2020 IMEKO TC-19 International Workshop on
Metrology for the Sea
Naples, Italy, October 5-7, 2020

Developing a low-cost GNSS/IMU data fusion platform for boat navigation

Matteo Cutugno[1], Giovanni Pugliano[1], Umberto Robustelli[1]

[1]*Department of Engineering, Parthenope University of Naples, 80133 Naples, Italy,*
(matteo.cutugno;umberto.robustelli;giovanni.pugliano)@uniparthenope.it

Abstract – **Methods of measuring a vessel's motioninvolve the use of expensive and complex Inertial Navigation Systems (INS). Cargo or passenger transport ships can afford the implementation of such systems while private small boat market has been cut off. The alternatives for small boat INS navigation are few. This paper investigates the potentiality of GNSS/IMU data fusion, experimenting low-cost hardware like Aceinna openIMU 300ZI and a GNSS receiver based on u-blox ZED-F9P module. The final goal is to build a low-cost self-powered system and assess the performance in small boats navigation tests. To address this goal the INS shall be light-weight, cost-effective, and easy-to-install.**

I. INTRODUCTION

Inertial Measurements Units (IMU) are made by sensors that comprise accelerometers and gyroscopes that use microcontrollers to analyze collected measurements. Nowadays, inertial navigation technology is used widely for aircraft, ships, and land vehicles motion tracking [1]. Inertial navigation is a self-sufficient navigation technique in which measurements provided by sensors are used to track the position, velocity, and attitude (orientation) of an object relative to a known start position, velocity, and attitude. IMU signal processing chain consists of high-speed sampling of the 9 degree-of-freedom (DOF) sensor cluster (accelerometers, angular-rate sensors, and magnetometers). Since an IMU doesn't rely on any external information sources that can be disturbed or jammed, it is an attractive means of navigation for many applications where 100% coverage and a high continuity-of-service is needed. Further, it also provides a full 9 degree of-freedom navigation solution and generally has a high update rate ($\geq$ 100 Hz). However, IMUs suffer from integration drift, which means that small errors in the accelerometers and gyroscopes accumulate over time and the position error will increase without bound.To overcome this problem, the navigation information provided by a low-cost IMU is frequently fused with the navigation information provided by a GNSS-receiver realizing a, so called, Inertial Navigation System (INS). On the other hand, a stand-alone GNSS receiver can be affected by signal outages due to the presence of buildings, vegetation, or jamming interference. An INS suitable for sea navigation has to provide a position accuracy as a GNSS-receiver and a full 6 degrees-of-freedom navigation solution at high update rate. Moreover, for short periods it can provide a suitable navigation solution even during GNSS signals outages [2]. This paper wants to investigate the potentiality of a low-cost INS navigation technique within the open-source software scenario. IMU/GNSS sensor integration relies on the fusion of estimate absolute positions from GNSS useful to correct the IMU sensor's drift. IMU/GNSS sensor integration strategies differ one from another depending on how deep the sensor fusion is realized. Uncoupled systems doesn't share any data providing two separate navigation solutions. In the loosely-coupled technique, the positions and velocities estimated by the GPS receiver are blended with the INS navigation solution, while in the case of tightly-coupled method, GPS raw measurements (i.e., pseudorange and Doppler observables) are processed through a unique Kalman filter with the measurements coming from the inertial sensors to estimate the PVT [3].

II. METHODOLOGY

INS navigation technique is well described in literature [4] and here we made just a brief recall of the key concepts. Remembering that an object position and orientation can only be measured referred to coordinate-frame, inertial sensors permit to calculate attitude and position of the object in a 3-D space related to an "inertial" frame, such as the Earth-Centered, Earth-Fixed frame (ECEF) [5]. The three attitude parameters that describe an object relative to the ECEF frame are direction cosine matrices, quaternion elements, and euler angles. The navigation solution obtained is based on a Kalman Filter (KF) that generates estimates of attitude, position, and velocity. The KF is an optimal linear estimator when the process noise and the measurement noise can be modeled by white Gaussian noise. In real-life situations, when the problems are nonlinear or the noise that distorts the signals is non-Gaussian, the Kalman filters provide a solution that may be far from optimal. Nonlinear problems can be solved with the extended Kalman Filter (EKF) [6]. KFs work on a predict/update cycle. The system state of the new time-step is predicted from current states plus system measurements, i.e. for attitude is the angular rate-sensor signal while for velocity and position is accelerometer signal. The update stage corrects the state estimates for errors in the measurement signals using measurements of the true values of position, velocity, and attitude.

