A Return-to-home Unmanned Aerial Vehicle Navigation Solution in Global Positioning System Denied Environments via Bidirectional Long Short-term Memory Reverse Flightpath Prediction

Mustafa Alkhatib(1), Mohammad Nayfeh(1), Khair Al Shamaileh(1), Naima Kaabouch(2), and Vijay Devabhaktuni(3)

(1) Department of Electrical and Computer Engineering, Purdue University Northwest, Hammond, IN 46323, USA
(2) School of Electrical Engineering and Computer Science, The University of North Dakota, Grand Forks, ND 58202, USA
(3) Department of Electrical Engineering, Illinois State University, Normal, IL 61761, USA

Corresponding author: Khair Al Shamaileh (e-mail: kalshama@pnw.edu)

***Abstract*–**In this paper, bidirectional long short-term memory (B-LSTM) deep learning modeling is proposed as an approach to facilitate autonomous return-to-home (RTH) aerial navigation in environments with compromised global positioning system (GPS) reception. Logged samples of ten radiometric features are extracted from onboard sensors (i.e., accelerometer, barometer, GPS, gyroscope, magnetometer) in two outdoor experimental scenarios of different altitudes and velocities. These samples are used for training and validating B-LSTM models with single and parallel architectures. The former architecture consists of a single B-LSTM model that processes all input features across the x-, y-, and z-axes to predict a three-dimensional local position, whereas the latter comprises three parallel B-LSTM models, each for processing only the features of a specific dimension (i.e., x, y, or z) and predicting local position in the respective axis. Evaluations demonstrate the validity of the proposed approach, with a four-meter average mean square error (MSE). This is achieved without imposing resource-consuming computational overhead, modifications to existing hardware, or changes to physical infrastructure and communication protocols. Due to using existing onboard sensors and accommodating varied scenarios, the proposed approach finds applications in autonomous navigation, including unmanned aerial vehicles (UAVs) and ground vehicles.

***Keywords*:** Bidirectional long short-term memory, global positioning system, return-to-home, unmanned aerial vehicles.

1. Introduction

Recently, the use of unmanned aerial vehicles (UAVs) has increased significantly in many applications, such as search and rescue missions, surveillance, construction, delivery of goods, agriculture, and smart cities [1–3]. Consequently, the UAV market has grown notably and is expected to reach USD 38.3B by 2027 [4]. This growth is thought to be due to the increasing demand for autonomous operations as well as the advances in enabling control and navigation technologies (e.g., swarm networking and communications).

The global positioning system (GPS) is instrumental in any operation that requires synchronization and location information. However, the GPS signal strength at the receiver is often in the magnitude of 10^{-14} watts (i.e., –110 dBm); and therefore, is subject to limitations that potentially compromise a reliable utilization in autonomous navigation [5–9]. For example, multipath fading, caused by signal reflections off surrounding objects, leads to multipath copies with varying arrival times, attenuations, and phases that result in inaccurate position calculations [10]. To address the multipath challenge, a solution encompassing omnidirectional infrared cameras in parallel with a satellite orbit simulator was exploited to identify surroundings in urban areas and exclude the satellites with obstructed or non-direct line of sight links from positioning estimation [11]. Although this solution demonstrated improved positioning accuracy, the associated additional hardware and computational requirements setback its ready acceptance. An adaptive code/carrier phase measurements filtering technique for multipath error reduction was studied in [12]. Nevertheless, the validity of this technique was limited to stationary GPS receivers without considering mobile scenarios, where multipath effects differ significantly. A multipath mitigation methodology via a nonlinear regression was explored in [13]. However, this methodology was tested with stationary receivers in environments with fixed surrounding objects. In [14], multipath mitigation via the fusion of GPS measurements was introduced, which benefits from a cooperative positioning of multiple GPS receivers.

One other challenge that affects the quality of GPS reception is the denial-of-service (e.g., jamming attack), which involves disrupting location awareness by launching interference that overpowers authentic transmissions [15]. Various solutions were introduced in the literature for such a particular challenge, including those with integrated angle of arrival, null steering, and adaptive notch filtering protocols [16–18]. However, these solutions require additional hardware (e.g., antenna arrays) and impose higher computational complexity. In [19], deep reinforcement learning and reconfigurable intelligent surfaces (RISs) were explored as an approach for GPS jamming mitigation. Nevertheless, this approach was validated in simulation environments; not to mention the limitations associated with integrating RISs with autonomous vehicles. Similarly, a RIS-based positioning, navigation, and timing approach utilizing reinforcement learning and game theory was introduced in [20]. However, this approach was developed and evaluated in a simulation environment. Moreover, it assumed the presence of anchor nodes, RISs, and that targets can receive beacon signals from such nodes.

Other solutions utilized visual localization (e.g., template matching, feature points matching, simultaneous localization and mapping, inertial navigation, computer vision) to offer a GPS-free navigation [21–25]. However, these solutions require contributions from multiple receiver nodes as well as added sensor devices (e.g., LiDAR, cameras). In [26, 27], sensor fusion methods were used as a navigation solution in GPS-denied environments. These methods, however, are susceptible to accumulating sensor errors, which may result in imprecise positioning estimation over time. The use of deep learning techniques to provide GPS-free navigation solutions was explored in [28–31]. However, these solutions were implemented in simulation environments or incorporated visual sensors; not to mention their imposed high computational requirements.

This work addresses the aftermath of losing GPS reception amid navigation and suggests a countermeasure strategy that enables a safe and autonomous return to the launch location, i.e., return-to-home (RTH), with the use of deep learning. The approach proposed herein exploits bidirectional long short-term memory (B-LSTM) modeling by leveraging data from the UAV onboard sensors during the flight. The primary reason for adopting the B-LSTM architecture stems from the sequential properties of UAV sensory measurements during navigation. In other words, B-LSTM is characterized for bidirectional information processing, which allows for capturing temporal dependencies and relationships among the features within a dataset. Therefore, it represents an attractive candidate for predicting and reversing flightpaths. On the other hand, conventional deep learning models (e.g., convolutional neural networks, transformer-based models) excel at spatial feature extraction. Nevertheless, these models face challenges in handling and predicting data of a sequential nature. It is paramount to point out that other models, e.g., recurrent neural networks (RNNs), can also be used to process sequential data. However, RNNs often suffer from the vanishing gradient, which limits their ability to capture long-term temporal dependencies. It is paramount to point out here that the proposed approach differs from those reported in the literature in the following aspects:

i. This approach utilizes feature samples extracted from readily available onboard sensors (e.g., accelerometer, barometer, GPS, gyroscope, magnetometer) and does not require added hardware devices, such as antenna arrays, Lidars, or cameras [11], [16–20], [22–24], [30].

ii. This approach does not require any cooperative communications between multiple receiver nodes and does not assume available backup radio connections with the control station [14, 16, 20, 23, 25].

iii. This solution is unaffected by weather or light conditions and does not require high computational resources in comparison to those that involve data from visual sensors [11], [21–27], [30–31].

