Article

Drone Trajectory Segmentation for Real-Time and Adaptive Time-Of-Flight Prediction

Claudia Conte ^{1,2} , Giorgio de Alteriis ^{1,2} , Rosario Schiano Lo Moriello ¹ , Domenico Accardo ^{1,*}
and Giancarlo Rufino ¹

¹ Department of Industrial Engineering, University of Naples Federico II, Piazzale Tecchio 80, 80125 Naples, Italy; claudia.conte2@unina.it (C.C.); giorgio.dealteriis@unina.it (G.d.A.); rschiano@unina.it (R.S.L.M.); giancarlo.rufino@unina.it (G.R.)

² Department of Management, Information and Production Engineering, University of Bergamo, Viale Marconi 5, 24044 Dalmine, Italy

* Correspondence: domenico.accardo@unina.it

Abstract: This paper presents a method developed to predict the flight-time employed by a drone to complete a planned path adopting a machine-learning-based approach. A generic path is cut in properly designed corner-shaped standard sub-paths and the flight-time needed to travel along a standard sub-path is predicted employing a properly trained neural network. The final flight-time over the complete path is computed summing the partial results related to the standard sub-paths. Real drone flight-tests were performed in order to realize an adequate database needed to train the adopted neural network as a classifier, employing the Bayesian regularization backpropagation algorithm as training function. For the network, the relative angle between two sides of a corner and the wind condition are the inputs, while the flight-time over the corner is the output parameter. Then, generic paths were designed and performed to test the method. The total flight-time as resulting from the drone telemetry was compared with the flight-time predicted by the developed method based on machine learning techniques. At the end of the paper, the proposed method was demonstrated as effective in predicting possible collisions among drones flying intersecting paths, as a possible application to support the development of unmanned traffic management procedures.

Keywords: drone; machine learning; neural network; trajectory prediction; unmanned traffic management

Citation: Conte, C.; de Alteriis, G.; Schiano Lo Moriello, R.; Accardo, D.; Rufino, G. Drone Trajectory Segmentation for Real-Time and Adaptive Time-Of-Flight Prediction. *Drones* **2021**, *5*, 62. <https://doi.org/10.3390/drones5030062>

Academic Editors: Marija Popović and Inkyu Sa

Received: 2 June 2021

Accepted: 12 July 2021

Published: 16 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The integration of drones in civil airspace involves several challenges related to the assessment of the total risk level. The prediction of the flight-time employed by the drone to perform a generic path is crucial for the strategical and tactical phases for the traffic analysis needed by unmanned traffic management (UTM), in particular for conflict prediction in order to prevent possible collisions among the platforms.

In the field of air traffic management (ATM), several efforts have been made by NextGen and the Single European Sky ATM Research Program (SESAR) to develop the key elements of trajectory management [1]. In particular, the adoption of trajectory based operations (TBO) requires integrated flights data systems to properly manage the flight plans before departure and modify the flight routes during trajectory execution [2]. Trajectory prediction (TP) techniques can be adopted to support trajectory management and the conflict detection and resolution tasks. Several studies have proposed TP methods based on aircraft physics parameters, such as model-based, probabilistic and Bayesian approaches [3–6]. Data-driven approaches can be adopted when historical data series are available, such as [7–10]. Moreover, considering the dynamic optimization of trajectories, several works were focused on the adoption of Karush–Kuhn–Tucker (KKT) conditions to achieve an optimal solution [11–15].

The adoption of trajectory prediction methods based on a machine learning (ML) approach for unmanned and manned platforms [16–26] allows to achieve a modular configuration of the prediction tool, which has several advantages with respect to other methods. Moreover, the ML methods do not need the development of model-based algorithms and the vehicle dynamic parameters are not required. By making use of deep learning, the prediction accuracy can be improved by adding new samples to the training database; even new features can be accounted for without needing to develop the overall process again by only upgrading the neural network (NN) structure and/or the training database. Another advantage is the deterministic operational time because the run-time of a given NN is independent of the specific input, which makes these methods eligible for real-time applications. Considering the need to train the developed neural network, several databases are currently available to replicate aircraft performance. Moreover, to obtain the needed samples, it is also possible to realize a custom database and to exploit any of the available NN libraries for training, validation, and the test phase, also to compare different structure solutions and training strategies, and selected the most appropriate approach.

On the other hand, the drone market is driven by an increase in user demand and so advanced methods are needed to manage a growing unmanned traffic at low altitude for different types of missions [27–29].

Moreover, current drone regulations have undergone several changes in the last few years. Many challenges are related to the operative environment in which the drones fly and the employment of several on-board systems that are characterized by different performance. The EASA's effort is to realize a homogeneous European regulatory scenario [30] to avoid disagreement among member countries rules. Thanks to the introduction of “open”, “specific”, and “certified” categories for drone operations described in the European Regulations [31,32], the risk level is assessed by evaluating, not only the drone technical specifications, but also the surrounding environment and the operative conditions. The reliable prediction of the flight-time employed by the platform to fly a planned path allows the optimal management of flight plans during the pre-tactical phase, in particular to reduce the collision risk among drones, supporting UTM development [33,34].

