DOI: 10.13009/EUCASS2023-965
Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS

A Simplified Digital Twin Framework of a Trainer Aircraft

Dominik Reinhardt and Abdessalem Bouferrouk*†*
**University of the West of Engand Bristol, Frenchay Campus, Bristol, BS16 1QY*
dominikrei@hotmail.de – abdessalem.bouferrouk@uwe.ac.uk
† Corresponding Author

Abstract

This paper aims to establish a simplified, user-friendly framework for a Digital Twin (DT) of a PC21 jet trainer aircraft. The implementation of the DT is carried out in MATLAB with the aid of embedded Neural Network tools. The aircraft is trained and tested with flight data provided by Qinetiq/ETPS, as well as simulation data obtained via UDP connection from X-Plane, via Simulink. The paper investigates the potential of DTs for enhancing the design, testing, and certification process of a trainer aircraft, with the goal of minimising human error, costs, effort, and environmental impact. The testing of the DT framework with manoeuvres from real flight data shows the feasibility of using a Neural Network to simulate the behaviour of a trainer aircraft. Despite the limitations faced during the execution of the project, the predictions of the network, for a similar manoeuvre as the one it was trained with, show promising results. This proves that the DT framework, with further development and careful adjusting, can be utilised as a simple tool for simulating the flight dynamics of an aircraft.

1. Introduction

With the evolving aviation industry and the shift toward products that need to be technologically smart and connected, the design and testing process needs to be an equally intelligent process. In recent years, the aviation industry has been exploring various methods to improve the design, testing, and certification process of aircraft. One such method is the development of DT technology, which represents a virtual copy of a physical system enabling real-time monitoring, prediction, and optimisation of the system's behaviour. The usage of DTs has the potential to minimise the costs, effort, and environmental impact that is related to the design, testing and certification processes of aircraft, due to the capability of simulating the flight dynamics and aircraft's behaviour. Through the implementation of a DT to simulate the behaviour of an aircraft in a virtual environment, improvements in efficiency, safety, and cost-effectiveness can be achieved.

A Digital Twin is a virtual replica of a physical system or product containing and processing information about the physical system. DTs usually are composed of three main components: the physical product, the virtual model, and the connection or communication between them. In aerospace, for example, a DT can be employed to predict the behaviour of an aircraft during flight by creating a virtual model of the aircraft [1]. To minimise costs, effort, and time put into the testing process, for example Software-in-the-Loop (SIL) simulations are being used to build virtual models of a product to test and adjust the digital prototype before building a physical product [2]. DTs generally offer the benefit of reducing the time and cost of physical prototyping and testing, while still being able to be employed throughout the whole product life cycle [3].

There are different types of DTs, mainly analytical and empirical models. Analytical models are based on sets of mathematical equations representing the physical behaviour of a product. To build an analytical model of a system a deep understanding of the physical principles governing the system is required, together with the ability to translate those principles into mathematical expressions. Analytical models can predict the behaviour of the physical system under different conditions, without needing any physical testing [4]. Empirical models are based on collected data from the physical system it represents. These models use machine learning techniques, such as Neural Networks, to create models by analysing large amounts of system data. Empirical models are useful for predicting the behaviour of complex systems where the physical principles are not well understood or cannot be properly represented by analytical models. The models can predict the behaviour of a product based on past behaviour [4].
DTs can be designed for a variety of purposes, for example, to simulate the behaviour of a system during the design phase, or for monitoring and maintenance purposes while already using the product. Additionally, DTs can be

DOI: 10.13009/EUCASS2023-965

differentiated based on the level of detail they provide, ranging from low-fidelity models providing a general overview of a system, to detailed models that simulate every aspect of a system's behaviour [5].

Several papers in the literature outline the work of Karen E. Wilcox and her team in creating DTs of unmanned aerial vehicles (UAV). The results achieved demonstrate methodologies to build DTs of UAVs serving different purposes, utilising physics-based modelling and data-driven flight capability estimations. One of the uses of the DT was to update, online, the flight envelope of an UAV that is subjected to structural degradation to cover a range of damage scenarios [6]. An alternative application of the DT was to enable path planning for an UAV, making the vehicle adjust its trajectory according to measurements it receives to retain a high probability of mission success [7].

Another methodology proposed included the scaling of DTs from individual components of an UAV to full-scale industrial systems, representing the whole product rather than each part individually [8]. This way, the required level of fidelity can be dictated by the needs of the application considered, thus reducing and adjusting the resolution of each part. The library of component-based reduced-order models for the data-driven physics-based DTs was proposed separately to give the DT methodologies access to the reduced-order models needed to add up to a model of a whole aircraft [9]. The different results indicate how several foundations and frameworks for building predictive DTs have been developed. The tools applied to achieve the required frameworks were component-based reduced-order models, simulation software, interpretable machine learning [9] and probabilistic graphical models together with a Bayesian neural network [10]. Furthermore, it was emphasised that the complexity of flight data presents a challenge for modelling and simulating flying systems [10].

