Swarm Intelligence Based Drone Flocking Model

Shuyan Dong, Saptarshi Das, Stuart Townley
Centre for Environmental Mathematics,
Faculty of Environment, Science and Economy,
University of Exeter, Penryn Campus,
Cornwall TR10 9FE, United Kingdom.
Email: {sd746, s.das3, s.b.townley}@exeter.ac.uk

Alex Thornton
Centre for Ecology and Conservation,
Faculty of Environment, Science and Economy,
University of Exeter, Penryn Campus,
Cornwall TR10 9FE, United Kingdom.
Email: alex.thornton@exeter.ac.uk

***Abstract*—Flocking, shoaling and swarming in animal groups serve a number of functions, including improving information transmission and reducing predation risks. Individuals in biological populations tend to make limited and simple responses to each other and also to stimuli in the environment. But by acting together they can accomplish collective tasks, which is referred to as swarm intelligence. Insights from natural systems have inspired work in numerous areas, such as meta-heuristic optimization, machine learning and image processing. However, the limitations of information sharing, and transfer make it difficult to solve real-world engineering problems in physical world using the swarm intelligence mechanism. This contrasts with natural systems where, for example, birds use social information to improve sensing of environmental cues and make decisions without lag during flight. Thus, behavioural modelling of animal swarming may provide new insights into this problem. Here, we show comparsion of two data-driven deep neural network models for drone flocking.**

***Index Terms*—collective behaviour, swarm intelligence, artificial intelligence, self-organization model**

I. Introduction

Animal collective behaviour refers to a biological phenomenon in which interactions between individuals give rise to coordinated group movement and decision-making. Tasks and goals may include foraging, breeding, and migration. Animal collectives can therefore arise through an emergent process of self-organisation. Animal collectives are self-organized, with no individual knowing the entire state of the system or acting as a leader [1]. The study of animal collective behaviour has attracted wide attention as it provides insights into the mechanisms through which group-level phenomena can emerge from local interactions between agents, generating major ecological and evolutionary consequences [2]–[6].

Although individuals in biological populations tend to make limited and simple responses to neighbours and external stimuli, these local responses can give rise to collective intelligence, enabling groups to accomplish collective tasks that would be beyond the abilities of any individual. This is known as collective or swarm intelligence.

In recent years, with the development of artificial intelligence (AI) technologies, especially in deep neural networks, more researchers have started to focus on the relationship between swarm intelligence and new AI technologies. Nowadays, swarm intelligence is applied in various areas such as complex function optimization, industrial automation, robotics, machine learning, and image processing.

Swarm intelligence can be used to find globally optimal solutions by simulating the collaborative swarm behaviour of agents. The majority of swarm intelligence algorithms are straightforward to implement, reliable, and robust for solving complex optimization problems [7]. For instance, ant colony optimization (ACO) and particle swarm optimization (PSO) algorithms have demonstrated excellent performance in solving a wide variety of constrained and unconstrained optimisation problems [8]–[10]. Furthermore, introducing swarm intelligence into natural language processing (NLP) has led to significant improvements in translation accuracy and performance [11]. Gao [12] proposed a new optimization method based on ant colony algorithms and neural networks. By simulating the foraging behaviour of ants, Gao's method achieves faster network convergence and better performance. Liu *et al.* proposed a method based on multi-objective optimization and swarm intelligence, which has shown superior performance and robustness across multiple tasks [13].

Swarm intelligence can also enhance the decision-making ability of artificial intelligence and neural networks by integrating the opinions of multiple agents through a collective "wisdom of the crowds" approach, generating more accurate and reliable decisions. This approach has applications in fields, such as finance, healthcare, and transportation [14]–[16].

Moreover, swarm intelligence can help artificial intelligence better adapt to the environment. In some applications, the environment may be uncertain or dynamic, requiring the artificial intelligence system to quickly adapt to changes. With swarm intelligence technology, artificial intelligence systems can better handle these environmental changes. For example, swarm intelligence algorithms can optimise agents' behaviour to better adapt to constantly changing environments.