This paper aims to develop a data-driven machine-learning-based method to predict drone flight time during a generic path. Several flight missions were performed to collect the training and test databases needed by the neural network (NN); each path was planned considering a standard characteristic angle between the consecutive segments. Analyzing different NN architectures, the best performance solution was identified and employed to compute the drone flight time over generic test paths, comparing the obtained results with the real time values. At the end, a possible UTM scenario for conflict detection is proposed, considering two crossing paths and adopting this method to predict the time to the closest point of approach (CPA).

2. Methodology

The presented study aims to predict the drone flight time needed to fly a planned path, adopting a machine learning method. A generic path is defined as a series of segments between planned waypoints. When the drone arrives at the waypoint, it performs a heading turn to the next waypoint. Around each waypoint, a corner can be identified selecting proper length legs, that are the segment parts before and after the turn. The total flight time over a generic path can be computed by summing the flight-time employed by the drone to fly a corner and the remaining parts of the segments that are out of the corner legs. The path can be cut so that distinguishing the corner sub-paths, which include the two legs of each corner, and the remaining parts; an example is shown in Figure 1. This is the strategy that is often used for path planning applications [35].

Figure 1. Example of the sub-path identification around the corners over a generic path.

After the evaluation of several commercial-off-the-shelf platforms, DJI Mavic 2 Enterprise[®] [36] was adopted for the operative phase for path execution, as shown in Figure 2. To manage and plan the flight paths, Ground Station Software UgCS[®] [37] was employed. The flight plans were uploaded as a series of waypoints (latitude, longitude and above ground level height) and the automatic mode was selected.

Figure 2. Drone model employed for the flight tests.

Several standard flights were performed in order to compute the flight time over different geometry corners. The standard paths included corners that were characterized by standard relative angles between the legs, in particular multiple 30-deg angles. Each standard path was defined by an identification symbol “TX”, where X was the angle between consecutive segments. TX path was specialized to collect the data that characterize the corners of size X degrees. Seven different paths were considered from the T0 path to the T180 path with step of 30 deg; it was assumed that right- and left- turns had analogous, symmetrical characteristics; asymmetry due to wind direction was accounted for separately as a specific NN input. During the T0 path, the drone came back along the same segment

in the opposite direction; instead, during the T180 path, the drone goes straight along two segments of the same length, stopping over an intermediate waypoint. The paths were designed to perform closed trajectories in the east north up reference frame. A full description of the performed paths is reported below:

- T0 path: Includes 5 segments of 30-m length that form a star shaped geometry, 5 external waypoints and one central waypoint, as shown in Figure 3a. Each segment is performed outwards and inwards with respect to the central waypoint so that it represents the case of a relative angle of 0 deg. Corners characterized by an angle of 72 deg that are performed in the center of the path were not considered as samples.
- T30 path: Includes 5 segments and 5 planned waypoints, as shown in Figure 3b. The first 4 segments have a 30-m length, and the corner legs involve a relative angle of 30 deg. Instead, the last segment has a 31-m length because it only allows to close the path. So, the relative angles of the couple legs fourth-fifth and fifth-first have not 30-deg size and the flight-time over them was not considered.
- T60 path. Includes 3 segments that represent an equilateral triangle shape and 3 planned waypoints.
- T90 path: Includes 4 segments that represent a square shape and 4 planned waypoints.
- T120 path: Includes 6 segments that represent a hexagon shape and 6 planned waypoints.
- T150 path: Includes 12 segments that represent a dodecagon shape and 12 planned waypoints.
- T180 path: Includes 2 segments and 3 planned waypoints. The path is a straight line with an intermediate stop at the central waypoint.

Figure 3. Examples of path design ENU reference frame centered at the first waypoint. (a) T0 path; (b) T30 path.

Considering the turn maneuver implementation, the time employed by the drone to change the heading is related to the amplitude of the path characteristic angle: the smaller the relative angle between consecutive segments is, the more time is needed to change the heading angle and so the corner flight-time increases.

In view of realistic future applications of drones, all the tests were conducted considering a ground speed limit of 5 m/s according to Regulation (EU) 2020/639 regarding the definition of the standard scenario STS-01 for “VLOS over a controlled area in an urban populated environment” [38]. A constant height of 20 m AGL was set for test development, in such a way as to have a controlled ground area that is compatible with several operations. However, if the height value changes, the controlled ground area specifications must be modified according to [38].

The underlying model of flight operations assumes that the drone flies at a constant speed along each trajectory segment between two waypoints except in proximity of each waypoint corresponding to a turn, i.e., a corner. The “stop and turn” mode was selected in order to guarantee a safe path execution, thus allowing the remote pilot to check the surrounding operative conditions by means of strapdown forward-looking cameras. Because of the turn maneuver and, standing the “stop and turn” mode set during the flight, each path segment involves an acceleration phase after leaving a waypoint and a deceleration phase when approaching the next one; the nominal ground speed is maintained only in the middle part of each segment. These sections are referred to as stable-speed sections in the following. On the whole, a segment between two waypoints consists of three sections: first the acceleration section, then the stable-speed section, and finally the deceleration section. For the best accuracy of the proposed technique, corner legs shall be long enough to fully include the acceleration and the deceleration sections to guarantee that the drone flies at the selected nominal ground speed out of the corners. Hence, preliminary flight test shall be carried out to fix the adequate corner legs’ length.

