In: AIAA SciTech Forum 2023, 23-27 January 2023, National Harbor, Maryland, USA
DOI:10.2514/6.2023-2226

GNSS/INS/VO fusion using Gated Recurrent Unit in GNSS denied environments

Sorin Andrei Negru[1], Patrick Geragersian [2], Ivan Petrunin [3], Argyrios Zolotas[4]

Cranfield University, Bedfordshire, MK43 0AL, United Kingdom

Raphael Grech[5]

Spirent Communications PLC, West Sussex, RH10 1BD, United Kingdom

Urban air mobility is a growing market, which will bring new ways to travel and to deliver items covering urban and suburban areas, at relatively low altitudes. To guarantee a safe and robust navigation, Unmanned Aerial Vehicles should be able to overcome all the navigational constraints. The paper is analyzing a novel sensor fusion framework with the aim to obtain a stable flight in a degraded GNSS environment. The sensor fusion framework is combining data coming from a GNSS receiver, an IMU and an optical camera under a loosely coupled scheme. A Federated Filter approach is implemented with the integration of two GRUs blocks. The first GRU is used to increase the accuracy in time of the INS, giving as output a more reliable position that it is fused, with the position information coming from, the GNSS receiver, and the developed Visual Odometry algorithm. Further, a master GRU block is used to select the best position information. The data is collected using a hardware in the loop setup, using AirSim, Pixhawk and Spirent GSS7000 hardware. As validation, the framework is tested, on a virtual UAV, performing a delivery mission on Cranfield university campus. Results showed that the developed fusion framework, can be used for short GNSS outages.

I. Nomenclature

AI	=	Artificial intelligence
EKF	=	Extended Kalman Filter
FF	=	Federated Filter
FO	=	Field of View
GNSS	=	Global Navigation Satellite System
GRU	=	Gated Recurrent Unit
HIL	=	Hardware in the loop
IMU	=	Inertial Measurement Unit
INS	=	Inertial Navigation Unit
KF	=	Kalman Filter
LC	=	Loosely Coupled
LLA	=	Latitude, Longitude and Altitude
LSTM	=	Long Short-Term Memory
MEMS	=	Micro Electromechanical System
MSKCF	=	Multi State Constrain Kalman Filter
NED	=	North, East and Down
NLOS	=	Non-Line of Sight
OKVIS	=	Open Keyframe based Visual Inertial SLAM
PNT	=	Position navigation and Timing
RMSE	=	Root Mean Squared Error

[1] MSc student, School of Aerospace, Transport and Manufacturing (SATM), Cranfield University

[2] PhD candidate, School of Aerospace, Transport and Manufacturing (SATM), Cranfield University

[3] Senior Lecturer, Centre for Autonomous and Cyberphysical Systems, Cranfield University

[4] Reader, Centre for Autonomous and Cyberphysical Systems, Cranfield University

[5] Technical Strategist in Emerging Technologies, Spirent Communications PLC

RNN	=	Recurrent Neural Network
ROVIO	=	Robust Visual Inertial Odometry
SLAM	=	Simultaneous Localization and Mapping
UAM	=	Urban Air Mobility
UAV	=	Unmanned Air Vehicle
VO	=	Visual Odometry
VPS	=	Visual Positioning System

II. Introduction

The future autonomous systems that will be implemented in urban environments have to rely on a stable navigation solution facing all the challenges encountered. Due to the nature of urban and suburban environments[1], Non-Line of Sight multipath signal propagation, can quickly degrade the quality of the GNSS-based positioning, leading to an erroneous localization (see Figure 1). Jamming and spoofing, are other types of signals, that can further affect GNSS-based position, navigation, and timing. Because of the low power of the GNSS receiver, a jammer device can easily induce interferences over the same frequencies used by the GNSS, resulting in an unreliable PNT solution. On the other hand, a spoofer device is capable to emulate a GNSS signal, modifying the actual ground truth[2].

