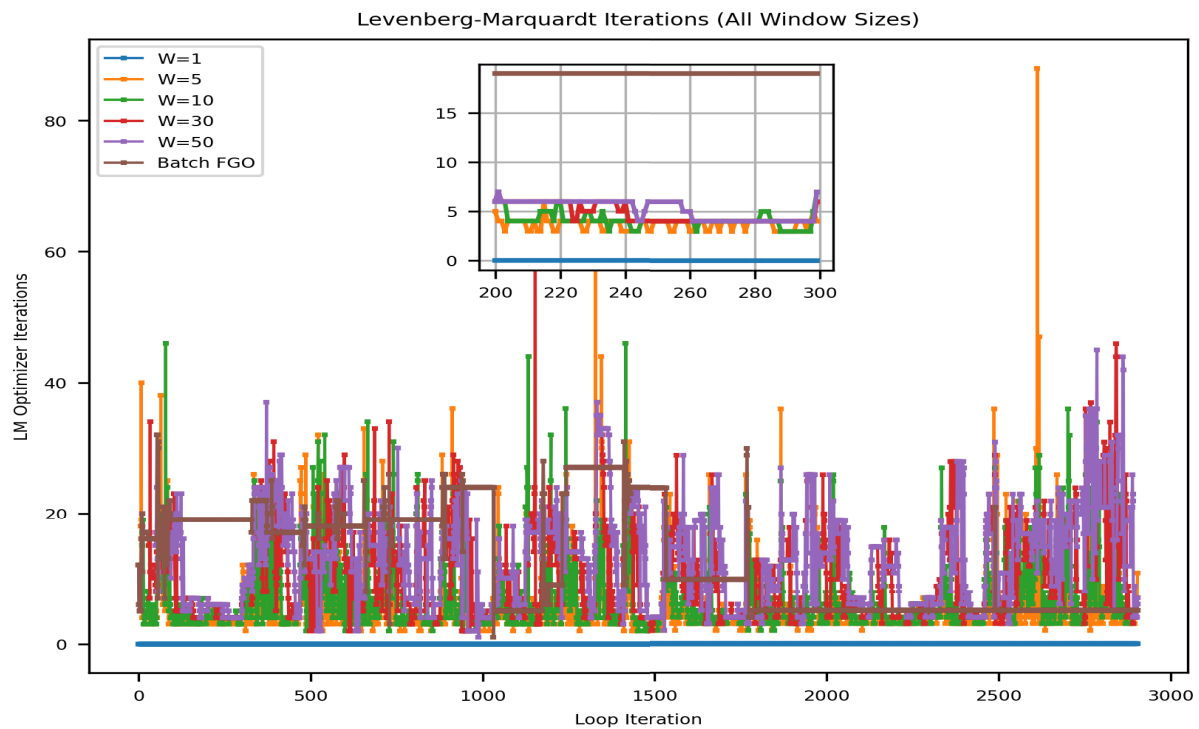


FIGURE 14. Number of iterations used by LM from different window sizes.

TABLE 2. 2D mean HPE comparison between NWFGO and MWFGO across different window sizes. The “Mean % Improvement w.r.t NWFGO” column reflects the average reduction in HPE achieved by MWFGO compared to NWFGO across all window sizes. The “Avg % Degradation w.r.t BFGO” indicates the average increase in error relative to the BFGO baseline.

Method	W 1	W 5	W 10	W 30	W 50	BFGO	Mean % Improvement w.r.t NWFGO	Avg % Degradation w.r.t BFGO
NWFGO	3.454	3.361	3.305	3.288	3.267	3.266	-	5.74%
MWFGO	3.288	3.279	3.274	3.269	3.267	-	0.67%	0.18%

TABLE 3. Comparison of mean 2D HPE and computational times for NWFGO and MWFGO across various window sizes. This table highlights the trade-off between accuracy and computational efficiency for both windowing approaches, showing how MWFGO maintains similar accuracy with reduced computation time.

Analysis	Approach	1	5	10	30	50	BFGO
Mean Errors (m)	NWFGO	3.454	3.360	3.305	3.288	3.267	3.266
	MWFGO	3.288	3.279	3.274	3.269	3.267	-
Time (s)	NWFGO	3	9	12	55	72	2450
	MWFGO	2	4	7	22	38	-

to NWFGO, making it a more effective approach to address transient variations in smartphone-based pedestrian navigation.

2) EFFECT OF VARYING WINDOW SIZES UNDER SHE-PDR OUTLIERS

Interestingly, in this scenario, the HPE for smaller window sizes shows lower errors compared to BFGO. The effect of different window sizes on 2D HPE is presented in Fig.18. As shown in Table 4, unlike the GNSS error

scenarios, the mean error for smaller window sizes is slightly better than that of BFGO. This is due to the smaller dataset size and significant sudden variations in how the smartphone is carried or disturbed, which allow smaller window sizes to capture and optimize results more effectively. Furthermore, the short path length of 52 meters and minimal displacements between epochs further highlight the suitability of smaller windows for capturing transient variations, demonstrating their advantage over batch data in this specific scenario.

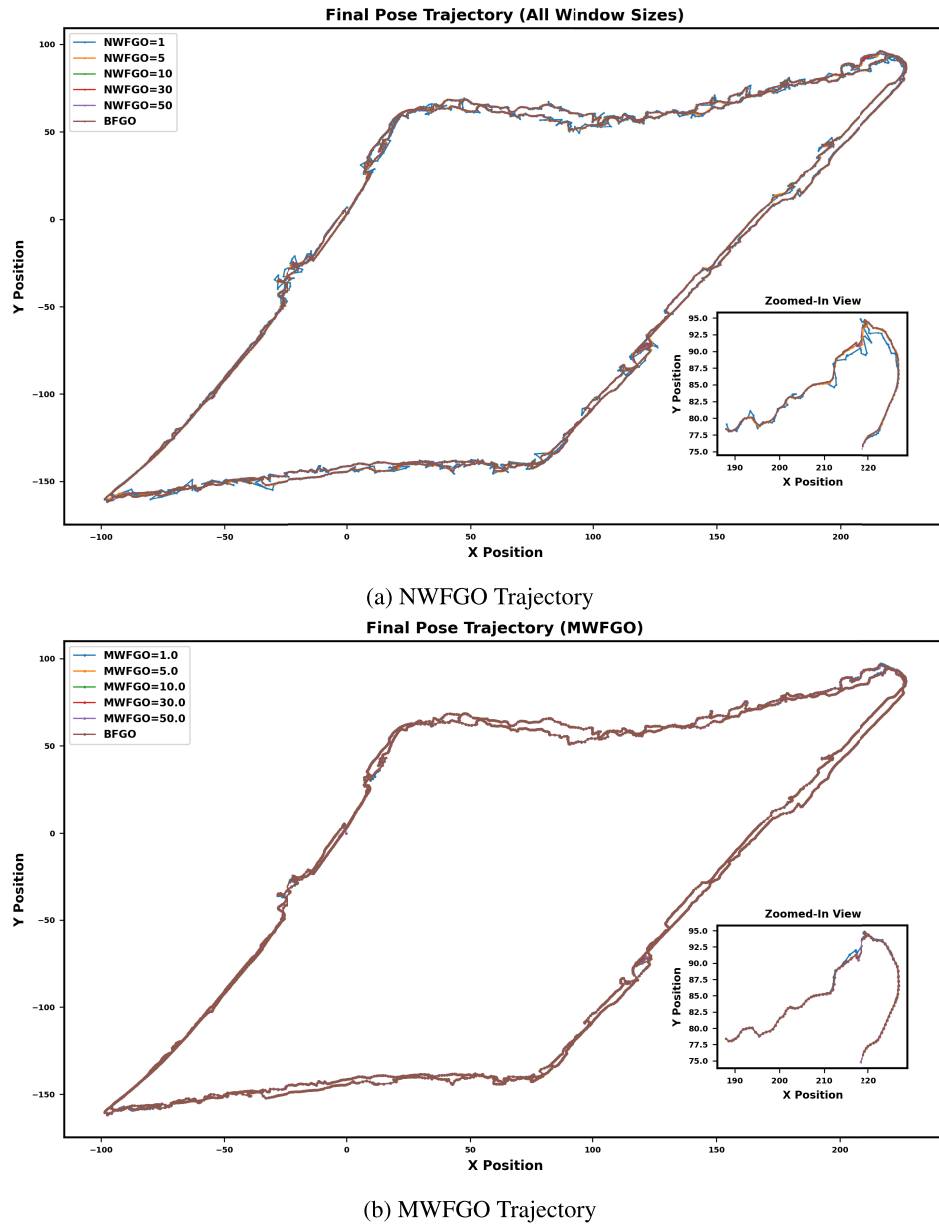


FIGURE 15. Comparison of final pose trajectories using (a) NWFGO and (b) MWFGO.

It is important to note that after an optimization window size of 30, the positional error stabilizes, producing consistent outputs with no further improvement in accuracy. This indicates that 30 is the upper threshold for the optimization window, beyond which increasing the size does not enhance accuracy. Based on this result, it can be concluded that using BFGO is no longer necessary if an effective strategy is implemented to determine and validate the upper threshold value of FGO. This approach can significantly reduce computational load while maintaining accuracy.

3) EFFECT OF MARGINALIZATION PDR-CLE ERRORS

To evaluate the impact of marginalization on PDR-CLE errors, the results are illustrated in the 2D HPE plot shown in Fig. 19. The figure highlights that the marginalization-based

windowing approach improved the FGO performance by 2.73% for walking and 2.00% for running, underscoring the effectiveness of this method in mitigating errors across diverse scenarios.

Additionally, insights from the accompanying Table 5 and box plot Fig. 20 reveal that at lower window sizes, the improvements slightly degrade compared to BFGO, similar to the GNSS error scenario. However, the percentage degradation is marginally higher for the PDR-CLE dataset. Specifically, the average degradation with NWFGO is 5.25% for walking, while marginalization reduces it to 4.75%. For running, the degradation averages 5.70% with NWFGO and decreases to 5.10% with MWFGO. This difference can be attributed to variations in human walking styles, speeds, and motions, which inherently affect overall accuracy.

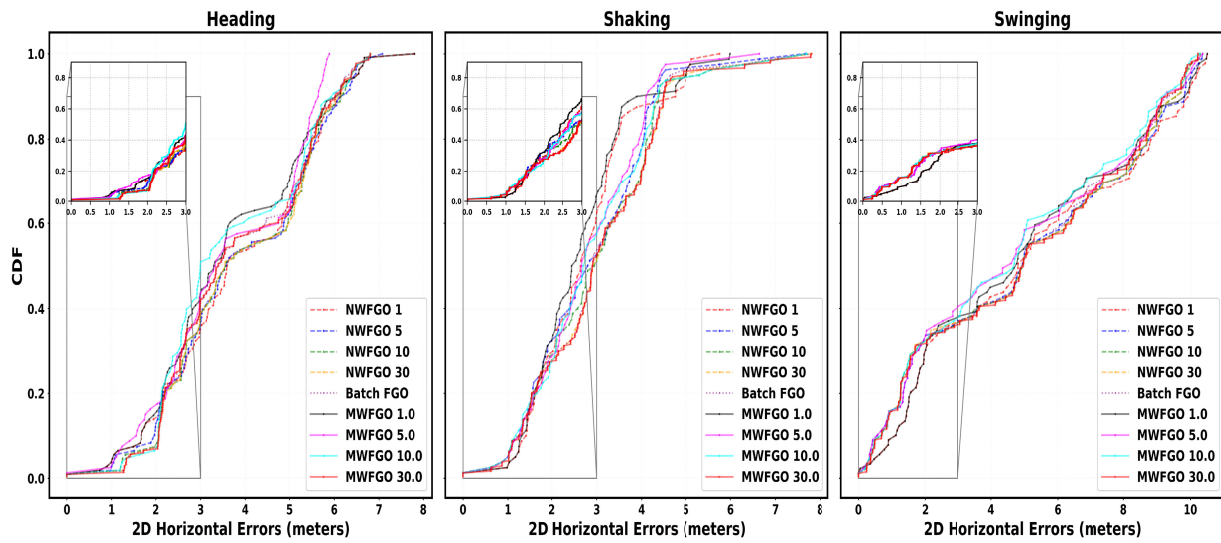


FIGURE 16. CDF analysis of SHE-PDR errors.

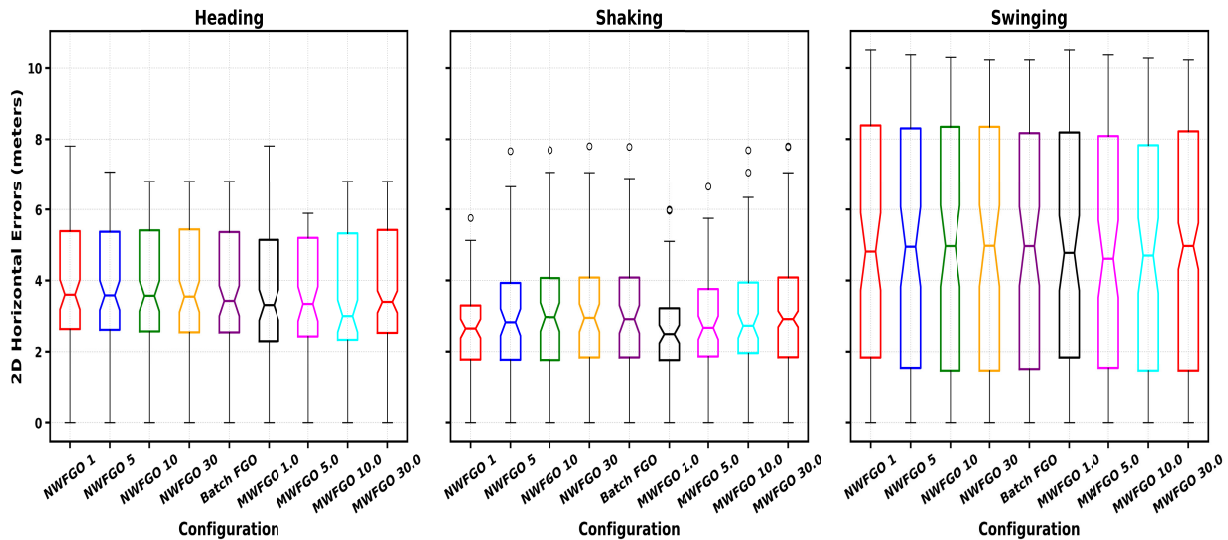


FIGURE 17. Box plot SHE-PDR errors.

TABLE 4. Comparison of 2D mean HPE between NWFGO and MWFGO under different SHE-PDR outlier scenarios: Heading, Shaking, and Swinging. For each case, mean HPE is reported across various window sizes. BFGO is included for the baseline. The final column reports the mean percentage improvement in accuracy of MWFGO relative to NWFGO, averaged across all window sizes.

Case	Method	W 1	W 5	W 10	W 30	BFGO	Mean % Improved w.r.t NWFGO
Heading	NWFGO	3.925	3.936	3.949	3.952	3.841	None
	MWFGO	3.675	3.659	3.709	3.841	-	5.57%
Shaking	NWFGO	2.724	2.876	2.998	3.046	2.985	None
	MWFGO	2.637	2.788	2.910	2.990	-	2.74%
Swinging	NWFGO	5.130	5.017	4.998	4.990	4.921	None
	MWFGO	4.951	4.781	4.672	4.924	-	4.21%

Notably, the improvements stabilize for both walking and running after a window size of 30, achieving results comparable to BFGO. This stabilization further validates the effectiveness of the proposed MWFGO approach in balancing computational efficiency with positioning accuracy.

4) EFFECT OF VARYING WINDOW SIZES ON CLE-PDR ERRORS

Fig. 19 provides a visualization of the 2D HPE for different window sizes. The results indicate that smaller window sizes, such as 1 and 5, show slightly higher errors due to limited contextual information, particularly in scenarios with diverse

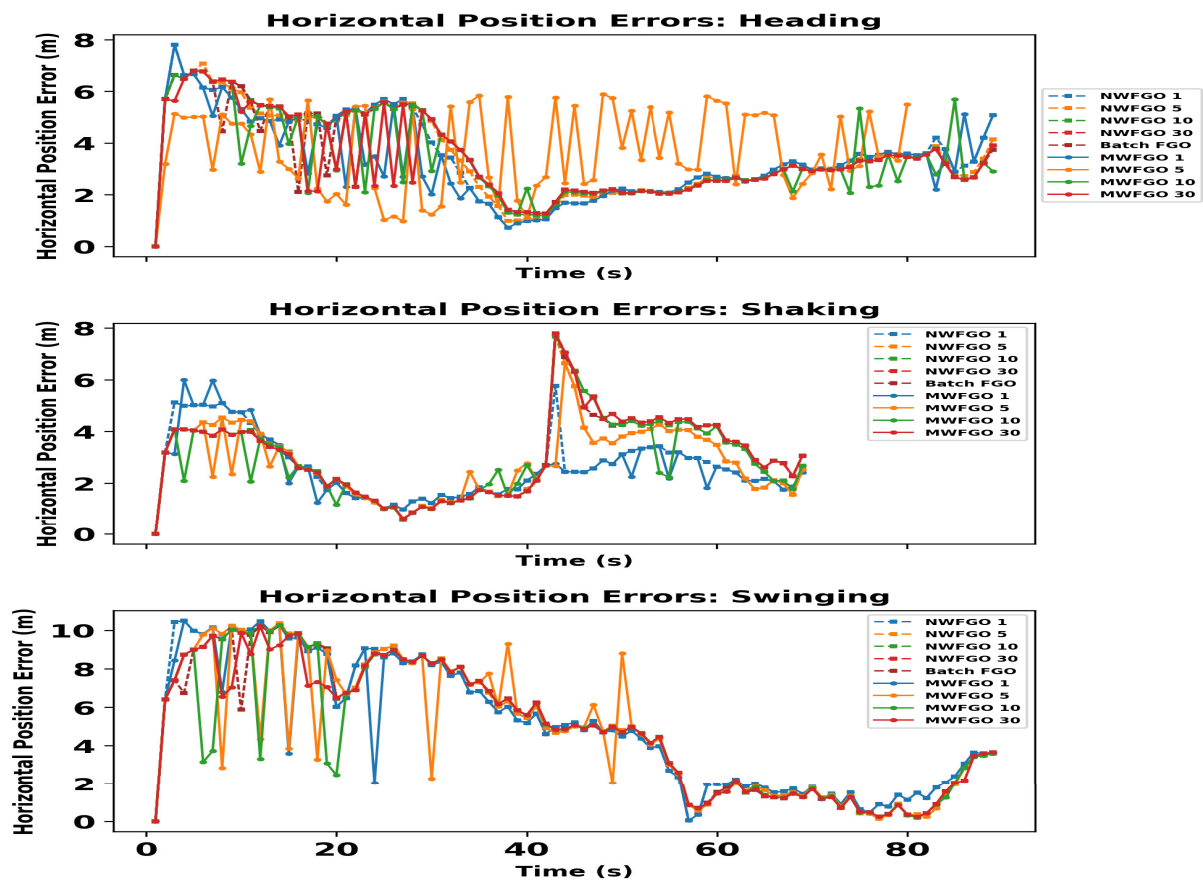


FIGURE 18. Both NWFGO vs MWFGO per iteration time used.

