Received: 3 April 2019 | Revised: 30 November 2019 | Accepted: 12 December 2019

DOI: 10.1002/eng2.12097

RESEARCH ARTICLE

WILEY

Implementation and performance evaluation of an inertial navigation system/global navigation satellite system real-time kinematic Ntrip navigation system aided by a robot operating system-based emulated odometer for high-accuracy land vehicle navigation in urban environments

Jorge P. M. Galdames[1] | Carlos E. Milhor[1] | Daniel V. Magalhães[1] | Glauco A. P. Caurin[2] | Marcelo Becker[1]

[1]SEM-EESC, University of São Paulo, São Carlos, SP, Brazil

[2]SAA-EESC, University of São Paulo, São Carlos, SP, Brazil

Correspondence
Jorge Pablo Moraga Galdames, SEM-EESC, University of São Paulo, São Carlos, SP, Brazil.
Email: galdames@sc.usp.br

Funding information
Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, DS00011/07-0, PNPD20131735; Fundação de Apoio à Física e à Química

Abstract

Advanced land-vehicle navigation commonly uses integrated systems to counteract global navigation satellite system (GNSS) solution degradation. This occurs mainly in urban environments due to blockage of the satellite signals. This paper presents a loosely coupled inertial navigation system/GNSS navigation system that combines an attitude and heading reference system (AHRS) device with a dual-frequency dual-antenna GNSS heading receiver. The integrated navigation system is aided by a low-cost odometer which replaces external wheel speed sensors usually installed in autonomous vehicles. The proposed odometer extracts the anti-lock braking system-generated pulses of rear wheels from vehicle controller area network messages. Following, it converts them into software-generated signal pulses which are sent to the AHRS device through the serial port. The system platform uses a mobile internet data link to get differential GNSS corrections in real-time from a public Ntrip—networked transport of Radio Technical Commission for Maritime Services via internet protocol—broadcaster in order to allow the GNSS receiver to operate in differential global positioning system/real-time kinematic (RTK) modes. Thus, the integrated navigation system provides centimeter-level positioning accuracy at 100 Hz. Since the positioning accuracy is severely affected by numerous factors, this work proposes a replicated 2^4 full factorial design with the purpose of evaluating the in-field obtained positioning performance under different factors combinations. The experimental design chosen allows to know under which conditions it is feasible to replace the available GNSS velocity by the proposed odometry solution, when they are used as navigation aids, and knowing that the proposed odometry has a low resolution. The analysis of 32-runs factorial design results, using a significance level of .05, demonstrated that the proposed odometry can overcome GNSS/RTK velocity.

KEYWORDS

ABS odometry, GNSS/INS, navigation, RTK Ntrip

1 INTRODUCTION

Navigation systems are used for land, sea, airborne, and space vehicles (SVs).[1] Generally, a global navigation satellite system (GNSS) alone provides satisfactory performance in open fields by tracking signals from one or more satellite constellations with global coverage, such as global positioning system (GPS), Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS), Galileo, and in the future BeiDou-2. In order to compute its location in 3D space, a GNSS receiver must be able to lock onto signals from at least four different satellites.[2] So that, when there is a direct line-of-sight (LOS) signal to four or more satellites, GPS system provides positioning information with a consistent and acceptable accuracy.[3,4] Regardless of the GNSS constellation in use, however, the navigation solution availability is fragmented in urban environments due to satellite blockages by buildings and other obstacles.[5] In addition to shadowing, GNSS receivers are also vulnerable to poor satellite constellation geometry, multipath propagation, and ionosphere variability, among other factors. For this reason, in general, a land-vehicle navigation system cannot continuously position a vehicle using a GNSS receiver alone, and other navigation aids are necessary to obtain the desired accuracy, integrity, availability, and continuity of service.[2,6]

In order to counteract GNSS navigation solution degradation, advanced land-vehicle navigation systems commonly use complementary navigation techniques, relying upon information from dead-reckoning sensors such as accelerometers, gyroscopes, and odometers.[6] Dead-reckoning sensors, however, generally cannot be also used alone to position a vehicle accurately for indefinitely long periods of time because dead-reckoning sensors, by definition, do not measure absolute position.[2] Thus, integrated navigation systems are used to take advantage of the complementary attributes of two or more navigation systems in order to yield a system that provides greater precision than either of the component systems operating in isolation.[7] In other words, integrated navigation systems combine an onboard navigation solution and independent navigation data to update or correct navigation solution.[1]

Potentially significant cost, size, weight advantages, and power consumption offered by micro-electro-mechanical systems (MEMS) devices' increased their use in navigation applications, in particular, the inertial measurement unit (IMU).[7,8] The MEMS-based IMU is an inertial instruments arrangement constructed into an electronic block that is usually composed by three mutually orthogonal accelerometers and three gyroscopes aligned with the accelerometers.[7] Another device that combines an IMU, a magnetic compass, and a processor which solves attitude equations is known as attitude and heading reference system (AHRS).[7,9] Generally, implementations that include at least an IMU together with a navigation processor are referred to as inertial navigation system (INS).[1,9] An INS is a complete 3D dead-reckoning navigation system that determines the position, velocity and attitude of a moving platform through the processing of the accelerations and angular velocity measurements of an IMU.[8,9]

The benefits from integrating a GNSS receiver and an INS are well known.[10] These benefits arise in part from the complementary nature of the errors that appear in GNSS position fixes and in the outputs of dead-reckoning sensors.[2,9] As the INS is not dependent on external information, its internal dead-reckoning sensors smooth out the short-term GNSS errors, consequently, assuring the continuous availability of navigation solution.[2,9,10] On the other hand, GNSS fixes prevent the dead-reckoning sensor drifts over long time periods.[2,9] Adding more sensors to the GNSS receiver, however, it is not merely a question of giving the navigation system higher accuracy and better integrity or providing a more continuous navigation solution; it also increases the update rate of the system and provides additional elements into navigation solution.[6]

A typical INS/GNSS navigation system compares the inertial navigation solution with GNSS output and estimates corrections to the inertial position, velocity, and attitude solution using an integration algorithm usually based on a Kalman filter.[9] The integration architecture of INS/GNSS systems widely varies and is mostly divided into four main classes: loosely coupled, tightly coupled, deep or ultra-tightly coupled, and uncoupled systems.[7,9,11] Because of its simplicity and redundancy, the loosely coupled is the most chosen approach.[7,10] Today, tightly coupled solutions are also commercially available because of their better performance when compared to previous versions based on loosely coupled systems. So that, the different types of coupling architectures can have an impact on navigation accuracy after short-term losses of the GNSS data, however, for long-term loss of GNSS it is the inertial sensors quality that will dominate system accuracy.[7]

In addition to variations in INS/GNSS integration architectures, there are several ways to improve the performance of an INS/GNSS system using additional navigation aids such as wheel-speed sensor (WSS). Kubo et al'[12] for example, proposed a dual-mode navigation system that switches between differential GPS (DGPS)/INS and INS/wheel sensor modes according to DGPS signal availability. Stephen[13] developed a multi-sensor navigation system using a decentralized Kalman filter to perform data fusion by combining GPS, gyroscope, barometer, and anti-lock braking system (ABS) differential odometry. Scherzinger[14] described the preliminary tests of an inertial/GPS real-time kinematic (RTK) navigation system, in two levels of integration: loosely coupled and tightly coupled. Carlson et al[15] in turn investigated how to use differential WSS from the factory installed ABS to estimate heading during periods of GPS unavailability. Years later, Hay[16] presented an automotive GPS/WSS dead-reckoning system for navigation assistance in difficult urban environments using wheel-speed messages transmitted by the ABS controller. Embedded solutions also became available, such as Applanix POS LV device used in positioning system installed on Boss[17] autonomous vehicle which combines GNSS, IMU, and wheel encoder to obtain a position estimation at 100 Hz. The authors reported a planar expected positioning error of 0.10 m using differential corrections from OMNISTAR[18] virtual base station system. Current version of this device model is able to provide a 0.035 m planar accuracy (X,Y position) at 200 Hz using inertially-aided RTK (IARTK) technology, as reported in Applanix.[19]

Focusing on algorithms, Hazlett et al[20] developed a six degrees of freedom (6 DOF) model that integrates the differential WSS with GPS and INS measurements, in two versions, one based on extended Kalman filter (EKF) and another on unscented Kalman filter (UKF). Guan et al[21] also proposed a Kalman filter-based navigation system which uses the INS output as positioning source and corrects deviations using GPS information or ABS dead-reckoning measurements when GPS is unavailable. In vehicle stability control applications, Bevly et al[22] proposed a methodology based on GPS/INS combination to estimate vehicle sideslip angle. Among the non-WSS aided improvements, Chiang et al[23] presented a modified tightly coupled INS/GNSS integration architecture to detect and eliminate abnormal GPS measurements before data fusion, for use in low-cost integrated navigation systems. Among the low-cost solutions, Liu et al[24] presented a low-cost implementation of a tightly coupled single-frequency GNSS precise point positioning (PPP)/INS navigation system. In order to improve land vehicles positioning by using the visual odometry (VO) as a navigation aid, Rahman et al[25] proposed an INS/VO/GPS navigation solution. In order to provide precise and continuous positioning solutions in urban environments, Li et al[26] proposed a tightly-coupled single-frequency multi-GNSS RTK/MEMS-IMU integrated navigation system, with resistance to measurement outliers.

In most cases, the inertial measurements, however, offer little correction to the GNSS positioning solution and are primarily used during periods of GNSS unavailability.[15] This is because the GNSS measurement data dominate navigation estimates in an integrated navigation system when the GNSS data are available.[7] Besides, in loosely coupled approach, the use of velocity measurements improves the observability of attitude errors and sensor biases.[7,9] For these reasons, improvements in velocity source of a loosely coupled INS/GNSS may improve the accuracy of the navigation solution but the best possible accuracy will be determined by the accuracy of the GNSS navigation solution.

GNSS positioning accuracy may be improved using some augmentation method, such as differential GNSS (DGNSS),[9] previously known as DGPS[27] or GPS relative positioning.[3] This method calibrates correlated range errors by comparing pseudo-range measurements with those made by equipment at a presurveyed location know as a reference station. Such that, the closer the user is to the reference station, the more accurate the navigation solution is.[9] DGPS techniques may be categorized in different ways: as absolute or relative differential positioning; as local area, regional area, or wide area; and as code-based or carrier-based.[27] Code-based differential systems can provide decimeter-level position accuracy, whereas state-of-the-art carried-based system can provide millimeter-level performance.[27] According to integer ambiguity resolution, DGPS systems may still provide a fixed or float positioning solution, being the fixed type more accurate. Carrier-based positioning techniques that can operate in real-time over moving baseline are know as RTK.[9] Different DGPS techniques implementations can be embedded into a GNSS receiver, so that, it selects one according to internal settings and the signals availability in order to provide a position, velocity and time (PVT) solution.

In a local area DGNSS system, corrections are transmitted through an arbitrary data link from a single reference station to users, generally using a version of Radio Technical Commission for Maritime Services (RTCM) Study Committee 104

(SC-104) protocol.[28] Data link type varies according to DGNSS system requirements. It can use the marine radio-beacon band, very high frequency (VHF) and ultra high frequency (UHF) bands, cellular frequencies, radio and television broadcasts, Loran signals, or even the Internet to transmit the RTCM messages.[9] The technique that uses the Internet to collect and exchange GNSS data streams in real-time was developed through an initiative of Bundesamt für Kartographie und Geodäsie (BKG), which established the format so called Ntrip (Networked Transport of RTCM via Internet Protocol).[29] This technique allows stationary and mobile users connected to the Internet, simultaneously retrieve DGNSS correction data from a Ntrip broadcaster, which in turn receives DGNSS corrections from multiple Ntrip servers each connected to at least a GNSS receiver of a reference station.

Differential corrections computed from a single reference station are subject to a single point of failure. In addition, the rover distance to the base restricts the validity of the corrections due to spatial decorrelation.[30] However, using data from multiple reference stations improves ambiguity resolution reliability and enhances GNSS positioning accuracy.[31] In addition, the use of a network of reference stations instead of a single reference station allows to model in-region systematic errors.[32] Network RTK (NRTK) techniques use a network of reference stations to provide a resolution in real time. These techniques collect raw measurements from the network of reference stations, solve the ambiguities within the reference network, generate error estimates, and then apply an interpolation scheme to generate the NRTK corrections for the user location.[33] Several NRTK solutions are used to generate NRTK corrections for rover users, such as virtual reference station (VRS),[32,34] multi reference station (MRS),[35-37] master auxiliary concept (MAC),[38] Flächen-korrektur-parameter (FKP),[31] individualised master-auxiliary (i-MAX),[39] and network adjustment (NetAdjust).[40] Takac and Zelzer[41] examine the relationship between some of these techniques aiming to improve estimation of the quality of the network correction information performed by the rover. In this context, Manzino and Dabove[42] presented a real-time quality control of NRTK positioning by using measurements of mass-market receivers.

Since Ntrip became the leading standard protocol for GNSS data streaming, it has become popular in various public and private sectors.[43] Today, there are Ntrip broadcasters that provide free access to DGNSS corrections of several reference stations to registered users. Uradzinski et al[43,44] used UNB3 mountpoint of BKG Ntrip broadcaster to carry out a field experiment with the purpose of evaluating the performance of Ntrip/RTK solution in rover kinematic mode. Souza et al[45] evaluated positioning accuracy obtained by Ntrip/RTK and real-time PPP methods using Ntrip broadcaster of the RBMC[46] (Brazilian Network for Continuous Monitoring of the GNSS-RBMC Systems) managed by IBGE (Brazilian Institute of Geography and Statistics). A first survey of IBGE/RBMC Ntrip service was presented by Costa et al.[47]

Whereas DGNSS/RTK corrections improve INS/GNSS navigation solution, velocity measurements may also improve the accuracy of loosely coupled INS/GNSS systems, as mentioned above. Information on the speed of the different wheels are often available through the ABS WSSs at no additional cost in terms of extra sensors but generally with a low resolution.[6] Natural tire size variation still increases land-vehicles odometry error (see Hay[16] for more details on WSS design challenges). In addition, wheel sensors can become unreliable for another reason such as wheel slipping, skidding, and error in the scale factor.[12] Furthermore, some ABS WSS design parameters may be unknown or inaccessible, such as sensor type (active or passive), sensor teeth number, and ABS controller transmission rate. The impact of ABS odometry accuracy on the INS/GNSS navigation solution, however, is determined by the parameters and methods of integration algorithm. Due to these reasons, the resulting performance improvement on integrated navigation systems by using ABS odometry aid is uncertain. Carlson et al,[15] for example, noted that the positioning error of their navigation system meaningly grows when the vehicle navigates on speed bumps and/or uneven road surfaces. Carlson et al[15] also noted that increasing the resolution of the wheel encoder from 100 PPR (pulses per revolution) to 2000 PPR did not improve the dead-reckoning global positioning accuracy by more than 2 cm.

GNSS/INS navigation systems evolved to devices with embedded integration architectures that make sensor integration easier because they avoid the complex algorithms implementation. These devices require a careful setting procedure, as well as right sensors installation and calibration to obtain best performance. On the other hand, implementations that employ this device type have a specific input of navigation aids that restricts the number and type of external sensors inputs. Furthermore, although an integrated navigation system can work in GNSS-denied environments, problems include the cost of the sensors and the length of time that the GNSS signals are unavailable.[23] In addition, to give absolute figures on the accuracy of the state-of-the-art land vehicle navigation systems is difficult, since not only does the performance of the systems depends on the characteristics of the sensors, GNSS receiver, vehicle model, and map information but on the trajectory dynamics and surrounding environment, as well.[6]

This paper is an important part of an outdoor mapping system presented in the author's doctoral thesis.[48] Since a positioning input is critical to perform robotic mapping tasks, a high-accuracy navigation system was implemented on a land vehicle platform. In robotic mapping applications, when a constant and high-accuracy positioning estimation is required to derive close-features positions, the navigation system output along the trajectory may not be sufficient to achieve the minimum accuracy required. Due to this, previous surveys and performance evaluations of the integrated navigation system are done. As the positioning solution is not only influenced by the equipment used but also by many other factors, using a model to predict the in-field performance of the navigation system is very useful to select the best conditions for collecting data. Thus, the methodology proposed to evaluate integrated navigation systems performance based on design of experiments (DOE) results in a model that allows to predict the performance of custom systems. The procedures presented in this paper can be adapted to obtain the performance model of other integrated navigation systems. This possibility of model replication represents the main contribution of this work.

This paper presents a high-accuracy INS/GNSS navigation system developed by combining an AHRS device, a GNSS receiver, and a custom odometer. The navigation system uses the loosely coupled INS/GNSS integration architecture embedded into the AHRS device. The GNSS receiver used is a dual-frequency dual-antenna multi-constellation GNSS heading receiver configured to operate in DGPS/RTK modes using RTCM messages retrieved from IBGE/RBMC Ntrip broadcaster connected via a 4G (LTE) mobile internet. Since the AHRS device used in this work, accepts only pulse signals as odometer input, an emulated odometer based on robot operating system (ROS) was developed in order to use ABS odometry extracted from vehicle controller area network (CAN) messages. The proposed odometry solution is called herein as CAN-odometry. The proposed navigation system was implemented on SENA[49] (Autonomous Embedded Navigation System) project vehicle. An experimental design based on DOE methodology was carried out in order to evaluate integrated navigation system performance under different experimental conditions. Section 2 describes devices physical installation, wired connections, and setup details applied in integration of the devices. Section 3 provides software architecture of the ROS-based odometer, and the ROS device driver developed for the AHRS device. Section 4 explains in detail experimental design, decisions, and procedures utilized to evaluate the positioning performance of the proposed integrated navigation system. Section 5 describes the factorial design analysis steps. It also provides the obtained models from the experimental results, as well as a discussion about models interpretation and validity. Finally, Section 6 provides the conclusions of this work, as well as the proposed future works.

2 | HARDWARE AND SENSORS SETUP

This section provides an overview on hardware and sensors used in this work with focus on their installation and setup. One may see in subsections that most of the configuration work was dedicated to AHRS and GNSS devices. Subsection 2.1 explains the installation and setup decisions applied in the AHRS and GNSS integration, as well as the AHRS integration with the CAN-based odometry system proposed in this work. Subsection 2.1.1 provides a brief description of the embedded EKF filter, and profile settings applied to obtain the best performance. Subsection 2.1.2 explains installation and extrinsic calibration of AHRS device. Subsection 2.1.3 provides a diagram of the circuit used to connect the external devices. Subsection 2.1.4 describes the AHRS settings for the correct interpretation of odometer signal. Subsection 2.1.5 provides the GNSS-related AHRS settings. Subsection 2.2 details the GNSS side settings of the integration, as well as the RTK-Ntrip configuration. Subsection 2.2.1 provides the configuration required to enable the GNSS true heading estimation. Subsection 2.2.2 provides the PVT mode setup. Subsection 2.2.3 provides the NMEA and x-pulse-per-second (xPPS) output setup. Finally, Subsection 2.3 describes main support equipments and CAN bus connection.

2.1 | AHRS settings

The AHRS sensor used in this work is an IG-500E-G4A2P1-S model manufactured by SBG Systems. The AHRS settings can be applied by using the SBG Center tool or via SBG SDK (Software Development Kit) low level commands. The presented settings in this section were configured by using the SBG Center tool and saved to the AHRS flash memory, because they do not need to be dynamically modified. Table 1 shows some other AHRS devices available on the market that allow a similar integration of odometer and GNSS external inputs, and that can be evaluated using the methodology presented in this work.

Model	Manufacturer	More information at
Ellipse 2 Micro INS	SBG Systems	www.sbg-systems.com
Ellipse2-D	SBG Systems	www.sbg-systems.com
Ekinox-E	SBG Systems	www.sbg-systems.com
Apogee-A	SBG Systems	www.sbg-systems.com
Applanix OEM products	Applanix	www.applanix.com
MTi-300	XSens	www.xsens.com
Motus	Advanced Navigation	www.advancednavigation.com
μINS[a]	Inertial Sense	www.inertialsense.com
VN-300[b]	VectorNav	www.vectornav.com

[a] Does not mention external GNNS input support.

[b] Does not mention odometer input support.

TABLE 1 AHRS devices currently available on the market that allow a similar integration

2.1.1 | EKF setup

The EKF is embedded in AHRS CPU and provides a real time 17 EKF states. According to the user selected settings, it can deliver precise orientation, position, and velocity, as well as sensors information to a host connected to the device. This information is obtained by data fusion between inertial sensors, magnetometers, temperature sensors, GNSS receiver, odometer, and sensors calibration data.[50,p.09] Figure 1 shows the simplified block diagram of the embedded EKF. In order to obtain the best performance from the embedded filter, the automotive motion profile was selected in the filter settings. It is the motion profile which fits better to experimental platform. The motion profile provides embedded EKF the adjustments and compensations to obtain the best performance of the filter. The magnetic declination estimated value of filter settings was manually inserted during preparation of each experimental run. The values were calculated using the procedure detailed in Subsection 4.2.1. The remote heading was selected as heading source in filter settings in accordance with SBG Systems support recommendation, thus enabling the EKF solution to use GNSS true heading in the Euler solution computation. The advanced options settings allows the user to enable some filter features; they were maintained at default values. The filter automatically starts and runs according to device settings described in next subsections. The device manufacturer documentation does not provide implementation details about the estimation algorithms used by the filter.

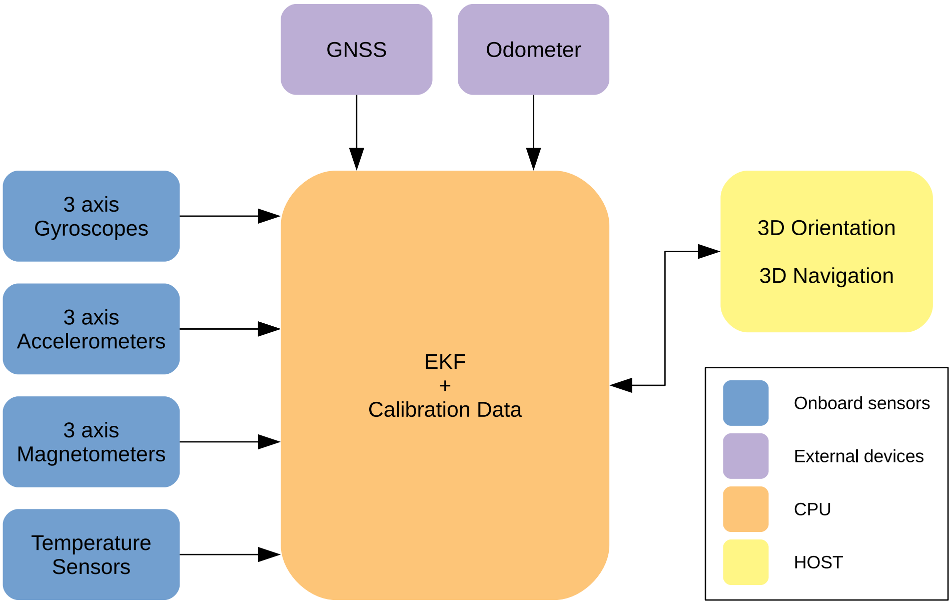


FIGURE 1 Simplified block diagram of the embedded EKF, adapted from SBG Systems.[50,p.09] EKF, extended Kalman filter

2.1.2 | AHRS installation and extrinsic calibration

The AHRS placement was chosen in order to attend SBG Systems placement requirements, as well as to ease its extrinsic calibration with other devices. According to SBG Systems,[50,p.15] sensor coordinate frame must be aligned within less than 1° to the vehicle coordinate frame, and *X*-axis must be turned toward vehicle direction of travel. Thus, the sensor mounting set development was guided to reach these SBG Systems optimal installation guidelines. Figure 2 shows the AHRS installation in vehicle roof rack using the designed mounting set.

In order to obtain highest precision position measurements, the origin of AHRS's coordinate system was based on real 3D accelerometer location shown in Figure 3 extracted from SBG Systems[50,p.46] with the SBG Systems authorization. The extrinsic calibration addressed here refers to the external aiding devices distances with respect to this AHRS frame. Table 2 shows the relative positions of the two connected external aiding devices in AHRS's coordinate system. The relative distances were obtained through manual measuring using a measuring tape with 1 mm resolution.