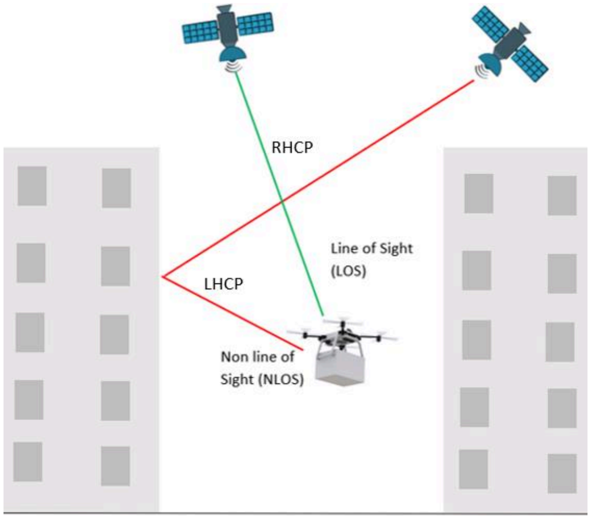

Figure 1 NLOS multipath environment

To mitigate all this errors a robust solution is required, to achieve the best navigation performances, even when the GNSS is not able to provide a PNT solution. Consequently, other sensors can be integrated along the GNSS receiver. A key sensor that can be found on each UAV is the IMU formed typically by a MEMS accelerometer and a MEMS gyroscope. It can measure the linear acceleration and the angular velocity of the UAV. To obtain the position and the velocity from the IMU, an INS[3,4] is used, giving in addition, the aerial vehicle attitude. Unfortunately, the two MEMSs sensors, accumulates errors in time, producing a substantial drift[5]. Further, during the INS mechanization process, the integrator factor is increasing more the errors.

Thus, to minimize further these errors, additional sensors and methods can be used, to extract alternative navigation solutions. Optical based techniques can be used to extract the motion of a UAV in a known or unknown environment. The motion estimation can be relative or absolute, depending on the method used. Relative motion estimation consists of VO[6,7] and SLAM algorithms[8,9]. VO methods involve the analysis of frames, captured by an optical sensor, to estimate its motion through the environment. Instead, the SLAM approach is a more complex navigation algorithm that can be used to estimate the relative motion of the UAV, building at the same time a map. Thus, VO can be considered a subset of a SLAM based navigation method. Although, a motion estimation can be obtained from SLAM and VO, challenging environments with low light conditions or poor in features, can lead the algorithm to diverge. On the other hand, VPS[10,11] is another algorithm that can be used, to extract the absolute user position while moving in a known environment. A proper dataset, formed by different georeferenced aerial images, should be provided, covering the area of interest. Secondly, the optical camera view, while flying, should match the dataset available, to extract the UAV's position as specified in.

Hence, to guarantee a good navigation and continuity, a fusion approach should be used. The sensor fusion framework should guarantee an adequate navigation output even if one sensor is suffering substantial degradation, combining the advantages and disadvantages of each sensor. Consequently, a VIO[12] can be implemented, using an IMU MEMS sensor along an optical camera. A tightly coupled or a loosely coupled approach can be implemented. The first method uses raw IMU, and VO data to compute the drone's ego-motion solution. Instead, the loosely coupled method, uses the solution measured independently from each sensor. In addition, different VIO algorithms are already developed as the MSKCF[13], OKVIS[14], ROVIO[15] or VINS-Mono[16].

Another fusion approach vastly spread along autonomous systems, is fusing INS and GNSS where a KF is used to track the changes of the system. The linearization of the system decreases the precision of the estimated INS errors, when GNSS signal outage occurs for long periods of time. Considering that the INS sensor accumulates errors in time, it is difficult to model the system dynamics and to tune the KF filter parameters optimally.

Besides all the advantages, that KFs can bring, AI techniques can be used to increase the performances of fusion frameworks. Usually, in a real environment, UAVs tend to face many issues and challenges, from meteorological, mechanical to electrical problems. This tends to change the UAV dynamics, and for that, KFs have some limitations. AI instead can bring new horizons if proper training data is provided. RNNs have been used in fusion frameworks as described in[17,18], combing INS and GNSS data. Some of the disadvantages in using such framework, is its high computational cost and its difficulty in storing data for long term[19]. This may cause different errors if used for long term missions. Considering the internal structure of an RNN cell, if the weights are too small the learning rate will be slow and consequently, handling data, in time, may be difficult, leading to the so called 'vanishing gradient' effect. On the other hand, if the weights are too large, the output can diverge, obtaining an unreliable result, leading to an 'exploding gradient' effect. Considering all the advantages and disadvantages of the RNN framework, LSTM[20,21] and GRUs, are valid alternatives in solving the losing memory of the RNN. Although, LSTMs presents good performances against RNNs, GRUs can achieve even better performances. The GRU cell has only two gates, a reset gate, and an update gate, having a less complex structure in comparison with a LSTM cell. This allows the GRU, to be more efficient during the training phase, having fewer gates and parameters to update. GRUs can be used also for navigation as stated in[22,23].

