










Article

OBU for Accurate Navigation through Sensor Fusion in the Framework of the EMERGE Project

Angel Luis Zuriarrain Sosa ^{1,2,*}, Valeria Ioannucci ^{1,3,*}, Marco Pratesi ^{1,4,*}, Roberto Alesii ^{1,4,5}, Carlo Albanese ², Francesco Valentini ^{1,3,4}, Elena Cinque ^{1,3,4}, Alessio Martinelli ² and Michele Brizzi ⁶

- ¹ Department of Information Engineering, Computer Science and Mathematics (DISIM), University of L'Aquila, Via Vetoio, 67100 L'Aquila, Italy; roberto.alesii@univaq.it (R.A.); francesco.valentini1@univaq.it (F.V.); elena.cinque@radiolabs.it (E.C.)
 - ² Telespazio Spa, Via Tiburtina, 965, 00156 Rome, Italy; carlo.albanese@telespazio.com (C.A.); alessio.martinelli@telespazio.com (A.M.)
 - ³ RadioLabs, Corso d'Italia, 19, 00198 Rome, Italy
 - ⁴ Centre of EXcellence on Connected, Geo-Localized and Cyber-Secure Vehicles (Ex-EMERGE), University of L'Aquila, Via Vetoio, 67100 L'Aquila, Italy
 - ⁵ Design Methodologies of Embedded Controllers, Wireless Interconnect and Systems-on-Chip (DEWS), University of L'Aquila, Via Vetoio, 67100 L'Aquila, Italy
 - ⁶ The Department of Industrial, Electronic and Mechanical Engineering (DIIEM), Roma Tre University, Via Vito Volterra, 62, 00146 Rome, Italy; michele.brizzi@uniroma3.it
- * Correspondence: angelluis.zuriarrainsosa@graduate.univaq.it (A.L.Z.S.); valeria.ioannucci@radiolabs.it (V.I.); marco.pratesi@univaq.it (M.P.)

Abstract: With the development of advanced driver assistance systems (ADAS) and autonomous vehicles (AV), recent years have seen an increasing evolution of onboard sensors and communication interfaces capable of interacting with available infrastructures, including satellite constellations, road structures, modern and heterogeneous network systems (e.g., 5G and beyond) and even adjacent vehicles. Consequently, it is essential to develop architectures that cover data fusion (multi-sensor approach), communication, power management, and system monitoring to ensure accurate and reliable perception in several navigation scenarios. Motivated by the EMERGE project, this paper describes the definition and implementation of an On Board Unit (OBU) dedicated to the navigation process. The OBU is equipped with the Xsens MTi-630 AHRS inertial sensor, a multi-constellation/multi-frequency Global Navigation Satellite System (GNSS) receiver with the u-blox ZED-F9P module and communication interfaces that afford access to the PointPerfect augmentation service. Experimental results show that GNSS, with corrections provided by augmentation, affords centimetre accuracy, with a Time To First Fix (TTFF) of about 30 s. During the on-road tests, we also collect: the output of fusion with inertial sensor data, monitoring information that assess correct operation of the module, and the OBU power consumption, that remains under 5 W even in high-power operating mode.

Keywords: on board unit; navigation system architecture; sensor fusion; loosely coupled algorithm; EMERGE project; IMU; GNSS; PointPerfect Correction Service



Citation: Zuriarrain Sosa, A.L.; Ioannucci, V.; Pratesi, M.; Alesii, R.; Albanese, C.; Valentini, F.; Cinque, E.; Martinelli, A.; Brizzi, M. OBU for Accurate Navigation through Sensor Fusion in the Framework of the EMERGE Project. *Appl. Sci.* **2024**, *14*, 4401. <https://doi.org/10.3390/app14114401>

Academic Editor: Koji Murai

Received: 29 February 2024

Revised: 11 May 2024

Accepted: 17 May 2024

Published: 22 May 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

Urban projections until 2050 [1] show an increased concentration of population in urban areas. This scenario presents challenges in terms of mobility and traffic that aim to efficiently and sustainably manage resources to increase the quality of life. There are many approaches behind the concept of smart mobility [2–4]:

- the use of technologies that provide helpful information (IoT, sensors, network);
- the creation of structures that include smart roads and monitoring platforms;
- the reduction of the ecological footprint (energy saving, bike lanes, etc.).

However, most solutions rely on efficient and accurate navigation processes.

The development of advanced driver assistance systems (ADAS) for connected and automated mobility makes the accuracy [5] of the navigation process a key issue. Sensors involved in the process of perception and comprehension of the environment, as well as fusion algorithms and decision-makers, have played a significant and evolving role in recent years. Nowadays, many resources provide helpful information to determine position, orientation, time or velocity. Nevertheless, some scenarios and phenomena directly affect the performance of sensor technologies. Thus, it is essential to develop architectures that use a multi-sensor and redundant approach to mitigate the adverse effects caused by the temporary under-performance of one or several sensors.

Recently, proposed architectural solutions have concentrated on sensor fusion [6,7] rather than implementing the entire On Board Unit (OBU) system [8–12]. This work addresses the fusion process as the core element of the OBU, dedicating a complete architectural layer. The elements used as sensors in our implementation are the inertial measurement unit (IMU) and the global navigation satellite system (GNSS) receiver, complemented by a GNSS augmentation system.

The primary obstacle in enabling AV navigation lies in establishing a consistently accurate and dependable navigation solution across all landscapes. GNSS stands out as the most prevalent source of navigation solutions due to its enduring stability in offering long-term solutions [13].

However, GNSS encounters limitations in providing accurate or any navigation solution in certain circumstances, notably within urban settings that include tunnels, underground parking areas, and high-rise buildings [14–16]. Satellite navigation in the urban environment is subject to some objectively critical issues, mainly related to the following items.

- Satellite visibility: due to the presence of urban canyons, the number of visible satellites, crucial for obtaining an accurate position, is limited.
- Multipath and non-line-of-sight reception: the typicality and singularity of the urban context, in which the vehicles carry out their service, produces a non-trivial multipath error that seriously compromises the performance of the GNSS signal.
- Other kinds of interference that can have a significant impact in an urban context.

These criticalities result in considerable variability in the reception of the GNSS signal, ranging from optimal conditions to complete absence. On the other hand, inertial navigation systems (INS) possess several advantages. They operate continuously, exhibit low short-term noise, and resist jamming and interference. However, INS accuracy degrades over time due to integration errors, and maintaining effective sole-means navigation is costly. While GNSS ensures high long-term accuracy, its output rate is lower, and signal obstruction leads to discontinuous navigation. By integrating INS and GNSS, their advantages complement each other. GNSS measurements prevent inertial solution drift, while INS smoothes GNSS output and bridges signal outages. This integration results in a continuous, high-bandwidth navigation solution with long and short-term accuracy [17].

The navigation system emerges as a pivotal enabling factor within the framework of the EMERGE project [18], co-funded by the European Union and spearheaded by RadioLabs in collaboration with esteemed partners such as the University of L'Aquila, Telespazio, Leonardo, and Elital. The project aims to design and develop a novel OBU to provide dual use capabilities to commercial vehicles (i.e., last mile delivery and emergency operations) enabled by dedicated cloud-edge services. The OBU is equipped with vehicle-to-everything (V2X) connectivity [19], procedures for cyber-secure operations [20] and an accurate geolocalization platform. The role of this latter is to guarantee real-time, precise positioning of all nodes and their relative positions, accomplished through innovative fusion techniques that leverage data from multi-constellation GNSS sensors and inertial sensors.

When elements with different characteristics regarding data rate, coding, data type, and time references are integrated, an infrastructure capable of handling all the information flows must be designed. For this reason, this work continues in Section 2, describing an architectural proposal applied in the OBU realisation process. Also, a multi-sensor approach is employed, consequently using data fusion techniques to achieve accurate and

robust navigation in different scenarios. Section 3 details the integration algorithm based on an Extended Kalman Filter. The experimental setup is divided into two parts: one dedicated to evaluating the individual sensors' performance and the other to evaluating the whole system's performance based on the integration algorithm. Section 4 describes the experimentation procedure, and the main results obtained are presented in Section 5. Finally, Section 6 concludes this article.

2. EMERGE Onboard System Architecture

This section describes the proposed architecture for achieving the EMERGE On board Unit (OBU). The objective is to present and use a generic architecture: easy to understand and implement, with a scalable structure compatible with different hardware platforms and operating systems. In this sense, the challenge is to define a harmonious level of abstraction that guarantees the following functionalities.

- Compatibility: taking advantage of existing components, both: HW and SW (COTS).
- Efficiency: exploiting the advantage of the HW architecture used.
- Performance: establishment of specific metrics in each context.
- Portability: to guarantee compatibility with Operating Systems and HW.
- Operation: real-time or post-processing execution capability.
- Scalability: add/delete components: Sensors, SW, algorithms.

We highlight the proposal of five specified layers based on their functionality and interaction with the navigation system. Including a monitoring layer allows the evaluation of prototypes and different scenarios, the detection of anomalies and problems in the singular components of the system and the evaluation of performances.

Figure 1 shows the diagram of the proposed architecture. Each of the layers to which each element belongs is identified by a different colour. The connections, paths and formats/encoding used by each element of the OBU are specified. Below, some details of the functional blocks are presented.

- Sensor (layer 1): includes all sensors capable of providing helpful information in the navigation process, i.e., GNSS receivers, inertial sensors, RTK stations, augmentation services, onboard sensors, and external clocks.
- Parser; Conditioner (layer 2): handles decoding/parsing and filtering information from the sensor level. When communication with sensors (configuration signals, request) is needed, this block also implements the appropriate coding/decoding process.
- Controller (layer 3): controls the start-up, interruptions, restart, and the change of parameters process (in the case of a dynamic scenario). In addition, it manages the possible changes in information flows that may occur between the Core layer and the Sensor layer.
- Core (layer 4): The system's centre contains all the navigation and information management algorithms. In this application, the Sensor Fusion algorithm and the Integrity SW components are located in this layer.
- Monitoring (layer 5): This layer monitors the system's performance in the navigation process and HW resources. It also introduces remote accessibility functionalities and possibly warning signals and notifications.

The proposed blocks or components in each layer are intended to provide the required functionalities of the OBU. The first element to guarantee is connectivity ("*always connected*"). The OBU COMM and CRYPTO ENGINE blocks are external subsystems ensuring remote communication and IP-based services access. The system uses the connection to receive information from the Augmentation GNSS (AGNSS) systems and exchange information with traffic control elements: control centres, other vehicles, or early alert systems. The system proposes solutions based on loosely coupled integration between the inertial sensor measurements and the previously corrected GNSS navigation solutions. The modules involved in the navigation process are dedicated to decoding the information coming from the sensors, managing the information flows and synchronisation. The main objective is

to integrate or fuse all useful navigation information to achieve higher levels of accuracy and integrity.

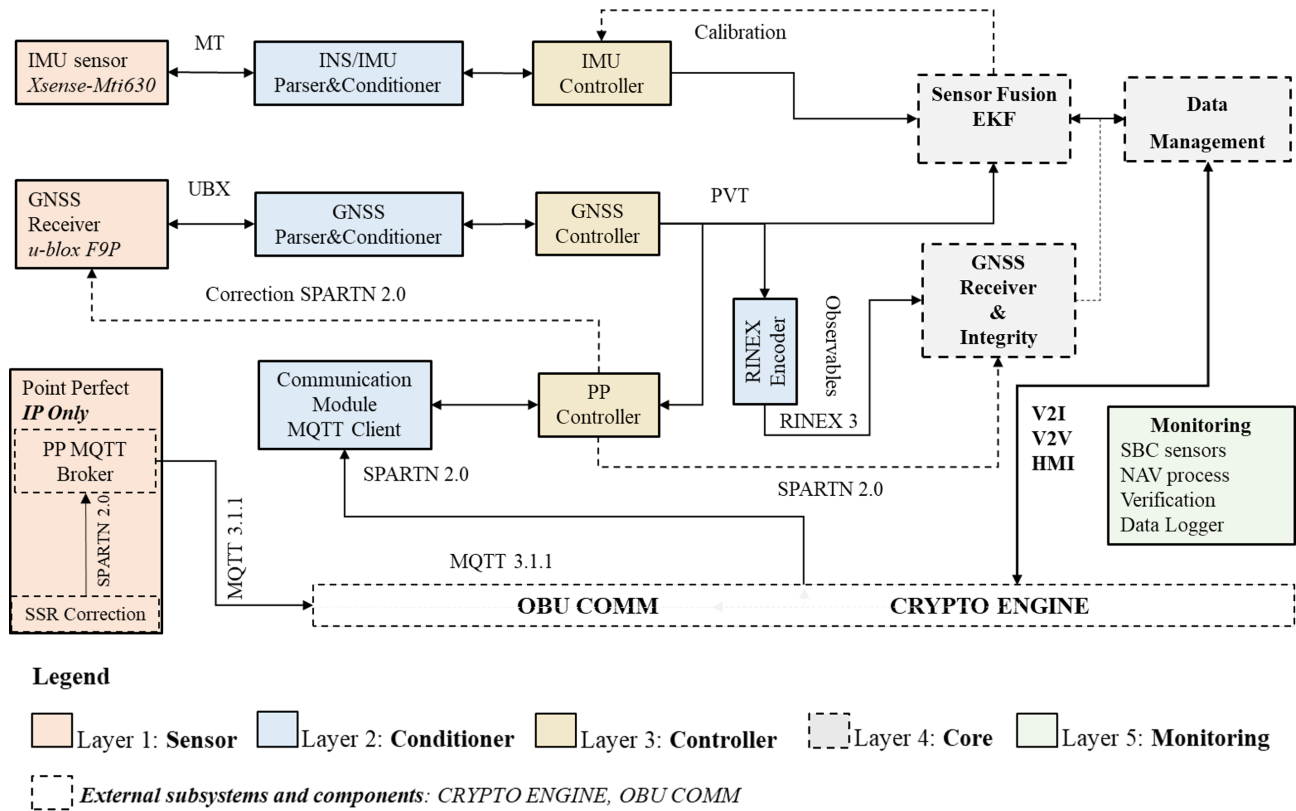


Figure 1. EMERGE Onboard System Architecture.

2.1. EMERGE Navigation System Implementation

The HW, SW or service components that drive up the onboard navigation system are selected according to the functional requirements of the EMERGE project. For each use case, requirements are mapped to performance measures that sensors must satisfy in the first layer. Specifically, two main elements are considered: the GNSS receiver and the inertial sensor. A GNSS augmentation service is used to improve the satellite positioning performance through corrections received via the OBU connection to the Internet.

2.1.1. Sensors and Services

The GNSS receiver uses the u-blox ZED-F9P positioning module [21] that provides multi-band GNSS for high-volume industrial applications. The ZED-F9P integrates multi-band RTK and PPP-RTK (Precise Point Positioning—Real Time Kinematic) for centimetre accuracy. The module is ideal for automotive and UAV applications due to its low power consumption, accuracy and integration capability. Another strength of this receiver is the ability to handle multiple channels dedicated to different constellations and frequency bands. In addition, the ZED-F9P module features native compatibility with the PointPerfect (PP) augmentation service [22].

PointPerfect is a GNSS augmentation service that supports PPP-RTK for position estimation, with 99.9% availability using the Internet or satellite (L-band) connection. The PP service has reported high availability and high levels of accuracy, making this product optimal for critical applications such as automotive. Specifically, in our configuration, we receive corrections over an IP connection. The GNSS corrections are received in SPARTN 2.0 format transparently and cost-effectively via the MQTT broker offered by the u-blox Thingstream platform [23]. This method of receiving GNSS corrections ensures efficient use of bandwidth, improves GNSS receiver initialisation times and reduces the power

consumption of the navigation system. It is important to note that the specific PointPerfect IP service (Unlimited access to IP data stream) costs 19.00 USD per month at the moment of this research.

