drones

MDPI

Article

Artificial Intelligence-Driven Aircraft Systems to Emulate Autopilot and GPS Functionality in GPS-Denied Scenarios Through Deep Learning

César García-Gascón [1,2,*], Pablo Castelló-Pedrero [1,2], Francisco Chinesta [2,3] and Juan A. García-Manrique [1]

[1] Design and Manufacturing Research Institute, Universitat Politècnica de València, 46022 Valencia, Spain; pabcaspe@idf.upv.es (P.C.-P.); jugarcia@upv.es (J.A.G.-M.)
[2] PIMM Laboratory, Arts et Métiers Institute of Technology, CNRS, ENSAM, 75013 Paris, France; francisco.chinesta@ensam.eu
[3] CNRS@CREATE Ltd., 1 Create Way, #08-01 CREATE Tower, Singapore 138602, Singapore
* Correspondence: cegarga3@etsid.upv.es

Abstract: This paper presents a methodology for training a Deep Learning model aimed at flight management tasks in a fixed-wing unmanned aerial vehicle (UAV), specifically autopilot control and GPS prediction. In this formulation, sensor data and the most recent GPS signal are first processed by an LSTM to produce an initial coordinate prediction. This preliminary estimate is then merged with additional sensor inputs and passed to an MLP, which replaces the conventional autopilot algorithm by generating the control commands for real-time navigation. The approach is particularly valuable in scenarios where the aircraft must follow a predetermined route—such as surveillance operations—or maintain a remote ground link under varying GPS availability. The study focuses on Class I UAVs; however, the proposed methodology can be adapted to larger classes (II and III) by adjusting sensor configurations and network parameters. To collect training data, a small fixed-wing aircraft was instrumented to record kinematic and control inputs, which then served as inputs to the neural network. Despite the limited sensor suite and the use of an open-source flight controller (SpeedyBee), the flexibility of the proposed approach allows for easy adaptation to more complex UAVs equipped with additional sensors, potentially improving prediction accuracy. The performance of the neural network, implemented in PyTorch, was evaluated by comparing its predicted data with actual flight logs. In addition, the method has been shown to be robust to both short and prolonged GPS outages, as it relies on waypoint-based navigation along previously explored routes, ensuring reliable performance in known operational contexts.

Keywords: artificial intelligence (AI); Deep Learning (DL); Unmanned Aerial Systems (UASs); unmanned aerial vehicle (UAV)

Academic Editors: Bo Li and Pablo Rodríguez-Gonzálvez

Received: 11 February 2025
Revised: 12 March 2025
Accepted: 24 March 2025
Published: 26 March 2025

Citation: García-Gascón, C.; Castelló-Pedrero, P.; Chinesta, F.; García-Manrique, J.A. Artificial Intelligence-Driven Aircraft Systems to Emulate Autopilot and GPS Functionality in GPS-Denied Scenarios Through Deep Learning. *Drones* **2025**, *9*, 250. https://doi.org/10.3390/drones9040250

1. Introduction

Recent years have seen significant growth in UAV technologies and artificial intelligence (AI), both for military and civilian purposes. Although these technologies have developed independently, their integration opens up almost limitless possibilities for new areas of research. Today, there is a wide and diverse range of UAV navigation systems that incorporate numerous sensors to enable autonomous navigation. It is important to highlight that each sensor or actuator has the capacity to record its behavior and position in a blackbox. This process captures a big amount of data, including various types of errors, such as the discrepancy between the actual position and the desired one, latency,

and the correct position of the aerodynamic actuators of the UAV. The recorded data can subsequently be utilized to train AI systems, thereby enabling them to model and compensate for these inaccuracies. These systems are widely employed in applications such as trajectory tracking, autonomous navigation, and scene recognition. Particularly, cameras play a crucial role in developing predictive control systems for target tracking by providing real-time visual data as input to the control algorithm [1–4]. However, mathematical modeling of UAVs remains highly sensitive to external disturbances, particularly in lightweight and low-speed UAVs, where environmental factors significantly impact flight stability. These disturbances are often treated as noise in the system, which is a significant challenge. Neural network-based control systems have been successfully used to model these disturbances with experimental data. For example, Dhayalan et al. [5] developed a neural network for the flight of a lightweight UAV by conducting six experimental flights and using mean and standard deviation to calculate errors. Their work shows that experimental flight information, when processed with statistical tools, can replace certain aircraft control systems and improve performance.

Traditional UAV control systems are based on linear control dynamics, but real flight conditions are clearly non-linear, which limits the capabilities of these systems. To overcome this problem, researchers have proposed DL techniques. For instance, Eivind Bohn et al. [6] have developed a system that enables control of fixed-wing aircraft with only three minutes of flight training. While this duration may seem short, modern flight controllers are capable of recording large amounts of data per second, allowing the neural network to learn efficiently from a condensed training period. Deep Reinforcement Learning (DRL) schemes are particularly effective for such problems. This technique has been used successfully in many applications that require decision making on large datasets, such as autonomous vehicles, robotic manufacturing, stock market investing, speech processing, medicine, games and advertising [7–10]. DRL techniques are particularly suited to highly complex problems where traditional control systems are inadequate, although they still lack sufficient safety guarantees for widespread use.

Most studies defining UAV flight parameters focus on multicopters in simulated environments, with few references to fixed-wing UAVs in real flight conditions [11]. Commercial autopilots such as Pixhawk, VECTOR-400, NAVIO2, Speedybee, MFD Crosshair, etc., have become widely available. These systems rely primarily on Proportional–Integral–Derivative (PID) loops to control pitch, roll and yaw. However, these autopilots do not account for high non-linearity, making them unreliable in complex flight scenarios. In addition, these controllers cannot model uncertainties related to wind gusts, thermal changes and other variables. Beyond conventional autopilots, robust or optimal multivariate control systems use linearization of UAV dynamics around one or more target values to improve performance [12,13].

The demand for greater autonomy and efficiency in UAVs has highlighted the importance of integrating autonomous navigation systems. Traditional UAV navigation systems rely on GPS, accelerometers, gyroscopes, and microcontrollers to stabilize and control flight [14,15]. However, these systems are prone to failure in environments with weak signals, interference, or complete GPS unavailability, such as dense urban areas, indoor settngs, or subterranean locations [16], necessitating alternative AI-based solutions. Recent research on UAV navigation has explored various methodologies to enhance autonomy and reliability, particularly in GPS-denied environments.

AI-driven approaches involve collecting flight data from conventional navigation systems to train models capable of replicating navigation decisions without relying on external positioning systems. By using Deep Neural Networks (DNNs) and Reinforcement Learning (RL), UAVs can learn to autonomously control flight, generating paths as if guided

by a human navigator [17]. AI-driven trajectory generation is a critical research area. While conceptually simple—moving from point A to point B—it requires interpreting multiple parameters, including environmental dynamics and real-time sensor inputs [5]. In addition, AI's adaptability allows it to learn and improve over time, with performance enhancements becoming evident after iterative training cycles that can span from a few minutes to several hours, depending on the complexity of the model and the amount of training data available.

In [18], the authors provide a comprehensive review of AI approaches applied to autonomous UAV navigation. The study highlights optimization techniques such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Genetic Algorithms (GA), and Grey Wolf Optimization (GWO), which improve trajectory planning in dynamic environments. Additionally, it explores DL-based methods, including Deep Q-Networks (DQNs), Convolutional Neural Networks (CNNs), and Deep Reinforcement Learning (DRL), which enhance obstacle avoidance and adaptive decision making in uncertain conditions.

The application of AI in UAV navigation has made significant progress, particularly in GPS-denied environments, where AI-driven methods such as Computer Vision, Deep Learning, and Sensor Fusion play a crucial role in enhancing autonomy and navigation accuracy [19].

Visual-based navigation is divided into Relative Visual Localization (RVL), which includes Visual Odometry (VO) and Simultaneous Localization and Mapping (SLAM), and Absolute Visual Localization (AVL), which relies on georeferenced datasets to improve accuracy and reduce drift. While RVL suffers from error accumulation over time, AVL requires high-quality reference data and robust matching algorithms. Traditional aerial localization relied on template matching and feature matching techniques, which improved robustness but were computationally expensive, limiting their real-time applicability in UAV operations [16]. The incorporation of DNNs has demonstrated great potential in enhancing state estimation, trajectory generation, and autonomous navigation of UAVs in complex scenarios [19].

Recently, in [20], the authors conduct an extensive review of the literature on DL-based methods for UAV AVL, covering significant advancements since 2019, highlighting how these methods have surpassed the limitations of traditional approaches based on manually extracted visual features. The study explains how DL models, particularly CNNs and Vision Transformers (ViTs), have surpassed classical computer vision approaches in UAV localization, offering increased robustness against variations in lighting, weather conditions, and terrain. Their review underscores the potential of learning-based AVL methods in providing accurate and drift-free positioning, making it a relevant reference for UAV applications in GPS-denied environments.

DL has significantly contributed to UAV navigation by enhancing feature extraction, object detection, and trajectory prediction. CNNs have been widely applied for geolocation and terrain recognition, while Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks improve obstacle avoidance and flight path optimization [21]. A significant study employs DRL to optimize UAV trajectories, allowing drones to adapt dynamically to environmental changes and unforeseen obstacles [22]. Al-Jarrah et al. [19] present a comprehensive review of DL-based visual localization techniques for UAVs operating in GPS-denied environments, highlighting the growing importance of AI-driven solutions for autonomous navigation. The study compares traditional visual localization methods, such as template matching and feature matching, with modern DL approaches, including CNNs, RNNs, Siamese Neural Networks (SNNs), and Generative Adversarial Networks (GANs).

The work conducted by Sarkar et Gul [23] provides a comprehensive survey of AI applications in UAV communications, localization, trajectory planning, and security. The study focuses particularly on AI-based localization techniques, which are crucial for UAV navigation in GPS-denied environments. Additionally, the survey addresses security concerns, emphasizing the use of CNNs and RNNs to enhance UAV cybersecurity and mitigate cyber-physical threats.

Other advancements integrate DL with visual SLAM to enhance feature extraction, target recognition, and path planning, thereby improving the stability and accuracy of UAV autonomous navigation in dynamic settings [21]. Recently, in [24], the authors present a Simultaneous Control, Localization, and Mapping (SCLAM) system based on monocular SLAM and real-time control strategies, providing an effective solution for UAV autonomous navigation in GPS-denied environments and laying the foundation for future improvements through AI techniques.