3. Initial Flight Test Campaign

Specifically, initial tests were performed at the selected nominal ground speed of 5 m/s and under the same wind conditions, considering different lengths of the linear segments. In the initial test phase two paths were tested, such as:

- T60 path;
- T180 path.

The above paths were planned for both 30-m and 40-m segment lengths, in order to find the smaller one, guaranteeing a constant ground speed of the drone out of the corners. Each path was flown three times. Each path segment involved an acceleration phase and a deceleration phase because the drone stops when reaching the waypoint. To conduct the presented flight tests, a constant speed segment was required along each segment between consecutive waypoints because the drone needed to enter and exit the corner at the same speed. The preliminary test drone telemetry was analyzed to identify the linear segments’ middle section flown at the nominal speed. The searched sections were identified as those in which the telemetry ground speed was never lower than one standard deviation below the nominal one. Then, the distance and the flight-time employed by the drone to cover the boundaries of the speed range were computed for each segment of the planned paths.

The results obtained from the drone telemetry data for the 40-m-segment path were compared with the ones obtained for the 30-m-segment path and are reported in Table 1. The reported statistics were computed over the identified stable-speed sections and then averaged for all the flown T60 and T180 paths, respectively.

Table 1. Preliminary test results of the T180 paths. Statistics are computed over the identified stable-speed sections of each flown waypoint-to-waypoint segment and then averaged for all the flown T60 and T180 paths, respectively.

	T60		T180	
	40 m	30 m	40 m	30 m
Ground speed mean value (m/s)	5.0	5.0	5.0	5.0
Ground speed std deviation (m/s)	0.2	0.2	0.2	0.2
Distance corner side length mean value (m)	32	22	33	22
Distance corner side length std deviation (m)	0.6	0.5	0.4	0.4
Flight-time mean value (s)	6.5	4.5	6.6	4.5
Flight-time std deviation (s)	0.13	0.07	0.16	0.12

A comparison between the different length paths is shown in Figure 4. Figure 4a,c displays the constant speed segments along the T180 paths for each of the three laps. Figure 4b,d reports the speed intensity along the T180 paths during the three laps. Consid-

ering the 40-m-segment, a standard 40-m corner can be defined around each turn, instead, for the 30-m-segment the reference corner length was 30 m. In brief, the stable-speed sections were practically in the middle of the trajectory segment, the ground speed was well controlled (mean equal to the nominal value, standard deviation within 5% of the nominal speed), and acceleration and deceleration section length was constant (4 m). So, the paths based on the 30-m length segment can be used, which means that 15-m corner legs were considered as reference turns. Moreover, a 15-m length is in the order of the size of a building. Therefore, this length was considered adequate for the proposed UTM application.

Figure 4. Ground speed analysis of the T180 path. (a) T180 path with 40-m sides, ENU reference frame centered at WP1; (b) ground speed amplitude behavior along the path for the 40-m T180 path; (c) T180 path with 30-m sides, ENU reference frame centered at WP1; (d) ground speed amplitude behavior along the path for the 30-m T180 path.

To plan the test campaign, the proper number of samples needed to realize the database was computed using the Chi-square method described in Equation (1) from [39]. Considering a confidence level of 95%, the Chi-Squared variables were $\chi_1^2 = 74.2219$ and $\chi_2^2 = 129.561$. The uncertainty parameters were referred to as the computed time measures from the drone telemetry. The desired standard deviation σ_s was set to 1 s for the presented study. Instead, the maximum value of the standard deviation σ was evaluated from previous test experiences [40], because the described preliminary tests involved too few samples to estimate a proper value of the standard deviation. Moreover, half of the samples coming from the preliminary tests were related to the T180 path that presents a lower standard deviation with respect to the other paths because the heading change is not performed during the straight path. So, previous tests data were employed in order to estimate the minimum number of samples required for each path using the right part of (1).

$$\frac{(n-1)\sigma_s^2}{\chi_2^2} < \sigma^2 < \frac{(n-1)\sigma_s^2}{\chi_1^2} \quad (1)$$

The maximum value of needed samples n was calculated using (1) and it was about 60 in the worst case. Thus, the new test campaign was performed and then the hypothesis needing 60 samples in the worst case was verified. Indeed, the maximum value of σ evaluated from the new performed flight tests was $\sigma = 0.87s$. Thus, at least 57 samples are required for the worst case. However, considering the specific value of the standard deviation for each type of path, as it results after the flight test campaign, the number of samples for some paths can be reduced according to (1). The minimum number of samples for each path is reported in Table 2.

Table 2. Evaluation of the minimum number of samples required for each template path.