The inertial sensor component used is the Xsens-MTi-630 AHRS (Attitude and Heading Reference System) [24], which contains a 3-axis gyroscope, a 3-axis accelerometer, a 3-axis barometer magnetometer, a high-precision oscillator and a low-power microcontroller unit (MCU). The MCU applies calibration models (unique to each sensor and including orientation, gain and polarisation offsets, plus more advanced relationships such as non-linear temperature effects and other higher order terms) and runs Xsens' optimised strap-down algorithm, which performs high-speed dead-reckoning calculations up to 2 kHz, enabling accurate capture of high-frequency motions and compensation for coning and sculling. The MTi-600's output data is easily configured and customised according to the application's requirements and can be set to use one of several inertial navigation filter profiles [25]. In particular, the MTi-630 AHRS device enhances the process of determining referenced magnetic north and 3D acceleration, velocity and heading calibration data. The project requirements motivating the use of this sensor are low in-run bias stability (accelerometer: 10 (x,y) and 15 (z) [micro-g]; gyroscope: 8 [deg/h]), low noise density (accelerometers: 60 [micro-g/sqrt(Hz)]) and the ability to output data at a sufficiently high frequency for the sensor fusion process. Successive models of the MTi-600 family (i.e., MTi-670/80) support the INS/GNSS integration process internally; in our case, they do not; therefore, the INS/GNSS synchronisation and integration process is performed by SW.

2.1.2. Sensor Fusion and Integrity

The navigation system's core comprises the components in charge of PNT (Positioning, Navigation and Timing) and Integrity estimation. The multi-sensor approach uses an algorithm that integrates all the elements that provide helpful information in the navigation process. Section 3 describes the loosely coupled algorithm that integrates inertial and GNSS data.

2.1.3. Signal and Information Exchange

The *Always Connected* paradigm opens up many possibilities and functionalities for the satellite navigation system. Therefore, a communication system between internal and external elements is essential. We use MQTT as a communication protocol to guarantee agile and lightweight internal communication between the OBU components (sensors, SW modules, monitoring services). MQTT is a standards-based communication protocol [26], or set of rules, used to communicate from one device/sensor/component to another. Intelligent sensors, wearable devices, and other IoT elements typically transmit and receive data that require modest bandwidth. Therefore, MQTT is used for data exchange as it is easy to implement and can communicate data efficiently.

Figure 2 presents a detailed view of exchanging messages via the MQTT broker (Identified as "SBC BROKER"). The arrowheads on the individual connections show whether it is only a subscriber, publishes and subscribes, or only publishes messages through the Broker. For instance, given its observation functionality, the Monitoring block only intends to receive messages and not publish them. Sensors and services publish correctly decoded measured/acquired data in a dedicated Topic, from where they are accessible to the processes or modules that process them. The *Controller4Controllers* module is responsible for configuring, calibrating and synchronising the sensors and coding/decoding SW components. The Broker is also involved in the external communication process via the *OBU COMM* module. After the appropriate encryption process, the information to be transmitted to the control centre is made available in a dedicated Topic managed by the Data Management element.

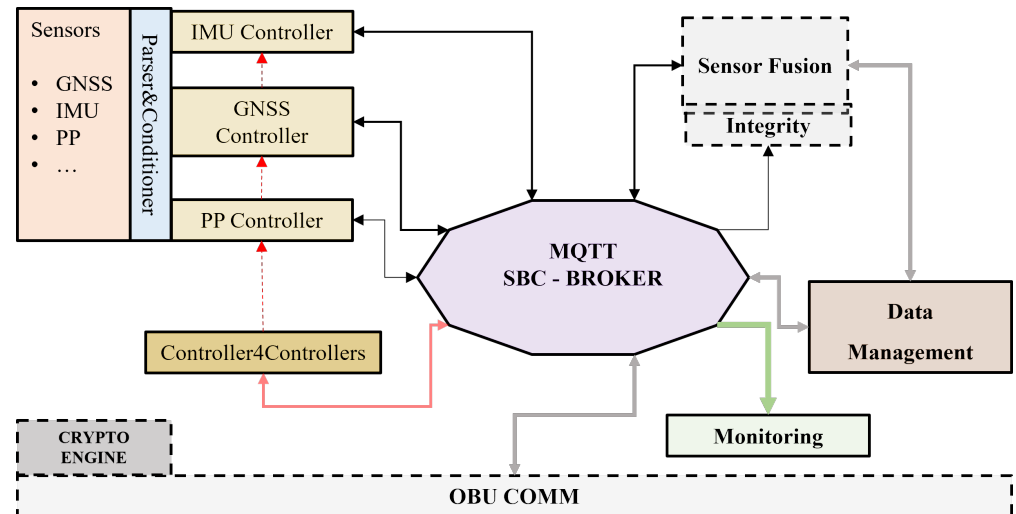


Figure 2. MQTT Broker Backend.

2.1.4. Power Management System

The Power Management System is the element that ensures that the system stays on and continues to operate regardless of the state of the vehicle. It consists of two elements: a HW and a SW element. The HW element is the Sixfab Power Management; UPS HAT [27], while the SW element is implemented using a dedicated API: Sixfab Power Python API [28]. In short, it is an uninterruptible power supply with a built-in 18650-Li-Ion battery, allowing it to automatically switch from one power supply to another without causing a reboot or system failure. The device can enter deep sleep mode in power-sensitive applications and save battery energy.

2.1.5. Monitoring

The Monitoring component consists of a Frontend, a graphical interface that displays the data of interest, and a Backend responsible for manipulating and storing internal and external information flows. This project's Monitoring layer is based on the Node-RED [29] tool or framework. Node-RED is a flow-based development tool developed to connect devices, APIs and online services as part of the IoT. In addition, it provides a web browser-based flow editor, which can be used to create JavaScript functions. Application elements can be saved or shared for reuse. The runtime uses Node.js, and the streams created are stored using JSON. The decision to use this technology is based on simplicity and support for the MQTT protocol and sensor information. In addition, the built-in WEB service allows remote monitoring of the system.

2.1.6. Hardware Description

The HW used as the On-Board Computer (OBC), shown in Figure 3, is the Raspberry Pi 4 Model B [30] equipped with BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8 GHz, 8 GB SDRAM memory and 32 GB Micro-SD card. The sensors (Inertial and GNSS) are connected via USB interfaces, while the Power Management and UPS system is connected to the standard 40-pin GPIO. The SIRIUS RTK GNSS ROVER multi-constellation receiver [31] (based on u-blox ZED-F9P) is connected to the Septentrio Polant-x MF antenna via the SMA interface provided. Both the antenna and the GNSS receiver verify multi-frequency and multi-constellation compatibility. Finally, the components are assembled inside a Sixfab enclosure specifically dedicated to Raspberry projects in outdoor environments with IP65 compliance. An IP65-rated enclosure gives protection against moisture, rain, snow, wind, dust and low-pressure water jets from any direction, as well as condensation and water spray. It is suitable for most outdoor enclosures that will not encounter extreme weather. Figure 4 shows the final Navigation Box, the interfaces and the GNSS antenna.

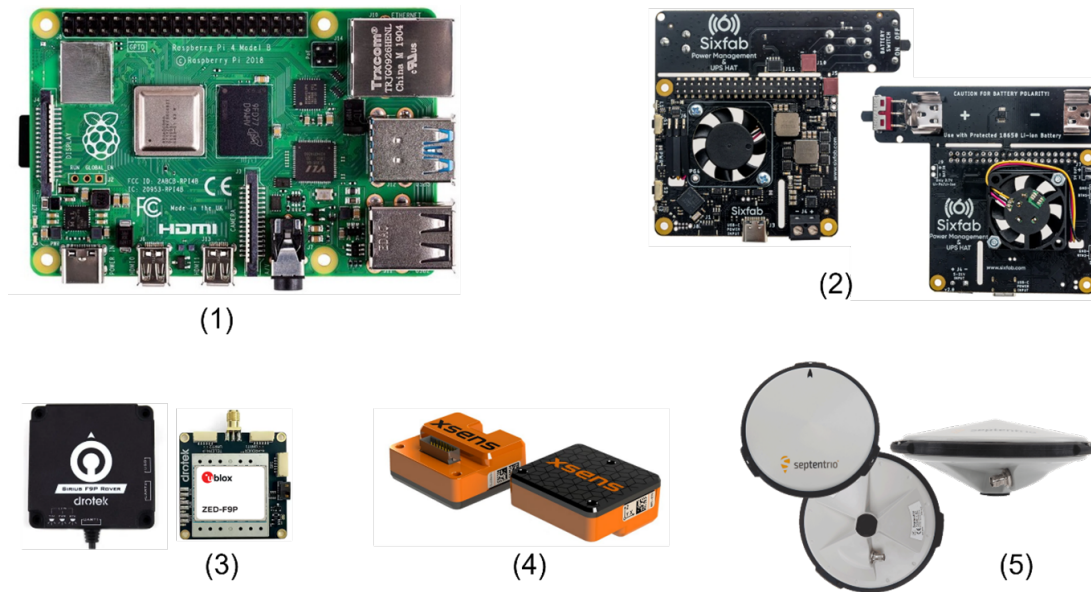


Figure 3. OBU HW components: (1) Onboard Computer Raspberry Pi 4 Model B; (2) Sixfab Power Management; UPS HAT power backup system; (3) Drotek (SIRIUS RTK GNSS ROVER: F9P) with u-blox ZED-F9P chip; (4) Xsens MTi630 AHRS Inertial Sensor; (5) Septentrio Polant-x MF antenna.



Figure 4. Final assembled system: Navigation EMERGE OBU.

3. Sensor Fusion: Loosely Coupled Algorithm Implementation

The primary advantage of a loosely coupled integration architecture lies in its simplicity. This architecture is versatile and compatible with any INS and GNSS user equipment, making it ideal for retrofit applications. In a loosely coupled INS/GNSS system, the integration algorithm utilizes GNSS position and velocity solutions as measurement inputs, regardless of the specific type of INS correction or GNSS aiding employed.

In the cascaded operation of this architecture, the GNSS user equipment, which incorporates a navigation filter [17], utilizes the Extended Kalman Filter (EKF) as the integration solution. The EKF is a recursive filter specifically designed to estimate the state of a dynamic system from multiple noisy measurements. It has evolved from the Standard Kalman Filter to address the complexities associated with nonlinear dynamic systems. The resulting integrated navigation solution consists of the INS navigation solution refined by the Kalman filter’s error estimates. Specifically, within the loosely coupled integration architecture, the sequential processes can be delineated as follows:

1. calculation of position and velocity with GNSS;

2. calculation of the difference between the estimated position and velocity from the GNSS and INS solutions to assess IMU errors by integrating the estimated differences in a Kalman filter;
3. correction of the INS solution using variational equations.

The tests were carried out in a confined area. To simplify the spatial representation in this local context, the North East Down (NED) coordinate frame was chosen. Consequently, the position of the OBU is represented using geodetic coordinates: latitude L_b , longitude λ_b , and ellipsoidal altitude h_b . The Earth-referenced velocity is resolved in the local navigation axis to give \mathbf{v}_{eb}^n , and the attitude is expressed as the body-to-local navigation frame coordinate transformation matrix \mathbf{C}_b^n . Here, e denotes the Earth-Centered Earth-Fixed (ECEF) reference frame, n is the NED frame, and b represents the inertial platform “body” coordinate frame.

Figure 5 illustrates the functional diagram of the GNSS-INS loose coupling process. It is a closed-loop correction architecture; consequently, the estimated errors in the NED reference frame, $\delta\hat{L}_b$ for the latitude, $\delta\hat{\lambda}_b$ for the longitude, $\delta\hat{h}_b$ for the height, $\delta\hat{\mathbf{v}}_{eb}^n$ for the velocity and $\delta\hat{\mathbf{C}}_b^n$ for the attitude are fed back to the inertial navigation processor, where they are used to correct the inertial navigation solution. In this way, the integrated navigation solution of the navigation system is the inertial navigation solution itself, and is obtained, respectively, for the orientation, $\hat{\mathbf{C}}_b^n$, the velocity, $\hat{\mathbf{v}}_{eb}^n$, and the position, as follows [17]:

$$\hat{\mathbf{C}}_b^{n+} = (\mathbf{I} - [\delta\hat{\boldsymbol{\psi}}_{nb}^n]_x) \hat{\mathbf{C}}_b^{n-} \tag{1}$$

$$\hat{\mathbf{v}}_{eb}^{n+} = \hat{\mathbf{v}}_{eb}^{n-} - \delta\hat{\mathbf{v}}_{eb}^n \tag{2}$$

$$\hat{L}_b^+ = \hat{L}_b^- - \delta\hat{L}_b \tag{3}$$

$$\hat{\lambda}_b^+ = \hat{\lambda}_b^- - \delta\hat{\lambda}_b \tag{4}$$

$$\hat{h}_b^+ = \hat{h}_b^- - \delta\hat{h}_b \tag{5}$$

where superscripts $-$ and $+$ are used to indicate the solution before and after correction, $\delta\hat{\boldsymbol{\psi}}_{nb}^n$ is the vector of Euler angles of the correction (i.e., yaw, pitch, and roll), and $[\delta\hat{\boldsymbol{\psi}}_{nb}^n]_x$ represents the antisymmetric matrix

$$[\delta\hat{\boldsymbol{\psi}}_{nb}^n]_x = \begin{bmatrix} 0 & -\delta\hat{\psi}_{nb,z}^n & \delta\hat{\psi}_{nb,y}^n \\ \delta\hat{\psi}_{nb,z}^n & 0 & -\delta\hat{\psi}_{nb,x}^n \\ -\delta\hat{\psi}_{nb,y}^n & \delta\hat{\psi}_{nb,x}^n & 0 \end{bmatrix}.$$

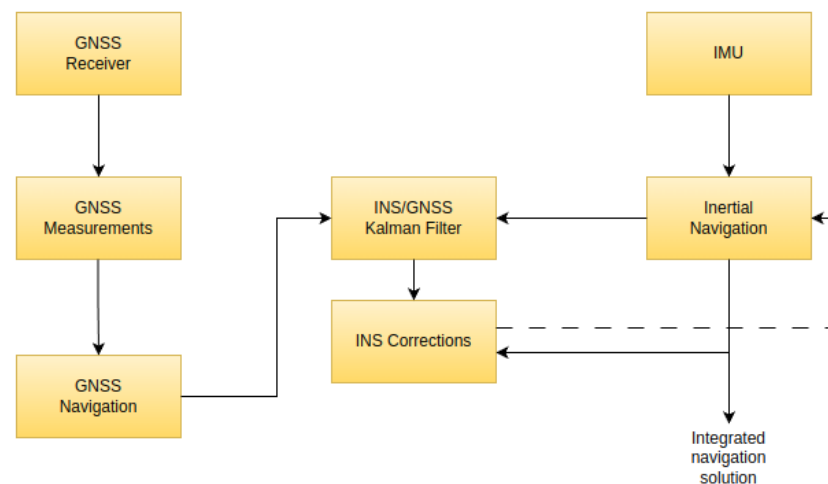
The pseudocode outlined in Algorithm 1 presents the procedural steps of the implemented INS-GNSS loosely coupled algorithm. Initially, it sets crucial parameters to account for IMU errors. The algorithm then begins with an initialisation loop which is used to estimate the initial pose of the vehicle and to configure Kalman filter parameters. After initialisation, it uses IMU data to update the vehicle’s pose, incorporating corrections from the Kalman filter when new GNSS measurements are available and providing the integrated navigation solution as output. A Zero Velocity Update (ZUPT) is also employed to mitigate the position drift introduced when the vehicle is stationary. All these steps are further described in the following.