Thus, in this paper a novel federated filter approach using GRUs and EKFs, combining data in a loosely coupled scheme, from an IMU MEMS sensor, a GNSS receiver and an optical camera. To test the novel sensor fusion framework, a custom Unreal Engine world is set-up with AirSim and linked with a Spirent SimGEN 7000 hardware to get more realistic IMU and GNSS data. The paper is organized as follow: in section III the fusion framework is presented, in section IV the experimental set-up, in section V the experimental results with the relevant metrics, and finally in section VI the conclusions.

III. Proposed fusion architecture

Taking into account all the navigation challenges that a UAV may face in an urban environment, a novel fusion framework betweeen IMU, GNSS and an optical sensor can be achived with the integration of two GRU blocks along EKFs, in a FF (Federated Filter)[24] approach as shown in Figure 2. An advantage in using a federated fusion configuration, is the possibility to add different sensors more easily in the future, as subsystems, and to split the computational cost. In this way it is possible to obtain a more robust, sensor fusion architecture, if GNSS outage occurs or the VO diverges. The core module of the sensor fusion framework, is represented by a first GRU block, used to correct and to increase the INS output accuracy. To correct in time the INS errors during the training phase, the GRU block is using GNSS data every 5 s, simulating GNSS outages. Further, during the testing phase, when GNSS outage occurs, the GRU block is predicting the errors, obtaining a more reliable output. From Figure 3 – Left, it is possible to see the IMU/GRU correction architecture, along the diagram with all inputs and outputs.

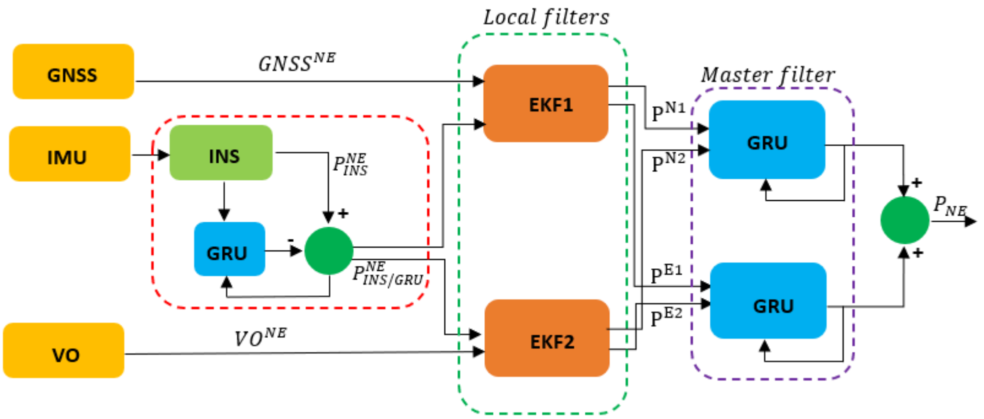

Figure 2 Proposed fusion architecture

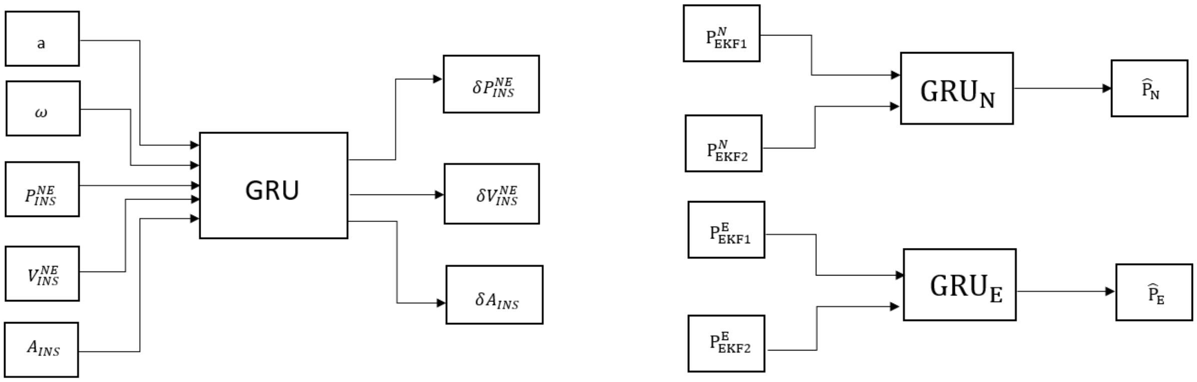

Figure 3 GRU input-output scheme for INS correction (Left) and respectively NE position correction (Right)

