

Systems Science & Control Engineering

An Open Access Journal

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tssc20

Drone motion prediction from flight data: a nonlinear time series approach

Shuyan Dong, Saptarshi Das & Stuart Townley

To cite this article: Shuyan Dong, Saptarshi Das & Stuart Townley (2024) Drone motion prediction from flight data: a nonlinear time series approach, Systems Science & Control Engineering, 12:1, 2409098, DOI: [10.1080/21642583.2024.2409098](https://doi.org/10.1080/21642583.2024.2409098)

To link to this article: <https://doi.org/10.1080/21642583.2024.2409098>



© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 06 Oct 2024.



Submit your article to this journal [↗](#)



View related articles [↗](#)



View Crossmark data [↗](#)

Drone motion prediction from flight data: a nonlinear time series approach

Shuyan Dong, Saptarshi Das and Stuart Townley

Centre for Environmental Mathematics, Faculty of Environment, Science and Economy, University of Exeter, Penryn Campus, Cornwall, UK

ABSTRACT

In this paper, we explore the application of data-driven predictive systems in enhancing unmanned aerial vehicle (UAV) control capabilities. We introduce a new model for predicting the motion of individual drones by utilizing fundamental flight control data. The model aims to improve the autonomy of individual drones and circumvent the complexity of traditional flight control systems, thus eliminating intricate nested controls. The proposed model lays the foundation for studying collective behaviours within a cluster of drones, thereby advancing the research into swarm behaviour exhibited by drones. The research findings demonstrate the potential of data-driven methods in the construction of UAV control systems. In particular, we here show a comparison of the prediction performances between two neural network architectures using real drone flight data involved in various kinds of motions. We explore the utility of using long short term memory (LSTM) and nonlinear autoregressive with exogenous inputs (NARX) family of nonlinear time series models in developing a virtual drone model using real experimental data.

ARTICLE HISTORY

Received 10 May 2024
Accepted 21 September 2024

KEYWORDS

Data-driven UAV model; artificial intelligence; time series analysis; motion prediction

1. Introduction

1.1. Drone control and technology support


As an emerging technology, unmanned aerial vehicles (UAVs) are increasingly being used across various fields, including power infrastructure inspection, security and fire protection, agriculture and farming, industry and the media (Deng et al., 2014; Skorput et al., 2016; Van der Merwe et al., 2020; Laghari et al., 2023). Autonomous drones enable closer, safer and more precise inspection of large industrial sites or hazardous equipment, especially offering the capability to gather visual or other survey data in extreme or perilous environments more effectively than humans (Deng et al., 2014; Nooralishahi et al., 2021; Samiappan et al., 2017).

Although the application of UAVs is expanding progressively, traditional control systems face difficulties in adequately addressing the dynamic nature of real flight conditions, resulting in limited autonomy and adaptability (Guzmán & Häggglund, 2024; Jembre et al., 2021). Traditional control systems are typically reliant on pre-determined flight plans and a global perspective, making it hard for UAVs to adapt to the inherent complexity in real-world environments (Gunetti, 2012; Ippolito et al., 2023), which then struggle to achieve optimal autonomy. Therefore, there is a pressing need for innovative methods to address these challenges and enhance the control capabilities of UAVs. Introducing data-driven

approaches, models utilize a combination of theoretical principles and empirical data to adapt to various flight conditions and mission scenarios while maintaining computational efficiency. By capitalizing on the scalability and efficiency of data-driven techniques, predictive systems can optimize flight paths and resource utilization, thereby improving mission effectiveness and reducing operational costs.

The advancement of machine learning and artificial intelligence has inevitably influenced drone research. One notable application area is the development of data-driven control systems (Ding, 2008; Freeman et al., 2013; Guo et al., 2021; Li et al., 2016). These systems utilize actual historical flight data for trajectory prediction and optimization thus contributing to improvements in both autonomy and safety (Ding, 2008; Li et al., 2016).

The autonomous operation of drones relies heavily on their adaptive and real-time predictive capabilities, enabling swift responses to rapidly changing mission demands and intricate environments. Data-driven predictive systems enable UAVs to adapt in real-time by continuously analyzing sensor data and predicting future states. By integrating data-driven approaches into UAV control systems, these adaptive models not only enhance responsiveness to dynamic environmental changes but also facilitate autonomous decision-making and trajectory adjustments, thereby bolstering overall mission efficiency and safety (Chen et al., 2021).

CONTACT Shuyan Dong  sd746@exeter.ac.uk

© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

Data-driven predictive systems represent a potential method for enhancing UAV control capabilities. Our study aims to address a significant gap in UAV trajectory prediction research, where reliance on simulation results prevails over actual flight data, by developing innovative methods that utilize real flight data to enhance UAV control capabilities in dynamic environments. Anticipating the utilization of the data-driven predictive system, we expect to achieve more efficient and adaptive flight operations in dynamic environments. With access to historical flight data and advanced machine-learning techniques, these systems can refine trajectory prediction, optimize flight paths, and elevate overall mission efficiency and safety. We expect that this autonomous control system is able to replace the usage of controllers that are normally used for integrated planning in multi-agent control systems so that the individual drones in a flock can cooperate to attain a target without any additional controller involved.

This paper uses data analysis techniques to statistically learn drone motion and control variables, delving into the correlations between *Pulse Width Modulations (PWMs)*, *poses*, and *positions*. PWM is an electronic method employed to represent the information through a changing signal. It involves the simulation of an analogue signal through the creation of a sequence of digital pulses, with the adjustment of pulse width serving as a means to communicate information. Correlation-based feature selection is carried out on the data and, based on the correlation coefficients, a new Nonlinear Autoregressive with Exogenous inputs (NARX) model for position prediction is designed. Specifically, it explores the innovative concept of a neural network-based drones' motion prediction model in the realms of drone control and trajectory prediction. With the advanced data processing capabilities of the NARX model, this study aims to establish a new drone motion prediction system.

The core of this new technique lies in utilizing extensive historical trajectory data and the internal data of the drone to train the drone's motion-predicting model. These datasets cover multiple factors, including motor outputs, poses, and three-dimensional positional information of the drone. The NARX network aims to decipher the intricate relationships within this data, enabling it to forecast the subsequent motion of a drone when given a particular set of *PWMs* or *poses*.

The NARX-based prediction of drone motion is not only an innovative approach but also shows potential value. This new designed model holds promise for providing new solutions to optimize flight efficiency and enhance UAV autonomy, potentially aiding the aviation industry in achieving greater autonomy and sustainability. Through interdisciplinary collaboration and ongoing

research efforts, we believe that data-driven predictive systems will play a pivotal role in shaping the future of UAV technology.

1.2. Research content and significance

This paper is an extension of our previous work (Dong et al., 2023), with additional data analysis results. The previous paper explored the shortcomings of both centralized and decentralized systems and discussed whether it would be possible to overcome existing problems in the centralized control system and the decentralized system by introducing swarm intelligence to the control system. By harnessing the adaptive and self-organizing nature of swarm algorithms (Blum & Roli, 2003; Dorigo et al., 2006; Gao, 2020; Hassanien & Emary, 2016; Liu et al., 2021; Procaccini et al., 2011; Teodorović, 2008), a new flocking system may be designed with minimal human intervention. The envisioned framework seeks to transcend existing limitations, promising breakthroughs in UAV cluster control and performance optimization. With this primary blueprint for a swarm intelligence-inspired drone flocking control system, we conducted an initial exploration of individual flock members' basic autonomy. We present two nonlinear time-series modelling methods, namely Long Short-Term Memory (LSTM) and NARX, applied to actual drone flight data, and compare the performance of each class of model.

In contrast to the previous paper, here we carried out a further exploration of predictive modelling and model optimization based on real flight data. We embark on a comprehensive investigation by initially applying Spearman correlation to the dataset for nonlinear analysis. Subsequently, we revisit the comparative results of *LSTM* and *NARX* models in terms of predictive capabilities. Remarkably, it is observed that, despite undergoing identical training durations, the *NARX* model exhibits superior predictive performance.

Expanding on this, we delve into the nuances of the *NARX* model's effectiveness, dissecting its ability to capture intricate patterns within flight data and adapt to dynamic environmental conditions. Furthermore, we elucidate the potential implications of these findings for enhancing UAV control systems and optimizing mission performance in real-world scenarios.

Considering the shortcomings in the experimental results of the closed-loop *NARX*, we delve further into the design and construction of the predictive model structure. By reconsidering the features of the model, we establish that incorporating the *Stabilizers* as the input yields improved model performance. This model is named '*Sta_POS NARX*'. To explore the impact of different input delays and the feedback delays of the *NARX* model,

we construct 144 variations of the *Sta_POS NARX* model by altering the delay parameters. Each model repeated the training for five times to mitigate for the effects of training randomness, resulting in a total of 720 models. These 720 trained models are subsequently applied individually to various types of flight data. Based on their predictive performance, the optimal models for each set of flight data are selected.

The optimal *NARX* models are chosen for each trajectory to integrate them into the closed-loop control system we are developing to enhance the autonomy of drone motion. This new closed-loop control system design is a novel autonomous control approach for drones, eliminating the need for nested complex control loops. Additionally, the new control system facilitates the accumulation of a large volume of reference data without incurring the costs and losses associated with collecting data through actual flight experimentation.

Key contributions of the paper are:

- an innovative method for predicting drone movements by projecting their future flight positions based on essential data such as *PWMs* and *Pose*, circumventing the intricate computations associated with conventional flight control systems;
- a comparative assessment between the newly introduced *NARX* model and the commonly employed *LSTM* model in autonomous flight technology, demonstrating that the *NARX* model offers quicker training and superior fitting;
- a formulation of the '*Sta_POS NARX*' model to capitalize on the time-dependent characteristics of data and so improving predictive precision.

2. Material and methods

2.1. Experimental setting and data collection

To implement the data-driven control system, we recorded a large amount of flight data, including, but not confined to, small regular patterns, different unidirectional flying patterns such as going left, going right, taking off, landing, flying forwards, flying backwards, along with random walks limited within a specific range.

Our experiments used the *Crazyflie* (see Figure 1), an open-source flying development platform known for its versatility and weighing a mere 27g. Due to its lightweight construction, the drones have been utilized in small robotics or multi-robotics research and applications (Giernacki et al., 2017; Lambert et al., 2019; Preiss et al., 2017).

Based on the aforementioned flight data, we modelled the motion of the drones within a machine-learning



Figure 1. Authentic photographs of the drone, used to generate experimental data for modelling.

framework. This may help perform experiments such as obstacle avoidance on a virtual model. With the data-driven virtual model, we are able to digitize the motion of the drones from manual control to quantitative control. This will lead theoretically to smoother flights and improved controllability. Upon a successful run of the virtual experiment, the model outputs are assigned to the real drone so that it can be controlled directly. Remarkably, the results obtained from the test are observed to closely match those derived from the virtual experiment.

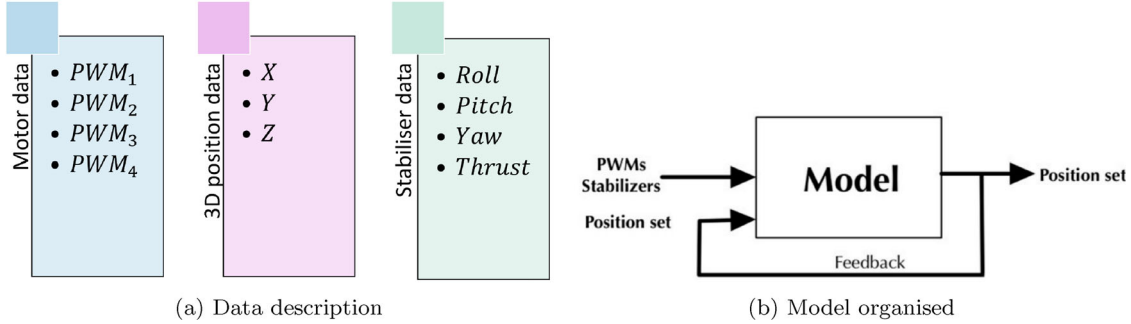
According to the mentioned experimental setup, data pertinent to the robotic control system, motor input data or pulse width modulation (*PWM*), 3D positional coordinates (X, Y, Z), and four features (*Roll*, *Pitch*, *Yaw*, and *Thrust*) from the robotic control system of the drone are recorded during the drone's operation. Comprehensive descriptions and sizes of these recorded data points are detailed in Table 1.

PWM inputs of the motors are recorded as the features of control technology for electronic power applications (Blaabjerg, 2018; W. Liu, 2013). *PWM* stands as an irreplaceable link in a stable and flexible method of controlling the speed of a DC motor (W. Liu, 2013). The three dimensions representing the position of the drone are recorded every 50 milliseconds. Roll, pitch, and yaw are terms used to describe the rotational movements of an object or body in three-dimensional space. *Roll* refers to the rotation around the longitudinal axis of an object. *Pitch* refers to the rotation around the lateral axis of an object. *Yaw* refers to the rotation around the vertical axis of an object. In the field of robotics, roll, pitch, and yaw are fundamental components for movement descriptions.

Flight data acquisition during drone operations inherently exhibits a temporal nature gathered over distinct time intervals. The inherent temporal and spatial dependencies in-flight data may result in adjacent data points being interrelated and potentially influencing each other. Moreover, flight data demonstrates spatial correlation,

Table 1. Flight data types used in the experiments and corresponding detailed descriptions with sizes.

Data name	Data description	Data size
Small Circle	A small circular motion including take-off and landing	300 × 11
Large Circle	A large circular motion including take-off and landing	325 × 11
Taking Off	100 repetitions of taking off and landing	5081 × 11
Going Forward	100 repetitions of going forward and backwards	720 × 11
Going Left	100 repetitions of going to the left direction	6441 × 11
Going Right	100 repetitions of going to the right direction	2802 × 11
Small Random	Multiple times of limited steps (50) random walk	7290 × 11

**Figure 2.** Variable arrangements and schematic of data generated to train the virtual drone model. (a) Data description and (b) Model organized.

wherein diverse flight paths and altitudes can significantly impact the characteristics of the data. In this study, we collected multi-dimensional flight data, consisting of motor data, flight position, and robotic control parameters. The specific parameters recorded and the schematic of data application to model structure are shown in Figure 2. These multiple dimensions of data are intricately related and require sophisticated analysis and modelling techniques to extract meaningful insights.