Algorithm 1 Loosely coupled algorithm

```

1: Set initialisation parameters for IMU errors;
2:  $frequency_{IMU} = 100$ ;
3:  $init\_time = 15$ ;
4:  $init\_imu\_samples = init\_time * frequency_{IMU}$ ;
5:  $level\_time = 10$ ;
6:  $init\_imu\_samples\_level = level\_time * frequency_{IMU}$ ;
7: for  $i = 1$  to  $init\_imu\_samples$  do
8:   Initialisation loop
9: end for
10: for  $i = init\_imu\_samples + 1$  to end do
11:   Specific force  $f_{ib}^b \leftarrow fb_{IMU}(i)$ ;
12:   Angular velocity  $\omega_{ib}^b \leftarrow wb_{IMU}(i)$ ;
13:   Correct  $f_{ib}^b$  and  $\omega_{ib}^b$  using estimated biases;
14:   Update navigation solution with mechanisation equations;
15:   Apply ZUPT detection algorithm;
16:   if There are new  $r, v$  solution from GNSS then
17:     Apply Kalman Filter;
18:     Update the navigation solution with Kalman filter estimates;
19:   end if
20: end for

```

**Figure 5.** Loosely coupled INS/GNSS coupling scheme.**3.1. Setting Initialisation Parameters for IMU Errors**

Inertial navigation systems are renowned for delivering highly accurate position, velocity, and attitude information, particularly over short time spans. However, this precision degrades significantly over extended periods due to inherent error sources within the sensors. To address this challenge, the algorithm's first step focuses on precisely defining the initialisation parameters related to IMU errors. This meticulous calibration lays the groundwork for mitigating sensor inaccuracies and ensures the subsequent integration of INS-GNSS data yields precise and reliable navigation estimates.

The Allan variance method is used to characterise various error types present in inertial sensor data. This method allows for representing root mean square (RMS) random drift error as a function of averaging time [32]. Inertial measurements were analysed using the Allan variance method; precisely, specific force and angular velocity measurements collected from the IMU during a stationary five-hour test were utilized. These measurements were crucial in deriving parameters such as velocity random walk, angle random walk, angle rate random walk, bias instability, correlation times, and dynamic bias root-PSD for the accelerometer and gyroscope.

3.2. The Initialisation Loop

As a systematic iteration over IMU samples, this Loop (2) calculates essential parameters and initializes the navigation solution when the system is stationary. Its execution sets the stage for optimal accuracy and performance of the Kalman filter during subsequent integration, establishing a robust foundation for a precise and reliable navigation solution. Within each iteration over the IMU measurement taken in the first seconds, the initialisation loop calculates the IMU sampling interval (tor_i) considering the IMU frequency, facilitating precise temporal alignment. Specific force (f_{ib}^b) and angular velocity (ω_{ib}^b) measurements are extracted from the IMU, providing essential motion-related data. The Algorithm 2 checks for the availability of new GNSS measurements within the current Inertial Navigation System (INS) time. If available, it updates the GNSS position. To account for synchronisation issues between the IMU and GNSS data, this check is performed within a time window of duration $2\varepsilon = 5$ ns. IMU measurements are accumulated for levelling purposes and also GNSS data, when available, is accumulated to initialize position parameters. Roll and pitch are computed based on the averaged specific force measurements through the levelling process. The estimated position is initialized using the averaged GNSS positions.

Algorithm 2 Initialisation Loop

```

1: gnss_eps =  $\varepsilon$ ; %duration in seconds
2: for  $i = 1$  to init_imu_samples do
3:   if  $i == 1$  then
4:      $tor_i \leftarrow \frac{1}{frequency_{IMU}}$ ;
5:   else
6:      $tor_i \leftarrow t_{IMU}(i) - t_{IMU}(i - 1)$ ;
7:   end if
8:    $f_{ib}^b \leftarrow fb_{IMU}(i)$ ;
9:    $\omega_{ib}^b \leftarrow wb_{IMU}(i)$ ;
10:   $is\_gnss\_available \leftarrow false$ ;
11:   $gdx \leftarrow find(t_{GNSS} \geq (t_{IMU}(i) - gnss\_eps) \text{ and } t_{GNSS} < (t_{IMU}(i) + gnss\_eps))$ ;
12:
13:  %Check if there is a new GNSS measurement to process at current INS time
14:  if (!isempty( $gdx$ ) and  $gdx > 1$ ) then
15:     $is\_gnss\_available \leftarrow true$ ;
16:     $last\_gdx \leftarrow gdx$ ;
17:    gnss position  $GNSS\_r \leftarrow [gnss\_lat(gdx); gnss\_lon(gdx); gnss\_h(gdx)]$ ;
18:  end if
19:   $f\_leveling_{ib}^b \leftarrow [f\_leveling_{ib}^b, f_{ib}^b]$ ;
20:   $\omega\_leveling_{ib}^b \leftarrow [\omega\_leveling_{ib}^b, \omega_{ib}^b]$ ;
21:  if  $is\_gnss\_available$  then
22:     $GNSS\_r\_init = [GNSS\_r\_init, GNSS\_r]$ ;
23:  end if
24:
25:  if ( $i > init\_imu\_samples\_level$  and !isempty( $GNSS\_r\_init$ )) then
26:     $ave\_f_{ib}^b \leftarrow mean(f\_leveling_{ib}^b, 2)$ ;
27:     $roll \leftarrow atan2(-ave\_f_{ib}^b(2), -ave\_f_{ib}^b(3))$ ;
28:     $pitch \leftarrow atan(ave\_f_{ib}^b(1) / \sqrt{ave\_f_{ib}^b(2)^2 + ave\_f_{ib}^b(3)^2})$ ;
29:     $est\_r_{eb}^n \leftarrow mean(GNSS\_r\_init, 2)$ ; %Initializes the estimated position
30:  end if
31: end for

```

3.3. Specific Force and Angular Velocity Error Model

The primary sources of error are modelled as follows. Given the true specific force and angular velocity measurements, their noisy counterparts $\tilde{\mathbf{f}}_{ib}^b$ and $\tilde{\boldsymbol{\omega}}_{ib}^b$ are given by:

$$\tilde{\mathbf{f}}_{ib}^b = \mathbf{f}_{ib}^b + \mathbf{b}_a + \mathbf{n}_a \tag{6}$$

$$\tilde{\boldsymbol{\omega}}_{ib}^b = \boldsymbol{\omega}_{ib}^b + \mathbf{b}_g + \mathbf{n}_g \tag{7}$$

where \mathbf{b}_a and \mathbf{b}_g represent the biases on accelerometers and gyroscopes measurements, respectively, and \mathbf{n}_a and \mathbf{n}_g are the corresponding random noises, assumed to follow a zero-mean normal distribution.

3.4. Navigation Solution Update

Mechanisation equations (in discrete form) [33] are used to update the solution of inertial navigation at the next time instant, using measurements of angular velocity $\boldsymbol{\omega}_{ib}^b$ and specific force \mathbf{f}_{ib}^b from the IMU sensors. Since mechanisation equations are the result of approximations and may introduce errors, an optimised version was used.

Orientation update: if the angular velocity remains constant during the integration interval, i.e., if τ is sufficiently small, the updated rotation matrix representing the attitude of the vehicle is given by

$$\mathbf{C}_b^n(t + \tau) \approx \mathbf{C}_b^n(t) \left(\mathbf{I} + [\boldsymbol{\omega}_{ib}^b]_x \tau \right) - (\boldsymbol{\Omega}_{ie}^n + \boldsymbol{\Omega}_{en}^n) \mathbf{C}_b^n(t) \tau, \tag{8}$$

where

$$\boldsymbol{\Omega}_{ie}^n = [\boldsymbol{\omega}_{ie}^n]_x = \boldsymbol{\omega}_{ie} \begin{pmatrix} 0 & \sin(L_b) & 0 \\ -\sin(L_b) & 0 & -\cos(L_b) \\ 0 & \cos(L_b) & 0 \end{pmatrix} \tag{9}$$

is the skew-symmetric matrix of the rotation vector of the Earth, $\boldsymbol{\omega}_{ie}$, resolved in the navigation frame,

$$\boldsymbol{\Omega}_{en}^n = [\boldsymbol{\omega}_{en}^n]_x \Leftarrow \boldsymbol{\omega}_{en}^n \begin{pmatrix} v_{eb,E}^n / R_E(L_b) + h_b & & \\ -v_{eb,E}^n / R_N(L_b) + h_b & & \\ v_{eb,E}^n \tan(L_b) / R_E(L_b) + h_b & & \end{pmatrix} \tag{10}$$

is the skew-symmetric matrix of the *transport rate* vector due to the rotation of the navigation frame with respect to the Earth, R_E and R_N are respectively the radius of transverse curvature and the radius of curvature of the meridian at that point. We further define the attitude axis update matrix as the coordinate transformation matrix from the body reference frame at the end of the update ($b+$) to the body reference system at the beginning ($b-$)

$$\mathbf{C}_{b+}^{b-} = \exp[\boldsymbol{\alpha}_{ib}^b]_x = \sum_{r=0}^{\infty} \frac{[\boldsymbol{\alpha}_{ib}^b]_x^r}{r!}, \tag{11}$$

where $\boldsymbol{\alpha}_{ib}^b = \boldsymbol{\omega}_{ib}^b \tau$ is the attitude increment. Truncating the formula to the fourth order gives the Rodrigues formula, used to calculate:

$$\mathbf{C}_b^n(t + \tau) \approx \left[\mathbf{I} - \left(\boldsymbol{\Omega}_{ie}^n + \frac{1}{2} \boldsymbol{\Omega}_{en}^n(t) + \frac{1}{2} \boldsymbol{\Omega}_{en}^n(t + \tau) \right) \tau \right] \mathbf{C}_b^n(t) \mathbf{C}_{b+}^{b-} \tag{12}$$

where $\boldsymbol{\Omega}_{en}^n(t + \tau)$ is calculated from $L_b(t + \tau), \lambda(t + \tau), h_b(t + \tau)$.

Velocity update:

$$\mathbf{v}_{eb}^n(t + \tau) = \mathbf{v}_{eb}^n(t) + [\mathbf{f}_{ib}^n + \mathbf{g}_b^n(L_b, h_b) - (\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n) \mathbf{v}_{eb}^n(t)] \tau \tag{13}$$

where the acceleration due to gravity, \mathbf{g}_b^n , is modeled as a function of latitude and height, and \mathbf{f}_{ib}^n denotes the measured specific force resolved in the local navigation frame using the estimated \mathbf{C}_b^n .

Position update:

$$\mathbf{h}_b(t + \tau) \approx \mathbf{h}_b(t) - \frac{1}{2}[\mathbf{v}_{eb,D}^n(t) + \mathbf{v}_{eb,D}^n(t + \tau)] \quad (14)$$

$$\mathbf{L}_b(t + \tau) \approx \mathbf{L}_b(t) + \frac{1}{2} \left[\frac{\mathbf{v}_{eb,N}^n(t)}{R_N(\mathbf{L}_b(t)) + \mathbf{h}_b(t)} + \frac{\mathbf{v}_{eb,N}^n(t + \tau)}{R_N(\mathbf{L}_b(t)) + \mathbf{h}_b(t + \tau)} \right] \tau \quad (15)$$

$$\lambda_b(t + \tau) \approx \lambda_b(t) + \frac{1}{2} \left[\frac{\mathbf{v}_{eb,E}^n(t)}{(R_N(\mathbf{L}_b(t)) + \mathbf{h}_b(t)) \cos \mathbf{L}_b(t)} + \frac{\mathbf{v}_{eb,E}^n(t + \tau)}{(R_N(\mathbf{L}_b(t + \tau)) + \mathbf{h}_b(t + \tau)) \cos \mathbf{L}_b(t + \tau)} \right] \tau \quad (16)$$

In order to have the velocity and position accuracy update, we considered the use of the mean transformation matrix $\bar{\mathbf{C}}_b^n$ in the transformation of the specific force in the NED coordinate system:

$$\bar{\mathbf{C}}_b^n = \mathbf{C}_b^{n-} \mathbf{C}_b^{b-} \quad (17)$$

where

$$\mathbf{C}_b^{b-} = \mathbf{I} + \frac{1 - \cos |\boldsymbol{\alpha}_{ib}^b|}{|\boldsymbol{\alpha}_{ib}^b|^2} [\boldsymbol{\alpha}_{ib}^b]_X + \frac{1}{|\boldsymbol{\alpha}_{ib}^b|^2} \frac{1 - \sin |\boldsymbol{\alpha}_{ib}^b|}{|\boldsymbol{\alpha}_{ib}^b|} [\boldsymbol{\alpha}_{ib}^b]_X^2 \quad (18)$$

obtaining

$$\mathbf{f}_{ib}^n = \bar{\mathbf{C}}_b^n \mathbf{f}_{ib}^b, \quad \bar{\mathbf{C}}_b^n = \mathbf{C}_b^{n-} \mathbf{C}_b^{b-} - \frac{1}{2} (\boldsymbol{\Omega}_{ie}^n + \boldsymbol{\Omega}_{en}^n) \mathbf{C}_b^{n-} \tau \quad (19)$$

3.5. ZUPT Detection Algorithm

To make the Algorithm 1 more efficient, the ZUPT algorithm is used to correct errors accumulated in the inertial navigation data when the system is stationary [34]. Thus, when the average velocity is below a fixed threshold of 0.5 m/s for a fixed time period of 4 s, the algorithm assumes the system is stationary and updates the current attitude and position based on the averages of the previous values.

3.6. Kalman Filter

The state vector of the Kalman filter includes orientation, velocity and position errors used in Equations (1)–(5), as well as the biases of the accelerometer and gyroscope, and is given by

$$\hat{\mathbf{x}}_k = [\delta \boldsymbol{\psi}_{eb}^n \quad \delta \mathbf{v}_{eb}^n \quad \delta \mathbf{p}_b \quad \mathbf{b}_a \quad \mathbf{b}_g]^T, \quad (20)$$

where the position error is

$$\delta \mathbf{p}_b = [\delta \hat{L}_b \quad \delta \hat{\lambda}_b \quad \delta \hat{h}_b]^T. \quad (21)$$

Whenever corrections from the integration filter are applied, the values of the corresponding states are reset to zero. The algorithm implemented for the Kalman filter is as follows.

1. Calculation of the covariance matrix \mathbf{P}_k^- of the prediction error at instant k :

$$\hat{\mathbf{x}}_k^- = 0 \quad (22)$$

$$\mathbf{P}_k^- = \boldsymbol{\Phi}_{k-1} \mathbf{P}_{k-1}^+ \boldsymbol{\Phi}_{k-1}^T + \mathbf{Q}_{k-1} \quad (23)$$

where \mathbf{Q} is the system noise covariance matrix, and the transition matrix $\boldsymbol{\Phi}$ is obtained by computing the expected value of the time derivative for each state.

2. Calculation of the Kalman gain matrix:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1}, \quad (24)$$

where \mathbf{H}_k is the measurement matrix and \mathbf{R}_k is the measurement noise covariance matrix at epoch k .

3. Filtering of the system state at instant k based on GNSS measurements, and calculation of the covariance matrix of the filtered estimation error:

$$\hat{\mathbf{x}}_k^+ = \mathbf{K}_k \delta \mathbf{z}_k^- \quad (25)$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (26)$$

where $\delta \mathbf{z}_k^-$ is the vector of observations (i.e., the difference between position and velocity measured by the GNSS and the corresponding values estimated by the inertial navigation system). It's important to note that all estimated quantities are derived based on preceding correction. The matrices Φ_{k-1} , \mathbf{Q}_{k-1} , \mathbf{R}_k used in this study are resolved in a local navigation (NED) frame and can be found in Appendix A of this paper.