Where a is the linear acceleration, ω the angular velocity, P^{NE}_{INS}, V^{NE}_{INS}, A_{INS} the position, velocity, and acceleration from the INS mechanization block, and $\delta P^{NE}_{INS/GNSS}$, $\delta V^{NE}_{INS/GNSS}$, $\delta A_{INS/GNSS}$ the position, velocity and attitude error between the INS and GNSS. For this application only the position is used from the INS block, being able to obtain a final output, defined as:

$$P^{NE}_{INS/GRU} = P^{NE}_{INS} - \delta P^{NE}_{INS/GNSS} \tag{1}$$

The first EKF, is fusing data from the GRU block, and GNSS, that is converted from LLA to a NED coordinated frame. Instead, the second EKF uses the output from the GRU block and the VO, that is generated from a monocular camera placed on the UAV, pointing downwards. To correctly use the VO data, a conversion is done from the camera frame to the navigation frame as specified in[25]. The final section of the fusion framework, is represented by two GRUs, used to correct the output from the two EKFs as it can be seen from Figure 2 and 3-Right. Since only N and E positions are considered, the first GRU is used to process only the N position data, meanwhile the second GRU is processing only the E position data. This split was done do decrease the computation load, required by the framework, thus increasing the overall efficiency. Finally, the 552233 data units that represents each dataset for each coordinate, is used as input, for the corresponding GRU. Each dataset is divided in 80% for training and 20% for testing. The final output of the framework, can be defined as:

$$P_{NE} = \hat{P}_N + \hat{P}_E \tag{2}$$

Where P_{NE} is the final position output of the fusion framework, represented by the sum of $\hat{P}_N$ (estimated North position) and $\hat{P}_E$ (estimated East position).

IV. Experimental set-up

To simulate a more realistic mission and to test the proposed fusion framework, a HIL simulation is set-up. For the testing purpose, a Pixhawk 2.4.8 board is used, along QGroundControl, Unreal Engine, Cesium and AirSim. It is possible in this way to develop a custom environment where to visualize the UAV trajectory and the visual data, coming from the monocular camera.

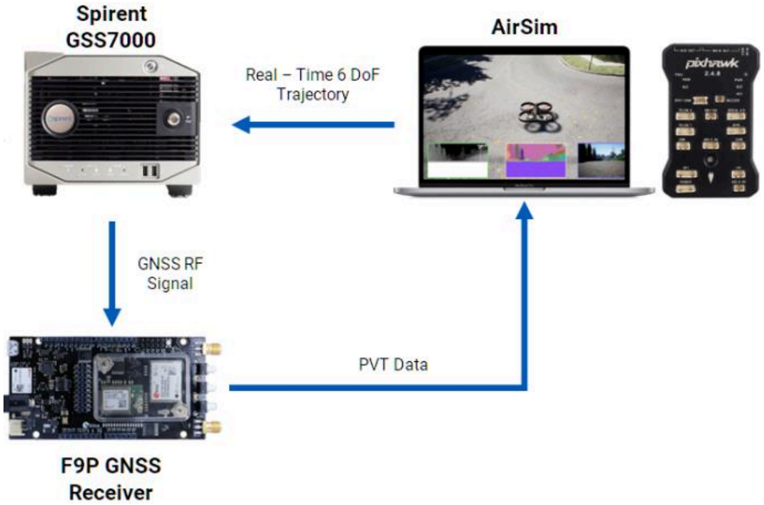

Figure 4 HIL set-up

Using the predefined trajectory from AirSim, it is possible to create a link with SimGEN and the Spirent GSS7000 platform. In this way, it is possible to generate GNSS RF signals, that are processed by the F9P GNSS receiver, obtaining more reliable GNSS data, to be used by the fusion framework. At the same time more realistic IMU data is generated by the SimGEN platform. All the parameters used to setup the accelerometer, gyroscope and GNSS in SimGEN, can be viewed in table below.

Table 1 Accelerometer, Gyroscope and GNSS parameter settings

Accelerometer		Gyroscope	
Scaling Factor (ppm)	500	Scaling Factor (ppm)	500
Bias (m/s^2)	5e-5	Bias (rad/s)	1.212e-4
ARW (m/s^2/$\sqrt{s}$)	3.7e-4	ARW (rad/s/$\sqrt{s}$)	7.33e-5
Update rate (Hz)	100	Update rate (Hz)	100
GNSS			
Pseudorange accuracy (m)		3	
Pseudorange rate accuracy (rad/s)		0.5	
Update rate (Hz)		10	