These developments underscore the critical role of ML, sensor fusion, and AI-driven decision making in ensuring robust and adaptive UAV operations [25]. The development of new hybrid architectures, combining traditional approaches with DL, will remain a key area of research in the coming years. Future applications include disaster response, where UAVs autonomously navigate hazardous terrain, precision agriculture, optimizing farming in large or GPS-denied areas, and urban planning, facilitating detailed aerial mapping even in restricted environments [21].

Advanced image processing and ML algorithms enable UAVs to detect and react to their environment in real time, which is crucial for obstacle avoidance, target identification, and following predefined or adaptive routes [26,27]. However, the algorithm proposed in this work does not rely on image processing but instead utilizes sensor-based data, such as kinematic and control parameters, to predict UAV trajectories and actuator commands.

This paper is a contribution to the field of UAV navigation, particularly in scenarios where GPS signals are unreliable or unavailable. To address this challenge, we propose a DL-based methodology comprising two neural networks: an LSTM network for GPS coordinate prediction and an MLP network for autopilot control. The LSTM model captures temporal dependencies in flight telemetry data to estimate missing GPS coordinates, while the MLP model translates these predictions into control surface commands, ensuring stable navigation. The proposed system is trained on real UAV flight data and validated through a five-fold cross-validation process to ensure robustness.

This work not only provides proof of concept for AI-based navigation systems, but also paves the way for future innovations. The data required to train and test the algorithm were obtained from the UAV's flight telemetry, which is performed using the Speedybee autopilot. Flight telemetry includes information such as the UAV's speed, GPS position, altitude, pitch, accelerations, and the autopilot signals sent to each of the servos (pitch, roll, and yaw), along with additional parameters like battery voltage or motor current. These parameters define the UAV's state and actuator behavior, encompassing its translational and rotational kinematics. The UAV dynamics are governed by non-linear equations of motion where external forces, such as aerodynamic forces and environmental disturbances, influence its trajectory. The model learns the relationship between these parameters and control inputs through real flight data, allowing it to estimate GPS coordinates and control surface positions in GPS-denied scenarios. By leveraging these kinematic relationships, the neural network adapts to changing flight conditions, ensuring stable navigation even in the absence of GPS.

A fixed-wing UAV weighing 2 kg and flying at reduced speed (around 8 m/s until 20 m/s) is used for the demonstration. The primary objective of this study is to validate the viability of the proposed methodology in enhancing operational safety during

waypoint-based missions by predicting GPS coordinates in cases where GPS signals are lost. Traditionally, maintaining flight stability in such missions relies on continuous GPS updates, whereas our approach leverages AI-based predictions to compensate for GPS outages. It is important to note that each specific mission, defined by a unique set of waypoints, would require retraining of the model under the same flight conditions, collecting new flight data to ensure accurate predictions. While the flights in this study were conducted under controlled conditions, the methodology itself is intended to be adaptable to real-world scenarios where GPS reliability is compromised.

Due to the UAV's weight and speed, external disturbances could significantly affect the measurements. The utilization of real flight data, in which such perturbations are inherently present, is a fundamental aspect of the methodology. Through the training of the neural network on flight telemetry that has been collected under varying environmental conditions, the network implicitly learns to compensate for these disturbances based on actual sensor data. This approach ensures that the model generalizes well to realistic operating conditions without requiring an explicit noise model. The UAV is controlled using non-linear dynamics, adding to the complexity of maintaining stability. It requires an aileron, an elevator, a motor, and a rudder to control roll, pitch, speed, and yaw. The aircraft is also equipped with altitude sensors, GPS and an inertial measurement unit (IMU). The data generated by the autopilot are recorded and made available in real time via the telemetry feed.

UAVs are broadly categorized into three classes based on weight, altitude and operational use. This study focuses on Class I UAVs (small platforms under 150 kg operating below 5000 ft AGL). However, the proposed methodology can be adapted to larger classes (II and III) by adjusting sensor configurations and network parameters.

The primary contribution of this work lies in the development of an AI-based navigation system that enables UAVs to autonomously predict GPS coordinates and control surfaces using DL. Unlike conventional autopilot systems that rely heavily on GPS, this approach leverages LSTM and MLP networks to maintain accurate trajectory control using exclusively sensor data, without any image processing, ensuring mission continuity in GPS-denied environments.

The rest of the paper is structured as follows. Section 2, the theoretical foundations of the neural network model used in this work are described. Section 3, the equipment and methodology used for the acquisition of numerical data are presented. Finally, the results obtained are presented and discussed in Section 4 and the conclusions are given in Section 5. Table 1 lists the acronyms and abbreviations used in this paper.

Table 1. Abbreviations/acronyms.

Acronym	Description
ACO	Ant Colony Optimization
ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
AVL	Absolute Visual Localization
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DL	Deep Learning
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
GA	Genetic Algorithm
GAN	Generative Adversarial Network
GPS	Global Positioning System
GWO	Grey Wolf Optimization
IMU	Inertial Measurement Unit

Table 1. *Cont.*

Acronym	Description
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
PID	Proportional-Integral-Derivative
PSO	Particle Swarm Optimization
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RVL	Relative Visual Localization
SCLAM	Simultaneous Control, Localization, and Mapping
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SNN	Siamese Neural Network
SURF	Speeded Up Robust Features
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
ViTs	Vision Transformers
VO	Visual Odometry

2. Theoretical Background

In this study, we propose a modular approach that leverages neural networks [28], both LSTM and MLP architectures, to generate control commands for UAV navigation. Specifically, the architecture processes sensor data and the last known GPS signal through an LSTM to generate an initial GPS coordinate estimate, which is then combined with the sensor information in an MLP. This MLP effectively replaces the traditional autopilot algorithm by providing the control commands for the aerodynamic surfaces. The model is designed to ensure robust real-time navigation by processing and integrating sensor data with GPS information. The approach is broken down in two steps:

1. Initial GPS Prediction via LSTM: The LSTM layer [29] uses the last known GPS position and sequential sensor data to predict an initial estimate of GPS coordinates (latitude and longitude), effectively substituting for missing GPS data. The neural network's input layer comprises 49 features extracted from the UAV's flight telemetry data. These features include sensor readings such as altitude, airspeed, accelerations, control surface positions, and target navigation waypoints, among others, as described in Section 4.
2. Autopilot Control with MLP: The estimated GPS coordinates are then merged with the sensor data, ensuring a comprehensive representation of the UAV's current state, and this combined information is fed into the MLP, which in turn produces the control commands necessary for real-time navigation (elevator, aileron, yaw, and motor throttle). This module effectively replaces the conventional autopilot system when GPS data are unavailable.

Figure 1 illustrates the research workflow that describes the previous steps. This modular design clearly distinguishes between long-term temporal modeling (via the LSTM for GPS trajectory estimation) and non-linear control mapping (via the MLP for aerodynamic surface actuation), enhancing both system interpretability and facilitating independent hyperparameter tuning for each component. Therefore, the hybrid architecture comprises two distinct types of neuronal networks: an MLP and an LSTM neural network.

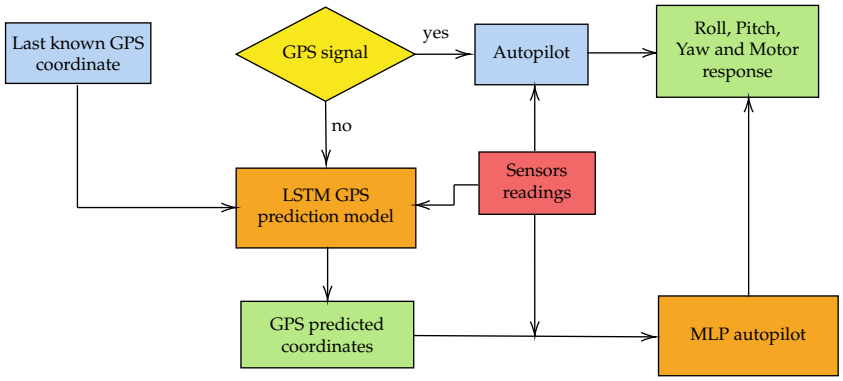

Figure 1. GPS and AI-based navigation decision flowchart.

An MLP is an artificial neural network composed of multiple layers of interconnected nodes (neurons), as depicted schematically in Figure 2. MLPs constitute one of the most fundamental and widely applied models in DL due to their capacity to learn nonlinear relationships in data. The MLP architecture typically includes an input layer that receives the pre-processed signals. In the proposed setup, these signals comprise both the original sensor measurements and the GPS coordinates predicted by the LSTM. Following the input layer, one or more hidden layers capture complex transformations, with the number of nodes and hidden layers depending on the complexity of the control task. These hidden layers extract informative features, enabling the network to learn intricate mappings between the inputs and the control outputs. Finally, the output layer produces the final output corresponding to the control commands on the aerodynamic surfaces. In this work, each output node encodes a particular control surface deflection, such as for ailerons or elevators, effectively replacing a traditional autopilot algorithm.

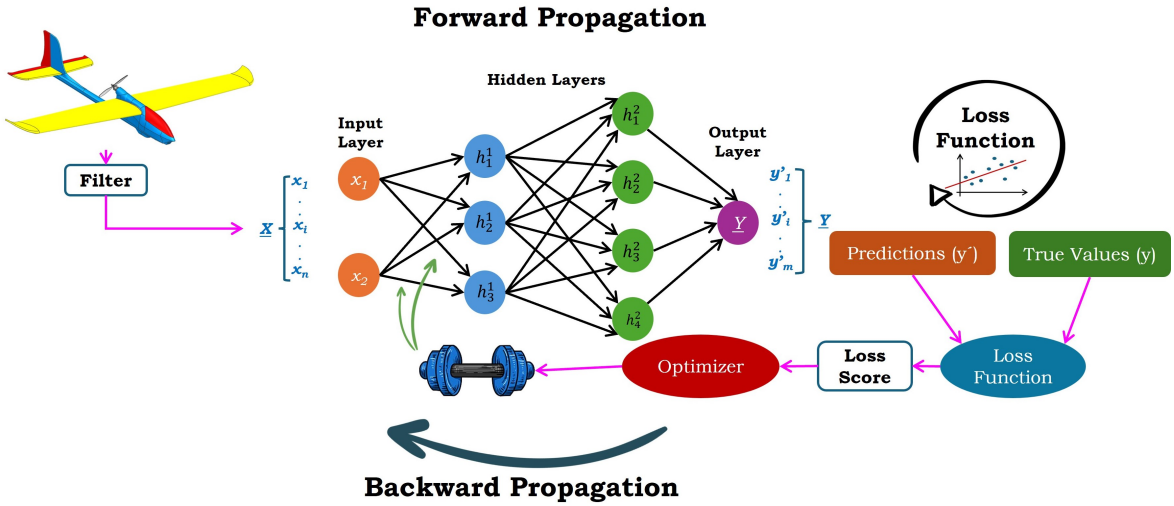

Figure 2. Generic representation of a Multilayer Perceptron (MLP) showing an input layer, one or more hidden layers, and an output layer. In the present work, sensor data and LSTM-based GPS predictions are combined at the MLP input stage to generate control commands.