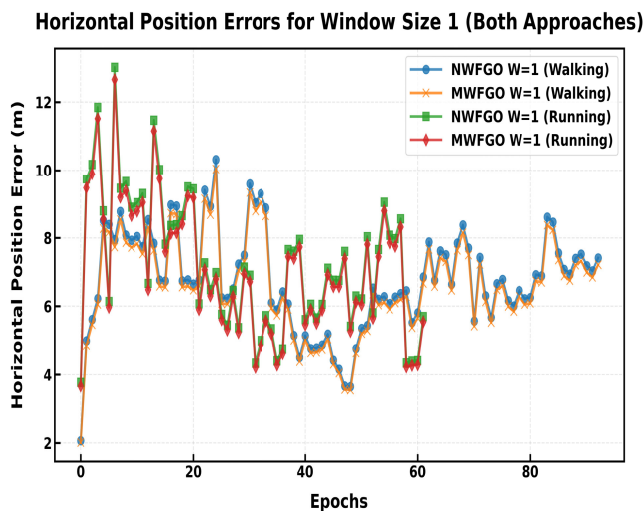


FIGURE 19. Marginalization effect on walking and running datasets for only window size 1 via NWFGO and MWFGO.

pedestrian motion patterns. However, as the window size increases, the errors reduce and gradually stabilize.

Quantitative analysis further supports this observation. The average degradation in NWFGO was 5.25% for walking and

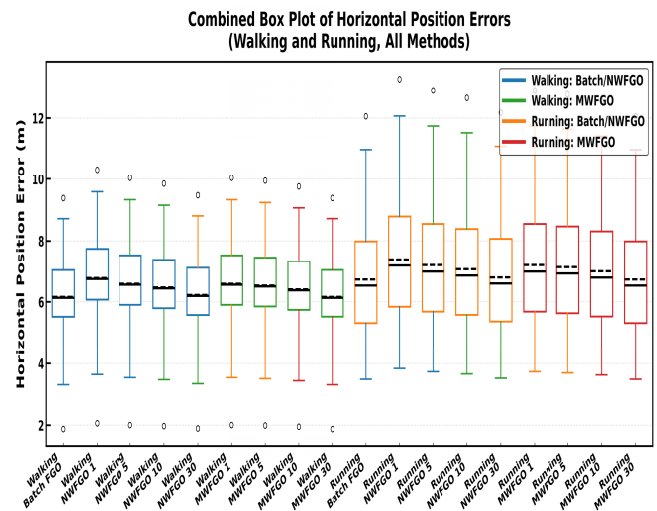
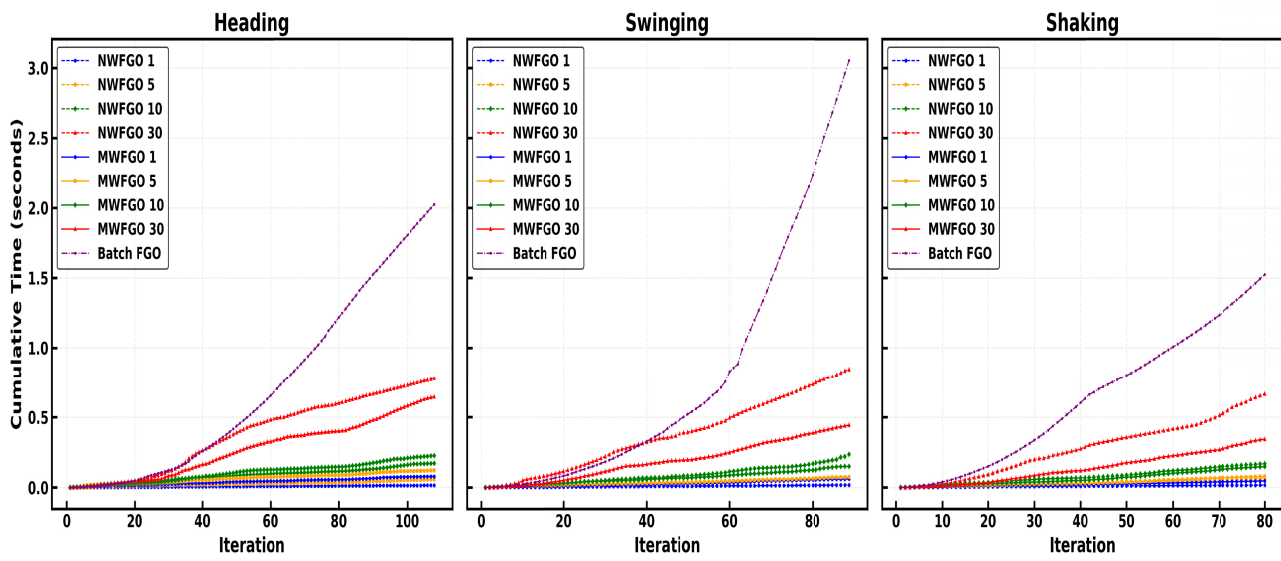
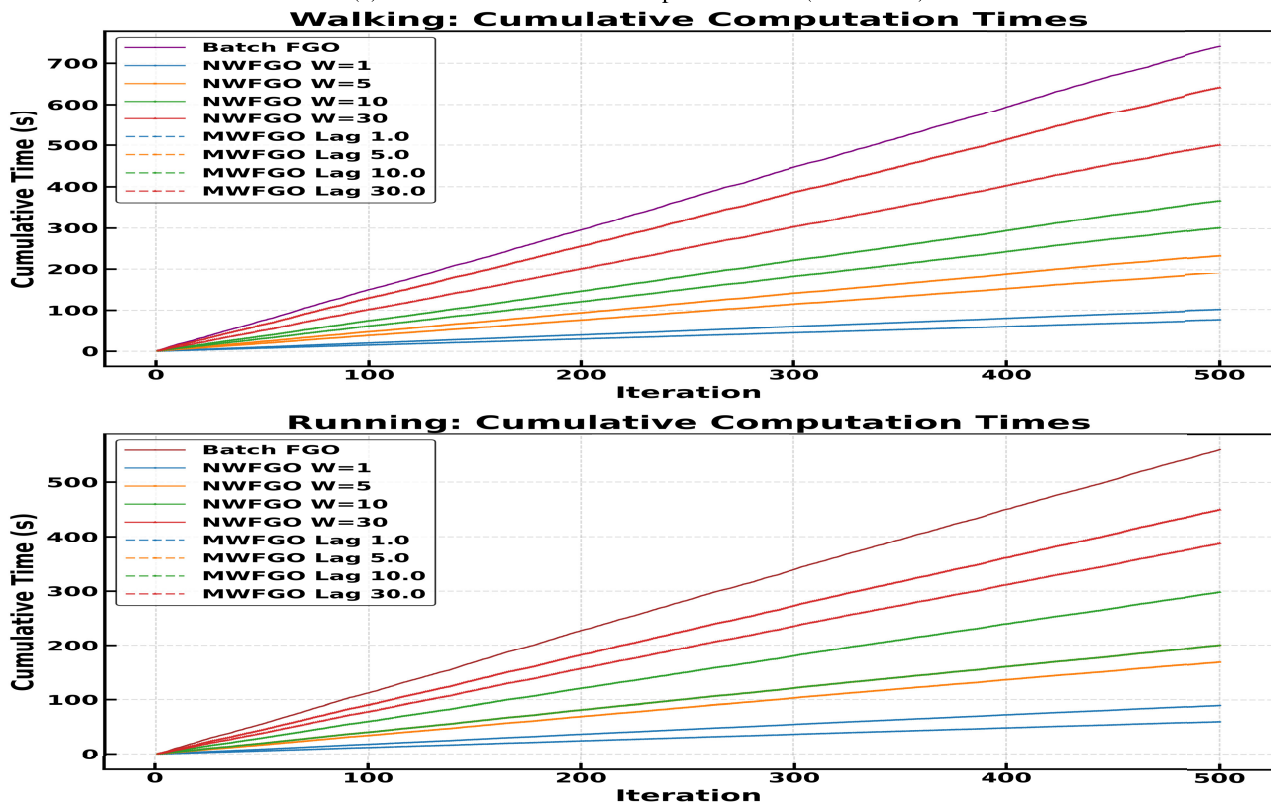


FIGURE 20. Box plot for walking and running motion from 10 different pedestrian datasets.

5.70% for running, whereas MWFGO reduced the degradation to 4.75% and 5.10%, respectively. These improvements demonstrate the ability of MWFGO to retain critical information from discarded factors, thereby enhancing accuracy under CLE-PDR conditions.



(a) NWFGO vs MWFGO computation time (SHE-PDR).



(b) Cumulative time for CLE-PDR (walking and running motion from 10 pedestrians)

FIGURE 21. Comparison of cumulative computation times for NWFGO and MWFGO approaches in different scenarios.

The observed behavior can be attributed to the inherent variability in pedestrian gait patterns, walking speeds, and smartphone carrying positions. Such factors introduce additional challenges for positioning systems, which are partially mitigated by using larger window sizes. Notably, after a window size of 30, the improvements in both NWFGO

and MWFGO approaches stabilize, achieving performance comparable to BFGO.

In summary of this detailed analysis above, it becomes evident that using BFGO is not always mandatory, as it is computationally expensive and provides similar positional accuracy when optimal threshold window sizes are reached.

TABLE 5. Comparison of 2D mean positioning errors for NWFGO and MWFGO under walking and running across various window size compared to BFGO. The “Mean % Improved w.r.t BFGO” column shows the relative improvement of MWFGO over BFGO, while the “Avg % Degradation in window sizes w.r.t BFGO” reflects the average loss in accuracy across windowed cases compared to BFGO.

Case	Method	W 1	W 5	W 10	W 30	BFGO	Mean % Improved w.r.t BFGO	Avg % Degradation in window sizes w.r.t BFGO
Walking	NWFGO	6.772	6.587	6.464	6.218	6.156	None	5.25%
	MWFGO	6.587	6.526	6.403	6.156	-	2.73%	4.75%
Running	NWFGO	7.398	7.196	7.061	6.792	6.725	None	5.70%
	MWFGO	7.196	7.129	6.994	6.725	-	2.00%	5.10%

As observed in the GNSS error conditions and the PDR scenarios, threshold window sizes of 50 and 30, respectively, achieved positional accuracy comparable to BFGO. Utilizing lower threshold values not only reduces computational power consumption but also maintains better accuracy.

Additionally, it is generally observed that smaller window sizes tend to result in reduced accuracy compared to BFGO. However, in the SHE-PDR case, BFGO demonstrated higher errors due to transient variations in how the smartphone was held. This emphasizes that the selection of windowing techniques and sizes must be carefully adapted to the specific conditions and scenarios where fusion is applied. These findings highlight the critical need for scenario-specific optimization to strike a balance between accuracy and computational efficiency.

Like the GNSS-error case a similar pattern is observed in the CLE-PDR error scenarios, where MWFGO delivers a 2.73% improvement in mean error compared to NWFGO. The differences in computational load between NWFGO and MWFGO are further highlighted in Fig. 21b, emphasizing MWFGO’s effectiveness in reducing computational cost while maintaining accuracy. However, In the case of SHE-PDR outliers, smaller window sizes achieved better overall accuracy compared to BFGO while significantly reducing computational cost, as shown in Fig. 21a.

Our analysis shows that positional accuracy stabilizes after reaching a certain threshold window size, making BFGO unnecessary in most scenarios. As we saw in the GNSS error case and CLE-PDR datasets, threshold window sizes of 50 and 30 achieved accuracy comparable to BFGO, while significantly reducing computational load, as shown in Table 3. This indicates that it is not always mandatory to use BFGO. Instead, determining and validating the optimal threshold window size for specific cases and scenarios can drastically reduce computational time without compromising positional accuracy. Importantly, BFGO does not always deliver superior accuracy, and careful consideration of threshold values can yield similar results with far greater efficiency.

In conclusion, while BFGO remains a benchmark for accuracy, its high computational cost limits its applicability. Windowed FGO, particularly MWFGO, emerges as a robust alternative, striking a balance between accuracy and efficiency. By tailoring window sizes and employing marginalization techniques, significant reductions in

computational load can be achieved, enabling scalable and practical solutions for diverse scenarios.

V. CONCLUSION

This study evaluated the performance of two windowing-based FGO approaches, NWFGO and MWFGO, to enhance smartphone-based pedestrian navigation while addressing the trade off between computational burden and accuracy of BFGO. The findings demonstrate that BFGO is not always necessary to achieve high positional accuracy, as its effectiveness varies based on the dataset, error characteristics, and fusion conditions. By optimizing threshold window sizes instead of relying on batch data, comparable accuracy can be achieved with significantly reduced computational load. This makes windowed FGO a practical and efficient alternative for real-time and resource-constrained applications.

Furthermore, the study highlighted that smaller window sizes do not always degrade accuracy. Under specific conditions, such as SHE-PDR errors, smaller windows demonstrated improved performance due to transient variations in smartphone handling. These findings emphasize the importance of tailoring windowing techniques and sizes according to the scenario, enabling adaptive and efficient navigation solutions.

In conclusion, windowing-based FGO, particularly the MWFGO approach, offers a robust alternative to BFGO by achieving a balance between computational efficiency and positional accuracy. This work provides a foundation for further exploration of scenario-specific optimizations in pedestrian navigation systems, paving the way for scalable and resource-efficient solutions for smartphone-based navigation.

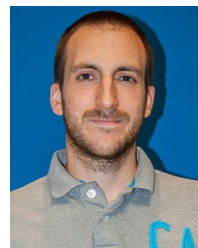
REFERENCES

- [1] J. Paziewski, “Recent advances and perspectives for positioning and applications with smartphone GNSS observations,” *Meas. Sci. Technol.*, vol. 31, no. 9, Apr. 2020, Art. no. 091001.
- [2] F. Zangenehnejad and Y. Gao, “GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives,” *Satell. Navigat.*, vol. 2, no. 1, pp. 1–23, Nov. 2021.
- [3] X. Wu, B. Xiao, C. Wu, Y. Guo, and L. Li, “Factor graph based navigation and positioning for control system design: A review,” *Chin. J. Aeronaut.*, vol. 35, no. 5, pp. 25–39, May 2022.
- [4] A. Sheta, A. Mohsen, B. Sheta, and M. Hassan, “Improved localization for Android smartphones based on integration of raw GNSS measurements and IMU sensors,” in *Proc. Int. Conf. Comput. Appl. (ICCA)*, Aug. 2018, pp. 297–302.

- [5] G. Falco, M. Pini, and G. Marucco, "Loose and tight GNSS/INS integrations: Comparison of performance assessed in real urban scenarios," *Sensors*, vol. 17, no. 2, p. 255, Jan. 2017.
- [6] A. Angrisano, "GNSS/INS integration methods," Ph.D. dissertation, Dept. Applied Sci., Università degli Studi di Napoli Parthenope, Naples, Italy, 2011.
- [7] J. A. Farrell and J. Wendel, "GNSS/INS integration," in *Springer Handbook of Global Navigation Satellite Systems*, P. Teunissen and O. Montenbruck, Eds., Cham, Switzerland: Springer, 2017, pp. 811–840.
- [8] D. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, vol. 17. Edison, NJ, USA: IET, 2004.
- [9] C. Jiang, Y. Chen, C. Chen, J. Jia, H. Sun, T. Wang, and J. Hyypä, "Implementation and performance analysis of the PDR/GNSS integration on a smartphone," *GPS Solutions*, vol. 26, no. 3, p. 81, May 2022.
- [10] C. Jiang, Y. Chen, C. Chen, J. Jia, H. Sun, T. Wang, and J. Hyypä, "Smartphone PDR/GNSS integration via factor graph optimization for pedestrian navigation," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.
- [11] Y. Zhong, W. Wen, and L.-T. Hsu, "Trajectory smoothing using GNSS/PDR integration via factor graph optimization in urban canyons," 2022, *arXiv:2212.14264*.
- [12] A. H. Magsi and L. E. Díez, "Comparison of FGO and KF for PDR-GNSS fusion under different PDR errors," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, 2024.
- [13] A. H. Magsi, L. E. Díez, and S. Knauth, "Continuous high-precision positioning in smartphones by FGO-based fusion of GNSS-PPK and PDR," *Micromachines*, vol. 15, no. 9, p. 1141, Sep. 2024.
- [14] A. Rehman, H. Shahid, M. A. Afzal, and H. M. A. Bhatti, "Accurate and direct GNSS/PDR integration using extended Kalman filter for pedestrian smartphone navigation," *Gyroscope Navigat.*, vol. 11, no. 2, pp. 124–137, Apr. 2020.
- [15] H. Lan, C. Yu, and N. El-Sheimy, "An integrated PDR/GNSS pedestrian navigation system," in *Proc. China Satell. Navigat. Conf. (CSNC)*, Jan. 2015, pp. 677–690.
- [16] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Found. Trends Robot.*, vol. 6, nos. 1–2, pp. 1–139, 2017.
- [17] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.
- [18] W. Li, X. Cui, and M. Lu, "A robust graph optimization realization of tightly coupled GNSS/INS integrated navigation system for urban vehicles," *Tsinghua Sci. Technol.*, vol. 23, no. 6, pp. 724–732, Dec. 2018.
- [19] W. Wen, X. Bai, Y. C. Kan, and L.-T. Hsu, "Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10651–10662, Nov. 2019.
- [20] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "Comparison of extended Kalman filter and factor graph optimization for GNSS/INS integrated navigation system," 2020, *arXiv:2004.10572*.
- [21] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter," *Navigat.*, vol. 68, no. 2, pp. 315–331, Jun. 2021.
- [22] W. Wen, Y. C. Kan, and L.-T. Hsu, "Performance comparison of GNSS/INS integrations based on EKF and factor graph optimization," in *Proc. ION GNSS+, Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2019, pp. 3019–3032.
- [23] Q. Li, L. Zhang, and X. Wang, "Loosely coupled GNSS/INS integration based on factor graph and aided by ARIMA model," *IEEE Sensors J.*, vol. 21, no. 21, pp. 24379–24387, Nov. 2021.
- [24] S. Xin, X. Wang, J. Zhang, K. Zhou, and Y. Chen, "A comparative study of factor graph optimization-based and extended Kalman filter-based PPP-B2b/INS integrated navigation," *Remote Sens.*, vol. 15, no. 21, p. 5144, Oct. 2023.
- [25] N. Boguspayev, D. Akhmedov, A. Raskaliyev, A. Kim, and A. Sukhenko, "A comprehensive review of GNSS/INS integration techniques for land and air vehicle applications," *Appl. Sci.*, vol. 13, no. 8, p. 4819, Apr. 2023.
- [26] C. Jiang, Y. Chen, C. Chen, S. Chen, Q. Meng, Y. Bo, and J. Hyypä, "Cooperative smartphone GNSS/PDR for pedestrian navigation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 6, pp. 2301–2305, Jun. 2023.
- [27] C. Jiang, Y. Chen, C. Chen, and J. Hyypä, "Walking gaits aided mobile GNSS for pedestrian navigation in urban areas," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8499–8510, May 2023.
- [28] A. Hussain and L. E. Díez, "Comparison of FGO and KF for PDR-GNSS fusion on smartphone for diverse pedestrian movements," in *Proc. 13th Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2023, pp. 1–6.
- [29] X. Dong, G. Hu, B. Gao, Y. Zhong, and R. Wei, "Windowing-based factor graph optimization with anomaly detection using Mahalanobis distance for underwater INS/DVL/USBL integration," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–13, 2024.
- [30] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," in *Proc. IEEE Int. Symp. Intell. Signal Process.*, Aug. 2009, pp. 37–42.
- [31] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," *Analog Devices*, vol. 2, no. 2, pp. 1–6, 2002.
- [32] Android Developers. *Sensors Overview—Android Developers*. Accessed: Jun. 3, 2025. [Online]. Available: https://developer.android.com/develop/sensors-and-location/sensors/sensors_position
- [33] F. Zhang, *The Schur Complement and Its Applications*, vol. 4. Cham, Switzerland: Springer, 2006.
- [34] Android Open Source Project. *GNSSLogger*. Accessed: Aug. 2024. [Online]. Available: <https://developer.android.com/studio/debug/gnss/gnss-logger>
- [35] A. R. Jiménez, F. Seco, and J. Torres-Sospedra, "Tools for smartphone multi-sensor data registration and GT mapping for positioning applications," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–8.