Although natural systems have inspired work in artificial intelligence areas, the limitations of information sharing and transfer make it difficult to solve engineering problems in the physical world. In contrast, in the natural world, birds for example can use social information (e.g. responding to the movement of other individuals) to improve sensing of environmental cues and make decisions without lag during flight. If considering this phenomenon as an algorithm, it can be viewed as self-organised. As we know that swarm intelligence enables efficient collaboration through the use of simple, local rules of interaction between agents, and shows high autonomy. Thus, swarm intelligence may provide new insights into information sharing and transfer. By introducing swarm intelligence to the multi-robot system, the autonomy of the system can be improved, such as each of the robots is able to show simple responses to neighbours. Therefore, information transfer problems are likely to be solved.

II. Drone Flocking as Technology Support

Being an emerging technology, drones are used in many different fields, such as power infrastructure inspection, security and fire protection, agriculture and farming, industrial, film, television and entertainment and light shows etc. [17]–[19]. Autonomous drones enable closer and more accurate inspection of large sites or hazardous equipment in the industry. In addition, they can provide visual or other survey data to equipment or other objects in extreme or dangerous environments more closely than humans [18], [20], [21].

The previous studies on unmanned aerial vehicles (UAVs) have mainly focused on UAVs' controlling via a centralized control system or distributed control system [22]–[24], or using dynamic programming for optimal flocking and control at a particular time, which requires personnel to be on-site for control or pre-coding. In a centralized control system, computation is mainly done in the control centre and agents are only used for performing instructions. So that the drones do not interact with each other. Although, the general cost is relatively low. One of the shortcomings of a centralized system is that with increasing number of agents, the response speed will be significantly reduced due to the overload of the computing centre. Moreover, if the communication between the command centre and the individuals is compromised, the entire cluster will not work.

In contrast, distributed control systems for UAVs do not require a centralised controller and they share information and work cooperative mode. 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. The system can still provide services normally to the outside world. The computing power of multiple computers makes distributed control faster than other system structures. However, distributed control are often designed to be complex to ensure data consistency and avoid the data hazards caused by machine failure.

We discuss whether it would be possible to overcome existing problems in the centralized control system and the decentralized system by designing a new flocking system that uses swarm intelligence in the control system. By harnessing the adaptive and self-organizing nature of swarm algorithms, a new flocking system may be designed with minimal human intervention. Introducing swarm intelligence into cluster modelling has the potential to enhance the adaptability, flexibility, and efficiency of UAV clusters and improve their environmental performance.

III. Methodology

A. Experimental Setting and Data Collection

To implement the self-organized model, we recorded a large amount of flight data, including but not limited to small regular patterns, different unidirectional flying patterns such as going left, going right, taking off, landing, flying forward, flying backwards and random walks with limited range. Our experiments are carried out on the *Crazyflie*, which is a versatile open-source flying development platform that only weighs 27g [25]. Due to its light weight, *Crazyflies* have been used in small robotics or multiple robotics research and applications.

Based on the above mentioned flight data, we modelled the motion of the drones within a machine learning framework. This may be helpful in performing 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 theoretically lead to smoother flight and improved controllability. After a successful run of the virtual experiment, the model outputs are assigned to the real drone so that it can be controlled directly. The results of the test run are observed to be the same as those of the virtual experiment. According to the above setting, the robotic control system, motor input data or pulse width modulation (PWM), 3D position data (X, Y, Z), and 4 features (Roll, Pitch, Yaw and Thrust) from the stabilizer of the drone are recorded during flying the drone. Detailed descriptions and sizes are shown in Table I.

TABLE I
Data description

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

PWM inputs of the motors are recorded as the features of control technology for electronic power applications [26], [27]. PWM is an irreplaceable link in a stable and flexible method of controlling the speed of a DC motor [27]. The position is in three dimensions that represent the position of the drone 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 robotics, roll, pitch, and yaw are important terms for describing the movements.