III. EXPERIMENTAL SETUP

In this research, the author's goal is to test the feasibility and the performances of a low-cost light-weight inertial navigation system made for small boats motion tracking.

A. *Hardware setup*

The hardware used in the experiment consists of an Aceinna openIMU 300ZI eval kit, a GNSS receiver equipped with u-blox ZED-F9P module [7] [8] and a Raspberry PI3 [9]. The IMU's main features are shown in Table 1. In Figure 2 is shown the equipment tested.

Table 1. Aceinna openIMU 300ZI main features.

Integrated 3-Axis Angular Rate
Integrated 3-Axis Accelerometer
Integrated 3-Axis Magnetic Sensor
168MHz STM32 M4 CPU
SPI / UART Interfaces
Update Rate up to 800Hz
In-System Upgrade
Small Size (24x37x9.5mm)
Drop-in Upgrade for IMU380ZA, IMU381ZA
Wide Temp Range -40 to 85 ° C
High Reliability > 50,000hr MTBF
1kHz time pulse input

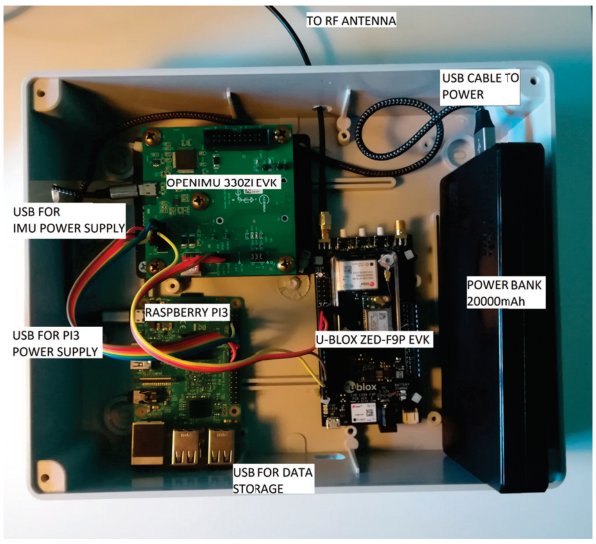

Fig. 1. Test equipment: Aceinna openIMU 300ZI EVK, Raspberry PI3, u-blox ZED-F9P EVK and a 20000 mAh Power bank

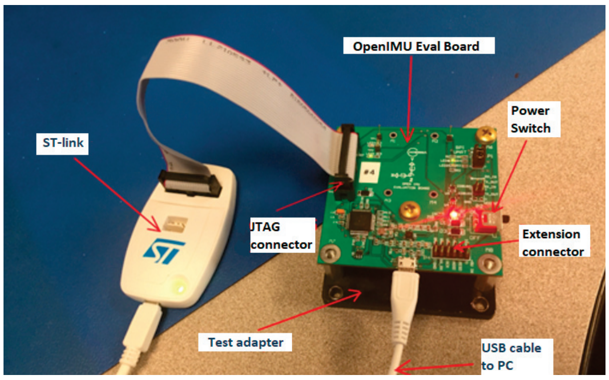

Fig. 2. Aceinna openIMU 300ZI evaluation kit (https://www.aceinna.com/).

The processing platform consist in a Raspberry PI3 shown in Figure 4. The INS system was realized providing NMEA messages to Aceinna openimu 300ZI EVK. The external GNSS receiver, as shown in Figure 3 is a u-blox receiver relying on ZED-F9P GNSS module. u-blox has been configured to interface with IMU on Serial Peripheral Interface (SPI). Also, Universal Asynchronous Receiver-Transmitter (UART) is available, but the choice has fallen on the former interface for data-rate speed reasons. Accelerometer, angular-rate sensor, and magnetometer of the IMU and GNSS sensor are used to obtain the information needed to estimate the position, velocity, and attitude. In particular, the position is obtained by the GNSS sensor while accelerometers provide the signal that is integrated to get velocity information. Then, GNSS provides velocity and supplemental information to the algorithm which is used to estimate the accelerometer bias. Roll and pitch are obtained by angular-rate sensor output where the errors due to integration are corrected with a gravity reference given by accelerometer. Lastly, heading is obtained combining data provided by three sensors: angular-rate sensor for the heading information, magnetometer for the North reference, and GNSS also provides more accurate low-rate heading information value.