2.2. Data preparation and wrangling

Throughout flights, the recorded data encounters instances of noise and outliers, which can introduce inaccuracies and instabilities in data analysis and modeling. This instability might affect the flight data in certain scenarios. Hence, data processing is of crucial significance in model design.

First, we addressed missing data and smoothed outliers using a sliding window approach. This method involves calculating the average of data points within a fixed-size window (where we use 2 as the window's size in outlier replacement), that moves across the dataset. This technique helps fill in missing values and smooth outlier values by leveraging the local neighbourhood of each data point. Moreover, we normalized three distinct parameter groups to ensure that the data was on a comparable scale, facilitating better analysis and model performance.

We considered the outlier as y_i , and the replacement value y'_i is calculated as the average of the non-outlier

neighbours. Thus, the $y'_i = (y_{i-1} + y_{i+1})/2$. If y_{i-1} or y_{i+1} is also an outlier, we continue to find the next valid non-outlier neighbour.

In the flight experiments, drones experienced an uncontrollable lateral offset that became more severe with each repetition of the flying pattern. Treating this offset as a discernible trend, we utilized Matlab's built-in function *detrend* to reduce the effect of the offset.

After an initial analysis of the position data, a notable discrepancy in magnitude between *PWM* and *POS* became apparent. Given our intention to employ the *NARX* model for *POS* prediction, understanding neural network characteristics became imperative. A significant magnitude gap between input and output values might bias the network toward learning larger magnitude outputs, potentially neglecting smaller ones, which will lead to considerable inaccuracies in predicting the smaller magnitude outputs. Furthermore, this magnitude discrepancy might result in increased errors and reduced precision in forecasting *POS* with smaller magnitudes, consequently affecting the overall model accuracy.

To address this significant magnitude disparity between *PWMs* and *POS*, while maintaining feature correlation, we implemented data rescaling. In MATLAB, *rescaling* typically involves adjusting data to specific ranges or standardizing it for improved model training or data processing.

Rescaling *PWMs* to the range $[0, 1]$, aided in mitigating dimensional differences between *PWM* and *POS*. This ensured a more balanced weight distribution during model training, which improved model stability and

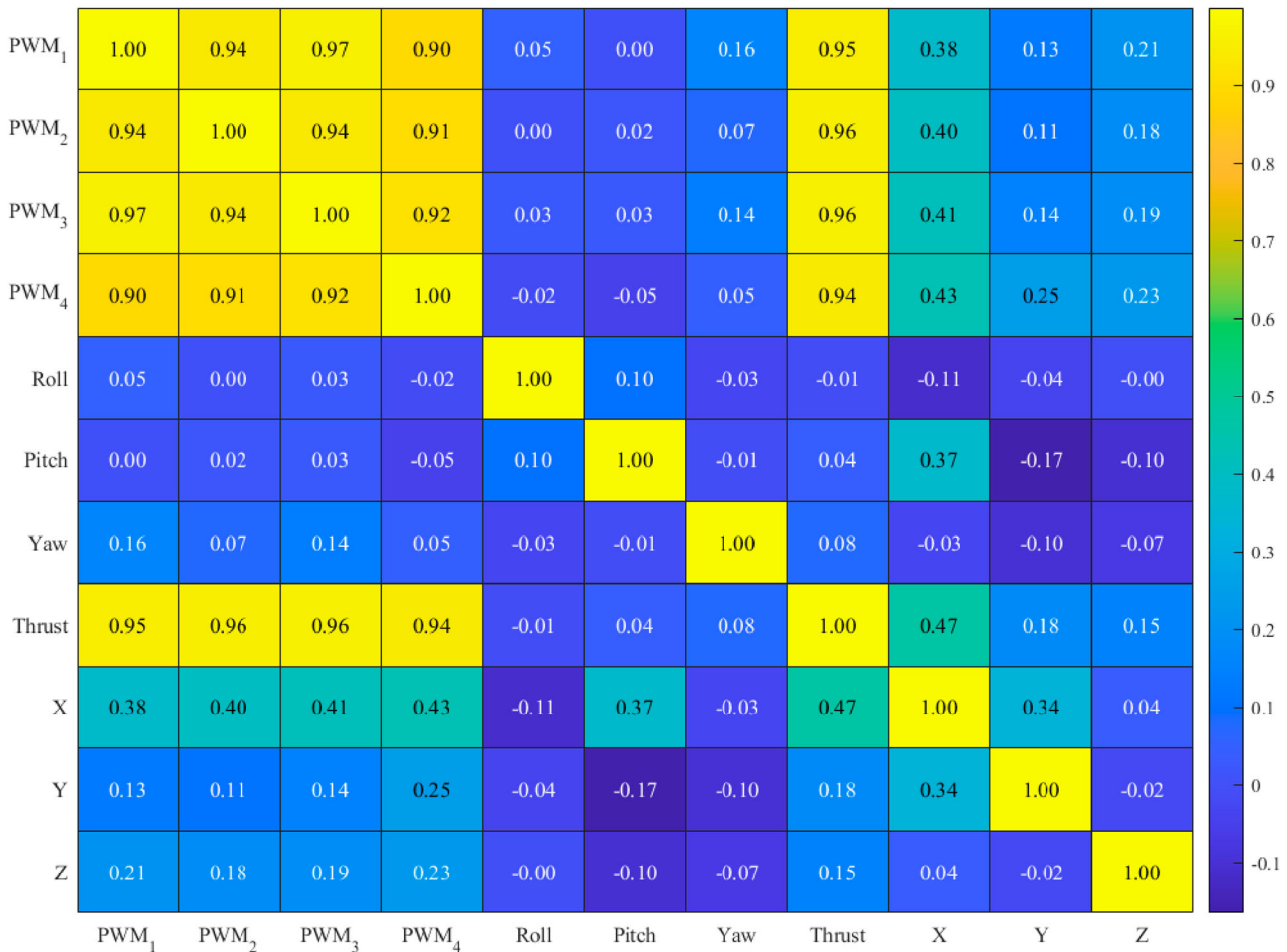


Figure 3. Feature correlations between 11 variables on the data including *Taking Off* experiment. This correlation matrix shows the correlations between pulse width modulation of four motors (PWM_{1-4}), drone's altitude (*Roll* and *Pitch*), heading angle (*Yaw*), reaction force (*Thrust*) and position (X, Y, Z).

convergence speed. Simultaneously, it aligned the general data within similar ranges, ensuring relative equilibrium in their influence on the model and preventing *PWMs* from dominating the model's training due to the larger numerical range.

2.3. Exploratory data analysis

To better understand the relationships between features, we investigated correlation metrics. The correlation metrics between pairs of features of the general flight data for multiple *detrended* flight trajectories were analyzed. However, when integrating flight data from different flight paths and performing data correlation analysis, it was found that the data recorded on *Taking off* would significantly impact the analysis. The schematic diagram of the correlation analysis of elements with and without *Taking off* data is shown in Figures 3 and 4, respectively.

It can be seen in Figure 3 that the position features (X, Y, Z) exhibit weak correlations with the majority of

other features and among themselves. Notably, a significant correlation is evident among *PWMs* ($PWM_1, PWM_2, PWM_3, PWM_4$ shown in Figure 3) when the dataset incorporates the *taking off* and the variable *thrust* demonstrates a high correlation with *PWMs*.

However, when only considering the rest of the data from the trajectories, the correlations of motors' output are not significantly high. Although *Thrust* shows a correlation of approximately 0.5 with the motors' output, the correlations are comparatively lower than that when including *Taking off* data (Figure 4).

It is interesting to see when modelling this specific behaviour of *Taking off*, we can exclude the output of certain motors as features, or use *thrust* as a substitute. This operation will not be performed for other behaviours. This is primarily because during take-off, maximum *thrust* and lift are required to overcome gravity and air resistance, resulting in a high correlation between *PWM* signals and *thrust*. In contrast, during other flight phases, the vehicle maintains equilibrium with balanced forces,

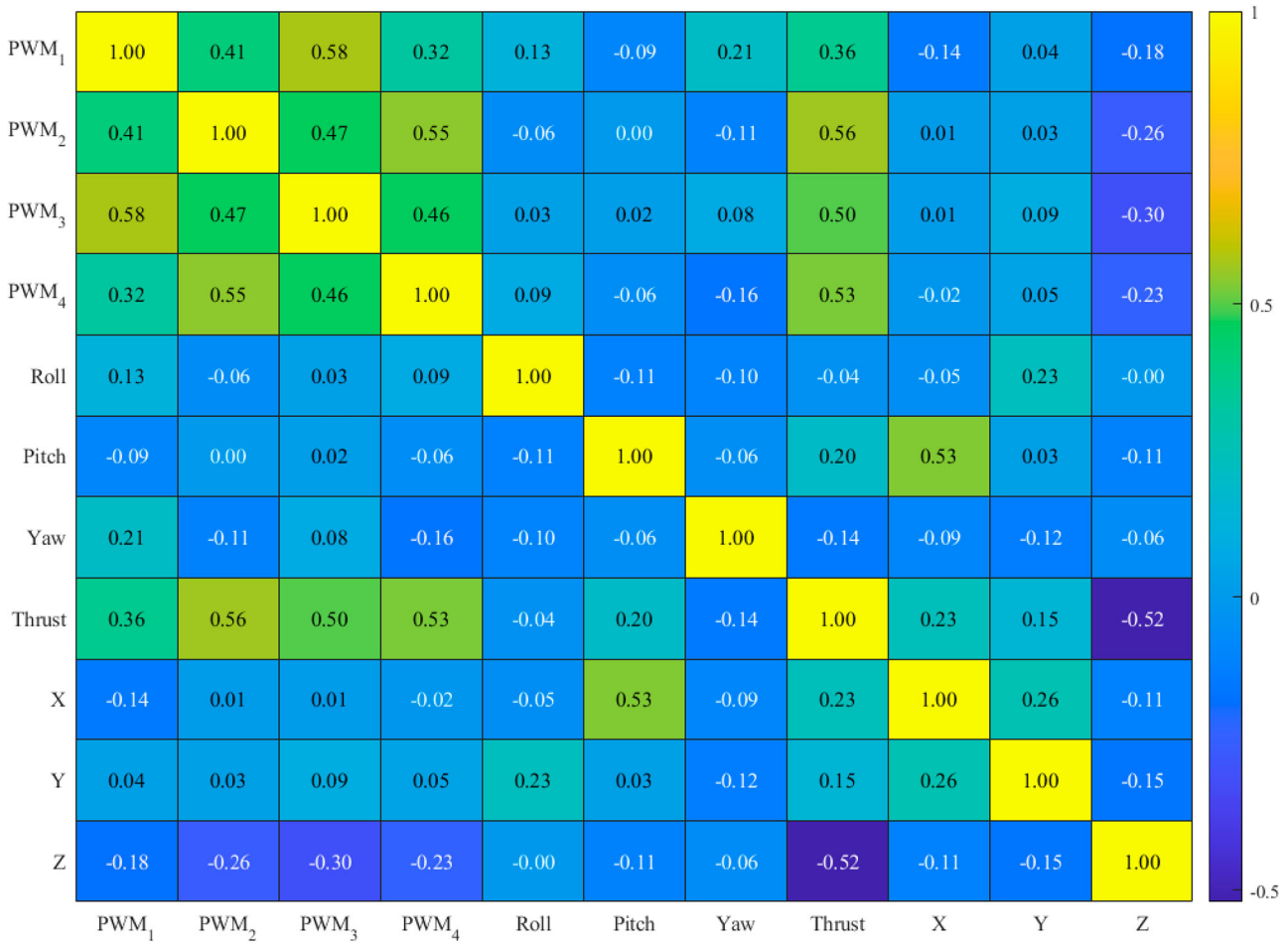


Figure 4. Feature correlations between 11 variables on the data without *Taking Off* experiment.

requiring only the necessary thrust to maintain speed and altitude, with specific engines adjusting rotation per minutes (RPMs) during maneuvers.

In the realm of flight control systems, *PWM* signals are utilized to regulate the motor speed of UAVs, thereby controlling the UAV's flight altitude and position. However, the relationship between *PWM* signals and *Stabilizer* (such as pitch, roll, and yaw) and *POS* is not always a simple linear correlation. This complexity arises due to intricate interactions between control algorithms, notably proportional integral derivative (PID) controllers within the flight controller system, and the dynamic behaviour of the UAV during flight.

Meanwhile, in traditional robotics control theory, through a series of computations involving altitude transformations, it is possible to deduce the current directional vector of the aircraft. This directional vector, combined with previously known positional information of the aircraft, can be integrated to compute the current coordinates of the UAV. Based on this, it can be broadly stated that the transformation between *Pose* and *position* involves multiple iterations of integral changes.

Therefore, we conducted the Pearson and Spearman correlation coefficient tests separately. The Pearson correlation coefficient assesses the strength of the linear relationship between two variables within a range of $[-1, 1]$. This measurement is ideally suited for continuous, normally distributed, and highly consistent linear data. Unlike this, the Spearman correlation is adept at measuring the strength of the non-linear relationships between two variables, upto a certain degree. Considering the inherent characteristics of drones, we expected to observe a lower correlation coefficient between the *PWMs* and *POS*, as well as between the *Stabilizers* and the *POS*. The specific correlation matrices are illustrated below (Figure 5). It shows a higher Spearman correlation coefficient between the *Stabilizers* and the positions than that between the *PWMs* and the *X* and *Y* directions. While, unlike the others, when providing *Small Random* to the drone, the *Z* direction position shows higher relevance with the *PWMs*. This observation implies that modelling based on *NARX*, focusing on the *Stabilizers* variables could yield greater accuracy in capturing and predicting positional changes.