4. Experimental Setup

This section describes the materials and methods used to test the navigation system. The testing process covered all the elements: sensors, services, communication elements, power consumption, synchronisation and especially the data integration/fusion for the PNT process. In the first part, this section describes the experimental activities to evaluate the performance of sensors/services used in the navigation process: GNSS receiver, GNSS Augmentation (PP) service and inertial sensor. Afterwards, it is shown the experimental methodology carried out during the final test, evaluating the impact of the sensor fusion algorithm on the navigation process in an on-road environment.

Figure 6 presents a detailed schematic of the experimental verification process conducted for each key system sensor. The purpose of this experimental setup is to evaluate each subsystem and check the individual performances of each sensor. The test also covers the process of interconnection, configuration, decoding, and calibration of the sensors involved. The verification process of the GNSS system (GNSS Antenna + SIRIUS ROVER: ZED-F9P) is a comprehensive series of experimental tests where all the components of the GNSS subsystem are thoroughly evaluated: the antenna, the communication interfaces, the SW block dedicated to decoding, as well as the performances in terms of position calculation (PVT). Furthermore, the impact of the PointPerfect (Augmentation service) is evaluated. Table 1 shows some performance results that cover the cases of using the GPS and Galileo constellations without PointPerfect service and the case of using the full capability of the GNSS receiver (GNSS + PointPerfect augmentation service). The reference system (known positions) used in the performance calculations was derived from previous long-term PPP-RTK measurements. The inertial measurement system (Xsens—MTi 630 AHRS) is also verified in a stationary scenario. In addition to validating the interfaces, configuration, and SW components dedicated to coding/decoding, this process allows the creation of a data set that was used to characterise the inertial error in the GNSS + IMU integration process.

Table 1. Experimental results of the satellite navigation module with two different configurations: using two GNSS constellations (GPS + GALILEO) and the GNSS (all constellations) integrated with the PointPerfect Augmentation service (GNSS + PointPerfect). For each parameter, μ and σ denote statistical mean and standard deviation, respectively.

| KPI ¹ | Parameters | GPS + GALILEO [μ, σ] | GNSS and PointPerfect [μ, σ] |
|------------------------------|----------------------------|---------------------------------|---|
| Accuracy | 3D Position [m] | 2.26, 0.52 | 0.10, 0.25 |
| | Horizontal Position [m] | 1.70, 0.44 | 0.06, 0.18 |
| | Vertical Position [m] | 1.46, 0.41 | 0.08, 0.18 |
| Availability and Continuity | SVs Used [U] | 11, 2 | 26, 2 |
| | SVs Tracked [U] | 24, 3 | 36, 3 |
| | SVs Received [U] | 28, 3 | 37, 3 |
| | C/N0 [dB-Hz] | 36.47, 1.5 | 34.31, 1.3 |
| | GDOP | 2.3, 0.6 | 1.7, 0.2 |
| | HDOP | 1.3, 0.5 | 0.8, 0.13 |
| | VDOP | 1.4, 0.4 | 1.2, 0.2 |
| Integrity | North Protection Level [m] | 8.86, 4.69 | 2.09, 5.59 |
| | East Protection Level [m] | 10.93, 6.61 | 2.44, 6.03 |
| | Down Protection Level [m] | 11.75, 6.41 | 4.08, 7.48 |
| TTFF | TTFF ² [s] | 28.75, 1.71 | 26.34, 1.33 |
| Other performance parameters | 3D Velocity Error [m/s] | 0.26, 0.09 | 0.13, 0.03 |
| | Time Accuracy [ns] | 0.0036, 0.0009 | 0.0011, 0.0009 |
| | TDOP | 1.1, 0.3 | 0.8, 0.1 |

¹ This table groups together the information collected that could provide answers to the KPIs organised in the first column. In the last years, some constellations have defined specific KPIs and parameters, so note that the parameters shown are only a portion of these. ² TTFF is time elapsed between a *coldstart* and a valid *Fix* detected in the GNSS receiver.

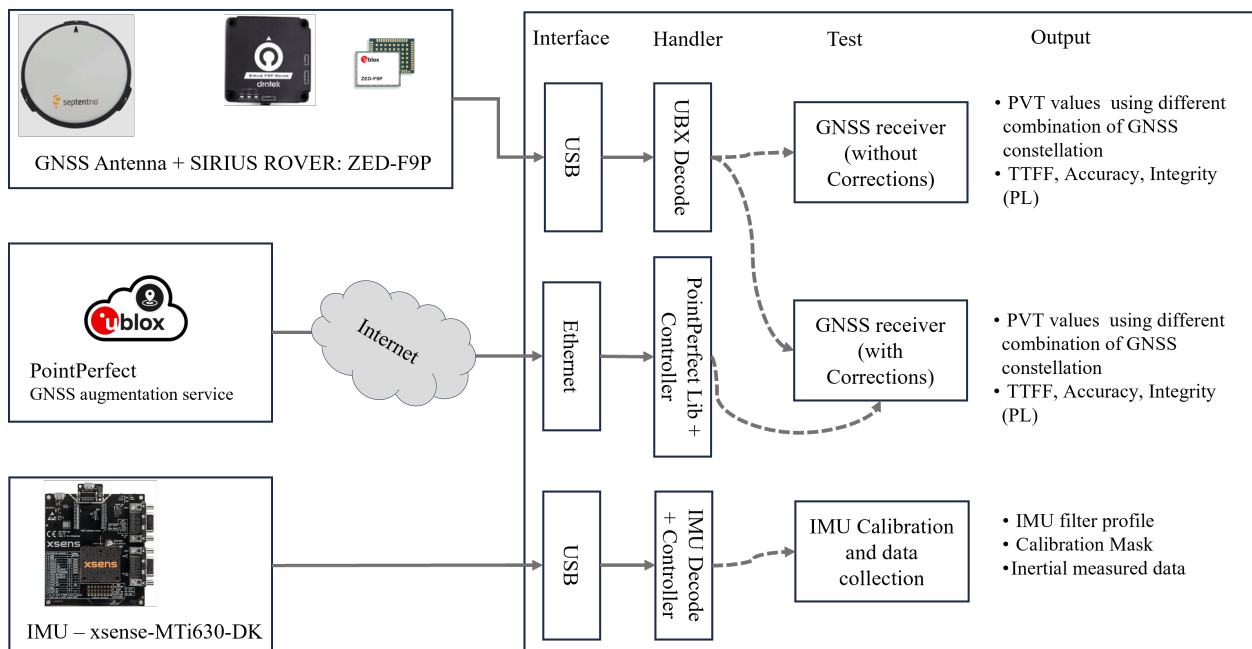


Figure 6. Experimental setup for sensor verification tests.

4.1. Sensors Verification

Before a complete and definitive test of the OBU in a real environment, we present a verification of the performance of some fundamental system elements. Particular interest falls on the satellite navigation system, which includes two fundamental components: the GNSS multi-constellation/multi-frequency receiver and the Augmentation Service, which

provides the relevant atmospheric and clock corrections. The sensor verification process allows the performance to be characterised locally through appropriate configuration. For example, it can evaluate satellite availability and coverage (DOP), the average Time To First Fix (TTFF), and the most effective combinations of constellations and frequencies in the case of GNSS.

4.2. Xsens MTi630 AHRS

The tests conducted on the inertial navigation component focused on the calibration process and the correct functionality of the sensor. For this, we used the Xsens Development Kit MTi630 connected to a PC and a SW dedicated to the configuration and storage of data measured by the sensor. The experiment configuration just involves connecting the PC and the MTi-630-DK using the corresponding USB interface. The acquisition SW is implemented in Python and uses the manufacturer open-source API (Xsens Device API) as a library. To summarise the flow followed by the acquisition SW: a first “startup” phase dedicated to the creation of the XDA library objects, scanning of the ports/interfaces and configuration of the device; a second “processing and storage” step dedicated to the manipulation and storage of the inertial data; and finally the step of closing and disconnection of the objects, ports and files used. In addition, a summary of the IMU configuration profile, including motion filters, calibration information and inertial system data, is stored.

4.3. GNSS and Augmentation Service

The whole satellite navigation system testing process is conducted in two steps: one dedicated only to verify the correct functionality of the GNSS receiver, and the other one to evaluate the impact of using the PointPerfect service. For the first step, the HW involved includes the GNSS antenna, the GNSS receiver, and a PC that connects to the receiver via the USB port. In the second step, an Internet connection is needed to access the PointPerfect service. This test is performed in a controlled outdoor environment with the antenna placed on a vehicle at rest.

The tests described in this section focus on the evaluation of the performance of the satellite navigation system, considering the most important parameters in automotive positioning: TTFF, Accuracy, Integrity and satellite availability metrics. The GNSS data acquisition process (with and without corrections) is carried out using an acquisition SW that has the following flow: *startup* (configuration of interfaces, objects, devices and services); *processing* (main cycle in charge of decoding the UBX data coming from the GNSS sensor, the corrections coming from the PointPerfect service, and the file storage process); and finally the *close* and *disconnection* phase of the elements used. In the case of the TTFF tests, repetitions are performed from a “coldstart” and, afterwards, the mean is applied to the measured values.

4.4. Sensor Integration: On-Road Navigation Test

The integration or fusion process is the core of the multi-sensor approach to navigation. This section describes the verification process applied to the algorithm described in Section 3. Having characterised the sensors/services in terms of noise, accuracy and response times, the improvements introduced by the integration process can be quantified. In this case we use a Raspberry Pi 4 Model B (On Board Controller), the satellite navigation system described in the previous section (Antenna, Receiver and GNSS Augmentation Service), the MTi630-DK, everything installed within a compact city car. Figure 7 specifies the coordinate system and reference frame used. In this case, we match the inertial reference frame (i-frame) with the body/vehicle reference frame (b-frame). The conversion process to a unique navigation system (n-frame) uses the corresponding transformations (i.e., Rotation Matrix). Likewise, the reference system used by the ground-referenced GNSS (e-frame) shall be in a compatible reference system to facilitate the fusion process.

The navigation information is captured during the experiment using a data-logger script. The main elements to be stored for future analysis are GNSS information, inertial

measurements, timing information, sensor fusion process results, and Kalman filter status information. This experiment has been conducted in the following sequence.

- *Startup phase*
 1. Power-up of the whole system: vehicle and OBU
 2. Initialisation of the acquisition SW
 - (a) Initialise libraries, dependencies, SW modules and objects: MQTT client/server, parser/deparsed UBX, XDA, PP credentials and files
 - (b) Sensor (interface) detection
 - (c) Calibrate and/or configure sensors
 - (d) Prepare for data handling (event-based), KF and synchronisation structures
 - (e) Change operation mode (in sensors) to measurement mode
- *Processing and Measurement phase*
 3. Handle incoming data: queuing, synchronisation, data availability and status of the fusion algorithm
 4. Write the information (synchronously) to the logger files
 5. Verify the conditions for the end of the experiment
- *Exit and Close phase*
 6. Save the final status of the system elements: GNSS, Inertial Unit and SF data
 7. Close open devices and files
 8. Clear and release all the structures used
 9. Print the successful experiment message

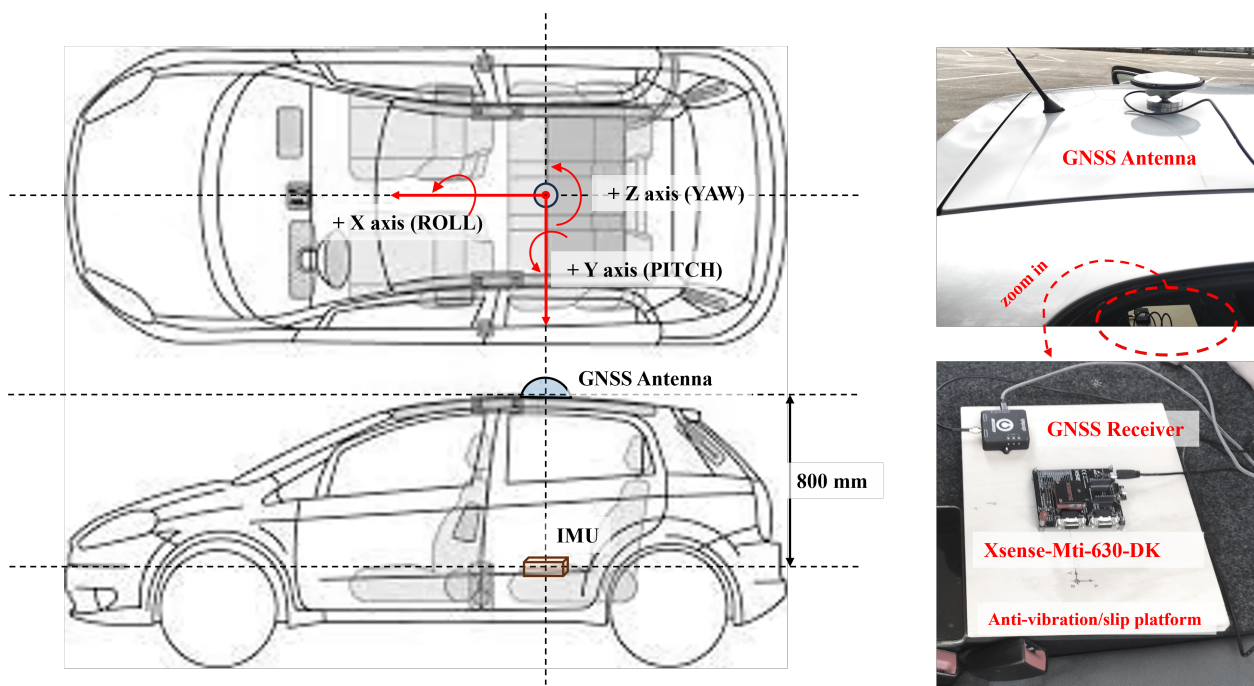


Figure 7. Coordinate system, frame and hardware for final Integration Test.

The On-road experiment has been conducted in the parking area of the Tecnopolo of Abruzzo, city of L'Aquila, Italy, and lasted 240 s (4 min). During the test, a trajectory along the marked tracks has been followed, with acceleration intervals and stops, to simulate a typical urban scenario. For this experiment, the inertial sensor samples the data with a frequency of 100 Hz, and the GNSS receives one updated PVT message every second.

5. Results

The most relevant results in evaluating the sensors lie in the accuracy of the satellite navigation system and the improvement obtained using an external GNSS augmentation service. In addition, considering the automotive application, we provide some statistics on the power consumption of the OBU. This section concludes by showing the final results of the sensor fusion application.

5.1. GNSS Performance

The first GNSS key issue analysed was the TTFF measured starting from a “coldstart” signal to the receiver until it reaches a 3D Fix. Experimental results show a mean value slightly below 30 s. This result has been verified and is consistent with the data provided by the manufacturer. However, even if this result is satisfactory, it is important to note that a lower TTFF may be required in automotive applications. This is why the OBU foresees the storage of the last position/status of the vehicle after it has been powered off, functionalities supported thanks to the internally incorporated Power Management and UPS system. The backup battery connected to the Protected 18650 Li-Ion Separable Battery Holder has a capacity of 3000 mAh (at 3.7 V). Assuming OBU consumption in the high operating mode (average power of 5 W and average current of 1 A), the system autonomy, regarding energy, is about 2 h.

Table 1 compiles the results obtained in the evaluation process by accumulating data for more than 5000 s (approximately one and a half hours) at different fixed known positions and times along a given day. The first two columns of the table match the key performance indicators (KPI) with the corresponding measured parameters. Two measurement sets have been collected: one using two GNSS constellations (GPS + GALILEO) without GNSS corrections and the other, including the PointPerfect Augmentation service and using all four GNSS constellations. Measurements collected at different times during multiple trials, are summarized statistically through their mean and variance. Accuracy measures are improved by using the augmentation service, as expected: performance obtained using the GALILEO and GPS constellations simultaneously, but without atmospheric and clock correction, are significantly lower than the ones obtained when the full potential of the GNSS receiver is used. Exploitation of the PointPerfect augmentation service noticeably improves the mean of the accuracy values and also reduces their variance, leading to an improved system’s reliability. Although requirements of the EMERGE reference use cases are satisfied, the recommendation points towards a configuration with a Multi-constellation/multi-frequency approach plus GNSS Augmentation Service. With the perspective to support high and full automated Connected and Automated Driving (CAD) functions, as detailed in 3GPP and ETSI latest specifications [35], a reduction of the order of magnitude (from metres to centimetres) in the accuracy parameters in the satellite navigation process is more than justified.