B. *Software setup*

Referring to IMU, the software used is VisualStudio core with PlatformIO wich give the possibility to upload different ready-to-go applications provided by the manufacturer as well as build customized ones. To compile and upload in the Electrically Erasable Programmable Read-Only Memory (EEPROM) the INS app a JTAG compiler

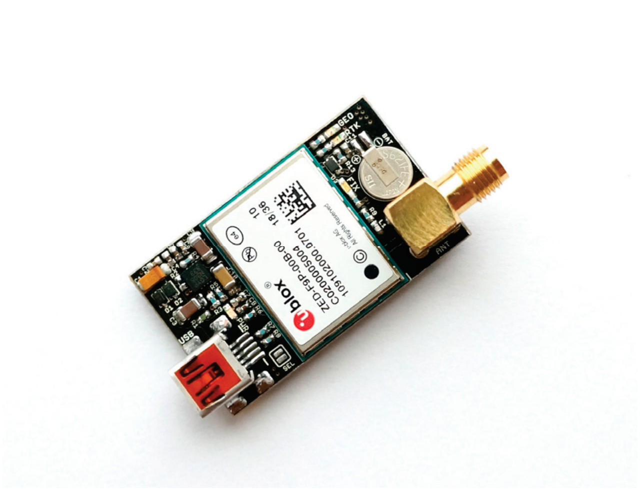

Fig. 3. u-blox ZED-F9P GNSS receiver.

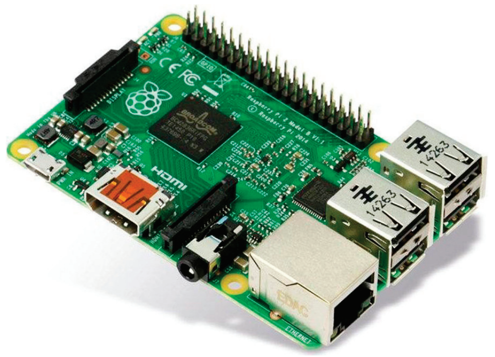

Fig. 4. Raspberry PI3.

has been provided as shown in Figure 2. In the INS app, a 16-state extended Kalman filter [9] [10] [11] [12] is implemented to process measurements from GPS receiver and IMU unit. For GNSS messaging protocol, the choice has fallen on NMEA protocol. In particular, *gpsBaudRate* parameter has been set to 115200, while *gpsProtocol* parameter to *NMEA_TEXT*. Finally, INS was configured to output estimated solutions at 100Hz.
On the GNSS receiver side, the software used to set up the u-blox to output proper navigation messages is U-center evaluation software version 20.01. Using *NMEA_TEXT* as protocol, u-blox receiver has been configured to output only NMEA messages making sure that other messages were disabled.

IV. PEDESTRIAN TEST

The first test carried out consisted in a pedestrian test in a highly degraded scenario in Centro Direzionale (Naples) [13] [14] where the navigation system was surrounded by tall glass skyscrapers resulting in strong multipath interference. NMEA were stored for post-mission processing in order to figure out difference between INS and stand-

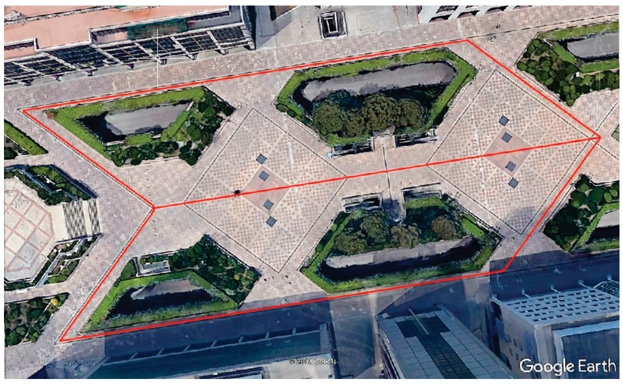

Fig. 5. Path trajectory (view from Google Earth).

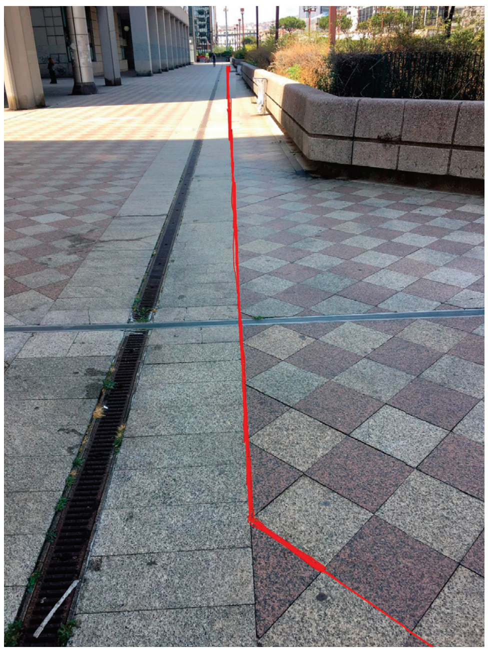

Fig. 6. Path detail: segment T3.

alone GNSS solution. The test was carried out along a precise path represented by different colors of the tiles of the paving well identifiable both from ground and satellite sight. Two laps were completed.