AMJAD HUSSAIN MAGSI received the Bachelor of Engineering degree in electronic engineering and the master's degree in electronic systems engineering from the Mehran University of Engineering Technology, Jamshoro, in 2017 and 2022, respectively. He is currently pursuing the Ph.D. degree in engineering for the information society and sustainable development with the University of Deusto, Bilbao, Spain, under the supervision of Luis Enrique Díez Blanco. He is with Deusto Smart Mobility Research Team. During his bachelor's degree, he was an Erasmus Scholar with the University of Limerick, Ireland. From 2019 to 2022, he was a Laboratory Engineer with the Dawood University of Engineering and Technology, Karachi. His research interests include data fusion for developing location-based services, smartphone positioning, factor graph optimization, and machine learning theory and algorithms for localization and intelligent environments.



LUIS ENRIQUE DÍEZ BLANCO received the degree in telecommunications engineering from the University of Deusto, in 2005, the master's degree in communications technologies and systems from the Telecommunications Engineering School, Polytechnic University of Madrid, in 2012, and the Ph.D. degree in engineering from the University of Deusto, in 2019. From 2005 to 2011, he was a Senior IT Consultant with Everis. From 2013 to 2014, he was a Research Support Technician with the SOFTLAB Research Group, Carlos III University of Madrid (UC3M). From 2014 to 2022, he was a Researcher with DeustoTech, the Research Institute, Engineering Faculty, University of Deusto. He is currently an Assistant Professor with the Faculty of Engineering, University of Deusto. He is also a member of the Research Group "Deusto Smart Mobility." He has worked (leading some of them) on more than ten regional, national, and international research projects and contracts. He has co-authored 15 research manuscripts published in international journals and more than ten communications in international congresses. His research interests include signal processing and data fusion for the development of location-based services, ambient assisted living, and intelligent environments.

...

Chapter 5

Case Study III: Impact of GNSS Quality on Fusion Accuracy

5.1 Introduction

This study investigates smartphone pedestrian localization by fusing PDR with GNSS-PPK via FGO and Kf. The goal is to assess whether higher quality GNSS processing (PPK) combined via FGO improves accuracy over standard Single-Point Positioning (SPP) and KF fusion strategies on smartphones under semi-obstructed, urban-like conditions.

A Google Pixel 4 was used to collect raw GNSS and IMU data along a ~ 1.8 km path (two laps) in partial open-sky conditions. GNSS raw observations were post-processed with RTKLIB 2.4.3 to obtain PPK. A u-blox ZED-F9P RTK GNSS receiver provided ground truth.

5.2 Key Findings

- **FGO + PDR + PPK improves over FGO + PDR + SPP.** Using PPK within FGO reduced the 2D mean error by **22.5%** relative to FGO with SPP (FGO+PPK: ~ 3.29 m vs. FGO+SPP: ~ 4.25 m).
- **FGO outperforms KF under both SPP and PPK.** FGO produced lower errors than KF in both fusion architectures that is with PPK: FGO ~ 3.29 m vs. KF ~ 3.44 m; with SPP: FGO ~ 4.25 m vs. KF ~ 4.60 m).
- **Smartphone PPK remained meter-level.** Despite PPK post-processing, smartphone results were still in the ~ 2 – 3 m range due to a low fixed-ambiguity rate on the phone dataset (while the u-blox reference achieved a high fixed-solution rate).
- **Compute cost is high for batch FGO.** Batch FGO incurred substantial cumulative computation time, highlighting the need for windowed formulations for real-time use.

Publication P4

Published in: *Micromachines* (MDPI)



Article

Continuous High-Precision Positioning in Smartphones by FGO-Based Fusion of GNSS–PPK and PDR

Amjad Hussain Magsi ^{1,*} , Luis Enrique Díez ¹ and Stefan Knauth ² ¹ Faculty of Engineering, University of Deusto, Avda. Universidades 24, 48007 Bilbao, Spain; luis.enrique.diez@deusto.es² Hochschule für Technik Stuttgart, Faculty of Computer Science, Geomatics and Mathematics, Schellingstraße 24, 70174 Stuttgart, Germany

* Correspondence: amjad.hussain@deusto.es

Abstract: The availability of raw Global Navigation Satellites System (GNSS) measurements in Android smartphones fosters advancements in high-precision positioning for mass-market devices. However, challenges like inconsistent pseudo-range and carrier phase observations, limited dual-frequency data integrity, and unidentified hardware biases on the receiver side prevent the ambiguity resolution of smartphone GNSS. Consequently, relying solely on GNSS for high-precision positioning may result in frequent cycle slips in complex conditions such as deep urban canyons, underpasses, forests, and indoor areas due to non-line-of-sight (NLOS) and multipath conditions. Inertial/GNSS fusion is the traditional common solution to tackle these challenges because of their complementary capabilities. For pedestrians and smartphones with low-cost inertial sensors, the usual architecture is Pedestrian Dead Reckoning (PDR)+ GNSS. In addition to this, different GNSS processing techniques like Precise Point Positioning (PPP) and Real-Time Kinematic (RTK) have also been integrated with INS. However, integration with PDR has been limited and only with Kalman Filter (KF) and its variants being the main fusion techniques. Recently, Factor Graph Optimization (FGO) has started to be used as a fusion technique due to its superior accuracy. To the best of our knowledge, on the one hand, no work has tested the fusion of GNSS Post-Processed Kinematics (PPK) and PDR on smartphones. And, on the other hand, the works that have evaluated the fusion of GNSS and PDR employing FGO have always performed it using the GNSS Single-Point Positioning (SPP) technique. Therefore, this work aims to combine the use of the GNSS PPK technique and the FGO fusion technique to evaluate the improvement in accuracy that can be obtained on a smartphone compared with the usual GNSS SPP and KF fusion strategies. We improved the Google Pixel 4 smartphone GNSS using Post-Processed Kinematics (PPK) with the open-source RTKLIB 2.4.3 software, then fused it with PDR via KF and FGO for comparison in offline mode. Our findings indicate that FGO-based PDR+GNSS–PPK improves accuracy by 22.5% compared with FGO-based PDR+GNSS–SPP, which shows smartphones obtain high-precision positioning with the implementation of GNSS–PPK via FGO.

Keywords: factor graph optimization; fusion; GNSS; Kalman Filter; PDR; post-processing kinematics; smartphone



Citation: Magsi, A.H.; Díez, L.E.; Knauth, S. Continuous High-Precision Positioning in Smartphones by FGO-Based Fusion of GNSS–PPK and PDR. *Micromachines* **2024**, *15*, 1141. <https://doi.org/10.3390/mi15091141>

Academic Editors: Changhui Jiang, Shuai Chen, Yuwei Chen and Jianxin Jia

Received: 10 August 2024

Revised: 2 September 2024

Accepted: 9 September 2024

Published: 11 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Global Navigation Satellite System (GNSS) provides Positioning, Navigation, and Timing (PNT) services across diverse fields such as transportation, autonomous vehicles, disaster forecasting, aerospace, and defense. Since these services have traditionally relied on specialized expertise, costly and high-precision professional GNSS receivers. Therefore, their widespread use has been limited. However, nowadays, chip-size GNSS receivers have empowered devices like smartphones, tablets, smartwatches, and other portable gadgets to provide positioning services broadly. The use of smartphones for

pedestrian positioning and navigation has become more widespread due to their mass availability and ease of access for individuals [1,2].

Before 2016, Android smartphones only provided position and velocity data, limiting research on smartphone-based positioning [3]. In May 2016, Google enabled access to raw GNSS data via the Android API, allowing for advancements in high-precision positioning. However, access to raw GNSS observations from smartphones does not solve the problem of high-accuracy localization systems via smartphones, as they are typically equipped with low-cost, poorly performed polarized GNSS antennas and unknown hardware biases on the receiver, resulting in noisy data and frequent cycle slips, especially in GNSS-challenging areas such as urban canyons, complex environments, tunnels, underpasses, etc., where these issues typically arise due to multipath interference and non-line-of-sight (NLOS) conditions [4,5]. Additionally, PNT signals from satellites are considerably impacted by various errors stemming from atmospheric conditions, including ionospheric delays, dry and wet tropospheric delays, as well as satellite clock offsets and ephemeris inaccuracies [6]. Therefore, the All Source Positioning and Navigation (APNS) program under the USA-based Defense Advanced Research Projects Agency (DARPA) has noted that single-source navigation is no longer reliable, prompting researchers to explore alternative navigation solutions [7,8]. This led researchers to fuse alternative navigation solutions to mitigate these problems. Different sensors are already available in modern smartphones, but among these, the Microelectrical Mechanical Systems (MEMS) Inertial Measurement Unit (IMU) holds significant importance due to its complementary behavior with GNSS, offering low errors in the short term and a high update rate without requiring additional infrastructure in the environment [9]. MEMS IMUs face limitations due to their high noise output, preventing the development of Strapdown Inertial Navigation Systems (SINS) and their integration with GNSS. This accumulates errors that are increased with the elapsed time, while the PDR system accumulates errors based on the detected steps, as they rely on pedestrian walking patterns [10]. Therefore, for the pedestrian navigation context with smartphones, PDR+GNSS fusion is preferable.

The fusion of PDR and GNSS in smartphones is still an open research topic. These fusion architectures mainly involve several steps, such as integration levels, fusion algorithms and frequencies, constellation, and GNSS processing techniques, as shown in Figure 1. Most of the researchers used KF and its variant models as the main fusion technique for these fusion architectures, such as Jiang et al. [11], who used the Krill Herd Algorithm (KHA) to optimize particle filters for GNSS/PDR integration, enhancing positioning accuracy. Zhu et al. [12] improved GNSS/PDR fusion positioning based on smartphone IMU error analysis. Zhang et al. [13] assessed GNSS/PDR fusion on smartphones in complex settings. Yan et al. [14] modified the KF using varied sampling frequencies from smartphone GNSS and IMU data for fusion. However, achieving better accuracy with a KF is challenging due to its first-order Markov assumption, Gaussian noise assumption, and reliance on the most recent observation.

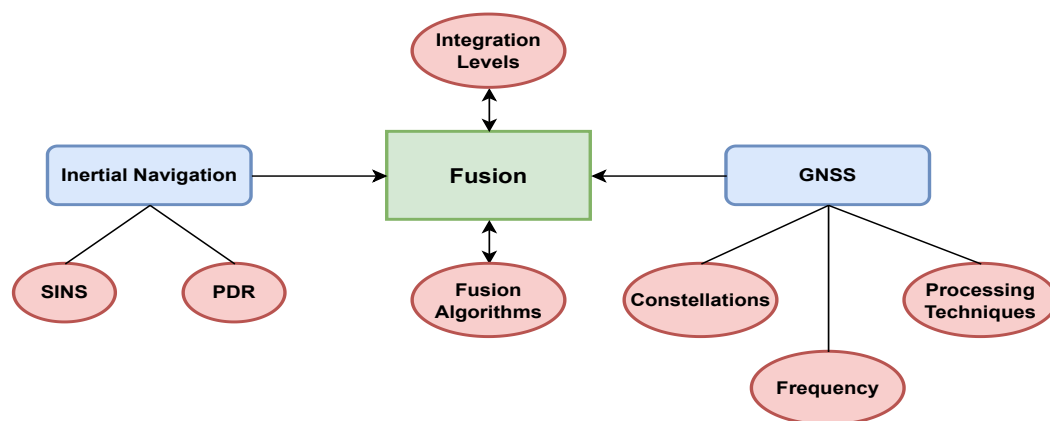


Figure 1. Inertial+GNSS fusion architectures.

Recently, FGO-based multi-sensor fusion has gained attraction among researchers. Initially, FGO demonstrated superior accuracy in SLAM-based complex problems in robotics and has been applied to INS/GNSS fusion architectures as well [15–22]. However, in the pedestrian navigation domain, significant analysis is still needed, and the application of FGO is in its early stages, presenting open research opportunities. Recently, Jiang et al. [23] implemented FGO for PDR+GNSS for a single walking dataset in open-sky conditions and compared its performance against KF. Later, ref. [24] introduced a PDR/GNSS integration method using FGO to boost accuracy in challenging GNSS conditions. Later, ref. [25] tested FGO for PDR/GNSS on smartphones by placing the smartphone in different positions, such as box, handheld, and pocket, and tested the fusion performance with open sky and urban canyon conditions. In our previous paper [26], we compared the performance of FGO and KF under two different motion scenarios (walking and running) in partially open sky conditions. We found that FGO performance deteriorated with varying motions, with running significantly affecting FGO performance more than walking. This degradation was attributed to differences in motion parameters such as step length, stride length, and speed. Therefore, in another paper [27], we conducted a more detailed analysis of FGO by examining a range of pedestrian motions from different individuals, including variations in gender, gait patterns, and heights. Additionally, we investigated how transient variations in how the smartphone is held affect errors in the PDR block. We then evaluated FGO's performance against these errors and compared it with KF and Smooth KF under these sources of PDR errors.

To the best of our knowledge, on the one hand, no work has tested the fusion of GNSS PPK and PDR on smartphones. And, on the other hand, the works that have evaluated the fusion of GNSS and PDR by means of FGO have always performed it using the GNSS SPP technique. Therefore, the aim of this work is to combine the use of the GNSS PPK technique and the FGO fusion technique to evaluate the improvement in accuracy that can be obtained on a smartphone compared with the usual GNSS SPP and KF fusion strategies.

The remaining sections detail the methodology, covering the PDR development and PDR+GNSS (SPP/PPK) fusion architectures via KF and FGO. The experimental evaluation section outlines the experiment's description, setup, and results, followed by the conclusion and future work section.

2. Methodology

The PDR development and its fusion with GNSS–SPP and GNSS–PPK by KF and FGO are presented in this section.

2.1. PDR

By using the number of detected steps, step length, and heading estimation calculation, the PDR can be developed, and its mechanism shown in Figure 2 can be mathematically written as follows:

$$\mathbf{Pos}_{k+1} = \mathbf{Pos}_k + \Delta\mathbf{Pos}_{k|k+1} \quad (1)$$

In this equation, $\mathbf{Pos}_{k+1}$ and $\mathbf{Pos}_k$ show position at steps $k + 1$ and k , respectively, and $\Delta\mathbf{Pos}_{k|k+1}$ is a positional increment from k to $k + 1$, which can be formulated as follows in east and north directions.

$$\Delta\mathbf{Pos}_{k|k+1} = \begin{bmatrix} \Delta\mathbf{Pos}_{k|k+1}^E \\ \Delta\mathbf{Pos}_{k|k+1}^N \end{bmatrix} = \begin{bmatrix} \mathbf{SL}_{k|k+1} \cdot \cos(\phi) \\ \mathbf{SL}_{k|k+1} \cdot \sin(\phi) \end{bmatrix} \quad (2)$$

ϕ is the heading angle, and $\mathbf{SL}_{k|k+1}$ is the step length between two steps.

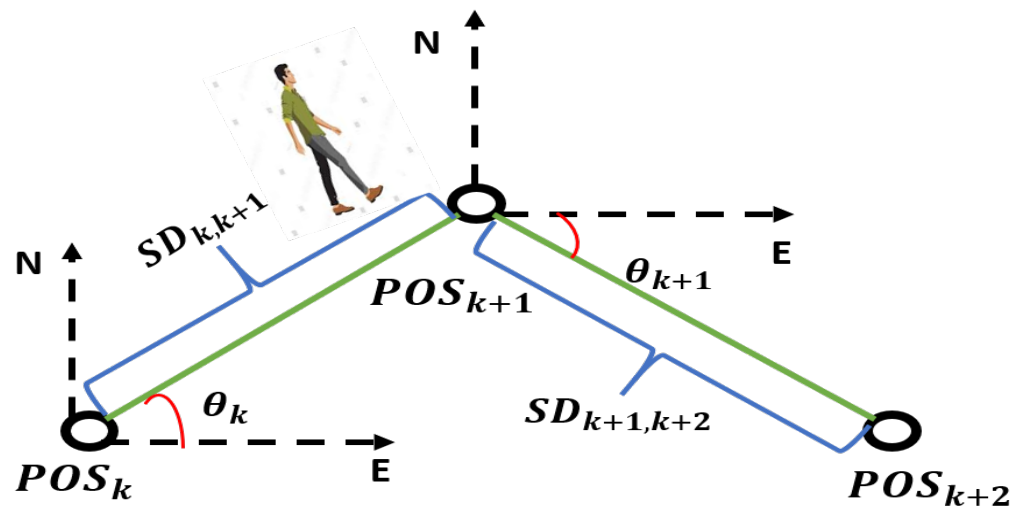


Figure 2. PDR mechanism.

The steps taken were detected by measuring the peak of acceleration magnitude, which can be written as

$$Acc_{mag_k} = \sqrt{Acc_{x_k}^2 + Acc_{y_k}^2 + Acc_{z_k}^2} \quad (3)$$

where Acc_{mag_k} denotes the magnitude of the triaxial accelerometer, and Acc_{x_k} , Acc_{y_k} , and Acc_{z_k} denote the accelerometer measurements in x , y , and z directions at time step k .