The flight data is captured during the operation of the drone, and is inherently temporal in nature, given that it is collected over a specific time interval. The inherent temporal and spatial dependencies in flight data may result in adjacent data points being interrelated and potentially influencing

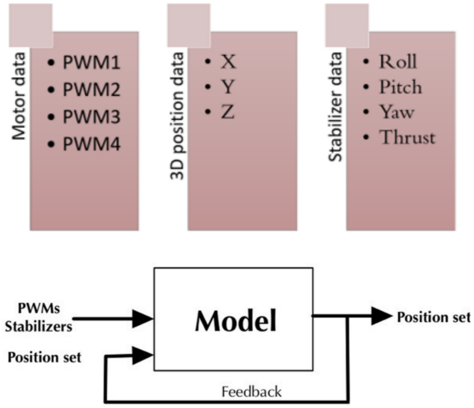

Fig. 1. Variable arrangements and schematic of data generated to train the virtual drone model.

each other. Furthermore, flight data is spatially correlated, as different flight paths and altitudes may have an impact on the data. In this paper, 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 Fig. 1. These multiple dimensions of data are intricately related and require sophisticated analysis and modelling technique to extract meaningful insights.

B. Data Pre-prossessing

During the flight, noise and outliers are frequently encountered in the recorded data, which can lead to inaccuracies and instabilities in data analysis and modelling. Flight data may be unstable in some cases. Therefore data processing is important for the design of the model.

First, we used a sliding window to supplement the missing data and smooth outliers. Then normalized each of the three groups of parameters. In the flight experiments in which the data was recorded, the drones experienced an uncontrollable lateral offset that became more severe with each repetition of the flying pattern. This offset can be interpreted as a trend, and we used Matlab's built-in function $detrend$ to reduce the effect of the offset.

C. Exploratory Data Analysis

In order to have a better exploration of the relationships between features, correlation metrics are investigated. The correlation metrics between pair of features of the general flight data for multiple $detrended$ flight trajectories were analysed. However, when integrating flight data from different flight paths and performing data correlation analysis, it was found that the data recorded on the *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 Fig. 2 and Fig. 3, respectively.

It can be seen from Fig. 2 that the position features (X, Y, Z) are weekly correlated to most of the rest features as

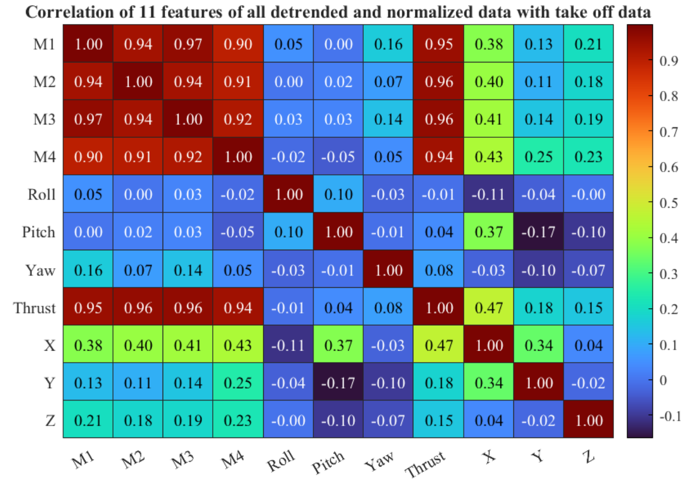

Fig. 2. Feature correlations between 11 variables on the data including *taking off* experiment.