In contrast, an LSTM network [29] is a type of recurrent neural network designed to learn long-term dependencies in data sequences. In this study, LSTM networks are employed to model the temporal dependencies that arise in GPS coordinate prediction, where current positions are influenced by historical movements. Unlike traditional RNN, LSTMs mitigate the problem of vanishing gradients through dedicated gating mechanisms (forget, input, and output gates) and a persistent cell state [30], allowing relevant data to be retained or discarded over time.

To incorporate nonlinearity, rectified Linear Unit [31] activation is employed in the MLP hidden layers. ReLU is computationally efficient, promotes sparse representations, and is less susceptible to vanishing gradients. Formally, for a neuron input x:

$$\text{ReLU}(x) = \max(0, x). \tag{1}$$

The Mean Absolute Error (MAE) is adopted to measure the discrepancy between model predictions and actual GPS coordinates:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{2}$$

where n is the number of samples, y_i is the ground-truth coordinate, and $\hat{y}_i$ is the predicted value. MAE is intuitive, robust to outliers relative to certain alternatives, and offers a direct interpretation of average prediction error in the coordinate space.

In addition, the Adaptive Moment Estimation (Adam) optimizer [32] is utilized due to its favorable convergence properties and adaptive learning rate. Adam computes exponentially decaying averages of both the first- and second-order moments of the gradient:

- First-order moment: $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$;
- Second-order moment: $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$.

where g_t is the gradient at time step t, and β_1, β_2 are typically set close to 1 (e.g., 0.9 and 0.999). Bias correction factors and a small constant ϵ are applied to improve numerical stability. This approach is well suited for non-stationary objectives, commonly encountered when sequential data and control actions must be learned simultaneously.

Finally, a k-fold cross-validation procedure with $k = 5$ is used to rigorously assess the model's ability to generalize. The complete dataset is partitioned into five folds of similar size, and in each iteration, four folds are employed for training while the remaining fold is used for validation. This process repeats five times, ensuring that each fold serves as the validation set exactly once. The reported performance metrics are then averaged over all five folds, thus reducing the variance associated with any single data split and offering a more robust indicator of the model's expected performance on unseen data.

A detailed analysis of the architecture of the MLP and LSTM networks designed in this work (including the number of layers and neurons), as well as the input features and hyperparameter tuning, is provided in Section 4.

3. Materials and Methods

The UAV platform used for data logging (Figure 3) is a VOLANTEXRC commercial fixed-wing model Ranger. It is a high-performance UAV designed for multiple applications. It has a wingspan of 2400 mm and is constructed from EPO and ABS, providing a robust yet lightweight structure. The UAV is powered by a 4023/1050 KV brushless motor paired with a 75 A ESC and a 6S 2400 mAh battery, providing ample power for extended flight loops.

Equipped with the SpeedyBee F405 WING APP Fixed Wing Flight Controller, this UAV offers advanced control capabilities. The flight controller is equipped with:

- MCU: STM32F405, 168MHz, 1MB Flash;
- Sensors: ICM-42688-P (Gyroscope and Accelerometer), SPL006-001 Barometer;
- OSD Chip: AT7456E;
- Storage: MicroSD slot;
- Communication: 6 UART sets, 1x I2C, 4x ADC, 12x PWM;
- Receivers: Supports ELRS/CRSF, SBUS;
- Battery Input: 7 36V (2 6S LiPo) with multiple BEC outputs.

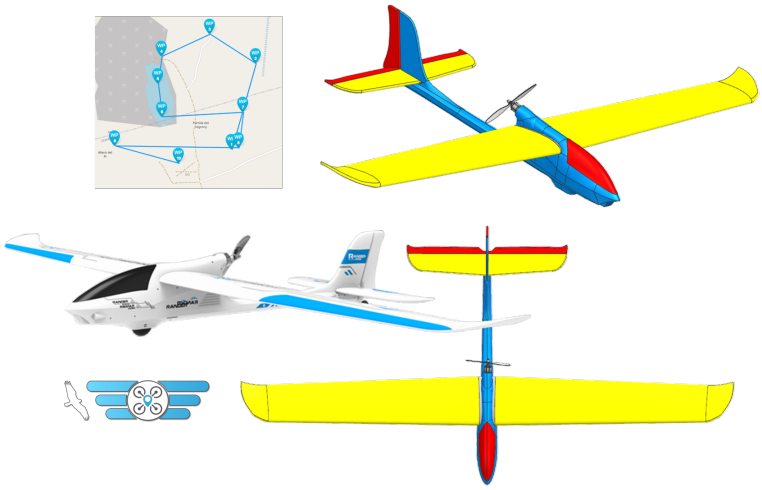

Figure 3. Fixed-wing platform for AI data acquisition.

This UAV possesses a payload capacity of 3 kg and is operated by six 9g servos, which provide precise control and stability during flight. The UAV's altitude is determined using the onboard barometric pressure sensor, which provides continuous altitude estimation based on atmospheric pressure variations. While this particular platform has been selected for demonstration purposes, it should be noted that the proposed methodology is not limited to this UAV model. Instead, it can be applied to different types of aircraft, including more complex systems equipped with additional sensors, which could further enhance the accuracy of predictions.

This study focuses on a case where the UAV is equipped with a minimal sensor suite, which makes it representative of a lower-bound scenario in terms of available flight data. In operational environments requiring higher levels of autonomy, such as defense or intelligence, surveillance, and reconnaissance (ISR) applications, UAVs may incorporate more advanced payloads, including LiDAR, thermal imaging, or multi-spectral cameras, which could further improve AI-driven navigation. Despite employing a relatively elementary UAV configuration, the study's findings underscore the viability of reliable trajectory predictions, thereby substantiating the methodology's adaptability. An illustrative representation of the interconnections within the UAV is provided in Figure 4.

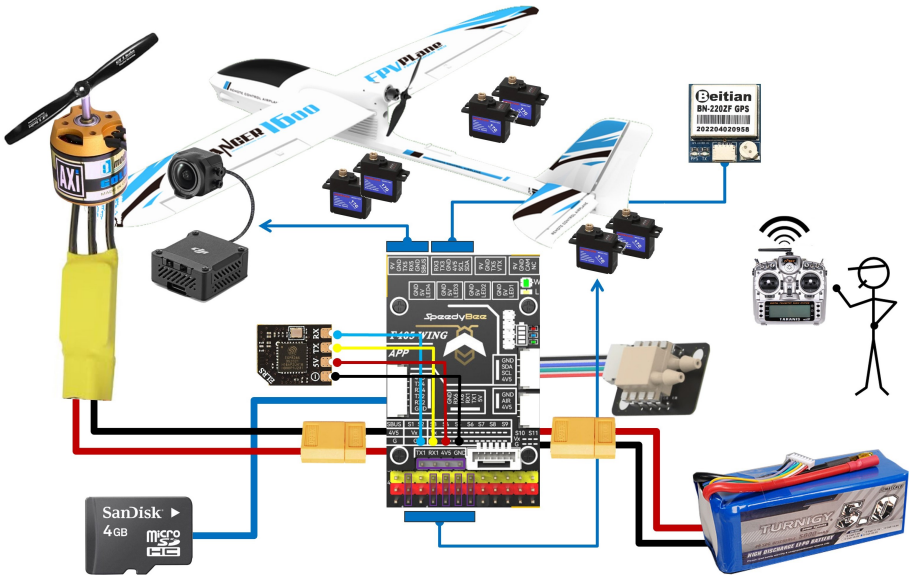

Figure 4. Wire diagram of the UAV electronics.

4. Results and Discussion

4.1. INAV Waypoint Mission System and GPS Dependency

The SpeedyBee flight controller uses an open-source flight control firmware designed for UAVs called INAV, providing waypoint-based autonomous navigation. It enables UAVs to follow pre-programmed flight paths using GPS waypoints. The autopilot operates by continuously computing position updates, ensuring the aircraft follows the designated trajectory while adjusting for wind disturbances and other external factors.

The INAV autopilot system employs PID control loops to navigate waypoints, thereby adjusting the aircraft's orientation and position in accordance with sensor feedback. The system relies on waypoint navigation, where each waypoint is defined by a latitude ϕ, longitude λ, and altitude h. The autopilot calculates the necessary control outputs (throttle, roll, pitch, and yaw) to guide the UAV towards the next waypoint.

The system is dependent on real-time GPS data, which are imperative for ensuring flight stability and trajectory accuracy. In the event of a loss of or unreliability in the GPS signal, the INAV system is unable to ascertain the UAV's precise position, resulting in potential navigation failures or even loss of control. Consequently, the absence of GPS results in the INAV's waypoint navigation being inoperable, thus necessitating the implementation of alternative localization methods to ensure uninterrupted autonomous flight.

As described in Section 2, in order to address the challenge presented by GPS-denied navigation, a DL-based approach is proposed in which two neural network models collaborate to substitute for GPS functionality:

1. LSTM Model for GPS Prediction: The model under consideration is responsible for predicting the UAV's GPS coordinates. This is achieved by means of historical flight telemetry and sensor data. The LSTM architecture is utilized due to its capability of handling sequential data and recognizing temporal dependencies, making it ideal for GPS coordinate estimation.
2. MLP Model for Autopilot Control: This model replaces the conventional autopilot system in cases where GPS data are unavailable, with the predicted GPS coordinates (from the LSTM model) along with current flight telemetry being used as input. The MLP architecture was selected for its ability to approximate complex control functions, mapping sensor inputs to control outputs.

The proposed methodology involves a transition to an AI-based control mechanism that compensates for missing GPS signals. Firstly, the LSTM-based GPS prediction model estimates real-time GPS coordinates using the UAV's sensor inputs and the last real GPS coordinates known, which are fed into the LSTM model. This model has been previously trained with real data enabling the model to identify patterns of UAV flight behavior. Consequently, it can produce estimates of the UAV's position that approximate the true GPS readings. This approach effectively substitutes for the missing GPS data. Concurrently, the MLP-based autopilot control module uses the estimated coordinates to determine corresponding control surface and throttle commands. The MLP has been trained using real flight data, ensuring that its internal control logic matches that of the standard autopilot. Consequently, even in the absence of genuine GPS readings, the UAV continues to receive valid instructions for maintaining stable flight, adjusting roll, pitch, yaw, and throttle as though it still had a reliable position fix. Thanks to this approach, the UAV can effectively follow the established mission, advancing from one waypoint to the next, executing turns and altitude changes based on the LSTM's predictions. This integrated LSTM-MLP architecture ensures the uninterrupted continuation of normal mission tasks, even in the absence of real GPS signals.