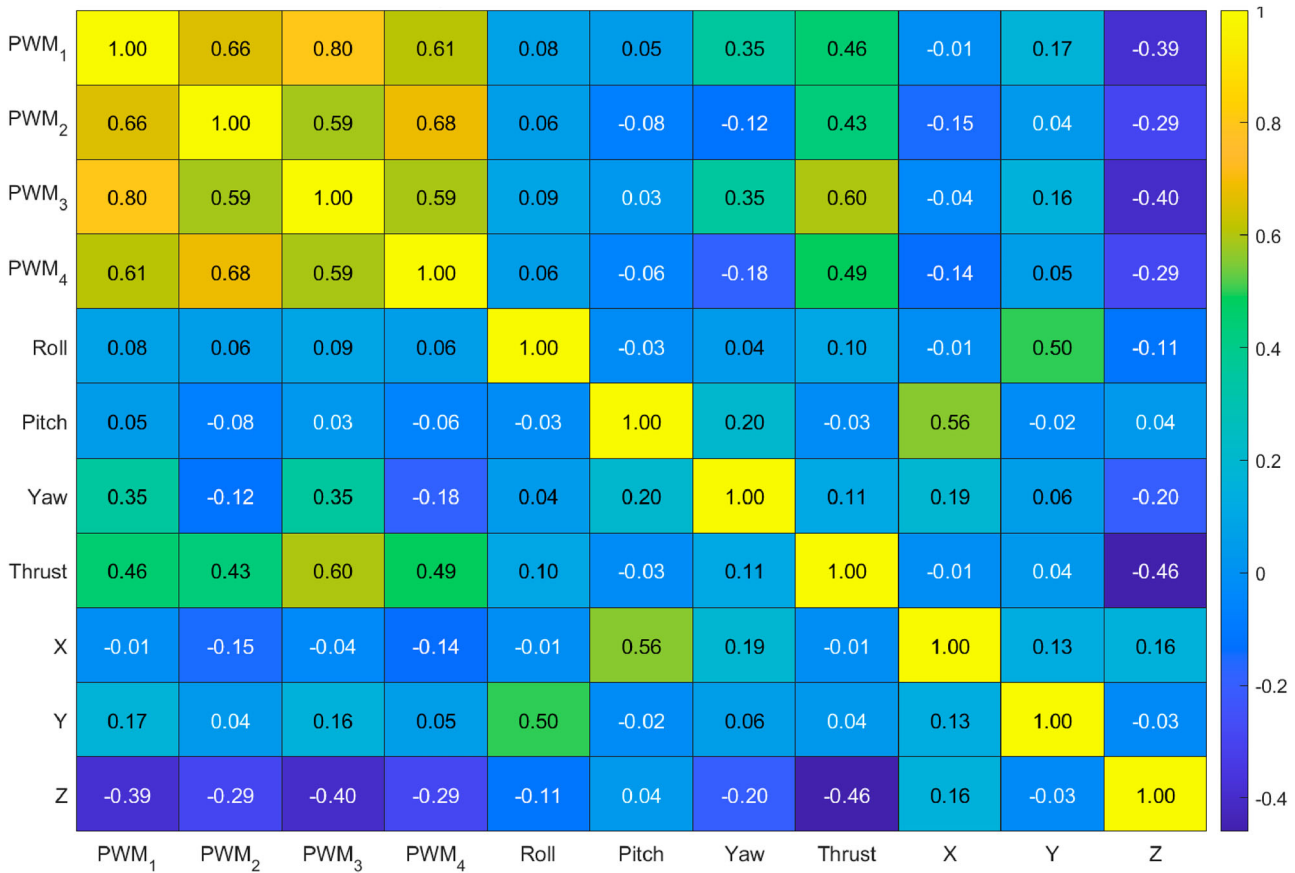


Figure 5. Spearman correlation coefficient of Small Random.

We also conducted the same data analysis with Spearman correlation for the more regular motions, which are *Going Left*, *Going Right* and *Going Forward*. Comparing Figures 5 and 6, the *PWMs*, *positions*, and *Stabilizer* performances during regular flights exhibited a similar correlation to those observed in the case of the *Small Random*.

The positive correlation coefficient between a drone's *Thrust* and *PWMs* is indicative of how *Thrust* in flight control is typically achieved by adjusting the output power of the motor. *PWMs* are a common method used to control the power of the motor. Generally, as *PWMs* increase (which means an increase in the pulse width of the motor's output), the motor's speed or output power also increases, resulting in greater *Thrust*.

The *Yaw* and two motors of the drone show a positive correlation, with the other two motors showing a negative correlation affirming the flight control system and power distribution of the quadcopter. In this type of drone, the control of flight direction and stability is achieved by adjusting the output of the four motors. *Yaw*, which involves rotation around the vertical axis of the drone, requires two pairs of symmetric motors to generate opposing torques. Consequently, changes in the speed of these two motors directly affect the direction

of the motor's *Yaw*, which results in a positive correlation. On the other hand, the other two motors are situated differently, and variations in their speeds produce torques in the opposite direction to the *Yaw*, leading to the observed negative correlation between them. This intricate correlation pattern emerges due to the dynamic balance and torque distribution inherent in this system.

3. Motion prediction with nonlinear time series models

Time series forecasting focuses on inferring future values by utilizing patterns and trends present in historical data. It involves analyzing the cyclic, trending, and other influential factors within the data to establish mathematical models for predicting future changes. In this process, correlation analysis provides valuable insights, serving as a guide for constructing time series models and aiding in the identification of variables that could substantially impact predictive outcomes.

Combining the findings from the earlier correlation analysis, we concentrated the drone motion prediction with two separate approaches: *LSTM* and *NARX* separately.

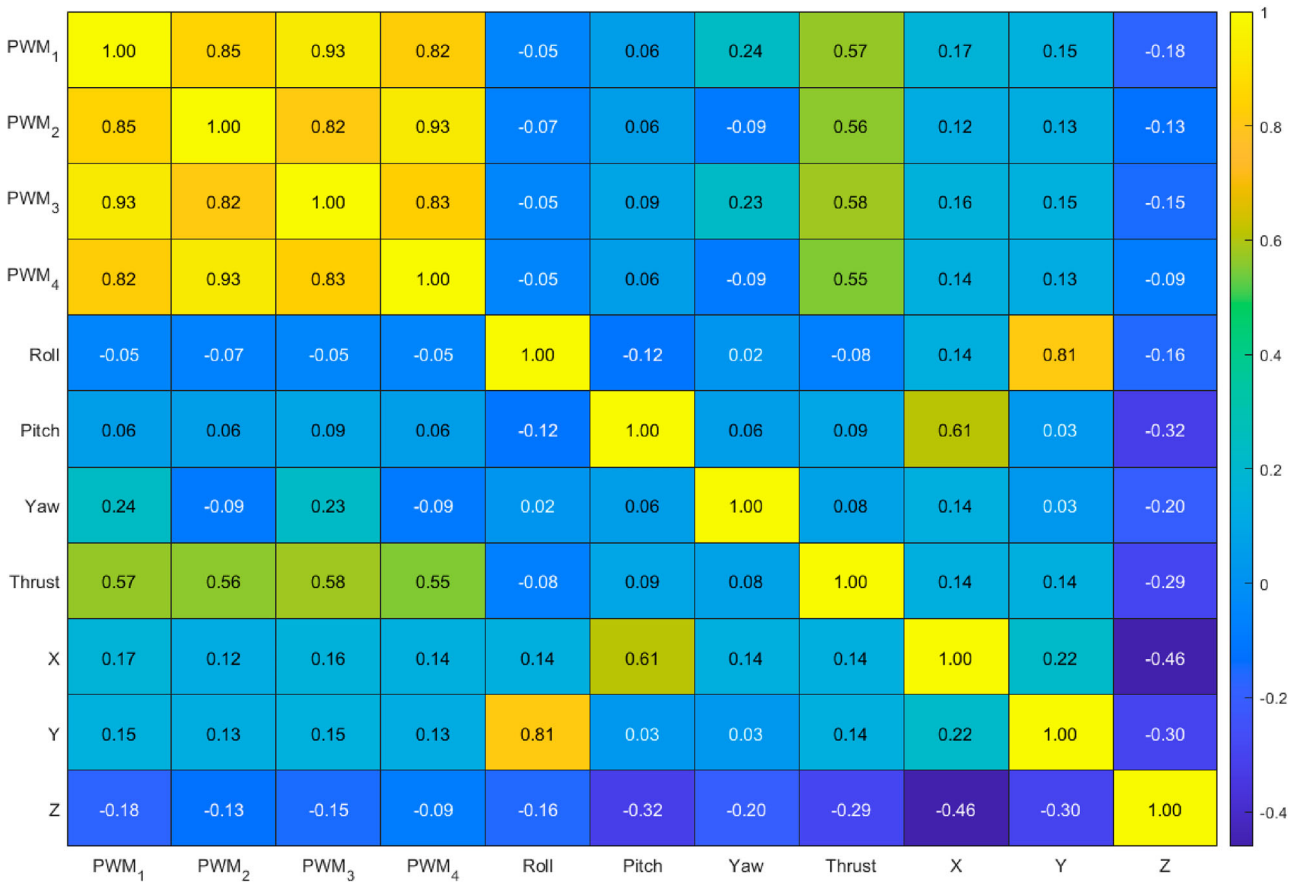


Figure 6. Spearman correlation coefficient of Going *LeftandRight*.

3.1. Motion prediction of LSTM models

Considering the characteristics of the data, firstly, we build our model based on the *LSTM* deep neural networks. *LSTM* is a kind of Recurrent Neural Network (RNN) that is suitable for processing sequential data such as speech, text, and time series (Greff et al., 2016; Hochreiter & Schmidhuber, 1997). Each unit of *LSTM* can continuously store and update information over multiple time steps, enabling solution of problems related to long-term and short-term dependencies. *LSTMs* are widely applied in fields such as natural language processing (NLP), speech recognition, time series forecasting, and action recognition. The application of *LSTM* is also found in autonomous driving by enabling the modeling and prediction of various aspects such as vehicle trajectories, behavioural patterns, and the surrounding environment. Through historical driving data, *LSTM* analyzes and learns from the vehicle's past movement patterns, enabling the anticipation of the vehicle's subsequent actions under specific circumstances. These actions encompass maneuvers like turning, lane changing, acceleration, or deceleration. By employing *LSTM* to learn from past experiences, autonomous vehicles are able to foresee and respond adeptly to diverse driving scenarios.

When examining the data associated with lateral movement (*Going Left*), the predicted results differ from the real ones, which can be easily noticed from Figure 7. While the model exhibits considerable precision in predicting movement along the *y*-direction, it displays a pronounced bias when forecasting the *z*-direction. Moreover, considering the diminutive scale of the experimental drones, the prediction accuracy is expected to be no more than 10 centimetres, which raises a challenge for the *LSTM*. The limitation may stem from *LSTM*'s inclination towards sequences with extended dependencies, rendering it susceptible to interference from noise.

To address this challenge, we explored the use of an *NARX* architecture as an alternative framework. *NARX* exhibits superior capabilities in predicting nonlinear time series, showcasing enhanced resilience against noise and outliers inherent in input data. Additionally, leveraging multiple time series as external inputs, *NARX* facilitates more adaptable modelling of diverse time series data.

In contrast to *LSTM*, *NARX* demonstrates accelerated computational efficiency when learning patterns from the drone flight data. This characteristic highlights the potential of *NARX* to overcome noise interference, leading to more accurate predictions.

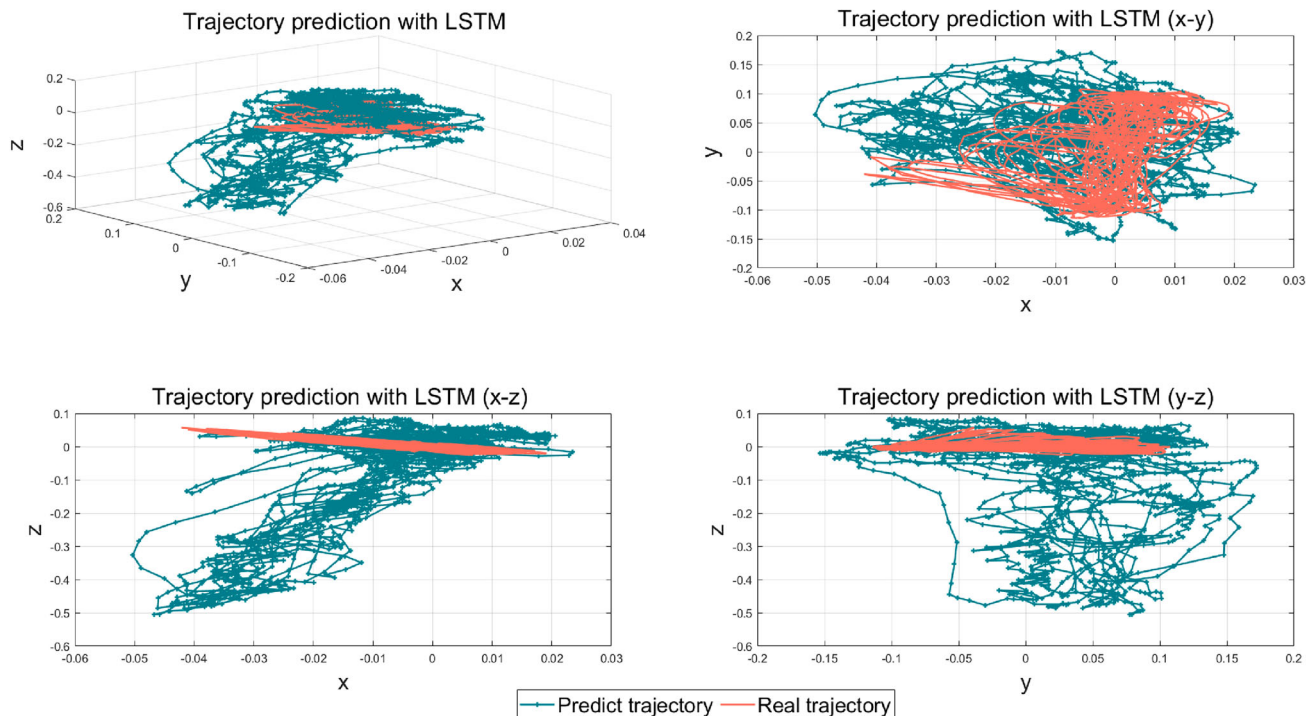


Figure 7. The LSTM model trained and tested on *Going Left*. The top left image displays the true trajectory and the predicted trajectory in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

3.2. Motion prediction of NARX models

The NARX model represents a hybrid architecture combining auto-regression and exogenous input components. It integrates the concepts of autoregression (AR) and external inputs (X), allowing for the effective handling of non-linear relationships and prediction issues within dynamic systems. The specific model structure is shown in Figure 8. This schematic illustrates the NARX model designed for motion prediction, structured in three layers. The model processes two input sequences: the stabilizers sequence and the initial positions sequence. The initial positions sequence undergoes preprocessing, including detrend and normalize operations, to remove linear trends and scale the data to a standard range. A crucial component of this model is the feedback loop, which connects the output back to the input, particularly to the feedback delay. This loop enables the model to use previous outputs to inform and improve future predictions.

The NARX model can be written as:

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-m), x(t-1), x(t-2), \dots, x(t-n)),$$

where, m - feedback delays; n - input delays.