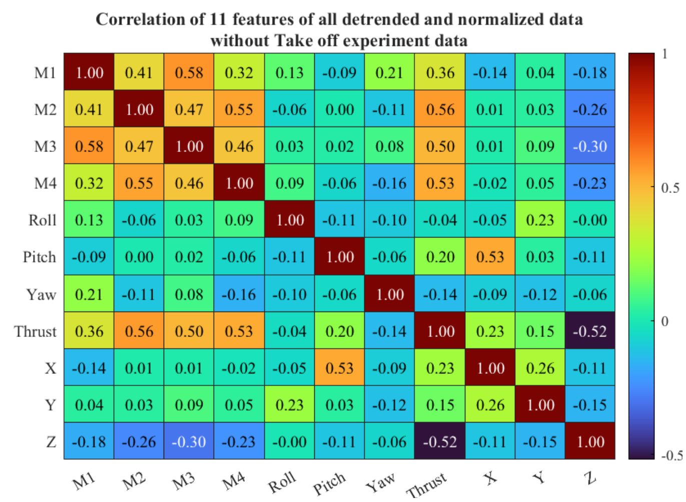

Fig. 3. Feature correlations between 11 variables on the data without *taking off* experiment.

well as with each other. It shows a significant correlation between PWM inputs of the Motors ($M1, M2, M3, M4$ shown in Fig. 2) when including *taking off* and $thrust$ is highly correlated with the output of the motors. When only considering the rest of the trajectories' data, the correlations of Motors' output are not significantly high. Although $thrust$ shows about 0.5 correlation to the motors' output, the correlations are not as high as the one including *taking off* data in Fig. 3.

So 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.

IV. Results and Discussions

A. LSTM Based Drone Model

Considering the characteristics of the data, firstly, we build our model based on the Long Short-Term Memory (LSTM) deep neural network. LSTM is a kind of Recurrent Neural

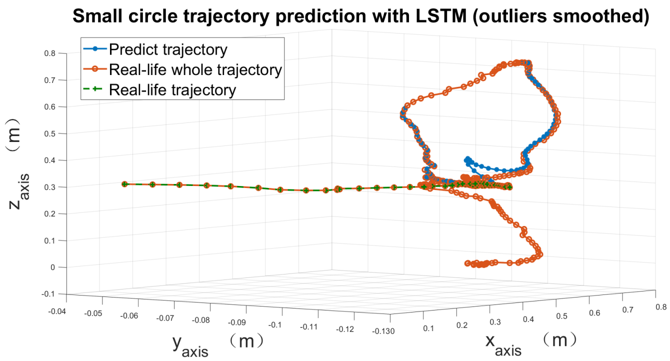

Fig. 4. The LSTM model trained and tested on *small circle trajectory*. The predicted trajectory for the last 100 steps is represented with a blue point line. The real trajectory for the last 100 steps is represented with purple short lines, while the orange line shows the whole trajectory of *small circle trajectory*.

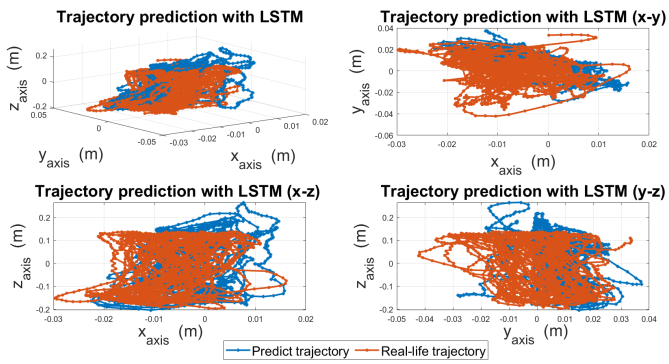

Fig. 5. The LSTM model trained and tested on *taking off and landing trajectory*. The top right image displays the true trajectory and the predicted trajectory of *taking off and landing trajectory* in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

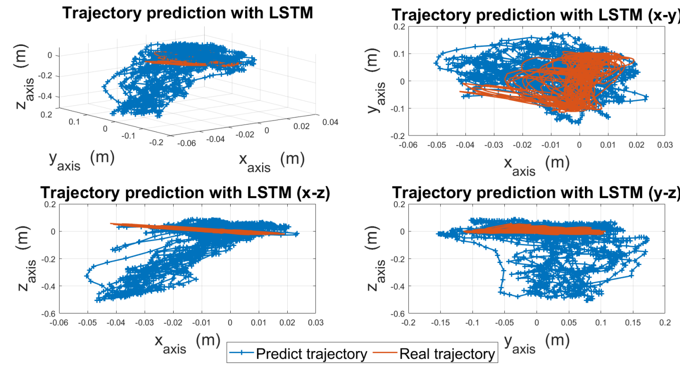

Fig. 6. The LSTM model trained and tested on *going left*. The top right image displays the true trajectory and the predicted trajectory in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