The primary innovation of the proposed methodology is that this architecture not only minimizes the requirement for human intervention but also extends the operational window in challenging environments. Rather than initiating a conventional failsafe that could result in the termination of the mission or the need for manual intervention, the methodology enables the UAV to continue operating autonomously. The dual-model approach provides a significant layer of redundancy, enabling critical missions to proceed in GPS-denied environments or under spoofing and jamming attacks. The flowchart seen in Figure 1 summarizes the design explained above.

4.2. Missions

The purpose of this study is to replace the automated piloting algorithm of a UAV with an AI system in the event of an emergency or in GPS-denied scenarios. The model is trained using flight data from predefined and known routes, specifically designed for missions that require continuous flight along a predetermined trajectory, such as surveillance operations. In such instances, UAVs adhere to a predetermined trajectory, though flight conditions and execution may vary between missions. The training data comprise flights along the same waypoints under different conditions, ensuring that the model generalizes within the operational framework. Validation and testing are performed using different flights with the same waypoints but under varying conditions. It is therefore evident that this methodology is only applicable to scenarios where the UAV follows a previously studied and known route and is not designed for completely arbitrary or free-flight navigation. The development of this study was based on the completion of a total of only two distinct missions. Each mission was carried out on three occasions, resulting in a total of six flights. As stated previously, the aircraft controller board is a SpeedyBee F405 WING. The model operates with open-source INAV software, version 7.1.1. The controller board is equipped with a microSD card slot, enabling the use of the blackbox function and the real-time saving of flight information.

The Proportional–Integral–Derivative (PID) loop update rate is 4 kHz, and the blackbox configuration allows for the capture of 6% of the information (240 measurements per second, approximately), having approximately around 432,000 individual measurements to train the Neural Network (NN). This limitation can be attributed to two primary factors. Firstly, the processing capabilities of the flight board's integrated circuit are insufficient to accommodate the entirety of the flight data, including real-time information, and to subsequently store it on the microSD card. Secondly, the read and write speeds of the blackbox on the F405 WING board are constrained. This approach allows us to achieve an optimal balance between the volume of data collected and the processing capacity of the controller board. The two missions were undertaken at the Alzira Model Aircraft Club in Valencia, Spain on 25 April 2024. The two missions are shown in Figure 5.

As observed, the waypoint design includes both sharp and gentle turns. Additionally, we have incorporated scenarios featuring left and right turns, as well as constant altitude changes between waypoints. After completing the final flight, the UAV was programmed to enter home mode, causing the blackbox on the controller board to store data only from the moment it began searching for the first waypoint until it reached the last one. The dataset used for training and validation consists of multiple flights following the same predefined waypoints, accounting for environmental variability. While the UAV trajectory may slightly differ in each flight due to changing conditions, the overall route remains consistent. The proposed methodology does not generalize to arbitrary flight paths but is designed to ensure that UAVs can continue navigating along pre-studied routes in cases of GPS signal loss.

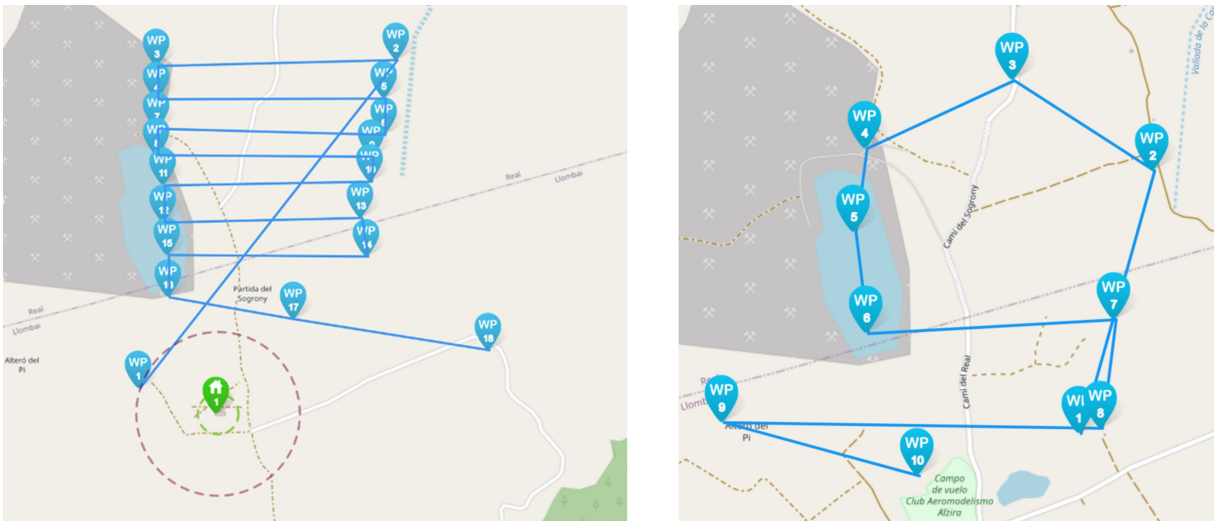

Figure 5. Waypoints of mission 1 (**left**) and mission 2 (**right**).

4.3. Predicting Autopilot Decisions Through Deep Learning

Predicting the positions of a UAV's aerodynamic control surfaces and engine speed is highly advantageous in various operational scenarios, particularly those in which the primary flight control algorithm must be substituted due to an emergency. This prediction is realized through an MLP, as previously introduced in this section. The MLP, which computes the positions of both aerodynamic control surfaces and motors, is especially critical in GPS-denied scenarios. In the event of GPS malfunction, the autopilot—normally reliant on accurate GPS information—becomes inoperative and must be disengaged. Under these conditions, the MLP-based model assumes control of the UAV, effectively replacing the conventional autopilot algorithm. Moreover, this model is continuously supplied with data from an auxiliary GPS coordinate prediction model, in conjunction with the suite of sensors typically employed by the standard autopilot. By integrating these sensor inputs, the MLP-based control system ensures the UAV maintains stable flight and navigational capabilities even in the absence of reliable GPS signals.

The neural network consists of an input layer with 49 features extracted from the UAV's flight telemetry data. These features include sensor readings such as altitude, airspeed, accelerations, control surface positions, and target navigation waypoints, among others. The feature selection process was conducted to ensure that only the most relevant variables were retained, reducing computational complexity while maintaining predictive accuracy.

The architecture of the MLP consists of five hidden layers with 128, 256, 128, 64, and 32 neurons, respectively. This configuration was selected through empirical testing and hyperparameter tuning, ensuring a balance between model complexity and computational efficiency. The increasing neuron count in the second layer (256) allows the network to capture higher-order interactions among the input features, while the gradual reduction in neurons in subsequent layers ensures effective feature extraction and dimensionality reduction without overfitting.

The output layer consists of four neurons, each corresponding to a control signal: elevator, aileron, yaw, and motor throttle. To facilitate convergence, the data from each layer are normalized and passed through a ReLU activation function, allowing the network to learn non-linear relationships between features. Additionally, a dropout rate of 0.1 was applied to prevent overfitting and improve the model's generalization performance. The loss function employed is the MAE, optimized with the Adam optimizer initialized with a learning rate of 0.001. The data were processed in batches of 32 tensors.

With regard to the division of the data, 66% of the dataset (comprising four flights) were utilized for training purposes, thus ensuring that each of the two missions was adequately represented. The remaining two flights were allocated for the purposes of validation and testing, respectively. The model was trained for 100 epochs.

As shown in Figure 6, both the training and validation losses decrease steadily over the training process, indicating the absence of overfitting or underfitting. Additionally, the prediction accuracy for each of the four outputs improves as the model is trained. Accuracy is measured as the percentage of predictions with an error of less than 5% of the true value. The average accuracy across the four outputs reached 99%.

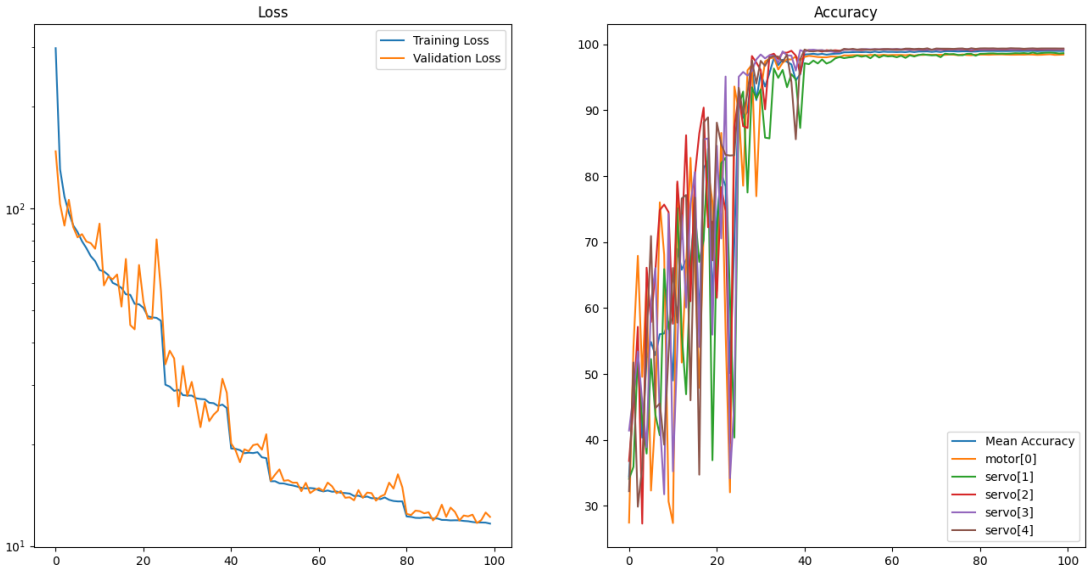

Figure 6. Evolution of the loss and the accuracy over the training loops.

Figure 7 shows a graph of the predicted values versus the actual values for the engine output. This graph exhibits the greatest dispersion among all the plotted results. Once the engine reaches a minimum output level, various values can be considered valid for maintaining altitude, as the only constraint at that point is sustaining a specific minimum speed. However, when an altitude increase is required, it becomes necessary to increase the engine's output. As shown, the dispersion of values becomes more pronounced when the engine provides higher power.