The distinctive feature of NARX models is their ability to capture complex nonlinear patterns within time series data. Unlike traditional linear models, NARX can accommodate external factors, enabling more accurate and

adaptable predictions. Its recurrent structure facilitates the modeling of time dependencies, making it well-suited for scenarios involving dynamic and interrelated data. By incorporating exogenous inputs, NARX can better adapt to diverse, real-world scenarios, making it a compelling choice for predictive analytics and forecasting tasks. With the growing demand for predictive models, NARX has been increasingly utilized across various domains, including weather forecasting (Bukhari et al., 2020), energy demand prediction (Ruiz et al., 2016), and many other areas (Raptodimos & Lazakis, 2020; Wunsch et al., 2021).

As a new approach, we proceeded to model the *Going Left* data using NARX methodology. The model takes a sequence of positions (X , Y , Z) as inputs and recurrent inputs which are 11-dimensional. After training, we compared the predicted trajectory as well as the predicted *Stabiliser* data (*Roll*, *Pitch*, *Yaw*, and *Thrust*), against their respective true values. Figure 9 shows the true and predicted trajectories across various dimension combinations.

When contrasting the trajectories predicted by the LSTM model, as depicted in Figure 7, with those derived from the NARX model, a markedly superior level of accuracy is evident in the latter. The detailed comparison of results is delineated in Table 2, illustrating the substantial improvement achieved by the NARX model over LSTM predictions. Additionally, Table 3 presents the goodness-of-fit (R^2) for individual variables, demonstrating the

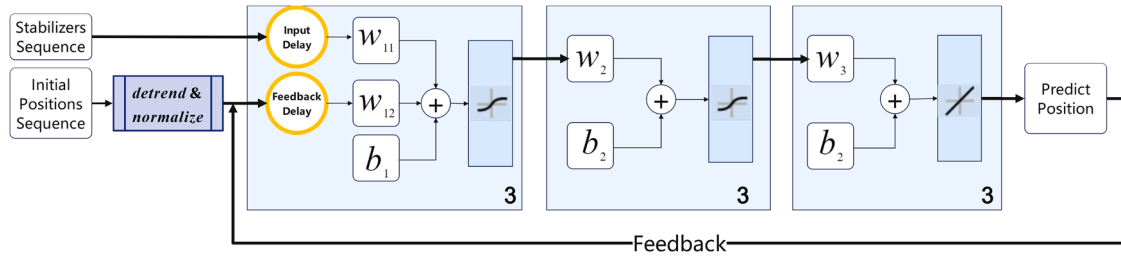


Figure 8. The schematic of the NARX structure. The model shown comprises three layers, the marked numbers '3' in each block stand for the size of the hidden layers in each block. Before being fed into the model, *detrend* and *normalize* operations are applied to the position inputs.

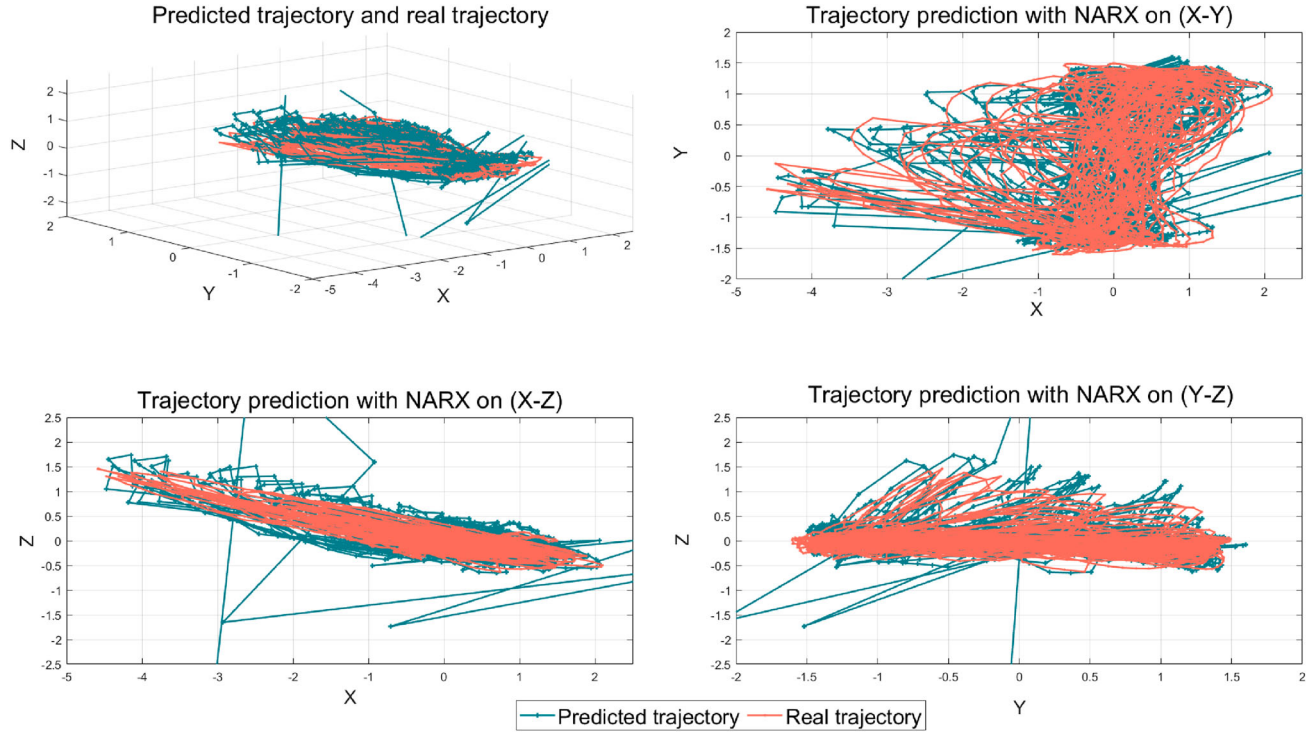


Figure 9. The NARX trained and tested on *Going Left*. The top left image displays the true trajectory and the predicted trajectory in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

Table 2. Model validation (MSE and R^2) of LSTM and NARX on *going left*.

Model name	Validation topic	Mean square error (MSE)	General R^2
Going Left LSTM	POS	0.094	$-1.28 \times e^{-5}$
Going Left NARX	POS	0.073	0.58
	PWM	0.435	-
	Stabilizer	0.142	-

overarching trend of superior predictive performance exhibited by the NARX model across various metrics.

Moreover, the NARX model shows a reasonable accuracy rate, reaching 70% in predicting path errors within a margin of 0.15 meters, while achieving an even higher accuracy rate of 90% for errors confined within 0.25 meters. These findings underline the heightened

precision and reliability inherent in the NARX model's predictions, solidifying its efficacy in forecasting trajectories with enhanced accuracy.

The comparative analysis between predicted and actual positions on individual dimensions is shown in Figure 10. After visualizing the predicted *Stabilizer* and the true stabilizers, as shown in Figure 11, it becomes apparent that the NARX model adeptly accommodates the variations in *Roll*, *Pitch*, and *Thrust*. However, concerning the comparatively steady data attribute, *Yaw*, the predicted values exhibit notable fluctuations.

To comprehensively assessing the model's performance, we examined the auto-correlation of the error. We found that the auto-correlation of the error of the model slightly deviates from the delta function, indicating that the model struggles to capture the underlying trends or

Table 3. Model validation (R^2 of different variables) using LSTM and NARX on *going left*.

Model name	Validation variables' name	R^2 of all predicted data	R^2 of the predicted data (from the 200th sample)
Going Left LSTM	X Direction	-0.37	-0.38
	Y Direction	-1.2	-1.13
	Z Direction	-1.5	-1.52
Going Left NARX	X Direction	0.76	0.92
	Y Direction	0.98	0.99
	Z Direction	-3.89	0.71
	Roll	0.84	0.87
	Pitch	0.75	0.89
	Yaw	-12.26	-0.85
	Thrust	-3.05	0.54

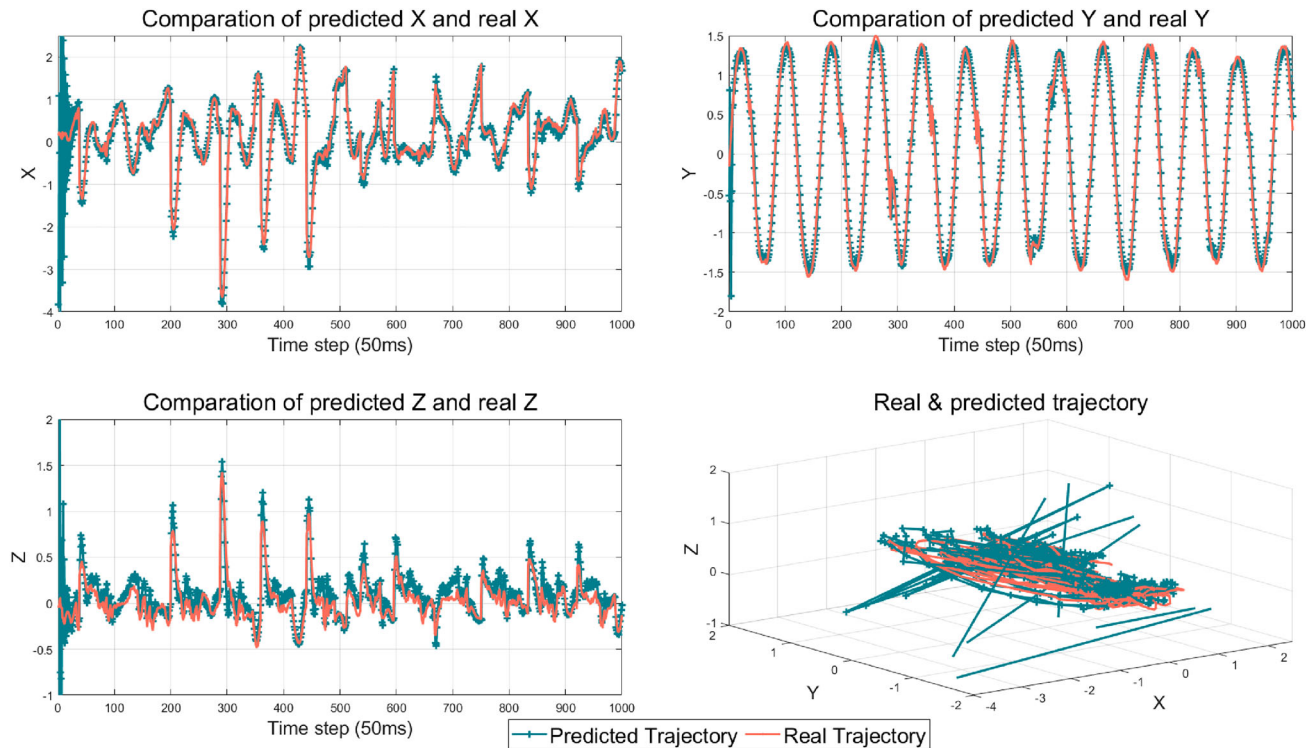


Figure 10. The comparison between the predicted and actual trajectory on a single dimension generated by the NARX applied on *Going Left* is shown. The true trajectory and the predicted trajectory are shown in the 3D image on the bottom right. The remaining three images show the true and predicted trajectory on separate dimensions.

seasonal factors of the data. This discrepancy could be indicative of under-fitting. Typically, improving the complexity, expanding the dataset, or reducing the number of features can alleviate this issue. Then we increased the size of the training set and observed a persistent auto-correlation in the error, although the correlation has slightly decreased.

This observation highlights the complexity of the underlying dynamics and indicates the need for further refinement in the model architecture or the inclusion of additional relevant features. While the model demonstrates competence in capturing certain variations, the persistence of auto-correlated errors implies an incomplete grasp of the intricate patterns governing the system's behaviour, particularly in relation

to yaw stability. Thus, subsequent iterations involving an expanded dataset, meticulous feature selection, or adjustments in the model's complexity are essential to address the persistent under-fitting issue and improve the accuracy of NARX model's prediction over all dimensions.

Comparing the experimental results, it can be seen that digitizing the flight of the drones with the NARX model generated relatively better results than that of LSTM. However, upon visualizing the output and analyzing the autocorrelation of the errors, we found that the model is under-fitting. Therefore, we further modify the current NARX model. The purpose of this model is data-driven learning of the flight pattern of the drone, which will provide us with a more stable and precise method of

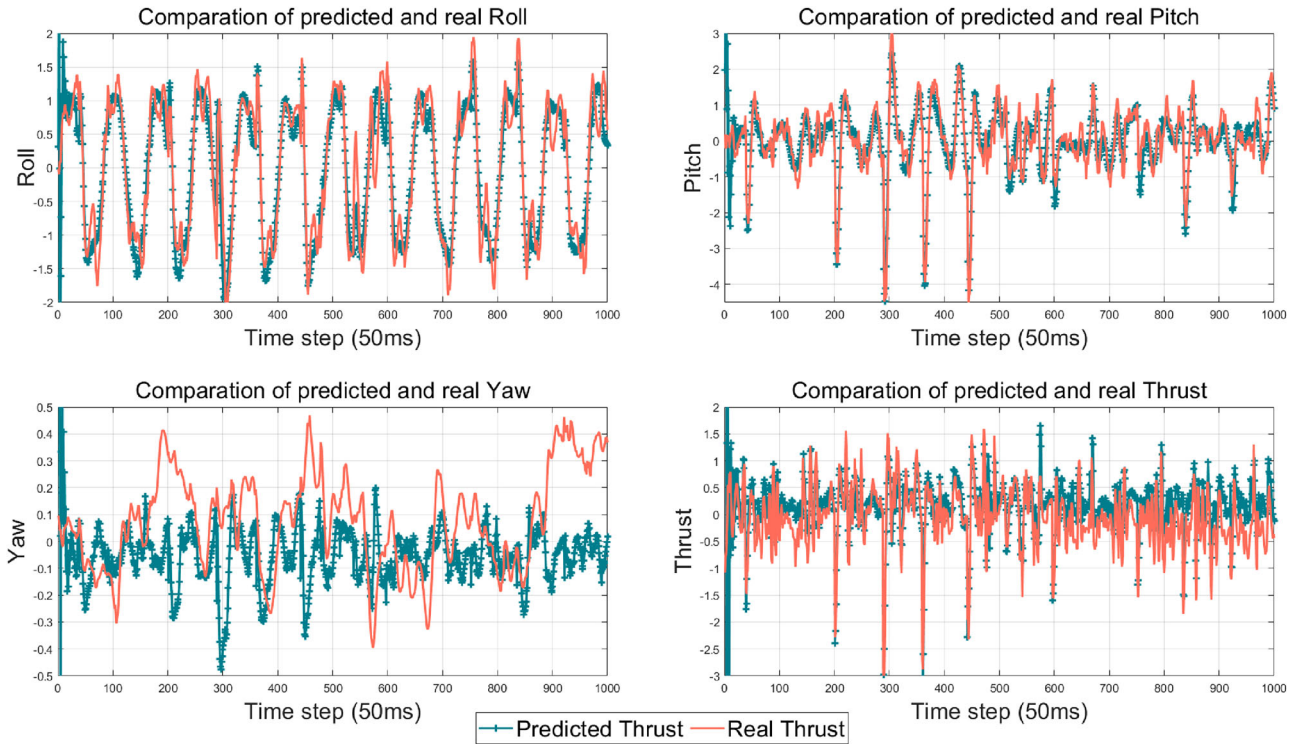


Figure 11. The comparison between predicted and actual *Stabilizers* generated by the *NARX* applied on *Going Left*.

automatic control of the drone. We refer to this model as a *Virtual Drone*.

