Uncertainty-based Sensor Fusion Architecture using Bayesian-LSTM Neural Network

Patrick Geragersian[1] Ivan Petrunin[2] Weisi Guo[3]
Cranfield University, Bedford, Bedfordshire, MK43 0AL, United Kingdom

Raphael Grech[4]
Spirent Communications PLC, Aspen Way, Paignton, Devon TQ4 7QR, United Kingdom

Uncertainty-based sensor management for positioning is an essential component in safe drone operations inside urban environments with large urban valleys. These canyons significantly restrict the Line-Of-Sight signal conditions required for accurate positioning using Global Navigation Satellite Systems (GNSS). Therefore, sensor fusion solutions need to be in place which can take advantage of alternative Positioning, Navigation and Timing (PNT) sensors such as accelerometers or gyroscopes to complement GNSS information. Recent state-of-art research has focused on Machine Learning (ML) techniques such as Support Vector Machines (SVM) that utilize statistical learning to provide an output for a given input. However, understanding the uncertainty of these predictions made by Deep Learning (DL) models can help improve integrity of fusion systems. Therefore, there is a need for a DL model that can also provide uncertainty-related information as the output. This paper proposes a Bayesian-LSTM Neural Network (BLSTMNN) that is used to fuse GNSS and Inertial Measurement Unit (IMU) data. Furthermore, Protection Level (PL) is estimated based on the uncertainty distribution given by the system. To test the algorithm, Hardware-In-the-Loop (HIL) simulation has been performed, utilizing Spirent's GSS7000 simulator and OKTAL-SE Sim3D to simulate GNSS propagation in urban canyons. SimSENSOR is used to simulate the accelerometer and gyroscope. Results show that Bayesian-LSTM provides the best fusion performance compared to GNSS alone, and GNSS/IMU fusion using EKF and SVM. Furthermore, regarding uncertainty estimates, the proposed algorithm can estimate the positioning boundaries correctly, with an error rate of 0.4% and with an accuracy of 99.6%.

I. Nomenclature

AL	=	Alert Limit
BLSTMNN	=	Bayesian-LSTM Neural Network
DL	=	Deep Learning
EKF	=	Extended Kalman Filter
GNSS	=	Global Navigation Satellite System
GPS	=	Global Positioning System
GRU	=	Gated Recurrent Unit
HIL	=	Hardware-In-the-Loop
ICAO	=	International Civil Aviation Organization
IMU	=	Inertial Measurement Unit
IR	=	Integrity Risk
LOS	=	Line-Of-Sight

[1] PhD candidate, School of Aerospace, Transport and Manufacturing (SATM), Cranfield University
[2] Senior Lecturer, Centre for Autonomous and Cyberphysical Systems, Cranfield University
[3] Professor of Human Machine Intelligence, Centre for Autonomous and Cyberphysical Systems, Cranfield University
[4] Technical Strategist in Emerging Technologies, Spirent Communications PLC

LSTM	=	Long Short-Term Memory
ML	=	Machine Learning
NLOS	=	None-Line-Of-Sight
PE	=	Position Estimate
PL	=	Protection Level
PNT	=	Position, Navigation and Timing
PVT	=	Position, Velocity, Time
RAIM	=	Receiver Autonomous Integrity Monitoring
RMSE	=	Root Mean Square Error
RNN	=	Recurrent Neural Network
SVM	=	Support Vector Machine
TTA	=	Time To Alert
UAV	=	Unmanned Aerial Vehicle

II. Introduction

One of the most prevalent challenges facing mounting interests in Unmanned Aerial Vehicles (UAVs) is the ability for autonomous navigation in densely populated environments, ensuring mission safety and availability [1]. Due to the nature of urban canyons, GNSS-based positioning struggle with occlusions caused by large buildings, as direct Line-Of-Sight (LOS) is required for accurate positioning [2]. Other factors, including intentional (jamming and spoofing) or unintentional interference, can lead to catastrophic failures. Alternative sensors that may be utilized include Inertial Measurement Unit (IMU) for improved Positioning, Velocity and Timing (PVT) accuracy [3]. However, such sensors are affected by biases (such as temperature effects) and noises that can cause drift errors known as random walk [4]. Consequently, it is essential to utilize multiple sensor information in conjunction with each other to improve PVT accuracy. Furthermore, understanding the uncertainty of the information provided by the position estimate is important, as it can help determine whether mission safety is at risk.

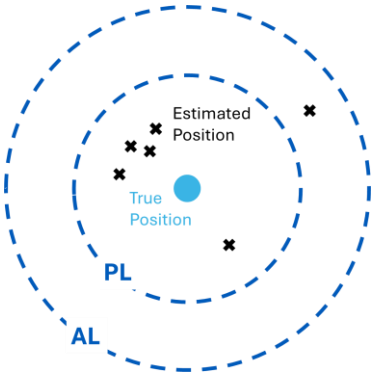


Fig 1: Illustration of relationship between estimated, true position and required integrity

To mitigate the erroneous effects mentioned, it is essential to facilitate an integrity-based sensor fusion approach for achieving the performance level (including integrity) that is required in civil aviation by international aviation bodies. As defined by the International Civil Aviation Organization (ICAO) on characterizing the performance of Position, Navigation and Timing (PNT) solutions, we can classify PNT algorithms in terms of accuracy, integrity, continuity, and availability [5]. Integrity, as defined by the measure of trust that can be placed in the correctness of the information supplied by a navigation system, is the focus of this paper. In low-altitude urban airspace, where error tolerances are tighter than in open sky conditions, integrity is an essential performance consideration that is key to mission safety. Integrity can be clarified in terms of four parameters. These are Alert Limit (AL), Integrity Risk (IR), Time to Alert (TTA) and Protection Level (PL) [6]. PL and AL are illustrated in Fig 1. PL is defined as the statistical bound errors computed to guarantee the probability of the position error is smaller than or equal to the targeted accuracy [7].