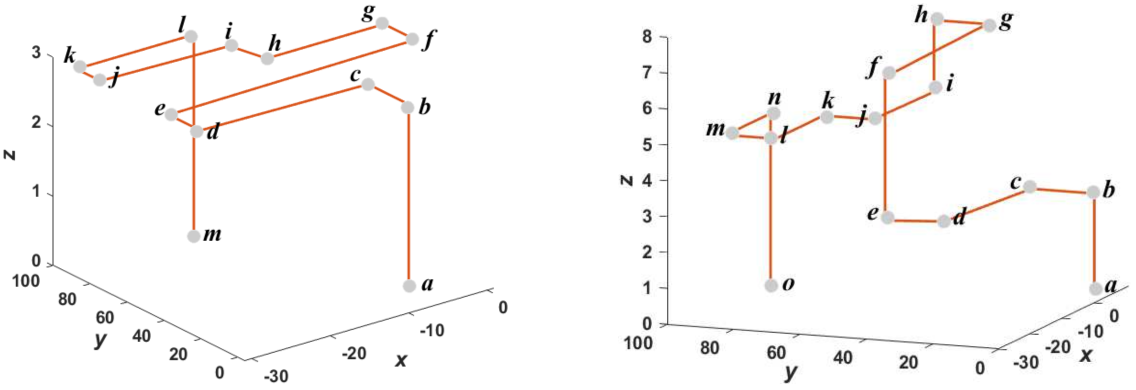
iv. This approach is based on datasets of feature samples collected from experimental setups under different flightpath scenarios. These datasets are used to develop realistic B-LSTM models that take into account radiometric factors which are often overlooked in simulation setups [19-20], [22, 25], [28–29].

The remaining of this article is organized as follows: Section 2 presents the experimental feature extraction setup for two flightpath scenarios as well as the associated extracted feature samples. Section 3 elaborates on the proposed RTH approach and includes an evaluation of two developed B-LSTM architectures. Finally, conclusions and future work are provided in Section 4.

2. Experimental Feature Extraction Setup

Figure 1 depicts the flightpaths for two experimental scenarios designed for extracting signal features and collecting samples. Such flightpaths are performed in an open field with a moderate presence of physical structures (e.g., dormitory buildings). Both scenarios are designed with QGroundControl software to ensure no damages occur to surrounding infrastructure during autonomous navigation [32]. This is realized via defining digital fences that the UAV cannot bypass as well as through precise calibration of onboard sensors and careful planning of the flightpaths. It is noteworthy to point out that the calibration of onboard sensors ensures accurate readings and stable navigation. A demonstration video for creating a flightpath can be found in [33]. The UAV used in these scenarios is an open-source drone from COEX that is equipped with a u-blox M8 GPS receiver and a PX4 flight controller for logging sensory measurements throughout the flights [34]. The adopted QGround Control software, GPS receiver, and flight controller are found in many commercial applications and research-grade equipment [35–37]. Scenario 1 characterizes a flightpath with fixed altitude and ground velocity; whereas Scenario 2 conveys an altitude and ground velocity that vary across the flightpath, as described in Table 1. The proposed RTH approach benefits from the readily available logged GPS data, sensory measurements (i.e., barometer, magnetometer), and inertial measurement unit (IMU) measurements (i.e., accelerometer, gyroscope) before GPS signal is compromised. Hence, flightpaths are launched without interfering with the GPS reception or other sensor measurements.

The accelerometer and gyroscope onboard sensors are utilized for measuring the rate of change in velocity and angle (i.e., roll, pitch, yaw), respectively, around the three axes. On the other hand, the barometer measures air pressure to assist in determining the altitude, while the magnetometer detects the earth's magnetic field to facilitate direction information. Finally, the GPS sensor provides location information to the UAV (e.g., longitude, latitude, altitude). It is worth pointing out that the GPS, barometer, magnetometer, and IMU measurements are processed with an extended Kalman filter (EKF) to obtain local position estimates [38]. A total of 10 features, showcased in Table 2, are logged with 989 samples per feature collected for scenario 1 and 739 samples per feature collected for scenario 2. The features are chosen for their attribute to offering a B-LSTM based RTH navigation solution. The timestamp provides the temporal context, while the onboard accelerometer and gyroscope sensors are utilized for measuring the rate of change in velocity and angle (i.e., roll, pitch, yaw), respectively, around the three axes. On the other hand, the barometer measures air pressure to assist in determining the altitude, while the magnetometer detects the earth's magnetic field to facilitate direction information.



(*a*) (*b*)

Figure 1. The flightpaths of the experimental scenarios: (*a*) scenario 1 with fixed altitude and ground velocity over the flightpath course, and (*b*) scenario 2 with varied altitude and ground velocity over the flightpath course.

Finally, the GPS sensor provides location information to the UAV (e.g., longitude, latitude, altitude). These are fundamental parameters for determining the UAV's position and orientation and allow for predicting a precise reverse flightpath. The developed B-LSTM model exploits all available samples (i.e., in this context, flight initiation to GPS signal loss). Consequently, there is no splitting of that dataset into training and testing sets. To avoid overfitting, 10% of the data is dropped during training. Collected samples (i.e., training datasets) can be found in [39].

3. RTH Solution Development and Evaluation

In Section 3.1, the development of the single and parallel B-LSTM architectures is elaborated, whereas their performance evaluations with the use of the mean square error (MSE) are detailed in Section 3.2.

Table 1: Altitudes and ground velocities for the two experimental scenarios illustrated in Figure 1.

Scenario 1			**Scenario 2**		
Waypoint	**Altitude (m)**	**Velocity (m/s)**	**Waypoint**	**Altitude (m)**	**Velocity (m/s)**
a	0	0	*a*	0	0
b-k	3	1	*b-e*	3	1
m	0	0	*f-h*	7	2
-	-	-	*i-n*	5	3
-	-	-	*o*	0	0

Table 2: The extracted features from onboard sensors.