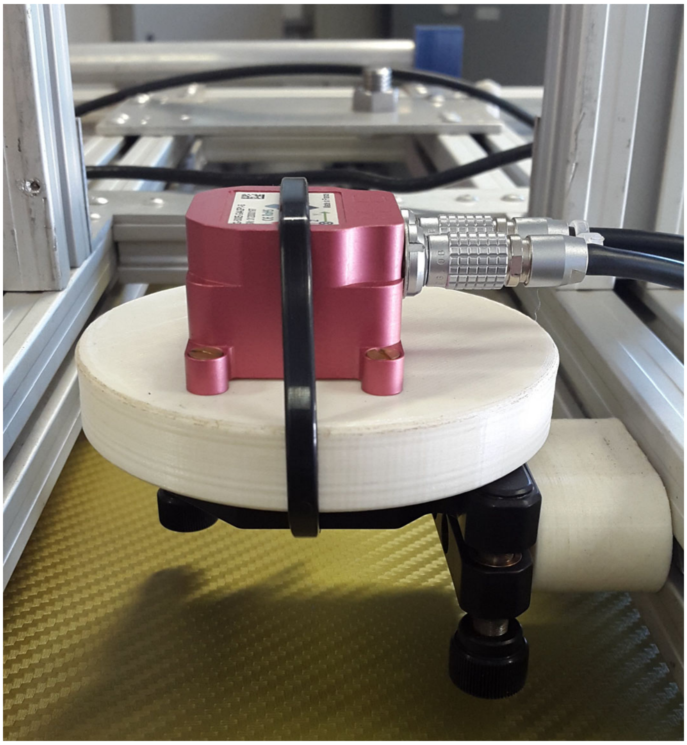


FIGURE 2 AHRS installation using the designed mounting set. AHRS, attitude and heading reference system

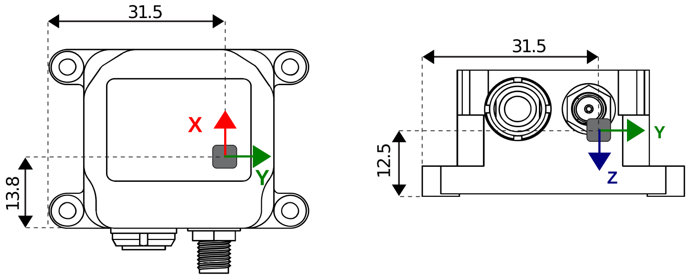


FIGURE 3 3D accelerometer physical location. All dimensions are in millimeters (mm)

TABLE 2 Devices relative positions in AHRS's coordinate system

	Coordinates in meters		
External aiding device	X_{AHRS}	Y_{AHRS}	Z_{AHRS}
Virtual odometer	−1.065	0	+1.287
GNSS main antenna	−0.577	0	+0.005

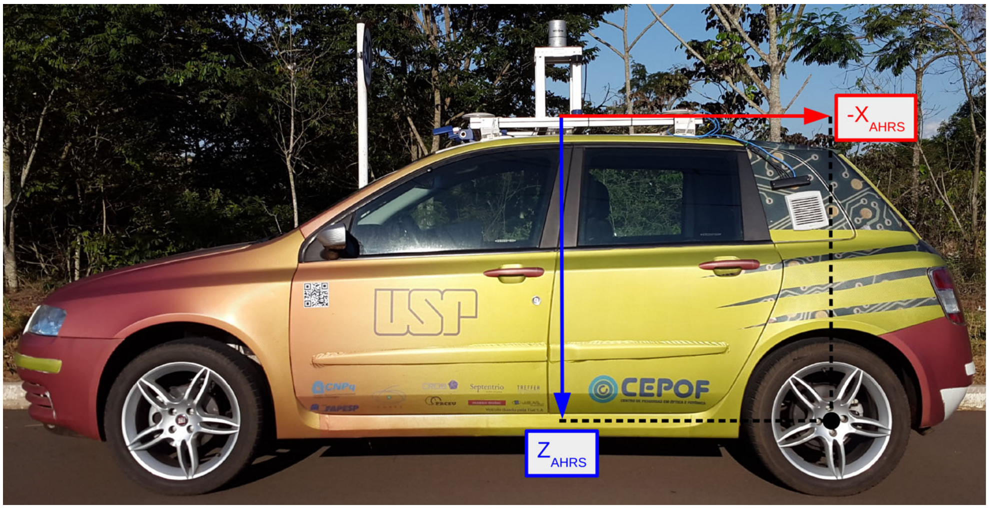


FIGURE 4 Odometer's relative distances to AHRS in *X* and *Z* directions

In AHRS settings, Table 2 rows are denominated as odometer lever arm and GPS lever arm, respectively. The lever arm between the AHRS device and the odometer needs to be properly entered for best performance.[50,p.35] The manufacturer available documentation does not inform the recommended accuracy for odometer lever arm, but it considers the alignment between AHRS device and odometer as crucial for dead reckoning performance.[50,p.35] It also recommends an accuracy within 10 cm for the distance between GNSS main antenna and AHRS device.[50,p.35]

As one may observe in Table 2, both distances in *Y* coordinate are zero. This is due to carefully chosen location to mount the AHRS and the GNSS antennas, as well as the choice to adopt a centralized virtual odometer. Following this procedure, the three devices frames keep aligned in longitudinal car symmetry line. Figure 4 shows relative distances between odometer and AHRS in *X* and *Z* directions of the AHRS's coordinates system. Figure 6 provides a better view of AHRS's relative position to GNSS antennas, as well as the AHRS's place in the vehicle roof rack.

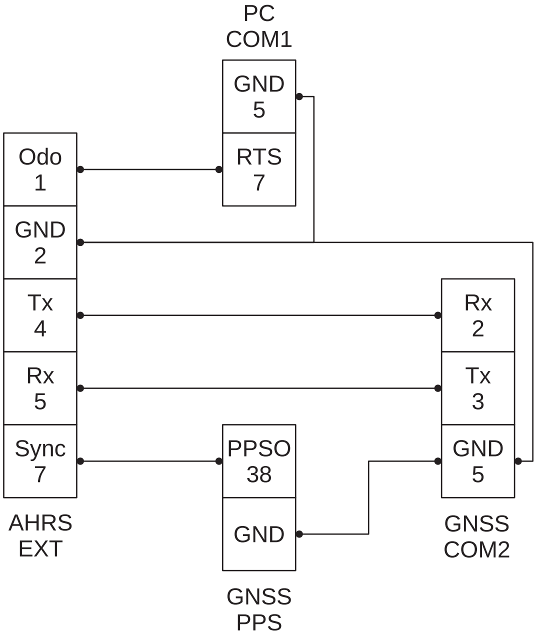
2.1.3 | AHRS external connections

Once that the pulse signal generation was tested and validated by an oscilloscope (see Subsection 3.2.3), a circuit was designed and built to connect GND (ground) pins, and RTS (request to send) pin of PC COM1 to odometer input (Odo) pin of AHRS extended connector (AHRS EXT). The connection between the AHRS and the GNSS, previously tested, was also included in the design. The GNSS connection is composed by two wired connections: a serial bus communication between GNSS COM2 and AHRS EXT; and a PPS (pulse-per-second) time synchronization. The serial bus connection uses two pairs of serial pins: transmitter (Tx) and receiver (Rx). The PPS time synchronization is carried out by using PPS output (PPSO) pin of GNSS connected to synchronization input (Sync) pin of AHRS EXT. Figure 5 shows the final version of the circuit diagram of the AHRS external connections.

2.1.4 | Odometer integration

This subsection provides necessary settings so that the AHRS translates rightly the signal received in the input channel to the velocity value sent by the pulse signal generator. The AHRS has two input channels with multiple purposes; in this work the odometer was assigned to input channel 1. In the input channel 1 settings, sensitivity is the option that determines how the AHRS triggers the count of one pulse with three possible choices: rising edge, falling edge, and level change, all of them working and resulting in an odometer velocity value. The loop block of pulse signal generator produces a square wave signal with one rising edge, one falling edge and two level changes. After several tests, the final choice for sensitivity was level change. It provides an 1:2 proportion in pulse signal transmission, so that one pulse sent by pulse signal generator is translated in two pulses in the AHRS side. This is a way to mitigate the imprecision of higher speeds by reducing to half the frequency that the pulse signal generator must emulate. Due to the software generation feature of the pulse signal, the processing time of RTS pin activation and deactivation instructions is not constant. Consequently, the generated pulse width may suffer little variations that are perceived as noise by the AHRS odometer velocity value. At

FIGURE 5 Pins connections of AHRS external connections (the numbers correspond to connectors pin numbering)



higher frequencies, these variations represent a larger fraction in pulse width than at lower frequencies, therefore higher speeds are subject to more noise.

The third setting to translate the odometer velocity correctly is the gain. In AHRS settings, the gain refers to the number of pulses computed by the AHRS to complete the distance of 1 m, namely, the number of pulses per meter. Due to fact that this value was not found in vehicle parts datasheet, it was estimated based on two parameters: wheel circumference (C_{wheel}) and encoder ring resolution ($\hat{R}_{\text{wheel}}$). The relation between the two parameters and gain is presented in Equation (1). The encoder ring resolution was estimated by using a developed ROS node (see Subsection 3.2.4). Unlike navigation systems of Carlson et al[15] and Guan et al[21] which used GPS to estimate the individual wheel radius, this work uses a value of wheel circumference extracted from a factory setting published by a CAN message (see Subsection 3.2.5).

$$\text{gain} = \frac{\hat{R}_{\text{wheel}}}{C_{\text{wheel}}} = \frac{81\ \text{PPR}}{1620\ \text{mm}} = 50\ \text{pulses/m}. \tag{1}$$

Once the pulse signal generator is started, the frequency of pulses is updated through ROS messages containing the new pulse rate. A ROS tool node (see Subsection 3.3.1) was employed to publish ROS messages with constant values in order to verify the odometer velocity deviations read from the AHRS. This test was repeated for several values; Table 3 shows the constant values that have been tested and their respective deviation range. This validation procedure was applied to evaluate the performance of the odometer and to verify if high velocities lead C code of pulse signal generator to crash. These tests were carried out by using this procedure, because it is very difficult to produce a constant velocity by manually rotating the wheel or by driving the car. The best performance, presented in Table 3, was accomplished just through connection of the GNSS PPS signal output to the AHRS input channel 0.

2.1.5 | GNSS integration

In order to integrate the GNSS NMEA output (see Subsection 2.2.3) to the AHRS, the external aiding of AHRS settings was set up as follows: the device type as NMEA, the baud rate according to GNSS output port which streams NMEA messages in 115200, the serial mode as RS-232, and slew rate as fast slew. The NMEA options of the external aiding were configured according to NMEA sentence sequence output by the GNSS. Since the HDT frame is sent after the RMC frame, the NMEA options were configured in accordance with this pattern. In order to accomplish the AHRS main loop synchronization with the received PPS signal (see Subsection 2.2.3), the input channel 0 settings of synchronization tab were set up as follows: type was set up to time pulse, sensitivity as rising edge, location at external connector, and delay to 0.

TABLE 3 Validation of odometer velocity read by the AHRS

		Velocity (m/s)	
Pulse variation (ΔP)	Frequency (Hz)	Input	AHRS
10	50	2	2 + [−0.0023; +0.0034]
20	100	4	4 + [−0.0050; +0.0074]
50	250	10	10 + [−0.0118; +0.0181]
100	500	20	20 + [−0.0230; +0.0376]
200	1000	40	40 + [−0.3693; +0.0480]
400	2000	80	80 + [−2.0370; +0.1072]

Abbreviation: AHRS, attitude and heading reference system.

2.2 | GNSS settings

The GNSS model used in this work is the AsteRx2eH OEM equipped with two PolaNt*_ GG antennas, all from the Septentrio Satellite Navigation manufacturer. It has six external communication ports; these are: four COM ports, one USB port, and one Ethernet port. The COM2 port was selected to send NMEA messages to the AHRS (see Subsection 2.2.3), and the COM3 port to receive the differential corrections from the Ntrip server (see Subsection 2.2.4). The COM1 and COM4 ports were reserved for other purposes not addressed in this work. The USB port was used exclusively to connect the RxControl,[51] that was used in the setup procedures, and in the GNSS monitoring during the preliminary tests and the experimental runs. The Ethernet port was also used exclusively to connect the RxLogger[51] in order to avoid possible connection overloads by getting log data from a port being used by another connection.

2.2.1 | Multi-antenna attitude

The multi-antenna attitude setup was carried out following default antenna configuration. It consists in placing the antennas aligned with the longitudinal axis of the vehicle, where the aux1 antenna should be in front of the main antenna.[52,p.26] The adoption of default antennas configuration eliminates the setup procedure required to remove the resulting GNSS attitude biases, such as nondefault antennas relative position, heading angle offset, and pitch angle offset. Figure 6 shows the antennas alignment in vehicle rear top view angle. The default antennas layout increased the distance between the antennas from 0.80 m to 1.00 m approximately. This layout improved the GNSS attitude accuracy, as well as it increased the GNSS heading availability. A new antenna support was designed to install the antennas in the default layout, and to reuse the current rack installation on the vehicle roof. Figure 7 shows the fixation of the GNSS main antenna by using the designed support. Some settings are required to activate the GNSS attitude computation in multi-antenna mode. In the

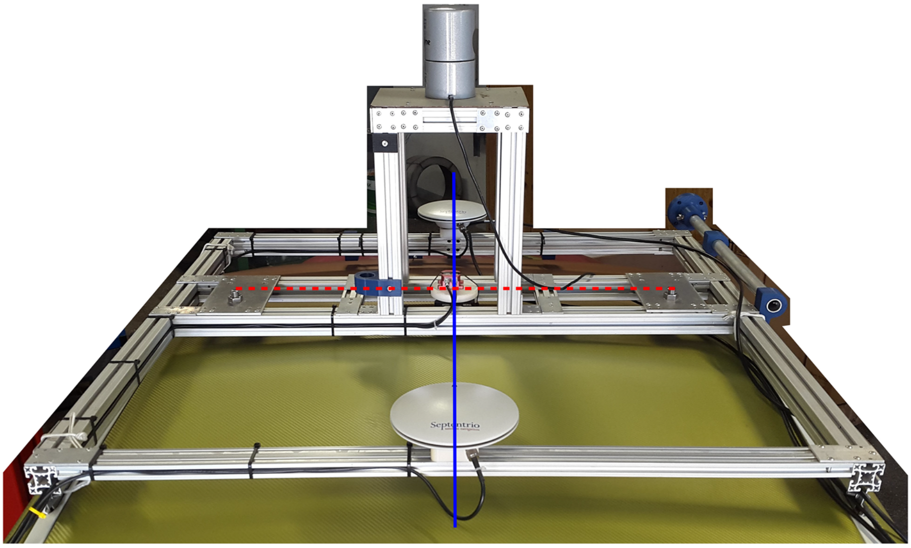
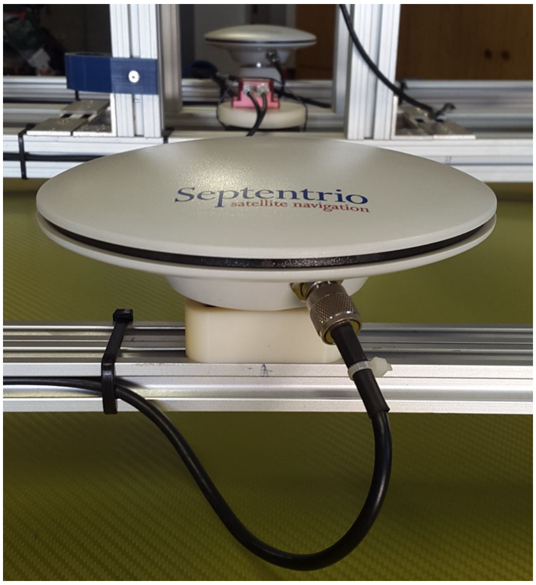


FIGURE 6 GNSS antennas placed in default antenna configuration (solid blue line) and previous installation places (dashed red line). GNSS, global navigation satellite system

FIGURE 7 GNSS main antenna fixation. GNSS, global navigation satellite system



Navigation>Positioning Mode>GNSS Attitude menu option of RxControl,[51] the source of GNSS attitude was selected to MultiAntenna, and the float and fixed multi-antenna modes were enabled. In the same menu option, the antenna location modes were selected to Auto in order to have the GNSS compute automatically the relative distance between the antennas.

2.2.2 | PVT mode setup

The GNSS model installed in the experimental platform allows the use of satellites from GPS, GLONASS, Galileo, and SBAS constellations in the PVT computation. In this work, the allowed PVT modes along with RTCM input (see Subsection 2.2.4) were configured in order to allow the GNSS to operate in GPS/GLONASS RTK fixed mode. The Rover mode was selected in Navigation>Positioning Mode>PVT Mode settings in order to inform the receiver that the receiver is moving, otherwise the receiver will assume that it is fixed when computing the PVT solution. Besides RTK modes, the PVT modes DGPS and Standalone were enabled in Rover mode options. When more than one PVT mode is enabled in Rover mode, the receiver automatically selects the PVT mode that provides the most accurate solution with the available data.[53,p.60] The reason for allowing other less accurate PVT modes was to keep sending NMEA messages by the receiver, as well as the time synchronization (see Subsection 2.2.3), under constraints of satellite signals and/or of differential corrections during the experimental runs. The satellite-based augmentation system (SBAS) mode was not enabled due to the fact that the experiment region is out of operational SBAS systems coverage. The Galileo and the SBAS constellations were disabled in Navigation>Advanced User Settings>PVT>Satellite Usage settings in order to keep a smoother transition under Standalone PVT mode. The Galileo and the SBAS constellations also were disabled in Navigation>Advanced User Settings>Tracking>Satellite Tracking settings in order to reduce the load of the automatic channel allocation mechanism. As the planned experiments included regions with different occlusions grades, it was found very difficult to choose an optimized elevation mask for all experimental runs. Therefore, the elevation mask was kept at default value of 10° for the two defined masks: tracking mask and PVT mask.

2.2.3 | NMEA and xPPS output setup

Two output streams were configured in order to make the GNSS to stream the NMEA messages required by the AHRS. Table 4 shows the NMEA sentences and the frequency of each stream, which were included in NMEA Output settings. Additionally, inside Input/Output Selection settings, the NMEA data type was selected at Output COM2 in order to enable the streaming. The NMEA messages requirements were obtained from SBG Systems.[54,p.5]

The xPPS output parameters were set up in Navigation>Receiver Operation>Timing. The time interval argument was set equal to period of ZDA message output, 1 s, in order to attend requirement of SBG Systems.[54,p.5] The signal polarity

TABLE 4 NMEA output streams

	Outputs	
Field	**Stream 1**	**Stream 2**
Port	COM2	COM2
NMEA sentences	GGA, HDT, RMC	ZDA
Frequency	10 Hz	1 Hz

TABLE 5 RTCM messages types sent by the selected reference stations

Reference stations									
Mountpoint	**Identifier**	**RTCM message types**							
EESC0	Sao Carlos	1004	1006	1008	1012	1013	1019	1020	1033
SPJA1	Jaboticabal	1004	1006	1008	1012	1013	1019	1020	1033

was set to Low2High, that is compatible with AHRS input channel sensitivity set as rising edge (see Subsection 2.1.5). The delay parameter was maintained at the zero nsec default value, because no method was established in this work to measure the delay coming from the tailor-made cable connecting AHRS input channel and PPSO (PPS output) pin of the GNSS. The universal time coordinated (UTC) was selected as time scale reference in accordance with requirement in SBG Systems.[54,p.5] The Max Sync Age was set up to 60 s. Using the UTC time as reference has the accuracy of the position of the xPPS pulse dependent on GNSS PVT computation. The xPPS position is extrapolated by the GNSS receiver during PVT outages. Thence, the xPPS pulse output will stop when the age of the last PVT is older than the age specified in the Max Sync Age argument.

2.2.4 | Ntrip broadcaster connection

Initially, the connection between the GNSS and the Ntrip broadcaster was established by using the BKG Ntrip Client (BNC) version 2.12.3, available for download on the BKG GNSS Data Center at BKG.[55] This software handles the HTTP communication and allows to transfer received real-time GNSS data streams from a Ntrip broadcaster to a serial port.[55] More details about the BNC development and resources can be found in Weber and Mervart.[56] The International GNSS Service (IGS)[57] provides the access to several Ntrip broadcasters in different regions worldwide through affiliated real-time service (RTS) station operators. In this work, we successfully applied to obtain the access to the IGS-IP broadcasters for research purposes. In addition, we required the registration as user in the RBMC-IP service provided by IBGE,[58] in order to get free real-time access to all data streams of stations of the RBMC. Each stream on Ntrip broadcaster is defined by using a unique source ID, also called mountpoint.[56] The source-table maintained by Ntrip broadcaster provides information about mountpoints available on the host. The Ntrip broadcaster host IP of RBMC-IP service is 186.228.51.52 and its port number is 2101. Only one mountpoint is required to have the GNSS operate in DGPS or RTK PVT modes. The Data Link software tool from Septentrio NV/SA[51] allows the configuration of a Ntrip connection, and the forwarding to a serial port in a similar way to the BNC configuration. Due to its easiness to change the connected mountpoint, it was used during the planned experimental runs.

In order to allow the incoming of RTCM messages in the COM3 port of the GNSS, the RTCMv3 was selected in the COM3 drop list of Input Communication>Input/Output Selection settings. Both reference stations used in this work publish differential corrections of GPS and GLONASS systems in RTCM 3.0 format. Table 5 shows the RTCM message types send by EESC0 and SPJA1 mountpoints. These message types should be enabled in the RTCMv3 of Communication>Input Settings>Differential Corrections settings, so that the GNSS receiver uses the received messages to compute PVT solution.

2.3 | Support equipments

The operation of the integrated navigation system presented in this work is supported by another equipment installed in the vehicle. Figures 8 and 9 show these equipments installation on the vehicle trunk. One may see in Figure 8 a

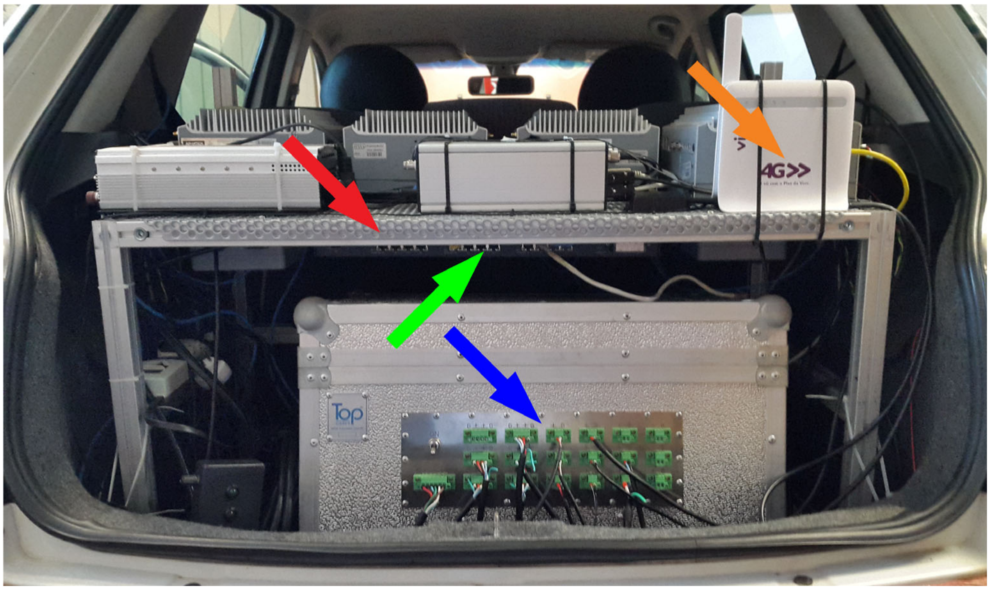


FIGURE 8 Installation of the support equipments on the vehicle trunk (front view)

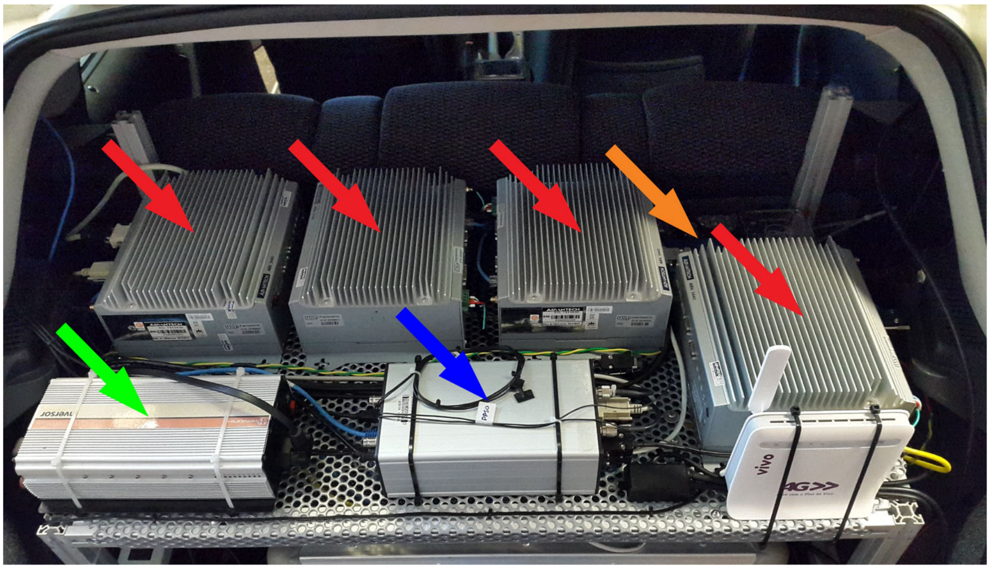


FIGURE 9 Installation of the support equipments on the vehicle trunk (top view)

metal shelf (red arrow), an industrial Ethernet switch (green arrow), a 12 V rechargeable battery case (blue arrow), and a 4G modem (orange arrow). Figure 9 shows the top view of the installation, one may see in it four industrial fanless computers (red arrows), an inverter (green arrow), a box of the GNSS receiver (blue arrow), and the location of the plastic box of the circuit represented in Figure 5 (orange arrow), among others. A top view of this plastic box is shown in Figure 10 (red arrow).