Path ID	n Samples	Std (m)
A0	17	0.5
A30	7	0.3
A60	8	0.3
A90	18	0.5
A120	57	0.9
A150	26	0.6
A180	5	0.2

4. Flight Tests

After test execution, the collected telemetry was analyzed in order to evaluate the deviations in the estimated travel time on the performed path with respect to the planned one. The initial study was focused on the following aspects:

- deviations of the performed trajectory with respect to the planned 20-m height AGL;
- deviations of the performed trajectory with respect to the planned 30-m length distance around each corner.

Regarding the height keeping tracking, DJI Mavic 2 Enterprise[®] can be considered as compliant with UTM requirements. The main values and the deviations were computed for each path. Analyzing the results reported in Table 3, it is possible to confirm the constant height hypothesis.

Table 3. Analysis of the height keeping performance along the template paths.

Path ID	Height Mean Value (m)	Height Std (m)
A0	19.9	0.2
A30	20.0	0.2
A60	20.0	0.2
A90	19.9	0.2
A120	20.0	0.2
A150	20.0	0.2
A180	19.9	0.2

Regarding the corner part execution, the flight time employed to complete the corner was affected by the trajectory deviations involved in the 30 m length distance performance. The nominal corner starting and final points were computed considering 15-m corner legs with respect to the corresponding waypoints.

Then, the real corner starting and final points were identified, evaluating the telemetry point nearest to the nominal corner points. Thus, the distance traveled from the real corner starting point to the real corner final point is computed and compared with the ideal 30-m length corner. The results are reported in Table 4 and they show that the performed corner trajectory matched the planned route.

Table 4. Analysis of the corner execution performance in the horizontal plane for the planned paths.

Path ID	Corner Length Mean Value (m)	Corner Length Std (m)
A0	30.0	0.4
A30	30.0	0.6
A60	30.0	0.4
A90	30.0	0.5
A120	30.0	0.4
A150	30.0	0.4
A180	30.0	0.4

The time taken by the drone to complete a corner—including in the different planned paths—was computed from the telemetry data. The mean value and the standard deviation were computed for each path type, i.e., for each corner class. The time data variability was analyzed and a threshold of 3σ was applied in order to identify the samples to discard singular cases, i.e., cases above 3σ . The new reduced database was built, and it included 442 samples. The mean value and the standard deviation were also evaluated in this case to verify the reduction of the data variability. The final number of valid samples for each path is reported in Table 5; the total number of samples was 442 and they constituted the reference database for NN training and the test phase.

Table 5. Total number of the valid samples for each performed path.

Path ID	Number of Valid Samples
A0	75
A30	55
A60	58
A90	69
A120	59
A150	68
A180	58
Total Number	442

5. Method Based on Machine Learning

The developed neural network was adopted as a classifier; this means that the samples could be defined in a limited number of categories where each category is characterized by the same value of input parameters, i.e., the same turn angle and the wind condition.

The MATLAB® Deep Learning Toolbox™ was adopted to develop and train a NN aimed at predicting the drone flight time over a corner out of the defined classes. A feedforward NN was implemented and the Bayesian regularization backpropagation algorithm [41] was adopted to train the NN, which achieved better generalization performance, considering the data variability [42]. The selected input parameters were the normalized following ones:

- The relative angle between consecutive segments, i.e., corner legs. It is the clockwise angle computed from the second segment direction and the opposite of the first segment direction;
- The clockwise angle computed from the first segment direction and the wind vector;
- The wind intensity category.

The output was the time employed by the drone to perform a corner of 30 m in length.

The wind vector was evaluated from the local official weather forecast. The Beaufort scale [43] was adopted to classify the wind conditions in order to define a limited number of categories to employ in the developed neural network as classifiers. Only low wind intensity values could be considered in order to guarantee a proper safety level during the flight. Thus, the number of different wind conditions that allow safe flights were limited to

a few cases, and they were expected to have an impact mainly on battery discharge rather than on performance when not flying at the borders of the flight envelope, as in a typical nominal operational condition with the aim of safety. For several mission types, only the first three levels of the Beaufort scale were involved in the current applications, from 0 kts up to 6 kts. Thus, the presented approach presents a proof of concept that involves a typical wind intensity condition. The most common flight condition over the year in the test area is wind intensity of about 5 kt, according to the local weather archives. Thus, it is possible performing the flight tests at the same wind intensity condition of 5 kt (2nd Beaufort category). Moreover, the exploited wind vector direction was 60 deg with respect to the north direction. It is worth noting that limited wind conditions must be evaluated to fly UTM aircraft [31]. Specific wind models can be developed according to available standards, such as MIL-STD-1797A and MIL-F8785C; however, a classification process is required to group the wind intensities according to a limited number of categories.