Feature	**Representation**	**Source**	**Unit**
Timestamp since system starts	T	Flight controller	µs
North position in NED* earth-fixed frame	x	EKF	m
East position in NED earth-fixed frame	y	EKF	m
Down position (negative altitude) in NED earth-fixed frame	z	EKF	m
Acceleration in FRD** board frame *x*-axis	A_x	Accelerometer	m/s^2
Acceleration in FRD board frame *y*-axis	A_y	Accelerometer	m/s^2
Acceleration in FRD board frame *z*-axis	A_z	Accelerometer	m/s^2
Angular velocity in FRD board frame *x*-axis	G_x	Gyroscope	rad/s
Angular velocity in FRD board frame *y*-axis	G_y	Gyroscope	rad/s
Angular velocity in FRD board frame *z*-axis	G_z	Gyroscope	rad/s

(*) *x*-north, *y*-east, and *z*-down
(**) *x*-forward, *y*-*r*ight, and *z*-down

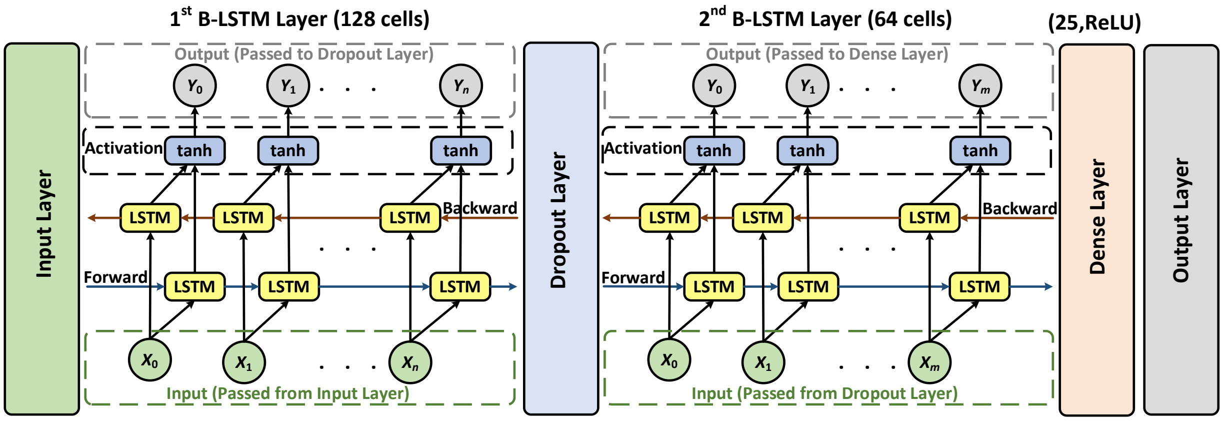

1st B-LSTM Layer (128 cells)
2nd B-LSTM Layer (64 cells)
(25,ReLU)
Input Layer
Output (Passed to Dropout Layer)
Y_0
Y_1
Y_n
Activation
tanh
LSTM
Backward
Forward
X_0
X_1
X_n
Input (Passed from Input Layer)
Dropout Layer
Output (Passed to Dense Layer)
Y_m
X_m
Input (Passed from Dropout Layer)
Dense Layer
Output Layer

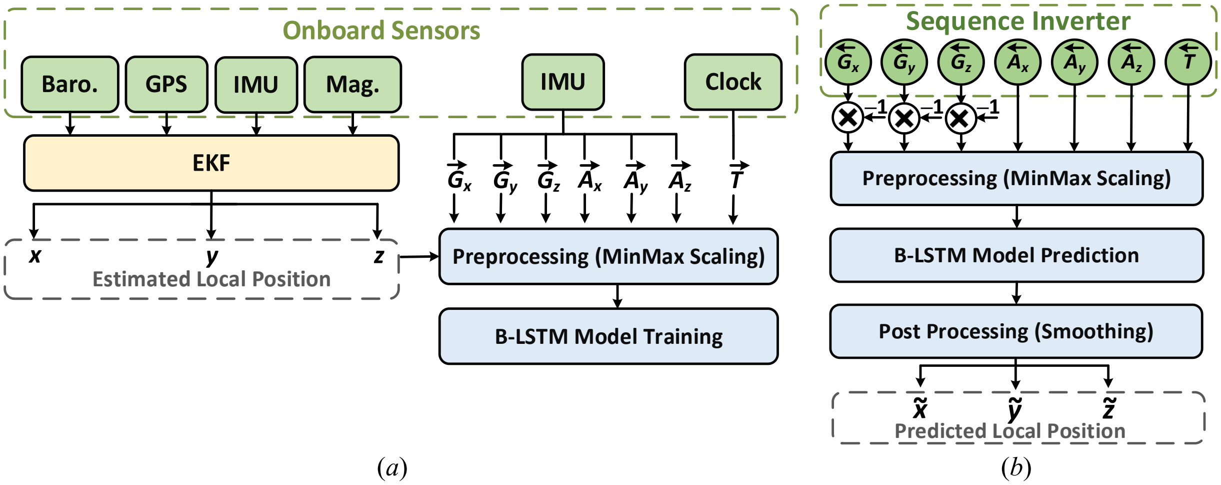
Figure 2. The configuration of the proposed B-LSTM deep learning model: "*n*" is the input/output sequence length in the 1st B-LSTM layer and "*m*" is the input/output sequence length in the 2nd B-LSTM layer after dropout ($m<n$).