4. The refinement of the *NARX* model structure

Considering the further application of this model for obstacle avoidance in unknown environmental conditions and its utilization for cooperative obstacle avoidance among multiple *Virtual Drones*, it becomes imperative to refine the model's structure. We opted to reconfigure the input and output structure of the virtual drone as an initial attempt to improve the fit of the model. This modification is aimed to better capture the encountered environmental information during the drone's flight.

After considering the drone's operational theory, our hypothesis centered on the notion that the motor operations induce position alterations. Therefore, we aimed to determine the feasibility of using *PWMs* or *Stabilizers* as external inputs for the model. Simultaneously, we anticipated the model's capability to forecast the drone's position. To evaluate these modifications, we conducted training and testing sessions using the restructured second-generation virtual drone model within a simulated environment referred to as the '*Small Random walk*'. Batch training is conducted separately on the model architectures utilizing *Stabilizers* (*Sta_POS*) and *PWMs* (*PWM_POS*) as inputs within the *Small Random*. This approach aimed to identify which attributes

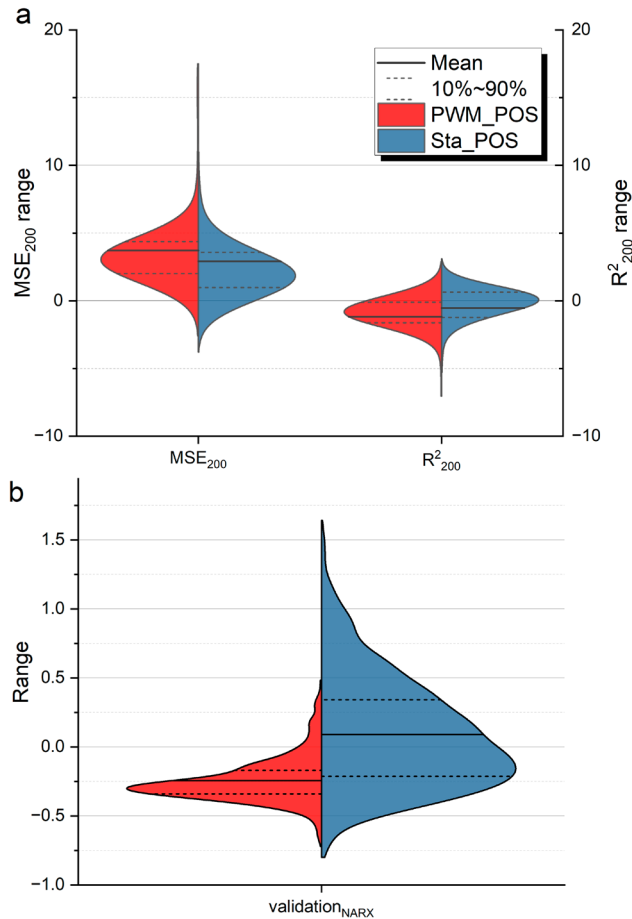
contribute most significantly to crafting a more high-performing model. The foundational structure of the model, characterized by the number of nodes in each hidden layer, was initially configured as [3, 3, 3].

We conducted training sessions involving varying numbers of input delays and feedback delays, ranging from 4 to 15 steps which are 720 models in total. The performance metrics, including Mean Squared Error (MSE) and coefficient of determination (R^2) scores were evaluated on the entire dataset as well as specifically the initial 200 steps, serving as validation metrics. Based on predictive visualization and the model's MSE and R^2 , we observed that for the *NARX* model, a single R^2 or MSE value alone does not adequately reflect the model's predictive capability. Even if the model gives a high R^2 but if the corresponding MSE is high, it does not necessarily imply accurate predictions by the model. Therefore, for *NARX* model validation, we predominantly utilized $validation_{NARX} = R^2/MSE$ as a validation metric. Higher $validation_{NARX}$ values corresponded to better predictive performance. Figure 12 compares the performance of *PWM_POS* and *Sta_POS* *NARX* based model in three validation methods intuitively. From the figure, it is evident that regardless of the methods used for the model validation, the *Sta_POS* model consistently demonstrates a higher degree of fit compared to the *PWM_POS* model.

Considering the performance comparison, we further delved into the internal structure of *Sta_POS*. As the

Table 4. Model validation (MSE and R^2) of general position of the best four model in batch training on *Small Random*.

Model	Input delays	Feedback delays	MSE general	R^2 general	MSE the first 200 steps	R^2 first 200 steps
Model 89	5	9	3.24	-2.2	0.73	0.59
Model 207	8	15	2.74	-1.77	0.70	0.75
Model 407	10	13	2.77	-1.78	0.55	0.80
Model 642	14	12	3.50	-2.53	0.68	0.71

**Figure 12.** The figure illustrates the comparison of performance between the *PWM_POS* model and the *Sta_POS* model. (a) displays the *MSE* and R^2 scores for the predictions with each model over the initial 200 steps, while (b) shows the comparison of the *validation_{NARX}*.

predicted position and the *MSE* and R^2 values of the models being visualized, we identified the top four performing models. The structures of the *Sta_POS* models and their respective validation results are presented in Table 4. The error chart of these four models across various dimensions is individually shown in Figure 13. It can be observed that *Model 407* exhibits superior regression performance on the data, particularly noticeable in the *y*-direction. Compared to other models, *Model 407* shows a notably more stable prediction trajectory. Additionally, when visualizing the predictive paths, we observed when confronted with abrupt changes in the data, *Model 407*

demonstrates a remarkable ability to predict this trend accurately. Upon examining the fitting of these four models, it becomes apparent that compared to actual flight data, the predictions given by the *NARX* model provide a smoother curve.

Predictions were made for 720 models across different datasets. We further explored the accuracy and fitting of predictions for these 720 models at different step lengths and conducted additional experiments. The 720 models were tested for variance step predictions, which ranged from 2 steps to 200 steps on multiple trajectories such as *Small Random*, *Going Left*, *Going Right*, and *Taking Off*.

After testing the models on *Small Random*, the four models (378, 176, 407, and 642) demonstrated the best *validation_{NARX}* are further visualized. *Model 378* generated the best *validation_{NARX}* among the 720 models. To gain an intuitive understanding of the trajectory prediction, we visualized the predicted results separately for the *x*-direction, *y*-direction, and *z*-direction (Figure 14). Additionally, we presented the visualization of the predicted results in Figure 15 across combined directions. From Figure 14, it is evident that the model provides more accurate predictions in the *y* and *z* directions, with a notable improvement, particularly in the *y* direction. This highlights the model's ability to capture the motion trend along the *y*-axis. From Figure 14, we can observe that *Model 378* exhibits a high degree of fitting for the first 40 steps of the predicted trajectory, as compared to the real trajectory. Analyzing the data, we found that R^2 reaches 99%, and *validation_{NARX}* achieves 99 (Table 5).

The visualizations of predictions for the remaining three optimal models under the *Small Random* trajectory, namely models 407, 176, and 642, are illustrated in Figures 16, 17, 18, 19, 20, and 21. The predictions from the four models are notably more accurate in the *y*-direction. When considering a direct prediction for 200 steps, we find that *Model 407*, emerges as optimal for the 200-step prediction in the previous section, remaining the best among models 378, 407, 176, and 642. However, in the case of predicting 120 steps in the *z*-direction, the prediction error of *Model 407* has increased by almost 50% compared to *Model 378*. Even though *Model 642* demonstrates a trajectory prediction trend very similar to the real path, the overall prediction error is significantly high. In the *x*-direction, the predicted range extends from its

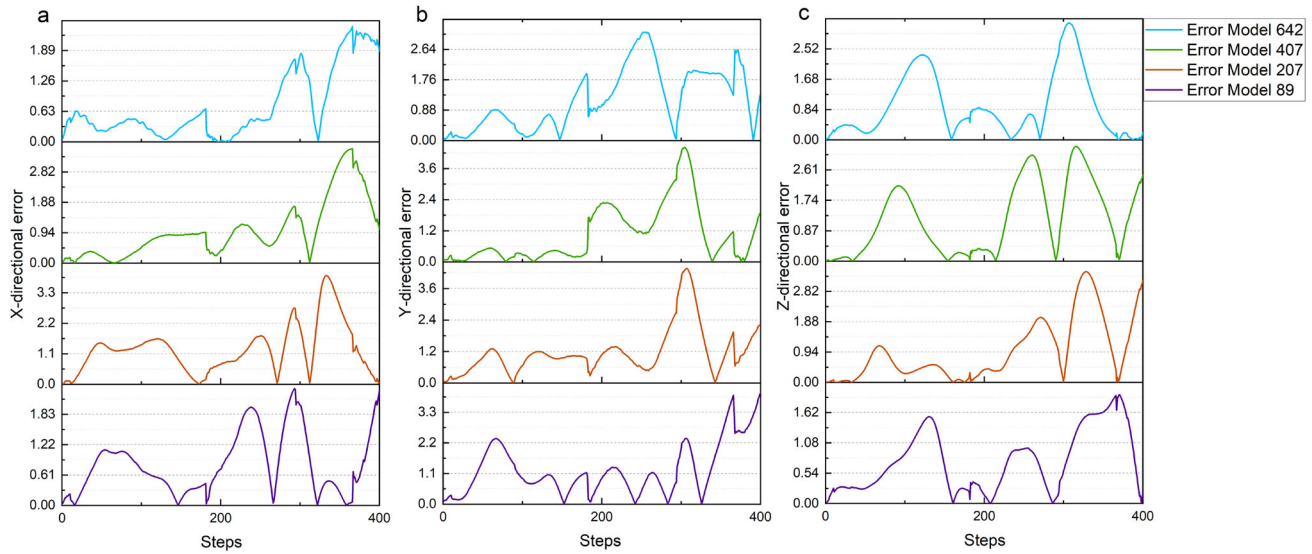


Figure 13. The error chart of the best four performing NARX models on *Small Random walk*. The error chart of each model on every dimension is shown in the subplots separately.

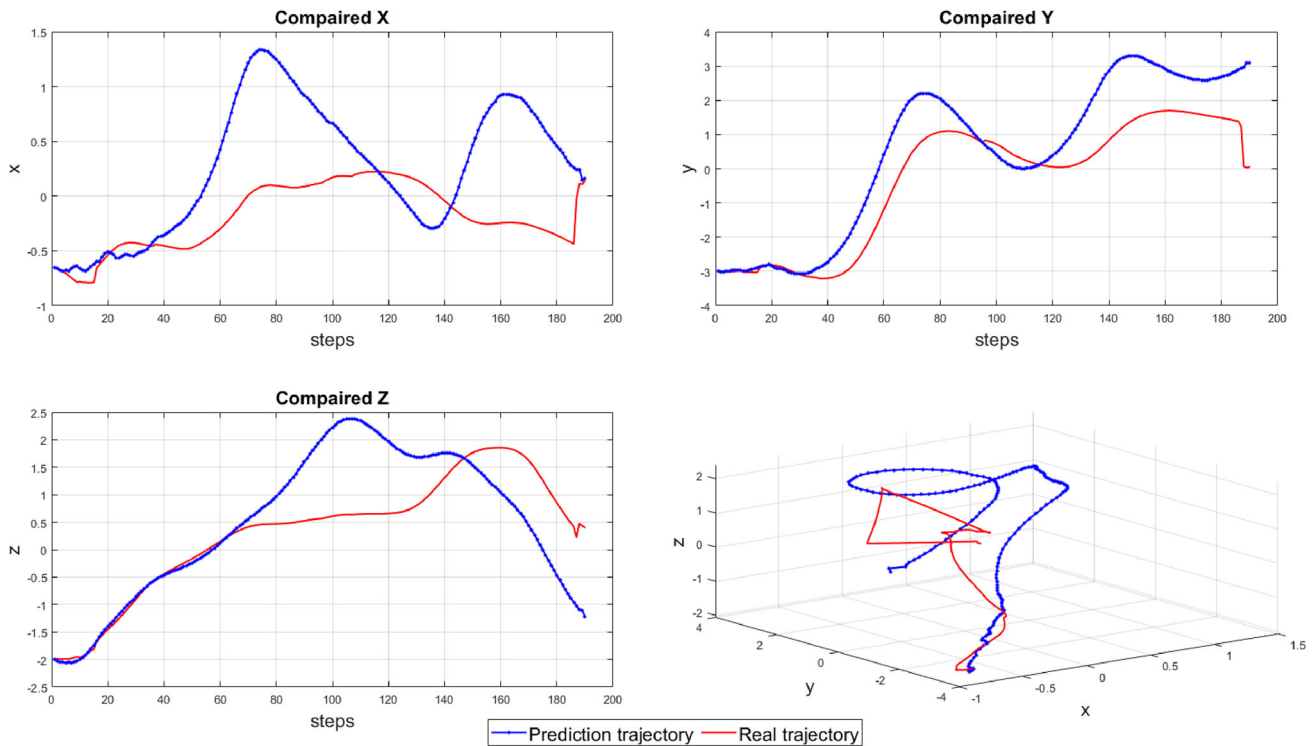


Figure 14. This figure shows the comparison between predicted and real trajectory on separate directions generated by the *Model 378* applied on *Small Random*. The bottom right image displays the real trajectory and the predicted trajectory in 3D. The remaining three images show the real and predicted trajectory on separate dimensions.