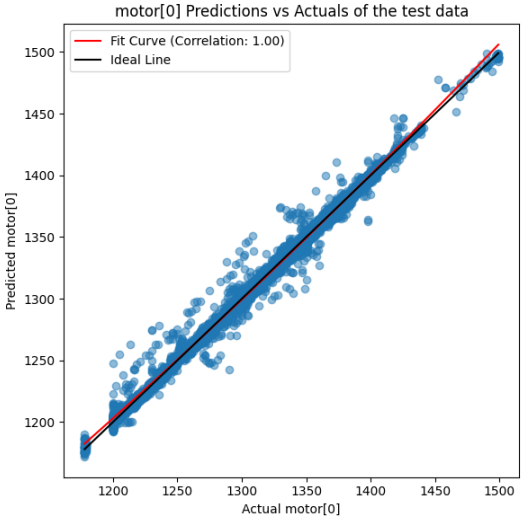

Figure 7. Plot predictions vs. actual values for the motor of the UAV.

Figures 7 and 8 display graphs illustrating the accuracy of the predictions. In these graphs, the actual measurement values are plotted on the x-axis, while the predicted values are plotted on the y-axis. The closer the points are to the diagonal and the less dispersed they are, the more accurate the predictions for each of the five parameters predicted using DL. It is important to note that the signal being analyzed is the one transmitted by the controller board to the electronic speed controllers and servomotors. The range of pulse width modulation (PWM) signals differs between the motor and the servomotors. For the servomotors (elevator, ailerons, and rudder), the signal typically ranges from 1000 µs to 2000 µs, where 1500 µs represents the neutral position. A value below 1500 µs moves the control surface in one direction, while a value above 1500 µs moves it in the opposite direction. For the motor (throttle control), the signal also operates within a range of 1000 µs to 2000 µs. However, in contrast to the servomotors, there is no neutral position at 1500 µs. Instead, 1000 µs corresponds to the motor being off, and increasing the signal above this value progressively increases the throttle output, reaching maximum power at 2000 µs.

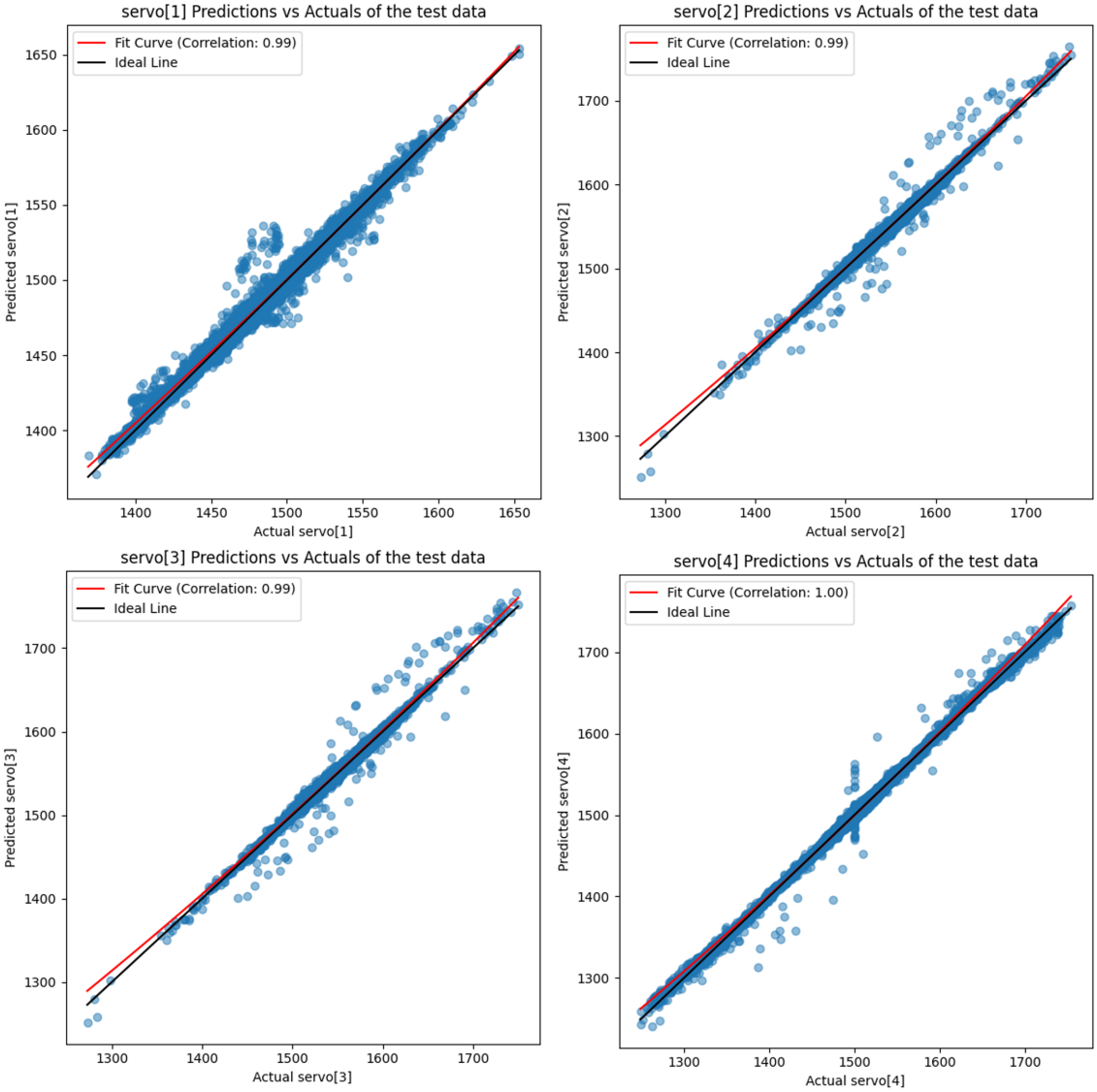

Figure 8. Plot predictions vs. actual values for the 4 servomotors of the UAV.

The UAV's inertial measurements, including angular rates and accelerations, come from the onboard ICM-42688-P gyroscope and accelerometer, which offers a $\pm 2000^{\circ}/\mathrm{s}$ measurement range and low noise density ($3.1\,\mathrm{mdps}/\sqrt{\mathrm{Hz}}$). These characteristics en-

sure high-resolution motion tracking with minimal drift, providing reliable data for DL-based predictions.

The first servomotor corresponds to the yaw movement. The scale of this graph differs from the others because, during flight, the autopilot makes only minor adjustments to yaw, requiring only slight corrections for optimal flight. In contrast, the graphs for servomotors 2 and 3 correspond to the aileron movements. As shown, the point distribution is nearly identical, indicating positive results. The ailerons are the only control surfaces actuated by two motors each. These servomotors are positioned symmetrically along the XZ plane, ensuring that for a given controller signal, their movements are reversed. This behavior is expected, as a turn in one direction requires one control surface to rise while the other is lowered.

The graph for servomotor 4, corresponding to the elevator, shows the widest range of values, which is expected given the nature of the flights and the potential for wind gusts encountered during flight. However, it can be seen that, in most cases, the points remain close to the diagonal, indicating accurate predictions.

Figure 9 illustrates the distribution of the relative importance of each of the characteristics of the model. Given the considerable number of inputs in the model, only those with a significant influence are presented. The following section provides a detailed explanation of the representation of each of these variables. Each of these variables represent the following features of the flight controller:

- navTgPos[0], navTgPos[1]: Target positions along the X and Y axes, representing the desired forward and lateral positions in the navigation system.
- navTgHdg: The target heading or orientation angle, indicating the desired direction the vehicle should face.
- navPos[0], navPos[1], navPos[2]: Current positions along the X, Y, and Z axes, representing the vehicle's forward, lateral, and vertical positions in the navigation frame.
- BaroAlt (cm): The altitude measured by the barometer, used for estimating the vehicle's height above ground.
- rcData[0], rcData[1], rcData[2]: Input signals from the remote control (RC) transmitter for the first three channels, typically used for roll, pitch, and throttle control.
- rcCommand[3]: The processed control command for the fourth channel, often linked to the vehicle's throttle or auxiliary functions.
- accSmooth[2]: The smoothed acceleration data along the Z-axis, used for stability control and navigation adjustments.

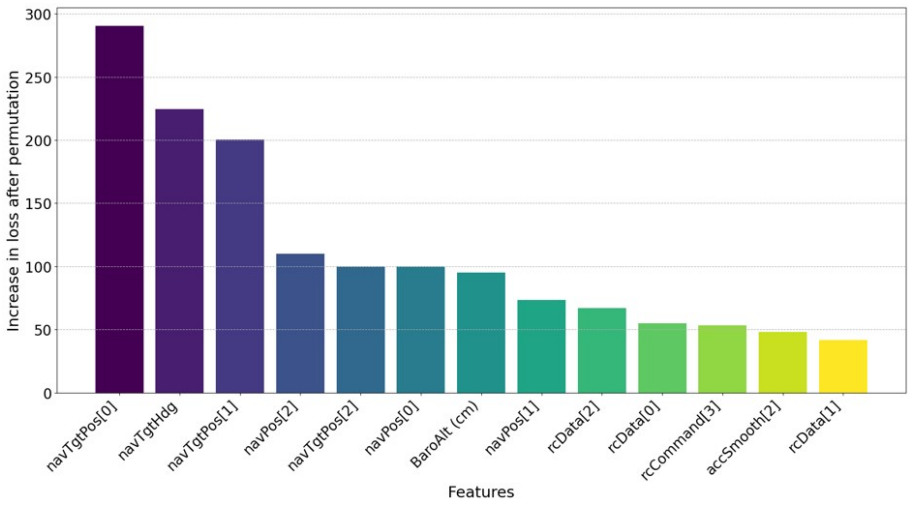

Figure 9. Plot predictions vs. actual values for the motor of the UAV.

Tables 2 and 3 present the principal metrics and characteristics of the autopilot model. In consideration of Table 2, the following observations can be made:

- Mean Squared Error (MSE): The MSE values, ranging from 16.61 to 21.36, indicate that the mean squared error in the predictions is moderate. This level of error is relatively low compared to the variable range (1000–2000), suggesting that the discrepancies between predicted and actual values are minimal.
- Root Mean Square Error (RMSE) and Mean Absolute Error (MAE): The RMSE and MAE values, ranging from 4.08 to 4.62 and from 2.16 to 2.91, respectively, support the conclusion that the absolute errors and the standard deviation of the errors are small relative to the variable ranges. This indicates accurate predictions.
- Coefficient of Determination (R^2): This statistical measure indicates the extent to which a dependent variable can be explained by one or more independent variables. The model demonstrates a high degree of explanatory power, with a constant R^2 of 0.998107, indicating an excellent fit.
- Mean Absolute Percentage Error (MAPE): The MAPE is consistently low at 0.164278, indicating a minimal average percentage error relative to the true values. This low percentage error is particularly significant given the wide range of values observed in the variables.