One of the advantages of using configurations where all available satellite resources in view are utilised is the ability to select (exclude) the SVs (space vehicles) to be used. This has a direct impact on other satellite navigation KPIs. When using all the potentialities of the receiver, it can be observed doubling of the number of used satellites, as well as an improvement in the Dilution of Precision (DOP) parameters that mitigate the (mathematical) error in the calculation of the position due to the effects of the spatial distribution of the GNSS satellites.

Values of the Integrity, which is a crucial element in automotive applications, may be insufficient for some specific automotive use cases, despite the usage of the GNSS and PointPerfect configuration. This is why other sensors such as Lidar, Radar and Video cameras must be incorporated in specific scenarios and applications. In fact, the EMERGE project foresees the integration of sensors (as external components) in the navigation OBU.

5.1.1. Sensor Integration

This section presents the experimental results obtained using the sensor fusion (SF) algorithm described in Section 3. To assess the algorithm's performance under realistic conditions, we conducted tests in an open-sky environment near our laboratory, ensuring clear visibility. We intentionally opted for this setting to maintain flexibility in test conditions and trajectory definition. We decided to focus our tests on real-world scenarios characterized by urban signal degradation, therefore augmenting the GNSS data further would not have provided significant insights. Our aim was to underscore the practical utility of INS in complementing GNSS signals, particularly in scenarios where augmentation may not substantially contribute to improving accuracy or robustness. Hence, we opted to concentrate on the primary GNSS signals from GPS and Galileo, augmented by the INS data. To replicate inaccuracies typically encountered in urban settings, a fault injection operation was applied to the GNSS-only signal (GPS + GALILEO). To simulate the effect of random variations and inaccuracies in the GNSS signal, we employed a Brownian motion process, that generated random additive variations for latitude, longitude, and altitude. In this way, we obtained the estimates of position provided by a degraded GNSS. Subsequently, we evaluated the impact of data fusion in terms of accuracy, stability, and response to errors that affected GNSS without augmentation and IMU sensors.

The tests were conducted near the Abruzzo Technopole, located at Strada Statale 17 Loc. Boschetto di Pile, 67100 L'Aquila (AQ), Italy.

Figures 8–12 illustrate the five test scenarios under consideration. Each figure depicts the starting point of the vehicle detected by each sensor with a circle. The dashed line represents the ground truth, derived from the GNSS signal (all constellations) with PP corrections applied every 5 s. The solid purple line represents the trajectory obtained with GNSS-only, while the solid blue line represents the trajectory obtained from the sensor fusion algorithm, integrating IMU/GNSS data. A preliminary analysis reveals that in each scenario, the trajectory from GNSS-only, which has undergone degradation, deviates further from the ground truth compared to the trajectory returned by the data fusion algorithm.

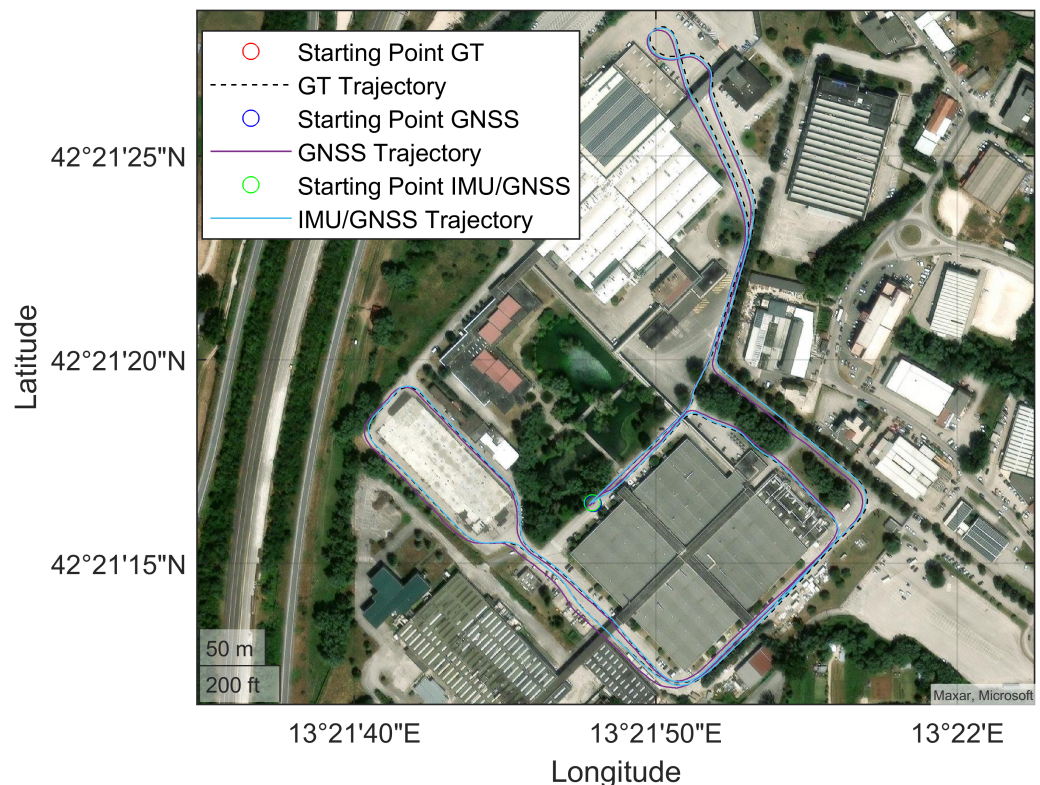


Figure 8. Scenario 1.

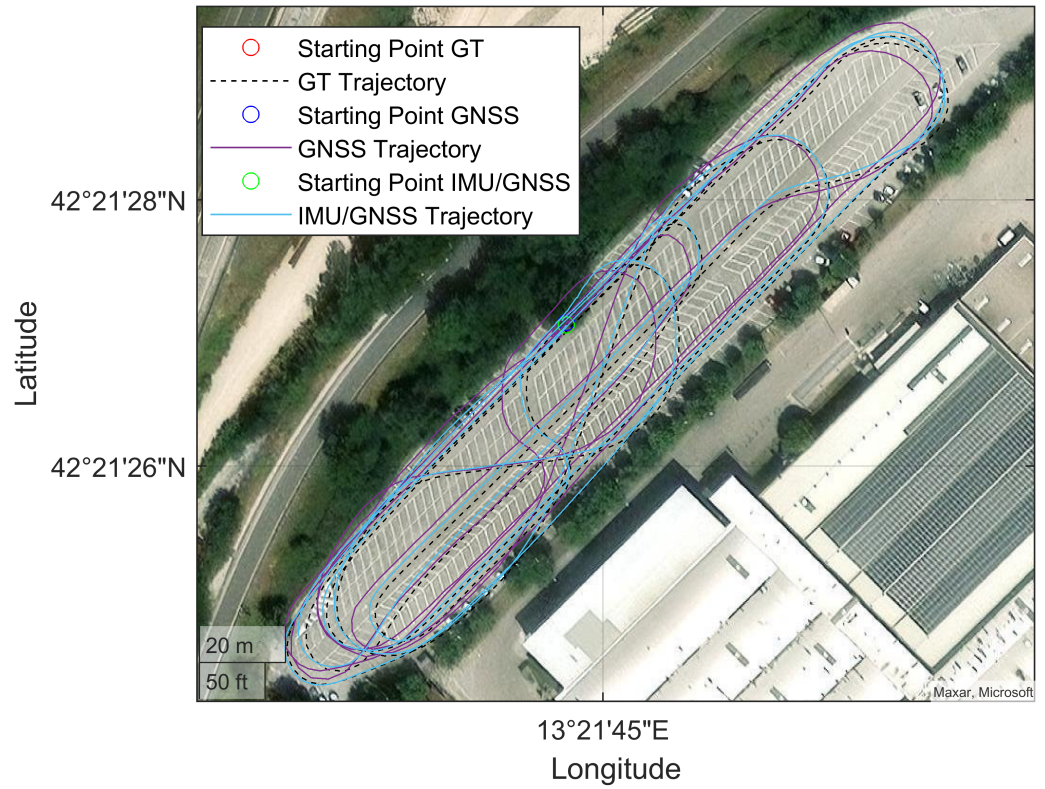


Figure 9. Scenario 2.

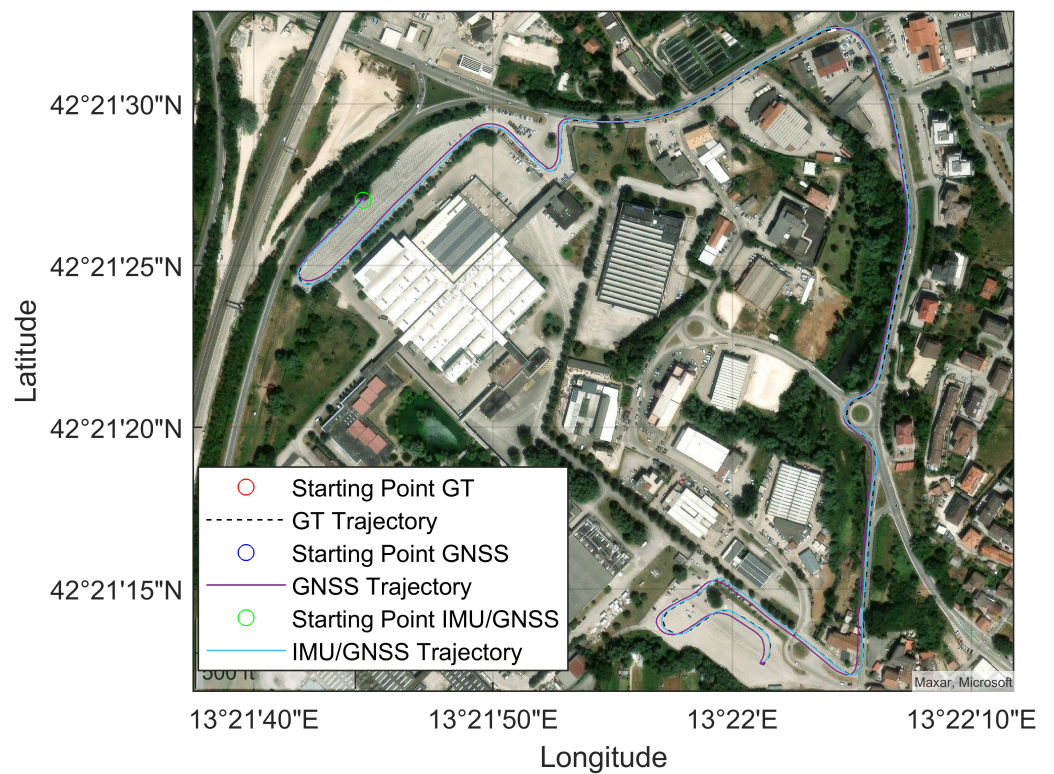


Figure 10. Scenario 3.

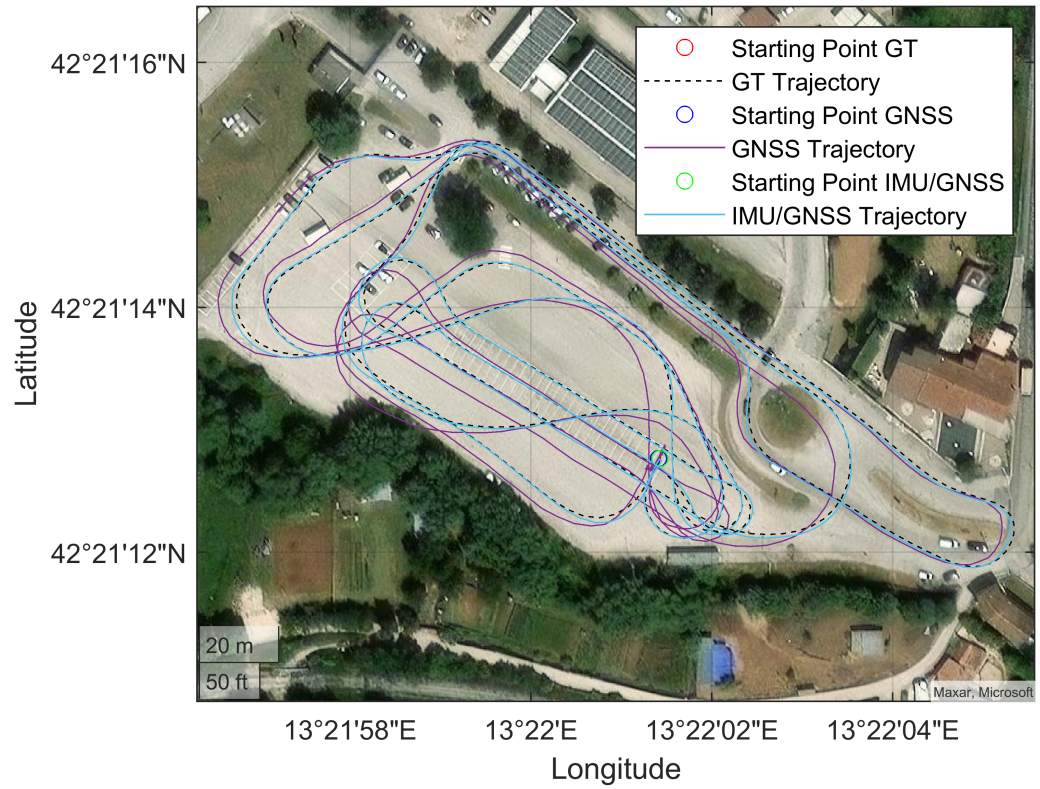


Figure 11. Scenario 4.

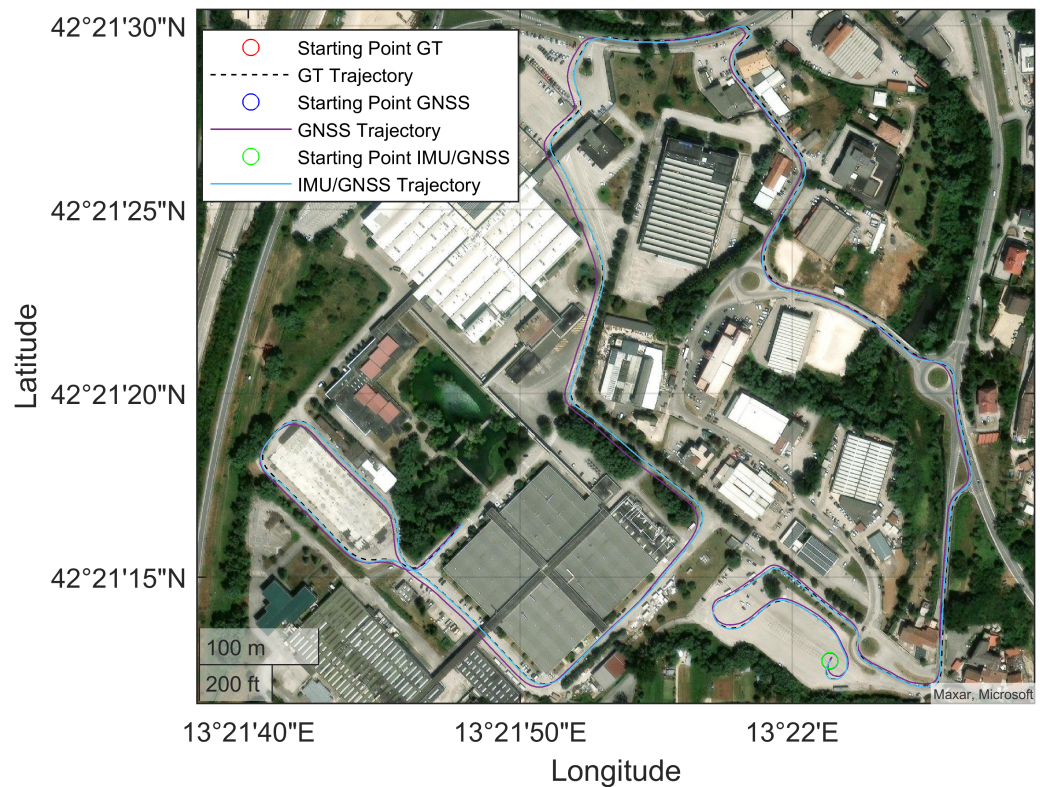


Figure 12. Scenario 5.