Other approaches have utilised Neural Networks to build the DT framework to simulate or predict the behaviour of complex systems like aircraft. For example, several approaches have successfully been used to model the landing phase of an aircraft, using only a Neural Network [11] or a Neural Network combined with a fuzzy supervisory control strategy [12]. In [11], it was noted that using a Neural Network may be inconvenient, because a similar level of effort is needed to compile data for the network, which could be the result of peculiarity of flight data recording. In addition, the Neural Network structures and learning models match only the type of task they have been designed for [11]. Other projects attempted to simulate the flight dynamics and trajectory of aircraft, such as of the Boeing 747-100 to provide a computationally efficient method to achieve real time prognostics of the air transportation system [13]. In the article a purely data-driven Neural Network is compared to a physics-based recurrent Neural Network structure, demonstrating that the integration of aircraft dynamics into a data-driven model significantly improves the prediction performance.

Although artificial Neural Networks are well-suited to handle complex flight data, they require large amounts of training data, and careful parameter tuning to achieve accurate results [14]. Additionally, the complexity and nonlinearity of flight data forces the need for appropriate model complexity in Neural Networks, to effectively simulate flight dynamics [15]. Several projects have encountered the issue of overfitting to training data, which occurs when a network becomes too complex while adjusting to training data, resulting in fitting too close to the training data and its noise instead of the underlying physical patterns. This leads to poor performance on new, unseen data, regardless of the amount of training conducted [16]. Overfitting may be prevented by implementing dropout regularisation and early stopping during training [18], applying data augmentation techniques to generate more training data [17], by introducing probabilistic modelling into the training [19] or by combining the network with an additional simulation method to improve the prediction accuracy.

The aim of this paper is to develop a DT framework of a trainer aircraft (PC21) using a Neural Network structure. In doing so, the feasibility of using machine learning to predict the behaviour of an aircraft is evaluated based on real flight data as well as simulation data from the X-Plane software. To achieve this, a few procedures need to be performed including processing of appropriate flight data from different sources, design of a capable Neural Network structure, crafting of interconnected framework structure to utilise flight data, comparison and testing of Neural Network structures, and evaluation of results achieved and feasibility of DT framework.

2. Elements of the DT

2.1 Neural Networks

A neural network, such as the one implemented in MATLAB in this paper, typically includes an input layer, an output layer, and one or more hidden layers in between. The purpose and complexity of each layer can vary, and the number of neurons in the input and output layers represent the number of inputs and outputs the neural network processes [20].

DOI: 10.13009/EUCASS2023-965

During the training of a Neural Network the network adjusts the weights and biases of its connections to minimise the difference between the predicted output, based on the training inputs, and the actual output of the training data, called targets [21]. The network processes a set of training data consisting of inputs and targets. When the input is fed into the network it generates an output based on its current weights and biases. The difference between the prediction and the target is then calculated to adjust the network's parameters to better fit to the target data, with the goal to accurately predict the outputs for unseen inputs. For updating the parameters of the network, the backpropagation algorithm is applied, which computes the gradient of the loss function with respect to the network's weights and biases. The gradient is then used to adjust the weights and biases of the network. The training process continues for a specified number of iterations, covering the whole set of training data each time, or until the accuracy on the validation data stops improving [22].

2.2 Digital Twin Blueprint

The approach used to build the DT was to apply an empirical model, as the virtual copy is based on real flight data. The DT framework implemented in the project predicts the reaction of the aircraft based on past behaviour, as it is being trained with a manoeuvre to then predict the reaction on a new series of inputs. The tool to build the DT is a Neural Network that has been executed in MATLAB, to model the behaviour of the aircraft through training with training data, including input and target time series that has been provided by an industry partner (Qinetiq/ETPS). The Neural Network is trained for inputting existing manoeuvres to predict the behaviour of the aircraft during similar manoeuvres. This is achieved using similar, provided manoeuvres, where the manoeuvre's aim itself remains the same, but the conditions are altered, meaning a slight change in control inputs by the pilot and thus in the reaction of the aircraft. As a result, the Neural Network can learn the behaviour of the aircraft during specific manoeuvres and predict the outcome of a similar manoeuvre to a certain degree of accuracy. To determine the best Neural Network structure and the best combination of network and training functions, testing has been conducted to try several combinations and select the best solution.