V. Evaluation

To start the mission, and to evaluate in time, the algorithm performances, a standard trajectory pseudocode is defined, as it can be seen below. The UAV starts from the local base, reaching firstly the cruise altitude for then, flying for a predefined distance, until it reaches the delivery location where it descends again to deliver the package. Once the package is delivered, it reaches again the cruise altitude, coming back to the base station. During every step, a message is sent to the base to confirm the UAV flying status.

```
Algorithm 1
Input: $h_c$, $distance_{UAV}$, $h_d$, $U_{UAV}$, $t_d$
If start=='y':
    Arm
    $0 \rightarrow h_c = 80m \rightarrow U_{UAV} = 2\ m/s$    ▻ $Take\ off\ from\ 0\ to\ the\ cruise\ altitude$
    $distance_{UAV} = 400m$
    $h_c \rightarrow h_d = 20\ m$    ▻ $Decrease\ altititude\ to\ the\ delivery\ altitude$
    $t_d = 10s$    ▻ $Deliver\ package$
    $h_d \rightarrow h_c$    ▻ $Increase\ altititude\ to\ the\ cruise\ altitude$
    $distance_{UAV} = 400m$    ▻ $Back\ to\ base$
    $h_c \rightarrow 0\ m \rightarrow U_{UAV} = 0\ m/s$    ▻ $Land$
else:
    Abort Mission
```

Figure 5 Trajectory pseudocode

Where h_c is the cruise altitude, $distance_{UAV}$ is the distance covered by the UAV, h_d the delivery altitude, U_{UAV} the UAV speed and, t_d the delivering time.

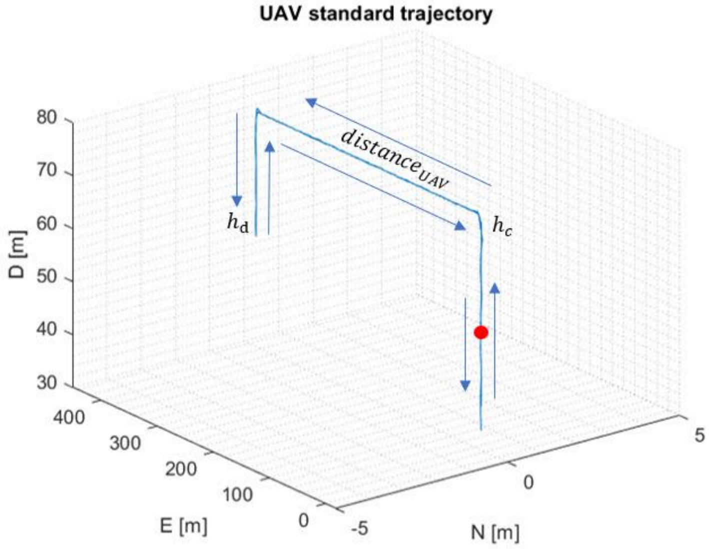

Figure 6 UAV mission trajectory

A. Local Filters evaluation

The first local filter, formed by the IMU/GNSS EKF, provides position information, that cannot be used in a standalone mode, for a UAV navigation as it can be seen from Figure 6. This is due the GNSS outages that are introduced into the system because of the GNSS degraded environment. On the other hand, the output from the second filter provides an alternative navigation solution. It can be seen clearly that the horizontal error in time, of the second local filter, increases during the first part of the mission. Considering that a feature-based algorithm is used, to estimate the ego-motion of the monocular camera during the mission, external features are directly affecting the output of algorithm, and this may introduce errors, if an environment portion if not rich in features. In fact, during the first part of the mission, because the UAV takes off from a green area, and due to the lack of features, the errors are increasing, only to decrease when it approaches a richer area in terms of features as it can be seen from Figure 7 -d. An additional source of error is the scale factor estimation, needed by the VO algorithm, along the entire mission, especially during altitude variations as the take-off and delivery phase. Although, errors are introduced into the fusion framework by the sensors used, the first GRU block helps, in increasing the robustness of the output from the two EKF local filters, as it can be seen from the figure below and from Table 2.