Table 2. Main metrics of the autopilot DL model. Servos [1–4] correspond to the primary control surfaces: Servo[1]—aileron (Left), Servo[2]—aileron (Right), Servo[3]—elevator, Servo[4]—rudder.

Metric	Motor	Servo[1]	Servo[2]	Servo[3]	Servo[4]
MSE	16.609	21.360	16.978	19.107	17.486
RMSE	4.075	4.622	4.120	4.371	4.182
MAE	2.158	2.912	2.197	2.420	2.458
R^2	0.998	0.998	0.998	0.998	0.998
MAPE	0.164	0.164	0.164	0.164	0.164

In the analysis of model performance metrics, it is clear that the model demonstrates exceptional results in both classification accuracy and regression error metrics. This is particularly impressive given the variable range between 1000 µs and 2000 µs. The features highlighted in Table 2 can be summarized as follows:

- The model achieves an outstanding mean accuracy of 99.00%, indicating that the majority of its predictions are highly accurate.
- The accuracy of the motor and servos is equally noteworthy, with individual accuracies ranging from 98% to 99.38. This demonstrates the model's effectiveness in correctly classifying the states and commands of these components.
- The reported classification errors are exceptionally low, all below 0.2%. This indicates that the model rarely makes incorrect predictions, which is critical for flight control applications where precision is essential.

Table 3. Summary of the statistics of the autopilot model.

Metric	Value	Metric	Value
Train Losses	11.64	Servo 4 accuracy [%]	99.38
Validation Losses	12.19	Motor error [%]	0.16
Mean accuracy [%]	99.00	Servo 1 error [%]	0.20
Motor accuracy [%]	98.46	Servo 2 error [%]	0.14
Servo 1 accuracy [%]	98.66	Servo 3 error [%]	0.16
Servo 2 accuracy [%]	99.24	Servo 4 error [%]	0.16
Servo 3 accuracy [%]	99.27	Time [min]	10.75

As shown in Figure 10, the mean validation accuracy curves indicate that the model reaches an accuracy level approaching 100% after approximately 40 epochs across all folds. This behavior suggests that the model has effectively learned the patterns present in the training data and can generalize these patterns to the validation data. The stabilization of the curves at high accuracy levels is a positive indicator of the model's ability to make precise predictions. The consistent reduction in both the training and validation loss curves reflects continuous model improvement over the epochs. The simultaneous decrease in both curves indicates a good model fit, with no clear signs of overfitting. Fluctuations observed in the validation loss curves are expected due to the inherent variability of the validation data and do not significantly affect the overall downward trend. This behavior confirms that the model is learning effectively and minimizing prediction errors.

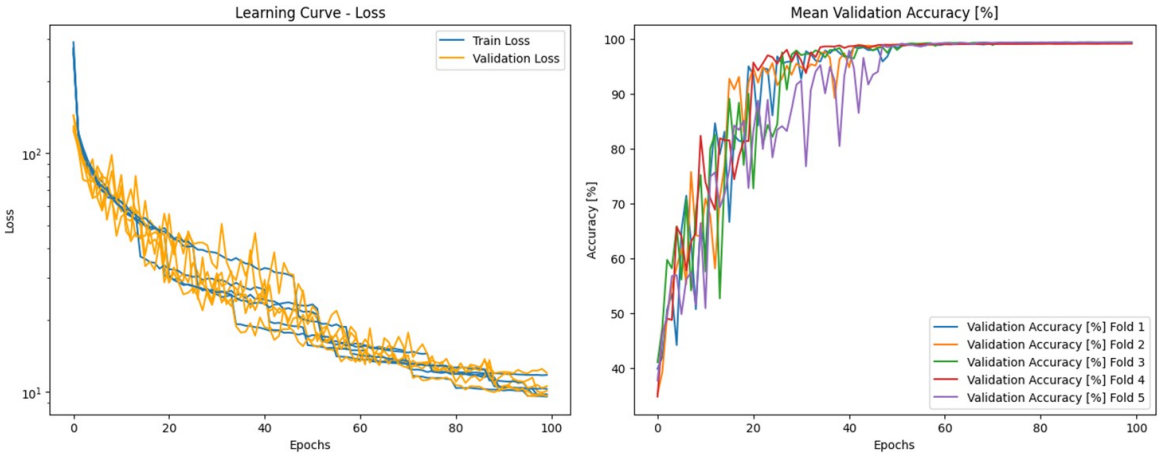

Figure 10. Evolution of the loss and the accuracy for the 5-fold cross-validation trainings of the autopilot model.

Regarding Table 4, the calculation of average performance metrics, including MSE, RMSE, MAE, R^2, and MAPE, provides an accurate assessment of the model's effectiveness. The low values of MSE, RMSE, and MAE indicate minimal absolute prediction errors. Notably, the R^2 value of approximately 0.998544 reflects an excellent proportion of the variance explained by the model. Additionally, the low MAPE value demonstrates minimal percentage errors, which is particularly important given the wide range of input variable variations (1000–2000 µs). These metrics confirm that the model is both accurate and reliable in its predictions. Collectively, this evidence suggests that the model is not only capable of learning patterns from the training data but can also generalize effectively to new, previously unseen data.

Table 4. Average metrics over the 5-fold cross-validation for the autopilot model.

Metric	Motor[0]	Servo[1]	Servo[2]	Servo[3]	Servo[4]
MSE	16.119	14.479	12.521	12.712	14.029
RMSE	4.006	3.772	3.524	3.553	3.726
MAE	1.999	2.453	1.973	1.996	1.986
R^2	0.999	0.999	0.999	0.999	0.999
MAPE	0.141	0.141	0.141	0.141	0.141

4.4. Predicting GPS Coordinates Through LSTM Neural Networks

This section has been developed using Long Short-Term Memory (LSTM) networks. As explained in the theoretical section, this type of network can store temporal information and use it to generate predictions. This capability is crucial in this context, as the objective is to predict GPS coordinates. Consequently, the prediction of the next coordinate is

significantly influenced by both the UAV's dynamic readings—such as velocities, altitudes, and accelerometer data—and the most recent coordinate information, whether measured by GPS or predicted by the model.

To predict the coordinates, a neural network capable of generating two outputs, corresponding to latitude and longitude, was designed. However, since the flights were conducted at the same location, the differences between various GPS coordinate measurements are minimal, amounting to only a few thousandths of a decimal degree. This makes generating precise predictions particularly challenging for the model. Therefore, the first step involved normalizing the coordinate data used in the training process, applying the following formula:

$$Z = \frac{X - \mu}{\sigma} \tag{3}$$

where X represents the actual measurement, μ is the mean of the training data and σ is the corresponding standard deviation. It is crucial to retain these values in order to facilitate the reversal of the normalization process, as the predictions of the trained model will undergo normalization.

In this context, a neural network with an LSTM layer containing 128 neurons has been employed. The output from the LSTM layer is passed to a four-layer MLP network with 64, 32, 16, and 2 neurons, with the last two forming the output layer. Each layer of the MLP network uses a ReLU activation function.

After evaluating several configurations, it was determined that an initial learning rate of 1×10^{-5}, combined with the Adam optimizer, yields the best results. A learning rate scheduler was applied, reducing the learning rate by half every five epochs if no significant reduction in the validation loss was observed. The loss function and the division of data into training, validation, and test sets remain consistent with the previous configuration, while the batch size was set to 128. The model was trained for 200 epochs.

Additionally, it is essential to consider whether the data should be analyzed bidirectionally in this type of neural network. This means determining whether the model should incorporate future context when predicting the current value. In this case, since the UAV is in flight and future readings are unknown, using bidirectional analysis is not beneficial. However, in applications such as text compression, where part of the relevant information needed to interpret the current input appears later, this feature can be highly advantageous.

Another crucial factor is the length of the sequence provided to the neural network, which influences how the model captures temporal evolution. In this case study, sequences of five measurements were used. It is important to balance between sequences that are too short, which may fail to capture meaningful temporal dependencies, and sequences that are too long, which can cause memory constraints.

In Figure 11 (left), the results obtained from the GPS coordinate prediction model are presented. Figure 11 (right) shows that the training and validation loss values are notably low, approximately 10^{-4}. This outcome is expected, as the loss function quantifies the discrepancy between predicted and actual values. Given that the differences in GPS coordinates are only a few thousandths of a decimal degree, the low loss values are not surprising.

Additionally, the similar values observed for validation and training losses indicate that overfitting has not occurred. In terms of prediction accuracy, approximately 90% of the GPS coordinate predictions exhibit a discrepancy of less than 5% compared to their actual values, demonstrating the model's reliability and precision.

Figure 12 shows the actual versus predicted values for both components of the GPS coordinates. It can be observed that all predictions are located close to the diagonal, indicating strong alignment between predicted and actual values. As presented in Table 5,

the model demonstrates a high overall accuracy, with an average accuracy of 92.15%. The accuracy of latitude predictions is slightly higher (93.47%) than that of longitude predictions (90.84%), suggesting a slight tendency for the model to predict latitudinal coordinates more accurately than longitudinal ones.

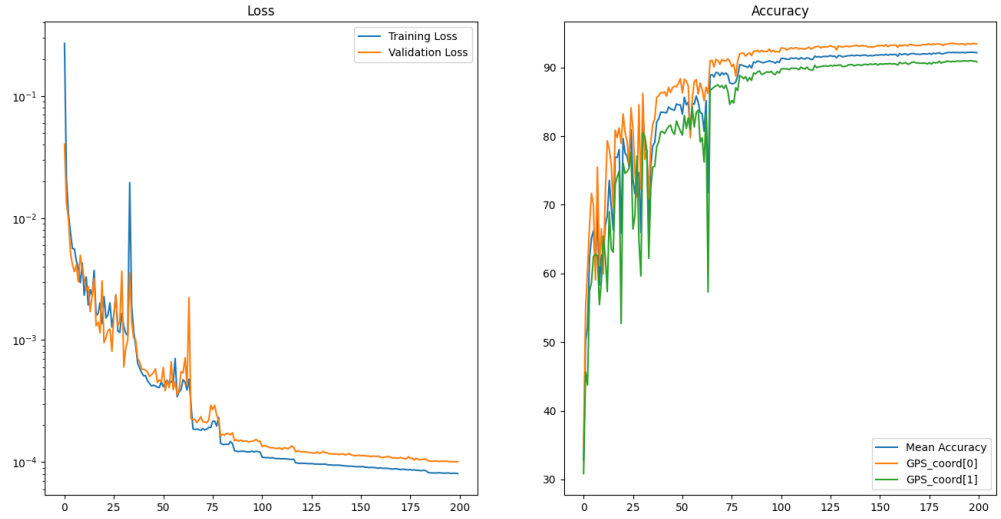

Figure 11. Evolution of the loss and the accuracy for GPS predictor model.