The step length was estimated using Weinberg's algorithm [28], which can be mathematically formulated as follows:

$$SL_{k|k+1} = C \cdot [\max(Acc_{mag_k}) - \min(Acc_{mag_k})]^{1/4} \quad (4)$$

$SL_{k|k+1}$ denotes the estimated step length between two successive acceleration peaks, while C is a constant factor to adjust the estimated step length, and the step heading was estimated via the Attitude and Heading Reference System (AHRS) from the smartphone. The factor value of C was determined through empirical testing. We conducted experiments by adjusting C under various conditions to see how it influenced the accuracy and consistency of the results. Through this process, we identified a value for C that yielded the best performance for our specific application.

2.2. GNSS

Modern Android smartphones can now provide raw GNSS observations, including pseudorange measurements, carrier phase data, and navigation messages. This capability allows for high-accuracy positioning directly on smartphones by processing these raw data to minimize associated errors [29–31]. In this study, we have utilized Post-Processing Kinematic (PPK) and compared it with Single-Point Positioning (SPP). A brief description of these techniques is provided below.

Single-Point Positioning (SPP): Also known as standalone positioning, it is a straightforward method that relies solely on the code-phase pseudorange measurements to determine a receiver's position. This method does not incorporate the more precise carrier phase measurements and typically does not apply error corrections [32,33]. The basic pseudorange equation used in SPP is as follows:

$$\rho_i = c(t_r - t_s) + d_{orb} + d_{iono} + d_{trop} + \epsilon \quad (5)$$

where

- ρ_i is the pseudorange from satellite i ;

- c is the speed of light;
- t_r and t_s are the receiver and satellite clock biases, respectively;
- d_{orb} represents satellite orbit errors;
- d_{iono} and d_{trop} are the ionospheric and tropospheric delays;
- ϵ encompasses various noise sources and multipath errors.

In SPP, the receiver calculates its position by solving for the location using pseudorange data from multiple satellites. However, due to the nature of code–phase measurements, this method is subject to several sources of error:

- **Clock offsets:** the receiver’s clock is less accurate compared with the atomic clocks on GNSS satellites, leading to timing errors.
- **Ephemeris errors:** inaccuracies in the satellite’s transmitted position.
- **Ionospheric and tropospheric delays:** signal propagation is affected by the Earth’s atmosphere, causing delays.
- **Multipath and NLOS (non-line-of-sight):** signals reflecting off surfaces before reaching the receiver, leading to errors in the measured pseudorange.

Because SPP does not correct for these errors and relies only on code data, the resulting position estimates are typically less accurate, often with errors in the range of several meters.

Post-Processed Kinematics (PPK): In contrast, PPK positioning is a more sophisticated method that uses carrier phase measurements, which are significantly more accurate than code–phase measurements. PPK applies a series of corrections to the raw GNSS data to achieve better accuracy, which is basically a post-processed RTK GNSS processing technique [34,35]. We opted for this approach because the SAPOS base station services in Germany did not provide real-time correction data. Instead, we had to download the RINEX observation files from the base station and process them afterward using RTKLIB Explorer. Additionally, we applied several other corrections, including ionospheric and tropospheric delay corrections, ambiguity fixing ratios, and adjustments to elevation angles. The basic equation for carrier phase measurement is as follows:

$$\Phi_i = \rho_i + \lambda N_i + c(t_r - t_s) + d_{orb} + d_{iono} + d_{trop} + \epsilon_\Phi \quad (6)$$

where

- Φ_i is the carrier phase measurement;
- λ is the carrier wavelength;
- N_i is the integer ambiguity (number of whole carrier wavelengths between the satellite and the receiver);
- Other terms are similar to those in the pseudorange equation, with ϵ_Φ representing the measurement noise specific to carrier phase.

Key aspects of PPK include the following:

- **Ambiguity resolution:** PPK involves resolving the integer ambiguity N_i , which is crucial for achieving high precision. This often requires a reference station (a nearby base station) with known coordinates to help resolve these ambiguities.
- **Error corrections:** PPK applies corrections for ionospheric and tropospheric delays, satellite clock errors, and ephemeris errors. These corrections are typically derived from precise models or additional reference stations.

By correcting for these errors and resolving ambiguities, PPK can achieve positioning accuracy down much better than SPP.

2.3. PDR Fusion with GNSS–SPP and PPK with KF in Smartphones

The fusion of PDR and GNSS (both SPP and PPK) using KF has been performed following an error state approach and a loosely coupled architecture. $\mathbf{X}$ can be the state vector for transitioning the state addressing position errors in both the x and y coordinates of PDR and can be written as follows:

$$\mathbf{X} = \begin{pmatrix} \Delta \text{PDR}_x \\ \Delta \text{PDR}_y \end{pmatrix}$$

The formulation of this transformation further can be written as below:

$$\mathbf{X}_{k+1} = \mathbf{F}_{k|k+1} \cdot \mathbf{X}_k + \mathbf{W}_{k+1} \quad (7)$$

In this equation, $\mathbf{X}_{k+1}$ and $\mathbf{X}_k$ are the state vectors. $\mathbf{W}_{k+1}$ is a process noise vector with the assumption of normal Gaussian distribution as $\mathbf{W}_{k+1} \sim \mathcal{N}(0, Q_k)$, where Q_k is the covariance matrix of the process noise and $\mathbf{F}_{k|k+1}$ is the transformation matrix at epochs $k+1$ and k , which can be expressed mathematically as

$$\mathbf{F}_{k|k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (8)$$

The measurement model is

$$\mathbf{Z}_{k+1} = \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1} + \mathbf{V}_{k+1} \quad (9)$$

where $\mathbf{Z}_{k+1}$ represents the measurement vector. $\mathbf{V}_{k+1}$ is the measurement noise vector with Gaussian distribution assumption expressed as $\mathbf{V}_{k+1} \sim \mathcal{N}(0, R)$, where R represents the covariance matrix of the measurement noise and $\mathbf{H}_{k+1}$ represents the observation matrix.

$$\mathbf{H}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (10)$$

The real observations represent the difference between PDR and GNSS–SPP for PDR+GNSS–SPP and PDR and GNSS–PPK for PDR+GNSS–PPK. Therefore, it can be rewritten and described as

$$\mathbf{Z}_{k+1} = \mathbf{Pos}_{k+1}^{\text{GNSS}} - \mathbf{Pos}_{k+1}^{\text{PDR}} \quad (11)$$

Prediction from KF will be as follows:

$$\mathbf{X}_{k+1}^- = \mathbf{F}_{k|k+1} \cdot \mathbf{X}_k \quad (12)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_{k|k+1} \cdot \mathbf{P}_k \cdot (\mathbf{F}_{k|k+1})^T + \mathbf{Q}_k \quad (13)$$

The prediction state $\mathbf{X}_{k+1}^-$ is derived via the transformation model. The matrix $\mathbf{P}_k$ denotes the predicted state covariance matrix at epoch k , while $\mathbf{P}_{k+1}^-$ represents the prediction at epoch $k+1$. Subsequently, the correction step of the Kalman filter updates the state estimation as follows:

$$\hat{\mathbf{X}}_{k+1} = \mathbf{X}_{k+1}^- + \mathbf{K}_{k+1} \cdot (\mathbf{Z}_{k+1} - \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1}^-) \quad (14)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \cdot (\mathbf{H}_{k+1})^T \cdot [\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1}^- \cdot (\mathbf{H}_{k+1})^T + \mathbf{R}_{k+1}]^{-1} \quad (15)$$

$$\mathbf{P}_{k+1} = (\mathbf{I}_{2 \times 2} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{P}_{k+1}^- \quad (16)$$

Here, $\hat{\mathbf{X}}_{k+1}$ denotes the revised state estimation at time step $k+1$, $\mathbf{K}_{k+1}$ symbolizes the Kalman gain matrix at time step $k+1$, and $\mathbf{I}_{2 \times 2}$ denotes the 2×2 identity matrix.

2.4. PDR Fusion with GNSS–SPP and GNSS–PPK with FGO in Smartphones

A factor graph serves as a probabilistic model that employs a graph structure to illustrate the probabilistic relationships among various variables. In essence, it consists of two distinct types of nodes: the unknown state variables and the factor nodes, which encode the conditional probabilities associated with certain sets of state variables. These nodes are connected by undirected edges, linking each state variable node to its corresponding

factor node. An edge exists between these nodes only if there's a functional dependence between the variable and factor. By employing a factor graph, one can break down a global probability function into the product of several local probability functions, thereby simplifying the resolution of the overall function.

In sensor integration, our focus typically lies on estimating a collection of state variables X based on a given set of measurements Z . Under the assumption of a first-order Markov model, we can express the conditional probability density function in the following manner:

$$P(X|Z) = \prod_{i=1}^k \frac{P(z_i|x_i)P(x_i|x_{i-1}, u_i)}{P(z_i)} P(x_0) \quad (17)$$

where z_i represents the measurements observed at epoch i (e.g., GNSS measurements), x_i represents the system state at i , and u_i denotes the control input (e.g., PDR measurements).

The most often used estimator for these unknown state variables X is the maximum a posteriori, or MAP estimate:

$$\hat{X} = \arg \max P(X|Z) = \arg \max \prod_{i=1}^k P(z_i|x_i)P(x_i|x_{i-1}, u_i) \quad (18)$$

In FGO-based integration, all these likelihood and transition probabilities are treated as the factorization of the global probability:

$$\hat{X} = \arg \max \prod_{j=1}^n f(x_j) \quad (19)$$

Each factor models a constraint and must include a measure of uncertainty. The most common model is a Gaussian noise:

$$P(z_i|x_i) \propto \exp\left(-\frac{1}{2}\|h_i(x_i - z_i)\|^2 \nabla_i\right) \quad (20)$$

$$P(x_i|x_{i-1}) \propto \exp\left(-\frac{1}{2}\|\Phi_i(x_{i-1} - x_i)\|^2 \Omega_i\right) \quad (21)$$

where the function $\Phi_i(\cdot)$ describes the relationship between the preceding states x_{i-1} and x_i , the function $h_i(\cdot)$ represents the relationship between the state x_i and the measurement z_i , and the covariance matrices are denoted by ∇_i and Ω_i .

Taking the negative log and dropping the factor $1/2$ transforms the problem into the minimization of an error function, which is a nonlinear least-squares problem.

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \|\Phi_i(x_{i-1}) - x_i\|_{\Omega_i}^2 + \sum_{i=1}^k \|h_i(x_i) - z_i\|_{\nabla_i}^2 \right) \quad (22)$$

In our case, there are two factors: PDR measurements and the GNSS measurements (GNSS-SPP in case of PDR+GNSS-SPP) and (GNSS-PPK in case of PDR+GNSS-PPK) fusion architectures as shown in Figure 3.

Factor by PDR position: this factor is obtained from the positions derived by the PDR block.

$$\begin{aligned} \|e_{k+1}^{PDR}\|_{\Sigma_{k+1}^{PDR}}^2 &= \|X_{k+1} - f(X_k)\|_{\Sigma_{k+1}^{PDR}}^2 \\ &= \left\| x_{k+1} - \left(x_k + \begin{bmatrix} \mathbf{S}\mathbf{L}_{t,t+1} \cdot \cos(\phi) \\ \mathbf{S}\mathbf{L}_{t,t+1} \cdot \sin(\phi) \end{bmatrix} \right) \right\|_{\Sigma_{k+1}^{PDR}}^2 \end{aligned} \quad (23)$$

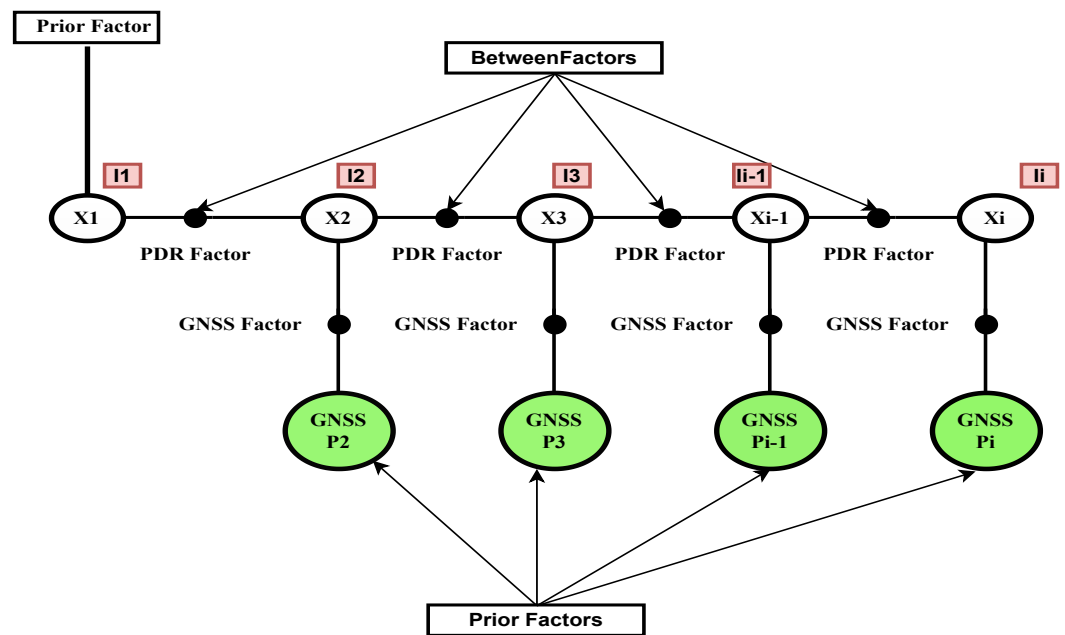


Figure 3. Factor graph for PDR+GNSS-SPP fusion architecture.

GNSS-SPP and PPK factor: GNSS-SPP factor is obtained by the smartphone raw GNSS observation with SPP solution.

$$\|e_{k+1}^{GNSS}\|_{\Sigma_{k+1}^{GNSS}}^2 = \|X_{k+1} - GNSS_{k+1}\|_{\Sigma_{k+1}^{GNSS}}^2 \quad (24)$$

Therefore,

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \left[\|e_{i+1}^{PDR}\|_{\Sigma_{i+1}^{PDR}}^2 + \|e_{i+1}^{GNSS}\|_{\Sigma_{i+1}^{GNSS}}^2 \right] \right) \quad (25)$$

The GNSS-PPK factor is created by replacing the GNSS with GNSS-PPK data.

We used the Georgia Tech Smoothing and Mapping (GTSAM) library [36] with the Levenberg–Marquardt (LM) optimizer [37].

3. Experimental Evaluation

3.1. Experimental Description

The experiment aimed to evaluate the reliability of the Google Pixel 4 smartphone's GNSS for high-precision localization by comparing its accuracy with that of a high-accuracy Ublox ZED F9P RTK GNSS receiver (Thalwil, Switzerland), which served as the ground truth. The Ublox antenna was placed on top of the pedestrian's head with a rod on the backpack, and the smartphone was held in a handheld position by the pedestrian, as shown in Figure 4. The experiment was conducted in Korntal Mönchingen, Stuttgart, Germany. A closed path of around 920 m, characterized by a partially open sky with trees and houses, was chosen for data collection, which can be seen in Figure 5. The data were collected at a normal walking speed by completing two laps (1.8 km) on the pre-defined path in approximately 27 min. A Google Pixel 4 smartphone was used to collect the smartphone's built-in GNSS and inertial sensor measurements. For ground truth, a Ublox ZED F9P RTK GNSS receiver was used, which is capable of providing centimeter-level accuracy. The pedestrian carried the smartphone in front of his chest, while the Ublox's ANN-MB-00 GNSS antenna was mounted on a backpack, as can be seen in Figure 4.



Figure 4. Data collection setup.



Figure 5. Difference between the ground truth, SPP-GNSS, and PPK-GNSS data.

The raw GNSS observations from the Android smartphone were logged using the GNSSLogger app [38], which conveniently allows data to be recorded directly in the RINEX format. For processing these observations with the PPK GNSS technique, we utilized the open-source RTKLIB Explorer2.4.3 software [39]. The RINEX observation file obtained from the smartphone contained numerous errors inherent to raw data, including satellite clock errors, inaccuracies in the satellite orbit and position, ionospheric delays, wet and dry tropospheric delays, receiver clock offsets, as well as multipath effects and non-line-of-sight (NLOS) errors caused by obstructions such as trees and nearby buildings.

To mitigate these errors, we employed the PPK technique, which applies corrections to the noisy observation file using data from a nearby base station. The RTKLIB software also offers built-in corrections for ionospheric and tropospheric delays, satellite clock errors,

and other necessary adjustments. In this study, we corrected the RINEX observation file by selecting signals from multiple GNSS constellations, including GPS, GLONASS, Galileo, and BeiDou. However, we disabled the SBAS correction system because our focus was on evaluating the fusion of PPK-based GNSS data with Pedestrian Dead Reckoning (PDR) rather than on SBAS corrections.

For satellite clocks and orbits, we relied on broadcast ephemerides rather than precise clock products, as our study did not focus on Precise Point Positioning (PPP). Ionospheric errors were corrected using the broadcast ionospheric model, with the impact of these errors further reduced by the use of dual-frequency observations, which are less susceptible to ionospheric disturbances compared with single-frequency data. For tropospheric corrections, we used an estimated Zenith Tropospheric Delay (ZTD) model, which is given by

$$\text{ZTD} = 0.0022768 \cdot \frac{P}{f(\phi, H)} + \text{ZWD}$$

where P is the atmospheric pressure at the receiver's location (in hPa). $f(\phi, H)$ is a function of the latitude ϕ and height H above sea level, accounting for the elevation and atmospheric conditions.

Carrier phase ambiguity resolution was implemented only for the GPS constellation, with ambiguity resolution disabled for other constellations. This selective ambiguity resolution allowed us to achieve integer-cycle resolution for GPS signals. These parameters, which are preconfigured in the RTKLIB software, were used to correct the observations. Additionally, we applied both forward and backward filtering, resulting in a combined filter approach that enhanced the accuracy and reliability of the corrections applied to the GNSS data. Moreover, smartphone low-cost chip size GNSS chip receivers and antennas are particularly susceptible to user-side errors such as clock biases, hardware inaccuracies, and antenna offsets [5,35]. Although we are aware of these issues, they were not addressed in this paper.

For processing the data, we utilized the L1, L2, and E5b frequencies. A 15-degree elevation mask was applied, with receiver dynamics enabled and tidal corrections disabled. For integer ambiguity resolution, we employed specific strategies for different satellite systems: continuous-resolution was used for GPS, while auto-calculation was selected for GLONASS (GLO) and BeiDou (BDS) was kept ON, as can be seen in Figure 6.

The figure displays two screenshots of the RTKLIB Options dialog box, showing settings for Setting1 and Setting2.