Figures 11 and 12 show CAN bus connection details. One may see in Figure 11 the pin connections that were used to make the cable that allows the communication between vehicle CAN bus network and CAN bus board of industrial computer. The specification of the on-board diagnostic (OBD) connector used in the cable end attached to vehicle CAN bus input is OBD-II J1962 type A. Figure 12 shows the cable connected to vehicle CAN bus input (red arrow), located near the vehicle instrument panel.

3 | SOFTWARE ARCHITECTURE

The software architecture of the system presented in this work was implemented over ROS framework in order to allow the integration of the system outputs with other experimental platform resources. A ROS package containing custom

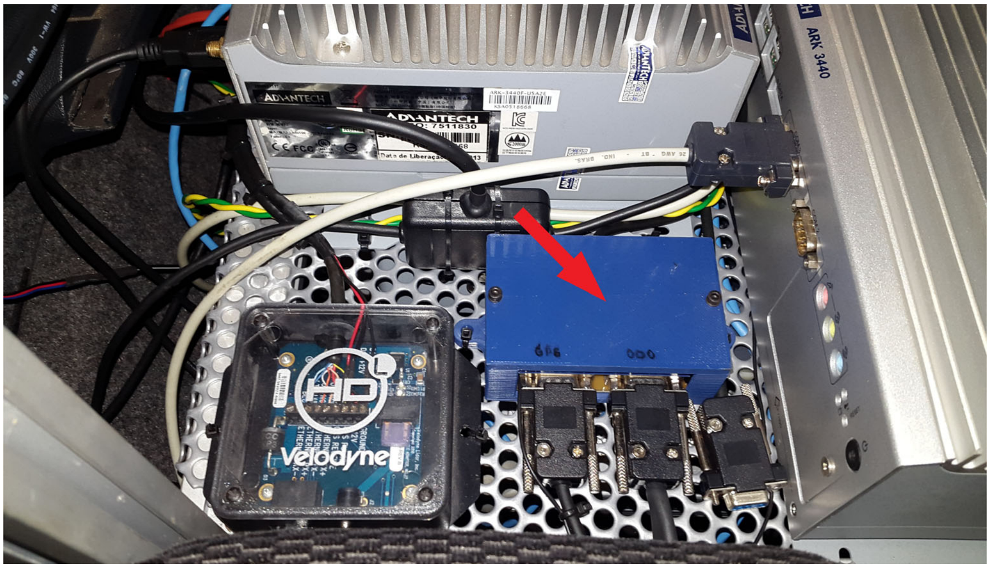


FIGURE 10 Installation top view of the integration circuit of the AHRS with the external connections. AHRS, attitude and heading reference system

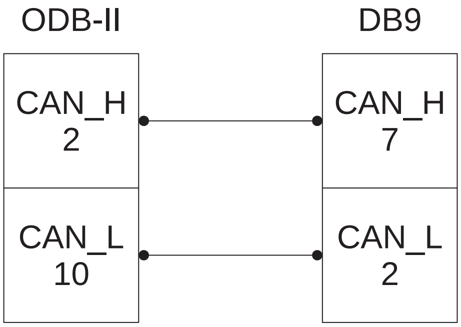


FIGURE 11 Pin connections of the CAN bus cable. CAN, controller area network

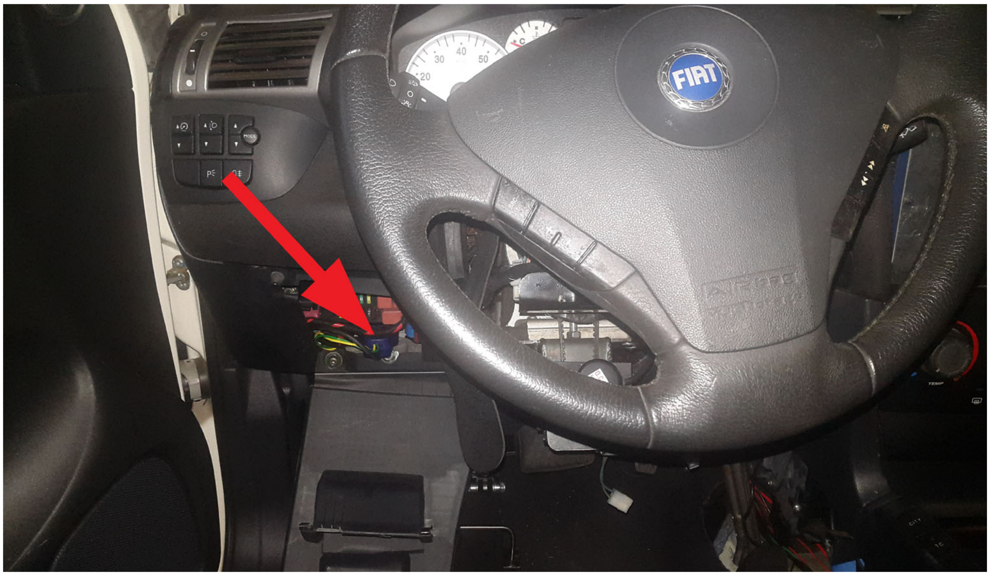


FIGURE 12 Vehicle CAN bus input. CAN, controller area network

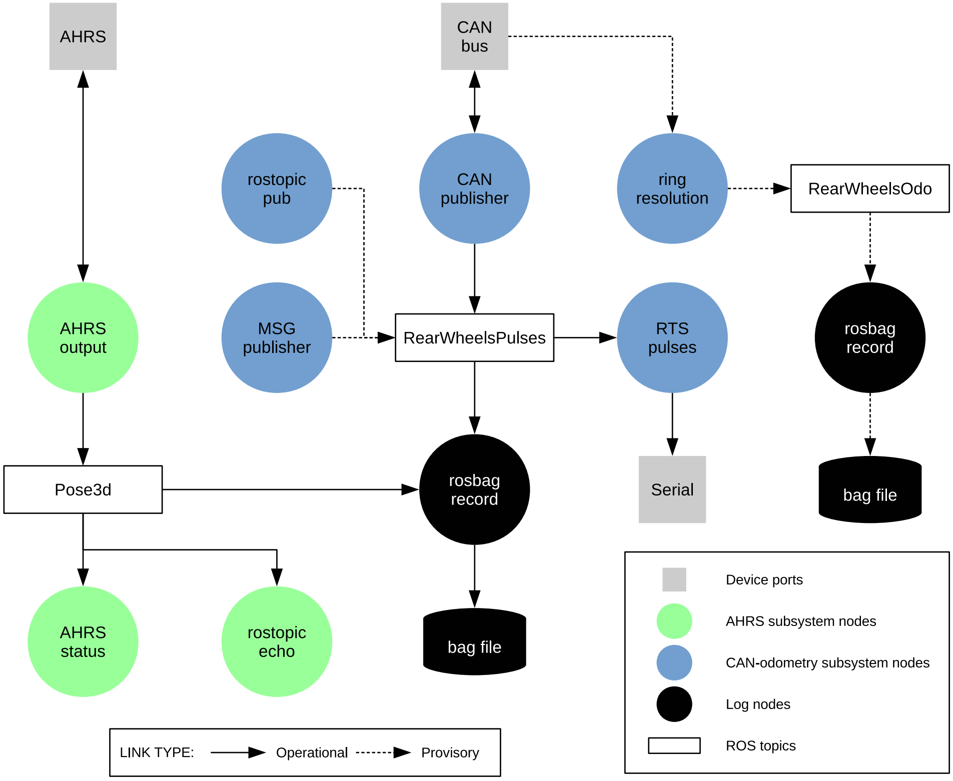


FIGURE 13 Software architecture components

messages definition was developed to support the software architecture operation under nodes and topics concepts of ROS framework, see Romero.[59] This package is described in Subsection 3.1.

One may observe in Figure 13 an overview of our system ROS elements. It shows the used device ports (represented by gray elements), AHRS subsystem nodes (green elements), CAN subsystem nodes (blue elements), log nodes and bag files (in black), ROS topics (white rectangles), and input/output links (solid and dashed arrows).

The description of CAN-odometry subsystem nodes is presented in Subsection 3.2. The CAN publisher node (Subsection 3.2.1) extracts the rear wheels pulse counter from vehicle CAN bus messages. The RTS pulses node (Subsection 3.2.2) utilizes the serial port to generate a pulse signal which emulates an odometer signal by using a pulse signal generator thread (Subsection 3.2.3). Subsection 3.2.4 describes a node used only during CAN-odometry subsystem development to obtain the wheel ring resolution. Subsection 3.2.5 describes the procedure adopted to obtain the nominal wheel circumference. Subsection 3.3 approaches the nodes used to test the CAN-odometry subsystem. The AHRS subsystem nodes are listed in Subsection 3.4. The AHRS output node (Subsection 3.4.1) accounts for device communication. The AHRS status node (Subsection 3.4.2) displays warning messages about AHRS operation. The ROS echo node (Subsection 3.4.3) displays Pose3d messages published by AHRS output node.

3.1 | Custom messages package

This package is composed by a set of .msg files and two common files (CMakeLists.txt and package.xml) of packages structure, see Hansen.[60] A subset of 31 .msg files matches the division of AHRS device output buffer available in SBG Systems.[61,p.12] Each file of this subset corresponds to one of the 31 outputs that may be activated in the device output mask. The outputs are composed by fields with a specific type definition. This type definition follows the SBG System serial protocol specifications available in SBG Systems.[61,p.11] The ROS message definition consists in listing a field type and a field name per line inside the .msg file in text format, where the assigned field type may be a ROS primitive type or other .msg file. An equivalent ROS primitive type was selected for each device type definition in order to keep memory allocation compatibility. Table 6 shows the ROS message definition of the SBG output Euler.

Beyond the SBG output messages subset, the package contains the .msg files shown in Table 7. The RearWheelsPulses.msg is used by CAN publisher node (Subsection 3.2.1) and RTS pulses node (Subsection 3.2.2). The

TABLE 6 ROS msg file of SBG output Euler

Activation mask: SBG_OUTPUT_EULER	ROS msg file: sbgEuler.msg
Roll real32	float32 roll
Pitch real32	float32 pitch
Yaw real32	float32 yaw

Abbreviation: ROS, robot operating system.

TABLE 7 ROS custom msg files

RearWheelsPulses.msg	RearWheelsOdo.msg	Pose3d.msg
Header header	Header header	Header header
uint16 right	uint16 pulses_right	sbgPosition position
uint16 left	uint16 pulses_left	**sbgEuler euler**
	float32 vel_right	sbgAttitudeAccuracy attitude_accuracy
	float32 vel_left	sbgNavAccuracy nav_accuracy
	float32 pulses_per_revolution_right	sbgDeviceStatus device_status
	float32 pulses_per_revolution_left	sbgUTCTime utc_time
		sbgOdoVelocity odo_velocity

Abbreviation: ROS, robot operating system.

RearWheelsOdo.msg is used by the Ring resolution estimator node (Subsection 3.2.4). The Pose3d.msg is used by AHRS subsystem nodes (Subsections 3.4.1, 3.4.2, and 3.4.3). One may see in Table 7 that the SBG Output Euler is a field of Pose3d message.

3.2 | CAN-odometry subsystem

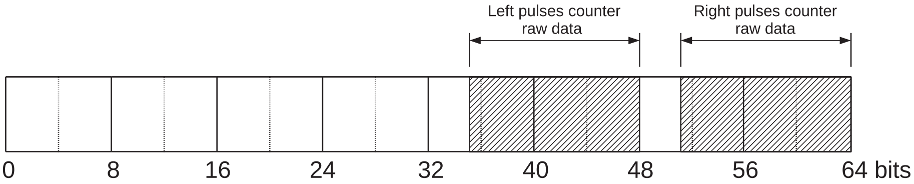
Three CAN messages ids were employed in CAN-odometry subsystem. The id codes were omitted in this work for reasons of confidentiality.

3.2.1 | CAN publisher node

This ROS node basically extracts rear wheels pulse counter from data field of CAN messages filtered according to a specific CAN message identifier and publishes a custom ROS message enclosing the rear wheels pulse counter difference between two consecutive CAN messages. Before CAN publisher node operation, the Advantech CAN device driver (advcan) was set to load automatically during Ubuntu system boot time by using Linux modprobe command. Additionally, the CAN devices permissions were configured to be changed at system boot time, too, in order to allow CAN-odometry ROS nodes have open read access to the CAN devices. The available version of advcan code was developed to be compatible with Ubuntu 12.10 and it was necessary to be modify it in order to enable code compilation under Ubuntu 16.04. After the ROS node is launched, the main code block initializes the CAN device in the selected port using can4linux library. If the CAN device was successfully initialized, the acceptance filtering is configured by using acceptance code and acceptance mask in base frame format with 11 identifier bits.

Once the CAN device boot is completed, this ROS node processes each incoming CAN message in a loop block. The bit fields that correspond to left and right rear wheels pulse counter are extracted from CAN data field by using their bit position as shown in Figure 14. This loop does not have elapsed time constraint, thus, messages are read from CAN buffer as the loop block processing is completed. One may see in Figure 14 that each rear wheel pulse counter has a 13-bit allocated space to receive pulses count information, which allows a maximum of 8191 pulses. When this number is achieved the pulse counter restarts from zero. After extracting the pulse counters, the node calculates the pulses variation

FIGURE 14 Rear wheels pulse counter raw data field bit position



of each rear wheel in relation to values obtained from previous CAN message and adjusts negative values resulting from maximum pulse counter transition. Finally, the pulse counter difference of each wheel is published in a ROS topic at 10 Hz.

The main reason to divide the CAN messages reading and the pulse signal generation in two ROS nodes was to avoid possible delays by extracting rear wheel pulses count and update pulse signal generation frequency at same loop step. The CAN buffer increases cumulatively if the loop step processing time exceeds the time window between two incoming CAN messages, consequently the pulse signal frequency will be updated based on old CAN messages, because they are read following FIFO (first-in, first-out) method. Moreover, this nodes division allows a better code reuse to obtain another useful information from CAN bus and convert them into ROS messages. Nodes division also improves testing procedure by testing CAN publisher node separately from RTS pulses node. Another benefit is that RTS pulses node can be adapted to reproduce other odometry types by changing the ROS message type of its subscriber.

3.2.2 | RTS pulses node

This ROS node produces a pulse signal switching RTS pin state of serial port and adjusts the frequency of the signal according with the ROS messages published by CAN publisher node. The main code block of this node initializes the access to a serial port and creates a thread object which accounts for pulse signal generation. This thread is the core of the node, its operation and validation are described in detail in Subsection 3.2.3. The Linux groups configuration of computer running ROS nodes was modified to allow the access of ROS nodes to serial port without sudo privileges. Two components, a global variable and a ROS subscriber control the frequency of pulse signal generation. The global variable is initialized with zero value and determines the frequency of pulse signal generation. The thread object checks this value at the beginning of each period of the square wave signal. As the thread object is running in parallel to ROS subscriber component, at initialization no pulse signal is generated until the value of global variable has been changed from zero to a positive value.

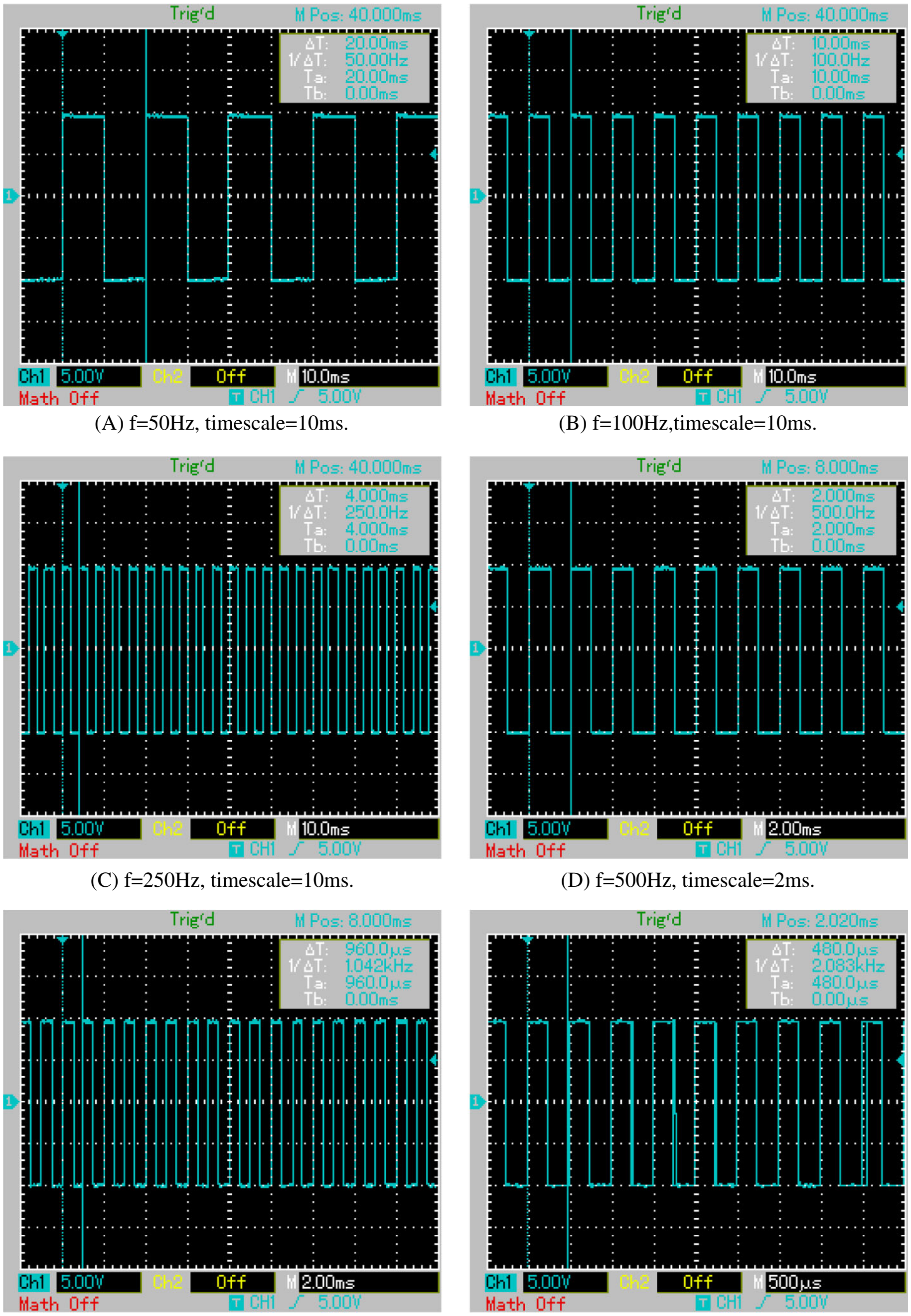
A callback function associated with the ROS subscriber is executed every time a ROS message is published in the ROS topic of subscriber, that is the same ROS topic where CAN publisher node publishes the ROS messages containing the rear wheel pulses variation. The callback function calculates the new frequency value using the pulses variation of the two rear wheels, a mean of these two values corresponds to the pulses variation of a virtual odometer located at the center between the two rear wheels. The use of a virtual odometer provides a better estimation of velocity than use of only the pulses of one of the two rear wheels, mainly in curved trajectories when each rear wheel has a different pulses variation. Moreover, the position of virtual odometer facilitates the extrinsic calibration, because the AHRS was installed in the same vehicle longitudinal axis of symmetry. The new frequency is obtained through Equation (2), where ΔP symbols are pulses variations. As the rear wheels pulses variations are acquired from CAN bus at 10 Hz, the pulses variation must be divided by the period of 0.1 s to obtain a pulses frequency value in Hz. The global variable is updated with the new frequency by using the reduced form of Equation (3).

$$f = \left(\frac{\Delta P_{\text{right}} + \Delta P_{\text{left}}}{2} \right) / 0.1 \text{ s}, \tag{2}$$

$$f = 10\,(\Delta P_{\text{virtual odometer}})\text{ Hz}. \tag{3}$$

3.2.3 | Pulse signal generator thread

The first reason to have chosen the serial port as output for the pulse signal generator was the availability of this port in the same computer with CAN interface. The next step was the development of a C code able to change the signal value in Volts of any pin of DB9 serial RS232 port. So, pin 7 (RTS) of this standard was selected because it is a control



(A) f=50Hz, timescale=10ms.

(B) f=100Hz,timescale=10ms.

(C) f=250Hz, timescale=10ms.

(D) f=500Hz, timescale=2ms.

(E) f=1,000 Hz, timescale=2ms.

(F) f=2,000Hz, timescale=0.5ms.

FIGURE 15 Signals output in the RTS pin by pulse signal generator as seen when probed by an oscilloscope Minipa MO-2200. RTS, request to send

pin and its state can be set via one function of Linux serial programming. The combination of this I/O control function with the sleep function made possible to obtain a pulse signal with C code, where the generation of a single pulse signal consists in changing the RTS pin state from inactive to active, wait for a preselected time, and then return the RTS pin state to inactive. The inclusion of a loop and a sleep time between pulses in the code resulted in a square wave signal. The pulse signal generator thread was probed in an oscilloscope at different frequencies using as input the output of testing node described in Subsection 3.3.1. This measurement procedure was used for testing the first versions of C code, to check signal shape, and to confirm if signal values obtained in the serial port are compatible with AHRS electrical specifications.

Figure 15 shows the pulse signals generated by final version code at different frequencies, as seen when probed by an oscilloscope Minipa MO-2200. One may see in Figure 15A-D that the frequencies estimated by the oscilloscope show no bias in the generated pulse frequencies. A bias of around +4.2% appears at 1000 Hz and 2000 Hz frequencies, as one may see in Figure 15E and F.

TABLE 8 Electrical compatibility verification

Electrical parameter	SBG IG-500E AHRS odometer input	Advantech serial port RTS pin signal values	Compatible devices?
Range	±20 V	±11 V	Yes
Low level	<0.8 V	−11 V	Yes
High level	>2.2 V	+11 V	Yes

Table 8 shows the electrical compatibility verification between SBG AHRS device and Advantech serial port. One may conclude that the serial port output is fully compatible with the AHRS odometer input.

3.2.4 | Ring resolution node

Considering that the wheel magnetic ring resolution is a integer given its constructive nature and that this value is used internally by the vehicle systems, to estimate the ring resolution is to obtain the integer which best fits to the velocities and pulses count values extracted from CAN messages. A ROS node was developed to estimate the odometer ring resolution adopting the hypothesis that the wheels velocity value extracted from a specific CAN message is based on the registered value of wheel circumference (see Subsection 3.2.5) and the wheels cumulative pulses (see Subsection 3.2.1).