Figures 13–17 compare, on one side, the graphs of latitude, longitude, and altitude, and on the other side, the error on the same calculated as the distance from the ground truth. The gray line corresponds to GNSS-only, the blue line represents the result of IMU/GNSS

integration, and the dashed line represents the ground truth. Here, it becomes even more evident how the algorithm reduces disturbances introduced by GNSS and minimizes error.

This observation is also numerically confirmed in Table 2 which presents the Root Mean Square Error data calculated with respect to the ground truth for the position obtained with GNSS-only and for the position estimated by the SF algorithm.

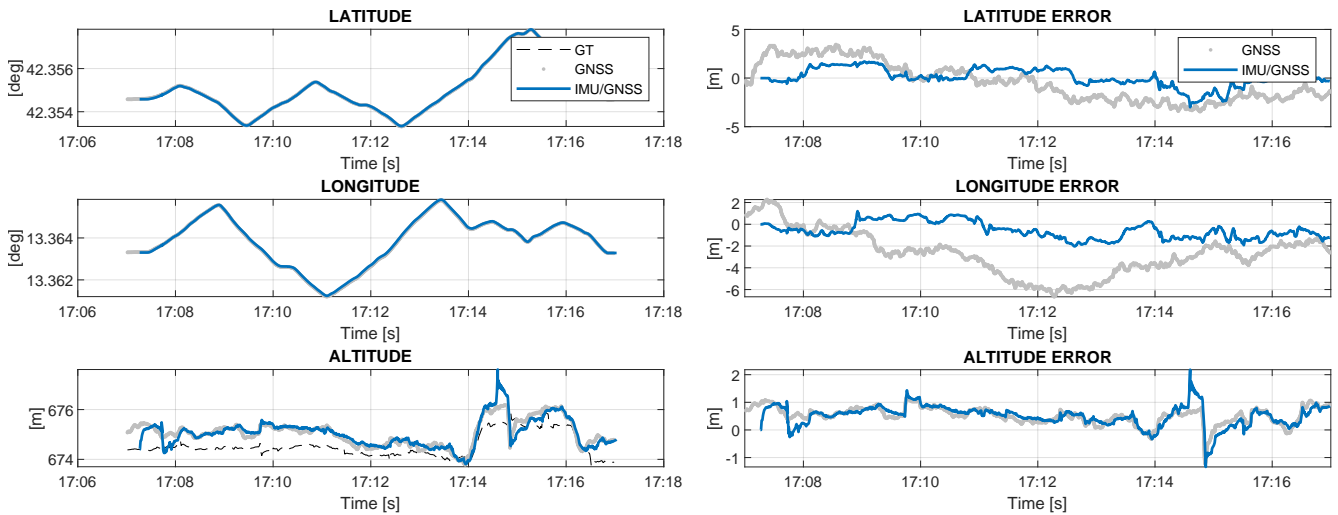


Figure 13. Scenario 1.

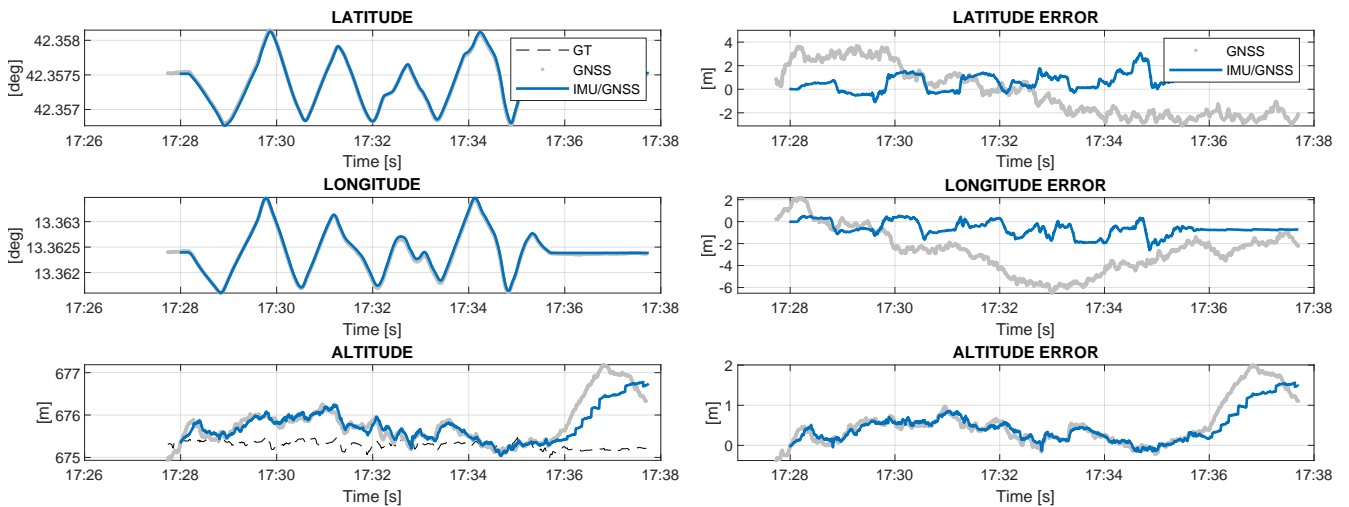


Figure 14. Scenario 2.

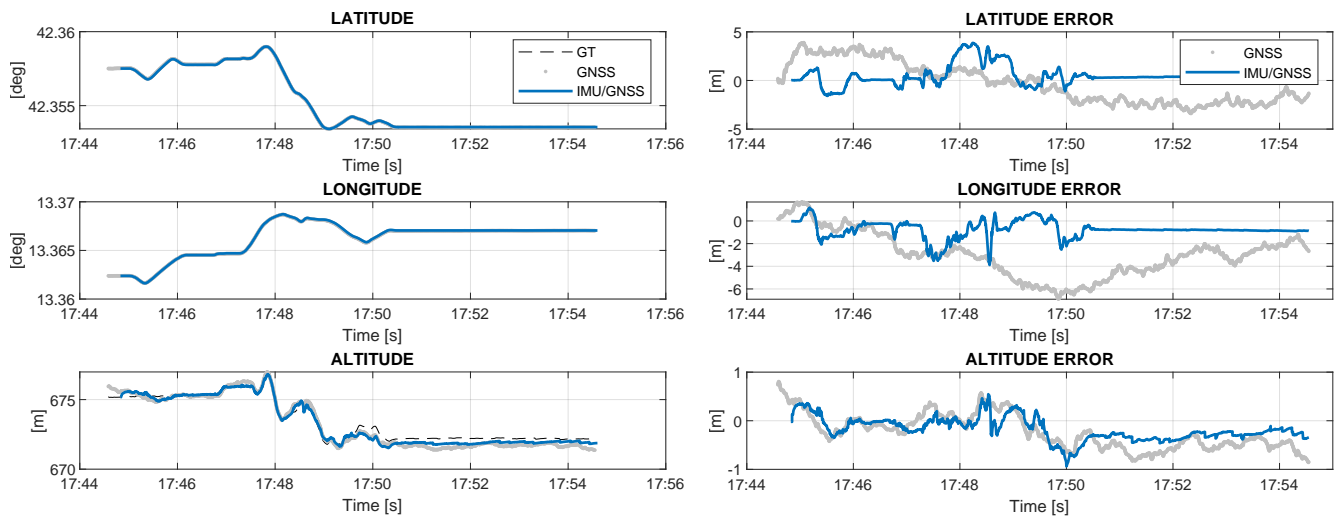


Figure 15. Scenario 3.

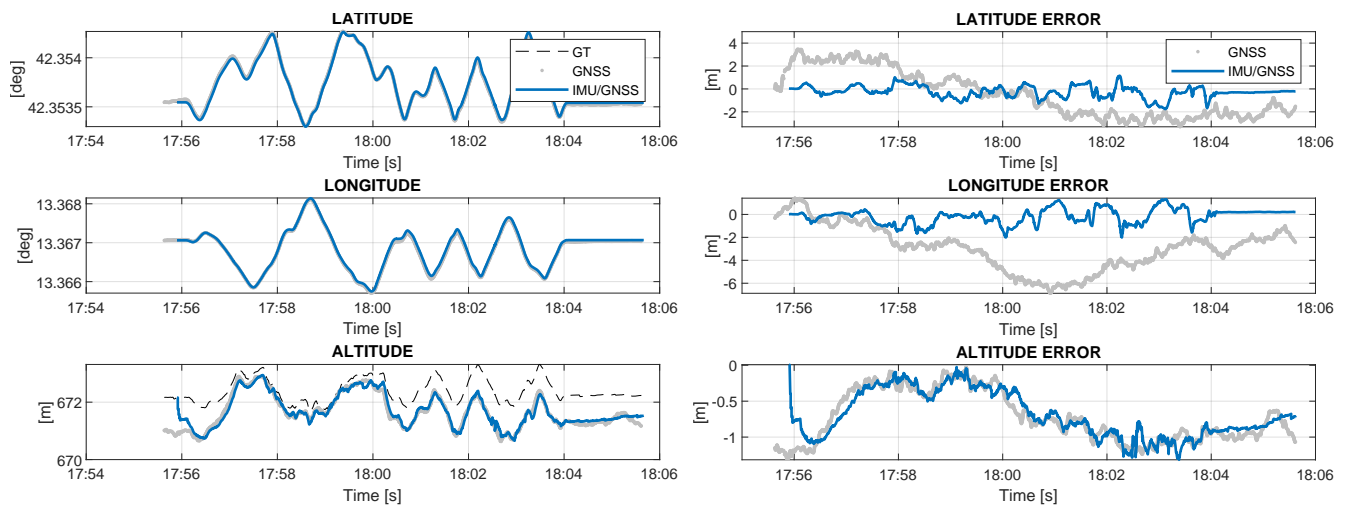


Figure 16. Scenario 4.

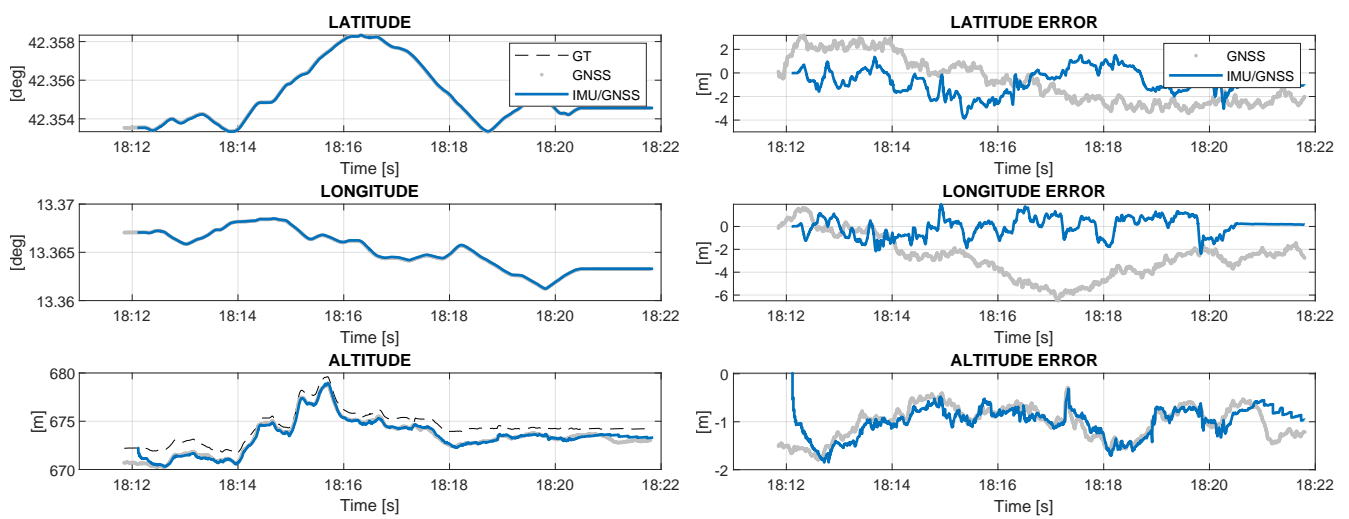


Figure 17. Scenario 5.

Table 2. Comparison between the estimated trajectory derived from degraded GNSS-only signals, simulating an urban environment, and the trajectory estimated by the SF algorithm using the Root Mean Square Error (RMSE) method, calculated relative to the ground truth.

| Scenario | RMSE GNSS [m] | RMSE SF [m] |
|----------|------------------------------------|------------------------------------|
| 1 | X = 1.9903, Y = 3.0736, Z = 1.5573 | X = 0.8272, Y = 0.8704, Z = 0.8553 |
| 2 | X = 2.0940, Y = 3.0223, Z = 1.5049 | X = 0.5824, Y = 0.9131, Z = 1.0307 |
| 3 | X = 1.7010, Y = 3.3717, Z = 1.7452 | X = 0.7041, Y = 1.1537, Z = 0.8255 |
| 4 | X = 1.6152, Y = 3.3778, Z = 1.8267 | X = 0.4870, Y = 0.6889, Z = 0.8041 |
| 5 | X = 1.6983, Y = 3.0971, Z = 1.8797 | X = 0.8312, Y = 0.8393, Z = 1.4352 |

These results indicate a substantial discrepancy between the positions estimated via GNSS and the actual positions measured in the field, and demonstrate that the combined use of IMU and GNSS significantly improved the accuracy of position estimation.

The velocity plots in Figures 18–22 confirm the previous analyses for the same scenarios, with a perfect overlap of the velocity estimates on all three axes for the ground truth, GNSS-only, and the result of data fusion. On the other side, the error on the three axes fluctuates slightly around zero, confirming the algorithm’s good performance.

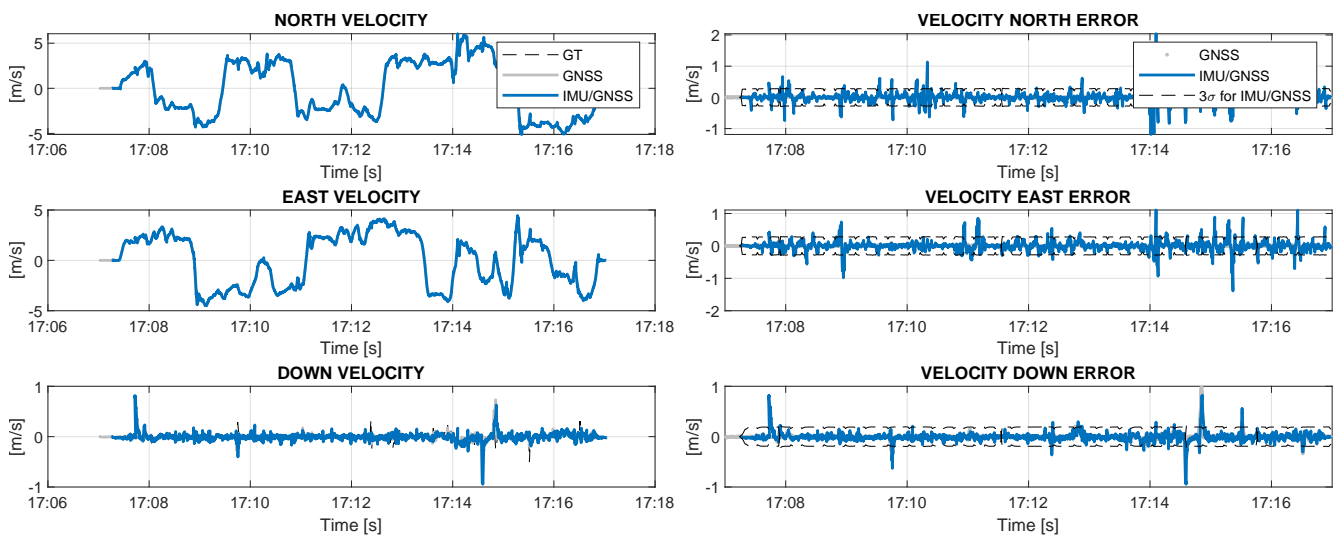


Figure 18. Scenario 1.