The overall database acquired during the described flight tests was employed to train the neural network, according to the selected training function. A total of 85% of the data was employed for the algorithm training phase and the remaining 15% for the test phase. The validation dataset was not required by the Bayesian regularization backpropagation function. The maximum epoch number was set to 1000. In order to estimate the NN performance over the training and test datasets, the output percent error was computed using (2).

$$\sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{x_t - x_p}{x_t} \right)^2} \quad (2)$$

In (2), N is the number of samples, x_t is the time computed by the drone telemetry and x_p is the predicted time. The time values are referred to as a single corner execution. Different neural network architectures with two hidden layers were analyzed, in order to identify the best performance solution. Figure 5 shows the values obtained using (2) during the test learning phase corresponding to different numbers of total neurons in the NN layers. For each reported value, the best performance architecture was selected, changing the neurons displacing in each layer. The NN that involves two hidden layers with 4 and 6 neurons, respectively, was identified as the best solution. The structure of the selected NN is displayed in the Figure 6 thanks to the MATLAB[®] Deep Learning Toolbox[™]. The performance of the selected architecture over the training and test database is reported in Table 6.

Figure 5. Neural network performance evaluated using (1) with respect to the total number of neurons.

Figure 6. Selected neural network configuration developed by the MATLAB[®] Deep Learning Toolbox[™].

Table 6. Neural network performance evaluated using (2). The neural network includes two hidden layers with 4 and 6 neurons, respectively.

NN Performance	Training Phase	Test Phase
Percentage Error	4.5%	5.0%

6. Analysis of Complete Paths

Exploiting the described NN architecture, generic paths were analyzed in order to test the presented method, comparing the obtained results with the real flight-time values. Considering future applications of drones in urban scenarios, the paths were designed fitting the architecture of real urban environments, and then they were performed at a proper test site. The exploited wind condition was the same as that of the training paths.

6.1. Travel Time Prediction over Complex Paths

The flight time computed by means of the telemetry was evaluated as the flight time from the start (from the first waypoint). Instead, the flight time predicted by the NN is computed as the sum of the estimated travel time over the performed corners. If the path included segments that were longer than 30 m, the added time values were computed dividing the traveled distance by the nominal speed of 5 m/s. This hypothesis can be accepted because the considered segments were included in the constant speed part of each segment.

Four test paths were performed in order to test the method and they are classified in two test cases, called Case 1 and Case 2. Case 1 includes Test path 1 and Test path 3 that have only same length segments of a 30-m length; instead, Case 2 case includes Test path 2 and Test path 4 that have different length segments with a minimum segment length of 30 m. This classification was defined because Case 1 allows to test the proposed method summing the NN outputs, instead, Case 2 allows to test the method with more general paths.

1. Case 1. Test path 1 was planned for a monitoring flight. For instance, it can be adopted for a drone to monitor a crowded area. The urban scenario selected for this example is the area in front of a university hall. The path is represented in Figure 7a,b and it includes 10 segments, each one has a length of 30 m. The turn angles between consecutive segments are multiples of the 30 deg angle. The path was performed three times. Summing the NN outputs to compute the total flight time over the path, the first and the last segment of the complete three-lap-path are not included in a corner, because they are included in a half-corner. However, this contribution can be neglected with respect to the order of magnitude of the involved flight-time values. Test path 3 includes 9 segments that have the same length of 30 m, and it was performed three times. It is the red path displayed in Figure 8.
2. Case 2. Test path 2 was designed considering an urban canyon scenario among buildings. The path involves 9 segments, as reported in Figure 7c,d, and it was performed once. In this case different length segments were considered to realize the path, assuming a minimum segment length of 30 m. To estimate the flight-time

employed to travel the first and the last segment, an additional corner coming from the T180 path was considered. Test path 4 included 8 segments that have different lengths, and it was performed three times. It is the blue path displayed in Figure 8.

(a)

(b)

(c)

(d)

Figure 7. Planning of the test paths. (a) Planning of the Test 1 path, ENU reference frame centered at WP1; (b) Test 1 path view in an urban environment; (c) planning of the Test 2 path, ENU reference frame centered at WP1; (d) Test 2 path view in an urban environment.

The flight-time values predicted by the proposed method over the four test paths were compared with the real values obtained by the telemetry data. Moreover, the flight-time employed to perform the described paths was also evaluated dividing the planned distance by the nominal velocity of 5 m/s to define a benchmark for the method test. As reported in the Table 7, the proposed method allows to predict the flight-time value with a percentage error less than 1% over the shortest time path. However, even for the longer time paths, the evaluated percentage error was less than 3.2% and it fit the needs for guaranteeing a proper flight-time trajectory prediction, as it resulted from the comparison with the defined benchmark.

Figure 8. Trajectories of the crossing paths. ENU reference frame centered at WP1. The black star shows the real CPA and the black circle shows the predicted CPA during the first lap.

Table 7. Comparison between the flight time computed by telemetry data and by the presented machine-learning-based method for the test paths.