The initialization of this node is similar to first steps of CAN publisher node code described in Subsection 3.2.1, such as CAN device access setup and ROS components initialization. Besides to compute rear wheels pulses count difference between two consecutive CAN messages, this node extracts the rear wheels velocities from another CAN message using bit fields presented in Figure 16 and Equation (4).

The equation terms: factor, offset, and unit were obtained from vehicle manufacturer documentation. These computed values were used to estimate the ring resolution integer that best fit the velocity and pulses variation pair.

$$v_{\text{wheel}} = (\text{raw} \cdot \text{factor} + \text{offset})_{\text{unit}} = ((\text{raw})_{\text{Dec}} \cdot 0.0625 + 0)_{\text{km/h}}. \tag{4}$$

Because velocity values less than 2.75 km/h are truncated at this value by vehicle device system, only velocity and pulses variation pairs with velocity values higher than this were considered as valid pairs in the estimation. Due to the 100 Hz rate of the CAN message which contains the wheels velocities raw values, the message use as velocity input to RTS pulses node (Subsection 3.2.2) seems a good choice but the truncation behavior precludes its adoption. One CAN message identifier is broadcast at 10 Hz and the second used CAN message identifier at 100 Hz, this difference makes it difficult to read the two message types simultaneously. A solution to filter the two types using a combination of acceptance code and mask was tested, but it does not work satisfactorily. The implemented solution switches the acceptance code register and running in a loop block at the rate of the slower CAN message identifier at 10 Hz. For each reading of the slower CAN message, 10 faster CAN messages are read in order to prevent the outdated messages accumulation in the CAN buffer due to the set loop rate.

Since pulses variation and velocities of rear wheels were computed in the loop block, the estimated ring resolution is calculated in PPR using the Equation (5), where ΔP_{wheel} is the pulse variation of left or right rear wheel, T is the time period of pulses variation, C_{wheel} is the wheel circumference estimated in Subsection 3.2.5, and v_{wheel} is the velocity of left or right rear wheel. The fraction $\frac{\Delta P_{\text{wheel}}}{T}$ corresponds to pulses rate and the fraction $\frac{C_{\text{wheel}}}{v_{\text{wheel}}}$ represents the time to complete a wheel revolution.

$$\hat{R}_{\text{wheel}} = \left(\frac{\Delta P_{\text{wheel}}}{T} \cdot \frac{C_{\text{wheel}}}{v_{\text{wheel}}} \right)_{\text{PPR}}. \tag{5}$$

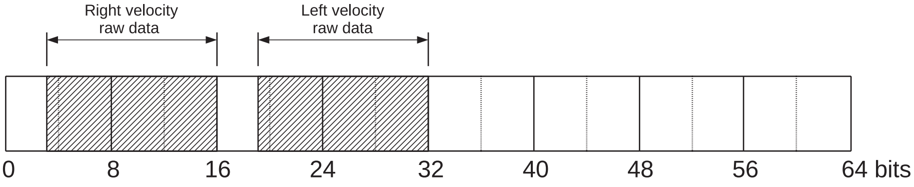


FIGURE 16 Rear wheel velocities data field bit position

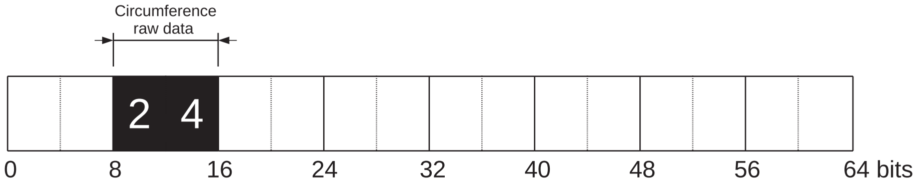


FIGURE 17 Extracted raw data of wheel circumference and data field bit position

Two ring resolutions ($\hat{R}_{\text{wheel}}$) are obtained, one for each rear wheel. At the loop block end, a custom ROS message RearWheelsOdo.msg (see Table 7) is employed to publish velocity, pulses variation, and resolution of each rear wheel. A test drive was carried out to collect and record the data published by this node. The tire pressure was set to 31 psi before test drive execution. The analysis of the recorded data determined that $\hat{R}_{\text{wheel}} = 81$ PPR is the best adjusted integer.

3.2.5 | Wheel circumference calculation

The car wheel circumference value is required to estimate the odometer gain parameter explained in Subsection 2.1.4. This value was obtained from the CAN bus by using a C code containing a library to access the CAN device by filtering user selected CAN messages. The C code shows on the screen the data frame of each read CAN message in hexadecimal. A specific CAN identifier was filtered to extract the wheel circumference-related value, the extracted raw data, and its bit position in the CAN data field is shown in Figure 17.

The extracted raw data is the first term of conversion formula of Equation (6) used to obtain the wheel circumference (C_{wheel}). The terms: factor, offset, and unit were obtained from vehicle manufacturer documentation.

$$C_{\text{wheel}} = (\text{raw} \cdot \text{factor} + \text{offset})_{\text{unit}} = ((24)_{\text{Dec}} \cdot 5 + 1440)_{\text{mm}} = 1620 \text{ mm}. \quad (6)$$

In order to check the obtained value coherence, this was compared with circumference value calculated from measured wheel diameter. The choice to adopt the CAN value came from the best practice of manufacturer value use instead of the manual measurement subjected to inaccuracies, such as tire pressure variations, tire wear, and measuring procedure. No ROS node was implemented to extract CAN raw data containing the wheel circumference, because this extracted value is constant and it needs to be calculated only once.

3.3 | CAN-odometry testing nodes

In order to verify odometer velocities read by AHRS, two nodes were launched during validation tests. The two test nodes publish same message type of CAN publisher node output at 10 Hz. This allows temporarily to replace the CAN publisher node output with custom values for testing purpose. The node launched to test constant velocity values is described in Subsection 3.3.1, and the node developed to test acceleration behavior is presented in Subsection 3.3.2.

3.3.1 | Pub node

The node was launched by using a ROS tool which publishes a ROS message with the inserted values at selected constant rate. Below, the command line used to launch a rostopic node which publishes RearWheelPulses messages in topic call as /test at 10 Hz:

```
rostopic pub -r 10 /test car_msgs/RearWheelsPulses ’{header: seq: 0, stamp: 0, frame_id: odom, right: 10, left: 10}’
```

The count value 10 for each rear wheel is equivalent to 2 m/s using the AHRS settings detailed in Subsection 2.1.4.

3.3.2 | MSG publisher node

A ROS node was developed to reproduce car acceleration behavior because the ROS tool used in the Pub node only works with fixed content messages. This node code creates a ROS publisher object and then enters into a loop block divided in two stages. In the first stage a pulses count variable initialized with zero value is increased at fixed steps. The updated value is assigned to the two ROS message fields which matches rear wheels pulses count, next the ROS message is published to /test topic. The second stage performs the same procedure, but instead, decreasing the last pulses count value. The fixed step value is set on the basis of acceleration parameter value input at node launch, and it has a minimum value of 1 pulse. The increase and decrease steps are limited to 100 steps at each stage, or 10 s at publishing rate at 10 Hz. The odometer velocity showed no strange delays when analysed in SBG Center with acceleration inputs of 2 m/s^2 and 4 m/s^2.

3.4 | AHRS subsystem

3.4.1 | AHRS output node

This node accounts for AHRS external communication. This device model uses the serial protocol to receive commands, to return acknowledge about commands execution, and to send data frames. The device has three types of communication which define how data frames are sent, they are: normal mode, continuous mode, and trigger mode. The main configurations of AHRS were made using SBG Center software and saved to nonvolatile flash memory. Therefore, few commands are sent to AHRS during node operation. The main code block of the node initializes the access protocol of AHRS device in the USB port. If the access was successfully established, the main code sends a command to define the default output mask and another command to enable the communication in continuous mode. The sent commands are C library functions of SBG SDK. The default output mask determines which data fields will be included in the output buffer according to active bit positions. Including all possible outputs in default output mask may lead to output buffer saturation and loss of data frames, as well as embedded EKF malfunction due to AHRS CPU overload.

To run the continuous mode, two handle functions are required; they are callback functions which solve in response to protocol handle call which is carried out in a loop block with controlled rate. One of the handle functions deals with error codes, it is called each time that an error occurs on a continuous frame. The second handle function processes the output buffer, it is called each time that new data are received on a continuous frame. It is in this second callback function that the output buffer data fields values are assigned to fields of a ROS message which is then published. A custom ROS message, as called Pose3d, was created to include the chosen fields of this project which are not present in AHRS message of the ROS sensors standard messages.

Initially, the node was developed to operate in normal mode in which a request is sent to AHRS and a frame is returned with the requested fields. The normal mode has a simpler implementation than continuous mode, but it demands additional AHRS CPU load that results in an output buffer saturation when requesting the set of fields of the implemented project. This wrong status of output buffer saturation was detected during preliminary tests through the warning messages of AHRS status node (Subsection 3.4.2). With the node in continuous mode it was possible to obtain the chosen set of fields without output buffer saturation at the maximum permissible operating rate at 100 Hz.

3.4.2 | AHRS status node

The purpose of this node is to convert the status frame of 32 bits output by the AHRS in human-readable messages. This feature provides the capacity of performance evaluation of AHRS, as well as it allows to inspect possible sources of errors or malfunctions during operation of integrated sensors. The node implementation was made apart from AHRS output node in order to avoid delays in AHRS output node rate. This node was employed during preliminary tests and planned data collection.

The first 21 bit positions of status frame represent 21 statuses defined in SBG Systems,[61,p.19] some of them are only updated during device initialization and others are updated in real-time. If any of these bit positions is set to 0, it means that something is wrong and that the AHRS operation can be affected. As the status frame is updated at 100 Hz, the node was programmed to print only the wrong statuses in the form of ROS warning messages on the screen. Each printed ROS warning message is bit-specific and has a message text according to SBG Systems.[61,p.19] The node subscribes to the ROS topic which the AHRS output node publishes Pose3d messages, the status frame is a field of Pose3d.msg (see Table 7). Using a callback function, the node processes each published status frame and prints applicable ROS warning messages. The bit 17 mask of status frame which corresponds to GPS leap seconds validation was removed from warning messages set, because it does not apply to the model IG-500E due to the fact that NMEA protocol does not support this datum.

TABLE 9 Factor levels of the 2^4 design

Factor	Name	Type	Low	High
A	SVs above horizon	Categorical	13-15	18-20
B	Reference station	Categorical	EESC	SPJA
C	Route	Categorical	UFSCar	USP2
D	Velocity source	Categorical	GNSS	CAN-odometry

3.4.3 | Echo node

The echo node was launched by using a ROS tool which displays messages published to a topic. The command line used to launch a rostopic node, which displays Pose3d messages published by AHRS output node, was: rostopic echo /Pose3d.

4 | EXPERIMENTAL DESIGN AND PROCEDURE

4.1 | Experimental design

An experimental design was carried out to establish if CAN-odometry system improves or worsens positioning accuracy of our INS/GNSS navigation system. The positioning accuracy herein refers to one of the navigation accuracy components of the accuracy output estimated by the EKF-embedded AHRS, that is specified in SBG Systems.[61,p.18] According to SBG Systems,[50,p.43] the position accuracy output of the AHRS device can provide good real-time information about robustness of the embedded filter. A 2^k factorial design was then chosen in order to evaluate the resulting position accuracy under different conditions. As the experimental platform was not equipped with external wheel encoders, the CAN-odometry input was compared with GNSS velocity input, which is the velocity source allowed by the AHRS in the absence of odometer pulses signal. Thus, velocity source was the first factor that was included in the 2^k factorial design.

The AHRS alone does not estimate a position, it requires the input from a position source. In this way, the AHRS provides a position estimation based on: the last estimated state, the internal sensors inputs (gyroscopes, accelerometers, magnetometers, and temperature sensors), and the external sensors inputs (odometer and GNSS). This estimation task is performed by an EKF embedded in the AHRS's CPU. The EKF loop rate allows the AHRS to provide a position estimation at 100 Hz rate, higher than the configured GNSS output at 10 Hz. The embedded EKF also provides an estimation during GNSS outages, as well as a smooth transition under high variations in accuracy of GNSS output. However, the position accuracy of EKF solution is limited by the position accuracy of GNSS output.

Several factors can affect the performance of the embedded EKF, including the calibration issues previously mentioned in Subsection 2.1. Assuming that calibration errors of AHRS internal sensors equally affect the position accuracy in the two velocity input modes, the choice of other factors of the 2^k factorial design was focused on effects that impact the positioning accuracy of the GNSS PVT solution. In this context, some effects that can be predicted or controlled were included in the final design. Besides the velocity source factor, three other factors were included in the 2^k factorial design, resulting in a 2^4 factorial design. Table 9 shows the four factors and their respective levels. The details about each factor are described in next subsections.

Initially, a replicated 2^4 full factorial design with 8 partial blocks was created in order to consider differences between days of the experiment. Blocking of the replicated 2^4 full factorial design reduces the influence of extraneous factors on the analysis, such as meteorological conditions, space weather influence, ionospheric irregularities, among others. In this way, four experimental runs were initially planned for each experiment day, totalling 32 runs over 8 days. The 4-run per block choice was established because of experimental time constraints per day, and due to the run order generated by randomization which may require a long waiting time for the change in the level of SVs above horizon, besides of the short time windows at each level (which are better explained in Subsection 4.1.1).

Moreover, if one of the runs in a block fails, it would be necessary to repeat all block runs, because the failing run must be repeated on the same day and it must follow the established run order. Consequently, this would increase the overall experimental time and the experiment costs. Furthermore, the partial blocking contribution of 1/8 is minimal on the analysis step of factorial design when compared to full block contribution. For this reason, the final replicated 2^4 full factorial design contains only one full block that ignores influence related to each day of the experiment on the analysis. With one block it was possible to allocate on the same day all runs allowed by the established run order. The two replicates choice, which increases the number of runs by 16, was necessary to compute p-numbers, required to classify significant effects.

4.1.1 | Factor A: SVs above horizon

The availability of SVs above horizon is a key factor in the computation of PVT solution by the GNSS. So that, a better accuracy in the solution is expected for a higher number of SVs above horizon. As position of this solution is sent to AHRS via NMEA messages, a better performance of EKF solution is expected with a greater number of SVs above horizon, too. This number varies according to day time and depends on geographical location. In order to control this factor during experimental runs, it became necessary to make a prediction of the SVs above horizon at the route waypoints. Some software and online tools provide this prediction based on user input parameters. In this work we used the RxPlanner tool from Septentrio NV/SA[51] to obtain the prediction of SVs above horizon for the experimental runs days. The RxPlanner computes the satellites trajectories based on an almanac automatically updated during the software loading procedure, and it provides among other information the satellites number above horizon with minutes resolution. Figures 18 and 19 show the linear chart of the predicted SVs above horizon generated by RxPlanner for two different locations, one on each route. The prediction period was from 07:00 AM to 07:00 PM on 19 September 2017.

As one may see in Figures 18 and 19, the time windows at the higher quantity of satellites, as well as the time windows at the lower quantity are very short, some of them have a duration of only 4 min. The mean estimated time to complete the routes is around 15 min and 12 min, for UFSCar and USP2 routes, respectively. So, these time windows are not enough to complete the routes. As the route is also an experiment factor, short time windows would restrict the route extension if a fixed number was applied at each level of SVs above horizon factor. Because of this, we adopted a quantity range for each Factor A level instead of fixed quantities, in order to allow the vehicle to always complete the routes at the planned level. Due to this, the factor type was defined as being of categorical type instead of numerical type. This choice reduces the expected accuracy of the model, but it provides a greater flexibility to plan the data acquisition. One may conclude that the 2^k design approach is not the best choice for models refinements, it applies better to explore significant factors and interactions.

On the other hand, the predicted quantity of satellites is computed based on the coordinates of a fixed location. It is not feasible for the planned experiment to estimate or to verify the satellite availability for many waypoints across the routes for each experiment day. One may observe that, by comparing the two predictions (Figures 18 and 19), the difference between the two is minimal considering that distance between the two routes is about 4 km. That way, it is possible to adopt only one reference point at each route with a minimal loss in prediction accuracy. The predicted number of SVs was confirmed at the beginning and at the ending of each experimental run using RxControl.[51] Sometimes, it was necessary to wait for about 2 min for the number of SVs in view to change to the predicted value.

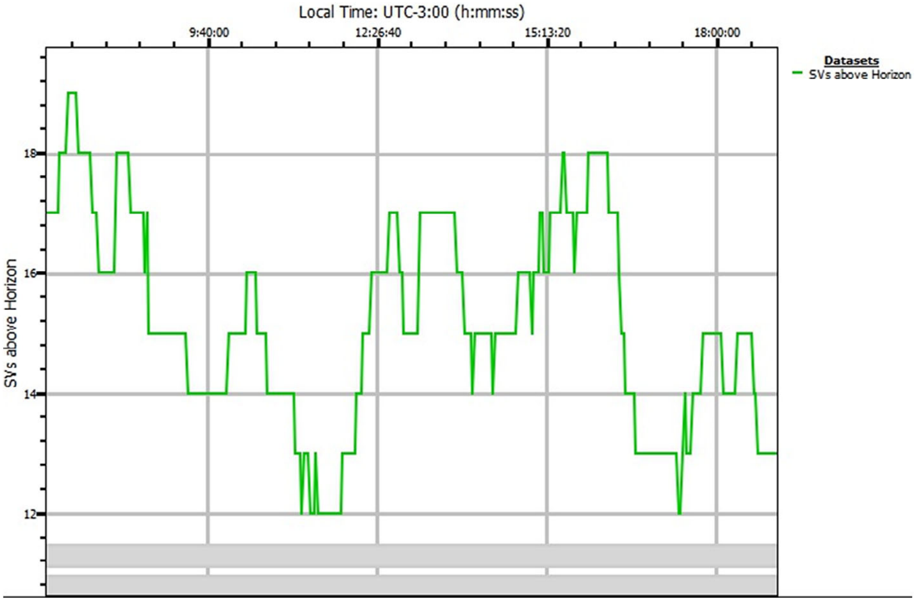


FIGURE 18 Prediction of satellites above horizon on 19 September 2017 (experiment day 7) at UFSCar reference point

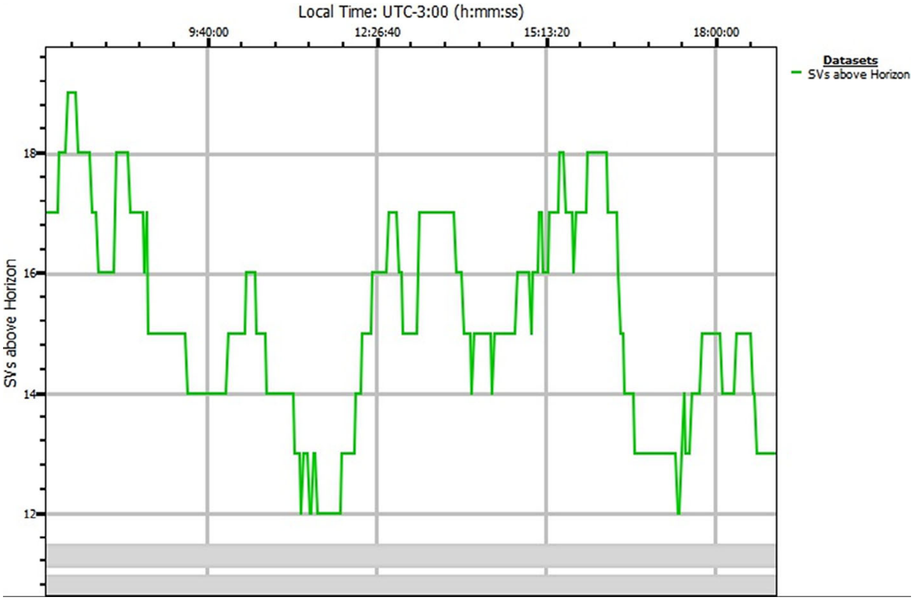


FIGURE 19 Prediction of satellites above horizon on 19 September 2017 (experiment day 7) at USP2 reference point

In addition to the location in geodetic or Cartesian coordinates, the elevation mask and the satellite selection are required for the SVs prediction computation. For all runs of the presented planned experiment an elevation mask of 10° was adopted, the same elevation mask configured in the GNSS settings (see Subsection 2.2.2). All satellites of the GPS and the GLONASS constellations were selected, the same satellite selection applied in the GNSS settings (see Subsection 2.2.2), no exception for unhealthy satellites in the Septentrio almanac was applied.

The fact of two sets having the same number of SVs above horizon does not imply that the sets will produce similar PVT solution accuracy. The different observation geometry of each set may result in a relevant difference in the dilution of precision (DOP). Due to this geometry factor, an experimental design approach that includes the DOP as a factor in place of the number of SVs above horizon may yield a more precise model of the effects. This model refinement based on the DOP will be addressed in future works.

4.1.2 | Factor B: Reference station

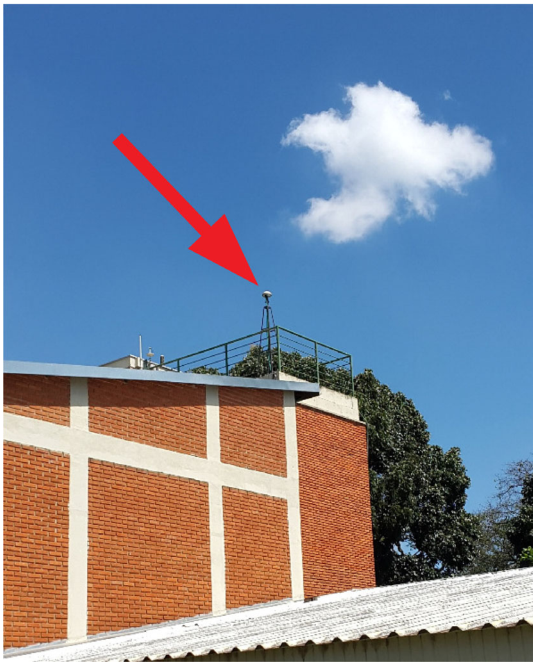
A better accuracy in the PVT solution is expected by using differential corrections coming from a nearby reference station than from one located far away. Among all the stations connected to IBGE/RBMC Ntrip broadcaster, the mountpoint located in the EESC-USP (Figure 20) is the nearest to this project facilities, and it was the first to be chosen.

One may observe in Figure 21 that there are two stations (SPJA and SPPI) nearby to EESC station, both about 90 km far away. As SPJA is the station nearest to the place of an experiment of a future work, a mountpoint of this one was chosen as second reference station to the planned experiment. The location of RBMC's stations in zipped KML (Keyhole Markup Language) format (.kmz) was retrieved from IBGE.[62] Besides the distance criterion for mountpoints selection, only those which transmit differential corrections of GPS and GLONASS systems in RTCM 3.0 format were used. No differentiation was made regarding equipment models installed in the stations.

4.1.3 | Factor C: Route

In order to evaluate impact of the occlusion effects during the experiment, we used two predefined routes during the experimental runs. Figures 22 and 23 show the two selected routes plotted on satellite images. The images were generated using Google Earth Pro. Aiming to insert the path of each route in the maps, two KML files were extracted from experimental run logs using SBF converter tool provided by Septentrio, and loaded in Google Earth Pro.