V. RESULTS AND CONCLUSIONS

Segment T1 and Segment T3 were the most difficult parts of the path because they are placed almost below the adjacent skyscrapers resulting in a very poor GNSS signal availability and quality. On the other hand the central Segment, i.e. Segment T2, represented the most open-sky condition during the test session. During the test session no GNSS signal leakage events happened so both navigation solutions (INS and GNSS) were available; in particular, INS estimated ∼65000 PVT solutions while u-blox ∼700. In Figure 7 are shown the errors (green lines) of the INS

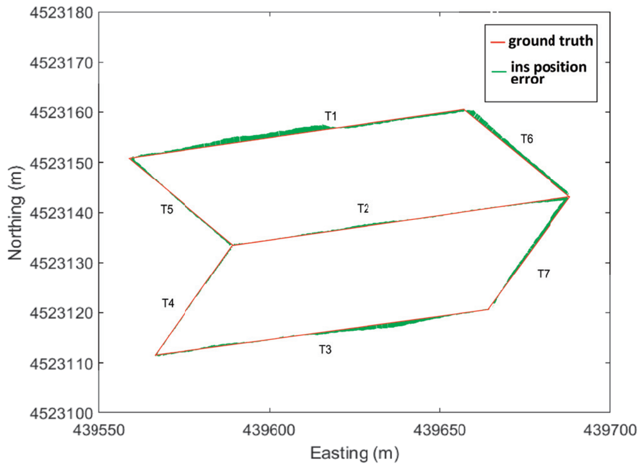

Fig. 7. First lap: true path (red line) and orthogonal distance errors of INS (green lines).

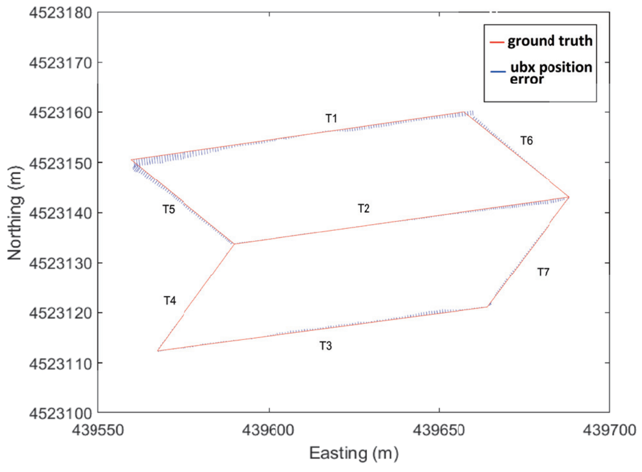

Fig. 9. First lap: true path (red line) and orthogonal distance errors of u-blox (blue lines).

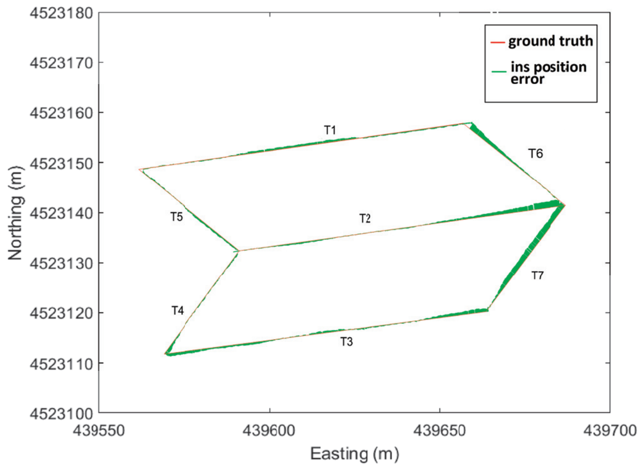

Fig. 8. Second lap: true path (red line) and orthogonal distance errors of INS (green lines).