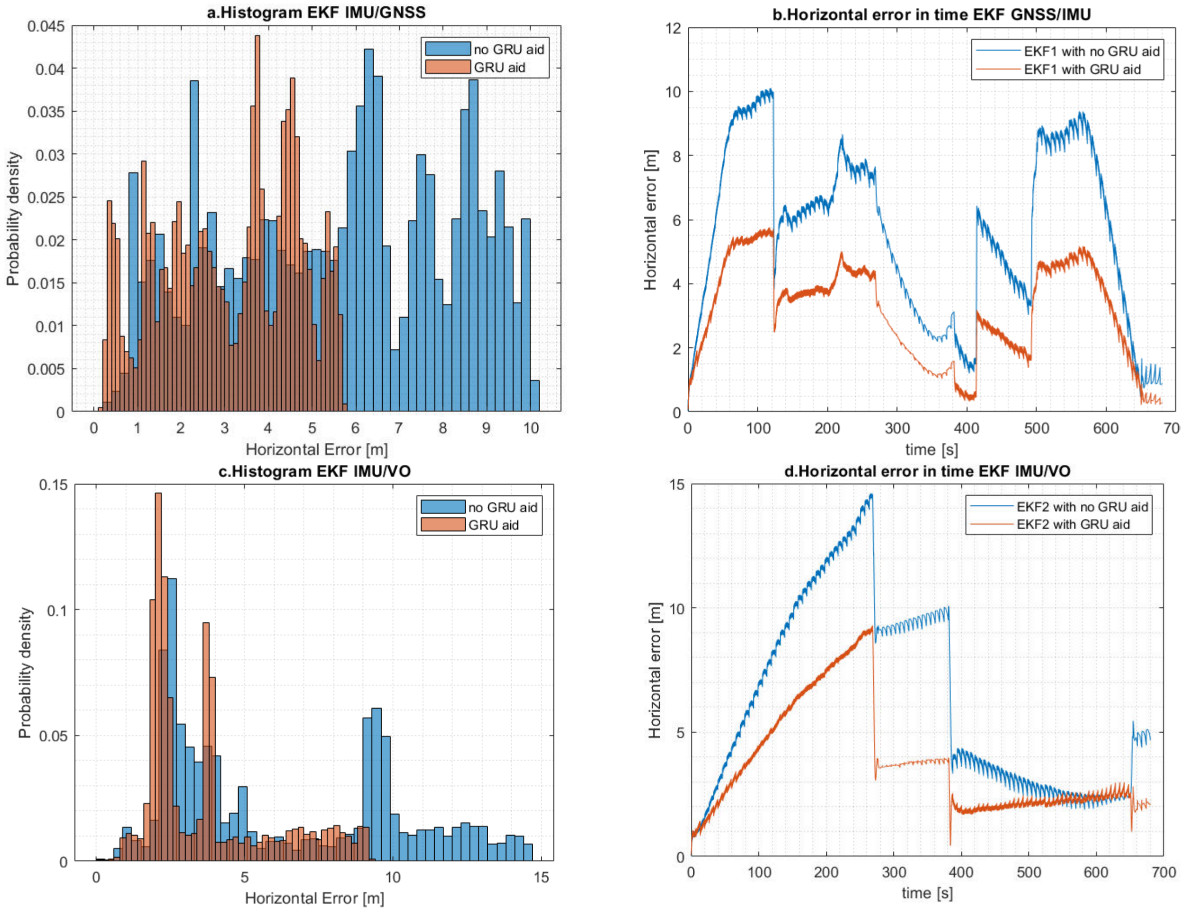

Figure 7 Local filters performance comparison with and without the GRU aid

Table 2 Local filters metrics

Filter	Mean horizontal RMSE [m]	RMSE N [m]	RMSE E [m]	Horizontal RMSE (95th percentile) [m]
EKF1 IMU/GRU/GNSS	3.4637	0.4069	3.4397	5.3880
EKF1 IMU/GNSS (no INS GRU aid)	6.2089	0.9065	6.1423	9.4749
EKF2 IMU/VO/GRU	4.2668	0.5152	4.2356	8.3203
EKF2 IMU/VO (no INS GRU aid)	7.3183	1.2043	7.2185	12.9876

B. Master filter Evaluation

Further, the output of the two local filters is processed by a master GRU block, that decreases consistently the errors introduced into the system as it can be seen from Table 3 and Figure 8 - a. In addition, it can be observed that an EKF block as master filter without the any GRU correction in the framework (see Figure 8 -b), does not increase the robustness of the fusion framework, achieving almost the same performances as the first local filter. Instead, if an EKF is used as master filter, including a GRU block, to predict the INS errors used by the local filters, the errors decrease. In comparison to the two master EKFs situations, slightly better results are obtained if a master GRU is used without the GRU INS correction block. On the other hand, the best performance can be achieved if a GRU block is used to predict the INS errors, and at the same time a master GRU block is used to predict the N and E errors coming from the two local filters (see Table 3). Thus, the GRU is efficient, having a RMSE lower than the output from the two local filters. Overall, the E errors are higher in comparison to the N errors. This is due to all the errors introduced into the fusion architecture by the GNSS, IMU and the monocular camera.