3.1 Development of the Bidirectional Long Short-term Memory Architectures

Figure 2 illustrates the configuration of a B-LSTM model, which conveys an input layer for receiving, scaling, and reshaping the feature samples. Such a layer is followed by a B-LSTM layer with 128 LSTM cells and return sequences. Sequence returning enables subsequent layers to access information from each time step of the input sequence, allowing the model to effectively capture temporal patterns and relationships in the data. The bidirectional aspect of the LSTM model is concerned with processing the input sequence in both forward direction and backward direction, unlike unidirectional models, which allows for capturing and extracting more temporal dependencies [40]. The output of the aforementioned layer is passed to a dropout layer for ensuring proper fitting considering an optimized dropout rate. In each training iteration, this layer randomly drops 10% of the cells output before they are passed to the next layer. Then, a second B-LSTM layer with 64 cells and no return sequences is incorporated for further processing the data and combining the information into a single hidden state vector. The resulting output is processed by a dense layer with 25 neurons and a ReLU activation function to introduce non-linearity to the model and allow for learning complex patterns and relationships within the data. This layer is connected to another dense layer that serves as the output layer. Figure 3 shows the proposed RTH approach with a single B-LSTM architecture, which consists of forward sequence training and backward sequence prediction, as follows:

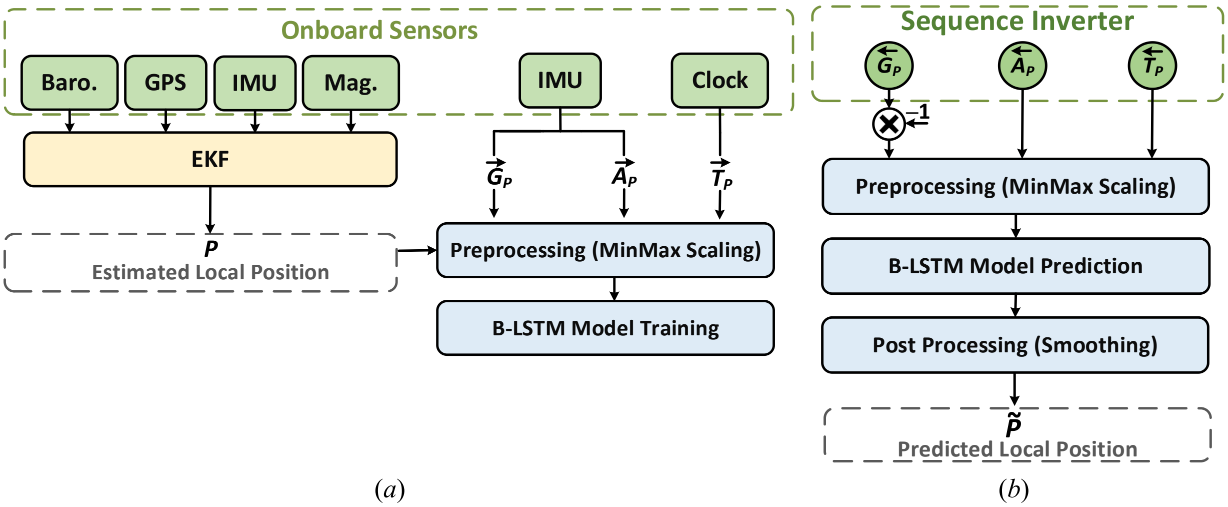
a. *Forward sequence training*: During the flight, the UAV logs multiple feature samples, including the accelerometer and gyroscope measurements, timestamp, and local position estimates obtained via the EKF. These samples are recorded at constant predefined rates until the GPS reception is no longer available. Then, the accelerometer, gyroscope, and timestamp measurements are used as an input vector, while the logged local position features estimated by the EKF are used as a target output. Logged samples are scaled before training to learn the relationship between the UAV sensory data and its position.

b. *Backward sequence prediction*: The logged feature samples are processed such that the timestamp and accelerometer sequence measurements are inverted; whereas the gyroscope sequence measurements are inverted and multiplied by –1 to reverse angular variations for achieving a mirrored trajectory. This procedure entails reversing the direction and movement sequence of the UAV, thereby enabling a return flightpath. The processed samples are fed to the flight controller as target readings to adjust the flightpath (i.e., altitude, velocity) accordingly, which is facilitated by the offboard mode on the flight controller. Simultaneously, the samples are scaled and used as an input to a B-LSTM configuration to predict the UAV local position. Finally, postprocessing is incorporated to refine predictions and eliminate noise.


Onboard Sensors
Baro.
GPS
IMU
Mag.
IMU
Clock
EKF
$\vec{G}_x$ $\vec{G}_y$ $\vec{G}_z$ $\vec{A}_x$ $\vec{A}_y$ $\vec{A}_z$ $\vec{T}$
x y z
Estimated Local Position
Preprocessing (MinMax Scaling)
B-LSTM Model Training
Sequence Inverter
$\overleftarrow{G}_x$ $\overleftarrow{G}_y$ $\overleftarrow{G}_z$ $\overleftarrow{A}_x$ $\overleftarrow{A}_y$ $\overleftarrow{A}_z$ $\overleftarrow{T}$
−1
Preprocessing (MinMax Scaling)
B-LSTM Model Prediction
Post Processing (Smoothing)
$\tilde{x}$ $\tilde{y}$ $\tilde{z}$
Predicted Local Position

(*a*) (*b*)

Figure 3. Proposed RTH approach with a single B-LSTM architecture: (*a*) forward sequence training, and (*b*) backward sequence prediction.


Onboard Sensors
Baro.
GPS
IMU
Mag.
IMU
Clock
EKF
$\vec{G}_P$ $\vec{A}_P$ $\vec{T}_P$
P
Estimated Local Position
Preprocessing (MinMax Scaling)
B-LSTM Model Training
Sequence Inverter
$\overleftarrow{G}_P$ $\overleftarrow{A}_P$ $\overleftarrow{T}_P$
−1
Preprocessing (MinMax Scaling)
B-LSTM Model Prediction
Post Processing (Smoothing)
$\tilde{P}$
Predicted Local Position

(*a*) (*b*)

Figure 4. Proposed RTH approach with a parallel B-LSTM architecture: (*a*) forward sequence training, and (*b*) backward sequence prediction. *P* denotes *x*-, *y*-, or *z*-dimension.

Figure 4 represents the proposed RTH approach with a parallel B-LSTM architecture. Here, three separate B-LSTM configurations are employed, one for each local coordinate denoted as *P* (for *x*, *y*, or *z*). Each configuration receives inputs specific to its assigned coordinate and produces a single output: *x*, *y*, or *z*. The B-LSTM configurations in both architectures are modeled considering the MSE as loss function. To this end, combinations of empirical experimentations and grid search optimizations are performed to optimize the hyperparameters; particularly, number of layers, number of cells in each layer, and dropout rate. It is worth noting that a fixed dropout rate is adopted in this research as it balances between modeling complexity and accuracy, which is of a grave importance, especially for resource-constraint applications (e.g., UAVs with limited processing and power supply capacities). MinMax scaling is adopted over standard scaling during training as it normalizes input features to a predefined range. This contributes to maintaining consistent magnitudes and improving stability against exploding or vanishing gradients, making it more suitable for handling sequential and temporal data. The optimized hyperparameters are given in Table 3. Figure 5 depicts the learning curves of the developed architectures. In both scenarios, the parallel architecture achieves lower average loss values as compared to the single architecture.