A common way to illustrate the relationship between AL, PL, Position Error (PE) and how these impact mission operation is the Stanford integrity diagram [8]. An example of this diagram is shown in Fig 2. Safe mission operation is achieved if AL or PL is above PE. Furthermore, correct integrity algorithm implementation is achieved if PL is

always above PE. This means that the system continually guarantees to alert the mission controller with information that is correct. Hazardous operation is when PE is above AL and PL. This means that the integrity algorithm was not able to correctly estimate the PL and therefore assumes no misleading messages are being given to the mission controller. These conditions can lead to dangerous scenarios such as collisions or potentially death.

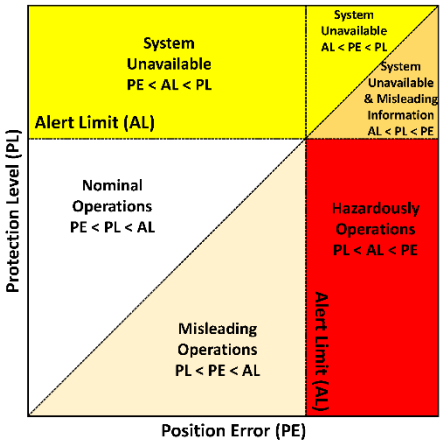


Fig 2: Stanford integrity diagram [8]

GNSS-based integrity monitoring techniques have extensively been researched for larger aircraft, as they are dependent on GNSS navigation. However, due to the nature of their operating environments, these methodologies may not be suitable for UAV navigation in urban canyons. An example integrity monitoring technique used in aviation is Receiver Autonomous Integrity Monitoring (RAIM), which uses pseudorange measurements, given by the receiver, to detect faulty signals and excludes them from PVT calculations [9]. However, this approach can only detect one fault at a time and requires 5 satellites to be in Line-Of-Sight (LOS) for the fault detection [2]. This may be difficult to achieve in urban canyons with a large field of view being obscured and high probability of interference. Furthermore, this approach only works with GNSS signals and does not take into consideration other PNT sources, which is crucial for a robust PNT systems.

Machine Learning (ML) based sensor fusion has been used for a variety of applications, from architectures that can estimate when a machinery needs servicing to better positioning performance by combining multiple sensors that take advantage of physical effects (e.g., speed of light) with different complementing advantages and disadvantages. These improved approximations are utilized in enhancing navigation positioning for vehicles used in urban canyons, which have limited Line-Of-Sight. The ability to observe relationships that are highly non-linear and the capability to use past information is of importance in improving prediction accuracies [10]. However, most of these systems are deterministic. This means that predictions provided by these sensor fusion approaches only provide their "best guess" that is derived by current timestep inputs. Consequently, this does not improve the integrity or robustness of the navigation system.

Current literature, exploring the monitoring of multi sensor-based systems, have proposed schemes such as Residuals Chi-square Test Method (RCTM), suitable for integrity monitoring of GNSS/Strapdown Inertial Navigation Sensor (SINS) fusion systems. This is based on an innovation analysis of a GNSS/SINS Extended Kalman Filter (EKF), where utilizing χ^2 testing, the decision of 'no-failure' or a 'failure' condition is made [11]. However, this system is not suitable for slow-growing ramp failures, such as random walk, that will cause position drifting. An uncertainty-based fusion approach previously discussed in literature is based on a combination of a YOLOv3 object detection network with a customized 1D radar segmentation network to fuse camera and radar data. Particular focus was centered around low light conditions that may prohibit camera performance. To obtain the uncertainty from the network, output of the probability a slice is occupied by an object was provided. This, along with the YOLOv3 network to characterize objects, was used for detecting vehicles at night. YOdar (YOLO radar) achieved a 39.4% mean average precision (mAP), compared to 31.36% mAP for YOLOv3 [12].

Other methods for uncertainty-based fusion are based on evaluating the integrity risk of EKF estimations when faults are present in a measurement. Authors of study [13] applied an EKF-based integrity evaluation approach for

GNSS spoofing scenarios to detect faults after GNSS/INS integration by assuming Inertial Measurement Units (IMU) exhibit fault-free conditions. However, this assumption is valid only for a short period of time as IMU random noises and biases may cause drifting in position estimates. Furthermore, no improvements in positioning performance were discussed in this paper. Extended Kalman Filter (EKF) typically exhibit lower performance , compared to other techniques, under highly non-linear conditions. Hence, there is a need for a fusion-based system, which both improves the position information provided by the fusion of other PNT sources and can quantify uncertainty for the position estimate in complex environments where EKF-based approaches are not effective.
Therefore, the aim of our work is to improve position estimates by fusion of various PNT sensors and to provide an estimated uncertainty distribution that can be utilized to evaluate the safety and reliability of the navigation system. This paper's contributions are summarized below:

A. Derivation and discussion of computational challenges regarding Bayesian-LSTM
B. Proposed navigation system architecture that utilizes both the benefits of Bayesian Neural Networks (BNN), which provides an uncertainty distribution of the estimated position, and Long Short-Term Memory (LSTM) that can use past error dependencies to improve the predictions of position and/or velocity estimates and quantify the corresponding uncertainties.