Table 5. Model validation (MSE and R^2) when testing on *Small Random*.

Model	Input delays	Feedback delays	MSE	R^2	validation _{NARX}	Predicted steps for comparison
Model 378	10	7	0.015	0.99	99	40
Model 407	10	13	0.02	0.97	48.5	40
Model 176	6	15	0.02	0.98	49	40
Model 642	14	12	0.11	0.76	6.91	40

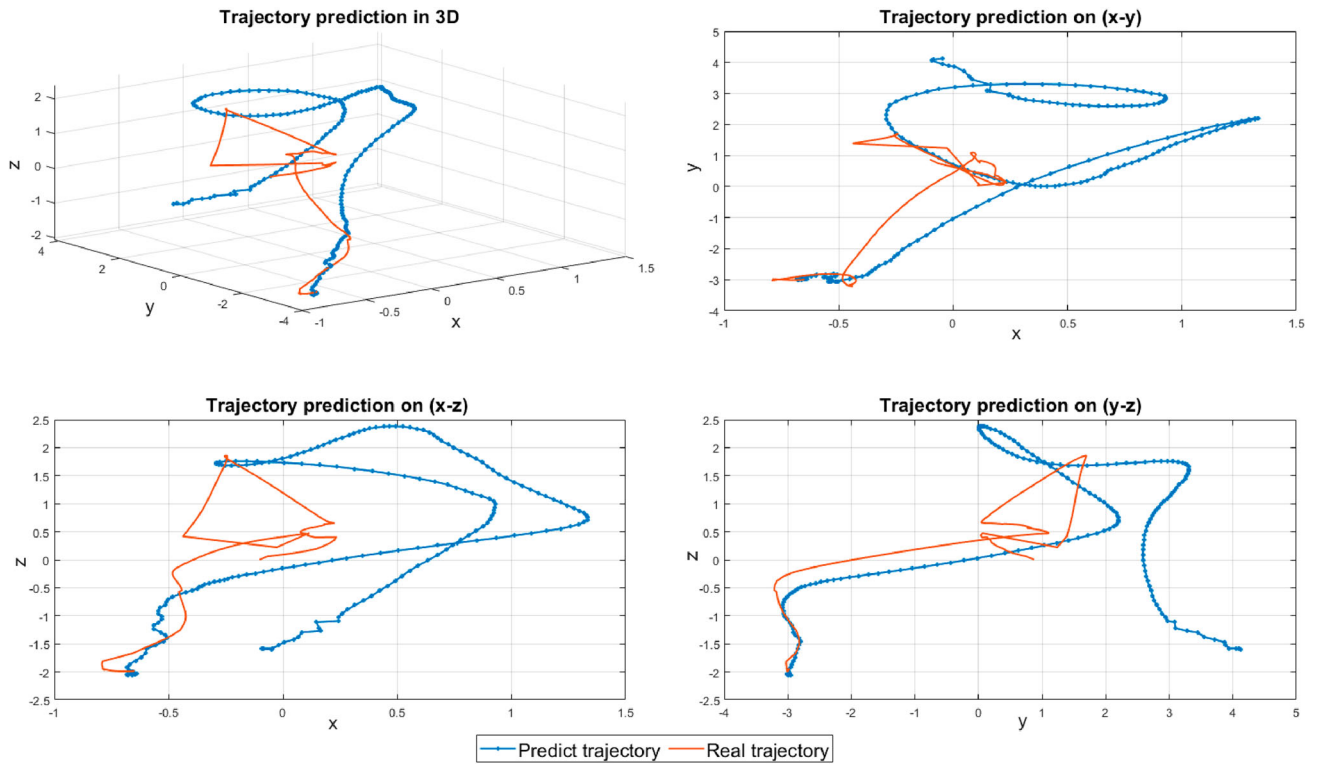


Figure 15. Model 378 tested on Small Random. The upper left image displays the true trajectory and the predicted trajectory in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

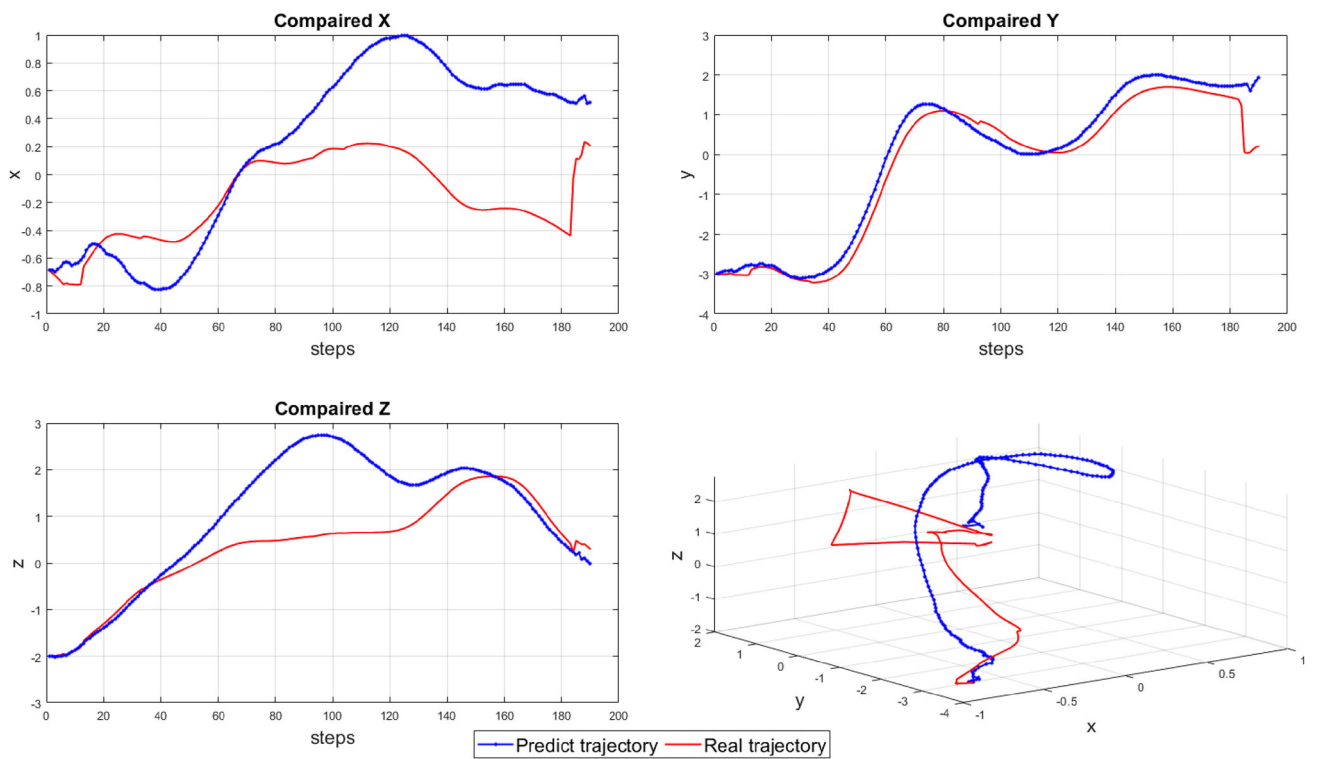


Figure 16. This figure shows the comparison between predicted and real trajectory on separate directions generated by the Model 407 applied on Small Random. The bottom right image displays the real trajectory and the predicted trajectory in 3D. The remaining three images show the real and predicted trajectory on separate dimensions.

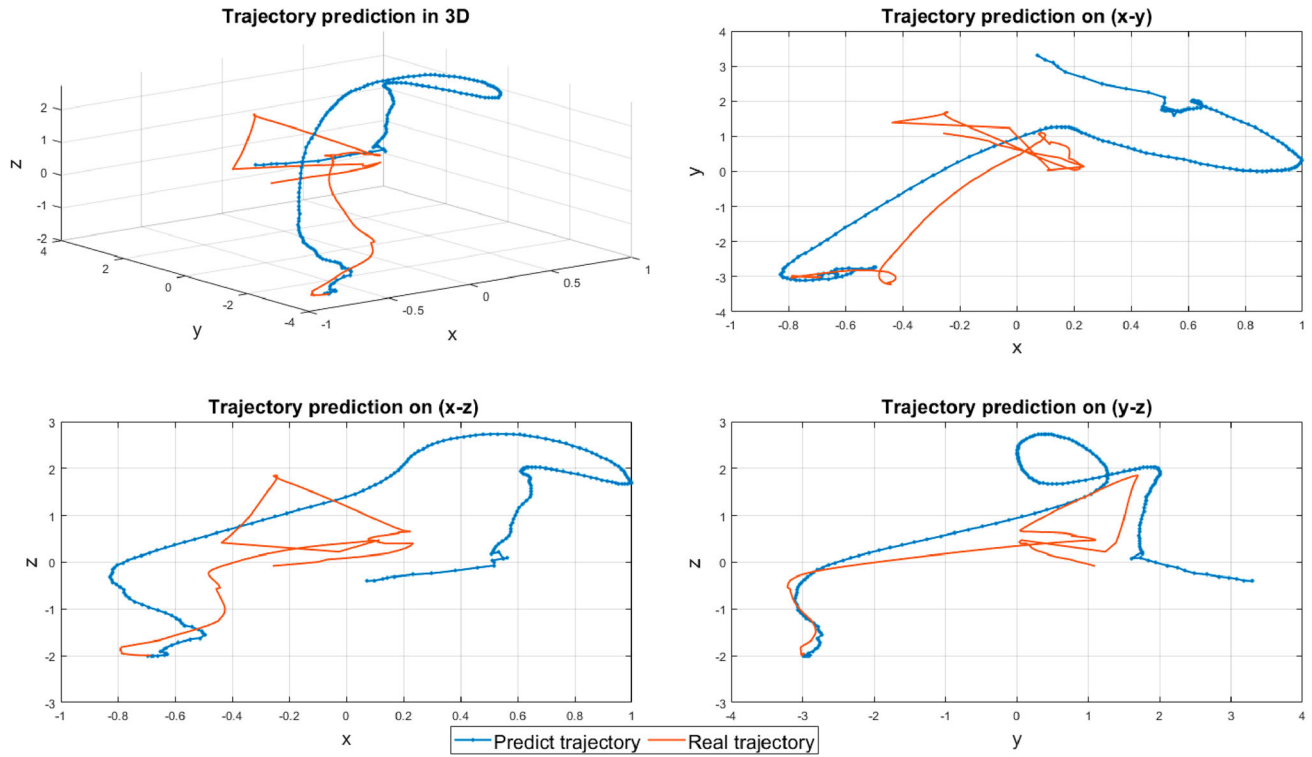


Figure 17. Model 407 tested on Small Random. The upper left image displays the true trajectory and the predicted trajectory in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

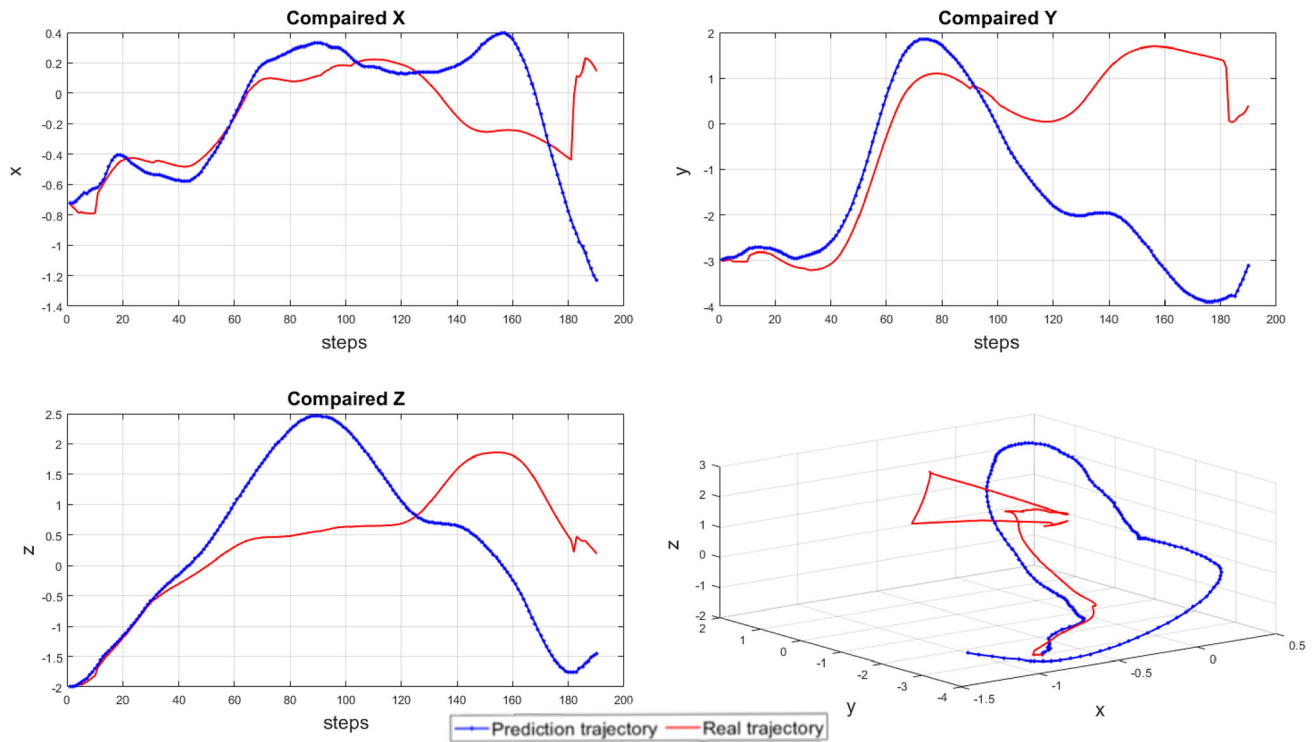


Figure 18. This figure shows the comparison between predicted and real trajectory on separate directions generated by the Model 176 applied on Small Random. The bottom right image displays the real trajectory and the predicted trajectory in 3D. The remaining three images show the real and predicted trajectory on separate dimensions.