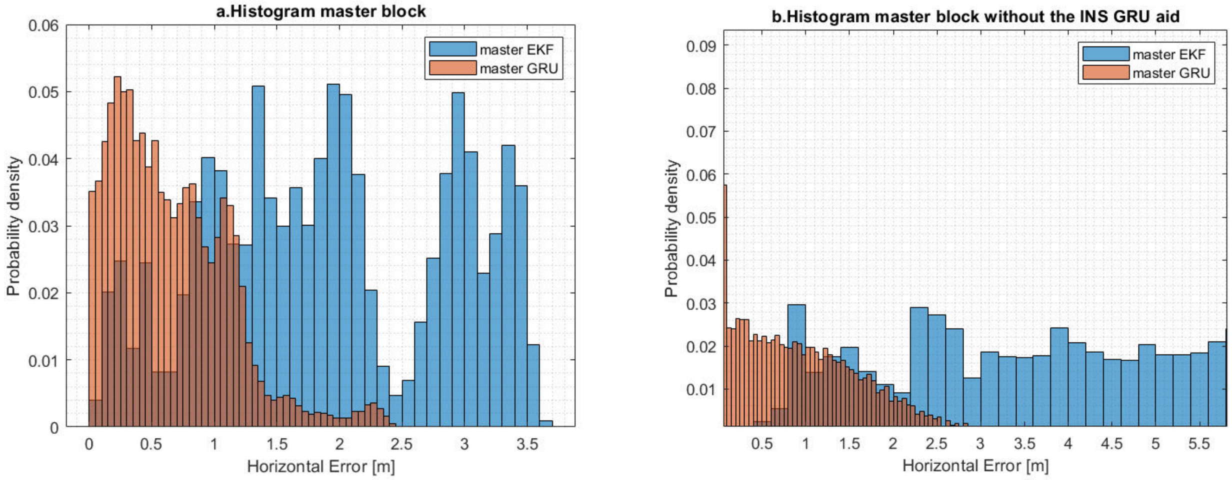

Figure 8 Comparison of performance for master filter based on GRU and EKF.

To evaluate further the performances of the fusion framework, the horizontal accuracy is calculated and evaluated in time along the 95% percentile. Again, the master GRU has the best performances as it can be seen from Table 3.

Table 3 Master filter performances

Filter	Mean horizontal RMSE [m]	RMSE N [m]	RMSE E [m]	Horizontal RMSE (95th percentile) [m]
Master EKF (no INS GRU aid)	6.2036	0.8943	6.1388	9.4665
Master EKF	2.1428	0.1356	2.1385	3.3980
Master GRU (no INS GRU aid)	1.0865	0.018	1.0865	2.1478
Master GRU	0.8035	0.0173	0.8033	1.4608

Conclusion

In the work the authors have presented and demonstrated the following:

- A novel sensor fusion framework based on an FF approach was developed and tested in a LC scheme, fusing data from a GNSS receiver, an IMU MEMS sensor and a monocular camera, performing a delivery mission with GNSS outages of 5 seconds.
- From the performances obtained, the novel fusion framework can provide a good mean horizontal accuracy of 0.8035 m with the aid of the two GRUs blocks and EKFs in comparison to other configurations (see Table 2 and 3). The VIO relies on the first GRU block, thus even if the VO degrades by external factors, a solution can still be provided. The dependency of the VO algorithm, for scale estimation, by the GRU, decreases the integrity of the system.
- A HIL simulation was successfully established between a Pixhawk board and a GSS7000 hardware provided by Spirent, through a UDP connection. Thus, the realism of GNSS and IMU are increased substantially.
- A virtual simulation environment was developed based on Unreal Engine, AirSim and Cesium. Therefore, the UAV can be deployed easily, in urban, sub urban and rural areas, by specifying the initial position in LLA frame. Small perturbances were observed during flights, due to the link between Unreal Engine, AirSim and Cesium, degrading the IMU data recorded. In addition, the Bing map introduced by Cesium, has different blurry parts, that affects the VO algorithm, during some stages of the mission.

References

[1] Castelli, T., Sharghi, A., Harper, D., Tremeau, A., and Shah, M. “Autonomous Navigation for Low-Altitude UAVs in Urban Areas.” 2016.

[2] Zhu, N., Marais, J., Betaille, D., Berbineau, M., Gnss, M. B., and Bétaille, D. “Position Integrity in Urban Envi-Ronments: A Review of Literature.” *IEEE Transactions on Intelligent Transportation Systems*, 2018. https://doi.org/10.1109/TITS.2017.2766768ï.

[3] Falco, G., Pini, M., and Marucco, G. “Loose and Tight GNSS/INS Integrations: Comparison of Performance Assessed in Real Urban Scenarios.” *Sensors (Switzerland)*, Vol. 17, No. 2, 2017. https://doi.org/10.3390/s17020255.