Method Performance	Test Path 1 3 Laps	Test Path 2 1 Lap	Test Path 3 3 Laps	Test Path 4 3 Laps
Real flight time (s)	272.3	97.1	239.6	245.3
Predicted flight time (s)	263.6	96.9	240.1	250.2
Predicted percentage error	3.2%	0.2%	−0.2%	−2.0%
Flight time at nominal speed (s)	180.0	69.1	162.0	177.0
Benchmark percentage error	34%	29%	32%	28%

6.2. Prediction of the Closest Point of Approach

Stating the mentioned results, a UTM application was investigated considering Test path 3 and Test path 4. These paths were performed using the same drone at different times. The telemetry position data of Test path 2 were processed in order to translate the path and intersect it with Test path 3. The resulting geometry is displayed in Figure 8, where Test path 3 is the red path and Test path 4 is the blue path. The telemetry time data of Test path 2 were synchronized to the first time instant of Test path 3. In this way, the described geometry can simulate two drones that, respectively, start the intersecting paths of Test path 3 and Test path 4 at the same time.

From the telemetry data, the closest point of approach, i.e., the point where the two drones are at a minimum distance of separation, was identified for each of the three laps of both paths. The time needed to reach the identified CPA from the initial time of the two paths was evaluated. The telemetry data were updated by the drone software considering a time difference of less than 0.3 s and thus it affected the estimation of the true drone position over the two paths.

Considering the described machine-learning-based method, the time to the CPA was computed as follows. The method allows to predict the flight time employed by the drone in strategical points of the path, i.e., at the waypoint, at the initial and final corner points, and along stable speed segments out of the corner, if present. For each known time value at the strategic points for the Test path 3 drone—that becomes the reference drone—the position of the Test path 4 drone was computed if these position data were related to strategical path points; otherwise, the position of the Test path 4 drone was estimated by

interpolating the known position data and evaluating the distance values between the two drones. Then, the described procedure was repeated considering the Test path 4 drone as the reference drone. Thus, the CPA is the point where the drones are at the minimum distance. The time to CPA was identified for each of the three laps as the minimum value of the distance between the drones during the execution time. The CPA condition must be analyzed in order to evaluate if the computed minimum distance is below a safety threshold in order to operate corrective actions.

The results of the prediction of the CPA are shown in Figure 9. The red line reported in Figure 9 represents the behavior of the distance parameter in time evaluated by the telemetry data. As the results show, the drone positions can be computed at very short time intervals. The blue stars identified in Figure 9 were computed starting from the known positions of one drone in the mentioned strategic points and through the interpolation of the positions. The blue stars refer to both cases in which the Test path 3 drone and the Test path 4 drone are, respectively, the reference drone.

Figure 9. Comparison of the distance among the drones during the path execution between the trajectory data and the output of the proposed method.

The time values to CPA computed by the telemetry data and by the proposed method are reported in Table 8 for each of the three laps. The difference between the real and the predicted value was less than 2 s after the first lap and 5 s after the third lap. The percentage error—computed dividing the difference between the real and the predicted value by the real value—is about 2.3% of the overall time to go in the worst case, i.e., after the third lap. Moreover, considering an example of management of 100 drones that are flying in the same area at a time interval of 5 min, about 1 million elementary operations are needed. Thus, the proposed approach can be implemented in UTM consoles.

Table 8. Flight time to the closest point of approach computed by the telemetry data and by the proposed method.

Time to CPA	1 Lap	2 Laps	3 Laps
Real time to CPA (s)	58.5	138.9	220.0
Predicted time to CPA (s)	57.3	140.8	215.0

The achieved performance shows that the described method can be applied to predict future risk of poor separation among drones after the acquisition of a proper database, supporting the development of standard UTM procedures within a proper timeframe to plan adequate avoidance maneuvers.

7. Conclusions

The presented paper exploits machine learning techniques to predict the flight time employed by a drone to complete a generic path. At the beginning, flight tests were performed with a commercial off-the-shelf drone by dividing a complex path into a sequence of standard paths to acquire a proper database in standard flight tests. The database was employed to train a neural network. Different neural network architectures were analyzed, and the best solution was identified. To assess the overall performance of the proposed method four generic paths were designed considering an urban environment scenario. The flight time employed by the drone to complete each path was computed from the telemetry data and compared with the flight-time forecast by the trained neural network, employing a benchmark to analyze the results. An error on the order of 3% of the travel time has been verified. At the end, a possible unmanned traffic management scenario was analyzed, computing the time to the closest point of approach and evaluating the method performance. The error in the prediction of the time at the closest point of approach resulted on the order of 2% of the current time to go. The proposed method was developed considering a selected drone model; however, a specific database can be acquired employing a different platform for test execution. Thus, the presented paper can support efficient conflict detection procedures that are required for the safe integration of drones in civil areas.

Author Contributions: Conceptualization, D.A.; methodology, D.A. and G.R.; software, C.C.; validation, G.R.; formal analysis, G.R.; investigation, C.C.; writing—original draft preparation, C.C.; writing—review and editing, R.S.L.M. and G.d.A.; visualization, G.d.A.; supervision, D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The flight tests were performed using the Ground Station Software UgCS® with the Education Program of the manufacturer.