Setting1 Options:

- Positioning Mode: Kinematic
- Frequencies: L1+L2/E5b
- Filter Type: Combined
- Elevation Mask (°) / SNR Mask (dBHz): 15
- Rec Dynamics / Earth Tides Correction: ON
- Iono/Tropo Correction: Broadcast
- Satellite Ephemeris/Clock: Broadcast
- Excluded Satellites (+PRN: Included):
- GPS: ☒ GLO: ☒ Galileo: ☒ QZSS: ☐ SBAS: ☐ BeiDou: ☒ IRNSS: ☐

Setting2 Options:

- Integer Ambiguity Res (GPS/GLO/BDS): Continuum
- Ratio to Fix Ambiguity (Min/Nom/Max): 3 / 3 / 3
- GLO HW Bias: 0
- Min Lock / Elevation (°) to Fix Amb: 0 / 15
- Min Fix / Elevation (°) to Hold Amb: 20 / 15
- Slip Thresholds: Doppler (Hz) / Geom-Free (m): 0.000 / 0.050
- Max Age of Diff (s) / Outs to Reset Amb: 30.0 / 20
- Outlier Threshold for Code/Phase (m): 30.0 / 5.0
- # of Filter Iter / Sync Solution: 1 / ON
- Baseline Length Constraint (m): 0.000 / 0.000
- Min Fix Sats / Min Hold Sats: 4 / 5
- Min Drop Sats: 10
- Max Pos Var for AR / AR Filter: 0.1000 / ON
- Hold Amb Var / Hold Amb Gain: 0.1000 / 0.0100

Figure 6. RTKLIB settings.

These settings were chosen after testing various configurations, as they provided a more reliable fixed solution. Specifically, we set the ratio for fixing ambiguity at 3. The elevation mask for fixing ambiguities was also set at 15 degrees, with the minimum fix elevation and minimum elevation to hold ambiguity configured at 20 degrees and 15 degrees, respectively. Additionally, an outlier threshold of 5 m was established for carrier phase measurements.

The base station, located approximately 17 km from the experimental site, provided Radio Technical Commission for Maritime Services (RTCM) correction data through the Satellite Positioning Service (SAPOS) in Baden-Württemberg, Germany. The RTKLIB parameters that we used in the software settings described above for post-processing the smartphone raw GNSS observations are listed in Table 1. With these parameters, we achieved almost 11% fixed and 89% float solutions for smartphone GNSS. In contrast, through Ublox ZED F9P, the raw GNSS data were logged in its .ubx (Ublox) format. Therefore, we first converted that into RINEX format using the RTKConv tool from the RTKLIB. By processing this RINEX data with RTCM correction data of the same base station and with the same RTKLIB parameters as listed in Table 1, we yielded a fixed solution rate of 96.9%. The difference between the implementation of PPK on the datasets from Ublox and smartphone GNSS can be further seen in Figure 5. This high fixed solution rate supports the use of the Ublox receiver as ground truth, demonstrating its reliability and efficiency compared with the smartphone GNSS.

Table 1. RTKLIB parameters.

Positioning Mode	Kinematic
Filter Type	Combined
GNSS Constellations	Disable QZSS, IRNSS, and SBAS
Ionospheric Correction	Broadcast
Tropospheric Correction	Estimate ZTD
Satellite Clock	Broadcast
Integer AR	Continuous for GPS

Accurate synchronization of ground truth and smartphone data: The data from both devices were timely synchronized by maintaining the sampling frequency of both devices. Moreover, while the Ublox receiver and the smartphone were kept approximately one meter apart during data collection, the resulting positional offset was considered but not specifically corrected in this study. This choice was made to maintain a straightforward setup, with our primary focus on the fusion strategy between PDR and GNSS using the PPK technique.

KF and FGO parameters: The efficacy of the FGO and KF fusion algorithms hinges on their adept utilization of parameter covariance settings. Throughout this study, we preserved uniform parameter settings for both fusion techniques. Regarding the KF, we set the KF parameters as follows when we parametrize it in our case:

$$P = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \quad Q_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

where P is the initial state covariance matrix, Q_k is the process noise covariance matrix, and R is the measurement noise covariance matrix.

In the domain of FGO, our approach involves two factors: PDR and GNSS. It is important to recognize that both of these factors come with uncertainties and noise. We have used GTSAM's odometry and prior noise model classes to handle these uncertainties. We set the same values for both KF and FGO.


```
odometryNoise = noiseModel.Diagonal.Sigmas([2;2;0.1])
```

```
priorNoise = noiseModel.Diagonal.Sigmas([1.4;1.4;0.1])
```

3.2. Results

3.2.1. Position Error

In the semi-open sky condition, the SPP solution of smartphone GNSS was affected by the influence of trees and houses. This effect can be seen in Figure 7 in the form of horizontal position error peaks shown in red. Therefore, it can also be seen that the fusion of this GNSS–SPP with PDR from both KF and FGO is producing better results as compared with only GNSS–SPP by mitigating the peak errors of GNSS–SPP. However, we achieved lower positioning errors with FGO compared with KF. This is because of FGO’s iterative optimization of historical measurements, which results in improved accuracy, while KF relies on current and last measurements only. We computed the Cumulative Distribution Function (CDF) of these positioning errors obtained by FGO, KF, and GNSS–SPP, as shown in Figure 8, and from this, it is clear that FGO-based fusion provides overall better accuracy and precision than KF. It can also be observed from the fused trajectories presented in Figure 9 that FGO is more robust than KF.

For PPK, we enhanced smartphone raw GNSS measurements using RTCM messages from a base station, processed with RTKLIB software using various parameters shown in Table 1. Consequentially, GNSS error peaks were lower than SPP, as shown in Figure 10 by the red color. Fusing these GNSS–PPK with PDR produced better, smoother, and more precise output than PDR+GNSS–SPP by both fusion algorithms, KF and FGO. However, again, FGO produced better output than KF and only GNSS–PPK, which can be clearly seen from the CDF of horizontal position errors in Figure 8. The trajectories from FGO, KF, and GNSS–PPK can be seen in Figure 11.

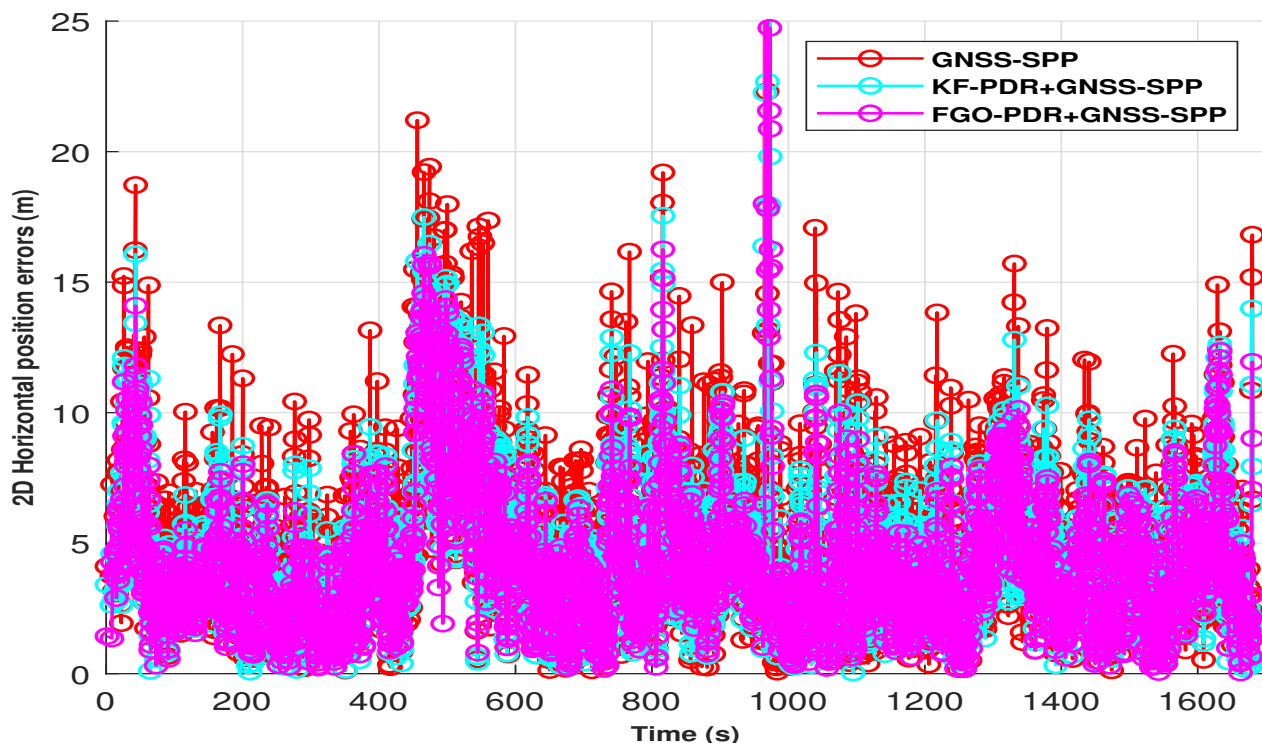


Figure 7. Two-dimensional horizontal positioning errors in PDR+GNSS–SPP fusion architecture.

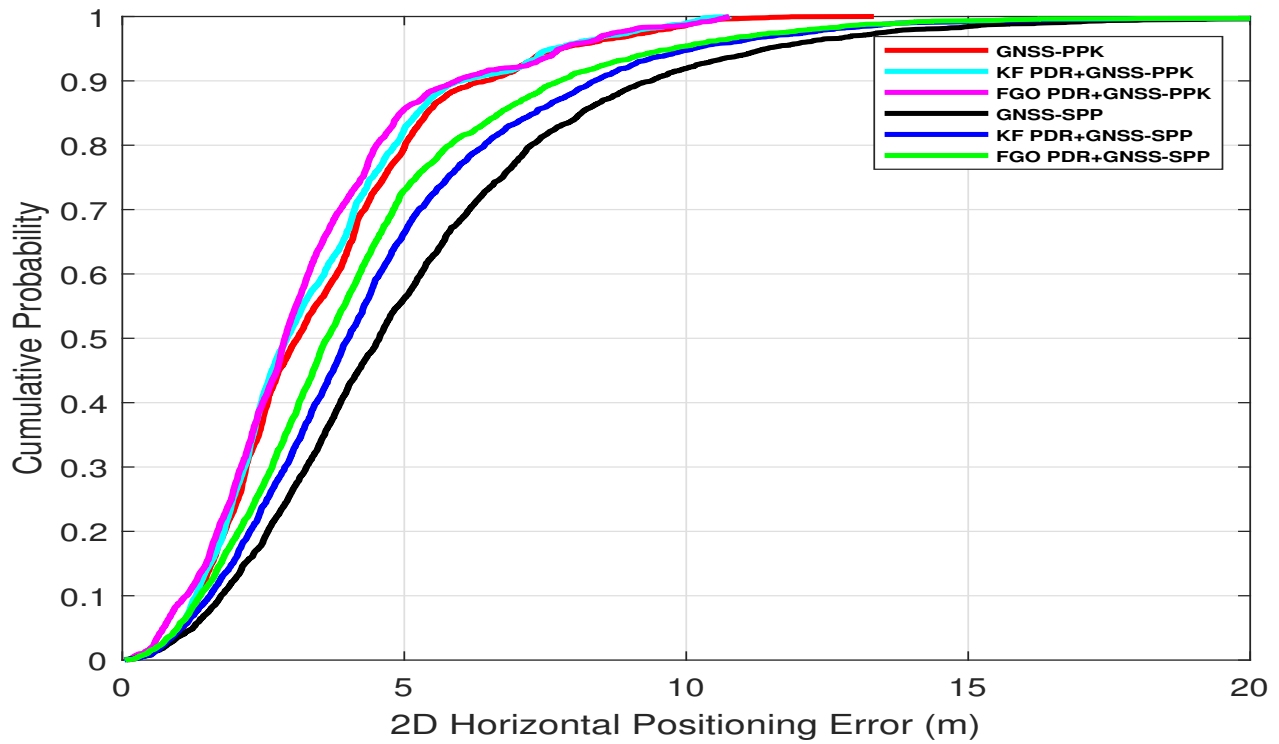


Figure 8. Combined overall CDF from both fusion architectures using FGO and KF.

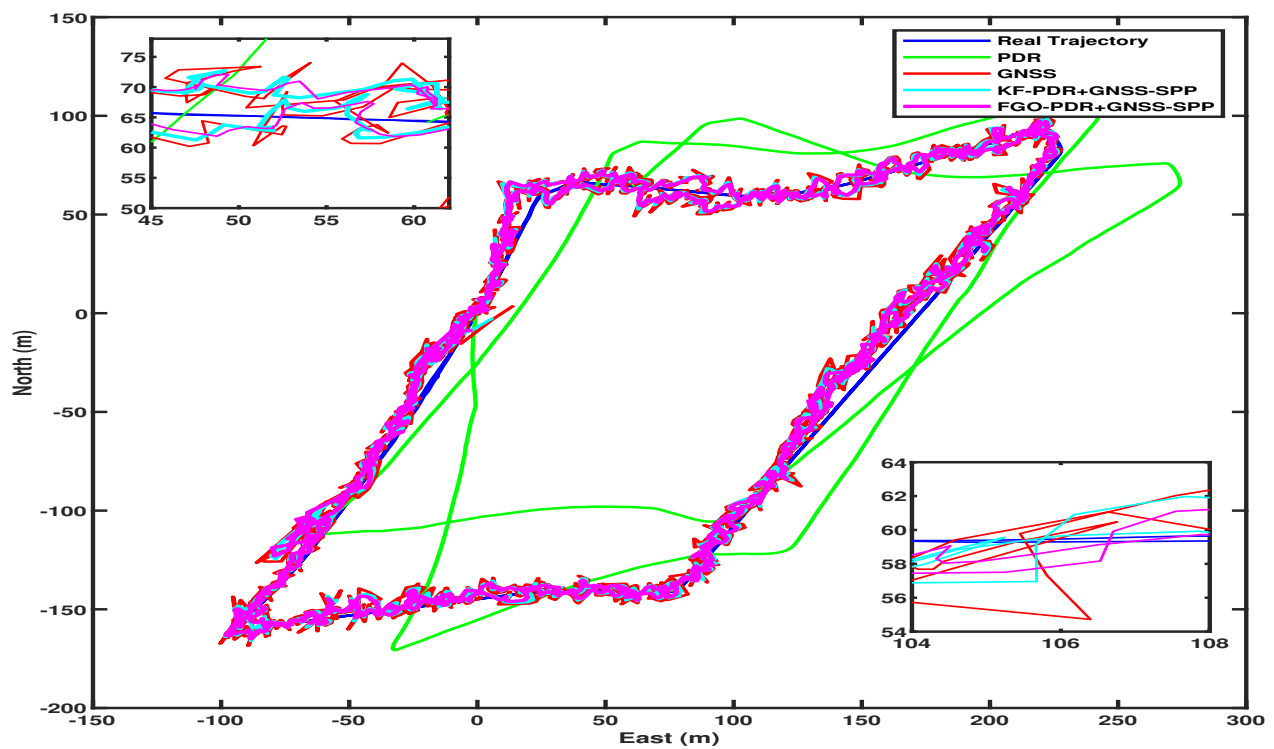


Figure 9. Comparison of trajectories in PDR+GNSS-SPP fusion architectures.

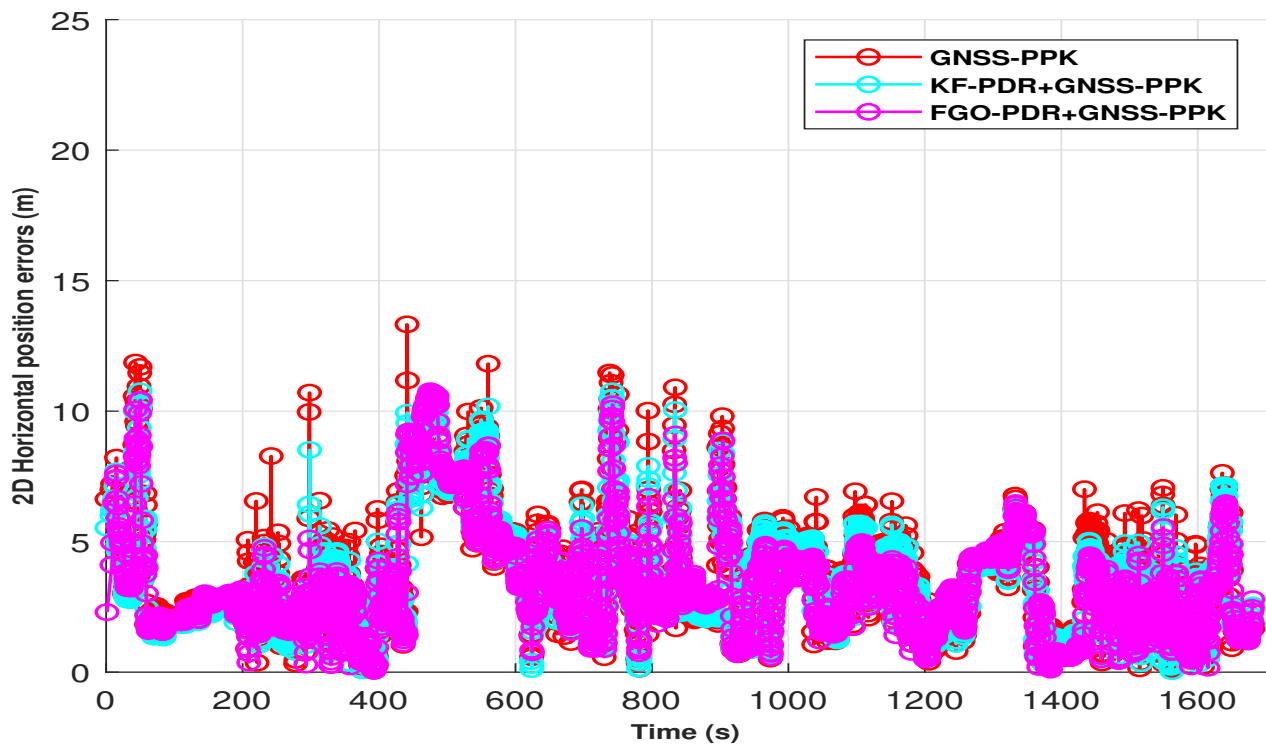


Figure 10. Two-dimensional horizontal positioning errors in PDR+GNSS-PPK fusion architecture.