FIGURE 20 GNSS antenna of EESC0 mountpoint (photograph taken by author on 6 April 2018)



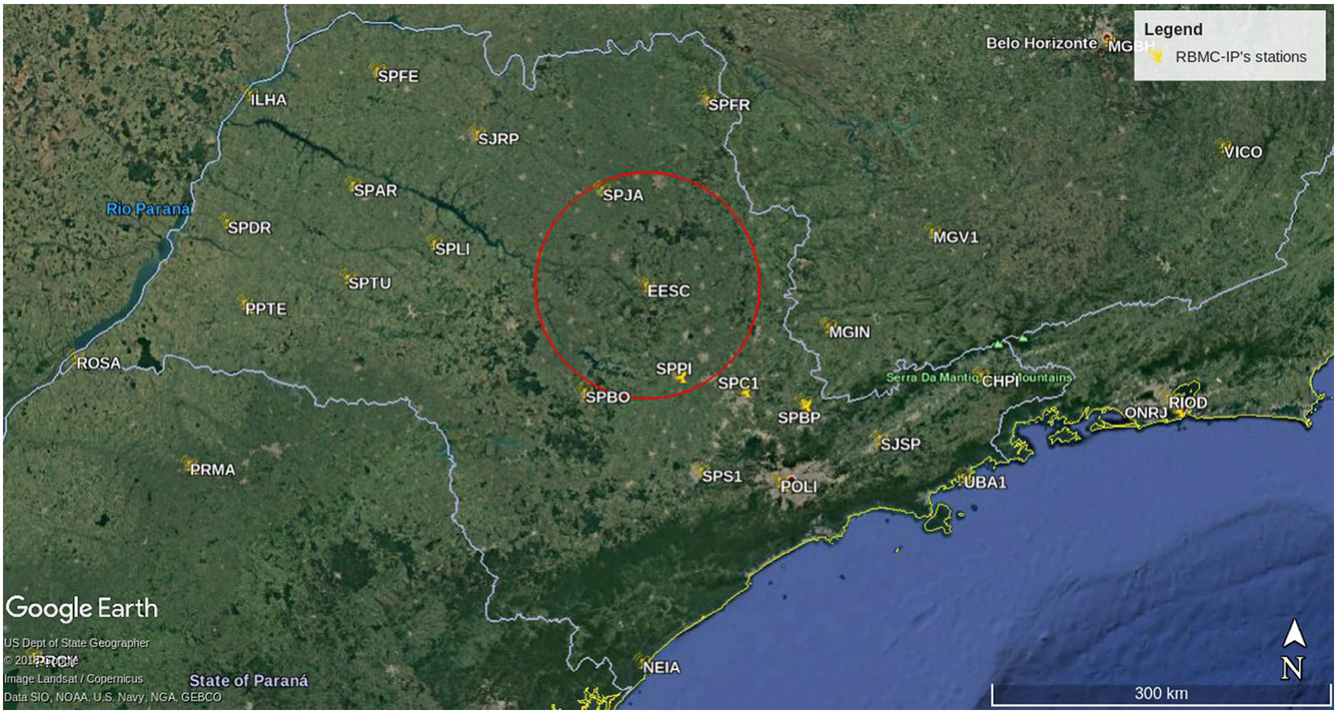


FIGURE 21 Satellite images generated by Google Earth Pro pointing out the RBMC's stations located nearby EESC station in a 100 km radius (red circle)

The two routes differ in their buildings density, as well as in type and number of roadside trees across the route path. One may observe in Figure 22 that the UFSCar route has a higher buildings density than USP campus 2 route (Figure 23). Also, the path of UFSCar has more route stretches surrounded by roadside trees than USP campus 2 path. One of the most occluded stretches is located in the south area of UFSCar path (left bottom side of cyan path). This route stretch is surrounded by tall eucalyptus trees on both sides of the road. Due to this, a higher impact in the embedded EKF performance due to occlusions is expected during the UFSCar runs.

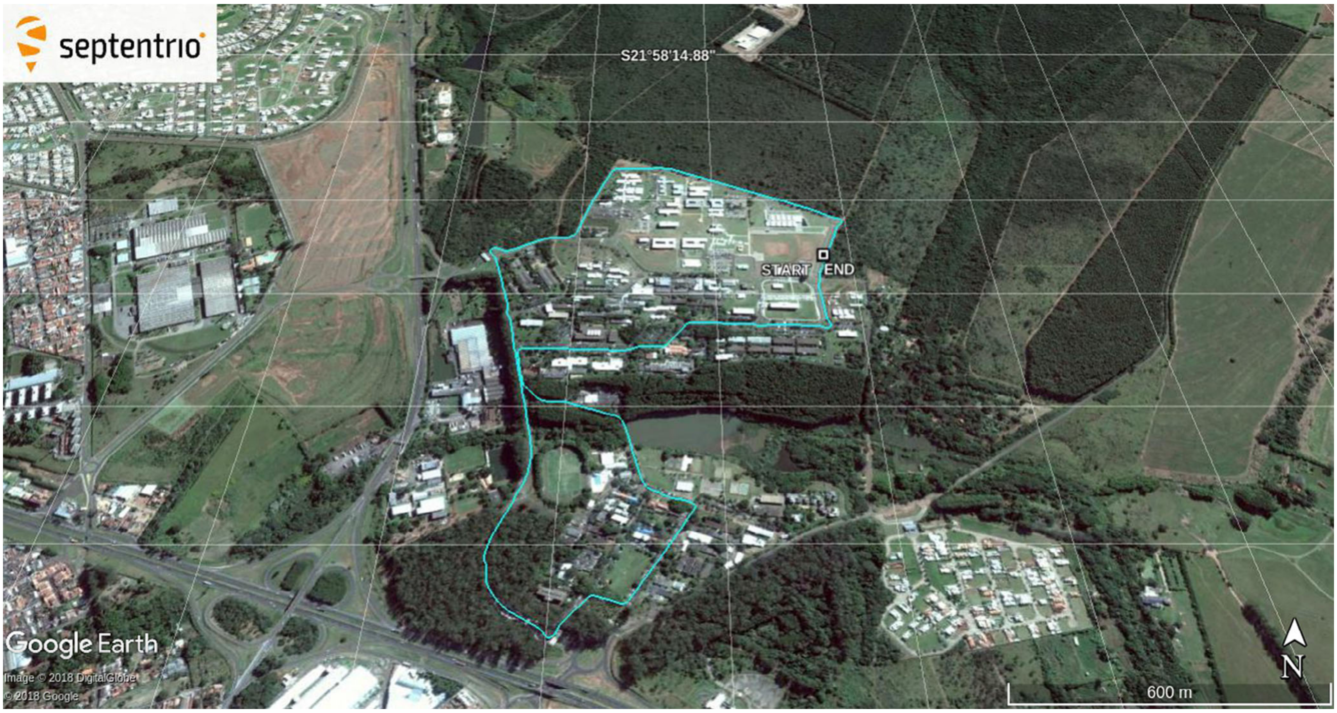


FIGURE 22 Satellite image generated by Google Earth Pro pointing out the UFSCar route (cyan path)

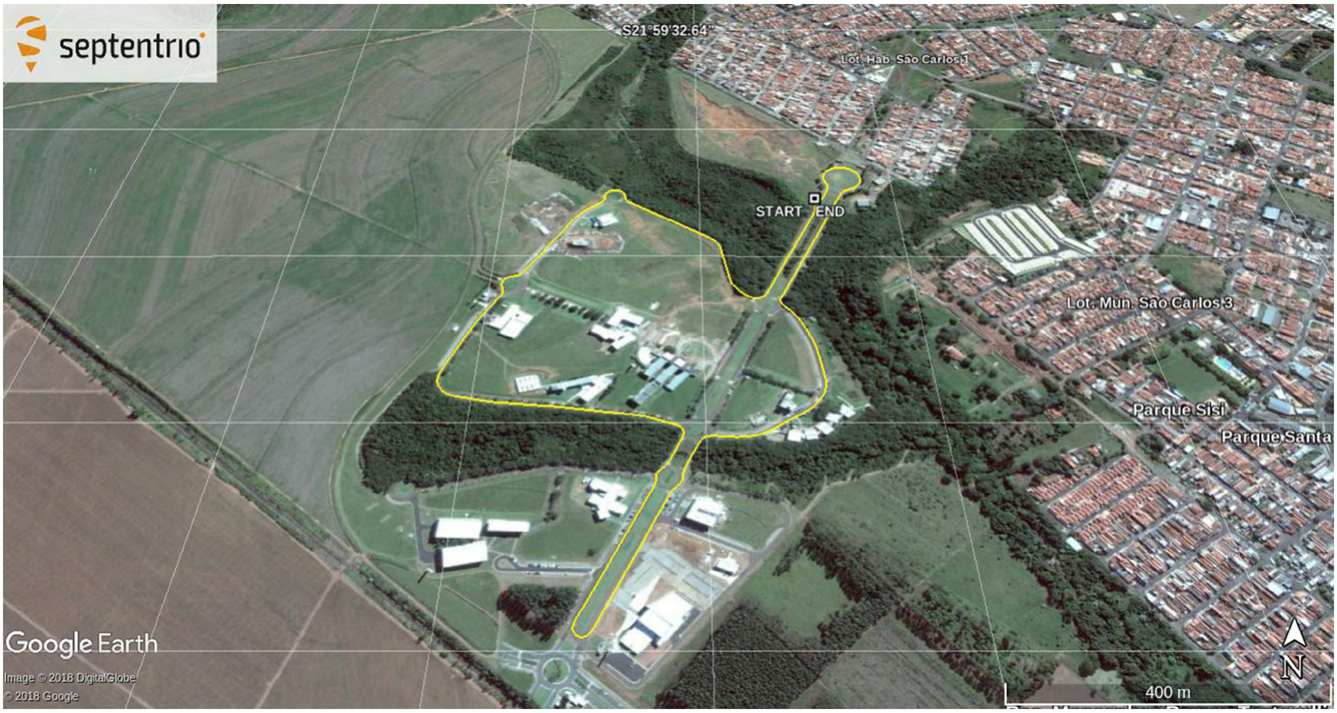


FIGURE 23 Satellite image generated by Google Earth Pro pointing out the USP2 route (yellow path)

TABLE 10 Approximate distance from route waypoints to reference stations in kilometers

	Mountpoint	
Route	**EESC0**	**SPJA1**
UFSCar	2.25-3.73	91.7-92.9
USP Campus 2	3.00-3.66	91.5-92.4

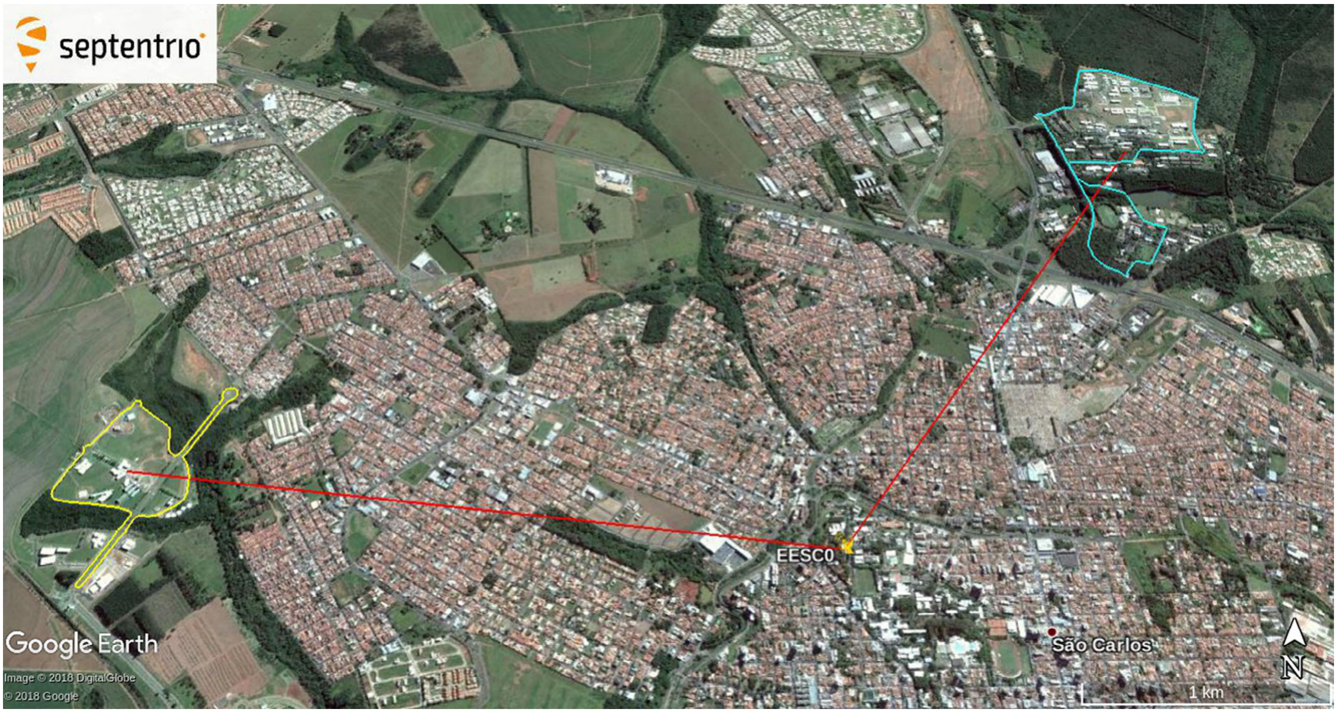


FIGURE 24 Satellite image generated by Google Earth Pro pointing out the routes UFSCar (cyan path) and USP campus 2 (yellow path), as well as the antenna of EESC0 mountpoint (yellow icon)

Since the distance to reference station is also a performance factor, the routes must be carefully chosen so that their distances to the reference stations are approximately equal. This procedure helps to mitigate that occlusion effects be confused with the accuracy variation coming from bias in the distance to the reference station. Table 10 shows the approximate distances between the routes waypoints and the selected reference stations. These values can be obtained by processing the GNSS logs, however as this choice should be made during the planning of the experiment, it was carried out by using Google Earth Pro measuring resource. Figure 24 illustrates through red lines the distance from the antenna of EESC0 mountpoint to the routes UFSCar and USP campus 2.

4.1.4 | Factor D: Velocity source

This factor allows to compare the EKF performance under two different velocity inputs, one provided by the GNSS and other obtained from the odometer emulated by the CAN-odometry framework. As the GNSS has been configured to operate in the RTK fix mode (see Subsection 2.2.2), always that the GPS/GLONASS signals availability and the received differential corrections allow it, the expected accuracy of GNSS velocity input will be the better possible. The setup procedure to set the factor level at each experimental run requires only one setting change and it was carried out by using SBG Center settings interface in Velocity source option of Navigation tab settings during experimental run preparation.

Whereas the CAN-odometry output has an independent behavior, the velocity estimation output by GNSS is subjected to variations that depend on the three other experiment factors, due to their influence in the accuracy of the PVT solution. This dependence leads to expect significant contributions of two-way interactions between factors and it was one of the reasons to adopt a 2^k factorial design for the planned experiment.

4.2 | Experimental procedure

In order to reduce unavoidable experimental errors, a standard procedure was established to carry out the experimental runs. A waiting time of at least 15 min was applied before the start of each run in order to allow the synchronization between the GNSS and the AHRS, as well as equipments warm-up. All runs were carried out only during sunny days. A start and end point was chosen for each route, and same points were used during all experimental runs. The criteria for selection of start and end points, called herein as reference points, were low occlusion of SVs, and impact on the local transit of each route. The settings of each experimental run were checked at the reference points before the start. Table 11 shows the geodetic coordinates of reference points used to plan day and time of each experimental run by using RxPlanner tool to predict the SVs above horizon. The pictures of the experimental platform parked at each location of the reference points are shown in Figures 25 and 26.

4.2.1 | Magnetic declination estimation

The magnetic declination refers to the angle of difference between true North and magnetic North.[63] In this work, we opted for manually setting the magnetic declination value instead of activating automatic estimation by the AHRS. This procedure allows the choice of model used in the estimation. The declination value was calculated using the most recent World Magnetic Model (WMM)[64] sponsored by the U.S. National Geospatial-Intelligence Agency (NGA) and the U.K. Defence Geographic Centre (DGC).[65] An online calculator provided by the National Centers for Environmental Information (NCEI)[63] was used to calculate declination value input in AHRS filter settings at each run of the designed experiment. These values were calculated by using as input parameters the latitude and the longitude of routes reference points as well as the experimental run date. Figure 27 shows the NCEI calculator result in HTML format for the reference point of USP Campus 2 on 4 September 2017. This result was used in the first four runs of the designed experiment.

	Reference point	
Geodetic coordinates	UFSCar	USP2
Latitude (φ)	S 21°58′49.93627″	S 21°59′54.20501″
Longitude (λ)	W 47°52′37.80188″	W 47°55′42.07644″
Height (h)	867.910 m	836.540 m

TABLE 11 Geodetic coordinates of reference points in DMS (degrees minutes seconds)

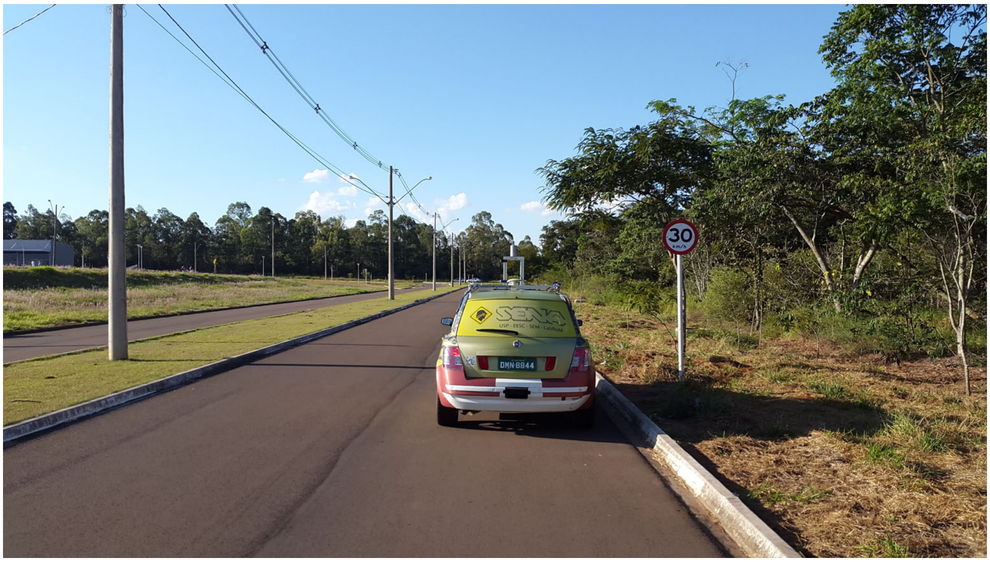


FIGURE 25 Picture of the location of the UFSCar reference point

FIGURE 26 Picture of the location of the USP2 reference point

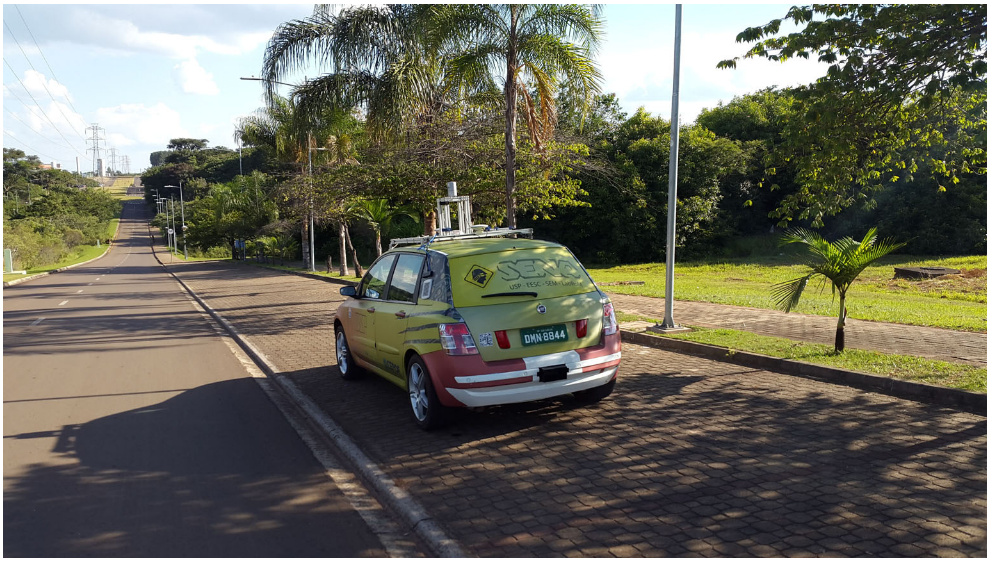
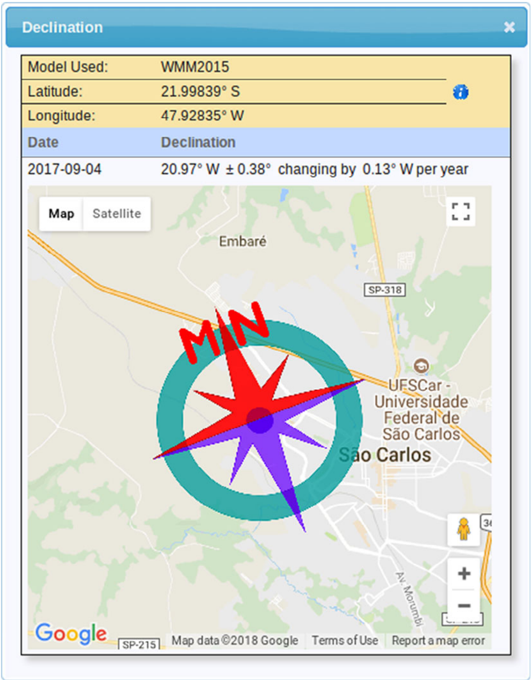


FIGURE 27 Estimated declination by NCEI calculator. NCEI, National Centers for Environmental Information



Declination

Model Used:	WMM2015
Latitude:	21.99839° S
Longitude:	47.92835° W
Date	Declination
2017-09-04	20.97° W ± 0.38° changing by 0.13° W per year

4.2.2 | Experiment logs

A set of logs was made from the experimental runs. A bag file was recorded to save the published messages in the ROS topics at each run as previously shown in Figure 13. A software tool called RxLogger provided by Septentrio NV/SA[51] was configured, it runs during each experimental run in order to produce two log files. One of them is used to keep a record of raw data received from GPS and GLONASS systems in a SBF (Septentrio Binary Format) file, as well as the PVT solution in geodetic coordinates, the received differential corrections, the status of GNSS, and the attitude estimation. The second log file contains the same NMEA streams configured in COM2 port (see Table 4) in order to look for fail conditions and to perform a deeper analysis of the NMEA messages values sent to the AHRS, if required.

4.2.3 | Logs processing

The EKF-embedded AHRS-estimated position metric performance is published in a subfield of sbgNavAccuracy field of Pose3d message type previously described in Table 7. One way to extract the message fields values from bag files is by playing them and reading the messages by using a ROS subscriber. This is a good way to combine messages from different topics and to test custom ROS nodes. However, beyond the time spent to write the custom subscriber code, it demands time to play the bag files. Due to this, a rostopic echo command was used in order to extract the Pose3d messages from the bag files in a faster way. The command below extracts the Pose3d messages from a bag file and writes them in a text file in CSV (Comma-Separated Values) format.

```
rostopic echo -p -b log_file.bag/Pose3d > log_file.Pose3d.
```

The CSV files were then analyzed in MATLAB environment. Before computing the responses presented in Table 12, a cut-off threshold was established for each experimental run in order to exclude values recorded during the delays between

TABLE 12 Replicated 2^4 full factorial design matrix with the extracted responses

Order				Factors				Position accuracy	
Standard	**Run**	**Center point**	**Blocks**	**A**	**B**	**C**	**D**	**Mean**	**Minimal**
2	1	1	1	18-20	EESC0	USP2	GNSS	0.1009	0.0273
3	2	1	1	13-15	SPJA1	USP2	GNSS	0.4110	0.2453
1	3	1	1	13-15	EESC0	USP2	GNSS	0.6408	0.2453
4	4	1	1	18-20	SPJA1	USP2	GNSS	0.4022	0.2277
31	5	1	1	13-15	SPJA1	UFSCar	CAN-odometry	0.9639	0.2435
22	6	1	1	18-20	EESC0	UFSCar	GNSS	5.9997	0.0273
13	7	1	1	13-15	EESC0	UFSCar	CAN-odometry	2.2471	0.0319
28	8	1	1	18-20	SPJA1	USP2	CAN-odometry	0.9893	0.2262
23	9	1	1	13-15	SPJA1	UFSCar	GNSS	9.4465	0.2453
5	10	1	1	13-15	EESC0	UFSCar	GNSS	8.5428	0.2612
21	11	1	1	13-15	EESC0	UFSCar	GNSS	5.9598	0.2453
12	12	1	1	18-20	SPJA1	USP2	CAN-odometry	0.2871	0.2262
7	13	1	1	13-15	SPJA1	UFSCar	GNSS	13.8355	0.2453
6	14	1	1	18-20	EESC0	UFSCar	GNSS	6.5145	0.0273
26	15	1	1	18-20	EESC0	USP2	CAN-odometry	0.7238	0.0273
16	16	1	1	18-20	SPJA1	UFSCar	CAN-odometry	5.2855	0.2263
9	17	1	1	13-15	EESC0	USP2	CAN-odometry	0.0802	0.0319
15	18	1	1	13-15	SPJA1	UFSCar	CAN-odometry	0.5471	0.0319
32	19	1	1	18-20	SPJA1	UFSCar	CAN-odometry	11.0421	0.2262
20	20	1	1	18-20	SPJA1	USP2	GNSS	0.3057	0.2277
8	21	1	1	18-20	SPJA1	UFSCar	GNSS	8.2502	0.2277
27	22	1	1	13-15	SPJA1	USP2	CAN-odometry	1.8200	0.2435
14	23	1	1	18-20	EESC0	UFSCar	CAN-odometry	7.9623	0.0273
25	24	1	1	13-15	EESC0	USP2	CAN-odometry	0.0796	0.0319
30	25	1	1	18-20	EESC0	UFSCar	CAN-odometry	0.9651	0.0273
18	26	1	1	18-20	EESC0	USP2	GNSS	1.5976	0.0273
10	27	1	1	18-20	EESC0	USP2	CAN-odometry	0.1397	0.0273
17	28	1	1	13-15	EESC0	USP2	GNSS	0.1087	0.0319
24	29	1	1	18-20	SPJA1	UFSCar	GNSS	2.6284	0.2453
19	30	1	1	13-15	SPJA1	USP2	GNSS	0.6918	0.2612
29	31	1	1	13-15	EESC0	UFSCar	CAN-odometry	3.4014	0.0364
11	32	1	1	13-15	SPJA1	USP2	CAN-odometry	2.4894	0.2435

the log recording and the vehicle movement at the start and the ending of each route. A MATLAB code was developed to perform the steps mentioned above, and the resulting outputs were saved using diary function. The diary files organized by experimental runs were then used to input the responses values in Minitab environment to perform the DOE analysis.