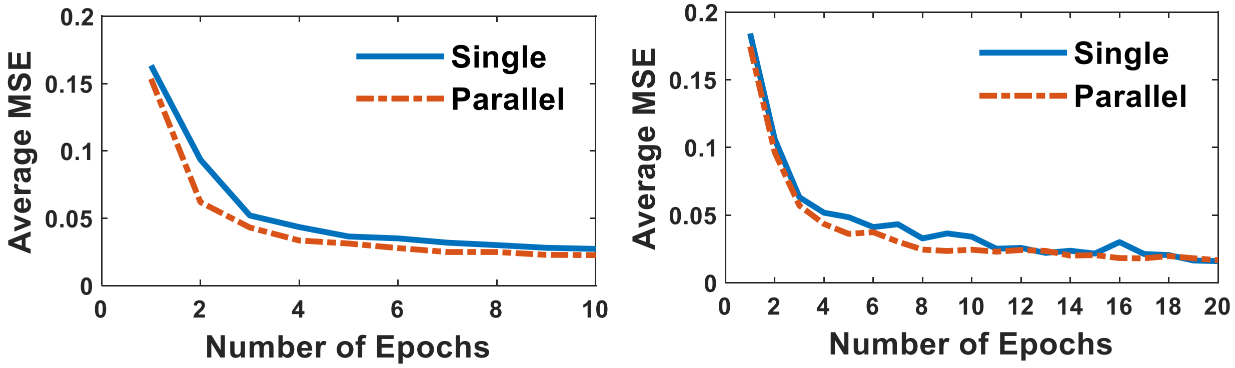
Table 3: Optimized hyperparameters of the developed B-LSTM architectures.

Hyperparameter	Single Architecture	Parallel Architecture
No. of B-LSTM layers	2	2
No. of cells in each B-LSTM layer	128/64	128/64
Number of neurons in each dense layer	25/3	25/1
Activation function	ReLU	ReLU
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Batch size	1	1
Number of epochs	10 (scenario 1)	10 (scenario 1)
	20 (scenario 2)	20 (scenario 2)
Dropout rate	0.1	0.1
Scaler	MinMax (–1, 1)	MinMax (–1, 1)

3.2 Model Evaluation

The MSE is used to evaluate the two developed architectures. In this evaluation, three cases are investigated according to the location where the GPS reception is assumed no longer available. Such cases, w.r.t. Figure 1, are *Case* 1: GPS reception loss occurs at the end of the flightpath, *Case* 2: GPS reception loss occurs after waypoint *i-a* (scenario 1) or *k-a* (scenario 2) of the path is executed, and *Case* 3: GPS reception loss occurs after waypoint *e-a* of the flightpath is executed. This investigation allows for examining the impact of the dataset size and altitude/velocity variations on the training of the architectures. A moving average filter is adopted as a postprocessing stage to improve the proposed RTH solution and mitigate the impact of sensor noise. Figures 6 and 7 illustrate the predicted trajectories in comparison to the planned and executed trajectories for the three GPS signal loss cases in each of scenarios 1 and 2 with and without incorporating the filter. The resulting predictions are in proximity to those executed by the UAV, with MSEs detailed in Tables 4 and 5.

The evaluation scores show that the parallel B-LSTM architecture has a lower average MSE in comparison to the single B-LSTM architecture, especially in cases 2 and 3 that exhibit lower training dataset sizes. The improved MSE performance of the parallel architecture is attributed to the reduced number of input-output features that this architecture processes. In other words, the single architecture processes the inputs-outputs of the three dimensions (i.e., *x*-, *y*-, and *z*-axes) simultaneously with one B-LSTM model to predict a three-dimensional local position. On the other hand, the parallel architecture employs three separate models; each is dedicated to processing features and predicting the local position of a single dimension (i.e., *x*, *y*, or *z*). This performance, however, comes at the expense of increased training and prediction times as indicated in Table 6.



(*a*) (*b*)

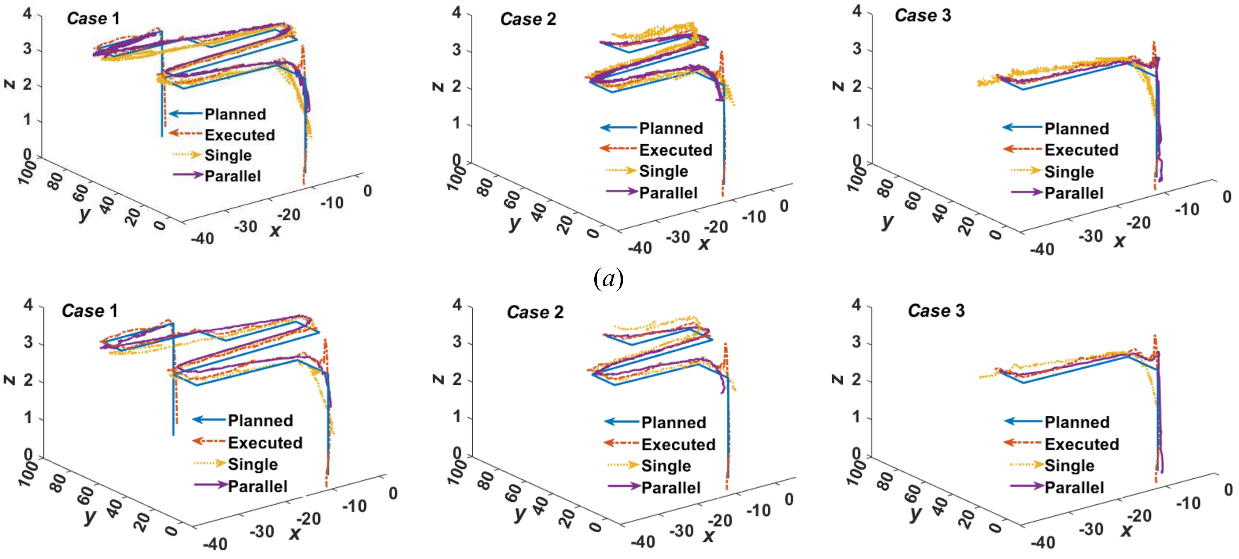
Figure 5. The learning curve of the developed B-LSTM models: (*a*) Scenario 1, and (*b*) Scenario 2.