The paper is organized as follows. Section III will introduce the concept of Bayesian-LSTM and the mathematical derivations relevant to this technique. Furthermore, an exploration of the use-case for Bayesian-LSTMs in the navigation domain is discussed. Section IV illustrates the proposed fusion architecture, methodology for testing the algorithm and an evaluation of its performance to current techniques that exist.

III. Bayesian-LSTM Fusion Algorithm

A. High-Level Overview & Navigational Applications

Bayesian-LSTM refers to the implementation of a Long-Short-Term-Memory (LSTM) neural network with Bayesian inference to allow for the quantification of output uncertainty. Traditionally, machine learning used in predicting position based on multi-sensor inputs (Tree, SVM, RNN, GRU, LSTM) are deterministic [14]. This means that their output is provided as the "best guess", without understanding the uncertainty in that estimate. This is due to the weights being treated as a point estimate. Bayesian Neural Networks (BNN) solve this problem by assigning prior distributions to the parameters [15]. However, BNNs suffer from issues related to intractability. This means that the computational complexity of the problem presented does not have an efficient algorithm to solve them. Therefore, researchers have focused on the development of various approximation methods to estimate the uncertainty of the neural network weights. Blundell et al. [16] proposed a backpropagation-based algorithm which could learn the posterior distribution of the weights of the network. Bayes by Backprop learns the uncertainty of the weights by minimizing the variational free energy on the marginal likelihood. Thus, applying a probability distribution of weights w rather than point estimates and applying Bayes theorem, the conditional probability distribution $P(w|D)$ can be obtained [16]. Obtaining the probability distribution from the Bayesian-LSTM aids in understanding the uncertainty the system places on the predicted outputs. This can be used in the domain of navigation to understand the fused output uncertainty and therefore calculate the PL required to determine system integrity. Furthermore, the advantages that LSTM provides with regards to the ability to observe relationships that are highly non-linear and the capability to use past information, is of importance to improve position prediction accuracy.

B. Mathematical derivation of Bayesian-LSTM

1. Bayes' Theorem

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{\int P(D|\theta)P(\theta)} \quad (1)$$

The classic bayes theorem equation is shown in equation 1, where the posterior $p(\theta|D)$ is given by the likelihood $p(D|\theta)$ of the data D given a model with parameters θ, $p(\theta)$ the prior, and $\int P(D|\theta)P(\theta)$ is the sum of all possible likelihoods for all possible prior values. However, with a large set of weights, the integral becomes an intractable problem. Therefore, classical Bayes theorem becomes impractical to calculate in a real-world scenario. Consequently, the posterior $p(\theta|D)$ needs to be estimated. Variational Inference (VI) allows for an additional posterior distribution $q(w|\Theta)$ to be used as an approximate for the first distribution. To obtain this, Kullback-Leibler (KL) is minimized

between $q(w|\Theta)$ and $p(\theta|D)$ using an optimization process. This, therefore, leads to the solution of Θ^* which is the best estimate $q(w|\Theta^*)$ for the distribution of $p(\theta|D)$ [17]:

$$\Theta^* = \arg\min_{\Theta} KL[q(w|\Theta^*)||p(\theta|D)] \quad (2)$$

$$\Theta^* = \arg\min_{\Theta} KL[q(w|\Theta^*)||p(\theta|D)] \quad (3)$$

$$\Theta^* = \arg\min_{\Theta} \int q(w|\Theta^*) log \frac{q(w|\Theta^*)}{p(\theta)p(D|\theta)} dw \quad (4)$$

$$\Theta^* = \arg\min_{\Theta} KL[q(w|\Theta^*)||p(\theta)] - \mathbb{E}_{q(w|\Theta^*)}[\log p(D|\theta)] \quad (5)$$

$$F(D,\Theta) = KL[q(w|\Theta^*)||p(\theta)] - \mathbb{E}_{q(w|\Theta^*)}[\log p(D|\theta)] \quad (6)$$

Equation 6 shows the resulting variational free energy equation or also known as the Evidence-based Lower Bound (ELBO) for the solution Θ^*. However, the KL term for $q(w|\Theta^*)$ is also intractable due to the integral required to solve for Θ^*. Therefore, $q(w|\Theta^*)$ requires approximation. Blundell et al. implemented a solution to this problem by reparametrizing to obtain variational posterior $q(w|\Theta^*)$ using stochastic gradient descend and sampling (commonly referred to as Bayes by Backprop). Reparameterization is achieved by assuming that the variational posterior is a diagonal Gaussian distribution, from which Θ is parameterized with mean μ and standard deviation σ. Therefore, equation 6 can now be written as shown in equation 7:

$$F(D,\Theta) \approx \sum_{i=1}^{n} \log q(w^i|\Theta) - \log p(\theta^i) - \log p(D|\theta^i) \quad (7)$$

As the integral has now been removed, the equation above becomes tractable and therefore a way to estimate the original posterior distribution.