5 | RESULTS

In a 2^4 full factorial design all 16 combinations of low and high levels for each of four factors are employed. This was the final design chosen to evaluate the performance of the position estimated by the INS/GNSS navigation system. The adoption of two replicates, in order to obtain an estimate of experimental error, resulted in 32 experimental runs. Table 12 shows the 32 experimental runs in randomized order with respective response values extracted from log files using the MATLAB codes. In Table 12, the low and high factor levels are uncoded, that is, they are in categorical text instead of coded as −1 and +1. This worksheet data format was chosen to better control the setup of experimental runs during data collection. The first column of Table 12 represents the standard order of runs after randomization performed by Minitab utilizing an optional decimal number as reference, and the second column, the order for execution of experimental runs. In Minitab options, the decimal number 11 was employed for randomization. The use of reference numbers allows the random order reproducibility by other researchers that use Minitab in their experiments. No additional center points and neither additional blocks were included in the design definition as can be seen in the third and fourth columns of the design matrix (Table 12).

Two response types were extracted and computed from the data logs of experimental runs using MATLAB codes. The response values were then recorded in DOE worksheet of Minitab 18 in order to carry out analysis of factorial design. The last two columns of Table 12 show responses of the mean values of position accuracy and the minimal values of position accuracy of each experimental run, respectively, where position accuracy refers to positioning accuracy computed by the EKF-embedded AHRS device. The analysis of factorial design was done separately for each response type, and it was divided in four steps. Subsection 5.1 describes the results analysis of "Mean Position Accuracy" response type; Subsection 5.2 describes the results analysis of "Minimal Position Accuracy" response type. Subsections of 5.1 and 5.2 describe details of the analysis steps. A statistical significance criterion was manually applied in the analysis procedure of factorial design. Thus a significance level of .05 was employed in the analysis of the two response types. The experimental design choice, as well as the factorial design analysis were based on concepts and procedures extracted from several references.[66-76]

5.1 | Response: Mean position accuracy

5.1.1 | Step 1: Determination of significant terms

Table 13 shows the analysis of variance (ANOVA) table generated by Stat > DOE > Factorial > Analyze Factorial Design menu option of Minitab 18, where the column associated with "Mean Position Accuracy" values was entered in Responses field. As shown in this ANOVA table, the *P* values for linear terms "Route" (C) and "Velocity source" (D), 2-way interactions "SVs above horizon" * "Velocity source" (AD) and "Route" * "Velocity source" (CD), as well as 3-way interaction "SVs above horizon" * "Route" * "Velocity source" (ACD) are significant because they are less than the employed significance level of .05.

Figure 28 shows Pareto chart of the standardized effects generated by Minitab 18. The red line threshold in the Pareto chart indicates the reference *T* value for a *P* value of .05. The effects that surpass this red line are statistically significant. As can be seen, Route (C) is the effect with greater impact. Out of the main effects, only "C" and "D" significantly impact the mean of position accuracy values output by the AHRS along the routes. It was expected a significant contribution of main effect "A" because more SVs above horizon tend to increase accuracy of PVT solution computed by GNSS. Also, a significant impact of main effect "B" was expected since differential corrections coming from a closer reference station provide a PVT solution with better accuracy. The ANOVA output (Table 13) does not imply that main effects "A" and "B" do not affect position accuracy of AHRS. Since literature states that variations in these factors impact position accuracy of PVT solution, this ANOVA result indicates that the chosen factorial design did not have sufficient statistical power to detect main effects "A" and "B" under the chosen experimental conditions, and with a significance level of .05.

The interpretation of the results as well as a deeper analysis about nondetected effects are provided in Step 4 (Subsection 5.1.4). A model reduction was carried out in order to simplify the obtained model and to increase the precision

TABLE 13 ANOVA table for response "Mean Position Accuracy"

Source	DF	Adj SS	Adj MS	*F* value	*P* value
Model	15	375.975	25.065	5.52	0.001
Linear	4	242.188	60.547	13.33	0.000
A	1	0.116	0.116	0.03	0.875
B	1	6.419	6.419	1.41	0.252
C	1	213.852	213.852	47.10	0.000
D	1	21.801	21.801	4.80	0.044
2-way interactions	6	60.433	10.072	2.22	0.095
AB	1	0.490	0.490	0.11	0.747
AC	1	0.938	0.938	0.21	0.656
AD	1	27.387	27.387	6.03	0.026
BC	1	1.313	1.313	0.29	0.598
BD	1	0.054	0.054	0.01	0.914
CD	1	30.251	30.251	6.66	0.020
3-way interactions	4	54.832	13.708	3.02	0.049
ABC	1	1.203	1.203	0.26	0.614
ABD	1	11.956	11.956	2.63	0.124
ACD	1	39.096	39.096	8.61	0.010
BCD	1	2.577	2.577	0.57	0.462
4-way interactions	1	18.522	18.522	4.08	0.060
ABCD	1	18.522	18.522	4.08	0.060
Error	16	72.652	4.541		
Total	31	448.627			

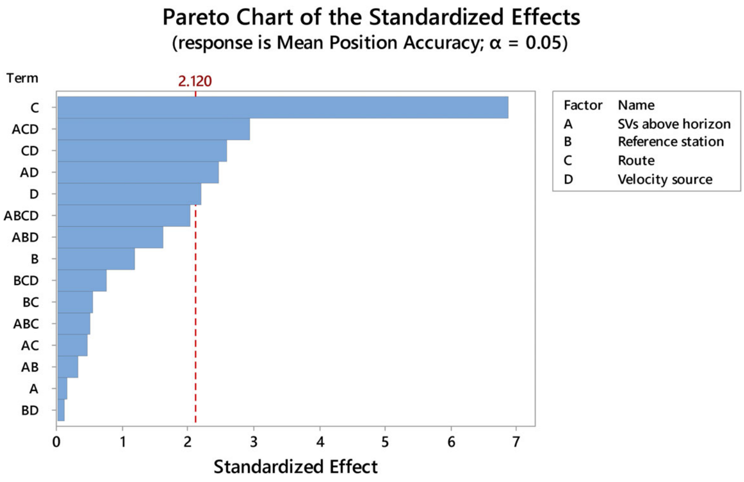


FIGURE 28 Pareto chart of the standardized effects for response "Mean Position Accuracy"

of predictions. This procedure consists of conducting a new analysis of factorial design using only significant terms. The results after model reduction are presented in the next step (Subsection 5.1.2).

5.1.2 | Step 2: Model reduction

Table 14 shows the ANOVA results after factorial design analysis of the reduced model. The significant terms before model reduction remained significant in the new ANOVA, however, with minor variations in their *P* values.

TABLE 14 ANOVA table for response “Mean Position Accuracy” after model reduction

Source	DF	Adj SS	Adj MS	*F* value	*P* value
Model	5	332.39	66.478	14.87	0.000
Linear	2	235.65	117.827	26.36	0.000
C	1	213.85	213.852	47.83	0.000
D	1	21.80	21.801	4.88	0.036
2-way interactions	2	57.64	28.819	6.45	0.005
AD	1	27.39	27.387	6.13	0.020
CD	1	30.25	30.251	6.77	0.015
3-way interactions	1	39.10	39.096	8.74	0.007
ACD	1	39.10	39.096	8.74	0.007
Error	26	116.24	4.471		
Lack-of-fit	10	43.59	4.359	0.96	0.510
Pure error	16	72.65	4.541		
Total	31	448.63			

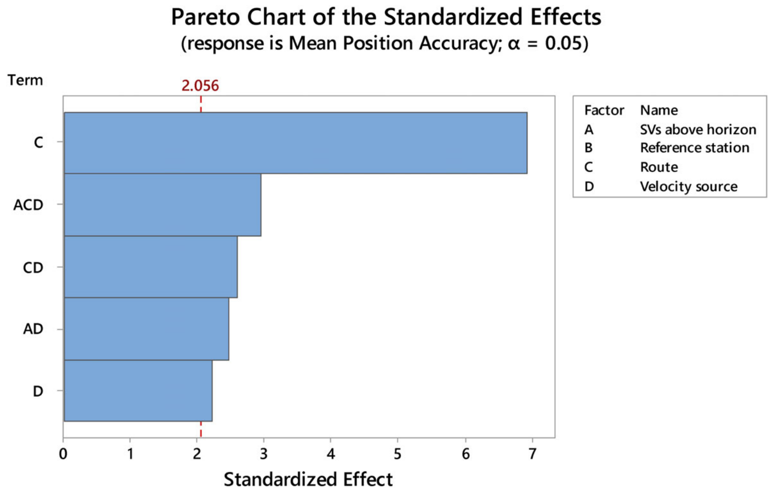


FIGURE 29 Pareto chart of the standardized effects for response “Mean Position Accuracy” after model reduction

TABLE 15 Model summaries for response “Mean Position Accuracy”

	Statistics			
Model	***S***	***R*²**	***R*² (adjusted)**	***R*² (predicted)**
Original	2.13090	83.81%	68.62%	35.22%
Reduced	2.11441	74.09%	69.11%	60.75%

The order of the significant terms was also kept as shown in the Pareto chart of Figure 29 generated by Minitab 18.

Table 15 shows comparison of model statistics before and after reduction. The decrease of *S* indicates that the reduced model fits slightly better the response “Mean Position Accuracy.” According to difference in R^2, the percentage of variation in the response explained by the reduced model is smaller due to the reduction of predictors. Since the two models have a different number of predictors, the adjusted R^2 provides a more appropriate evaluation of the model improvement. Therefore, the increase of the adjusted R^2 indicates a real improvement in the reduced model. In original model, the substantially less value of predicted R^2 than R^2 indicates that the first model may be over-fit. Lastly, the largest value of predicted R^2 of the reduced model, as compared to the original model, confirms the improvement after model reduction.

5.1.3 | Step 3: Validation of model assumptions

Figure 30 and Figure 31 show residuals plots for response "Mean Position Accuracy" before and after model reduction, respectively. The histogram (bottom left plot) of both figures shows no skewness or suggests outliers. Nonetheless, the normal probability plot (top left plot) of both models shows a nonlinear pattern, therefore the residuals do not follow a normal distribution. This pattern indicates a symmetric light-tailed distribution. This means that outliers are produced in both sides of the distribution at a reduced rate from what it would be expected from a normal distribution. The residuals versus fits plot (top right plot) of both models suggests that the variance of the residuals are unequal due to the uneven spreading across fitted values. The residuals versus order plot (bottom right plot) of both figures shows no trends or patterns in the points that indicate correlation between near residuals in time order. In this plot, it is possible to identify large residual observations diagnosed by Minitab 18 (observations 23 and 25 in Figure 30, and observations 13, 19, and 25 in Figure 31).

Since both models did not meet the assumption that the residuals are normally distributed, as well as violate the assumption that the variance of the residuals is constant, the obtained models may not fit well the data.

5.1.4 | Step 4: Interpretation of results

As the obtained model does not meet all factorial analysis assumptions, the magnitude of the effects can be inaccurate. However, the results analysis provides a good insight about factors influence on the positioning accuracy of INS/GNSS navigation system, as well as about factors interaction. Figure 32 shows the main effects that are statistically significant for "Mean Position Accuracy" response. As can be seen in the plot, the UFSCar route increases the mean position accuracy value, on average, by 5.170 m. This effect is associated with the highest number of occlusion of the UFSCar route. This term is specific because it reflects the difference between occlusion levels of the two routes. The second main effect in the plot indicates that the CAN-odometry reduces the mean position accuracy value, on average, by 1.651 m. As there are significant interactions, the interpretation of the results should also consider the impact of these interactions.

Figure 33 shows the 2-way interactions that are statistically significant for response "Mean Position Accuracy." The left interaction of Figure 33 shows the effect of the SVs above horizon when combined with the velocity sources. In the range of 13-15 SVs above horizon, the GNSS velocity increases the mean position accuracy value, on average, by 3.501 m more

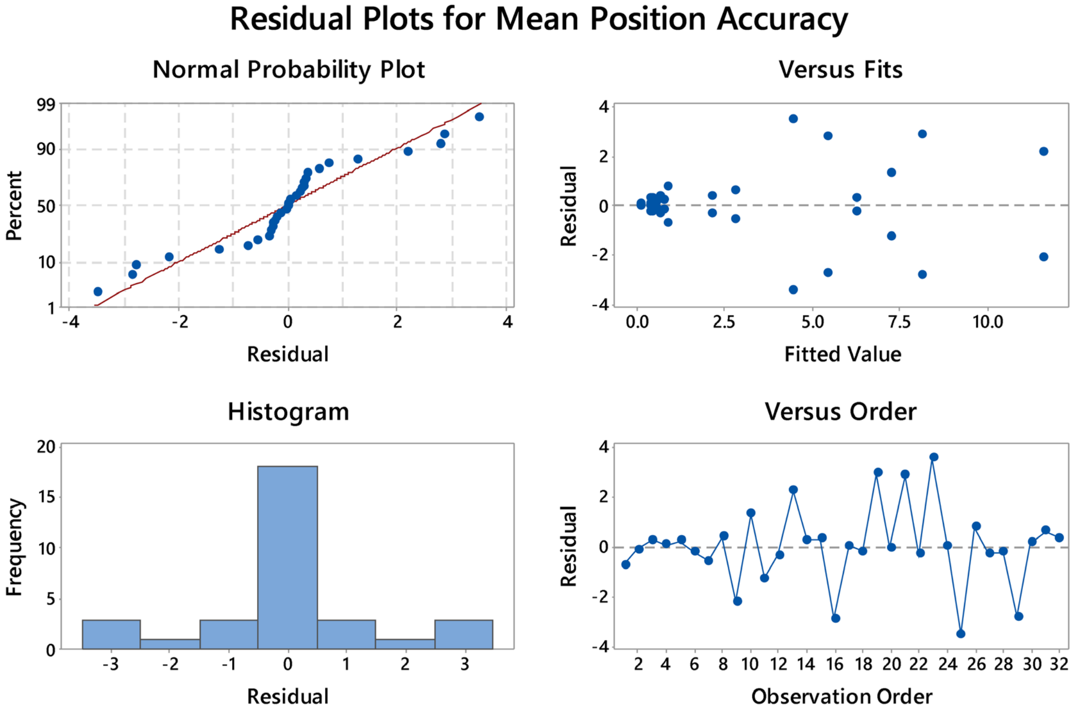


FIGURE 30 Residual plots for response "Mean Position Accuracy"

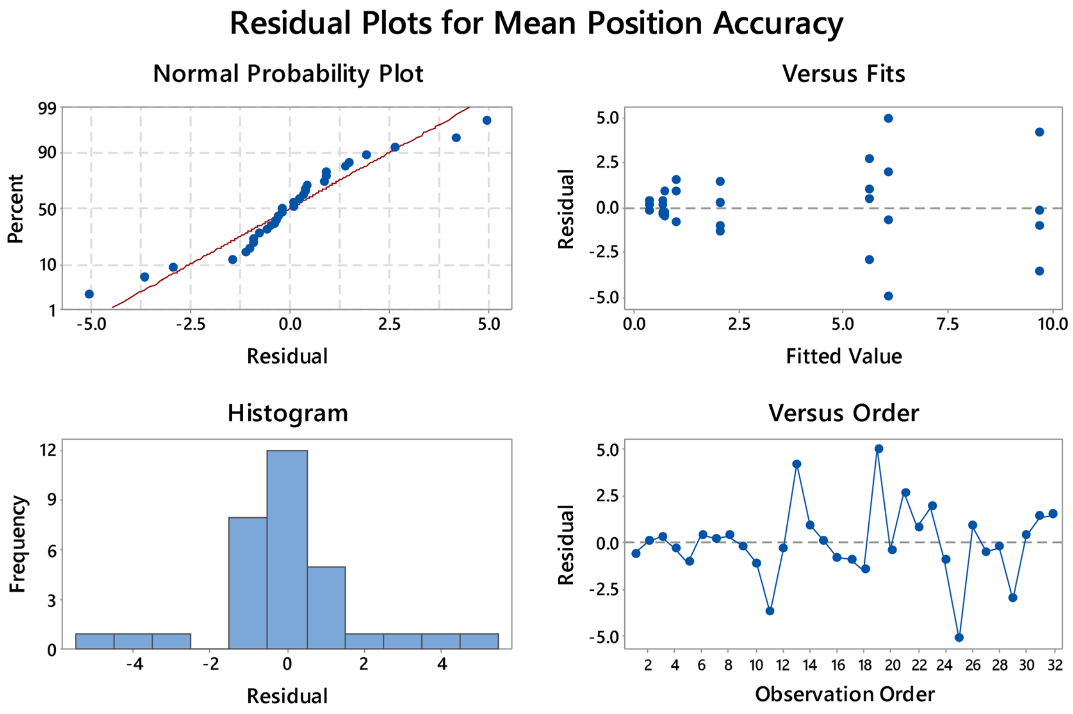
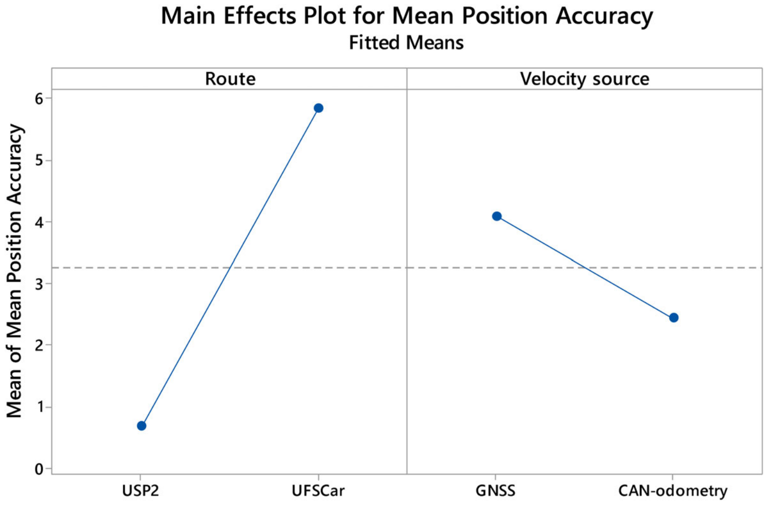


FIGURE 31 Residual plots for response "Mean Position Accuracy" after model reduction

FIGURE 32 Main effects plot for response: "Mean Position Accuracy"



than CAN-odometry velocity, whereas in the range of 18-20 SVs above horizon the obtained averages of mean position accuracy values with both velocity sources are similar. As a result, the average for GNSS velocity decreases and the average for CAN-odometry velocity increases. In fact, the highest number of SVs above horizon improves the accuracy of PVT solution of the GNSS, consequently, the velocity accuracy estimation, which impacts in the position accuracy of AHRS. However, there is not an interpretation for the reduction of the position accuracy performance, when the CAN-odometry velocity is combined with SVs above horizon in the range of 18-20.

The right interaction of Figure 33 shows the effect of the routes when combined with the velocity sources. In USP2 route, the obtained averages of mean position accuracy values with both velocity sources are similar, whereas in UFSCar route the average increases for both velocities but at an higher rate for GNSS velocity. In fact, the highest occlusion level of the UFSCar route also negatively impacts the velocity component of PVT solution of the GNSS, whereas the CAN-odometry velocity accuracy is not affected by occlusions due to the nature of its sensors. The GNSS velocity increases the mean position accuracy value, on average, by 3.595 m more than CAN-odometry velocity in the UFSCar route. Each vertex of the cube represents one combination of factor settings

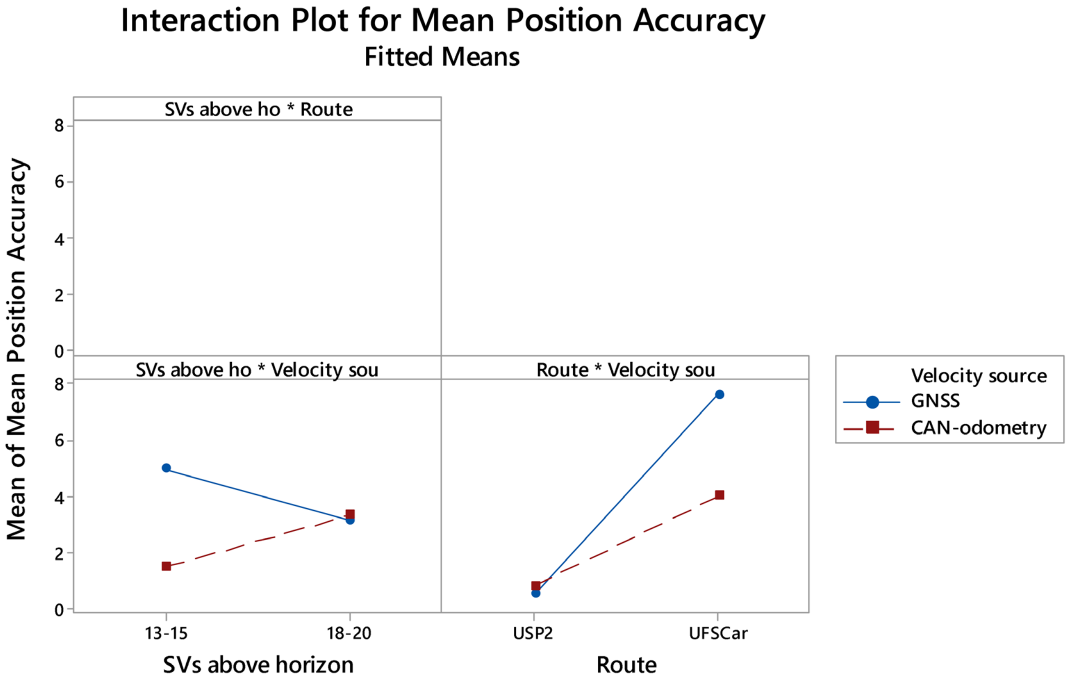


FIGURE 33 Interaction plot for response: "Mean Position Accuracy"

Although only the significant predictors have been kept in the reduced model, it still contains terms of 2-way and 3-way interactions that makes hard the synthesization of the effects during model interpretation. The regression equation in coded units (Equation (7)) shows the synthesization of the effects of each term in the form of coefficients. This equation, generated by Minitab 18 during analysis of factorial design, was used to make a prediction of "Mean Position Accuracy" response for all possible combinations of factor settings of the reduced model.

$$\begin{aligned}\text{Mean Position Accuracy} = {} & 3.264 + 2.585\ \text{Route} - 0.825\ \text{Velocity source} \\ & + 0.925\ \text{SVs above horizon*Velocity source} - 0.972\ \text{Route*Velocity source} \\ & + 1.105\ \text{SVs above horizon*Route*Velocity source.}\end{aligned} \tag{7}$$

Table 16 shows the result of the prediction. As reference station factor is not present in the significant terms, it is also not present in settings of the prediction. The values fitted for the response were ranked by quality of the predicted mean accuracy. The position of each combination of variables in the ranking is shown in the last column of Table 16. The high values of the standard error of the fit (SE fit) indicate a low accuracy of the prediction, which demands a careful interpretation of predictions. The USP2 route in the settings of top four ranking and the fit difference to the remaining

TABLE 16 Prediction for mean position accuracy

	Settings			Prediction		
Item	SVs above horizon	Route	Velocity source	Fit (m)	SE fit	Accuracy ranking
1	13-15	USP2	GNSS	0.35	0.916	1
2	13-15	USP2	CAN-odometry	1.01	0.916	4
3	13-15	UFSCar	GNSS	9.68	0.916	8
4	13-15	UFSCar	CAN-odometry	2.02	0.916	5
5	18-20	USP2	GNSS	0.71	0.916	3
6	18-20	USP2	CAN-odometry	0.65	0.916	2
7	18-20	UFSCar	GNSS	5.62	0.916	6
8	18-20	UFSCar	CAN-odometry	6.08	0.916	7

settings confirm the occlusion effect. The difference between fits of items 3 and 4 indicates that CAN-odometry overcomes GNSS under unfavorable condition of SVs in view, that is, low level of SVs above horizon and high occlusion level. Due to the value of SE fit, the difference between the top four ranking fits does not allow the researches to assert which velocity source is better under favorable conditions of SVs in view but it provides a hypothesis that GNSS can overcome the CAN-odometry under high levels of SVs above horizon and low occlusion levels.