Table 4: Scenario 1 evaluation scores for the developed architectures.

	MSE (Single B-LSTM Architecture)			MSE (Parallel B-LSTM Architecture)		
	x-axis (m)	*y*-axis (m)	*z*-axis (m)	*x*-axis (m)	*y*-axis (m)	*z*-axis (m)
Case* 1: *m-a	2.63	7.88	0.21	3.05	6.74	0.15
Case* 2: *i-a	5.62	3.78	0.19	1.24	6.08	0.19
Case* 3: *e-a	5.16	29.12	0.30	0.22	4.45	0.04
Average MSE	4.48	13.59	0.23	1.50	5.76	0.13

Table 5: Scenario 2 evaluation scores for the developed architectures.

	MSE (Single B-LSTM Architecture)			MSE (Parallel B-LSTM Architecture)		
	x-axis (m)	*y*-axis (m)	*z*-axis (m)	*x*-axis (m)	*y*-axis (m)	*z*-axis (m)
Case* 1: *o-a	4.53	3.62	0.24	2.24	7.95	0.38
Case* 2: *k-a	3.24	3.06	0.24	1.21	2.76	0.46
Case* 3: *e-a	0.32	0.74	0.12	0.08	0.28	0.06
Average MSE	2.70	2.47	0.20	1.17	3.67	0.30

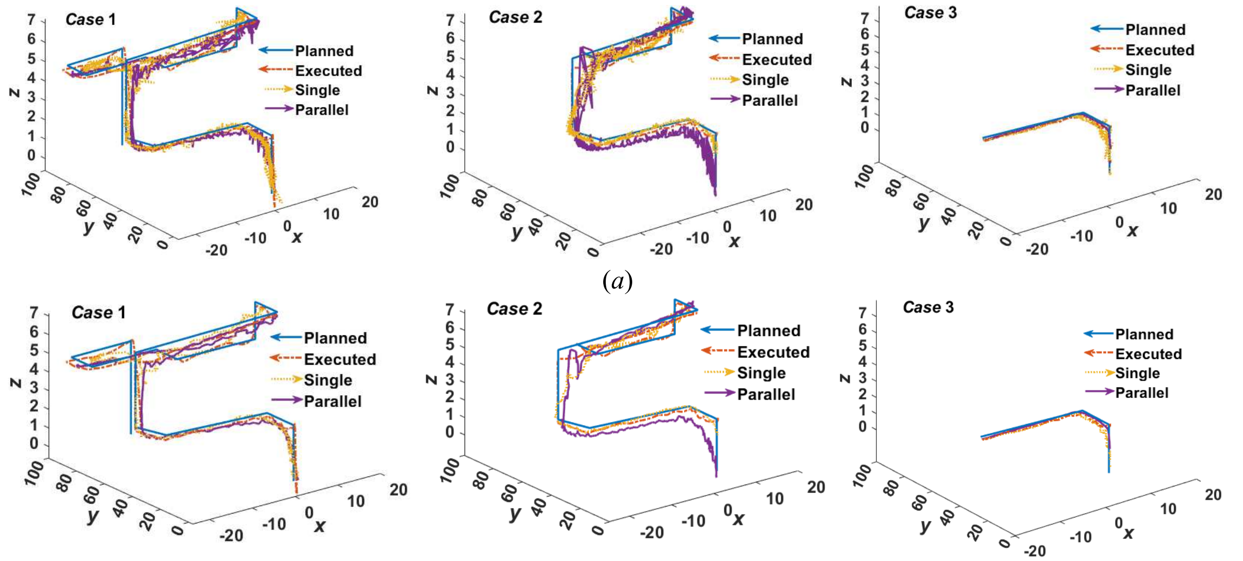


(*a*)

(*b*)

Figure 6. Scenario 1 comparative flightpath performance for the developed architectures: (*a*) cases 1–3 without incorporating a moving average filter, and (*b*) cases 1–3 with incorporating a moving average filter.

It is noteworthy to point out that the reported times and memory are from averaging five independent executions per architecture. The prediction time per sample set in scenario 2 (*Case* 1) is 1.1 ms, which results from dividing the overall prediction time (i.e., 0.87 second) by the total number of samples in the dataset (i.e., 739 samples). This demonstrates the timely prediction of the proposed solution without necessitating any parallelization techniques (i.e., prediction time less than GPS sampling time of 100–1000 ms). Scenario 2, which characterizes varying altitudes and velocities, demonstrates an improved MSE as compared to Scenario 1. This variation facilitates a dynamic dataset that enables the model to better capture complex relationships within features and leads to improved generalizations. Table 6 also illustrates a comparative analysis benchmarking the performance of each architecture with the use of conventional unidirectional LSTM layers and stacked unidirectional LSTM layers (i.e., two stacks) instead of the B-LSTM layers in the configuration presented in Figure 2. Such an analysis shows that the utilization of B-LSTM layers improves the average MSE. Nevertheless, it increases the overall training and prediction times as well as the required memory.



(*a*)

(*b*)

Figure 7. Scenario 2 comparative flightpath performance for the developed architectures: (*a*) cases 1–3 without incorporating a moving average filter, and (*b*) cases 1–3 with incorporating a moving average filter.

Table 6: Comparison between the proposed architectures with different LSTM configurations considering *Case* 1.