2.3 Flight Test Data

In the received flight data, there are similar manoeuvres, where the manoeuvre itself stays the same, but the initial conditions, in this case in particular the speed, changes. These manoeuvres are the Aileron Only Turn (AOT), Pitch Doublet, Roll Performance (Roll Perf), Rudder Doublet, Steady Heading Sideslip (SHSS), and Steady Pulls/Pushes manoeuvres. During the project, these manoeuvres data was used by training the Neural Network with one manoeuvre and then predicting similar manoeuvre. There are two more related manoeuvres, due to their similarity of movement, that have been used to train and test a Neural Network. These manoeuvres are Steady Pulls to 4g and Sudden Pulls. Additionally, some manoeuvres have no similar manoeuvre related to it, with their usage being limited to the training of a fully trained Neural Network. These manoeuvres are the Flight Control Mechanical Characteristics (FCMC), Level Acceleration Deceleration (Level Accel Decel), Sinusoidal Stick Pumping (SSSP), Steady Pushes to -1g, Trim C_x Airbrake and Wind Up Turns (WUT). The received flight data had to be filtered so that it only contained the measures needed for the project, and then formatted for use in MATLAB.

3. Neural Network Structure and Implementation

The network needs to be capable of being trained with the provided inputs and targets from either real flight data or X-Plane software data, and to be able to predict the targets of a manoeuvre by inputting the inputs to a previously trained network. Inputs refer to the control inputs of the pilot, being the elevator, aileron, rudder, and throttle input. Targets refer to the reaction of the aircraft to the control inputs, being the behaviour of the aircraft. The targets consist of the pitch, roll, TAS, and Altitude.

The building of the Neural Network ultimately required a try-and-error approach. The final approach was implemented using the provided Neural Network functionality in MATLAB. The implemented Neural Network structure was supposed to learn and simulate the relationship between the control inputs by the pilot and the flight dynamics, being the reaction of the aircraft. The final version of the Digital Twin framework consisted of 7 different MATLAB files, each with their individual purpose (see Table 1).

DOI: 10.13009/EUCASS2023-965

Table 1: MATLAB files

File	Purpose
neural_network_creation.m	Create a Neural Network and train it with a selected manoeuvre from real flight data.
neural_network_creation_XPlane.m	Create a Neural Network and train it with simulated flight data by flying a manoeuvre in X-Plane.
neural_network_train.m	Train an existing Neural Network with a selected manoeuvre from real flight data.
neural_network_train_multiple.m	Train an existing Neural Network with all the existing manoeuvres from real flight data.
neural_network_train_XPlane.m	Train an existing Neural Network with simulated flight data by flying a manoeuvre in X-Plane.
neural_network_test.m	Test a Neural Network with a selected manoeuvre from real flight data and receive the predictions of the Network compared with the real targets.
neural_network_test_XPlane.m	Test a Neural Network with simulated flight data by flying a manoeuvre in X-Plane and receive the predictions of the Network compared with the real targets.

4. Building the Digital Twin Framework

4.1 Preparation of Flight Data

The provided data has been measured in standard orientations, meaning a positive input leads to a negative response from the aircraft. In practice, this means that pushing the stick forward is defined as a positive elevator input and leads to the nose of the aircraft facing down, being a negative pitch response of the aircraft. Pushing the stick to the left is defined as a positive aileron input and leads to the aircraft to roll to the left, which is defined as a negative roll response. Pushing the left rudder is defined as a positive rudder input and it leads to the nose of the aircraft to turn left, being defined as a negative yaw response. Figure 1 displays the principal axes of an aircraft, with the positive turning direction of each axis indicated.

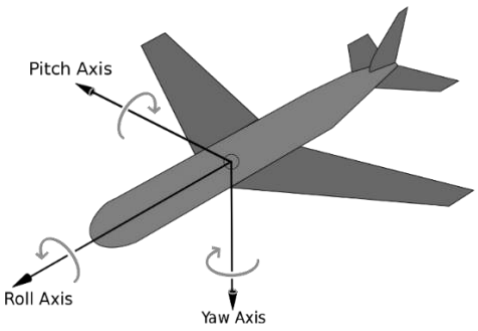

Figure 1: Standard orientation of the Axes of an aircraft

DOI: 10.13009/EUCASS2023-965

The real measurements received by QinetiQ/ETPS were different to the simulation data obtained from X-Plane, because X-Plane defines a positive control input to lead to a positive reaction from the airplane. This means that whenever a control input is negative in the real flight data, the same control input would be positive in X-Plane, with both inputs leading to a positive reaction of the aircraft. To provide consistency throughout the paper, the control inputs in the real flight data were formatted to match the control inputs from X-Plane. This means whenever a control input value is negative in the flight data, it is formatted to be positive when processed.