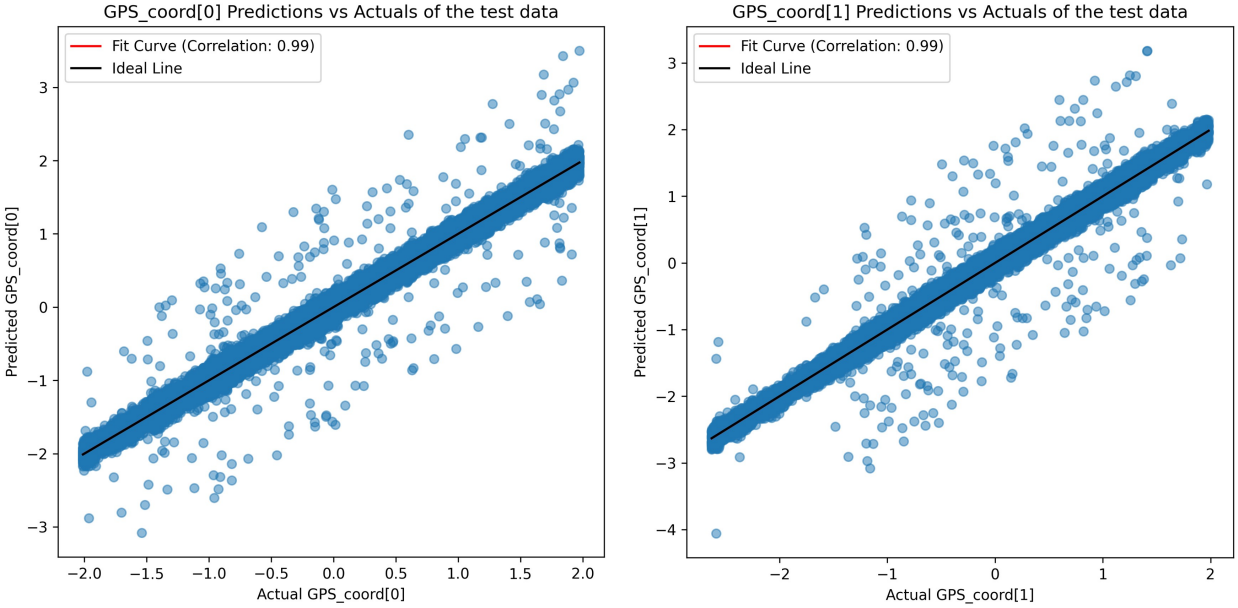

Figure 12. Plot predictions vs. actual values for the 2 coordinates of the GPS of the UAV.

Table 5. Summary of the statistics of the GPS predictor model.

Metric	Value
Train Losses	0.00008
Validation Losses	0.0001
Mean Accuracy [%]	92.15
Latitude Accuracy [%]	93.47
Longitude Accuracy [%]	90.84
Mean Relative Error [%]	0.87
Time [min]	27.87

The training loss is notably low (0.00008), indicating that the model has been effectively calibrated to the training data. However, the validation loss is marginally higher (0.0001), reflecting a slight discrepancy between the training and validation data. This difference is expected and generally acceptable for models with a high degree of generalization. The average relative error is approximately 0.87%, which is relatively low and demonstrates that the model is making accurate predictions compared to the true values. The total training time was 27.87 minutes, which is reasonable given the level of accuracy achieved. It is important to highlight that in Figure 12, the axes do not represent the actual real coordinates of the UAV. This is because, during flight, the variations in GPS coordinates are minimal due to the global reference system used by GPS, which applies anywhere on Earth. Consequently, even though the UAV may cover relatively long distances locally, the changes in its GPS coordinates are very subtle.

To address this, the output data have been normalized to facilitate the training process of the neural network (NN). After training, when the model produces predictions, these normalized outputs are denormalized using the stored normalization parameters, allowing the retrieval of the actual GPS coordinates.

It should also be noted that this normalization affects the analysis of the precision metrics discussed below, as these metrics are calculated on the normalized predictions rather than the denormalized outputs.

An examination of the normalization metrics in Table 5 reveals that the MSE for GPS[0] (latitude) coordinates is 0.000030, while for GPS[1] (longitude), it is 0.000072. The RMSE values for latitude and longitude are 0.005509 and 0.008461, respectively, indicating low prediction errors in absolute terms. The MAE values are 0.003706 for latitude and 0.006192 for longitude, reflecting a comparable level of accuracy across metrics. The R^2 value is nearly perfect (0.999949) for both coordinates, confirming that the model explains a substantial proportion of the observed variability in the data. The MAPE is 3.32% for both coordinates, indicating that, on average, the model's predictions are within 3.32% of the true values, highlighting the model's high accuracy and robustness.

Figure 13 illustrates two critical aspects of the model's training process and its evaluation through cross-validation. On the left, the loss learning curves for both the training and validation sets are displayed. Notably, both curves exhibit a gradual decline over the epochs, indicating effective learning by the model. The training and validation losses eventually stabilize, demonstrating clear convergence. It is important to note that the curves are presented on a logarithmic scale, which amplifies differences at lower values. As a result, the apparent gap between training and validation losses is more pronounced visually, though the absolute differences are minimal, particularly in the later stages of training.

On the right, Figure 13 shows the average validation accuracy across five cross-validation folds. The graph illustrates consistent accuracy improvements over the epochs, reaching a stable level of approximately 90%. Each fold follows a similar trend, indicating that the model exhibits robust generalization across diverse subsets of the data. The initial variability in accuracy reflects the model's fitting process, which stabilizes as training progresses.

Regarding Table 6, the cross-validation results demonstrate the model's remarkable performance. The MSE for GPS[0] coordinates (latitude) is 0.000116, while for GPS[1] (longitude), it is 0.000272. The RMSE values are 0.010751 for latitude and 0.016479 for longitude, indicating small absolute prediction errors. These values align with the visual observations from the figures, confirming the model's low overall error.

The MAE values are 0.006131 for latitude and 0.010795 for longitude, further reflecting the high accuracy of the model's predictions. The R^2 value of 0.999806 for both coordinates suggests that the model explains nearly all of the observed variability in the data.

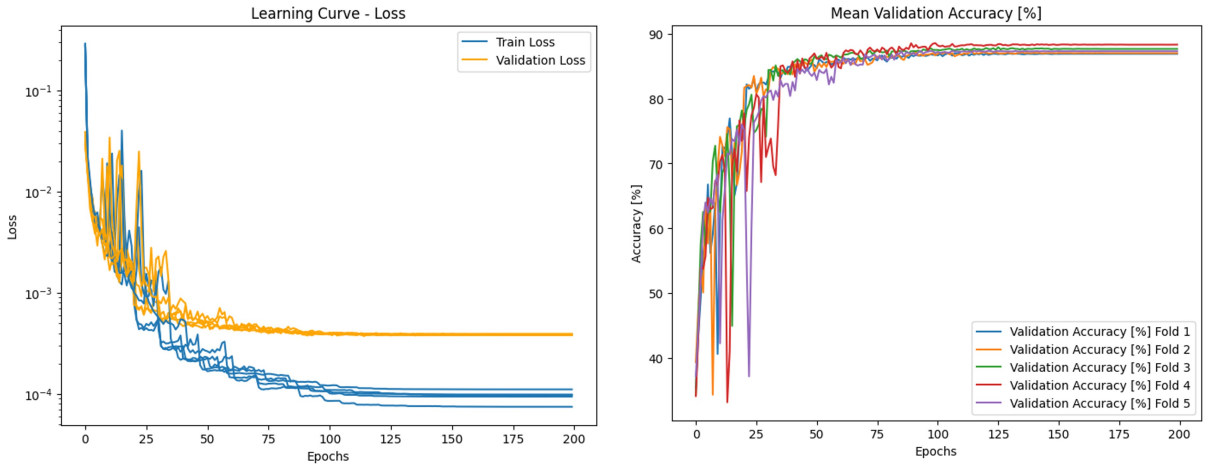

Figure 13. Evolution of the loss and the accuracy for the 5-fold cross-validation training of the GPS model.

Although the MAPE is approximately 12.29% for both coordinates, this reflects the percentage-based nature of the error. While higher than other metrics, it remains acceptable given the specific characteristics of the dataset, particularly the scale of the GPS coordinate values. This result underscores the model's strong predictive capability despite the inherent challenges posed by the data's numerical scale.

Table 6. Average metrics over the 5-fold cross-validation for the GPS model.

Metric	GPS_coord[0]	GPS_coord[1]
MSE	0.000116	0.000272
RMSE	0.010751	0.016479
MAE	0.006131	0.010795
R^2	0.999806	0.999806
MAPE	12.290785	12.290785

4.5. GPS Prediction Model Deployment

The following section outlines the methodology for testing the GPS coordinate prediction model. To accomplish this, one of the flights saved in the test dataset will be used. The data from this flight have never been presented to the model during the training phase; however, a flight with identical waypoints was previously included in the training set.

The test begins by collecting flight data under normal conditions with a fully functioning GPS. Next, a scenario will be simulated in which the GPS antenna malfunctions or the GPS satellite signal is lost, leaving the UAV unable to determine its position using GPS. In this situation, the final five sensor readings will be collected and combined with existing UAV sensor data to estimate the UAV's current position.

Since the model outputs normalized values, it is necessary to transform the predicted data back to actual coordinates using the mean and standard deviation of the training data.

As shown in Figure 14 (left), the predicted trajectory for a waypoint-based autonomous flight without GPS appears to be reasonably accurate. However, Figure 14 (right) provides a more detailed view, revealing a considerable amount of noise in the predictions. This noise could hinder the UAV's ability to navigate accurately to the next waypoint.

To improve this behavior, a smoothing technique based on an exponential moving average (EMA) is proposed. This technique uses only previous predicted values to adjust the current prediction, resulting in less noisy outputs. Consequently, this enables a more stable flight for the UAV. The formula for calculating the exponential moving average is as follows:

$$\mathrm{EMA}_t = \alpha \cdot X_t + (1 - \alpha) \cdot \mathrm{EMA}_{t-1} \tag{4}$$

In this context, EMA_t represents the exponential moving average at time t, X_t denotes the current value at time t, and α is the smoothing factor, calculated as $\alpha = \frac{2}{N+1}$, where N is the number of periods within the specified moving window. Additionally, EMA_{t-1} represents the exponential moving average from the preceding period.