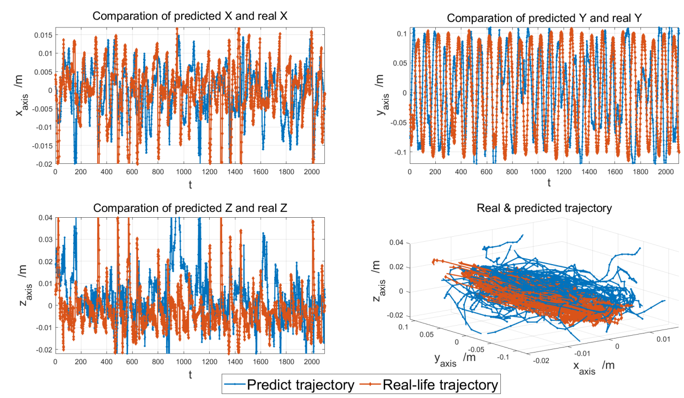

Fig. 7. The comparison between predicted and actual trajectory on a single dimension generated by the LSTM model applied on *going left* is shown. The bottom right image displays the true trajectory and the predicted trajectory of *going left* in 3D. The remaining three images show the true and predicted trajectory on separate dimensions.

Network (RNN) which is suitable for processing sequence data such as speech, text, and time series [28], [29]. Each unit of LSTM can continuously store and update information over multiple time steps, so that it can solve the problem of both long-term and short-term dependencies better.

We first conduct experiments on a small circle trajectory. The *small circle trajectory* has 300 sets of data-points, as shown in Table I. The first 200 sets of data-points are used to train the model and predict the next paths. The last step of the training dataset is used as the first step of the prediction. As an initial experiment, we mainly explored the relationship between the output data-points PWMs of the four motors and the position of the drone that is *(X, Y, Z)*. Therefore, both input and output features are 7-dimensional. The result is shown in Fig. 4. It can be noticed that the predicted trajectory of the drone was found to be very similar to the first 200 sets of data-points. Although the predicted trajectory is far from the real one, it is very similar to the trained trajectory, which indicates that LSTM does have the ability to predict the flight paths. When applying the LSTM model with the same structure to the larger datasets, it produced very similar patterns for regular flight path prediction (Fig.5). However, in the data of *going left*, the predicted results differ from the real ones, which can be easily noticed from Fig. 6. The model provides a more accurate prediction on the y-direction, however, there is a significant bias in the predition of the z-direction. A further comparison of The predicted and true values in each of the three directions is shown in Fig. 7.

Also, due to the small size of the experimental drones, the prediction accuracy is expected to be no greater than 10 centimetres, which is difficult for the LSTM model. When further predicting *roll*, *pitch*, *yaw*, and *thrust*, we found that the accuracy was extremely low. This may be because LSTM is more suitable for handling sequences with long-term dependencies and is susceptible to noise interference. Therefore, we tried to use the NARX model as the structural basis, which can better predict nonlinear time series. NARX has better robustness to noise and outliers in the input data, and by using multiple time-series as external inputs, it allows for more flexible modelling of multiple time series. As compared to LSTM, NARX provides faster computational speed for learning patterns from the drone flight data.