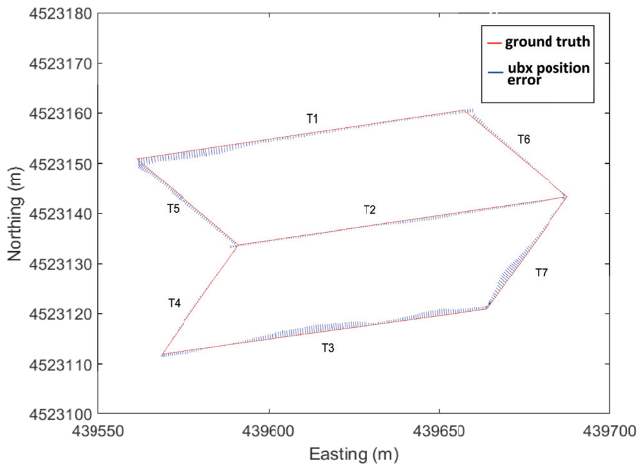

Fig. 10. Second lap: true path (red line) and orthogonal distance errors of u-blox (blue lines).

navigation solutions respect to true path (red lines) for the data acquired during first lap while in Figure 8 are shown the same values for second lap.
In Figure 9 are shown the errors of the positions obtained with stand-alone u-blox (blue lines) respect to true path for second lap while Figure 10 depicts the same values for second lap.
Moreover, Table 2 summarizes the mean values of orthogonal distance for INS. One can notice that the values of INS errors are always under 1 meter except for Second lap of Segment T7.
Lastly, Table 3 summarizes the mean of orthogonal distance for u-blox. Here, the mean values for u-blox errors are quite bigger than the corresponding values of INS.

VI. FUTURE WORKS

The next future work will be employing a Software Defined Receiver (SDR) instead of Commercial off-the-shelf (COTS) receivers. This approach can address improvements in terms of cost and customizability. Different kind of tests, i.e. pedestrian and marine, will be carried out in order to assess the performances of the system in terms of reliability and solution accuracy when stressed in a long time real-life situations.

VII. REFERENCES

VIII. *

References

[1] Noureldin, A., Karamat, T. B., Georgy, J. (2013). Fundamentals of inertial navigation, satellite-based posi-

Table 2. Mean values of the orthogonal distance between INS positions and true path for each Segment.

Segment	lap 1 (m)	lap 2 (m)
T1	0.8212	0.3044
T2	0.1906	0.5072
T3	0.4419	0.4234
T4	0.1358	0.1739
T5	0.3666	0.2729
T6	0.9037	0.5883
T7	0.3029	1.0977

Table 3. Mean values of the orthogonal distance between NMEA positions and true path for each Segment

Segment	lap 1 (m)	lap 2 (m)
T1	0.9875	1.0802
T2	0.3156	0.4669
T3	0.4977	1.1390
T4	0.1940	0.2279
T5	0.9656	0.9486
T6	0.4035	0.4785
T7	0.6377	1.1800

tioning and their integration (1st ed.). Berlin: Springer.

[2] Groves, P. D. (2015). Navigation using inertial sensors [Tutorial]. IEEE Areospace and Electronic Systems Magazine, 30(2), 42-69.

[3] Falco G, Pini M, Marucco G. Loose and Tight GNSS/INS Integrations: Comparison of Performance Assessed in Real Urban Scenarios. Sensors (Basel). 2017;17(2):255. Published 2017 Jan 29. doi:10.3390/s17020255.

[4] Mehdi, J. (2017). Quaternions Algebra and Its Applications: An Overview. International Journal of Theoretical and Applied Mathematics. 2. 79-85. 10.11648/j.ijtam.20160202.18.

[5] Salychev, O.S. Voronov, V.V. Cannon, M. E. Lachapelle, G. "Low cost INS/GPS integration: concepts and testing". Proceeding of the Institute of Navigation National Technical Meeting. 2000.

[6] Montella, C. (2011). The Kalman Filter and Related Algorithms: A Literature Review.

[7] Aceinna website. https://openimu.readthedocs.io/en/latest/. Accessed on 10/04/2020.

[8] u-blox website. https://www.u-blox.com/en/product/zed-f9p-module. Accessed on 12/04/2020

[9] Raspberry website. https://www.raspberrypi.org/documentation/. Accessed on 11/04/2020

[10] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. ASME. J. Basic Eng. March 1960; 82(1): 35â45. https://doi.org/10.1115/1.3662552

[11] N.J. Gordon D.J. Salmond A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation

[12] Simon J. Julier and Jeffrey K. Uhlmann "New extension of the Kalman filter to nonlinear systems", Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI, (28 July 1997); https://doi.org/10.1117/12.280797

[13] Cutugno, M. Robustelli, U. Pugliano, G. (2019). Testing a GNSS software receiver for end-user utilization.

[14] Cutugno, M. Robustelli, U. Pugliano, G. (2020). Low-Cost GNSS Software Receiver Performance Assessment. Geosciences 2020, 10, 79.