4.2 Data Transfer

To use the data that was available either from real flight data or X-Plane in the form of simulated flight data, it had to be transferred to MATLAB, where it could be utilised to feed the Neural Network for further utilisation. The data transfer had to be differentiated between the sources of data, being either real flight measurements or X-Plane simulated flight data. Figure 2 and Figure 3 display the utilised data transfer for each source of flight data for an exemplary manoeuvre and a Neural Network with 2 layers and 32 neurons per layer.

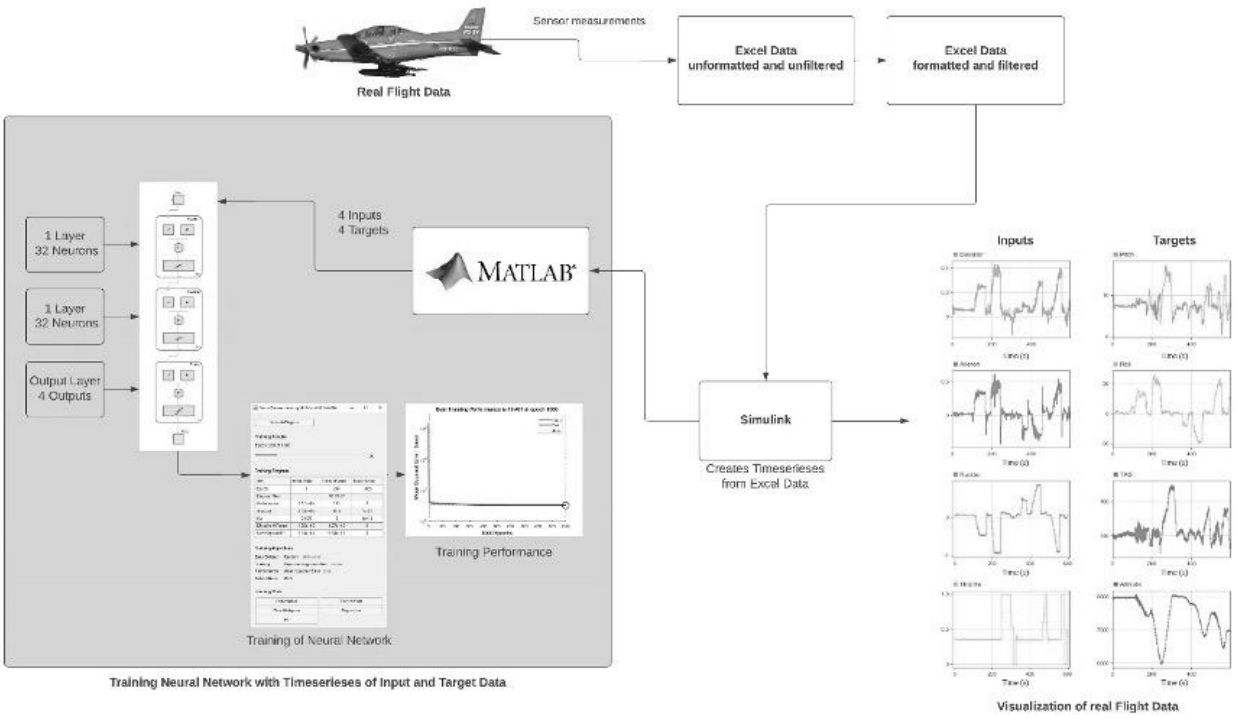

Figure 2: Data transfer procedure of real flight data

DOI: 10.13009/EUCASS2023-965

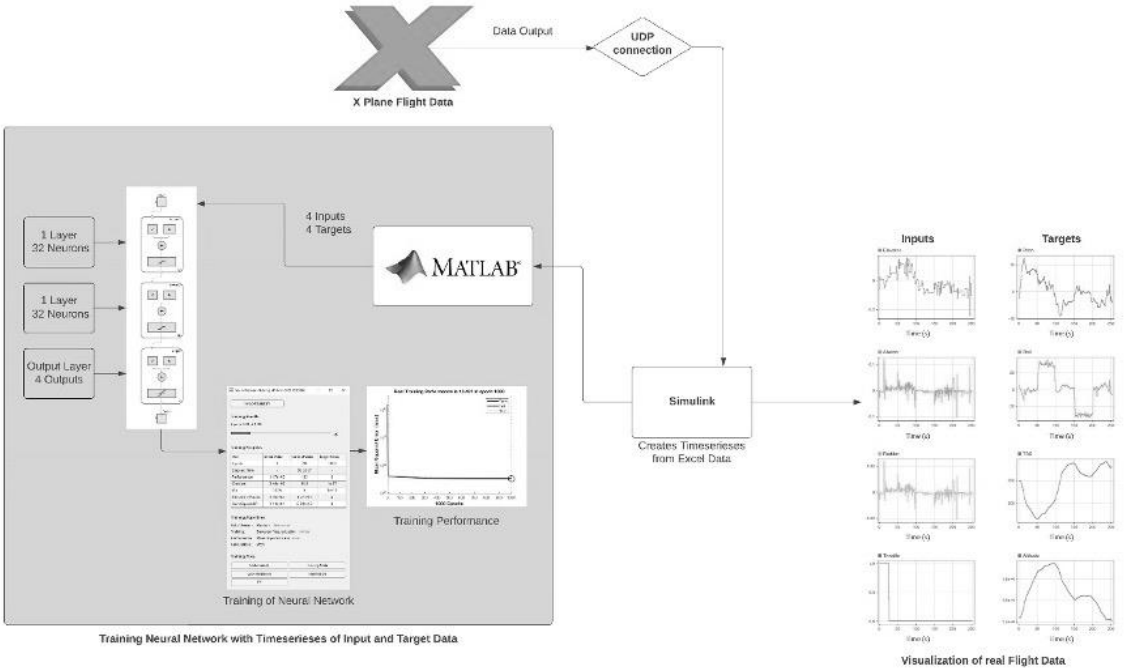

Figure 3: Data transfer procedure of simulated flight data (X-Plane).

5. Results of Digital Twin Framework

5.1 Comparison of Neural Network Structures

To create the graphs to be displayed and calculate the average accuracies of the predictions, the predicted flight data was saved separately to compare several data points with one another. Equation 1 was used to calculate the average accuracy of each prediction with n being the number of timesteps.

$$Accuracy = 100 * \frac{\sum\left(1 - \frac{\left(\sqrt{x_{pred} - x_{target}}\right)^2}{x_{target}}\right)}{n} \quad \textbf{(1)}$$

The overall accuracy of each network was calculated by taking the average of all accuracies of every individual prediction produced by each network, see Table 2 for full comparison details.

Table 2: Accuracy comparison of neural network structures

	Training	**Pitch**	**Roll**	**TAS**	**Altitude**	**Overall**
fitnet[32]	trainlm	50.42 %	19.72 %	96.08 %	69.80 %	54.94 %
fitnet[10]	trainbr	51.25 %	19.97 %	87.13 %	88.35 %	61.67 %
fitnet[10 10]	trainbr	46.97 %	23.24 %	97.24 %	87.35 %	63.67 %