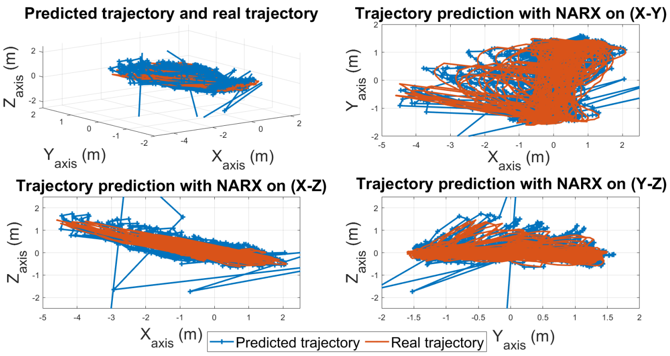

Fig. 8. The NARX model trained and tested on *going left*. The top right image displays the true trajectory and the predicted trajectory of *going left* in 3D. The remaining three images show the true and predicted trajectory on different combinations of dimensions.

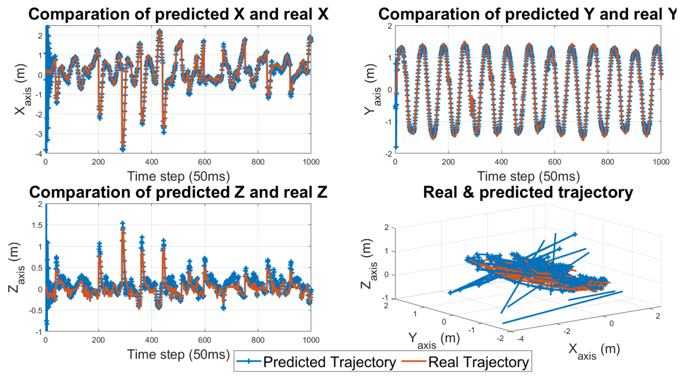

Fig. 9. The comparison between predicted and actual trajectory on a single dimension generated by the NARX model applied on *going left* is shown. The bottom right image displays the true trajectory and the predicted trajectory of *going left* in 3D. The remaining three images show the true and predicted trajectory on separate dimensions.

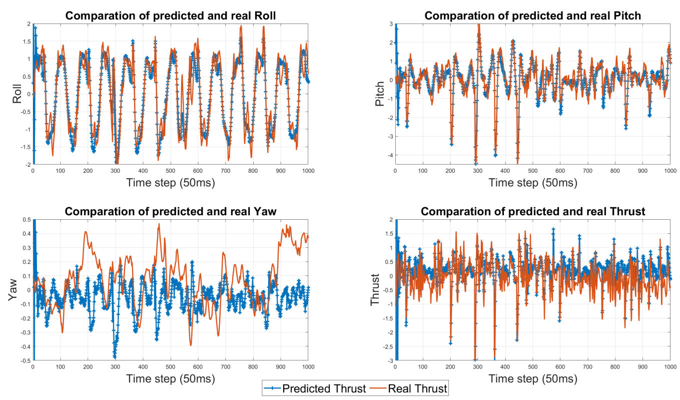

Fig. 10. The comparison between predicted and actual *stabilizers* generated by the NARX model applied on *going left*.

TABLE II
MODEL VALIDATION (MSE & R^2) OF LSTM AND NARX ON *going left*

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

B. NARX Based Drone Model

Next, we modelled the *going left* data using the nonlinear autoregressive with exogenous input (NARX) neural network model. The model takes a series of positions *(X, Y, Z)* as inputs and recurrent inputs which is 11 dimensional. After training, we compared the predicted trajectory and true trajectory, as well as the predicted stabilizer data (*roll*, *pitch*, *yaw*, and *thrust*) and the true values. Fig. 8 shows the true and predicted trajectory on different combinations of dimensions.

As compared to the trajectory predicted by the LSTM model shown in Fig. 6, the NARX model provides a significantly more accurate prediction. The specific comparison of the results can be seen in Table II. Table III further shows the goodness of fit statistics (R^2) for each variable which shows that in general NARX gives better predictions. Furthermore, the NARX model has an accuracy rate of 70% for predicting path errors within 0.15 meters and an even higher accuracy rate of 90% for errors less than 0.25 meters.