The magnitude of the effect of Route factor (C) may have hidden the effect of Reference station factor (B), as well as the main effect of SVs above horizon factor (A). This is due to the found effect of C which is in the range of meters, whereas the effect of B may result in centimeter variations only. Due to this, a factorial analysis using the minimal position accuracy values was carried out with the purpose of finding these effects without the need to perform a new full experiment. The idea was finding the best possible accuracy allowed by the other three factors based on the fact that the two routes have path excerpts with partially open sky which are less affected by occlusion. Subsection 5.2 presents the results of this second analysis.

5.2 | Response: Minimal position accuracy

5.2.1 | Step 1: Determination of significant terms

Table 17 shows the ANOVA table associated with response: "Minimal Position Accuracy." As shown in the ANOVA table, the *P* values for linear terms "Reference station" (B) and "Velocity source" (D), as well as 2-way interactions "SVs above horizon" * "Reference station" (AB) and "SVs above horizon" * "Velocity source" (AD) are significant since the *P* values are less than the significance level of .05.

TABLE 17 ANOVA table for response "Minimal Position Accuracy"

Source	DF	Adj SS	Adj MS	*F* value	*P* value
Model	15	0.304147	0.020276	7.12	0.000
Linear	4	0.226934	0.056734	19.92	0.000
A	1	0.012152	0.012152	4.27	0.055
B	1	0.188897	0.188897	66.32	0.000
C	1	0.000018	0.000018	0.01	0.938
D	1	0.025867	0.025867	9.08	0.008
2-way interactions	6	0.058834	0.009806	3.44	0.022
AB	1	0.018586	0.018586	6.52	0.021
AC	1	0.000004	0.000004	0.00	0.970
AD	1	0.023263	0.023263	8.17	0.011
BC	1	0.006149	0.006149	2.16	0.161
BD	1	0.004836	0.004836	1.70	0.211
CD	1	0.005995	0.005995	2.10	0.166
3-way interactions	4	0.018311	0.004578	1.61	0.221
ABC	1	0.007170	0.007170	2.52	0.132
ABD	1	0.006061	0.006061	2.13	0.164
ACD	1	0.005075	0.005075	1.78	0.201
BCD	1	0.000004	0.000004	0.00	0.970
4-way interactions	1	0.000068	0.000068	0.02	0.879
ABCD	1	0.000068	0.000068	0.02	0.879
Error	16	0.045575	0.002848		
Total	31	0.349722			

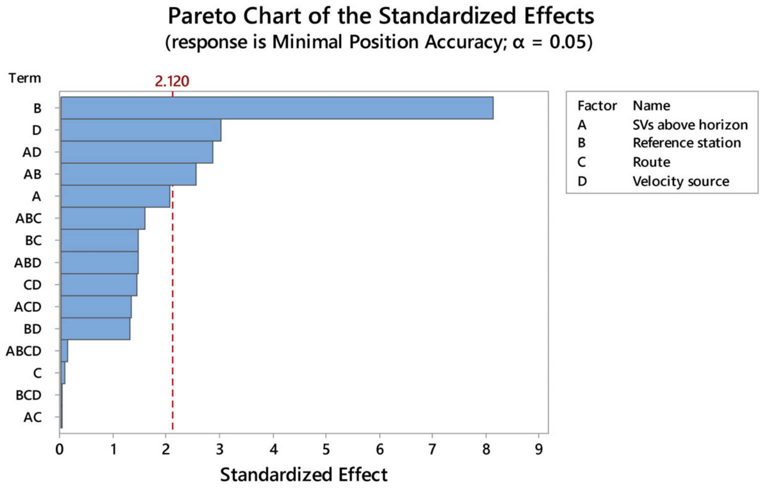


FIGURE 34 Pareto chart of the standardized effects for response "Minimal Position Accuracy"

TABLE 18 ANOVA table for response "Minimal Position Accuracy" after model reduction

Source	DF	Adj SS	Adj MS	*F* value	*P* value
Model	4	0.25661	0.064153	18.60	0.000
Linear	2	0.21476	0.107382	31.14	0.000
B	1	0.18890	0.188897	54.78	0.000
D	1	0.02587	0.025867	7.50	0.011
2-way interactions	2	0.04185	0.020925	6.07	0.007
AB	1	0.01859	0.018586	5.39	0.028
AD	1	0.02326	0.023263	6.75	0.015
Error	27	0.09311	0.003448		
Lack-of-fit	11	0.04753	0.004321	1.52	0.218
Pure error	16	0.04557	0.002848		
Total	31	0.34972			

Figure 34 shows Pareto chart of the standardized effects for response "Minimal Position Accuracy." The red line threshold in the Pareto chart indicates the reference *T* value for a *P* value of .05. The effects that surpass this red line are statistically significant. As can be seen, the linear term Reference station (B) became the effect with greater impact, whereas the occlusion effect of Route (C) factor was suppressed by picking the best possible position accuracy across each trajectory. The main effect of "D" continued to be significant at threshold of minimal value for position accuracy. The expected significant contribution of main effect "A" did not appear with a significance level of .05 but its obtained *P* value indicates that the effect may become significant in an analysis with a significance level of .10.

5.2.2 | Step 2: Model reduction

A model reduction was also carried out for response "Minimal Position Accuracy" by following the same procedure applied for response "Mean Position Accuracy." Table 18 shows the ANOVA table after model reduction. The significant terms remained significant after model reduction, although with minor variations in their *P* values.

Figure 35 shows the Pareto chart after model reduction. The order of the significant terms remains the same.

Table 19 shows comparison of model statistics before and after reduction. The increase of *S* indicates that the original model fits slightly better the response "Minimal Position Accuracy." According to the difference in R^2, the percentage of variation in the response explained by the reduced model is smaller, partially due to the reduction of predictors. As previously explained, since the two models have a different number of predictors, the adjusted R^2 provides a more appropriate evaluation of the model improvement. Therefore, the decrease of the adjusted R^2 indicates a degradation in the reduced model. In the original model the substantially less value of predicted R^2 than R^2 indicates that the first model may be

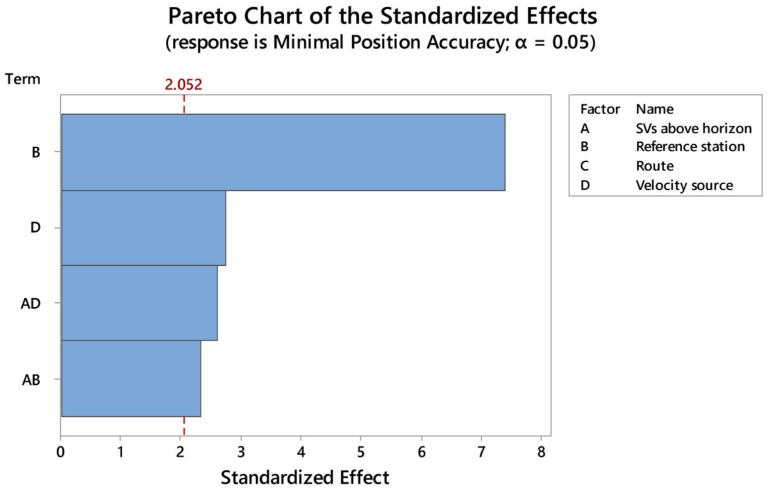


FIGURE 35 Pareto chart of the standardized effects for response "Minimal Position Accuracy" after model reduction

TABLE 19 Model summaries for response "Minimal Position Accuracy"

	Statistics			
Model	***S***	***R*²**	***R*² (adjusted)**	***R*² (predicted)**
Original	0.0533707	86.97%	74.75%	47.87%
Reduced	0.0587237	73.38%	69.43%	62.60%

over-fit. Lastly, the largest value of predicted R^2 of the reduced model, as compared to the original model, indicates the reduction of the over-fit behavior but not necessarily an improvement due to the decrease of adjusted R^2.

5.2.3 | Step 3: Validation of model assumptions

Figures 36 and 37 show residuals plots for response "Minimal Position Accuracy" before and after model reduction, respectively. The histogram (bottom left plot) of Figure 36 does not indicate skewness but shows the presence of outliers. The histogram (bottom left plot) of Figure 37 suggests a little skewness, and also shows the presence of outliers. The normal probability plot (top left plot) of both models shows a nonlinear pattern, since this pattern indicates a distribution with thin tails. This means that outliers are produced in both sides of the distribution. The residuals versus fits plot (top right plot) of both models indicates the presence of outliers. The residuals versus order plot (bottom right plot) of both figures shows no trends or patterns in the points that indicate correlation between near residuals in time order. In this plot, it is possible to identify large residual observations diagnosed by Minitab 18 (observations 03, 05, 18, and 28 in Figure 36, and observations 10, 18, and 28 in Figure 37).

Since both models did not meet the assumption that the residuals are normally distributed the obtained models may not fit well the data.

5.2.4 | Step 4: Interpretation of results

As the obtained model does not meet all the factorial analysis assumptions, the magnitude of the effects can be inaccurate. However, the results analysis provide a good insight about factors influence in the positioning accuracy of INS/GNSS navigation system without occlusion presence, as well as about factors interaction. Figure 38 shows the main effects that are statistically significant for response "Minimal Position Accuracy." As can be seen in the plot, the SPJA1 station increases the minimal position accuracy value, on average, by 0.154 m. This effect is associated with the distance of SPJA1 station to the data collection points, which is larger than that of EESC0 station, which is very close to both data collection routes. In fact, differential corrections from closer stations better reflect local atmospheric-related errors. The second main effect in the plot indicates that the CAN-odometry reduces the mean position accuracy value, on average, by 0.057 m. As there are significant interactions, the interpretation of the results should also consider their impact.

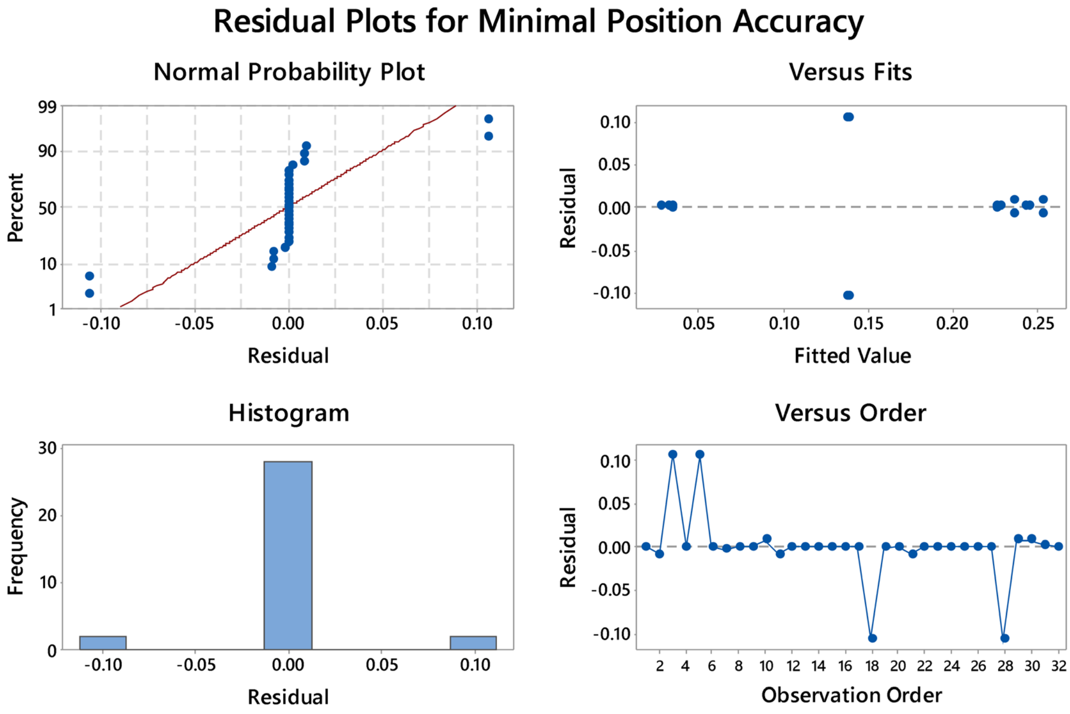


FIGURE 36 Residual plots for response “Minimal Position Accuracy”

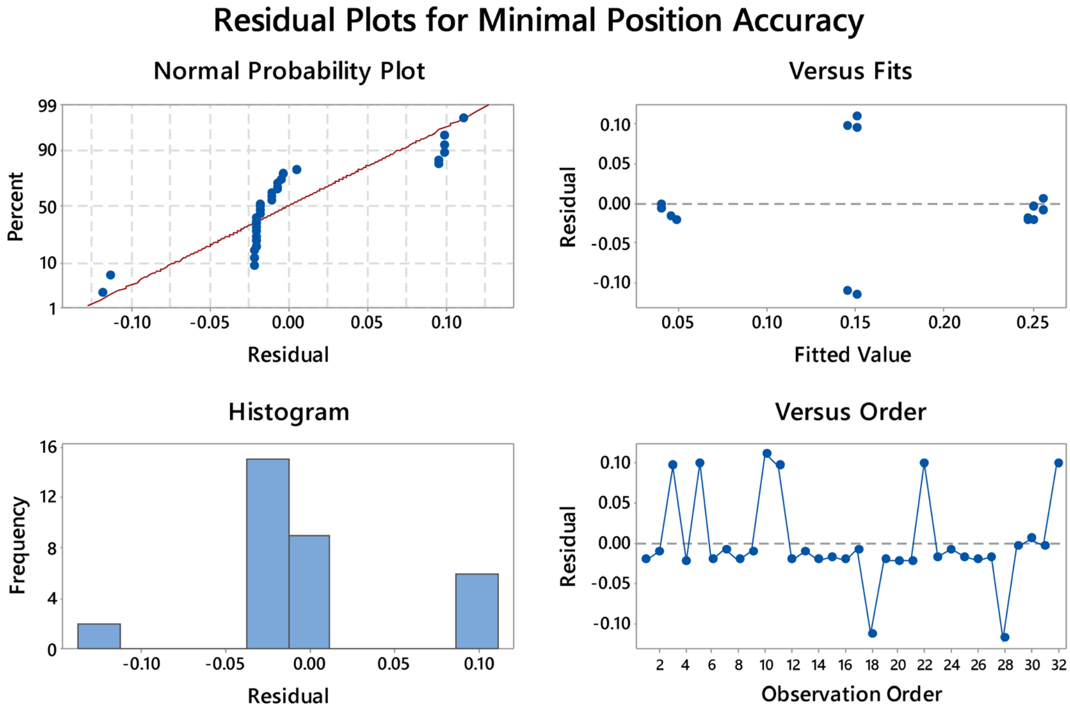
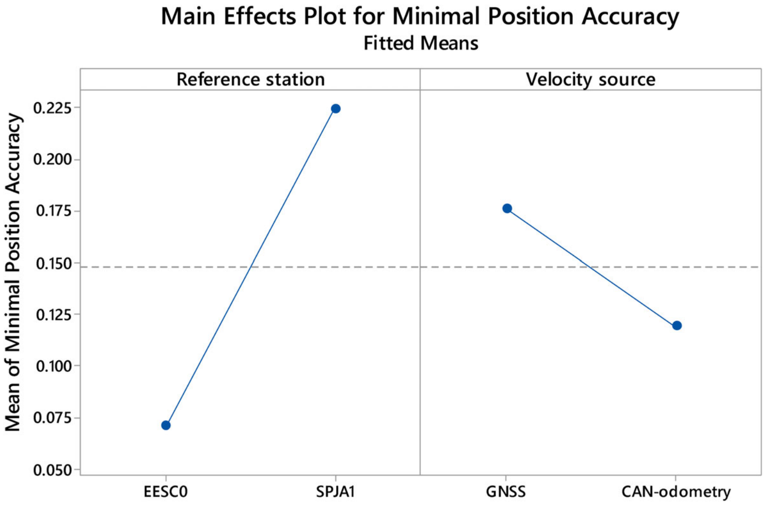


FIGURE 37 Residual plots for response “Minimal Position Accuracy” after model reduction

FIGURE 38 Main effects plot for response: "Minimal Position Accuracy"



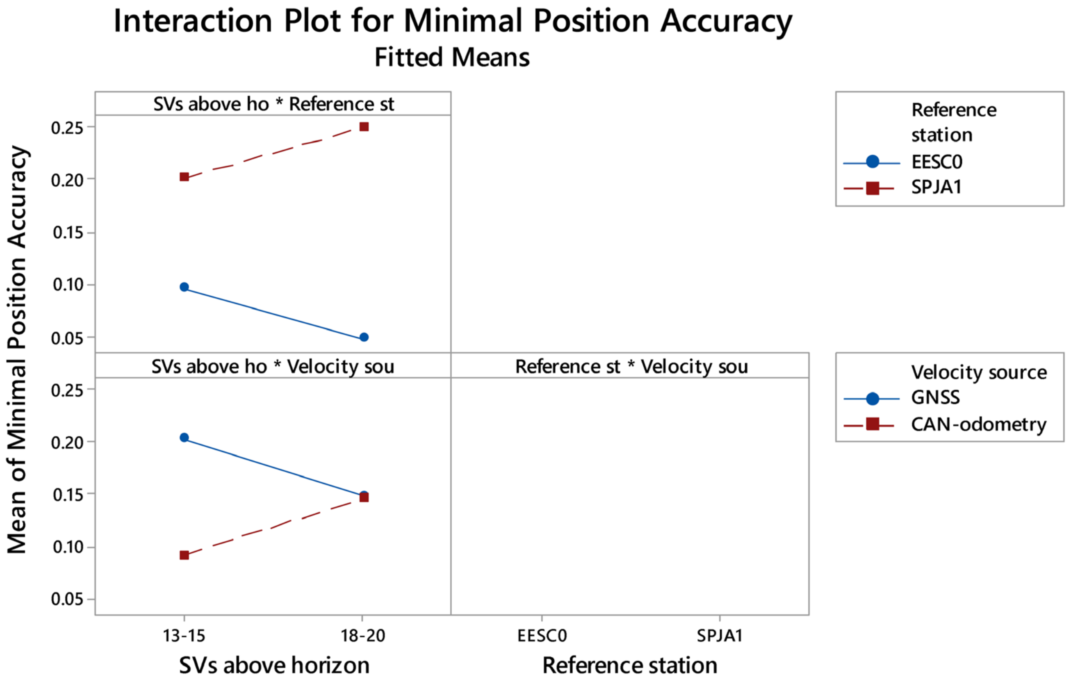


FIGURE 39 Interaction plot for response: "Minimal Position Accuracy"

Figure 39 shows the 2-way interactions that are statistically significant for response "Minimal Position Accuracy." The top interaction of Figure 39 shows the effect of the SVs above horizon when combined with the reference stations. The SPJA1 station increases the minimal position accuracy value, on average, by 0.105 m more than EESC0 station in the range of 13-15 SVs above horizon, whereas in the range of 18-20 SVs above horizon the average for EESC0 station decreases and the average for SPJA1 station increases resulting in a difference, on average, of 0.202 m. This result indicates that the EESC0 station produces a better performance under more SVs above horizon. However, so far there is not an interpretation for the position accuracy performance reduction with the SPJA1 station in the range of 18-20 SVs above horizon.

The bottom interaction of Figure 39 shows the effect of the SVs above horizon when combined with the velocity sources. The GNSS velocity increases the minimal position accuracy value, on average, by 0.111 m more than CAN-odometry velocity in the range of 13-15 SVs above horizon, whereas in the range of 18-20 SVs above horizon the obtained averages of minimal position accuracy values with both velocity sources are similar. Therefore, the average for GNSS velocity decreases and the average for CAN-odometry velocity increases. This interaction repeats the behavior of the same 2-way interaction of the first response analysis (see Subsection 5.1.4), except that now this interaction results

TABLE 20 Prediction for minimal position accuracy

	Settings			Prediction		
Item	SVs above horizon	Reference station	Velocity source	Fit (m)	SE fit	Accuracy ranking
1	13-15	EESC0	GNSS	0.150	0.0232	5
2	13-15	EESC0	CAN-odometry	0.040	0.0232	1
3	13-15	SPJA1	GNSS	0.256	0.0232	8
4	13-15	SPJA1	CAN-odometry	0.145	0.0232	4
5	18-20	EESC0	GNSS	0.048	0.0232	3
6	18-20	EESC0	CAN-odometry	0.045	0.0232	2
7	18-20	SPJA1	GNSS	0.250	0.0232	7
8	18-20	SPJA1	CAN-odometry	0.247	0.0232	6

in a variation of some centimeters instead of meters. The interpretation of this interaction is the same applied to the first response analysis (see Subsection 5.1.4).

Although only the significant predictors have been kept in the reduced model, it still contains terms of 2-way interactions that makes hard the synthesization of the effects during model interpretation again. The regression equation in coded units (Equation (8)) shows the synthesization of the effects of each term in the form of coefficients. This equation, generated by Minitab 18 during the analysis of factorial design, was used to make a prediction of Minimal Position Accuracy response for all possible combinations of factor settings of reduced model.

$$\begin{aligned}\text{Minimal Position Accuracy} =\ & 0.1477 + 0.0768\ \text{ Reference station} - 0.0284\ \text{ Velocity source} \\ & + 0.0241\ \text{ SVs above horizon*Reference station} \\ & + 0.0270\ \text{ SVs above horizon*Velocity source.}\end{aligned} \tag{8}$$

Table 20 shows the result of the prediction. As Route factor became absent in the significant terms, it is also not present in settings of this prediction. The values fitted for the response were ranked by quality of the predicted mean accuracy. The position of each combination of variables in the ranking is shown in the last column of Table 20. The high values of the standard error of the fit (SE fit) indicate a low accuracy of the prediction, and it demands a careful interpretation of predictions again. The EESC0 station in the settings of top three ranking confirms the station distance effect. The difference between fits of items 3 and 4 indicates that CAN-odometry can overcomes GNSS under unfavorable GNSS inputs, that is, low level of SVs above horizon and differential corrections from a farther reference station. Due to the value of SE fit, the difference between the top three ranking fits does not allow to assert which velocity source is better under favorable GNSS inputs but it provides a hypothesis that CAN-odometry can overcome the GNSS velocity even under high levels of SVs above horizon and differential corrections from a closer station.

5.3 | Discussion

The performance evaluation of this INS/GNSS system without using any multi-factor approach, such as 2^k factorial design, could lead to wrong conclusions about viability of the odometry proposal. Relying on the values shown in Table 12, it can be stated that the system presented a performance similar to fully integrated solutions which are currently commercialized. This statement is valid, considering an ideal surrounding environment, and assuming that the error estimate of the implemented INS/GNSS navigation system is close to realistic error.

Regarding the proposed experimental design, it is possible to obtain a better model fit by adding more replicates and repeats. This action increases the statistical power of the experimental design, consequently, the ability to detect minor effect sizes. However, due to the increase in the number of experimental runs, this also results in an higher experiment cost. Besides, this procedure does not cut out the outliers stemmed from extraneous factors or from factors that are difficult to control. Another alternative to refine the model is to add one more factor with known influence, but this will double the number of experimental runs.