2. Bayesian-LSTM

Conventional LSTM cells, as shown in Fig 3, are made up of three gates. These are the input gate, forget gate, and output gate. This can also be written as [18]:

$$f_t = \sigma(W_{fx}x_t + w_{fh}h_{t-1} + b_f) \quad (8)$$

$$i_t = \sigma(W_{ix}x_t + w_{ih}h_{t-1} + b_i) \quad (9)$$

$$\bar{c_t} = tanh(W_{cx}x_t + w_{ch}h_{t-1} + b_c) \quad (10)$$

$$c_t = f_t \,.\, c_{t-1} + i_t \,.\, \bar{c_t} \quad (11)$$

$$o_t = \sigma(W_{ox}x_t + w_{oh}h_{t-1} + b_o) \quad (12)$$

$$h_t = o_t \,.\, \tanh(c_t) \quad (13)$$

Where t is the recent time to predict, $w = [W_{fx}, W_{fh}, W_{cx}, W_{ox}, W_{oh}]$ are the weight, and $b = [b_f, b_i, b_c, b_o]$ are the biases. LSTM weights and bias parameters are normally constant (deterministic). However, in Bayesian LSTM the

weights and biases are based on Bayesian Inference. To represent this, weight and bias sampling can be written as [19]:

$$W_n^i = N(0,1) * \log(1 + p^i) + \mu^i \quad (14)$$

$$b_n^i = N(0,1) * \log(1 + p^i) + \mu^i \quad (15)$$

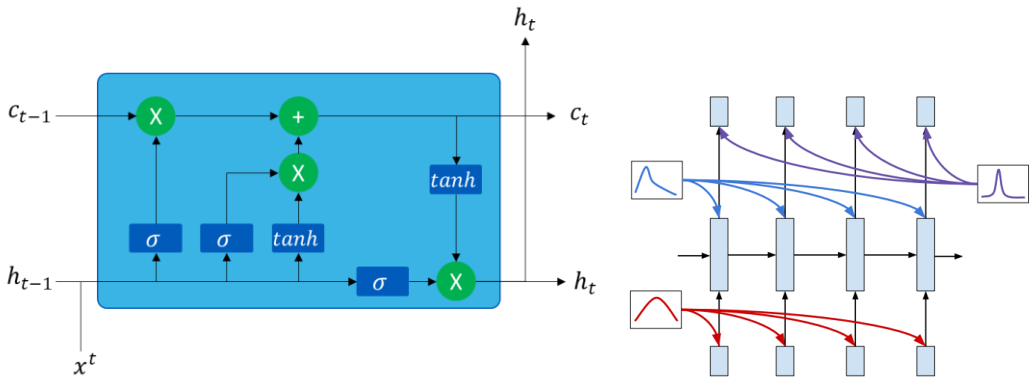


Fig 3: Diagram of an individual LSTM cell with input gate, forget gate and output gate [20]

The distribution of weights and biases are now parameterized by the mean and standard deviation. Once KL is minimized, the parameters that represent the distribution are set. To obtain a distribution of the possible position estimates from the network, each sequence is provided as a sample to the trained NN and is then run multiple times to obtain a spread. The distribution of those results is then processed to calculate the mean and standard deviation of each time frame.

C. Proposed Architecture

Figure 4 shows a high-level diagram of the proposed fusion design. GNSS position measurements are converted from latitude, longitude, and altitude form into NED frame. This, along with Inertial Measurement Unit (IMU) raw readings are fed as the input to the Bayesian-LSTM Neural Network. The output of the trained Neural Network (NN) is the position estimate, the mean and the standard deviation of that output. The uncertainty distribution is used to calculate the PL of the positioning estimate.

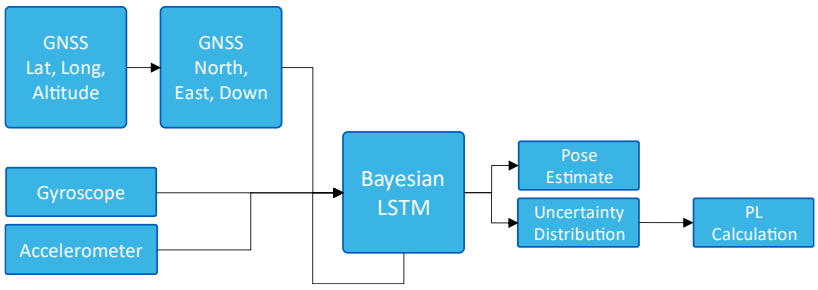


Figure 4: High level diagram illustrating the proposed architecture and data processing