		Single Architecture			**Parallel Architecture**		
		LSTM	**Stacked LSTM**	**B-LSTM**	**LSTM**	**Stacked LSTM**	**B-LSTM**
Avg. MSE±Dev (m)	Scenario 1	5.84±2.40	5.80±4.83	3.59±0.76	5.16±0.71	4.77±1.89	3.82±0.39
	Scenario 2	3.68±2.03	4.08±1.86	3.13±0.89	3.13±1.23	3.04±1.17	4.02±0.51
Training (s)	Scenario 1	124.39	158.55	220.30	240.32	381.43	435.97
	Scenario 2	201.93	239	293.12	401.31	518.78	542.64
Prediction (s)	Scenario 1	0.33	0.44	0.72	0.67	0.88	1.31
	Scenario 2	0.29	0.35	0.43	0.66	0.74	0.87
Memory (MB)	Scenario 1	48.25	58.97	78.92	36.50	65.35	90.14
	Scenario 2	48.83	56.25	96.59	35.52	65.12	78.69

Therefore, the proposed RTH approach offers a flexible offline navigation solution for meeting the requirements of a given application. For example, short autonomous missions with low memory requirements can benefit from a higher location accuracy. On the other hand, long missions with high memory expectations can adopt the proposed approach with a marginal impact, characterized by an increase in location error. All the developed architectures presented in this article can be found in [39]. Finally, a comparison between the proposed approach and other state-of-the-art methods that facilitate navigation solutions in GPS-denied environments is provided in Table 7. The proposed approach offers a unique solution for a reliable and autonomous RTH under sudden losses of GPS reception. This solution leverages the readily available measurements of onboard sensors. Thus, it does not incorporate extra hardware components (e.g., antenna arrays, LiDARs, cameras) that are often required to facilitate visual-assisted GPS-free navigation. While these added components can improve performance, the resulting navigation accuracy is susceptible to external channel conditions (i.e., weather, lighting). The proposed approach also offers a low computational overhead as compared to other solutions that adopt cumbersome and resource-consuming image processing routines. In addition, the proposed solution is validated with UAV-specific data obtained via realistic experiments rather than laboratory simulations. Lastly, the approach proposed herein does not require supporting infrastructure (i.e., transceiver nodes with access to GPS), and most importantly, does not assume any cooperative communications between multiple transceiver nodes.

Table 7: Comparison between the proposed approach and other state-of-the-art research for UAV applications.

Ref.	Data source	Utilized Hardware	Adopted Approach	Offered Solution
[16]	Experiments	antenna arrays	trilateration	RTH
[19]	Simulations	RIS	reinforcement learning	path planning
[24]	Experiments	camera, IMU, sonar	EKF with SLAM*	GPS-free navigation
[25]	Experiments	altimeter, camera, IMU, Lidar	UKF** with SLAM	GPS-free navigation
[26]	Experiments	barometer, IMU, range finder, magnetometer, optical sensor	EKF	GPS-free navigation
This work	Experiments	barometer, GPS, IMU, magnetometer	EKF with B-LSTM	RTH

(*) Simultaneous localization and mapping
(**) Unscented Kalman filter

Overall, the solution presented herein is developed with realistic experimental setups. It also features the advantages of reduced cost and improved robustness to varying navigation environments as well as flightpaths (altitudes, velocities). Such advantages makes it suitable for resource-constrained platforms that benefit from autonomous navigation.

4. Conclusion and Future Work

To conclude, a B-LSTM approach is utilized as an RTH autonomous navigation solution in the absence of GPS reception. Two architectures are trained for predicting the path of an already executed mission with the use of logged samples from ten features. To validate this approach, samples were collected in two experimental scenarios comprising different altitudes and ground velocities. Evaluation results showcased the efficacy of the proposed approach, yielding an average MSE as low as 1.17 m, 3.67 m, and 0.30 m w.r.t the x-axis, y-axis, and z-axis, respectively, with an average MSE uncertainty of ±0.89 m. Furthermore, the training of the proposed B-LSTM architectures utilized a memory of less than 100 MB and exhibited a timely prediction close to 1 ms per sample set. This approach does not require additional hardware; rather, samples from sensors that are commonly found in autonomous aerial vehicles are exploited, which makes it attractive and scalable. Future work will entail exploring the integration of LSTM configurations with reinforcement learning to develop a hybrid navigation mechanism for tactical, mission-critical applications. The environment in such applications is dynamic in the sense that a forward flightpath may not include threats (i.e., physical obstacles, cyberattacks), while during performing the RTH reverse flightpath, the presence of these threats may compromise the vehicle. Hence, this integration is expected to significantly improve the robustness of the RTH navigation, especially in complex or unpredictable situations of unforeseen environments. In addition, there is a tradeoff between performance accuracy, energy efficiency, and memory in artificial intelligence-based models. Therefore, another future research may be concerned with investigating an analysis scheme that addresses real-time energy consumption, model precision, and memory in different use cases, such as multiple channel conditions, dataset sizes, and preprocessing methods. Finally, the adaptation of the proposed RTH solution to other types of autonomous vehicles (e.g., ground vehicles) has the potential to impact the transportation industry, traffic management, as well as safety and accessibility.

Acknowledgement

This research is funded by the National Science Foundation, Secure and Trustworthy Cyberspace Program, under award no. 2006662.

References

[1] S. Hayat, E. Yanmaz, and R. Muzaffar, “Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint,” *IEEE Communications Surveys & Tutorials*, vol.18, no. 4, pp. 2624–2661, 2016.

[2] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, “A compilation of UAV applications for precision agriculture,“ *Computer Networks*, vol. 172, p.107148, 2020.

[3] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, “Unmanned aerial vehicles applications in future smart cities,” *Technological Forecasting and Social Change*, vol. 153, 2020.

[4] UAV (Drone) market [Online]. Available:https://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html.

[5] J. Zidan, E. Adegoke, E. Kampert, S. Birrell, C. Ford, and M. Higgins, “GNSS Vulnerabilities and Existing Solutions: A Review of the Literature,” *IEEE Access*, vol. 9, pp. 153960–153976, 2021.

[6] P. Craven, R. Wong, N. Fedora, and P. Crampton, “Studying the effects of interference on GNSS signals,” *International Technical Meeting of the Institute of Navigation*, pp. 893–186, 2013.

[7] A. Hussain, F. Akhtar, Z.H. Khand, A. Rajput, and Z. Shaukat, Z, “Complexity and limitations of GNSS signal reception in highly obstructed environments,” *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp.6864–6868, 2021.

[8] F. Zangenehnejad and Y. Gao, “GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives,” *Satellite navigation*, vol. 2, pp. 1–23, 2021.

[9] L.N. Thin, L.Y. Ting, N.A. Husna, and M.H. Husin, “GPS systems literature:Inaccuracy factors and effective solutions,” *International Journal of Computer Networks & Communications*, vol. 8, no. 2, pp. 123–131, 2016.

[10] T. Kos, I. Markezic, and J. Pokrajcic, “Effects of multipath reception on GPS positioning performance,” *ELMAR Proceedings*, pp. 399–402, 2010.

[11] J. Meguro, T. Murata, J. Takiguchi, Y. Amano, and T. Hashizume, “GPS multipath mitigation for urban area using omnidirectional infrared camera,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 22–30, 2009.