For these reasons, it is better to reduce the experimental error by solving the control issues of experimental conditions. As mentioned in Subsection 4.1.1, the control of Factor A is improved by adopting the DOP as metric instead of the number of SVs above horizon, as well as by removing the unhealthy satellites from the prediction. Another way to better control Factor A is to reduce the extension of routes.

On the other hand, the effects of occlusion are difficult to control due to their time dependence on the satellite constellation geometry. In addition, the occlusion impact on PVT solution accuracy is satellite-specific. Thus, occlusions blocking the same satellites being tracked by the reference station directly affect DGNSS/RTK solution. In addition, due to the effect size of occlusion, which is much larger than that of the reference station, it becomes difficult to evaluate in-motion the reference station effects in high-occlusion environments, when carrying out at the same experimental design. This issue can be solved by using only low-occluded route segments.

The CAN-odometry uses low-resolution ABS encoders, as compared to current options of external encoders. However, due to the fact that encoder-based velocities are less affected by the trajectory dynamics and that are not affected by the occlusions, they always provide a velocity value with a smooth transition. Besides, the proposed ROS-based odometry provides the velocity value equal to zero during vehicle stop times. Therefore, these features make the average performance of CAN-odometry to overcome that of GNSS/RTK velocity during some factor combinations.

6 | CONCLUSION AND FUTURE WORKS

This work presented a framework called CAN-odometry, a low-cost odometry based on CAN protocol messages and flow control pins of RS232 standard. The proposed framework was integrated with an INS/GNSS system able to operate in RTK fixed mode with the aid of a local area DGNSS system connected to a Ntrip broadcaster. In order to evaluate the performance of the integrated system, a kinematic survey was planned into a 2^k factorial design and carried out under different controlled conditions. In order to avoid miscalculations, the obtained results were analyzed using the Minitab 18 software tools. The interpretation of results demonstrated that CAN-odometry can overcome the GNSS estimation when used as velocity source input to INS/GNSS system. It also provides an insight into magnitude of impacts of studied effects on integrated system performance, whereas sky obstructions was the factor of greater impact. Furthermore, by running the INS/GNSS system output on ROS framework, the experimental platform can read position and attitude values by high-level functions. Finally, the implemented integrated navigation system is a high-accuracy positioning and attitude system but is still limited by the satellite blockages.

Some improvements were proposed during the development of this work to enhance the CAN-odometry and the INS/GNSS navigation system accuracy. The first proposal is to develop a calibration model based on velocity values of the GNSS/RTK fixed mode, in order to mitigate the CAN-odometry low-accuracy at slow velocities. Another improvement is to modify the current SENA vehicle roof rack, or design a new one, in order to increment the distance between the GNSS main antenna and the aux1 antenna. This allows to obtain a better GNSS heading accuracy and availability.

In the robotics applications scope, the presented INS/GNSS navigation system will be integrated and synchronized with the currently installed perception sensors, such as LiDARs, and cameras. This task will include sensors extrinsic calibration and additional ROS nodes development. In addition, the potential use of the integrated navigation system in rural zones stemmed from the current project purpose. Since, under the current scenario, the cell phone signal is quite weak in these areas due to distance to the cell towers, it is proposed to install a cellular quad-band antenna to increase the modem reception signal strength.

The software architecture of CAN-odometry allows to be reused to emulate other odometry types. This could allow to transform odometry algorithms outputs in RTS pulses that can be sent to the AHRS device. Thus, one can transform the visual odometry in an odometer input of the AHRS. This is also a proposal for future performance evaluation using DOE approach.

ACKNOWLEDGMENTS

This work was financially supported by CAPES award numbers DS00011/07-0 and PNPD20131735. This project was supported by FAFQ, CNPq, FAPESP, CePOF-INOF, EESC-USP, Embrapa, and Fiat. We gratefully acknowledge the technical support of Septentrio NV/SA and SBG Systems. We would also like to gratefully acknowledge the access to Ntrip services granted by IBGE and BKG. We would like to give special thanks to Mateus Valverde Gasparino for his help in data collection.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

AUTHOR CONTRIBUTIONS

Jorge P. M. Galdames, Conceptualization-Lead, Data curation-Equal, Formal analysis-Equal, Investigation-Lead, Methodology-Equal, Project administration-Supporting, Software-Lead, Validation-Equal, Visualization-Lead, Writing-original draft-Lead, Writing-review & editing-Equal. Carlos E. Milhor, Conceptualization-Supporting, Investigation-Supporting, Project administration-Supporting, Software-Supporting, Visualization-Supporting, Writing-original draft-Supporting. Daniel V. Magalhães, Conceptualization-Supporting, Funding acquisition-Supporting, Resources-Supporting, Supervision-Supporting. Glauco A. P. Caurin, Conceptualization-Supporting, Funding acquisition-Supporting, Resources-Supporting, Supervision-Supporting. Marcelo Becker, Conceptualization-Supporting, Funding acquisition-Lead, Project administration-Lead, Resources-Lead, Supervision-Lead, Writing-original draft-Supporting.

ORCID

Jorge P. M. Galdames https://orcid.org/0000-0002-0742-5891
Carlos E. Milhor https://orcid.org/0000-0003-0393-1276
Daniel V. Magalhães https://orcid.org/0000-0002-1850-3103
Glauco A. P. Caurin https://orcid.org/0000-0003-0898-1379
Marcelo Becker https://orcid.org/0000-0002-7508-5817

REFERENCES

1. Rogers RM. *Applied Mathematics in Integrated Navigation Systems*. 2nd ed. Reston, VA: American Institute of Aeronautics and Astronautics; 2003.
2. Abbott E, Powell D. Land-vehicle navigation using GPS. *Proc IEEE*. 1999;87(1):145-162. https://doi.org/10.1109/5.736347.
3. El-Rabbany A. *Introduction to GPS: The Global Positioning System*. 2nd ed. Boston, MA: Artech House; 2006.
4. Noureldin A, Karamat TB, Eberts MD, El-Shafie A. Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications. *IEEE Trans Veh Technol*. 2009;58(3):1077-1096. https://doi.org/10.1109/TVT.2008.926076.
5. Soloviev A, Miller MM. Navigation in difficult environments: multi-sensor fusion techniques. In: Boginski V, Commander C, Pardalos P, Ye Y, eds. *Sensors: Theory, Algorithms, and Applications. 61 of Springer Optimization and Its Applications*. New York, NY: Springer; 2012:199-229. https://doi.org/10.1007/978-0-387-88619-0_9.
6. Skog I, Handel P. In-car positioning and navigation technologies–a survey. *IEEE Trans Intell Transp Syst*. 2009;10(1):4-21. https://doi.org/10.1109/TITS.2008.2011712.
7. Titterton DH, Weston JL. *Strapdown Inertial Navigation Technology*. 2nd ed. Reston, VA: American Institute of Aeronautics and Astronautics; 2004.
8. De Agostino M, Manzino AM, Piras M. Performances comparison of different MEMS-based IMUs. Paper presented at: IEEE/ION Position, Location and Navigation Symposium; May 04–06, 2010; Indian Wells, CA, 187–201. https://doi.org/10.1109/PLANS.2010.5507128.
9. Groves PD. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Boston, MA: Artech House; 2008.
10. Wendel J, Trommer GF. Tightly coupled GPS/INS integration for missile applications. *Aerosp Sci Technol*. 2004;8(7):627-634. https://doi.org/10.1016/j.ast.2004.07.003.
11. Schmidt GT, Phillips RE. INS/GPS integration architectures. Technical Report ADA581020, DTIC; 2010. https://apps.dtic.mil/docs/citations/ADA581020. Accessed January 24, 2019.
12. Kubo Y, Kindo T, Ito A, Sugimoto S. DGPS/INS/wheel sensor integration for high accuracy land-vehicle positioning. Paper presented at: Proceedings of the 12th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1999); September 14–17, 1999; Nashville, TN, 555–564. https://www.ion.org/publications/abstract.cfm?articleID=3202. Accessed September 23, 2019.
13. Stephen J. Development of a multi-sensor GNSS based vehicle navigation system [Master's thesis]. University of Calgary, Calgary, Canada, 2000. https://www.ucalgary.ca/engo_webdocs/GL/00.20140.JStephen.pdf. Accessed September 29, 2019.
14. Scherzinger BM. Precise robust positioning with Inertial/GPS RTK. Paper presented at: Proceedings of the 13th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 2000); September 19–22, 2000; Salt Lake City, Utah, 155–162. https://www.ion.org/publications/abstract.cfm?articleID=1399. Accessed September 23, 2019.
15. Carlson CR, Gerdes JC, Powell JD. Practical position and yaw rate estimation with gps and differential wheelspeeds. Paper presented at: Proceedings of AVEC '02, 6th International Symposium on Advanced Vehicle Control 2002; September 09–13, 2002; Hiroshima, Japan, 1–8. http://www-cdr.stanford.edu/dynamic/navigation/WheelSpeedsAVEC2002.pdf. Accessed May 29, 2019.
16. Hay C. Turn, turn, turn: whell-speed dead reckoning for vehicle navigation. *GPS World*. 2005;16(10):37-42. https://www.gpsworld.com/transportationroadturn-turn-turn-6513/.
17. Urmson C, Anhalt J, Bagnell D, et al. Autonomous driving in urban environments: boss and the urban challenge. *J Field Robotics*. 2008;25(8):425-466. https://doi.org/10.1002/rob.20255.
18. Ott JE. The OMNISTAR virtual base station system. Paper presented at: Proceedings of Position, Location and Navigation Symposium - PLANS '96; April 22–25, 1996; Atlanta, GA. https://doi.org/10.1109/PLANS.1996.509132.
19. Applanix. POS LV datasheet: design for integration, built for performance. 2017. Richmond Hill, Canada: Applanix. https://www.applanix.com/downloads/products/specs/POS_LV_Datasheet_web.pdf. Accessed January 21, 2018.

20. Hazlett AC, Crassidis JL, Fuglewicz DP, Miller P. Differential wheel speed sensor integration with gps/ins for land vehicle navigation. Paper presented at: AIAA Guidance, Navigation, and Control Conference; August 08–11, 2011; Portland, Oregon. https://doi.org/10.2514/6.2011-6577.
21. Guan H, Li L, Jia X. Multi-sensor fusion vehicle positioning based on Kalman Filter. Paper presented at: 2013 IEEE 3rd International Conference on Information Science and Technology (ICIST); March 23–25, 2013; Yangzhou, China. https://doi.org/10.1109/ICIST.2013.6747554.
22. Bevly DM, Ryu J, Gerdes JC. Integrating INS sensors with GPS measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness. *IEEE Trans Intell Transp Syst*. 2006;7(4):483-493. https://doi.org/10.1109/TITS.2006.883110.
23. Chiang KW, Duong TT, Liao JK. The performance analysis of a real-time integrated INS/GPS vehicle navigation system with abnormal GPS measurement elimination. *Sensors*. 2013;13(8):10599-10622. https://doi.org/10.3390/s130810599.
24. Liu Y, Liu F, Gao Y, Zhao L. Implementation and analysis of tightly coupled global navigation satellite system precise point positioning/inertial navigation system (GNSS PPP/INS) with insufficient satellites for land vehicle navigation. *Sensors*. 2018;18(12):1-20. https://doi.org/10.3390/s18124305.
25. Rahman MT, Karamat T, Givigi S, Noureldin A. Improving multisensor positioning of land vehicles with integrated visual odometry for next-generation self-driving cars. *J Adv Transp*. 2018;2018(6513970):1-12. https://doi.org/10.1155/2018/6513970.
26. Li T, Zhang H, Gao Z, Chen Q, Niu X. High-accuracy positioning in urban environments using single-frequency multi-GNSS RTK/MEMS-IMU integration. *Remote Sens*. 2018;10(2):1-21. https://doi.org/10.3390/rs10020205.
27. Kaplan ED, Hegarty CJ. *Understanding GPS: Principles and Applications*. 2nd ed. Boston, MA: Artech House; 2006.
28. Kalafus RM, Van Dierendonck AJ, Pealer NJ. Special committee 104 recommendations for differential GPS service navigations. *J Inst Navigat*. 1986;33(1):26-41. https://doi.org/10.1002/j.2161-4296.1986.tb00922.x.
29. Weber G, Dettmering D, Gebhard H. Networked transport of RTCM via internet protocol (NTRIP). In: Sansò F, ed. *A Window on the Future of Geodesy. 128 of International Association of Geodesy Symposia*. Berlin, Heidelberg: Springer; 2005:60-64. https://doi.org/10.1007/3-540-27432-4_11.
30. Cina A, Dabove P, Manzino AM, Piras M. Network real time kinematic (NRTK) positioning – description, architectures and performances. In: Jin S, ed. *Satellite Positioning - Methods, Models and Applications*. London, UK: IntechOpen; 2015:23-45. https://doi.org/10.5772/59083.
31. Wübbena G, Bagge A, Seeber G, Böder V, Hankemeier P. Reducing distance dependent errors for real-time precise DGPS applications by establishing reference station networks. Paper presented at: Proceedings of the 9th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1996); September 17–20, 1996; Kansas City, Missouri, 1845–1852. https://www.ion.org/publications/abstract.cfm?articleID=2722. Accessed September 29, 2019.
32. Landau H, Vollath U, Chen X. Virtual reference station systems. *J Global Position Syst*. 2002;1(2):137-143. https://www.scirp.org/pdf/nav20090200008_97580842.pdf.
33. Ouassou M, Jensen ABO. Network real-time kinematic data screening by means of multivariate statistical analysis. *SN Appl Sci*. 2019;1(6):512. https://doi.org/10.1007/s42452-019-0531-3.
34. Vollath U, Buecherl A, Landau H, Pagels C, Wagner B. Multi-Base RTK positioning using virtual reference stations. Paper presented at: Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000); September 19–22, 2000; Salt Lake City, Utah, 123–131. https://www.ion.org/publications/abstract.cfm?articleID=1396. Accessed September 23, 2019.
35. Fortes LP, Lachapelle G, Cannon ME, et al. Use of a multi-reference GPS station network for precise 3D positioning in constricted waterways. *Int Hydrogr Rev*. 2000;1(1):15-29. https://journals.hil.unb.ca/index.php/ihr/article/view/20480.
36. Rizos C, Han S, Chen HY. Regional-scale multiple reference stations for carrier phase-based GPS positioning: a correction generation algorithm. *Earth Planets Space*. 2000;56(10):795-800. https://doi.org/10.1186/BF03352284.
37. Fortes LP. Optimizing the use of GPS multi-reference stations for kinematic positioning. Paper presented at: Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002); September 24–27, 2002; Portland, ON, 2359–2372. https://www.ion.org/publications/abstract.cfm?articleID=2257. Accessed September 29, 2019.
38. Euler HJ, Keenan CR, Zebhause BE. Study of a simplified approach in utilizing information from permanent reference station arrays. Paper presented at: Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001); September 11–14, 2001; Salt Lake City, Utah, 379–391. https://www.ion.org/publications/abstract.cfm?articleID=1727. Accessed September 29, 2019.
39. Brown N, Keenan R, Richter B, Troyer L. Advances in ambiguity resolution for RTK applications using the new RTCM V3.0 master-auxiliary messages. Paper presented at: Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2005); September 13–16, 2005; Long Beach, CA, 73–80. https://www.ion.org/publications/abstract.cfm?articleID=6196. Accessed November 26, 2019.
40. Raquet JF. Development of a method for kinematic GPS carrier-phase ambiguity resolution using multiple reference receivers [Ph.D thesis]. University of Calgary, Calgary, CA, 1998. https://doi.org/10.11575/PRISM/14742.
41. Takac F, Zelzer O. The relationship between network RTK solutions MAC, VRS, PRS, FKP and i-MAX. Paper presented at: Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008); September 16–19, 2008; Savannah, GA, 348–355. https://www.ion.org/publications/abstract.cfm?articleID=7964. Accessed September 23, 2019.
42. Manzino A, Dabove P. Quality control of the NRTK positioning with mass-market receivers. In: Hsueh YH, ed. *Global Positioning Systems: Signal Structure, Applications and Sources of Error and Biases*. Systems Engineering Methods, Developments and Technology. Hauppauge, NY: Nova Science Publishers; 2013:17-40. https://novapublishers.com/wp-content/uploads/2019/07/978-1-62808-022-3_ch2.pdf.

43. Uradzinski M, Kim D, Langley RB. The usefulness of internet-based (NTrip) RTK for navigation and intelligent transportation systems. Paper presented at: Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008); September 16–19, 2008; Savannah, GA, 1437–1445. https://www.ion.org/publications/abstract.cfm?articleID=8056. Accessed September 23, 2019.
44. Uradzinski M, Kim D, Langley RB. Usefulness of the internet-based (NTrip) RTK for precise navigation. Warsaw, Poland: WAT; 2010; 59(2): 197–209. https://redakcjawydawnictw.wat.edu.pl/_Biuletyn/3_baza_artykulow_list.php?qs=Uradzinski.
45. Souza AN, Garnés SJA, Marques HA. Evaluation of the GNSS positioning obtained by method kinematic RTK/NTRIP and PPP in real time [in Portuguese]. *Revista Brasileira de Cartografia*. 2014;66(5):1117-1133. http://www.seer.ufu.br/index.php/revistabrasileiracartografia/article/view/44706.
46. IBGE. Brazilian network for continuous monitoring of the GNSS-RBMC systems - RBMC. 2019. https://www.ibge.gov.br/en/geosciences/geodetic-positioning/geodetic-networks/20079-brazilian-network-for-continuous-monitoring-of-the-gnss-systems-2.html?=&t=o-que-e. Accessed February 25, 2019.
47. Costa SMA, Lima MAA, De Moura Junior NJ, Abreu MA, Da Silva AL. Fortes LPS. RBMC em Tempo Real, via NTRIP, e seus benefícios nos levantamentos RTK e DGPS. Paper presented at: II Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação; September 08–11, 2008; Recife, Brazil, 1–8. https://www3.ufpe.br/cgtg/SIMGEOII_CD/Organizado/geo/202.pdf. Accessed February 25, 2019.
48. Galdames JPM. In-field citrus fruit counting from color image streams using deep learning based detection and random finite sets based robotic mapping [Ph.D thesis]. University of Sao Paulo, Sao Carlos, SP, Brazil; 2019. Final manuscript in preparation.
49. Hernandes AC, Brito AS, Roncancio H, et al. GISA: a Brazilian platform for autonomous cars trials. Paper presented at: 2013 IEEE International Conference on Industrial Technology (ICIT); February 25–28, 2013; Cape Town, South Africa, 82–87. https://doi.org/10.1109/ICIT.2013.6505652
50. Systems SBG. *IG-500E Aided Navigation & AHRS: User Manual*. Rueil-Malmaison, France: SBG Systems; 2012 Document: IG500EUM.3.
51. Septentrio NV. SA. RxTools. RxTools version: 16.4.1. 2017. https://www.septentrio.com/support/software/rxtools. Accessed June 19, 2017.
52. Septentrio NV/SA. *Firmware User Manual: Applicable to SSRC2 3.4.0*. Leuven, Belgium: Septentrio Satellite Navigation; 2014. https://www.septentrio.com/en/support/asterx/asterx2eh.
53. Septentrio NV/SA. *AsteRx2eH Command Line Interface Reference Guide: Applicable to version 3.4.0 of the GNSS Firmware*. Leuven, Belgium: Septentrio Satellite Navigation; 2014 https://www.septentrio.com/en/support/asterx/asterx2eh.
54. SBG Systems. *IG-500E Generic NMEA Integration Manual*. Rueil-Malmaison, France: SBG Systems; 2012 Document: IG500EINMEA.3.
55. BKG. *BKG Ntrip Client (BNC)*. Frankfurt, Germany: BKG; 2018. https://igs.bkg.bund.de/ntrip/bnc.
56. Weber G, Mervart L. *The BKG Ntrip Client (BNC)*. Frankfurt, Germany: BKG; 2016. https://www.researchgate.net/publication/267718077_The_BKG_Ntrip_Client_BNC.
57. Johnston G, Riddell A, Hausler G. The international GNSS service. In: Teunissen P, Montenbruck O, eds. *Springer Handbook of Global Navigation Satellite Systems*. Cham, Switzerland: Springer; 2017:967-982. https://doi.org/10.1007/978-3-319-42928-1_33.
58. IBGE. RBMC-IP - Rede Brasileira de Monitoramento Contínuo dos Sistemas GNSS em tempo real. 2017. https://ww2.ibge.gov.br/home/geociencias/geodesia/rbmc/ntrip/. Accessed April 05, 2017.
59. Romero AM. Wiki: ROS/Concepts. 2014. http://wiki.ros.org/ROS/Concepts. Accessed March 12, 2018.
60. Hansen K. Wiki. Packages. 2015. http://wiki.ros.org/Packages. Accessed March 12, 2018.
61. SBG Systems. *IG Devices Serial Protocol Specifications*. Rueil-Malmaison, France: SBG Systems; 2012 Document: IGDSPS.12.
62. IBGE. Estações da RBMC (arquivo KMZ). 2018. ftp://geoftp.ibge.gov.br/informacoes_sobre_posicionamento_geodesico/rbmc/cartogramas/RBMC_\2017.kmz. Accessed April 07, 2018.
63. NCEI. Magnetic declination calculator. 2018. https://www.ngdc.noaa.gov/geomag-web/#declination. Accessed March 28, 2018.
64. Chulliat A, Macmillan S, Alken P, et al. The US/UK World Magnetic Model for 2015-2020. Technical Report; 2015; National Geophysical Data Center, NOAA. https://www.ngdc.noaa.gov/geomag/WMM/data/WMM2015/WMM2015_Report.pdf. Accessed March 28, 2018.
65. NGA DGC. World Magnetic Model (WMM). 2018. https://www.ngdc.noaa.gov/geomag/WMM/. Accessed March 28, 2018.
66. Simon S. P. Mean: what is a normal probability plot? 2009. http://www.pmean.com/09/NormalPlot.html. Accessed July 23, 2018.
67. Montgomery DC. *Design and Analysis of Experiments*. 8th ed. Hoboken, NJ: John Wiley & Sons; 2013.
68. NIST. SEMATECH . NIST/SEMATECH e-Handbook of Statistical Methods. 2013. https://www.itl.nist.gov/div898/handbook/index.htm. Accessed July 23, 2018.
69. Richard NM. How to use minitab: design of experiments. 2014. https://web.wpi.edu/Pubs/ETD/Available/etd-082714-135653/unrestricted/How_to_Use_Minitab_4_Design_of_Experiments.pdf. Accessed August 02, 2018.
70. Minitab. Interpret the key results for analyze factorial design. 2018. https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/doe/how-to/factorial/analyze-factorial-design/interpret-the-results/key-results/. Accessed July 23, 2018.
71. Minitab. Interpret the key results for Factorial Plots. 2018. https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/using-fitted-models/how-to/factorial-plots/interpret-the-results/key-results/. Accessed July 25, 2018.
72. Minitab. Model reduction. 2018. https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/regression/supporting-topics/regression-models/model-reduction/. Accessed July 16, 2018.
73. Minitab. Model summary table for Analyze Factorial Design. 2018. https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/doe/how-to/factorial/analyze-factorial-design/interpret-the-results/all-statistics-and-graphs/model-summary-table/. Accessed July 19, 2018.

74. STATWING. Interpreting residual plots to improve your regression. 2018. http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression/. Accessed July 24, 2018.

75. Pan J. Minitab Tutorials for Design and Analysis of Experiments. 2018. https://web.calpoly.edu/~pan/teaching/Minitab%20DOE%20Tutorial.pdf. Accessed August 02, 2018.

76. PennState. STAT 501 | Regression Methods - Lesson 4: SLR Model Assumptions - 4.6 Normal Probability Plot of Residuals. 2019. https://newonlinecourses.science.psu.edu/stat501/lesson/4/4.6. Accessed September 29, 2019.

How to cite this article: Galdames JPM, Milhor CE, Magalhães DV, Caurin GAP, Becker M. Implementation and performance evaluation of an inertial navigation system/global navigation satellite system real-time kinematic Ntrip navigation system aided by a robot operating system-based emulated odometer for high-accuracy land vehicle navigation in urban environments. *Engineering Reports*. 2020;2:e12097. https://doi.org/10.1002/eng2.12097