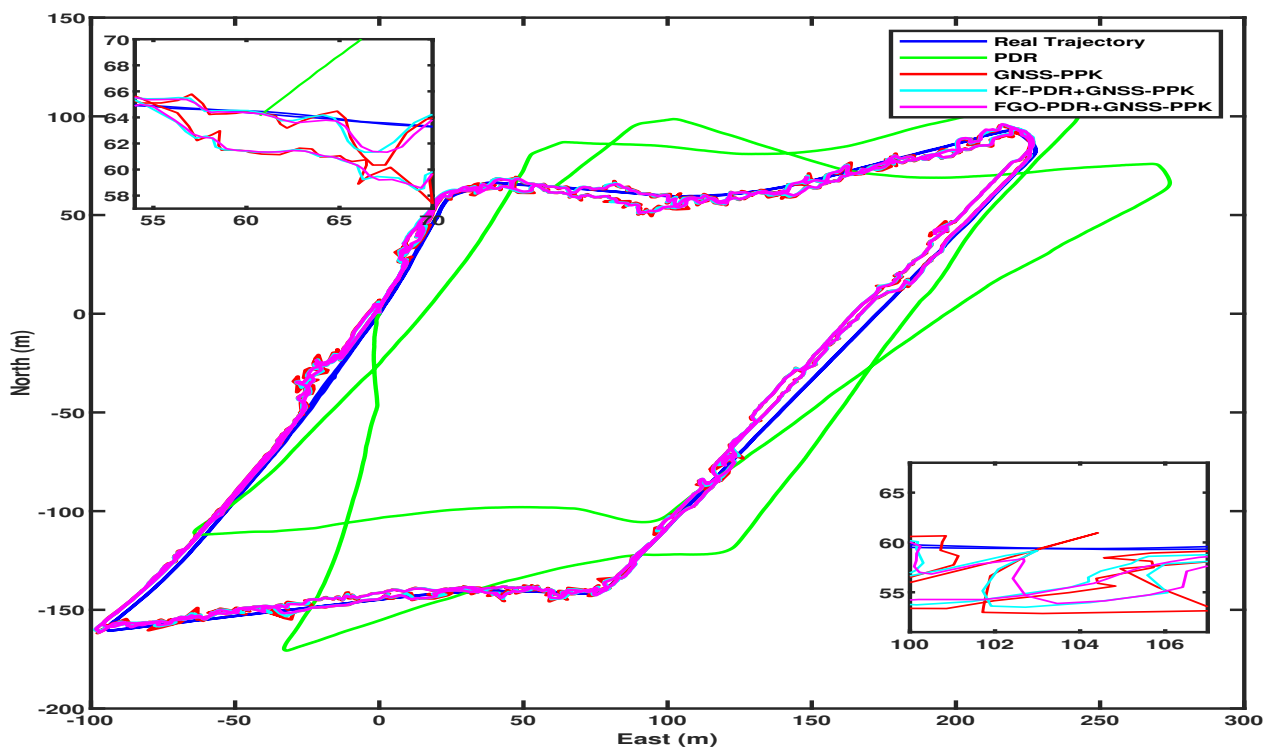


Figure 11. Comparison of trajectories in PDR+GNSS-PPK fusion architecture.

3.2.2. Computational Load

FGO-based PDR+GNSS-PPK has demonstrated superior positional accuracy compared with other methods discussed in this study. However, it is crucial to assess the computational demands of the FGO approach due to its iterative optimization process. FGO performs optimization at each epoch, incorporating the entire dataset, which can significantly impact computational efficiency. To evaluate the computational load, we

analyzed the performance of FGO at each epoch, measuring the amount of computational power utilized. This analysis includes both the per-epoch computational load and the cumulative computational load, which accounts for the computational resources used across all previous epochs. The results of this evaluation are illustrated in Figures 12 and 13. It can be noted that Figure 12 has some peaks at several epochs. This is because of unexpected variations in the data, such as noise or outliers, which require additional computations. The LM optimizer may also encounter conditions at specific epochs that demand more iterations or complex calculations to converge. Additionally, system-related overheads, like memory management or concurrent processes, could contribute to these temporary spikes. The slight increase in computational load per epoch might result from subtle changes in system state, solver efficiency, or the gradual accumulation of numerical errors over time.

Figure 13 shows that the computational load of FGO increases with each new epoch. This happens because FGO needs to optimize all factors from scratch at each epoch, which requires more computing power. By the end, the computational load has grown significantly, reaching around 5320 s. This increase is due to FGO's iterative process, which re-optimizes the entire dataset at every epoch. This repeated optimization and consideration of all factors makes FGO quite resource-intensive, which makes it less practical for real-time applications where efficiency is important.

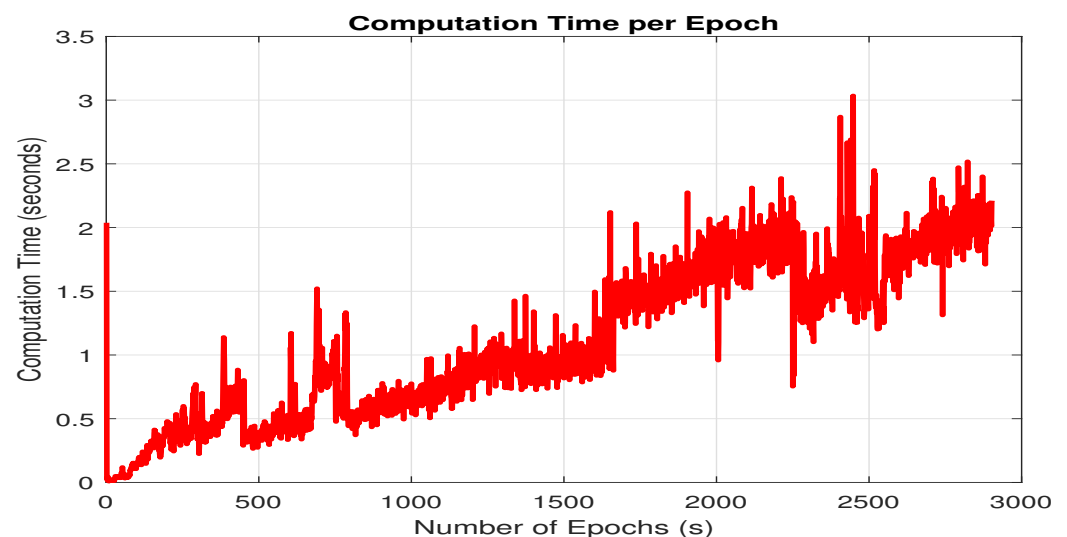


Figure 12. Computational time consumed by FGO-based PDR+GNSS-PPK fusion architecture.

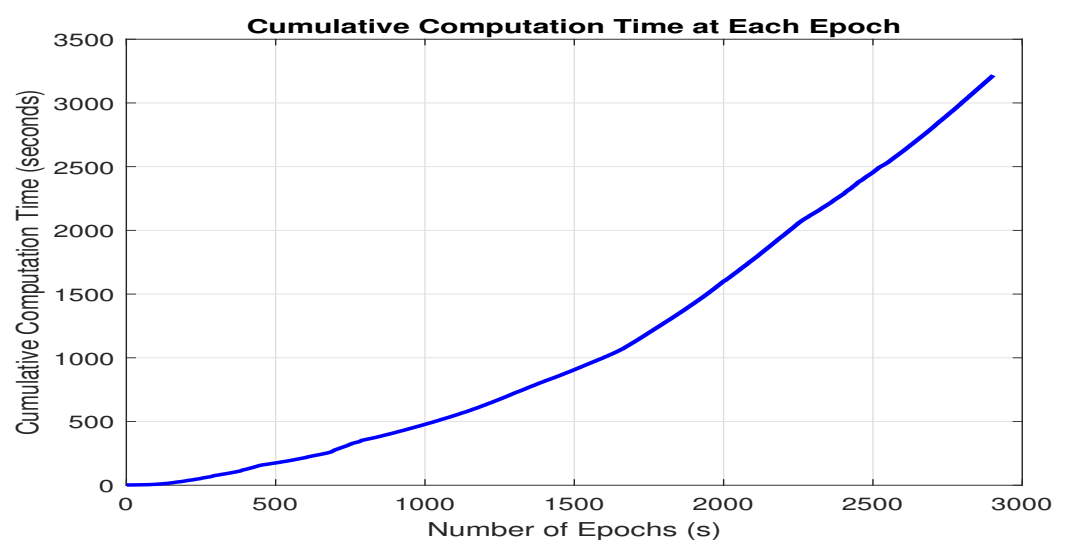


Figure 13. Cumulative computational time consumed by FGO-based PDR+GNSS-PPK fusion.

3.3. Discussion

To characterize the positional performance of PDR+GNSS-PPK with FGO, we computed the mean, median, and standard deviation (STD). Subsequently, we compared its mean % improvement with PDR+GNSS-SPP, GNSS-SPP, GNSS-PPK, KF-PDR+GNSS-SPP, and KF-PDR+GNSS-PPK, as detailed in Table 2. FGO with PDR+GNSS-PPK has achieved 22.5% better 2D mean error compared with FGO fusion with PDR+GNSS-SPP, which shows the higher capability of smartphone positioning in partial-open sky conditions, but still, for PPK, we have received positional errors in the range of 2–3 m. This is due to the fact that the fixed ambiguity solution from smartphone raw GNSS observation was very low, which is 11%. This is mainly due to the limitations of the smartphone's low-quality chip size GNSS receiver, which resulted in continuous cycle slips across all satellites for significant portions of the data and also resulted in hardware clock discontinuities at every epoch, likely caused by the phone's duty cycling of GNSS tracking during data collection, which can be seen in Figure 14, which shows simultaneous observations from a Google Pixel 4, where the red ticks indicate the flagged cycle slips. With few to no valid carrier phase observations in the degraded files, these solutions often exhibited very large position errors. Similarly, it was highlighted by other studies as well that a smartphone's GNSS performance is prone to several limitations [5,34,40–43].

Table 2. Two-dimensional error analysis.

Approach	GNSS Processing Techniques						% FGO Mean Improvement over GNSS and KF
	PPK			SPP			
	Mean Error	Median Error	STD	Mean Error	Median Error	STD	
GNSS (only)	3.56 m	3.11 m	2.10 m	5.23 m	4.55 m	3.56 m	7.28% (PPK) and 18.5% (SPP)
KF-PDR+GNSS	3.44 m	2.92 m	2.04 m	4.60 m	4.00 m	3.16 m	4.36% (PPK) and 7.60% (SPP)
FGO-PDR+GNSS	3.29 m	2.88 m	2.03 m	4.25 m	3.62 m	3.02 m	as compared with SPP 22.5%

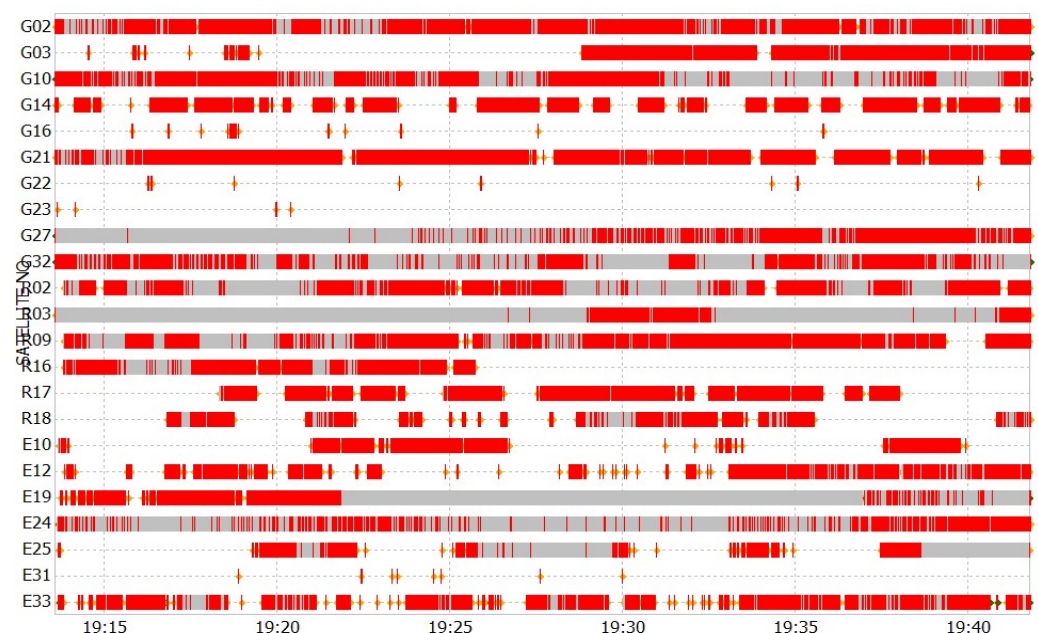


Figure 14. Google Pixel 4 raw GNSS observations.

Despite the limited corrections applied to the raw GNSS observations, we observed improved positioning accuracy compared with KF-based fusion. This enhancement is ad-

vantageous for various applications. However, FGO's computational performance is quite high, making it impractical for real-time applications. This limitation arises from FGO's batch processing mode, which optimizes the entire dataset in each iteration. A potential solution to manage this issue could be the use of a sliding window-based optimization approach. While our offline implementation showcases promising results, that can be useful for some applications. However, further examination is needed to assess its feasibility and practicality in diverse operational scenarios.

4. Conclusions and Future Work

Our study implemented GNSS–PPK and its fusion with PDR using FGO under partially open sky conditions, surrounded by homes and trees. We compared its performance with GNSS–SPP fusion with PDR via both FGO and KF, PDR+GNSS–PPK via KF, as well as only GNSS–SPP and GNSS–PPK. From the results obtained, we observe that for enhanced accuracy, the FGO-based PDR+GNSS–PPK offers a superior option for seamless localization. However, the computational load for FGO-based PDR+GNSS–PPK is much higher to be implemented in real-time in smartphones; the computational load at each epoch can be seen in Figure 12, and the overall accumulative computational time consumed can be seen in Figure 13. Moreover, it can be noted that PPK is a post-processed GNSS technique, which makes it unsuitable for real-time applications. Therefore, in the future, we plan to test our system in real-time by setting up our own base station. We believe that establishing a base station for real-time testing is a crucial next step. Consequently, we believe that the following works are essential for future research:

- (1) It is essential to explore the balance between computational load and the performance of FGO–PDR/GNSS systems. Implementing a sliding window for optimization could be a promising approach to reduce computational demands while preserving the system's performance.
- (2) We want to assess our fusion architecture's performance using more datasets in a variety of complex environments, including real-time smartphone locations and deep urban canyons, using sliding window-based FGO to maintain a healthy balance between computational complexity and accuracy.

Author Contributions: Conceptualization, A.H.M.; methodology, A.H.M. and L.E.D.; software RTKLIB Explorer 2.4.3, A.H.M.; validation, L.E.D.; writing—original draft preparation, A.H.M.; writing—review and editing, A.H.M., L.E.D., and S.K.; supervision, L.E.D. and S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported in part by the Research Training Grants Programme of the University of Deusto and in part by REPIN++ under Grant RED2022-134355-T.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FGO	Factor Graph Optimization
GNSS	Global Navigation Satellite Systems
IMU	Inertial Measurement Unit
KF	Kalman Filter
NLOS	Non-Line-of-Sight
PNT	Positioning Navigation and Timing
PDR	Pedestrian Dead Reckoning
SPP	Single-Point Positioning
PPK	Post-Processing Kinematics

References

- Paziewski, J. Recent advances and perspectives for positioning and applications with smartphone GNSS observations. *Meas. Sci. Technol.* **2020**, *31*, 091001. [\[CrossRef\]](#)
- Weng, D.; Chen, W.; Ji, S.; Wang, J. Intelligent Urban Positioning Using Smartphone-Based GNSS and Pedestrian Network. *IEEE Internet Things J.* **2024**, *11*, 22537–22549. [\[CrossRef\]](#)
- Banville, S.; Van Diggelen, F. Precise positioning using raw GPS measurements from Android smartphones. *GPS World* **2016**, *27*, 43–48.
- Specht, C.; Szot, T.; Dąbrowski, P.; Specht, M. Testing GNSS receiver accuracy in Samsung Galaxy series mobile phones at a sports stadium. *Meas. Sci. Technol.* **2020**, *31*, 064006. [\[CrossRef\]](#)
- Yun, J.; Lim, C.; Park, B. Inherent limitations of Smartphone GNSS Positioning and effective methods to increase the accuracy utilizing dual-frequency measurements. *Sensors* **2022**, *22*, 9879. [\[CrossRef\]](#)
- Karaim, M.; Elsheikh, M.; Noureldin, A.; Rustamov, R. GNSS error sources. *Multifunct. Oper. Appl. GPS* **2018**, *32*, 137–144.
- Burke, J. DARPA positioning, navigation, and timing (PNT) technology and their impacts on GPS users. In Proceedings of the 23rd Meeting Space-Based Positioning, Navigation, and Timing (PNT) Advisory Board, Alexandria, VA, USA, 6–7 June 2019.
- Zhuang, Y.; Sun, X.; Li, Y.; Huai, J.; Hua, L.; Yang, X.; Cao, X.; Zhang, P.; Cao, Y.; Qi, L.; et al. Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches. *Inf. Fusion* **2023**, *95*, 62–90. [\[CrossRef\]](#)
- Elmezayen, A.; Elsheikh, M.; Iqbal, U.; Noureldin, A. Examining the Capabilities of Smartphone-Based High Precision Positioning in Challenging Suburban Environments. In Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, CO, USA, 19–23 September 2022; pp. 2329–2338.
- Kuang, J.; Niu, X.; Chen, X. Robust pedestrian dead reckoning based on MEMS-IMU for smartphones. *Sensors* **2018**, *18*, 1391. [\[CrossRef\]](#)
- Jiang, C.; Chen, Y.; Liu, Z.; Xia, Q.; Chen, C.; Hyypä, J. A Probabilistic Method-Based Smartphone GNSS Fault Detection and Exclusion System Utilizing PDR Step Length. *Remote Sens.* **2023**, *15*, 4993. [\[CrossRef\]](#)
- Zhu, F.; Tao, X.; Liu, W.; Shi, X.; Wang, F.; Zhang, X. Walker: Continuous and precise navigation by fusing GNSS and MEMS in smartphone chipsets for pedestrians. *Remote Sens.* **2019**, *11*, 139. [\[CrossRef\]](#)
- Zhang, R.; Mi, J.; Li, J.; Wang, Q. A Continuous PDR and GNSS Fusing Algorithm for Smartphone Positioning. *Remote Sens.* **2022**, *14*, 5171. [\[CrossRef\]](#)
- Yan, W.; Zhang, Q.; Wang, L.; Mao, Y.; Wang, A.; Zhao, C. A modified kalman filter for integrating the different rate data of gyros and accelerometers retrieved from android smartphones in the GNSS/IMU coupled navigation. *Sensors* **2020**, *20*, 5208. [\[CrossRef\]](#)
- Li, W.; Cui, X.; Lu, M. A robust graph optimization realization of tightly coupled GNSS/INS integrated navigation system for urban vehicles. *Tsinghua Sci. Technol.* **2018**, *23*, 724–732. [\[CrossRef\]](#)
- Wen, W.; Bai, X.; Kan, Y.C.; Hsu, L.T. Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10651–10662. [\[CrossRef\]](#)
- Wen, W.; Pfeifer, T.; Bai, X.; Hsu, L.T. Comparison of extended Kalman filter and factor graph optimization for GNSS/INS integrated navigation system. *arXiv* **2020**, arXiv:2004.10572.
- Wen, W.; Pfeifer, T.; Bai, X.; Hsu, L.T. Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter. *NAVIGATION J. Inst. Navig.* **2021**, *68*, 315–331. [\[CrossRef\]](#)
- Wen, W.; Kan, Y.C.; Hsu, L.T. Performance comparison of GNSS/INS integrations based on EKF and factor graph optimization. In Proceedings of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019), Miami, FL, USA, 16–20 September 2019; pp. 3019–3032.
- Li, Q.; Zhang, L.; Wang, X. Loosely coupled GNSS/INS integration based on factor graph and aided by ARIMA model. *IEEE Sens. J.* **2021**, *21*, 24379–24387. [\[CrossRef\]](#)
- Xin, S.; Wang, X.; Zhang, J.; Zhou, K.; Chen, Y. A Comparative Study of Factor Graph Optimization-Based and Extended Kalman Filter-Based PPP-B2b/INS Integrated Navigation. *Remote Sens.* **2023**, *15*, 5144. [\[CrossRef\]](#)
- Boguspayev, N.; Akhmedov, D.; Raskaliyev, A.; Kim, A.; Sukhenko, A. A Comprehensive Review of GNSS/INS Integration Techniques for Land and Air Vehicle Applications. *Appl. Sci.* **2023**, *13*, 4819. [\[CrossRef\]](#)
- Jiang, C.; Chen, Y.; Chen, C.; Jia, J.; Sun, H.; Wang, T.; Hyypä, J. Implementation and performance analysis of the PDR/GNSS integration on a smartphone. *GPS Solut.* **2022**, *26*, 81. [\[CrossRef\]](#)
- Jiang, C.; Chen, Y.; Chen, C.; Chen, S.; Meng, Q.; Bo, Y.; Hyypä, J. Cooperative Smartphone GNSS/PDR for Pedestrian Navigation. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *70*, 2301–2305. [\[CrossRef\]](#)
- Jiang, C.; Chen, Y.; Chen, C.; Jia, J.; Sun, H.; Wang, T.; Hyypä, J. Smartphone PDR/GNSS Integration via Factor Graph Optimization for Pedestrian Navigation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 8504112. [\[CrossRef\]](#)
- Hussain, A.; Díez, L.E. Comparison of FGO and KF for PDR-GNSS Fusion on Smartphone for Diverse Pedestrian Movements. In Proceedings of the 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nuremberg, Germany, 25–28 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.