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

References

1. NextGen; SESAR. *State of Harmonization*, 3rd ed.; Publications Office of the European Union: Luxembourg, 2018; Available online: <https://www.sesarju.eu/newsroom/brochures-publications/state-harmonisation> (accessed on 13 May 2021).
2. Mondoloni, S.; Rozen, N. Aircraft trajectory prediction and synchronization for air traffic management applications. *Prog. Aerosp. Sci.* **2020**, *119*, 100640. [CrossRef]
3. Prandini, M.; Hu, J.; Lygeros, J.; Sastry, S. A probabilistic approach to aircraft conflict detection. *IEEE Trans. Intell. Transp. Syst.* **2000**, *1*, 199–220. [CrossRef]
4. Bashllari, A.; Kaciroti, N.; Nace, D.; Fundo, A. Conflict Probability Estimations Based on Geometrical and Bayesian Approaches. In Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference, Bellevue, WA, USA, 30 September–3 October 2007; pp. 479–484. [CrossRef]
5. Zhou, Z.; Chen, J.; Shen, B.; Xiong, Z.; Shen, H.; Guo, F. A trajectory prediction method based on aircraft motion model and grey theory. In Proceedings of the 2016 IEEE Advanced Information Management, Communication, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 3–5 October 2016; pp. 1523–1527. [CrossRef]
6. Schuster, W. Trajectory prediction for future air traffic management—Complex manoeuvres and taxiing. *Aeronaut. J.* **2015**, *119*, 121–143. [CrossRef]
7. Base of Aircraft Data (BADA). Available online: <https://www.eurocontrol.int/model/bada> (accessed on 13 May 2021).