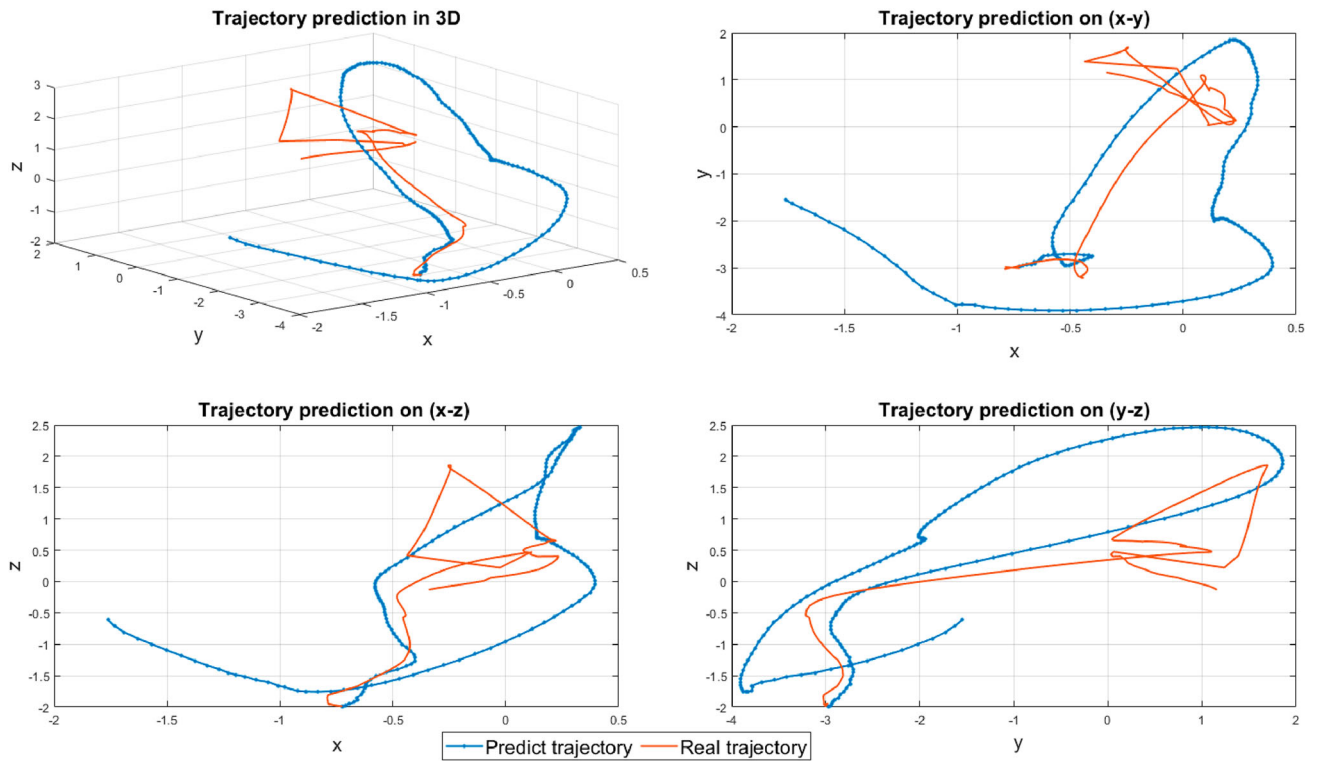


Figure 19. Model 176 tested on Small Random. The upper left image displays the true trajectory and the predicted trajectory in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

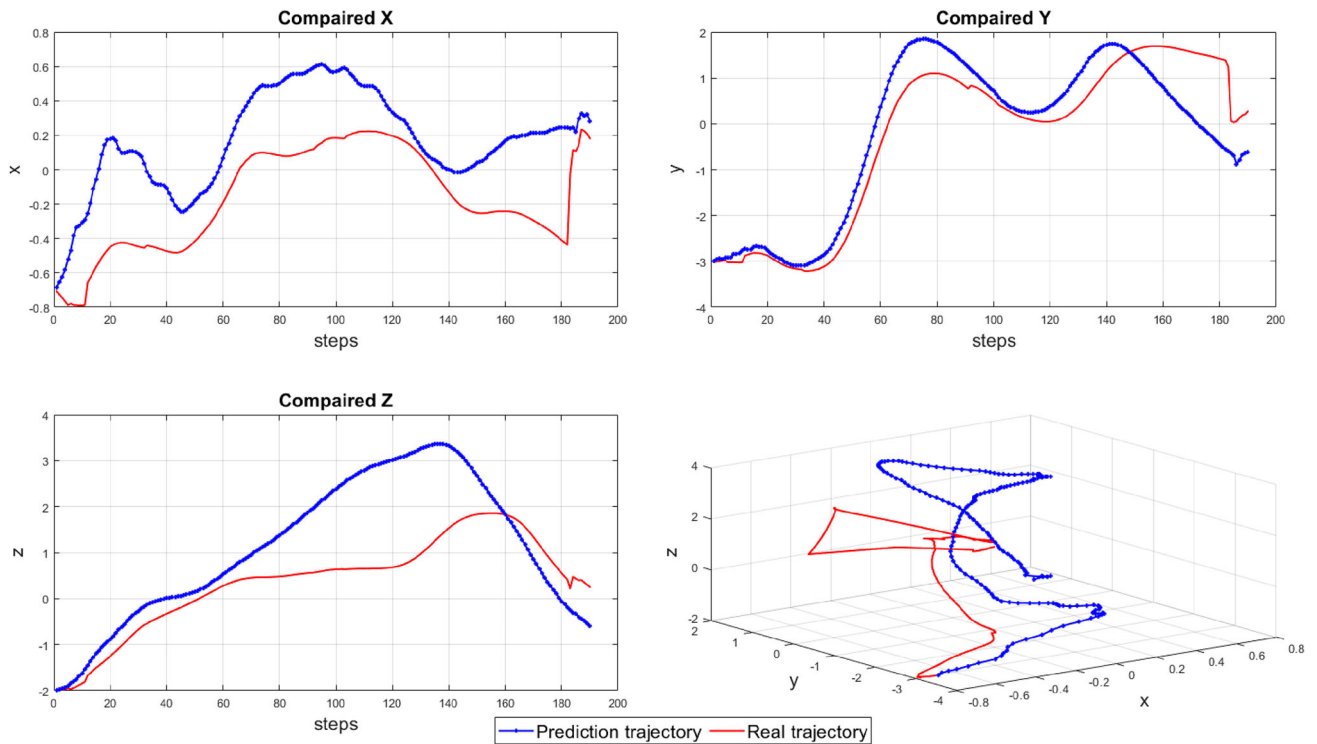


Figure 20. This figure shows the comparison between predicted and real trajectory on separate directions generated by the Model 642 applied on Small Random. The bottom right image displays the real trajectory and the predicted trajectory in 3D. The remaining three images show the real and predicted trajectory on separate dimensions.

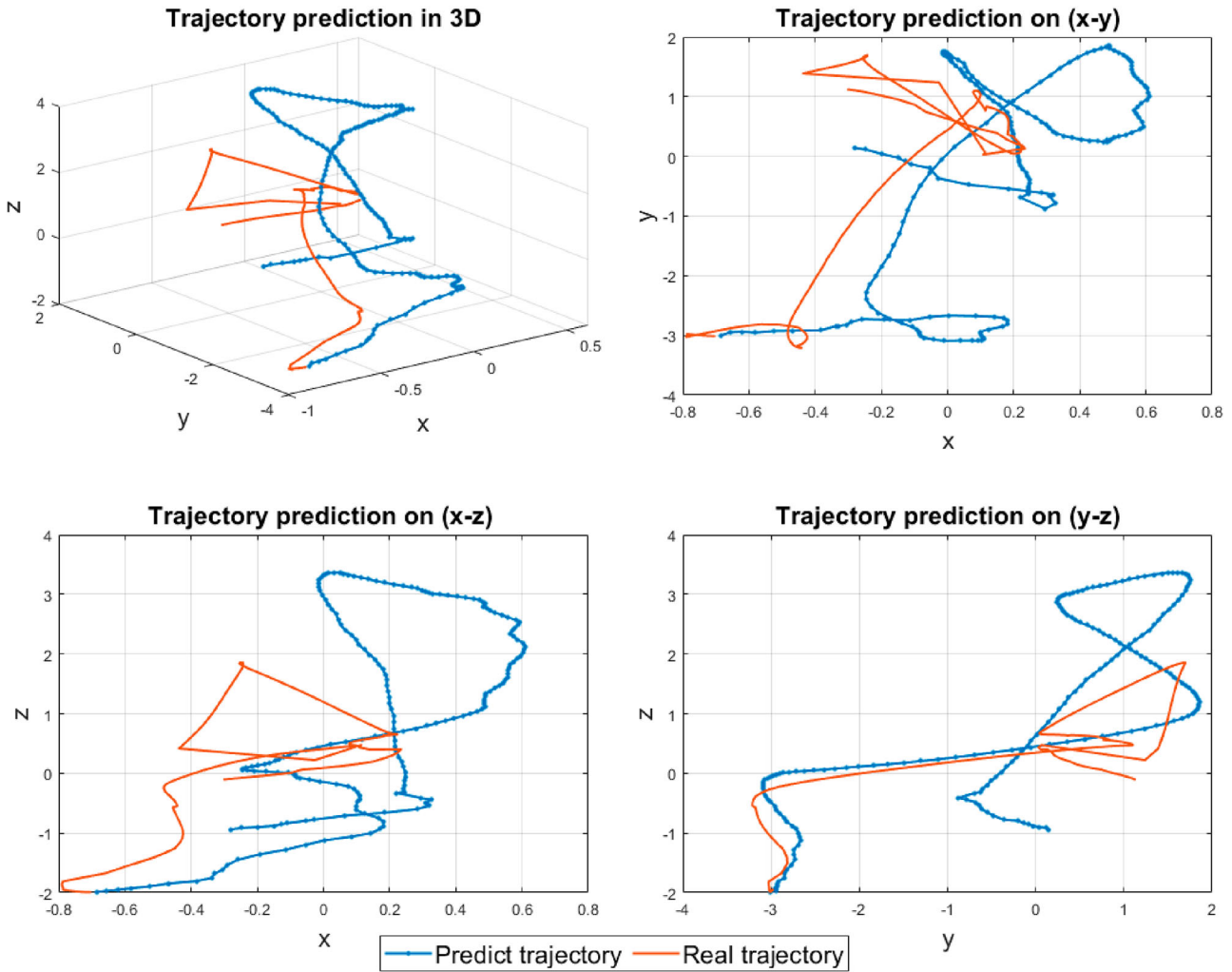


Figure 21. Model 642 tested on Small Random. The upper left image displays the true trajectory and the predicted trajectory in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

original 0.2m to 0.6m, while, the prediction of z-direction showed a larger error when doing more than 100 steps of prediction.

We applied the model to various flight paths, including *Going Left*, *Going Right*, *Going Forward*, and *Taking Off*. Simultaneously, we validated the prediction results for different step counts to assess the model's predictive capabilities across diverse flight trajectories. The following section presents a comparison between the predicted paths and the real paths for the models that demonstrated superior performance on distinct datasets. Model 207 and Model 176 emerged as the top-performing prediction models for the *Going Left* and *Going Right* flight trajectories, respectively.

There was significant fluctuation in the early predictions, stabilizing gradually after forecasting 20 steps. Upon comparing the predicted values in the three directions, we observe that these models exhibit greater accuracy in predicting the x and z directions while

encountering larger errors in the y direction. This phenomenon contrasts sharply with the substantial-high regression observed in the y-direction predictions during the Small Random, indicating potential overfitting issues in the y-direction for the model ensemble.

Regarding predictions for different underlying paths in the environment besides *Going Left* and *Going Right*, the specific optimal models and their corresponding structures, MSE (Mean Squared Error), R^2 values, and $validation_{NARX}$ are detailed in the Table 6. It is notable that predictions for *Going Right* and *Going Left* are not as accurate as those for other simpler trajectories. This discrepancy arises from the inherent challenge of maintaining the drone's stability in the x and z directions during actual flight conditions. To sustain lateral motion only, the drone's motors require additional outputs. Despite detrending the data in the initial stages, fluctuations generated during flight still exert a non-negligible impact on the model predictions.

Table 6. Model validation (MSE and R^2) when testing on *Going Left*, *Going Right*, *Going Forward* and *Taking Off*.

Dataset	Model	Input delays	Feedback delays	MSE	R^2	$validation_{NARX}$	Best Predicted steps
Going Left	Model 207	7	9	4.25	0.45	0.12	30
Gong Right	Model 176	6	15	80	–	–	–
Going Forward	Model 594	13	14	13.315	0.52	0.04	62
Taking Off	Model 338	9	11	15	0.67	0.04	42

These findings provide promising strides toward refining the virtual drone model for enhanced predictive accuracy and stability, crucial for its deployment in real-world scenarios involving autonomous flight control and obstacle avoidance. Further iterations and refinements can be pursued to optimize model performance across diverse environmental and operational conditions.

Based on our findings, each pathway presents its distinctive applicable model. When we endeavour to apply Model 378, which exhibited optimal performance on the *Small Random* pathway, to various scenarios such as *Going left*, *Going Right*, *Going Forward*, and *Taking Off*, we observe that the Model 378 adequately predicts data trends in the *Taking Off* experiments. However, its predictive performance on the other three pathways falls short of expectations. Similarly, the optimal models for *Going left*, *Going Right*, *Going Forward*, and *Taking Off* also exhibit analogous behaviours when applied to the remaining datasets. Considering these observations, we are confronted with the question of model selection for the virtual drone to be incorporated into our subsequent closed-loop control system research. However, considering the complexity of the pathways and the adaptability to the environment, we will initially employ Model 378 as the embedded virtual drone in the closed-loop control system.

In the upcoming research, we intend to leverage trained drone models extensively to establish a closed-loop control system for the autonomous control of aircraft motion paths. The traditional closed-loop control system, also known as the feedback control system, is a common system design in the engineering and control domains, characterized by the integration of a feedback loop within the system. This feedback loop allows the system to monitor its output in real time and make adjustments based on this output information, ensuring that the system's behaviour aligns more closely with or meets the expected performance requirements. By integrating virtual drones and feedback loops, our objective is to construct a control system for aircraft flight paths, enabling the aircraft to closely adhere to the desired trajectory, enhance overall performance, and reduce resource consumption associated with extensive repetitive flights.

This research direction not only holds the potential to advance drone technology but also carries significant theoretical and practical implications for exploring

the application of closed-loop control systems in complex environments. Our innovative approach aims to fully exploit the intelligence and adaptability acquired by drones during model training, allowing the closed-loop control system to flexibly adapt to changes and uncertainties in real-world environments.