[12] K. Yedukondalu, A. Sarma, and S. Vemuri, “Estimation and mitigation of GPS multipath interference using adaptive filtering,” *Progress in electromagnetics research m*, vol. 21, pp.133–148, 2011.

[13] Q. Phan, S. Tan, and I. McLoughlin, “GPS multipath mitigation: a nonlinear regression approach,” *GPS solutions*, vol. 17, no. 3, pp. 371–380, 2013.

[14] Y. Yuan, F. Shen, and X. Li, “GPS multipath and NLOS mitigation for relative positioning in urban environments,” *Aerospace Science and Technology*, vol. 107, pp. 106315, 2020.

[15] A. Novák, K. Kováčiková, B. Kandera, and A. Sedláčková, “Global navigation satellite systems signal vulnerabilities in unmanned aerial vehicle operations: Impact of affordable software-defined radio,” *Drones*, vol. 8, no. 3, pp. 109, 2024.

[16] B. Van den Bergh and S. Pollin, “Keeping UAVs under control during GPS jamming,” *IEEE Systems Journal*, vol. 13, no. 2, pp. 2010–2021, 2018.

[17] A. Osman, M. Moussa, M. Tamazin, M. Korenberg, and A. Noureldin, “DOA elevation and azimuth angles estimation of GPS jamming signals using fast orthogonal search,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 5, pp. 3812–3821, 2020.

[18] Y. Chien, “Design of GPS anti-jamming systems using adaptive notch filters,” *IEEE Systems Journal*, vol. 9, no. 2, pp. 451–460, 2013.

[19] S. Hu, X. Yuan, W. Ni, X. Wang, and A. Jamalipour, “RIS-assisted jamming rejection and path planning for UAV-borne IoT platform: A new deep reinforcement learning framework*,” IEEE Internet of Things Journal*, vol. 10, no. 22, 2023.

[20] M. S. Siraj, A. Rahman, M. Diamanti, E. Tsiropoulou, and S. Papavassiliou, “Alternative positioning, navigation, and timing enabled by games in satisfaction form and reconfigurable intelligent surfaces,” *IEEE Systems Journal*, vol. 17, no. 3, pp. 5035–5046, 2023.

[21] A. Couturier and M. Akhloufi, “A review on absolute visual localization for UAV,” *Robotics and Autonomous Systems*, vol. 135, pp. 103666, 2021.

[22] F. Vanegas, K. Gaston, J. Roberts, and F. Gonzalez, “A framework for UAV navigation and exploration in GPS-denied environments,” *IEEE aerospace conference*, pp. 1–6, 2019.

[23] Y. Tang, Y. Hu, J. Cui, F. Liao, M. Lao, F. Lin, and R. Teo, “Vision-aided multi-UAV autonomous flocking in GPS-denied environment,” *IEEE Transactions on industrial electronics*, vol. 66, no. 1, pp. 616–626, 2018.

[24] F. Valenti, D. Giaquinto, L. Musto, A. Zinelli, M. Bertozzi, and A. Broggi, “Enabling computer vision-based autonomous navigation for unmanned aerial vehicles in cluttered gps-denied environments,” *International Conference on Intelligent Transportation Systems*, pp. 3886–3891, 2018.

[25] J. Hardy, J. Strader, J. Gross, Y. Gu, M. Keck, J. Douglas, and C. Taylor, “Unmanned aerial vehicle relative navigation in GPS denied environments,” *IEEE/ION Position, Location and Navigation Symposium*, pp. 344–352, 2016.

[26] H. Lu, H. Shen, B. Tian, X. Zhang, Z. Yang, and Q. Zong, “Flight in GPS-denied environment: Autonomous navigation system for micro-aerial vehicle,” *Aerospace Science and Technology*, vol. 124, pp. 107521, 2022.

[27] S. Ashraf, P. Aggarwal, P. Damacharla, H. Wang, A. Javaid, and V. Devabhaktuni, “A low-cost solution for unmanned aerial vehicle navigation in a global positioning system–denied environment,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 6, pp. 1–17, 2018.

[28] Y. Xue and W. Chen, “A UAV navigation approach based on deep reinforcement learning in large cluttered 3D environments,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3001–3014, 2023.

[29] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud, “Autonomous UAV Navigation: A DDPG-based deep reinforcement learning approach,” *IEEE International Symposium on Circuits and Systems* (*ISCAS*), pp. 1–5, 2020.

[30] F. Mumuni, A. Mumuni, and C. Amuzuvi, “Deep learning of monocular depth, optical flow and ego-motion with geometric guidance for UAV navigation in dynamic environments,” *Machine Learning with Applications*, vol. 10, p.100416, 2022.

[31] A. Deraz, O. Badawy, M. Elhosseini, M. Mostafa, H. Ali, and A. El-Desouky, “Deep learning based on LSTM model for enhanced visual odometry navigation system,” *Ain Shams Engineering Journal*, vol. 14, no. 8, p.102050, 2023.

[32] QroundControl [Online]. Available: QGC - QGroundControl - Drone Control

[33] Flightpath creation [Online]. Available: Introduction to QGroundControl

[34] COEX Clover [Online]. Available: Introduction · Clover (coex.tech)

[35] L. Tirado, L. Vinces, and J. Ronceros, “An interface based on QGroundControl for the rapid parameterization of flights from an embedded system for the control of an inspection drone,” *Congreso Internacional de Innovación y Tendencias en Ingeniería* (*CONIITI*), pp. 1–5, 2022.

[36] C. Mongrédien, J. Doyen, M. Stromand, and D. Ammann, “Centimeter-level positioning for UAVs and other mass-market applications,” *Proceedings of the 29th International Technical Meeting of the Satellite Division of the Institute of Navigation* (*ION GNSS+*), pp. 1441–1454, 2016.

[37] E. Ebeid, M. Skriver, K. Terkildsen, K. Jensen, and U. Schultz, “A survey of open-source UAV flight controllers and flight simulators,” *Microprocessors and Microsystems*, vol. 61, pp. 11–20, 2018.

[38] EKF [Online]. Available: https://docs.px4.io/main/en/advanced_config/tuning_the_ecl_ekf.html

[39] Developed Datasets and Models [Online]. Available: https://github.com/sherlock9999/BI-LSTM.git

[40] B.Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, “A survey on long short-term memory networks for time series prediction,” *Procedia Cirp*, vol. 99, pp. 650–655, 2021.