IV.Uncertainty & Fusion Algorithm Evaluation

A. Simulation System Setup

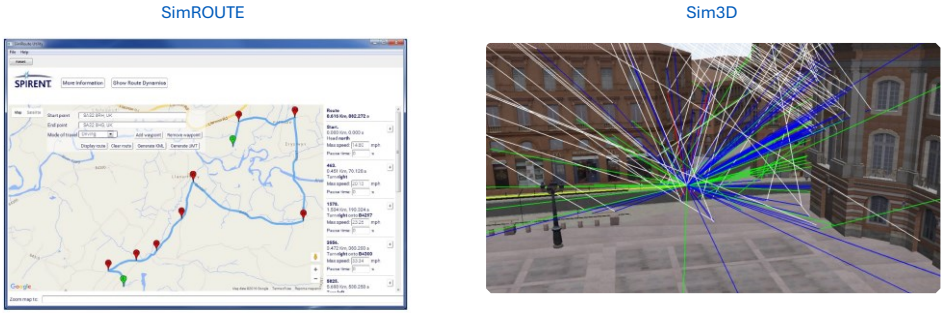


Fig 5: SimROUTE & Sim3D screenshot of route planning and multipath simulation

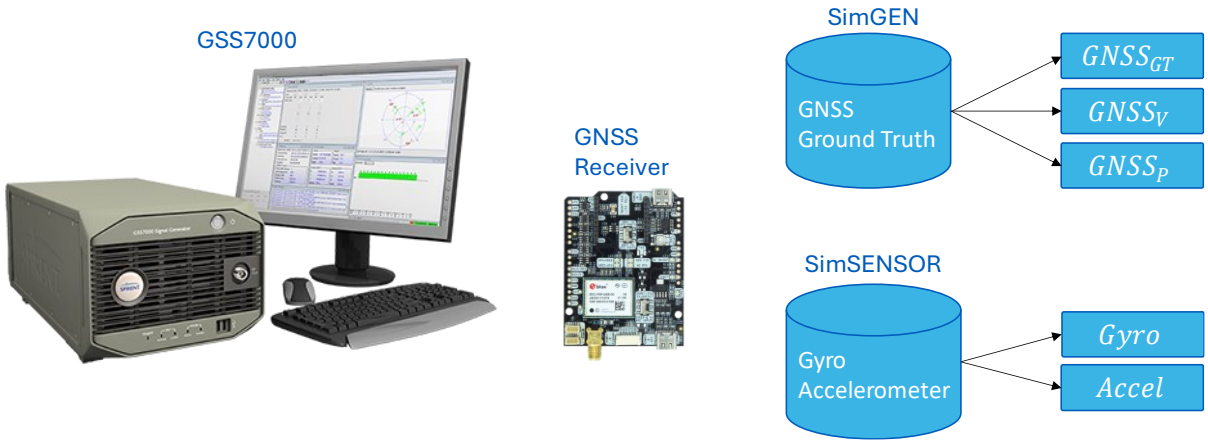


Fig 6: Illustration showing the process from simulation to data generation and processing

To generate training data, Hardware-In-the-Loop (HIL) simulation was utilized using Spirent's SimGEN software and GSS7000 hardware. To improve the simulation realism, OKTAL-SE's Sim3D software was used to replicate the effects of multipath in urban canyons. The software/hardware and their corresponding data being collected is illustrated in Fig 5 and Fig 6. The generated Radio Frequency (RF) from GSS7000 was then fed into a Ublox ZED-F9P receiver to calculate the position and velocity information. Only GPS constellation was simulated at the time and date of 00:00:00 September 05, 2022. To obtain IMU sensor data, Spirent's SimSENSOR software was used to generate accelerometer and gyroscope records. Stochastic and deterministic errors were tuned in SimSENSOR to mimic an Advanced Navigation Orientus IMU sensor. The specifications of those sensors mentioned are shown in Table 1 and Table 2.

Data collected from these sensors and simulation systems was stored in a CSV file format with the time and date of each timestep recorded. The data was then processed in MATLAB to generate the ground truth data used as the training target. The input data consists of the IMU sensor raw output (accelerometer and gyroscope in XYZ) and the GNSS latitude, longitude and height information that was recorded at 1Hz. Using interpolation, the data was synchronized to the same timestep as the accelerometer and gyroscope at 100Hz. In regards to the trajectory, it was generated as a grid-based path in an urban canyon scenario with the UAV travelling at a height of between 60 m to 120 m. This trajectory provides an overview of the typical maneuvers and patterns exhibited when flying a drone. The simulated drone flight is conducted in the Wanhua district in Taipei, Taiwan as there is a diverse representation of urban, suburban, and clear sky conditions. Ground truth, representing the actual trajectory of the UAV, is shown in Fig 7.

Table 1: Accelerometer and Gyroscope specification

Accelerometer		*Gyroscope*	
Scaling factor (ppm)	400	***Scaling factor (ppm)***	400
Bias (mg)	0.099	***Bias (deg/h)***	0.001
Accelerometer Random Walk (ARW) (m/s/sqrt(h))	0.0023	***Gyroscope Random Walk GRW (deg/sqrt(h))***	0.003
Update rate (Hz)	100		

Table 2: GNSS receiver specification

Pseudo range accuracy (m)	3
Pseudo range rate accuracy (m/s)	0.5
Update rate (Hz)	1

Fig 7: Actual UAV trajectory in the simulated scenario

The IMU, GNSS, and ground truth data collected was then split into training data (80%) for the NN to train on and testing data (20%) used to analyze and evaluate the proposed system. Hyperparameter tuning was carried out to maximize performance of the NN. The final ML layers were based on two LSTM layers (128 and 32 cells) with a dropout probability of 20%, a Tanh layer, a fully connected layer and ADAM used as the optimizer for training. Training was carried out at a learning rate of 0.001 and iterated through 2000 epochs, with the output from the NN being the combined position/uncertainty estimate. The PL was then calculated based on the standard deviation provided by the Bayesian-LSTM. To calculate the PL for fused position estimation, the equation below was used [21]:

$$HPL = K_H d \tag{16}$$

$$d = \sqrt{\frac{\sigma_E^2 + \sigma_N^2}{2} + \sqrt{\left(\frac{\sigma_E^2 - \sigma_N^2}{2}\right)^2 + (\sigma_E \sigma_N)^2}} \tag{17}$$

Where σ_E is the standard deviation in the East direction (m) and σ_N is the standard deviation in the North direction. K_H is the scalar factor that are both set to one in this scenario which represents an integrity risk of 0.01%.

Analysis of the proposed architecture performance is split into two sections. The first section is the evaluation of the fusion performance. Root-Mean-Square Error (RMSE) is used to compare existing techniques that aim to reduce the positioning error. Furthermore, the mean, standard deviation, and 95% error bounds were compared. A comparison of the proposed BLSTMNN with the Ground Truth and GNSS is performed. The second phase of analysis is based on the uncertainty and PL parameters calculated by the proposed system. 99% boundaries are used to compare against the ground truth data.