The clearer comparison between predicted and actual positions on a single dimension is shown in Fig. 9. After visualizing the predicted *stabilizer data* and the true stabilizers, as shown in Fig. 10, it can be noticed that the NARX model can fit the variations of roll, pitch, and thrust well, but for relatively stable data, yaw, the predicted values showed significant fluctuations. In order to better analyse the performance of the model, the auto-correlation of the error are analysed. We found that the model slightly deviates from the delta function like auto-correlation of the error, which means there are trends or seasonal features that are not involved in the current model. This canbe considered as under-fitting. Normally, improving the complexity, larger dataset and reducing the number of features can solve this problem. Then we increased the size of the training set and found that the error is still auto-correlated, but the correlation has slightly decreased.

TABLE III
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 200^{th})
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

V. CONCLUSION AND FUTURE WORK

Comparing the experimental results, it can be seen that digitalizing the flight of the drones with the NARX model generated relatively better results. However, upon visualizing the output and analyzing the autocorrelation of the errors, we found that the model is under-fitting. Therefore, we will further modify the current NARX model. The purpose of this model is

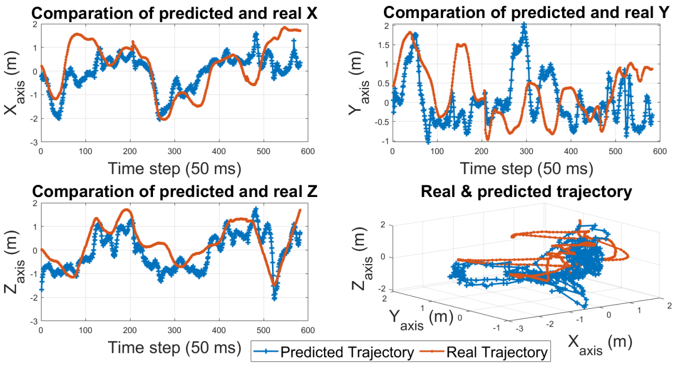

Fig. 11. Predicted and real trajectory of *small random walk* with the 2^{nd} edition NARX. The bottom right image displays the true trajectory and the predicted trajectory in 3D. The remaining three images show the true and predicted trajectory on separate dimensions.

data-driven learning of the flight pattern of the drone, which will provide us with a more stable and precise method of automatic control. This model is named as *virtual drone*.

Due to the upcoming application of the model for obstacle avoidance under unknown environmental conditions, as well as for cooperative obstacle avoidance among multiple Virtual Drones, it is necessary to further improve the model's structure. As a preliminary attempt, we decided to modify the input and output structure of the virtual drone to better reflect the environmental information encountered during flight.

Based on the operating theory of the drone, we hypothesized that the operation of the motor causes changes in position, and therefore, we tested whether the PWM could be used as an external input for the model. In addition, we expect the model to provide predictions related to the drone's position. To test these modifications, we trained and tested the second-generation virtual drone model structure on a simulated environment called *small random walk*. The results of comparing the predicted and actual positions are shown in Fig. 11. It can be seen that the model is able to predict the positions or trends in the X and Z directions relatively well, but there is a significant bias in the prediction of the Y direction.

References

[1] N. T. Ouellette and D. M. Gordon, "Goals and limitations of modeling collective behavior in biological systems," *Frontiers in Physics*, vol. 9, p. 687823, 2021.

[2] M. M. Sosna, C. R. Twomey, J. Bak-Coleman, W. Poel, B. C. Daniels, P. Romanczuk, and I. D. Couzin, "Individual and collective encoding of risk in animal groups," *Proceedings of the National Academy of Sciences*, vol. 116, no. 41, pp. 20 556–20 561, 2019.

[3] J. Krause and G. Ruxton, "Living in groups: Oxford univ. press," 2002.

[4] A. Ward and M. Webster, "Sociality: the behaviour of group-living animals," 2016.

[5] J. W. Jolles, A. J. King, and S. S. Killen, "The role of individual heterogeneity in collective animal behaviour," *Trends in Ecology & Evolution*, vol. 35, no. 3, pp. 278–291, 2020.