[4] Parthenope, ", and Angrisano, A. *UNIVERSITA' DEGLI STUDI DI NAPOLI GNSS/INS Integration Methods*.

[5] *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems Second Edition*.

[6] Scaramuzza, D., and Fraundorfer, F. “Tutorial: Visual Odometry.” *IEEE Robotics and Automation Magazine*, Vol. 18, No. 4, 2011, pp. 80–92. https://doi.org/10.1109/MRA.2011.943233.

[7] Aqel, M. O. A., Marhaban, M. H., Saripan, M. I., and Ismail, N. B. Review of Visual Odometry: Types, Approaches, Challenges, and Applications. *SpringerPlus*. 1. Volume 5.

[8] Hening, S., Ippolito, C., Krishnakumar, K., Stepanyan, V., and Teodorescu, M. 3D LiDAR SLAM Integration with GPS/INS for UAVs in Urban GPS-Degraded Environments. 2017.

[9] Servières, M., Renaudin, V., Dupuis, A., and Antigny, N. “Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking.” *Journal of Sensors*, Vol. 2021, 2021. https://doi.org/10.1155/2021/2054828.

[10] Jiang, W., Liu, D., Cai, B., Rizos, C., Wang, J., and Shangguan, W. “A Fault-Tolerant Tightly Coupled GNSS/INS/OVS Integration Vehicle Navigation System Based on an FDP Algorithm.” *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 7, 2019, pp. 6365–6378. https://doi.org/10.1109/TVT.2019.2916852.

[11] Shan, M., Wang, F., Lin, F., Gao, Z., Tang, Y. Z., and Chen, B. M. “Google Map Aided Visual Navigation for UAVs in GPS-Denied Environment.” 2017.

[12] Scaramuzza, D., and Zhang, Z. *Visual-Inertial Odometry of Aerial Robots*.

[13] Mourikis, A. I., and Roumeliotis, S. I. *A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation*. 2007.

[14] Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P. *Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization*.

[15] Bloesch, M. ;, Omari, S. ;, Hutter, M. ;, Siegwart, R., Bloesch, M., Omari, S., and Hutter, M. “ETH Library Robust Visual Inertial Odometry Using a Direct EKF-Based Approach Conference Paper Robust Visual Inertial Odometry Using a Direct EKF-Based Approach.” https://doi.org/10.3929/ethz-a-010566547.

[16] Qin, T., Li, P., and Shen, S. “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator.” 2017. https://doi.org/10.1109/TRO.2018.2853729.

[17] Dai, H. fa, Bian, H. wei, Wang, R. ying, and Ma, H. “An INS/GNSS Integrated Navigation in GNSS Denied Environment Using Recurrent Neural Network.” *Defence Technology*, Vol. 16, No. 2, 2020, pp. 334–340. https://doi.org/10.1016/j.dt.2019.08.011.

[18] Jeon, B. J., Petrunin, I., and Tsourdos, A. Recurrent Neural Network Based Sensor Fusion Algorithm for Alternative Position, Navigation and Timing. No. 2021-October, 2021.

[19] Ribeiro, A. H., Tiels, K., Aguirre, L. A., and Schön, T. B. *Beyond Exploding and Vanishing Gradients: Analysing RNN Training Using Attractors and Smoothness*. 2020.

[20] Staudemeyer, R. C., and Morris, E. R. “Understanding LSTM -- a Tutorial into Long Short-Term Memory Recurrent Neural Networks.” 2019.

[21] Lopez, A., Lapata, M., and Keller, F. *Natural Language Understanding Lecture 12: Recurrent Neural Networks and LSTMs Recap: Probability, Language Models, and Feedforward Networks Simple Recurrent Networks Backpropagation Through Time*. 2018.

[22] Xu, S., Petrunin, I., and Tsourdos, A. Experimental Evaluation of GNSS and IMU Fusion Using Gated Recurrent Unit. 2022.

[23] Geragersian, P., Petrunin, I., Guo, W., and Grech, R. An INS/GNSS Fusion Architecture in GNSS Denied Environments Using Gated Recurrent Units. 2022.

[24] Carlson, N. A. “Federated Square Root Filter for Decentralized Parallel Processes.” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 26, No. 3, 1990, pp. 517–525. https://doi.org/10.1109/7.106130.

[25] Xiang, H., and Tian, L. “Method for Automatic Georeferencing Aerial Remote Sensing (RS) Images from an Unmanned Aerial Vehicle (UAV) Platform.” *Biosystems Engineering*, Vol. 108, No. 2, 2011, pp. 104–113. https://doi.org/10.1016/j.biosystemseng.2010.11.003.