B. Results – Sensor Fusion

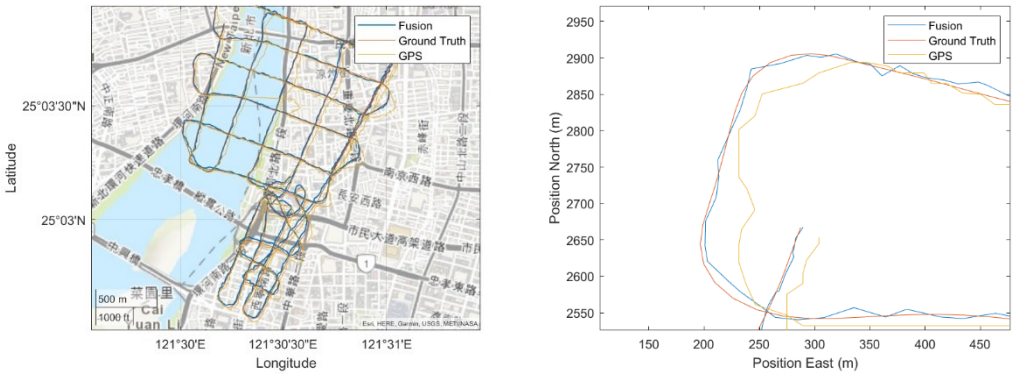


Fig 8: Positioning performance for GPS alone and BLSTMNN fusion vs ground truth

Fig 8 shows the positioning performance comparison between the GNSS only positioning and the BLSTMNN fusion architecture proposed in this paper. GPS data plotted shows deviation from the ground truth. This is due to the environmental and processing noises such as multipath and weather effects simulated, which contribute to the divergence of estimate and actual position. The proposed architecture reduces these errors with fusion of IMU data provided by the accelerometer and gyroscope. This is especially true in heading changes, where the fusion architecture is still able to reduce GNSS errors, despite this representing a highly non-linear case for fusion.

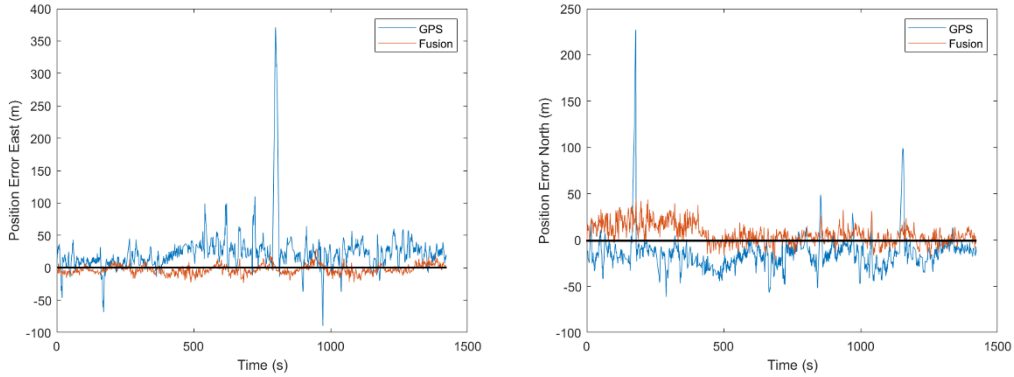


Fig 9: Graphs showing the position error in the east (left) and north (right) direction

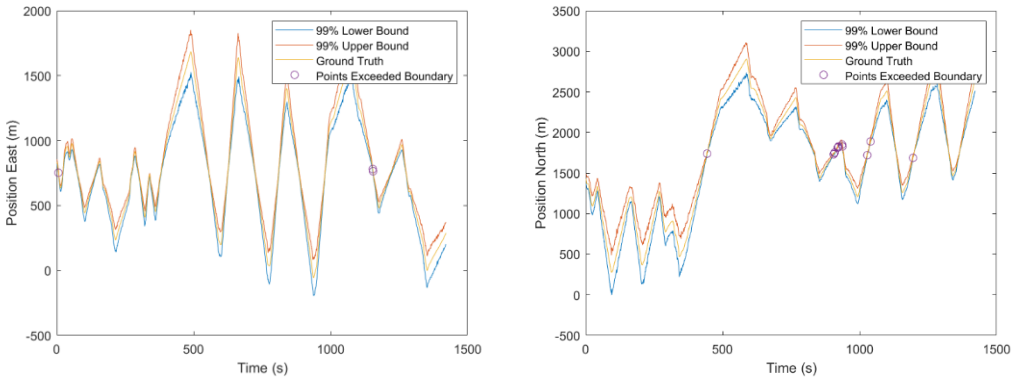
Fig 9 is a comparison of position error in the east and north direction for GPS and the proposed fusion architecture. In both directions, the BLSTM performs better than GPS only. The perturbation exhibited by BLSTM from the 0 m line is less than from those from GNSS. Furthermore, during GNSS outages (as can be seen by the high peaks totaling 7 cases), the proposed architecture is able to mitigate for these events with raw data provided by the IMU.