8. Uzun, M.; Koyuncu, E. Data-Driven Trajectory Uncertainty Quantification For Climbing Aircraft To Improve Ground-Based Trajectory Prediction. *Anadolu Univ. J. Sci. Technol. Appl. Sci. Eng.* **2017**, *18*, 1. [\[CrossRef\]](#)
9. Fernández, E.C.; Cordero, J.M.; Vouros, G.; Pelekis, N.; Kravaris, T.; Georgiou, H.; Fuchs, G.; Andrienko, N.; Andrienko, G.; Casado, E.; et al. DART: A Machine-Learning Approach to Trajectory Prediction and Demand-Capacity Balancing. In Proceedings of the Seventh SESAR Innovation Days, Belgrade, Serbia, 28–30 November 2017.
10. Sun, J.; Ellerbroek, J.; Hoekstra, J. Modeling Aircraft Performance Parameters with Open ADS-B Data. In Proceedings of the 12th USA/Europe Air Traffic Management Research and Development Seminar, Seattle, WA, USA, 27–30 June 2017.
11. Xu, G.; Long, T.; Wang, Z.; Cao, Y. Matrix Structure Driven Interior Point Method for Quadrotor Real-Time Trajectory Planning. *IEEE Access* **2019**, *7*, 90941–90953. [\[CrossRef\]](#)
12. Xiang, L.; Lei, L.; Chatzinotas, S.; Ottersten, B.; Schober, R. Towards Power-Efficient Aerial Communications via Dynamic Multi-UAV Cooperation. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea, 25–28 May 2020; pp. 1–7.
13. Hu, S.; Wu, Q.; Wang, X. Energy Management and Trajectory Optimization for UAV-Enabled Legitimate Monitoring Systems. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 142–155. [\[CrossRef\]](#)
14. Zhang, X.; Ma, J.; Cheng, Z.; Huang, S.; Ge, S.S.; Lee, T.H. Trajectory Generation by Chance-Constrained Nonlinear MPC With Probabilistic Prediction. *IEEE Trans. Cybern.* **2020**, 1–14. [\[CrossRef\]](#)
15. Sun, C.; Ni, W.; Wang, X. Joint Computation Offloading and Trajectory Planning for UAV-Assisted Edge Computing. *IEEE Trans. Wirel. Commun.* **2021**, 1. [\[CrossRef\]](#)
16. Gui, G.; Liu, F.; Sun, J.; Yang, J.; Zhou, Z.; Zhao, D. Flight Delay Prediction Based on Aviation Big Data and Machine Learning. *IEEE Trans. Veh. Technol.* **2020**, *69*, 140–150. [\[CrossRef\]](#)
17. Wang, X.; Liu, J.; Qiu, T.; Mu, C.; Chen, C.; Zhou, P. A Real-Time Collision Prediction Mechanism With Deep Learning for Intelligent Transportation System. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9497–9508. [\[CrossRef\]](#)
18. Xue, M. UAV Trajectory Modeling Using Neural Networks. In Proceedings of the 17th AIAA Aviation Technology, Integration, and Operations Conference, Denver, CO, USA, 5–9 June 2017.
19. Alligier, R.; Gianazza, D. Learning aircraft operational factors to improve aircraft climb prediction: A large scale multi-airport study. *Transp. Res. Part C: Emerg. Technol.* **2018**, *96*, 72–95. [\[CrossRef\]](#)
20. Cherian, A.K.; Rai, A.; Jain, V. Flight trajectory prediction for air traffic management. *J. Crit. Rev.* **2020**, *7*, 412–416. [\[CrossRef\]](#)
21. Barratt, S.T.; Kochenderfer, M.J.; Boyd, S.P. Learning Probabilistic Trajectory Models of Aircraft in Terminal Airspace From Position Data. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 3536–3545. [\[CrossRef\]](#)
22. Wang, Z.; Liang, M.; Delahaye, D. Data-driven Conflict Detection Enhancement in 3D Airspace with Machine Learning. In Proceedings of the 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT), Singapore; 2020; pp. 1–9. [\[CrossRef\]](#)
23. Ma, L.; Tian, S. A Hybrid CNN-LSTM Model for Aircraft 4D Trajectory Prediction. *IEEE Access* **2020**, *8*, 134668–134680. [\[CrossRef\]](#)
24. Zeng, W.; Quan, Z.; Zhao, Z.; Xie, C.; Lu, X. A Deep Learning Approach for Aircraft Trajectory Prediction in Terminal Airspace. *IEEE Access* **2020**, *8*, 151250–151266. [\[CrossRef\]](#)
25. Hashemi, S.M.; Botez, R.M.; Grigorie, L.T. New Reliability Studies of Data-Driven Aircraft Trajectory Prediction. *Aerospace* **2020**, *7*, 145. [\[CrossRef\]](#)
26. Casado, R.; Bermúdez, A. Neural Network-Based Aircraft Conflict Prediction in Final Approach Maneuvers. *Electronics* **2020**, *9*, 1708. [\[CrossRef\]](#)
27. Galkin, B.; Kibilda, J.; DaSilva, L.A. A Stochastic Model for UAV Networks Positioned Above Demand Hotspots in Urban Environments. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6985–6996. [\[CrossRef\]](#)
28. Qadir, Z.; Ullah, F.; Munawar, H.S.; Al-Turjman, F. Addressing disasters in smart cities through UAVs path planning and 5G communications: A systematic review. *Comput. Commun.* **2021**, *168*, 114–135. [\[CrossRef\]](#)
29. Khoufi, I.; Laouiti, A.; Adjih, C.; Haddad, M. UAVs Trajectory Optimization for Data Pick Up and Delivery with Time Window. *Drones* **2021**, *5*, 27. [\[CrossRef\]](#)
30. Barrado, C.; Boyero, M.; Bruculeri, L.; Ferrara, G.; Hatley, A.; Hullah, P.; Martin-Marrero, D.; Pastor, E.; Rushton, A.P.; Volkert, A. U-Space Concept of Operations: A Key Enabler for Opening Airspace to Emerging Low-Altitude Operations. *Aerospace* **2020**, *7*, 24. [\[CrossRef\]](#)
31. Regulation (EU) 2019/945 of 12 March 2019 on Unmanned Aircraft Systems and on Third-Country Operators of Unmanned Aircraft Systems. Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019R0945> (accessed on 13 July 2021).
32. Regulation (EU) 2019/947 of 24 May 2019 on the Rules and Procedures for the Operation of Unmanned Aircraft. Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019R0947> (accessed on 13 July 2021).
33. Eurocontrol, Concept of Operations for European UTM Systems. Available online: <https://www.eurocontrol.int/project/concept-operations-european-utm-systems>. (accessed on 13 May 2021).
34. SESAR Joint Undertaking, “AMU-LED”. Available online: <https://www.sesarju.eu/projects/AMU-LED>. (accessed on 13 May 2021).
35. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
36. DJI, Mavic 2 Enterprise. Available online: <https://www.dji.com/it/mavic-2-enterprise>. (accessed on 13 May 2021).

-
37. UgCS. *Leading Drone Control Software to Elevate Your Productivity*. 2019. Available online: <https://www.ugcs.com/> (accessed on 4 March 2019).
 38. Regulation (EU) 2020/639 of 12 May 2020 Amending Implementing Regulation (EU) 2019/947 as Regards Standard Scenarios for Operations Executed in or Beyond the Visual Line of Sight. Available online: https://eur-lex.europa.eu/eli/reg_impl/2020/639/oj. (accessed on 13 July 2021).
 39. Chatfield, A.B. *Fundamentals of High Accuracy Inertial Navigation*; AIAA: Reston, VA, USA, 1997.
 40. Conte, C.; Accardo, D.; Rufino, G. Trajectory Flight-Time Prediction based on Machine Learning for Unmanned Traffic Management. In Proceedings of the 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 11–15 October 2020; pp. 1–6.
 41. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
 42. Hassan, H.; Ahmed, I.; Ahmad, R.; Khammari, H.; Bhatti, G.; Ahmed, W.; Alam, M.M. A Machine Learning Approach to Achieving Energy Efficiency in Relay-Assisted LTE-A Downlink System. *Sensors* **2019**, *19*, 3461. [[CrossRef](#)]
 43. Garbett, L.G. Admiral Sir Francis Beaufort and the Beaufort Scales of wind and weather. *Q. J. R. Meteorol. Soc.* **1926**, *52*, 161–172. [[CrossRef](#)]