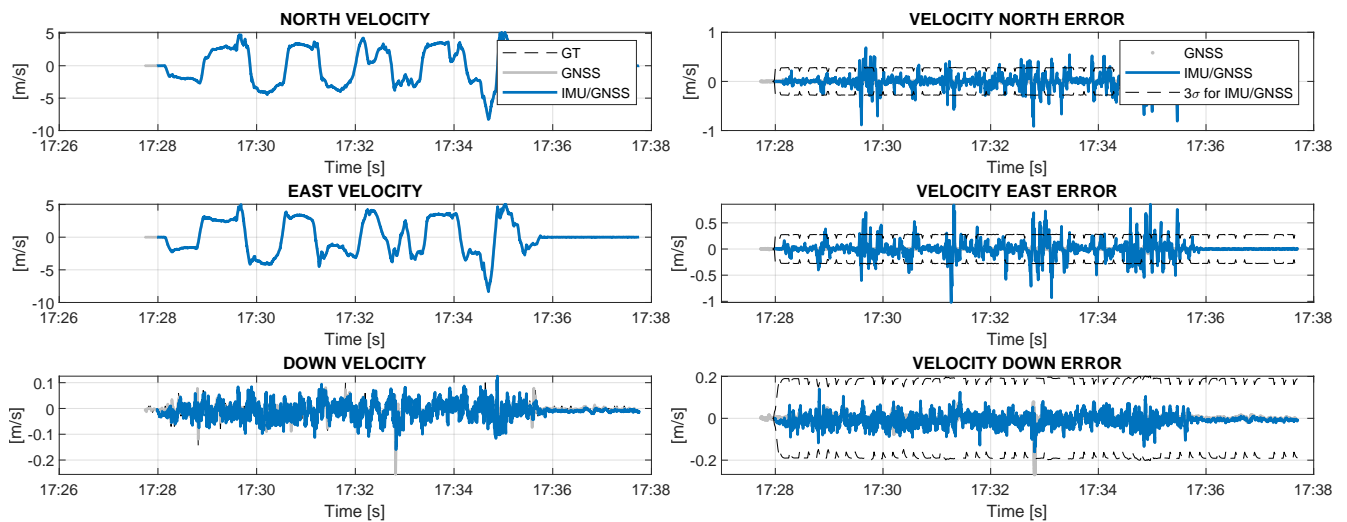


Figure 19. Scenario 2.

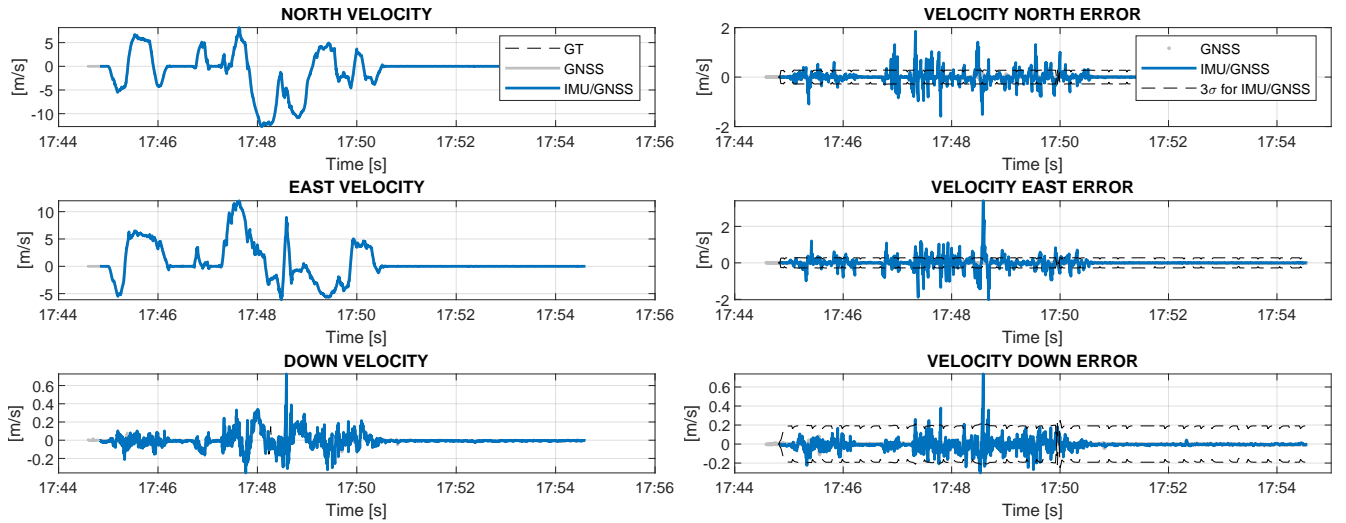


Figure 20. Scenario 3.

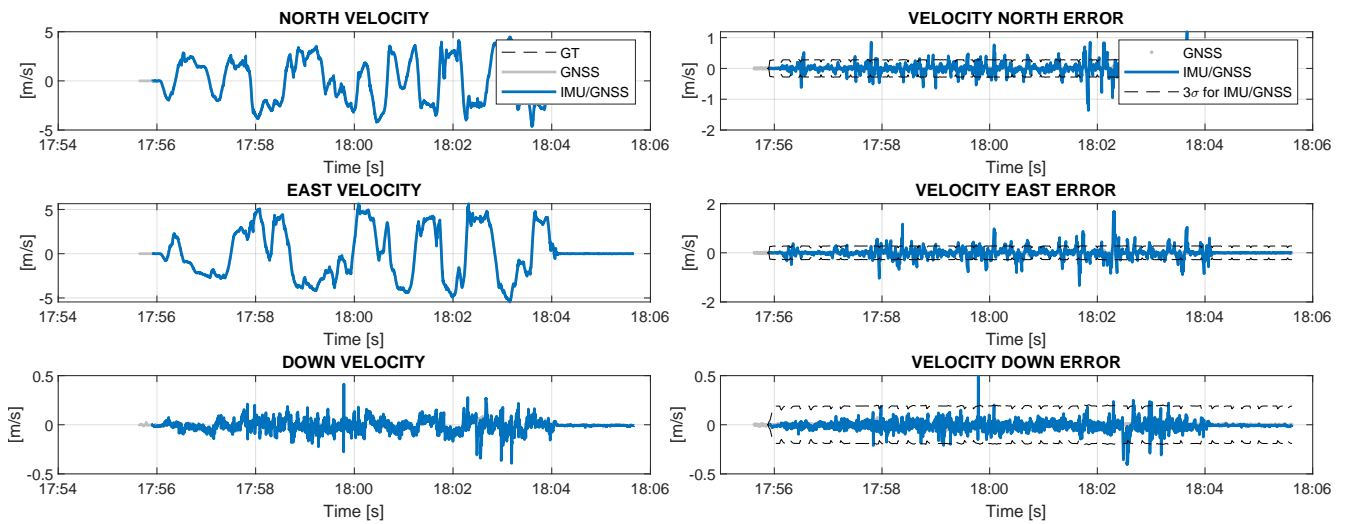


Figure 21. Scenario 4.

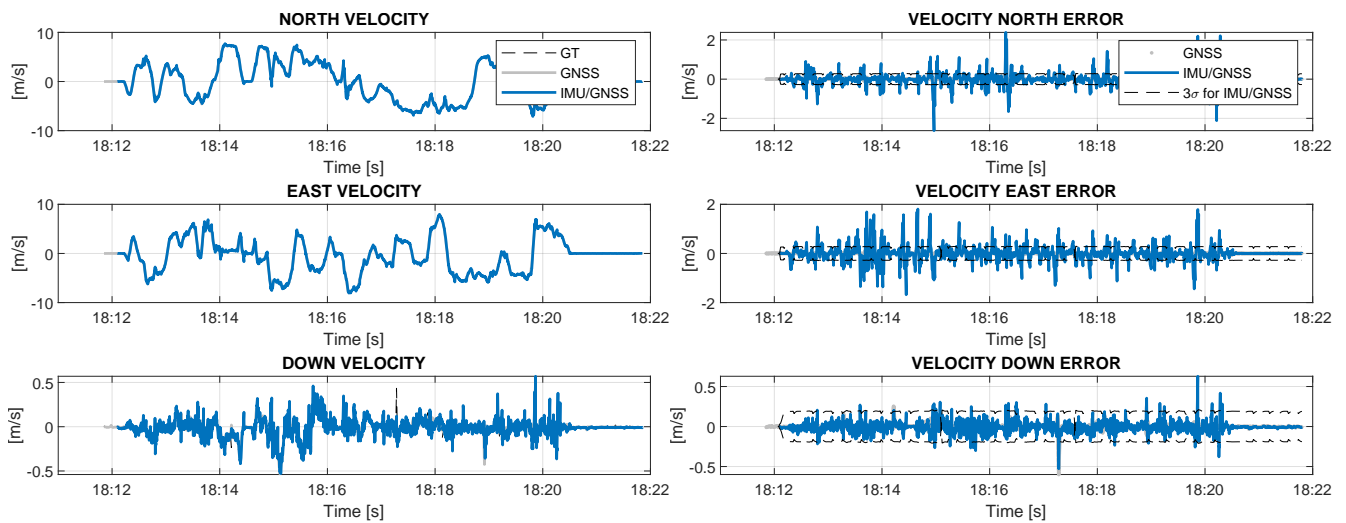


Figure 22. Scenario 5.

6. Conclusions

This paper presents the OBU architecture for the EMERGE project, and the implementation of the navigation system, based on a multi-sensor approach.

The proposal of a concrete architecture for OBU development provides a starting point in the automotive field for other application areas and services. Although in the test-bed used for this work most of the processing has been performed locally, the presented architecture can be applied to systems that foresee a microservice approach, cloud computing [36], and/or edge computing. This transformation entails moving some elements (mainly from the core layer of the onboard system architecture, i.e., layer 4 in Figure 1) out of the OBU and enhancing elements dedicated to communication, according to specific performance requirements. The modular architecture has helped in using the agile methodology for developing the product, facilitating the selection process of HW components and technologies, and error debugging.

Experimental tests conducted on inertial and GNSS sensors have demonstrated their accuracy, that satisfies performance requirements of the EMERGE project and justifies their use in real automotive scenarios. The GNSS receiver provides excellent performance when integrated with the u-blox augmentation system, i.e., PointPerfect, that enhances the accuracy of the satellite navigation system from metres to centimetres. This work uses the u-blox commercial IP plan, which provides corrections through an MQTT broker. The subscription to the root topic provides to the experiment all the set of corrections—i.e., orbits, bias, atmosphere, and clock—whenever they are available. The effectiveness of the multi-sensor approach has also been validated in challenging scenarios like urban contexts. In fact, tests documented in Section 5.1.1 show that merging inertial and satellite data provides a more accurate estimate of the trajectory, particularly when the GNSS is degraded.

Implementing an OBU in the automotive context also involves the power analysis of the system. The Power Management and UPS systems guarantee the operating autonomy needed by the OBU to conclude and store the states of the active processes.

Author Contributions: Conceptualisation, all authors; methodology, all authors; software, A.L.Z.S., V.I., M.B. and M.P.; validation, A.L.Z.S., V.I., M.P., M.B., F.V. and E.C.; formal analysis, all authors; investigation, all authors; resources, all authors; data curation, A.L.Z.S., V.I., M.P., M.B., F.V. and E.C.; writing—original draft preparation, A.L.Z.S., V.I., M.P., F.V. and E.C.; writing—review and editing, A.L.Z.S., V.I., M.P., M.B., F.V. and E.C.; visualisation, all authors; supervision, M.P., R.A., F.V., E.C., C.A. and A.M.; project administration, F.V. and M.P.; funding acquisition, all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by the project “Centre of Excellence on Connected, Geo-Localized and Cyber-secure Vehicles”—Italian Government (Grant Number: CIPE Resolution 70/2017) and the project EMERGE-Navigation (Ministry of Innovation and Made in Italy and Regione Abruzzo).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding authors, A.Z. and V.I., upon reasonable request.

Acknowledgments: The authors acknowledge the Ex-EMERGE centre, Radiolabs and Telespazio Spa for the support, resources and facilities provided. Special thanks to Graziano Battisti/DISIM-UNIVAQ for his invaluable technical assistance during this research. Their expertise and support have played a crucial role in successfully executing this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---------|---|
| ADAS | Advanced Driver Assistance System |
| AGNSS | Augmented GNSS |
| AHRS | Attitude and Heading Reference Systems |
| API | Application programming interface |
| AV | Autonomous Vehicle |
| C/N0 | Carrier-to-Noise Density Ratio |
| CAD | Connected and Automated Driving |
| COTS | Commercial Off-The-Shelf |
| DOP | Dilution Of Precision |
| ECEF | Earth-centred, Earth-fixed Coordinate System |
| EKF | Extended Kalman Filter |
| GNSS | Global Navigation Satellite System |
| GPIO | General Purpose Input/Output |
| GT | Ground Truth |
| HW | Hardware |
| IMU | Inertial Measurement Unit |
| INS | Inertial Navigation System |
| IoT | Internet of Things |
| IP | Internet Protocol |
| JSON | JavaScript Object Notation |
| KPI | Key Performance Indicators |
| MCU | Microcontroller Unit |
| MQTT | Message Queuing Telemetry Transport |
| MT | Motion Trackers |
| NED | North-East-Down |
| OBC | On Board Computer |
| OBU | On Board Unit |
| PNT | Positioning, Navigation and Timing |
| PP | PointPerfect |
| PPP | Precise Point Positioning |
| PPP-RTK | Precise Point Positioning - Real Time Kinematic |
| PVT | Position, Velocity and Time |
| RMSE | Root Mean Square Error |
| RTK | Real Time Kinematic |
| SBC | Single Board Computer |
| SDRAM | Synchronous Dynamic Random Access Memory |
| SF | Sensor Fusion |
| SMA | SubMiniature version A - coaxial RF connector |
| SoC | System-on-a-Chip |
| SVs | Space Vehicles |
| SW | Software |
| TTF | Time To First Fix |
| UAV | Unmanned Aerial Vehicle |
| UBX | u-blox proprietary binary protocol |
| UPS | Uninterruptible Power Supply |
| V2V | Vehicle to Vehicle communications |
| V2X | Vehicle to Everything communications |
| XDA | Xsens Device API |

Appendix A. Extended Kalman Filter (NED Implementation)

In this appendix, the expressions for the matrices used in the Kalman filter prediction and correction steps when attitude and velocity are Earth-referenced and resolved in the local (NED) navigation frame, and the position is expressed using geodetic coordinates, are reported. The index denoting the epoch is omitted in the equations for readability.

However, it should be understood that all equations are formulated explicitly for a specific epoch k .