Table 3: Comparison of the mean, standard deviation, 95% error bounds and RMSE of the GPS, EKF, SVM and BLSTM

Technique (Error component)	*Mean (m)*	*Std (m)*	*95th (m)*	*RMSE (m)*	*% Reduction*
GPS Only (East)	23.16	33.7	90.56	40.91	0 (Reference)
EKF (East)	12.35	15.22	42.79	17.88	52.75
SVM (East)	7.52	10.92	29.36	11.01	67.58
BLSTM (East)	***2.81***	***7.01***	***16.82***	***7.55***	***81.42***
GPS Only (North)	14.55	20.91	56.35	25.46	0 (Reference)
EKF (North)	7.53	15.34	38.21	15.55	32.19
SVM (North)	4.34	12.22	28.78	10.34	48.92
BLSTM (North)	***6.65***	***10.58***	***27.82***	***12.50***	***50.63***
GPS Only (Horizontal)	27.35	39.66	106.66	48.18	0 (Reference)
EKF (Horizontal)	14.46	21.61	57.36	23.70	46.21
SVM (Horizontal)	8.68	16.39	41.11	15.10	61.45
BLSTM (Horizontal)	***7.22***	***12.70***	***32.51***	***14.61***	***69.52***

Table 3 shows a comparison of the mean, standard deviation, 95% error bounds and RMSE of the GPS, Extended Kalman Filter (EKF), Support Vector Machine (SVM) and the Bayesian-LSTM proposed in this paper. In all cases except for North, BLSTM mean error is the lowest as compared to the other techniques or sensor outputs. Furthermore, in all cases regarding standard deviation, the proposed system performs best. This is due to the nature of LSTMS which can look at past dependencies, that could influence the predicted output from the architecture, and therefore reduce output error and variation. Looking at the combined 95th percentile error, BLSTM reduces this by 70% as compared to GNSS only.

C. Results – Uncertainty



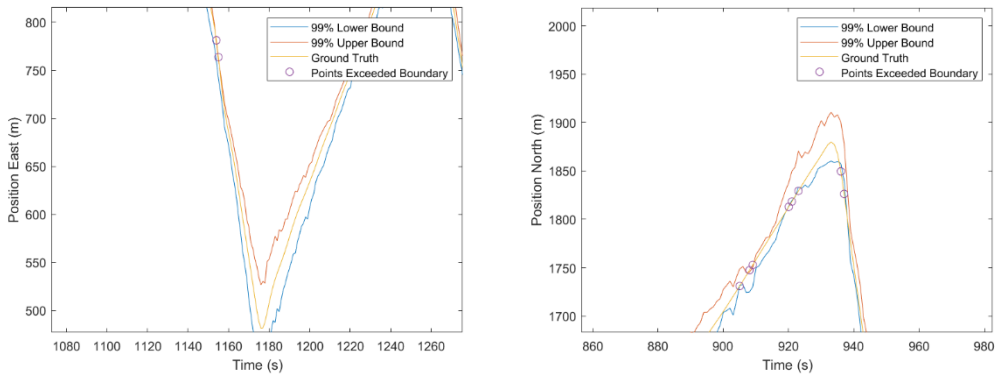


Fig 10: East (left) and North (right) estimated position boundaries compared to the ground truth with points exceeding the distribution highlighted

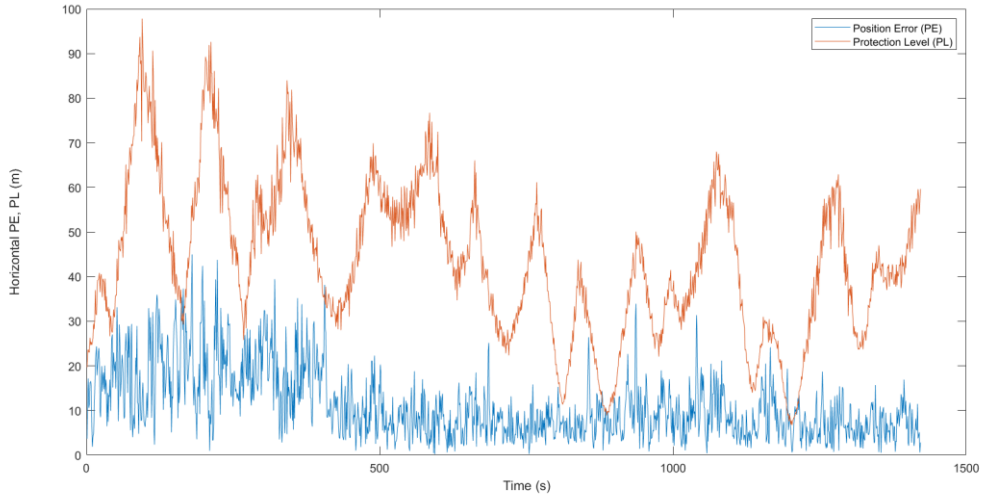


Fig 11: Comparison of Protection Level (PL) and Position Error (PE) in flight

Fig 10 shows the position distribution output from the Bayesian-LSTM compared to the ground truth in the east and north direction. Fig 11 shows a comparison between the PL calculated from the uncertainty distribution and PE in the horizontal direction. From these comparisons, most of the ground truth is between the 99% lower bound and upper bound with a few cases (totaling 10 data points) outside those boundaries. This represents a 0.4% error rate with 99.6% accuracy. Furthermore, regarding the PL, only 4 cases exist where PE>PL. Error cases correlate with the narrowing of the distribution which may indicate overconfidence or overfitting from the BLSTM. Furthermore, during heading changes, the distribution widens as the uncertainty in the fusion increases. This indicates that there is an increase in the complexity of the relationship between the input and the output and is therefore identified by the system as an increase in the uncertainty of the position estimate.