27. Magsi, A.H.; Díez, L.E. Comparison of FGO and KF for PDR-GNSS Fusion under Different PDR Errors. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 8506312. [CrossRef]
28. Weinberg, H. Using the ADXL202 in pedometer and personal navigation applications. *Analog Devices AN-602 Appl. Note* **2002**, *2*, 1–6.
29. Li, G.; Geng, J. Characteristics of raw multi-GNSS measurement error from Google Android smart devices. *GPS Solut.* **2019**, *23*, 90. [CrossRef]
30. Fu, G.M.; Khider, M.; Van Diggelen, F. Android raw GNSS measurement datasets for precise positioning. In Proceedings of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020), Online, 22–25 September 2020; pp. 1925–1937.
31. Lachapelle, G.; Gratton, P.; Horreli, J.; Lemieux, E.; Broumandan, A. Evaluation of a low cost hand held unit with GNSS raw data capability and comparison with an android smartphone. *Sensors* **2018**, *18*, 4185. [CrossRef]
32. Jiménez-Martínez, M.J.; Farjas-Abadia, M.; Quesada-Olmo, N. An approach to improving GNSS positioning accuracy using several GNSS devices. *Remote Sens.* **2021**, *13*, 1149. [CrossRef]
33. Paziewski, J.; Pugliano, G.; Robustelli, U. Performance assessment of GNSS single point positioning with recent smartphones. In Proceedings of the IMEKO TC-19 International Workshop on Metrology for the Sea, Naples, Italy, 5–7 October 2020; pp. 5–7.
34. Everett, T.; Taylor, T.; Lee, D.K.; Akos, D.M. Optimizing the use of RTKLIB for smartphone-based GNSS measurements. *Sensors* **2022**, *22*, 3825. [CrossRef]
35. Darugna, F.; Wübbena, J.; Ito, A.; Wübbena, T.; Wübbena, G.; Schmitz, M. RTK and PPP-RTK using smartphones: From short-baseline to long-baseline applications. In Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019), Miami, FL, USA, 16–20 September 2019; pp. 3932–3945.
36. Dellaert, F. *Factor Graphs and GTSAM: A Hands-On Introduction*; Technical Report; Georgia Institute of Technology: Atlanta, GA, USA, 2012.
37. Gavin, H.P. *The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems*; Department of Civil and Environmental Engineering Duke University: Durham, NC, USA, 2019; Volume 19.
38. GNSSLogger. 2016. Available online: <https://github.com/google/gps-measurement-tools> (accessed on 15 May 2024).
39. Takasu, T. RTKLIB: An Open Source Program Package for GNSS Positioning. Version 2.4.3. 2023. Available online: <https://www.rtklib.com> (accessed on 15 May 2024).
40. Zangenehjad, F.; Gao, Y. GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives. *Satell. Navig.* **2021**, *2*, 1–23. [CrossRef]
41. Wu, Q.; Sun, M.; Zhou, C.; Zhang, P. Precise point positioning using dual-frequency GNSS observations on smartphone. *Sensors* **2019**, *19*, 2189. [CrossRef]
42. Glaner, M.F.; Weber, R. Breaking the one-meter accuracy level with smartphone GNSS data. *Eng. Proc.* **2023**, *54*, 16. [CrossRef]
43. Takasu, T.; Yasuda, A. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In Proceedings of the International Symposium on GPS/GNSS, Seogwipo-si, Republic of Korea, 4–6 November 2009; Volume 1, pp. 1–6.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Chapter 6

Conclusions and Future Works

The conclusions presented in this chapter pertain exclusively to smartphone-based PDR+GNSS fusion, as investigated throughout this thesis. This chapter synthesizes the research outcomes presented in the preceding publications, unifying them into an overarching perspective on the use of FGO for pedestrian localization on smartphones. In particular, it emphasizes the research gap that FGO has been little explored for the fusion of PDR and GNSS data, where most existing studies still rely on traditional filtering methods. The motivation for this work stemmed from the growing demand for robust, accurate, and adaptive localization in real-world pedestrian scenarios, particularly in urban environments with frequent signal degradation and interruptions. Across the included studies, multiple fusion architectures, error scenarios, and computational strategies were investigated to assess and enhance the performance of FGO in comparison to traditional filtering methods.

The results collectively confirm that FGO-based fusion consistently delivers better positional accuracy, resilience to transient errors, and better long-term trajectory consistency than KF-based approaches. The progression of work from comparing FGO with KF under different motion dynamics, to evaluating robustness against carrying mode disturbances, to optimizing computational efficiency via windowing strategies, and finally to integrating high-precision GNSS–PPK establishes a comprehensive framework for advancing smartphone-based pedestrian localization.

6.1 Main Findings

The main findings emerging from this thesis are:

- **Accuracy and robustness:** FGO-based fusion of PDR and GNSS outperforms KF and smoothed KF variants under a wide range of pedestrian motion types, including walking, running, and irregular gait patterns, maintaining higher accuracy and lower drift.
- **Resilience to disturbances:** FGO demonstrates stronger recovery from transient and continuous PDR errors caused by variations in smartphone carrying modes such as shaking, swinging, rotation and natural inter-user differences.

- **Computational efficiency:** Sliding-window strategies for FGO can significantly reduce processing cost. In particular, marginalization-based windowing preserves most of the accuracy benefits of batch FGO while improving real-time feasibility.
- **Continuity under degraded GNSS:** Using PPK position fixes as factors within an FGO fusion with PDR improved trajectory continuity and reduced drift during short GNSS degradations/outages. On smartphone data, accuracy remained at the meter level (limited by a low ambiguity-fix rate), but FGO still outperformed KF/SKF baselines.
- **Methodological flexibility:** The approaches developed are adaptable across different pedestrian contexts and GNSS quality levels, offering a practical pathway for deployment on mobile and wearable devices.

6.2 Limitations

Despite the promising results, several limitations remain:

- PDR still exhibits errors in step detection and step length estimation, especially under fast or irregular movements.
- FGO’s computational requirements remain non-trivial, potentially challenging for low-power or resource-constrained devices without further optimization.
- GNSS–PPK requires access to correction data, which may not always be available in real-time in certain environments.
- Most evaluations were conducted in semi-controlled or urban outdoor environments; broader testing in varied indoor, mixed, and densely obstructed scenarios is needed.

6.3 Future Research Directions

Building on this thesis, several avenues for further investigation are identified:

- Development of adaptive FGO models that dynamically adjust parameters such as window size and factor weighting based on motion dynamics or sensor quality.
- Real-time implementation of FGO on smartphones using optimized solvers or hardware acceleration to address computational constraints.
- Multi-sensor integration, incorporating visual-inertial odometry, magnetometers, or barometric sensors, to improve performance in GNSS-denied conditions.
- Hybrid localization strategies that fuse PDR and GNSS with other positioning technologies such as 5G, Wi-Fi RTT, or ultra-wideband.
- Large-scale field trials across diverse user groups and environments to assess scalability and long-term reliability.

6.4 Closing Remarks

The research presented in this compendium advances the state of the art in pedestrian localization by demonstrating that FGO can serve as a powerful and flexible fusion framework for combining inertial and satellite navigation data on smartphones. Through systematic experimentation and methodological innovation, this work has addressed both accuracy and practical deployment challenges, moving toward solutions that are robust, computationally viable, and adaptable to real-world pedestrian use cases.

The findings have broader implications for the development of mobile and wearable navigation systems capable of operating reliably in GNSS-challenged environments such as dense urban areas, transport hubs, and partially obstructed outdoor spaces. By bridging the gap between high-precision positioning techniques and the constraints of consumer-grade devices, this thesis contributes both to academic understanding and to the practical foundation for next-generation pedestrian navigation applications.

Bibliography

- [1] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, “Gnss position integrity in urban environments: A review of literature,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2762–2778, 2018.
- [2] United Nations Department of Economic and Social Affairs, “68% of the world population projected to live in urban areas by 2050,” UN DESA, 2018. [Online]. Available: <https://www.un.org/uk/desa/68-world-population-projected-live-urban-areas-2050-says-un>
- [3] W. Cox, “Demographia world urban areas, 17th annual edition,” Demographia, 2021. [Online]. Available: <https://www.newgeography.com/content/007127-focusing-world-megacities-demographia-world-urban-areas-2021>
- [4] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert, “Density-aware person detection and tracking in crowds,” in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2423–2430.
- [5] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.
- [6] M. Billinghurst, A. Clark, G. Lee *et al.*, “A survey of augmented reality,” *Foundations and Trends® in Human–Computer Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015.
- [7] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran, “Rfid in robot-assisted indoor navigation for the visually impaired,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 2. IEEE, 2004, pp. 1979–1984.
- [8] Q. Wang, M. Fu, J. Wang, H. Luo, L. Sun, Z. Ma, W. Li, C. Zhang, R. Huang, X. Li *et al.*, “Recent advances in pedestrian inertial navigation based on smartphone: A review,” *IEEE Sensors Journal*, vol. 22, no. 23, pp. 22 319–22 343, 2022.
- [9] B. Dean, “Smartphone usage statistics for 2025 (surprising),” Demand Sage, 2025, accessed: May 23, 2025. [Online]. Available: <https://www.demandsage.com/smartphone-usage-statistics/>

- [10] D. Weng, W. Chen, M. Ding, S. Liu, and J. Wang, "Sidewalk matching: a smartphone-based gnss positioning technique for pedestrians in urban canyons," *Satellite Navigation*, vol. 6, no. 1, p. 4, 2025.
- [11] J. Yun, C. Lim, and B. Park, "Inherent limitations of smartphone gnss positioning and effective methods to increase the accuracy utilizing dual-frequency measurements," *Sensors*, vol. 22, no. 24, p. 9879, 2022.
- [12] L. Wang, P. D. Groves, and M. K. Ziebart, "Urban positioning on a smartphone: Real-time shadow matching using gnss and 3d city models." The Institute of Navigation, 2013.
- [13] Inside GNSS, "Darpa seeks pnt robustness, flexibility with all source positioning and navigation program," Inside GNSS, 2012. [Online]. Available: <https://insidegnss.com/darpa-seeks-pnt-robustness-flexibility-with-all-source-positioning-and-navigation-program/>
- [14] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.
- [15] T.-M. T. Dinh, N.-S. Duong, and K. Sandrasegaran, "Smartphone-based indoor positioning using ble ibeacon and reliable lightweight fingerprint map," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 283–10 294, 2020.
- [16] K.-H. Lin and Z.-H. Tseng, "Enhancing coordinate accuracy in three-dimensional uwb indoor positioning systems," *IEEE Transactions on Consumer Electronics*, 2025.
- [17] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual slam: Applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.
- [18] M. Holmberg, O. Karlsson, and M. Tulldahl, "Lidar positioning for indoor precision navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 359–368.
- [19] A. Sheta, A. Mohsen, B. Sheta, and M. Hassan, "Improved localization for android smartphones based on integration of raw gnss measurements and imu sensors," in *2018 International conference on computer and applications (ICCA)*. IEEE, 2018, pp. 297–302.
- [20] G. Falco, M. Pini, and G. Maruccio, "Loose and tight gnss/ins integrations: Comparison of performance assessed in real urban scenarios," *Sensors*, vol. 17, no. 2, p. 255, 2017.
- [21] A. Angrisano *et al.*, "Gnss/ins integration methods," Ph.D. dissertation, 2010.
- [22] J. A. Farrell and J. Wendel, "Gnss/ins integration," in *Springer handbook of global navigation satellite systems*. Springer, 2017, pp. 811–840.

- [23] D. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.
- [24] C. Jiang, Y. Chen, C. Chen, J. Jia, H. Sun, T. Wang, and J. Hyypä, “Implementation and performance analysis of the pdr/gnss integration on a smartphone,” *GPS Solutions*, vol. 26, no. 3, p. 81, 2022.
- [25] C. Jiang, Y. Chen, C. Chen, J. Jia, H. Sun, T. Wang, and J. Hyypä, “Smartphone pdr/gnss integration via factor graph optimization for pedestrian navigation,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.
- [26] Y. Zhong, W. Wen, and L.-T. Hsu, “Trajectory smoothing using gnss/pdr integration via factor graph optimization in urban canyons,” *IEEE Internet of Things Journal*, vol. 11, no. 14, pp. 25 425–25 439, 2024.
- [27] A. Rehman, H. Shahid, M. Afzal, and H. Bhatti, “Accurate and direct gnss/pdr integration using extended kalman filter for pedestrian smartphone navigation,” *Gyroscope and Navigation*, vol. 11, no. 2, pp. 124–137, 2020.
- [28] J. Zhang, B. Yu, Y. Ge, J. Gao, and C. Sheng, “An effective gnss/pdr fusion positioning algorithm on smartphones for challenging scenarios,” *Sensors*, vol. 24, no. 5, p. 1452, 2024.
- [29] J. Uhlmann and S. Julier, “Gaussianity and the kalman filter: A simple yet complicated relationship,” *arXiv preprint arXiv:2405.00058*, 2024.
- [30] M. Jahja, D. Farrow, R. Rosenfeld, and R. J. Tibshirani, “Kalman filter, sensor fusion, and constrained regression: Equivalences and insights,” *Advances in neural information processing systems*, vol. 32, 2019.
- [31] C. Urrea and R. Agramonte, “Kalman filter: historical overview and review of its use in robotics 60 years after its creation,” *Journal of Sensors*, vol. 2021, no. 1, p. 9674015, 2021.
- [32] F. Dellaert, M. Kaess *et al.*, “Factor graphs for robot perception,” *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.
- [33] W. Li, X. Cui, and M. Lu, “A robust graph optimization realization of tightly coupled gnss/ins integrated navigation system for urban vehicles,” *Tsinghua Science and Technology*, vol. 23, no. 6, pp. 724–732, 2018.
- [34] W. Wen, X. Bai, Y. C. Kan, and L.-T. Hsu, “Tightly coupled gnss/ins integration via factor graph and aided by fish-eye camera,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10 651–10 662, 2019.
- [35] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, “Factor graph optimization for gnss/ins integration: A comparison with the extended kalman filter,” *NAVIGATION: Journal of the Institute of Navigation*, vol. 68, no. 2, pp. 315–331, 2021.

- [36] Q. Li, L. Zhang, and X. Wang, "Loosely coupled gnss/ins integration based on factor graph and aided by arima model," *IEEE Sensors Journal*, vol. 21, no. 21, pp. 24 379–24 387, 2021.
- [37] S. Xin, X. Wang, J. Zhang, K. Zhou, and Y. Chen, "A comparative study of factor graph optimization-based and extended kalman filter-based ppp-b2b/ins integrated navigation," *Remote Sensing*, vol. 15, no. 21, p. 5144, 2023.
- [38] N. Boguspayev, D. Akhmedov, A. Raskaliyev, A. Kim, and A. Sukhenko, "A comprehensive review of gnss/ins integration techniques for land and air vehicle applications," *Applied Sciences*, vol. 13, no. 8, p. 4819, 2023.
- [39] P. JG and O. Montenbruck, *Springer handbook of global navigation satellite systems*. Springer., 2017.
- [40] J. Zidan, E. I. Adegoke, E. Kampert, S. A. Birrell, C. R. Ford, and M. D. Higgins, "Gnss vulnerabilities and existing solutions: A review of the literature," *IEEE Access*, vol. 9, pp. 153 960–153 976, 2020.
- [41] Y. Zhou, J. Lai, X. Guo, and J. Yang, "A research on all source navigation and positioning and its critical technology," in *China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III*. Springer, 2015, pp. 801–808.
- [42] D. Tremper, "Relying solely on gps provides users with a single point of failure," *National Defense Magazine*, 2018. [Online]. Available: <https://www.nationaldefensemagazine.org/articles/2018/5/31/darpa-pursuing-global-positioning-system-alternatives>
- [43] Y. Zhuang, X. Sun, Y. Li, J. Huai, L. Hua, X. Yang, X. Cao, P. Zhang, Y. Cao, L. Qi *et al.*, "Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches," *Information Fusion*, vol. 95, pp. 62–90, 2023.
- [44] N. El-Sheimy and A. Youssef, "Inertial sensors technologies for navigation applications: State of the art and future trends," *Satellite navigation*, vol. 1, no. 1, p. 2, 2020.
- [45] A. Budiyo, "Principles of gnss, inertial, and multi-sensor integrated navigation systems," *Industrial Robot: An International Journal*, vol. 39, no. 3, 2012.
- [46] X. Hou and J. Bergmann, "Pedestrian dead reckoning with wearable sensors: A systematic review," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 143–152, 2020.
- [47] A. H. Magsi and L. E. Díez, "Comparison of fgo and kf for pdr-gnss fusion under different pdr errors," *IEEE Transactions on Instrumentation and Measurement*, 2024.