Figure 14. Predicted (red line) vs. real coordinates (blue line) obtained through the GPS predictor model (**left**) and detail of the predictions (**right**).

The implementation of this smoothing procedure has resulted in significantly more linear trajectories, as shown in Figure 15. The model's training for the specified autonomous flights has been considered successful, allowing it to predict coordinates in emergency scenarios. As observed in the figure, the trajectory predicted by the model does not exactly match the actual path but is sufficiently accurate to enable automatic waypoint navigation without GPS in emergency situations.

Figure 15. Smoothened flight path prediction vs. reality.

5. Conclusions

The ability to maintain accurate control of a UAV's flight dynamics without GPS inputs is essential for ensuring safe and reliable operations under challenging conditions. This study explores the use of two types of neural networks, an MLP and an LSTM, demonstrating significant potential for enhancing autonomous UAV navigation in GPS-denied environments.

On the one hand, an MLP model was proposed and experimentally tested to emulate autopilot. This model was tasked with predicting the positions of the UAV's aerodynamic control surfaces and engine speed, achieving an average accuracy of 99.00%. The model's performance is supported by its low MSE and RMSE values, indicating minimal discrepancies between predicted and actual values. Such precision suggests that the MLP model could serve as an effective substitute for traditional autopilot systems, particularly in emergency scenarios where standard navigation aids are unavailable.

On the other hand, the LSTM model was employed to forecast the UAV's GPS coordinates, achieving an average accuracy of 92.15%. Notably, the model demonstrated proficiency in handling temporal sequences and predicting future coordinates based on historical data. Its mean absolute error (MAE) of approximately 0.87% highlights its ability to provide accurate location data, critical for maintaining the UAV's trajectory in GPS-denied environments. Furthermore, the stability of the predictions was enhanced by implementing a smoothing technique based on an exponential moving average. This approach generated more stable flight paths, improving the UAV's ability to navigate autonomously without external navigation aids.

These advancements open promising applications in sectors such as disaster relief, precision agriculture, and urban planning, where UAVs can operate effectively even in areas with limited GPS coverage. The findings of this study highlight the significant potential for advancing autonomous UAV systems capable of operating reliably and effectively in diverse and challenging environments. The study suggests that future research could focus on integrating additional sensor data, such as visual inputs from cameras, to further enhance the models' accuracy and robustness. Exploring applications in different UAV types and flight conditions could also broaden the models' versatility. Further development and refinement of these technologies could enable UAVs to become more autonomous and capable of performing complex missions with minimal human intervention.

Author Contributions: Conceptualization, C.G.-G. and P.C.-P.; methodology, C.G.-G. and P.C.-P.; software, C.G.-G.; validation, P.C.-P.; formal analysis, P.C.-P.; investigation, P.C.-P.; resources, F.C. and J.A.G.-M.; writing—original draft preparation, C.G.-G., P.C.-P. and J.A.G.-M.; writing—review and editing, C.G.-G., P.C.-P., F.C. and J.A.G.-M.; funding acquisition, J.A.G.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received partial funding from the Government of Spain under the project PID2023-151110OB-I00 and Generalitat Valenciana under CIPROM/2022/3 and CIACIF/2021/286. This research project is part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Funding for open access charge: Universitat Politècnica de València

Data Availability Statement: The data from this work is available upon request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yang, L.; Wang, X.; Zhou, Y.; Liu, Z.; Shen, L. Online Predictive Visual Servo Control for Constrained Target Tracking of Fixed-Wing Unmanned Aerial Vehicles. *Drones* **2024**, *8*, 136. [CrossRef]
2. Katara, P.; Harish, Y.V.S.; Pandya, H.; Gupta, A.; Sanchawala, A.M.; Kumar, G.; Bhowmick, B.; Krishna, M.K. DeepMPCVS: Deep Model Predictive Control for Visual Servoing. In Proceedings of the 2020 Conference on Robot Learning, PMLR, Cambridge, MA, USA, 16–18 November 2020; Kober, J., Ramos, F., Tomlin, C., Eds.; Proceedings of Machine Learning Research; MLR Press: Philadelphia, PA, USA, 2020; Volume 155; pp. 2006–2015. Available online: https://proceedings.mlr.press/v155/katara21a.html (accessed on 7 June 2024).

3. Ulus, Ş.; Eski, İ. Neural network and fuzzy logic-based hybrid attitude controller designs of a fixed-wing UAV. *Neural Comput. Appl.* **2021**, *33*, 8821–8843. [CrossRef]
4. Al-lQubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Razaque, A.; Hariri, S. Deep learning for unmanned aerial vehicles detection: A review. *Comput. Sci. Rev.* **2024**, *51*, 100614. [CrossRef]
5. Dhajaram, R.; Saderla, S.; Ghosh, A.K. Parameter estimation of UAV from flight data using neural network. *Aircr. Eng. Aerosp. Technol.* **2018**, *90*, 302–311. [CrossRef]
6. Bøhn, E.; Coates, E.M.; Reinhardt, D.; Johansen, T.A. Data-Efficient Deep Reinforcement Learning for Attitude Control of Fixed-Wing UAVs: Field Experiments. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 3168–3180. [CrossRef]
7. Tournier, V.; Topham, C.M.; Gilles, A.; David, B.; Folgoas, C.; Moya-Leclair, E.; Kamionka, E.; Desrousseaux, M.L.; Texier, H.; Gavalda, S.; et al. An engineered PET depolymerase to break down and recycle plastic bottles. *Nature* **2020**, *580*, 216–219. [CrossRef]
8. Peng, W.; Lei, J.; Ding, C.; Yue, C.; Ma, G.; Sun, J.; Zhang, D. A novel deep ensemble reinforcement learning based control method for strip flatness in cold rolling steel industry. *Eng. Appl. Artif. Intell.* **2024**, *134*, 108695. [CrossRef]
9. Avramelou, L.; Nousi, P.; Passalis, N.; Tefas, A. Deep reinforcement learning for financial trading using multi-modal features. *Expert Syst. Appl.* **2024**, *238*, 121849. [CrossRef]
10. Chkirbene, Z.; Ridha, H.; Ünal, D.; Gabbouj, M.; Hamdi, M. Enhancing Healthcare Systems With Deep Reinforcement Learning: Insights Into D2D Communications and Remote Monitoring. *IEEE Open J. Commun. Soc.* **2024**, *5*, 3824–3838. [CrossRef]
11. Chowdhury, M.; Keshmiri, S. A Unified Inner-Outer Loop Reinforcement Learning Flight Controller for Fixed-Wing Aircraft. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Chania-Crete, Greece, 4–7 June 2024; pp. 556–563. [CrossRef]
12. Osco, L.P.; Marcato Junior, J.; Ramos, A.P.M.; de Castro Jorge, L.A.; Fatholahi, S.N.; de Andrade Silva, J.; Matsubara, E.T.; Pistori, H.; Gonçalves, W.N.; Li, J. A review on deep learning in UAV remote sensing. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *102*, 102456. [CrossRef]
13. Li, L.; Zhang, X.; Qian, C.; Wang, R.; Zhao, M. Autopilot Controller of Fixed-Wing Planes Based on Curriculum Reinforcement Learning Scheduled by Adaptive Learning Curve. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *8*, 2182–2196. [CrossRef]
14. Derbali, M.; Jerbi, H.; Khan, F.; Jan, S.; Piromalis, D.; Tsaramirsis, G. A Technical Framework for Selection of Autonomous UAV Navigation Technologies and Sensors. *Comput. Mater. Contin.* **2021**, *68*, 2772–2790. [CrossRef]
15. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
16. Lu, Y.; Xue, Z.; Xia, G.-S.; Zhang, L. A survey on vision-based UAV navigation. *Geo-Spat. Inf. Sci.* **2018**, *21*, 21–32. [CrossRef]
17. Wang, C.; Wang, J.; Wang, J.; Zhang, X. Deep-reinforcement-learning-based autonomous UAV navigation with sparse rewards. *IEEE Internet Things J.* **2020**, *7*, 6180–6190. [CrossRef]
18. Rezwan, S.; Choi, W. Artificial Intelligence Approaches for UAV Navigation: Recent Advances and Future Challenges. *IEEE Access* **2022**, *10*, 26320–26339. [CrossRef]
19. Al-Jarrah, O.Y.; Shatnawi, A.S.; Shurman, M.M.; Ramadan, O.A.; Muhaidat, S. Exploring Deep Learning-Based Visual Localization Techniques for UAVs in GPS-Denied Environments. *IEEE Access* **2024**, *12*, 113049–113065. [CrossRef]
20. Couturier, A.; Akhloufi, M.A. A Review on Deep Learning for UAV Absolute Visual Localization. *Drones* **2024**, *8*, 622. [CrossRef]
21. Long, K. Deep learning and visual SLAM for autonomous navigation of UAVs: Status, challenges, and future perspectives. *Highlights Sci. Eng. Technol.* **2024**, *119*, 746–751. [CrossRef]
22. Challita, U.; Saad, W.; Bettstetter, C. Interference management for cellular-connected UAVs: A deep reinforcement learning approach. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2125–2140. [CrossRef]
23. Sarkar, N.I.; Gul, S. Artificial Intelligence-Based Autonomous UAV Networks: A Survey. *Drones* **2023**, *7*, 322. [CrossRef]
24. Munguia, R.; Grau, A.; Bolea, Y.; Obregón-Pulido, G. A Simultaneous Control, Localization, and Mapping System for UAVs in GPS-Denied Environments. *Drones* **2025**, *9*, 69. [CrossRef]
25. Bithas, P.S.; Michailidis, E.T.; Nomikos, N.; Vouyioukas, D.; Kanatas, A.G. A Survey on Machine-Learning Techniques for UAV-Based Communications. *Sensors* **2019**, *19*, 5170. [CrossRef]
26. El Alaoui, M.; Amraoui, K.E.; Masmoudi, L.; Ettouhami, A.; Rouchdi, M. Unleashing the potential of IoT, Artificial Intelligence, and UAVs in contemporary agriculture: A comprehensive review. *J. Terramech.* **2024**, *115*, 100986. [CrossRef]
27. Al Radi, M.; AlMallahi, M.N.; Al-Sumaiti, A.S.; Semeraro, C.; Abdelkareem, M.A.; Olabi, A.G. Progress in artificial intelligence-based visual servoing of autonomous unmanned aerial vehicles (UAVs). *Int. J. Thermofluids* **2024**, *21*, 100590.
28. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
29. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
30. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]

31. Fukushima, K. Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements. *IEEE Trans. Syst. Sci. Cybern.* **1969**, *5*, 322–333. [CrossRef]
32. Kingma, D.P.; Lei Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980. [CrossRef]