5. Conclusion

In this study, we designed a novel method for predicting the position of drones by directly forecasting the next flight location based on experimental data such as *PWMs* and *Pose*. This approach circumvents the complex computations and nested control loops typically found in traditional flight control systems (Ogata, 2009). To achieve this, we devised a dynamic time series model, termed the *Sta_POS NARX* model, based on the temporal nature of the data. Subsequently, we conducted a comparative analysis with the *LSTM* model which is widely utilized in self-driving vehicle technology for position prediction.

Our findings demonstrate that, when applied to the drone dataset, the *NARX* model exhibited faster training speeds and relatively better fitting, as compared to the *LSTM* model. Specifically, the *LSTM* model often required longer training times and performed comparatively poorly in terms of regression model fitting. These results hold significant implications for UAV control and position prediction. In particular, the rapid training pace of the *NARX* model contributes to a more efficient real-time response capability within drone systems. This is crucial for real-time decision-making and flight path adjusting, particularly in situations requiring rapid adaptations to dynamic environments and task requirements. Moreover, the relatively higher degree of fitting of the *NARX* model indicates that it is better suited to the temporal characteristics present in data from drones. This improved ability to capture temporal patterns contributes to improved prediction accuracy, thereby reinforcing the stability and safety of the drones' automatic control systems. However, it is essential to note that although we observed the advantages of the *NARX* model over the *LSTM* model, this does not imply that the *LSTM* model is inapplicable in all scenarios. The *LSTM* model may demonstrate distinct advantages in particular contexts and possesses the capability to manage more intricate temporal data structures or data that is

comparatively stable or exhibits regular patterns. Following further refinement in the model's structure, we discovered that the *Sta_POS* model yielded comparatively superior predictions. Furthermore, through batch training, we identified the most optimal configuration for the *Sta_POS NARX* model in the closed-loop scenario. The detailed weight and structure of the optimal models on multiple simple trajectories are attached to the paper.

In conclusion, our study provides a fresh perspective on UAV position prediction and underscores the benefits of the data-driven *NARX* model in terms of speed and fitting accuracy. This provides valuable insights and directions for optimizing and advancing drone navigation systems. Nevertheless, further meticulous research and practical validation remain necessary for precise model selection in specific scenarios.

6. Future research plans

Due to complex goals as well as the uncertainty of the environment, the cooperative control of swarms of drones is one of the enduring hot topics in UAV research. Research on UAV cooperative control initially focused on fundamental concepts and theoretical aspects (Yu et al., 2019).

Currently, the most prevalent controlling methods for UAVs involve either centralized control systems or distributed control systems (Cajo et al., 2019; Kada et al., 2020; Maza et al., 2015; Ying Ani Hsieh et al., 2004). Additionally, dynamic programming is utilized for optimal flocking, often necessitating at least one remote controller with residential, reactive capabilities, and a comprehensive view of the environment.

In a centralized control system, computation primarily occurs within a centralized control centre, while agents execute instructions without interacting with one another. This approach tends to result in lower overall costs. However, one drawback of centralized systems is that as the number of agents increases, the response speed can significantly diminish due to computational overload at the control centre. Furthermore, if communication between the command centre and the individual agents is compromised, the entire cluster's functionality can be affected.

In contrast, distributed control systems for UAVs, like animal flocks and swarms, do not require a centralized controller; they share information and work cooperatively. One of the most important advantages of distributed computing systems is reliability: the failure of one server's system will not affect the rest of the servers, so the system can still provide services normally. The computing power of multiple computers makes distributed control faster than other system structures. However,

distributed controls are often designed to be complex to ensure data consistency and avoid the data hazards caused by machine failure.

In the crucial initial steps to achieve swarm control without main controllers, our focus lies in constructing a closed-loop control model. This model enables drones to predict motion based on trajectory memory, thereby enhancing their autonomy in trajectory forecasting. In future research, we will transform the path prediction system of virtual agents into a closed-loop system, enabling the enhancement of autonomous path prediction for drones in dynamic environments. Ultimately, the improvement in this autonomous capability not only enhances the accuracy of motion prediction but could also play a pivotal role in optimizing overall coordination within aerial vehicle swarm. Ultimately, individual entities will possess fundamental autonomy, while the group can rely on these individuals to collectively achieve the objectives of group behaviour (Krause & Ruxton, 2002; Ouellette & Gordon, 2021; Ward & Webster, 2016).

Acknowledgments

For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

Data availability statement

The data that support the findings of this study are available from the corresponding author [S. Dong], upon reasonable request.

Author contributions statement

Shuyan Dong designed and performed the experiments, derived the models and analyzed the data. Shuyan Dong wrote the manuscript with support from Saptarshi Das and Stuart Townley. Saptarshi Das and Stuart Townley helped supervise the project. Shuyan Dong, Saptarshi Das and Stuart Townley conceived the original idea.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

- Blaabjerg, F. (2018). *Control of power electronic converters and systems: Volume 2* (Vol. 2). Academic Press.
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3), 268–308. <https://doi.org/10.1145/937503.937505>
- Bukhari, A. H., Sulaiman, M., Islam, S., Shoaib, M., Kumam, P., & M. A. Zahoor Raja (2020). Neuro-fuzzy modeling and prediction of summer precipitation with application to different meteorological stations. *Alexandria Engineering Journal*, 59(1), 101–116. <https://doi.org/10.1016/j.aej.2019.12.011>

- Cajo, R., Mac, T. T., Plaza, D., Copot, C., De Keyser, R., & Ionescu, C. (2019). A survey on fractional order control techniques for unmanned aerial and ground vehicles. *IEEE Access*, 7, 66864–66878. <https://doi.org/10.1109/Access.6287639>
- Chen, W.-H., Rhodes, C., & Liu, C. (2021). Dual control for exploitation and exploration (DCEE) in autonomous search. *Automatica*, 133, 109851. <https://doi.org/10.1016/j.automatica.2021.109851>
- Deng, C., Wang, S., Huang, Z., Tan, Z., & Liu, J. (2014). Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications. *Journal of Communications*, 9(9), 687–692. <https://doi.org/10.12720/jcm.9.9>
- Ding, S. X. (2008). *Model-based fault diagnosis techniques: Design schemes, algorithms, and tools*. Springer Science & Business Media.
- Dong, S., Das, S., Townley, S., & Thornton, A. (2023). Swarm intelligence based drone flocking model. In *2023 28th International Conference on Automation and Computing (ICAC)* (pp. 1–6).
- Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4), 28–39. <https://doi.org/10.1109/MCI.2006.329691>
- Freeman, P., Pandita, R., Srivastava, N., & Balas, G. J. (2013). Model-based and data-driven fault detection performance for a small UAV. *IEEE/ASME Transactions on Mechatronics*, 18(4), 1300–1309. <https://doi.org/10.1109/TMECH.2013.2258678>
- Gao, W. (2020). New ant colony optimization algorithm for the traveling salesman problem. *International Journal of Computational Intelligence Systems*, 13(1), 44–55. <https://doi.org/10.2991/ijcis.d.200117.001>
- Giernacki, W., Skwierczyński, M., Witwicki, W., Wroński, P., & Koziński, P. (2017). Crazyflye 2.0 quadrotor as a platform for research and education in robotics and control engineering. In *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)* (p. 37–42). IEEE.
- Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
- Gunetti, P. (2012). *An intelligent agent based autonomous mission management system for UAVs* [PhD thesis, University of Sheffield].
- Guo, K., Ye, Z., Liu, D., & Peng, X. (2021). UAV flight control sensing enhancement with a data-driven adaptive fusion model. *Reliability Engineering & System Safety*, 213, 107–654.
- Guzmán, J. L., & Hägglund, T. (2024). Tuning rules for feedforward control from measurable disturbances combined with PID control: A review. *International Journal of Control*, 97(1), 2–15. <https://doi.org/10.1080/00207179.2021.1978537>
- Hassani, A. E., & Emary, E. (2016). *Swarm intelligence: Principles, advances, and applications* (1st ed.). CRC Press. <https://doi.org/10.1201/9781315222455>.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Ippolito, C. A., Hashemi, K., Kawamura, E., Gorospe, G., Holforty, W., Kannan, K., Stepanyan, V., Lombaerts, T., Brown, N., Jaffe, A., & Dolph, C. (2023). Concepts for distributed sensing and collaborative airspace autonomy in advanced urban air mobility. In *AIAA SciTech forum and exposition*.
- Jembre, Y. Z., Nugroho, Y. W., Khan, M. T. R., Attique, M., Paul, R., S. H. A. Shah, & Kim, B. (2021). Evaluation of reinforcement and deep learning algorithms in controlling unmanned aerial vehicles. *Applied Sciences*, 11(16)240. <https://doi.org/10.3390/app11167240>
- Kada, B., Khalid, M., & Shaikh, M. S. (2020). Distributed cooperative control of autonomous multi-agent UAV systems using smooth control. *Journal of Systems Engineering and Electronics*, 31(6), 1297–1307. <https://doi.org/10.23919/JSEE.5971804>
- Krause, J., & Ruxton, G. (2002). *Living in groups*. Oxford University Press.
- Laghari, A. A., Jumani, A. K., Laghari, R. A., & Nawaz, H. (2023). Unmanned aerial vehicles: A review. *Cognitive Robotics*, 3, 8–22. <https://doi.org/10.1016/j.cogr.2022.12.004>
- Lambert, N. O., Drew, D. S., Yaconelli, J., Levine, S., Calandra, R., & K. S. J. Pister (2019). Low-level control of a quadrotor with deep model-based reinforcement learning. *IEEE Robotics and Automation Letters*, 4(4), 4224–4230. <https://doi.org/10.1109/LSP.2016>.
- Li, M., Li, G., & Zhong, M. (2016). A data driven fault detection and isolation scheme for UAV flight control system. In *2016 35th Chinese Control Conference (CCC)* (pp. 6778–6783).
- Liu, W. (2013). *Introduction to hybrid vehicle system modeling and control*. John Wiley & Sons.
- Liu, X., Liu, X., Zhou, Z., & Hu, L. (2021). An efficient multi-objective optimization method based on the adaptive approximation model of the radial basis function. *Structural and Multidisciplinary Optimization*, 63(3), 1385–1403. <https://doi.org/10.1007/s00158-020-02766-2>
- Maza, I., Ollero, A., Casado, E., & Scarlatti, D. (2015). Classification of multi-UAV architectures. In K. Valavanis & G. Vachtsevanos (Eds.), *Handbook of unmanned aerial vehicles* (pp. 953–975). Dordrecht: Springer. https://doi.org/10.1007/978-90-481-9707-1_119
- Nooralishahi, P., Ibarra-Castaneda, C., Deane, S., López, F., Pant, S., Genest, M., Avdelidis, N. P., & Maldague, X. P. (2021). Drone-based non-destructive inspection of industrial sites: A review and case studies. *Drones*, 5(4), 106. <https://doi.org/10.3390/drones5040106>
- Ogata, K. (2009). *Modern control engineering*. India: Prentice Hall.
- Ouellette, N. T., & Gordon, D. M. (2021). Goals and limitations of modeling collective behavior in biological systems. *Frontiers in Physics*, 9, 687823. <https://doi.org/10.3389/fphy.2021.687823>
- Preiss, J. A., Honig, W., Sukhatme, G. S., & Ayanian, N. (2017). CrazySwarm: A large nano-quadcopter swarm. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3299–3304).
- Procaccini, A., Orlandi, A., Cavagna, A., Giardina, I., Zoratto, F., Santucci, D., Chiarotti, F., C. K. Hemelrijk, Alleva, E., Parisi, G., & Carere, C. (2011). Propagating waves in starling, *Sturnus vulgaris*, flocks under predation. *Animal Behaviour*, 82(4), 759–765. <https://doi.org/10.1016/j.anbehav.2011.07.006>
- Raptodimos, Y., & Lazakis, I. (2020). Application of NARX neural network for predicting marine engine performance parameters. *Ships and Offshore Structures*, 15(4), 443–452. <https://doi.org/10.1080/17445302.2019.1661619>
- Ruiz, L. G. B., Cuellar, M. P., Calvo-Flores, M. D., & Jimenez, M. D. C. P. (2016). An application of non-linear autoregressive neural networks to predict energy consumption in public buildings. *Energies*, 9(9), 684. <https://doi.org/10.3390/en9090684>
- Samiappan, S., Turnage, G., Hathcock, L., Casagrande, L., Stinson, P., & Moorhead, R. (2017). Using unmanned aerial vehicles for high-resolution remote sensing to map invasive *Phragmites*

- australis* in coastal wetlands. *International Journal of Remote Sensing*, 38(8-10), 2199–2217. <https://doi.org/10.1080/01431161.2016.1239288>
- Skorput, P., Mandzuka, S., & Vojvodic, H. (2016). The use of unmanned aerial vehicles for forest fire monitoring. In *2016 International Symposium ELMAR* (pp. 93–96). IEEE.
- Teodorović, D. (2008). Swarm intelligence systems for transportation engineering: Principles and applications. *Transportation Research Part C: Emerging Technologies*, 16(6), 651–667. <https://doi.org/10.1016/j.trc.2008.03.002>
- Van der Merwe, D., Burchfield, D. R., Witt, T. D., Price, K. P., & Sharda, A. (2020). Drones in agriculture. *Advances in Agronomy*, 162, 1–30. <https://doi.org/10.1016/bs.agron.2020.03.001>
- Ward, A., & Webster, M. (2016). *Sociality: The behaviour of group-living animals* (Vol. XIII, 276). Springer Cham.
- Wunsch, A., Liesch, T., & Broda, S. (2021). Groundwater level forecasting with artificial neural networks: A comparison of long short-term memory (LSTM), convolutional neural networks (CNNs), and non-linear autoregressive networks with exogenous input (NARX). *Hydrology and Earth System Sciences*, 25(3), 1671–1687. <https://doi.org/10.5194/hess-25-1671-2021>
- Ying Ani Hsieh, M., Srivastava, P., Kumar, V., & Taylor, C. J. (2004). Composable communication constraint-based control. In D. W. Gage (Ed.), *Mobile robots XVII. International society for optics and photonics* (Vol. 5609, pp. 192 – 200). SPIE.
- Yu, Z., Qu, Y., & Zhang, Y. (2019). Distributed fault-tolerant cooperative control for multi-UAVs under actuator fault and input saturation. *IEEE Transactions on Control Systems Technology*, 27(6), 2417–2429. <https://doi.org/10.1109/TCST.87>