- [48] Y. Peng and W. A. Scales, "Ionospheric remote sensing with gnss," *Encyclopedia*, vol. 1, no. 4, pp. 1246–1256, 2021. [Online]. Available: <https://www.mdpi.com/2673-8392/1/4/94>
- [49] R. Gao, L. Xu, B. Zhang, and T. Liu, "Raw gnss observations from android smartphones: Characteristics and short-baseline rtk positioning performance," *Measurement Science and Technology*, vol. 32, no. 8, p. 084012, 2021.
- [50] D. Weng, Z. Hou, Y. Meng, M. Cai, and Y. Chan, "Characterization and mitigation of urban gnss multipath effects on smartphones," *Measurement*, vol. 223, p. 113766, 2023.
- [51] K. Su, S. Jin, and M. M. Hoque, "Evaluation of ionospheric delay effects on multi-gnss positioning performance," *Remote Sensing*, vol. 11, no. 2, p. 171, 2019.
- [52] P. Collins and R. B. Langley, "Tropospheric delay," *GPS world*, vol. 53, pp. 52–58, 1999.
- [53] O. Montenbruck, R. Schmid, F. Mercier, P. Steigenberger, C. Noll, R. Fatkulin, S. Kogure, and A. S. Ganeshan, "Gnss satellite geometry and attitude models," *Advances in Space Research*, vol. 56, no. 6, pp. 1015–1029, 2015.
- [54] O. L. Colombo, "Ephemeris errors of gps satellites," *Bulletin géodésique (1976-1995)*, vol. 60, no. 1, pp. 64–84, 1986.
- [55] K. Radoš, M. Brkić, and D. Begušić, "Recent advances on jamming and spoofing detection in gnss," *Sensors*, vol. 24, no. 13, p. 4210, 2024.
- [56] J. D. Gutiérrez, A. R. Jiménez, F. Seco, F. J. Álvarez, T. Aguilera, J. Torres-Sospedra, and F. Melchor, "Getsensordata: An extensible android-based application for multi-sensor data registration," *SoftwareX*, vol. 19, p. 101186, 2022.
- [57] Google, "Gnss logger application," Android Location Tools, accessed July 8, 2025. [Online]. Available: <https://developer.android.com/guide/topics/sensors/gnss#tools>
- [58] G. Wubben, M. Schmitz, A. Bagge, and G. Seeber, "Geo++ rinex logger app," Geo++ GmbH, 2018, accessed July 8, 2025. [Online]. Available: <https://www.geopp.de/geopp-rinex-logger/>
- [59] Nottingham Scientific Ltd., "rinexon app," FLAMINGO Project, 2018, accessed July 8, 2025. [Online]. Available: <https://www.flamingognss.com/>
- [60] P. Crosta, J. Paziewski, J. Samson, and G. Molino, "Introducing galileopvt: An open-source android app for gnss data collection," in *Proceedings of the European Space Agency GNSS Raw Measurements Task Force Workshop*. ESA, 2018.

- [61] H. Sharma, M. Bochkati, and T. Pany, "Time-synchronized gnss/imu data logging from android smartphone and its influence on the positioning accuracy," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 2000–2011.
- [62] S. M. Asaad and H. S. Maghdid, "A comprehensive review of indoor/outdoor localization solutions in iot era: Research challenges and future perspectives," *Computer Networks*, vol. 212, p. 109041, 2022.
- [63] M. Elsanhoury, J. Koljonen, P. Välisuo, M. Elmusrati, and H. Kuusniemi, "Survey on recent advances in integrated gnss towards seamless navigation using multi-sensor fusion technology," in *Proceedings of the 34th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2021)*, 2021, pp. 2754–2765.
- [64] F. Zangenehnejad and Y. Gao, "Gnss smartphones positioning: Advances, challenges, opportunities, and future perspectives," *Satellite navigation*, vol. 2, no. 1, p. 24, 2021.
- [65] F. W. Gobana, "Survey of inertial/magnetic sensors based pedestrian dead reckoning by multi-sensor fusion method," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2018, pp. 1327–1334.
- [66] N. Al Bitar, A. Gavrilov, and W. Khalaf, "Artificial intelligence based methods for accuracy improvement of integrated navigation systems during gnss signal outages: an analytical overview," *Gyroscopy and Navigation*, vol. 11, no. 1, pp. 41–58, 2020.
- [67] R. Nagula, K. Srivastava, and K. Surender, "An overview of multi-sensor fusion methods with partial reliance on gnss," in *2022 1st International Conference on the Paradigm Shifts in Communication, Embedded Systems, Machine Learning and Signal Processing (PCEMS)*. IEEE, 2022, pp. 98–103.
- [68] N. Cohen and I. Klein, "Inertial navigation meets deep learning: A survey of current trends and future directions," *Results in Engineering*, p. 103565, 2024.
- [69] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan *et al.*, "The prisma 2020 statement: an updated guideline for reporting systematic reviews," *bmj*, vol. 372, 2021.
- [70] S. Malkos, "Google to provide raw gnss measurements: User location takes center stage in new android os," *GPS World*, vol. 27, no. 7, p. 36, 2016.
- [71] L. Zheng, X. Zhan, and X. Zhang, "A probabilistic graphic model based gps/sins integration algorithm for low-cost sensors on smartphone," in *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, 2018, pp. 271–283.

- [72] M. Grochowski, M. Schweigler, B. Alrifae, and S. Kowalewski, "A gps-aided inertial navigation system for vehicular navigation using a smartphone," *IFAC-PapersOnLine*, vol. 51, no. 10, pp. 121–126, 2018.
- [73] M. F. El-Khalea, H. M. Hendy, A. M. Kamel, and I. I. Arafa, "Loosely coupled gnss/ins for android platforms," in *2022 13th International Conference on Electrical Engineering (ICEENG)*. IEEE, 2022, pp. 37–41.
- [74] Z. Niu, P. Nie, L. Tao, J. Sun, and B. Zhu, "Rtk with the assistance of an imu-based pedestrian navigation algorithm for smartphones," *Sensors*, vol. 19, no. 14, p. 3228, 2019.
- [75] K. Park, W. Kim, and J. Seo, "Effects of initial attitude estimation errors on loosely coupled smartphone gps/imu integration system," in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2020, pp. 800–803.
- [76] W. Yan, L. Bastos, and A. Magalhães, "Performance assessment of the android smartphone's imu in a gnss/ins coupled navigation model," *IEEE Access*, vol. 7, pp. 171 073–171 083, 2019.
- [77] S. Yang, D. Yi, S. Vana, and S. Bisnath, "Resilient smartphone positioning using native sensors and ppp augmentation," *NAVIGATION: Journal of the Institute of Navigation*, vol. 70, no. 2, 2023.
- [78] K. Han, S. Lee, Y.-J. Song, H.-B. Lee, D.-H. Park, and J.-H. Won, "Precise positioning with machine learning based kalman filter using gnss/imu measurements from android smartphone," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 3094–3102.
- [79] D. Tomaszewski, "Concept of ins/gps integration algorithm designed for mems based navigation platform," 2017.
- [80] D. Yi, S. Yang, and S. Bisnath, "Native smartphone single-and dual-frequency gnss-ppp/imu solution in real-world driving scenarios," *Remote Sensing*, vol. 14, no. 14, p. 3286, 2022.
- [81] A. Elmezayen, M. Elsheikh, U. Iqbal, and A. Noureldin, "Examining the capabilities of smartphone-based high precision positioning in challenging suburban environments," in *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, 2022, pp. 2329–2338.
- [82] M. Fortunato, G. Tagliaferro, E. Fernández-Rodríguez, and J. Critchley-Marrows, "The whole works: A gnss/imu tight coupled filter for android raw gnss measurements with local ground augmentation strategies," in *Proceedings of the ION GNSS*, 2021.

- [83] S. Paul and T. Maiti, "Accurate kinematic-parameters estimation using imu and gps sensors fusion," *IEEE Sensors Journal*, 2024.
- [84] C. Jiang, Y. Chen, Z. Liu, Q. Xia, C. Chen, and J. Hyypä, "A probabilistic method-based smartphone gnss fault detection and exclusion system utilizing pdr step length," *Remote Sensing*, vol. 15, no. 20, p. 4993, 2023.
- [85] C. Jiang, Y. Chen, C. Chen, S. Chen, Q. Meng, Y. Bo, and J. Hyypä, "Cooperative smartphone gnss/pdr for pedestrian navigation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 6, pp. 2301–2305, 2022.
- [86] C. Jiang, Y. Chen, C. Chen, and J. Hyypä, "Walking gaits aided mobile gnss for pedestrian navigation in urban areas," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 8499–8510, 2023.
- [87] M. Basso, A. Martinelli, S. Morosi, and F. Sera, "A real-time gnss/pdr navigation system for mobile devices," *Remote Sensing*, vol. 13, no. 8, p. 1567, 2021.
- [88] M. Basso, M. Galanti, G. Innocenti, and D. Miceli, "Triggered ins/gnss data fusion algorithms for enhanced pedestrian navigation system," *IEEE Sensors Journal*, vol. 20, no. 13, pp. 7447–7459, 2020.
- [89] R. Zhang, J. Mi, J. Li, and Q. Wang, "A continuous pdr and gnss fusing algorithm for smartphone positioning," *Remote Sensing*, vol. 14, no. 20, p. 5171, 2022.
- [90] A. Angrisano, M. Vultaggio, S. Gaglione, and N. Crocetto, "Pedestrian localization with pdr supplemented by gnss," in *2019 European Navigation Conference (ENC)*. IEEE, 2019, pp. 1–6.
- [91] K.-S. Kim and Y. Shin, "Deep learning-based pdr scheme that fuses smartphone sensors and gps location changes," *IEEE Access*, vol. 9, pp. 158 616–158 631, 2021.
- [92] D. Dahlke, P. Drakoulis, A. Fernández García, S. Kaiser, S. Karavarsamis, M. Mallis, W. Oliff, G. Sakellari, A. Belmonte-Hernández, F. Alvarez *et al.*, "Seamless fusion: multi-modal localization for first responders in challenging environments," *Sensors*, vol. 24, no. 9, p. 2864, 2024.
- [93] A. Rehman, Q. Liu, Z. Wu, H. Zhu, J. Qian, Y. Wang, and P. Liu, "Pdr/gnss fusion algorithm based on joint heading estimation," in *China Satellite Navigation Conference (CSNC) 2019 Proceedings: Volume I*. Springer, 2019, pp. 326–339.
- [94] F. Zhu, X. Tao, W. Liu, X. Shi, F. Wang, and X. Zhang, "Walker: Continuous and precise navigation by fusing gnss and mems in smartphone chipsets for pedestrians," *Remote Sensing*, vol. 11, no. 2, p. 139, 2019.

- [95] X. Li, D. Wei, W. Zhang, Y. Xu, and H. Yuan, "Integrated pdr/gnss at different times for pedestrian localization in urban canyon," *IEEE Access*, vol. 9, pp. 22 698–22 710, 2021.
- [96] Q. Liu, R. Ying, Z. Dai, Y. Wang, J. Qian, and P. Liu, "Multi-phase fusion for pedestrian localization using mass-market gnss and mems sensors," *Sensors*, vol. 23, no. 7, p. 3624, 2023.
- [97] L.-T. Hsu, Y. Gu, Y. Huang, and S. Kamijo, "Urban pedestrian navigation using smartphone-based dead reckoning and 3-d map-aided gnss," *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1281–1293, 2015.
- [98] A. Sheta, A. Mohsen, B. Sheta, and M. Hassan, "Improved localization for android smartphones based on integration of raw gnss measurements and imu sensors," in *2018 International conference on computer and applications (ICCA)*. IEEE, 2018, pp. 297–302.
- [99] A. Suprem, V. Deep, and T. Elarabi, "Orientation and displacement detection for smartphone device based imus," *IEEE Access*, vol. 5, pp. 987–997, 2016.
- [100] A. Hussain and L. E. Díez, "Comparison of fgo and kf for pdr-gnss fusion on smartphone for diverse pedestrian movements," in *2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2023, pp. 1–6.
- [101] A. H. Magsi, L. E. Díez, and S. Knauth, "Continuous high-precision positioning in smartphones by fgo-based fusion of gnss–ppk and pdr," *Micromachines*, vol. 15, no. 9, p. 1141, 2024.
- [102] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.
- [103] N. Kovvali, M. Banavar, and A. Spanias, *An introduction to kalman filtering with matlab examples*. Springer Nature, 2022.
- [104] Y. Kim and H. Bang, "Introduction to kalman filter and its applications," in *Introduction and implementations of the Kalman filter*. IntechOpen, 2018.
- [105] T. M. Apostol, *One-variable calculus, with an introduction to linear algebra*. Blaisdell Waltham, Toronto, London, 1967, vol. 1.
- [106] P. S. Maybeck, *Stochastic models, estimation, and control*. Academic press, 1982, vol. 3.
- [107] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer graphics and applications*, vol. 25, no. 6, pp. 38–46, 2005.
- [108] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.

- [109] B. M. Bell and F. W. Cathey, “The iterated kalman filter update as a gauss-newton method,” *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.
- [110] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
- [111] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [112] A. Doucet, N. De Freitas, N. J. Gordon *et al.*, *Sequential Monte Carlo methods in practice*. Springer, 2001, vol. 1, no. 2.
- [113] S. Thrun, *Probabilistic robotics*. ACM New York, NY, USA, 2002, vol. 45, no. 3.
- [114] A. H. Gandomi and A. H. Alavi, “Krill herd: a new bio-inspired optimization algorithm,” *Communications in nonlinear science and numerical simulation*, vol. 17, no. 12, pp. 4831–4845, 2012.
- [115] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [116] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (slam) problem,” *IEEE Transactions on robotics and automation*, vol. 17, no. 3, pp. 229–241, 2001.
- [117] F. Dellaert, “Factor graphs and gtsam: A hands-on introduction,” Tech. Rep. 4, 2012.
- [118] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, “Factor graph based incremental smoothing in inertial navigation systems,” in *2012 15th International Conference on Information Fusion*. IEEE, 2012, pp. 2154–2161.
- [119] N. Sünderhauf, M. Obst, G. Wanielik, and P. Protzel, “Multipath mitigation in gnss-based localization using robust optimization,” in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 784–789.
- [120] T. Qin, S. Cao, J. Pan, and S. Shen, “A general optimization-based framework for global pose estimation with multiple sensors,” *arXiv preprint arXiv:1901.03642*, 2019.
- [121] Z. Huo, L. Jin, H. Wang, X. Sun, and Y. He, “A robust factor graph optimization method of gnss/ins/odo integrated navigation system for autonomous vehicle,” *Measurement Science and Technology*, vol. 36, no. 1, p. 016301, 2024.
- [122] K. Wang, L. Wu, Y. Ben, Q. Li, C. Lv, and L. Hou, “Robust factor graph optimization for gnss/ins tightly coupled integration: A flexible strategy for urban navigation resilience,” *IEEE Sensors Journal*, 2025.

- [123] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual–inertial odometry,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.
- [124] T. Suzuki, “First place award winner of the smartphone decimeter challenge: Global optimization of position and velocity by factor graph optimization,” in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 2974–2985.
- [125] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.
- [126] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “isam2: Incremental smoothing and mapping using the bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
- [127] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, “It is time for factor graph optimization for gnss/ins integration: Comparison between fgo and ekf,” *arXiv preprint arXiv:2004.10572*, 2020.
- [128] S. Agarwal, K. Mierle *et al.*, “Ceres solver: Tutorial & reference,” p. 8, 2012.
- [129] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g 2 o: A general framework for graph optimization,” in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3607–3613.
- [130] C. Taylor and J. Gross, “Factor graphs for navigation applications: A tutorial,” *NAVIGATION: Journal of the Institute of Navigation*, vol. 71, no. 3, 2024.
- [131] B. Song, R. Xu, and L.-T. Hsu, “Fgo mythbusters: Explaining how kalman filter variants achieve the same performance as fgo in navigation applications,” *arXiv preprint arXiv:2511.00306*, 2025.
- [132] W. Wen, Y. C. Kan, and L.-T. Hsu, “Performance comparison of gnss/ins integrations based on ekf and factor graph optimization,” in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, pp. 3019–3032.
- [133] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, “Deep learning sensor fusion for autonomous vehicle perception and localization: A review,” *Sensors*, vol. 20, no. 15, p. 4220, 2020.
- [134] Z. Huang, G. Ye, P. Yang, and W. Yu, “Application of multi-sensor fusion localization algorithm based on recurrent neural networks,” *Scientific reports*, vol. 15, no. 1, p. 8195, 2025.
- [135] A. Abdulkamsoud and R. Ahmed, “Transformer-based sensor fusion for autonomous vehicles: A comprehensive review,” *IEEE Access*, 2025.
- [136] P. Aggarwal, Z. Syed, X. Niu, and N. El-Sheimy, *MEMS-Based Integrated Navigation*. Artech House, 2010.

- [137] D. Hwang, D. Lim, S. Cho, and S. Lee, "Unified approach to ultra-tightly-coupled gps/ins integrated navigation system," *IEEE Aerospace and Electronic Systems Magazine*, vol. 26, no. 3, pp. 30–38, 2011.
- [138] J. Kouba and P. Héroux, "Precise point positioning using igs orbit and clock products," *GPS solutions*, vol. 5, no. 2, pp. 12–28, 2001.
- [139] J. F. Zumberge, M. B. Heflin, D. C. Jefferson, M. M. Watkins, and F. H. Webb, "Precise point positioning for the efficient and robust analysis of gps data from large networks," *Journal of geophysical research: solid earth*, vol. 102, no. B3, pp. 5005–5017, 1997.
- [140] J. Kouba, "A guide to using international gnss service (igs) products," 2009.
- [141] C. Klatt, S. Banville, and E. Hassen, "The canadian spatial reference system precise point positioning service: Today and tomorrow." 2019.
- [142] A. Angrisano *et al.*, "Gnss/ins integration methods," Ph.D. dissertation, 2010.
- [143] R. V. Vitali, R. S. McGinnis, and N. C. Perkins, "Robust error-state kalman filter for estimating imu orientation," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3561–3569, 2020.
- [144] I. Skog and P. Handel, "Effects of time synchronization errors in gnss-aided ins," in *2008 IEEE /ION Position, Location and Navigation Symposium*. IEEE, 2008, pp. 82–88.
- [145] A. H. Magsi and L. E. Díez Blanco, "Performance analysis of fgo windowing strategies for pdr+gnss fusion architecture," *IEEE Access*, 2025.