State prediction matrix:

$$\Phi = \begin{bmatrix} \mathbf{I} + \mathbf{F}_{11}\tau & \mathbf{F}_{12}\tau & \mathbf{F}_{13}\tau & 0 & \hat{\mathbf{C}}_b^n \tau \\ \mathbf{F}_{21}\tau & \mathbf{I} + \mathbf{F}_{22}\tau & \mathbf{F}_{23}\tau & \hat{\mathbf{C}}_b^n \tau & 0 \\ 0 & \mathbf{F}_{32}\tau & \mathbf{I} + \mathbf{F}_{33}\tau & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \quad (\text{A1})$$

where

$$\mathbf{F}_{11} = -[\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n]_x, \quad (\text{A2})$$

$$\mathbf{F}_{12} = \begin{bmatrix} 0 & \frac{-1}{R_E(L_b)+h_b} & 0 \\ \frac{1}{R_N(L_b)+h_b} & 0 & 0 \\ 0 & \frac{\tan(L_b)}{R_E(L_b)+h_b} & 0 \end{bmatrix}, \quad (\text{A3})$$

$$\mathbf{F}_{13} = \begin{bmatrix} \omega_{ie} \sin(L_b) & 0 & \frac{v_{eb,E}^n}{(R_E(L_b)+h_b)^2} \\ 0 & 0 & \frac{-v_{eb,N}^n}{(R_N(L_b)+h_b)^2} \\ \omega_{ie} \cos(L_b) + \frac{v_{eb,E}^n}{(R_E(L_b)+h_b) \cos^2(L_b)} & 0 & \frac{-v_{eb,E}^n \tan(L_b)}{(R_E(L_b)+h_b)^2} \end{bmatrix}, \quad (\text{A4})$$

$$\mathbf{F}_{21} = -[\hat{\mathbf{C}}_b^n \hat{\mathbf{f}}_{ib}^b]_x, \quad (\text{A5})$$

$$\mathbf{F}_{22} = \begin{bmatrix} \frac{v_{eb,D}^n}{R_N(L_b)+h_b} & 0 & \frac{v_{eb,N}^n}{R_N(L_b)+h_b} \\ \frac{v_{eb,E}^n \tan(L_b)}{R_E(L_b)+h_b} + 2\omega_{ie} \sin(L_b) & \frac{v_{eb,N} \tan(L_b) + v_{eb,D}^n}{R_E(L_b)+h_b} & \frac{v_{eb,E}^n}{R_E(L_b)+h_b} + 2\omega_{ie} \cos(L_b) \\ -\frac{2v_{eb,N}^n}{R_N(L_b)+h_b} & -\frac{2v_{eb,E}^n}{R_E(L_b)+h_b} - 2\omega_{ie} \cos(L_b) & 0 \end{bmatrix}, \quad (\text{A6})$$

$$\mathbf{F}_{23} = \begin{bmatrix} \left(\frac{(v_{eb,E}^n)^2 \sec^2(L_b)}{R_E(L_b)+h_b} - 2v_{eb,E}^n \omega_{ie} \cos(L_b) \right) & 0 & \frac{(v_{eb,E}^n)^2 \tan(L_b)}{(R_E(L_b)+h_b)^2} - \frac{v_{eb,N}^n v_{eb,D}^n}{(R_N(L_b)+h_b)^2} \\ \left(\frac{v_{eb,N}^n v_{eb,E}^n \sec^2(L_b)}{R_E(L_b)+h_b} + 2v_{eb,N}^n \omega_{ie} \cos(L_b) \right) & 0 & \frac{v_{eb,N}^n v_{eb,E}^n \tan(L_b) + v_{eb,E}^n v_{eb,D}^n}{(R_E(L_b)+h_b)^2} \\ -2v_{eb,D}^n \omega_{ie} \sin(L_b) & 0 & 0 \\ 2v_{eb,E}^n \omega_{ie} \sin(L_b) & 0 & \left(\frac{(v_{eb,E}^n)^2}{(R_E(L_b)+h_b)^2} + \frac{(v_{eb,N}^n)^2}{(R_N(L_b)+h_b)^2} \right) \\ & & -\frac{2\mathbf{g}_0(L_b)}{r_{eS}^e(L_b)} \end{bmatrix}, \quad (\text{A7})$$

$$\mathbf{F}_{32} = \begin{bmatrix} \frac{1}{R_N(L_b)+h_b} & 0 & 0 \\ 0 & \frac{1}{(R_E(L_b)+h_b) \cos(L_b)} & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad (\text{A8})$$

$$\mathbf{F}_{33} = \begin{bmatrix} 0 & 0 & -\frac{v_{eb,N}^n}{(R_N(L_b)+h_b)^2} \\ \frac{v_{eb,E}^n \sin(L_b)}{(R_E(L_b)+h_b)^2 \cos^2(L_b)} & 0 & \frac{v_{eb,E}^n}{(R_N(L_b)+h_b)^2 \cos(L_b)} \\ 0 & 0 & 0 \end{bmatrix}. \quad (\text{A9})$$

Prediction error covariance matrix:

$$\mathbf{Q} \approx \begin{bmatrix} S_{rg}\mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & S_{ra}\mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & S_{bad}\mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & S_{bgd}\mathbf{I} \end{bmatrix} \tau, \tag{A10}$$

where S_{rg} , S_{ra} , S_{bad} , and S_{bgd} denote, respectively, the power spectral densities of the noise of the gyroscope and accelerometer, and the power spectral densities of the bias variations of the gyroscope and accelerometer.

Vector of observations:

$$\delta \mathbf{z}_k = \left(\hat{\mathbf{v}}_{GNSS}^n - \hat{\mathbf{v}}_{eb}^n - \hat{\mathbf{C}}_b^n (\hat{\boldsymbol{\omega}}_{ib}^b \times \mathbf{1}_{ba}^b) + [\boldsymbol{\omega}_{ie}^n]_x \hat{\mathbf{C}}_b^n \mathbf{1}_{ba}^b \right), \tag{A11}$$

is given by the difference between the position and velocity solution of the GNSS and the corrected inertial navigation solution, plus a term to account for the displacement between the inertial platform and the GNSS antenna, $\mathbf{1}_{ba}^b$.

Measurement matrix:

$$\mathbf{H}^e = \begin{pmatrix} [\hat{\mathbf{C}}_b^n (\hat{\boldsymbol{\omega}}_{ib}^b \times \mathbf{1}_{ba}^b)]_x & 0 & -\mathbf{I} & 0 & 0 \\ [\hat{\mathbf{C}}_b^n (\hat{\boldsymbol{\omega}}_{ib}^b \times \mathbf{1}_{ba}^b)]_x & -\mathbf{I} & 0 & 0 & \hat{\mathbf{C}}_b^n [\mathbf{1}_{ba}^b]_x \end{pmatrix}. \tag{A12}$$

References

1. United Nations: Department of Economic and Social Affairs Population Dynamics. World Urbanisation Prospects 2018—Processed by Our World in Data. “Urban”. Available online: <https://population.un.org/wup/Download/> (accessed on 12 September 2023).
2. Faria, R.; Brito, L.; Baras, K.; Silva, J. Smart mobility: A survey. In Proceedings of the 2017 International Conference on Internet of Things for the Global Community (IoTGC), Funchal, Portugal, 10–13 July 2017; pp. 1–8. [CrossRef]
3. Paiva, S.; Ahad, M.; Tripathi, G.; Feroz, N.; Casalino, G. Enabling technologies for urban smart mobility: Recent trends, opportunities and challenges. *Sensors* **2021**, *21*, 2143. [CrossRef] [PubMed]
4. Ribeiro, P.; Dias, G.; Pereira, P. Transport systems and mobility for smart cities. *Appl. Syst. Innov.* **2021**, *4*, 61. [CrossRef]
5. Xiang, C.; Feng, C.; Xie, X.; Shi, B.; Lu, H.; Lv, Y.; Yang, M.; Niu, Z. Multi-Sensor Fusion and Cooperative Perception for Autonomous Driving: A Review. *IEEE Intell. Transp. Syst. Mag.* **2023**, *15*, 36–58. [CrossRef]
6. Yeong, D.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors* **2021**, *21*, 2140. [CrossRef]
7. Mailka, H.; Abouzahir, M.; Ramzi, M. An efficient end-to-end EKF-SLAM architecture based on LiDAR, GNSS, and IMU data sensor fusion for autonomous ground vehicles. *Multimed. Tools Appl.* **2023**, *83*, 56183–56206. [CrossRef]
8. Santa, J.; Bernal-Escobedo, L.; Sanchez-Iborra, R. On-board unit to connect personal mobility vehicles to the IoT. In Proceedings of the 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), Leuven, Belgium, 9–12 August 2020; Volume 175, pp. 173–180, ISSN 18770509. [CrossRef]
9. Ganeshkumar, N.; Kumar, S. OBU (On-Board Unit) Wireless Devices in VANET(s) for Effective Communication—A Review. In *Computational Methods and Data Engineering. Advances in Intelligent Systems and Computing*; Singh, V., Asari, V.K., Kumar, S., Patel, R.B., Eds.; Springer: Singapore, 2021; Volume 1257. [CrossRef]
10. He, L.; Deng, Z.; Huang, J. Navigation and communication platform for on board unit of logistics traffic. In Proceedings of the 2009 Joint Conferences On Pervasive Computing (JCPC), Tamsui, Taiwan, 3–5 December 2009; pp. 305–308. [CrossRef]
11. Schindler, N. On Board Unit for the European Electronic Tolling Service. In Proceedings of the 9th ITS European Congress, Dublin, Ireland, 4–7 June 2013. ERTICO (ITS Europe). Available online: <https://www.gnss-consulting.com/wp-content/uploads/2018/12/2013-Schindler-OBU-for-the-European-Electronic-Tolling-Service-Hybrid-Electronic-Tolling-Solution-9thITS-Dublin.pdf> (accessed on 6 June 2023).
12. Santa, J.; Skarmeta, A.; Ubeda, B. An Embedded Service Platform for the Vehicle Domain. In Proceedings of the 2007 IEEE International Conference on Portable Information Devices, Orlando, FL, USA, 25–29 May 2007; pp. 1–5. [CrossRef]

13. Noureldin, A.; Karamat, T.B.; Georgy, J. Fundamentals of Inertial Navigation. In *Satellite-Based Positioning and Their Integration*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 1, p. 313.
14. Tamazin, M.; Karaim, M.; Noureldin, A. GNSSs, Signals, and Receivers. In *Multifunctional Operation and Application of GPS*; Rustamov, R.B., Hashimov, A.M., Eds.; IntechOpen: Rijeka, Croatia, 2018; Chapter 6, pp. 69–85.
15. Zhu, N.; Marais, J.; Bétaille, D.; Berbineau, M. GNSS Position Integrity in Urban Environments: A Review of Literature. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2762–2778. [[CrossRef](#)]
16. Reid, T.G.R.; Houts, S.E.; Cammarata, R.; Mills, G.; Agarwal, S.; Vora, A.; Pandey, G. Localization Requirements for Autonomous Vehicles. *SAE Intl. J CAV* **2019**, *2*, 173–190. [[CrossRef](#)]
17. Groves, P.D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*; Artech House: Boston, MA, USA, 2013; Chapter 14, pp. 559–589.
18. Overview of the EMERGE Initiative: Connected, Geo-Localized and Cybersecure Vehicles. Available online: http://exemerge.disim.univaq.it/?page_id=342 (accessed on 11 December 2023).
19. Di Sciullo, G.; Zitella, L.; Cinque, E.; Santucci, F.; Pratesi, M.; Valentini, F. Experimental Validation of C-V2X Mode 4 Sidelink PC5 Interface for Vehicular Communications. In Proceedings of the 2022 61st FITCE International Congress Future Telecommunications: Infrastructure and Sustainability (FITCE), Rome, Italy, 29–30 September 2022; pp. 1–6. [[CrossRef](#)]
20. Tiberti, W.; Civino, R.; Gavioli, N.; Pugliese, M.; Santucci, F. A Hybrid-Cryptography Engine for Securing Intra-Vehicle Communications. *Appl. Sci.* **2023**, *13*, 13024. [[CrossRef](#)]
21. u-blox. ZED-F9P High Precision GNSS Module Integration Manual. 30 August 2023. Available online: https://content.u-blox.com/sites/default/files/ZED-F9P_IntegrationManual_UBX-18010802.pdf (accessed on 6 September 2023).
22. u-blox. PointPerfect Product Summary. 2021. Available online: https://content.u-blox.com/sites/default/files/PointPerfect_ProductSummary_UBX-21024758.pdf (accessed on 11 June 2023).
23. u-blox. u-blox Services Product Overview: IoT Location-as-a-Service. 2023. Available online: https://content.u-blox.com/sites/default/files/SER-product_Overview_UBX-21029993.pdf (accessed on 10 November 2023).
24. Movella. MTi-630 AHRS Datasheet. May 2023. Available online: <https://www.xsens.com/hubfs/Downloads/Leaflets/MTi-630.pdf> (accessed on 10 November 2023).
25. Movella. MTi 600-Series Datasheet: IMU, VRU, AHRS and GNSS/INS. June 2023. Available online: <https://www.xsens.com/hubfs/Downloads/Leaflets/MTi%20600-series%20Datasheet.pdf> (accessed on 10 November 2023).
26. MQTT Version 5.0 OASIS Standard. 7 March 2019. Available online: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf> (accessed on 10 November 2023).
27. Sixfab. Sixfab Raspberry Pi 3G-4G/LTE Base HAT Datasheet: V1.0. 2019. Available online: https://sixfab.com/wp-content/uploads/2020/03/Sixfab_3G-4G-LTE_Base_HAT_Datasheet_V1.0.pdf (accessed on 22 May 2023).
28. Sixfab. Sixfab API Documentation. Available online: <https://sixfab.github.io/sixfab-power-python-api/> (accessed on 9 September 2023).
29. Node-RED: Low-Code Programming for Event-Driven. Node-RED Documentation. Available online: <https://nodered.org/docs/> (accessed on 14 September 2023).
30. Raspberry Pi (Trading) Ltd. Raspberry Pi 4 Model B DATASHEET: Release 1. June 2019. Available online: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf> (accessed on 12 February 2023).
31. Drotek. Sirius RTK GNSS Rover (F9P): Compact; High-Precision L1/L2 GNSS Rover Based on U-blox ZED-F9P (GPS/GLONASS/BeiDou/Galileo). March 2020. Available online: https://static1.squarespace.com/static/5619914ce4b07e497b442de9/t/5f29cc9638612318c900f5c1/1596574874260/DrotekDoc_0911A++Sirius+RTK+GNSS+Rover+%28F9P%29.pdf (accessed on 22 June 2023).
32. El-Sheimy, N.; Hou, H.; Niu, X. Analysis and Modeling of Inertial Sensors Using Allan Variance. *IEEE Trans. Instrum. Meas.* **2008**, *57*, 140–149. [[CrossRef](#)]
33. Groves, P.D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*; Artech House: Boston, MA, USA, 2013; Chapter 5, pp. 163–187.
34. Groves, P.D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*; Artech House: Boston, MA, USA, 2013; Chapter 15, pp. 638–641.
35. ETSI TS 122 186 V17.0.0. 5G; Service Requirements for Enhanced V2X Scenarios (3GPP TS 22.186 Version 17.0.0 Release 17). (2022-04). Available online: https://www.etsi.org/deliver/etsi_ts/122100_122199/122186/17.00.00_60/ts_122186v170000p.pdf (accessed on 10 September 2023).
36. Desirena Lopez, H.; Siller, M.; Huerta, I. Internet of vehicles: Cloud and fog computing approaches. In Proceedings of the 2017 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Bari, Italy, 18–20 September 2017; pp. 211–216. [[CrossRef](#)]

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.