[6] L. F. Hughey, A. M. Hein, A. Strandburg-Peshkin, and F. H. Jensen, "Challenges and solutions for studying collective animal behaviour in the wild," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 373, no. 1746, p. 20170005, 2018.

[7] M. P. Saka, E. Doğan, and I. Aydogdu, "Analysis of swarm intelligence–based algorithms for constrained optimization," in *Swarm intelligence and bio-inspired computation*. Elsevier, 2013, pp. 25–48.

[8] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[9] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[10] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[11] W. Al-Saeedan and M. E. B. Menai, "Swarm intelligence for natural language processing," *International Journal of Artificial Intelligence and Soft Computing*, vol. 5, no. 2, pp. 117–150, 2015.

[12] W. Gao, "New ant colony optimization algorithm for the traveling salesman problem," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 44–55, 2020.

[13] X. Liu, X. Liu, Z. Zhou, and L. Hu, "An efficient multi-objective optimization method based on the adaptive approximation model of the radial basis function," *Structural and Multidisciplinary Optimization*, vol. 63, pp. 1385–1403, 2021.

[14] J. Uthayakumar, N. Metawa, K. Shankar, and S. Lakshmanaprabu, "Financial crisis prediction model using ant colony optimization," *International Journal of Information Management*, vol. 50, pp. 538–556, 2020.

[15] A. Bohr and K. Memarzadeh, "The rise of artificial intelligence in healthcare applications," in *Artificial Intelligence in healthcare*. Elsevier, 2020, pp. 25–60.

[16] D. Teodorović, "Swarm intelligence systems for transportation engineering: Principles and applications," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 651–667, 2008.

[17] D. van der Merwe, D. R. Burchfield, T. D. Witt, K. P. Price, and A. Sharda, "Drones in agriculture," *Advances in Agronomy*, vol. 162, pp. 1–30, 2020.

[18] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications." *Journal of Communications*, vol. 9, no. 9, pp. 687–692, 2014.

[19] P. Skorput, S. Mandzuka, and H. Vojvodic, "The use of unmanned aerial vehicles for forest fire monitoring," in *2016 International Symposium ELMAR*. IEEE, 2016, pp. 93–96.

[20] S. Samiappan, G. Turnage, L. Hathcock, L. Casagrande, P. Stinson, and R. Moorhead, "Using unmanned aerial vehicles for high-resolution remote sensing to map invasive phragmites australis in coastal wetlands," *International Journal of Remote Sensing*, vol. 38, no. 8-10, pp. 2199–2217, 2017.

[21] P. Nooralishahi, C. Ibarra-Castanedo, S. Deane, F. López, S. Pant, M. Genest, N. P. Avdelidis, and X. P. Maldague, "Drone-based non-destructive inspection of industrial sites: A review and case studies," *Drones*, vol. 5, no. 4, p. 106, 2021.

[22] R. Cajo, T. T. Mac, D. Plaza, C. Copot, R. De Keyser, and C. Ionescu, "A survey on fractional order control techniques for unmanned aerial and ground vehicles," *IEEE Access*, vol. 7, pp. 66 864–66 878, 2019.

[23] B. Kada, M. Khalid, and M. S. Shaikh, "Distributed cooperative control of autonomous multi-agent uav systems using smooth control," *Journal of Systems Engineering and Electronics*, vol. 31, no. 6, pp. 1297–1307, 2020.

[24] I. Maza, A. Ollero, E. Casado, and D. Scarlatti, "Classification of multi-uav architectures," *Handbook of Unmanned Aerial Vehicles*, pp. 953–975, 2015.

[25] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 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)*. IEEE, 2017, pp. 37–42.

[26] F. Blaabjerg, *Control of Power Electronic Converters and Systems: Volume 2*. Academic Press, 2018, vol. 2.

[27] W. Liu, *Introduction to hybrid vehicle system modeling and control*. John Wiley & Sons, 2013.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.