V.Conclusion

This paper presents a new approach for sensor fusion of IMU and GNSS using deep machine learning. The system also provides an estimate PL based on the uncertainty obtained from the stochastic outputs of the neural network. The proposed method predicts the position and uncertainty by utilizing Bayes by Backpropagation to understand the posterior distribution. It overcomes the disadvantages of traditional deep learning methods that have been researched based on deterministic approaches. The combination of advantages associated to LSTM and the ability to estimate PL

benefits mission safety. This system can, therefore, be utilized in areas such as urban canyons, which have reduced GNSS availability/accuracy but require high integrity. Results show a reduction in position error in the horizontal direction by up to 70% as compared to GPS only. Furthermore, compared to other techniques such as EKF and SVM, BLSTMNN was shown to be the best performer. In terms of uncertainty estimates, an error rate of 0.4% was observed with only 4 PL cases that exceeded PE.

VI.References

[1] M. V. Rajasekhar and A. K. Jaswal, "Autonomous vehicles: The future of automobiles," in *2015 IEEE International Transportation Electrification Conference (ITEC)*, Chennai, 2015.

[2] P. Geragersian, I. Petrunin, W. Guo and R. Grech, "Multipath Detection from GNSS Observables Using Gated Recurrent Unit," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, Portsmouth, 2022.

[3] X. Shuoyuan, I. Petrunin and A. Tsourdos, "Experimental Evaluation of GNSS and IMU Fusion Using Gated Recurrent Unit," in *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Dulles, 2022.

[4] Q. Zhang, X. Niu and C. Shi, "Impact Assessment of Various IMU Error Sources on the Relative Accuracy of the GNSS/INS Systems," *IEEE Sensors Journal ,* vol. 20, no. 9, pp. 5026-5038, 2020.

[5] O. K. Isik, J. Hong, I. Petrunin and A. Tsourdos, "Integrity Analysis for GPS-Based Navigation of UAVs in Urban Environment," *Robotics,* vol. 9, no. 3, p. 66, 2020.

[6] Y. Zhai, X. Zhan and B. Pervan, "Bounding Integrity Risk and False Alert Probability Over Exposure Time Intervals," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 56, no. 3, pp. 1873-1885, 2020.

[7] H. Jiang, T. Li, D. Song and C. Shi, "An Effective Integrity Monitoring Scheme for GNSS/INS/Vision Integration Based on Error State EKF Model," *IEEE Sensors Journal,* vol. 22, no. 7, pp. 7063-7073, 2022.

[8] M. Tossaint, J. Samson, Toran F, J. Ventura_traveset, J. Sanz, M. Hernandez-Pajares and J. M. Juan, "The Stanford - ESA Integrity Diagram: Focusing on SBAS Integrity," in *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, Fort Worth, 2006.

[9] S. PAternostro, T. Mooore, C. Hill, J. Atkin and H. P. Morvan, "Evaluation of advanced receiver autonomous integrity monitoring performance on predicted aircraft trajectories," in *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Savannah, 2016.

[10] P. Geragersian, I. Petrunin, W. Guo and R. Grech, "An INS/GNSS fusion architecture in GNSS denied environment using gated recurrent unit," in *AIAA SCITECH 2022 Forum*, San Diego, 2022.

[11] C. Yang, A. Mohammadi and Q.-W. Chen, "Multi-Sensor Fusion with Interaction Multiple Model and Chi-Square Test Tolerant Filter," *Sensors,* vol. 16, no. 11, p. 1835, 2016.

[12] K. Kowol, M. Rottmann, S. Bracke and H. Gottschalk, "Uncertainty-based Sensor Fusion for Vehicle Detection with Camera and Radar Sensors," *arXiv,* 2020.

[13] S. Khanafseh, N. Roshan, S. Langel, F.-C. Chan, M. Joerger and B. Pervan, "GPS spoofing detection using RAIM with INS coupling," in *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, Monterey, 2014.

[14] Y. Han, J. C. Lam, V. O. Li and D. Reiner, "A Bayesian LSTM model to evaluate the effects of air pollution control regulations in Beijing, China," *Environmental Science & Policy,* vol. 115, pp. 26-34, 2021.

[15] D. J. MacKay, "Bayesian neural networks and density networks," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment,* vol. 354, no. 1, pp. 73-80, 1995.

[16] C. Blundell, J. Cornebise, K. Kavukcuoglu and D. Wierstra, "Weight uncertainty in neural networks," *ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning,* vol. 37, pp. 1613-1622, 2015.

[17] N. Zhang, X. Chen and L. Quan, "Layerwise Approximate Inference for Bayesian Uncertainty Estimates on Deep Neural Networks," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, 2021.

[18] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Long Short-Term Memory,* vol. 9, no. 8, pp. 1735-1780, 1997.

[19] P. Esposito, "Bayesian LSTM on PyTorch — with BLiTZ, a PyTorch Bayesian Deep Learning library," *Towards Data Science,* p. 1, 2020.

[20] M. Fortunato, C. Blundell and O. Vinyals, "Bayesian Recurrent Neural Networks," in *31st Conference on Neural Information Processing Systems (NeurIPS 2017)*, Long Beach, 2017.

[21] J. Oliveira and C. Tiberius, "Quality Control in SBAS: Protection Levels and Reliability Levels," *The Journal of Navigation,* vol. 62, no. 3, pp. 509-522, 2009.

Uncertainty-based sensor fusion architecture using Bayesian-LSTM neural network

Geragersian, Patrick

2023-01-19

Geragersian P, Petrunin I, Guo W, Grech R. (2023) Uncertainty-based sensor fusion architecture using Bayesian-LSTM neural network. In: AIAA SciTech Forum 2023, 23-27 January 2023, National Harbor, Maryland, USA. Paper number AIAA 2023-0193
https://doi.org/10.2514/6.2023-0193