fitnet[32]	trainbr	66.86 %	43.21 %	60.76 %	63.89 %	59.00 %

DOI: 10.13009/EUCASS2023-965

fitnet[32 32]	trainbr	64.17 %	25.11 %	96.22 %	86.92 %	68.10 %
fitnet[64]	trainbr	52.36 %	18.93 %	94.91 %	54.89 %	55.27 %
feedforwardnet[32]	trainlm	37.70 %	51.01 %	95.46 %	70.81 %	63.75 %
feedforwardnet[32]	trainbr	10.46 %	19.55 %	97.52 %	95.32 %	55.71 %
feedforwardnet[32 32]	trainbr	37.79 %	26.38 %	93.58 %	65.41 %	55.79 %
feedforwardnet[64]	trainbr	47.30 %	19.45 %	94.94 %	50.33 %	53.00 %

Due to the described comparison, a fitnet combined with the trainbr training function has been applied for further analysis of the remaining manoeuvres. Although the structure with 2 layers and 32 neurons each performed slightly better, because of limited computational resources, the network structure with 1 layer and 32 neurons has been selected for the training and testing of the flight data. The fitnet and trainbr combination using 1 layer of 10 neurons also performed slightly better in terms of average accuracy, due to a closer resemblance to the targets in its TAS and altitude predictions, but the network with 32 neurons has been chosen for further use based on the close resemblance to the targets in the pitch and roll angle predictions. It was clear that a simpler network structure usually performed better on the TAS and altitude predictions, while a slightly more complex structure performed better on the pitch and roll angle predictions, although a careful balance in complexity had to be maintained to provide consistent results.
Figure 4 displays one of the comparisons conducted, specifically of the fitnet with 1 layer and with 2 layers, each containing 32 neurons and applying the trainbr training function. The comparison shows the closer resemblance of the prediction by the fitnet with 1 layer in the pitch and roll data compared to the fitnet with 2 layers.

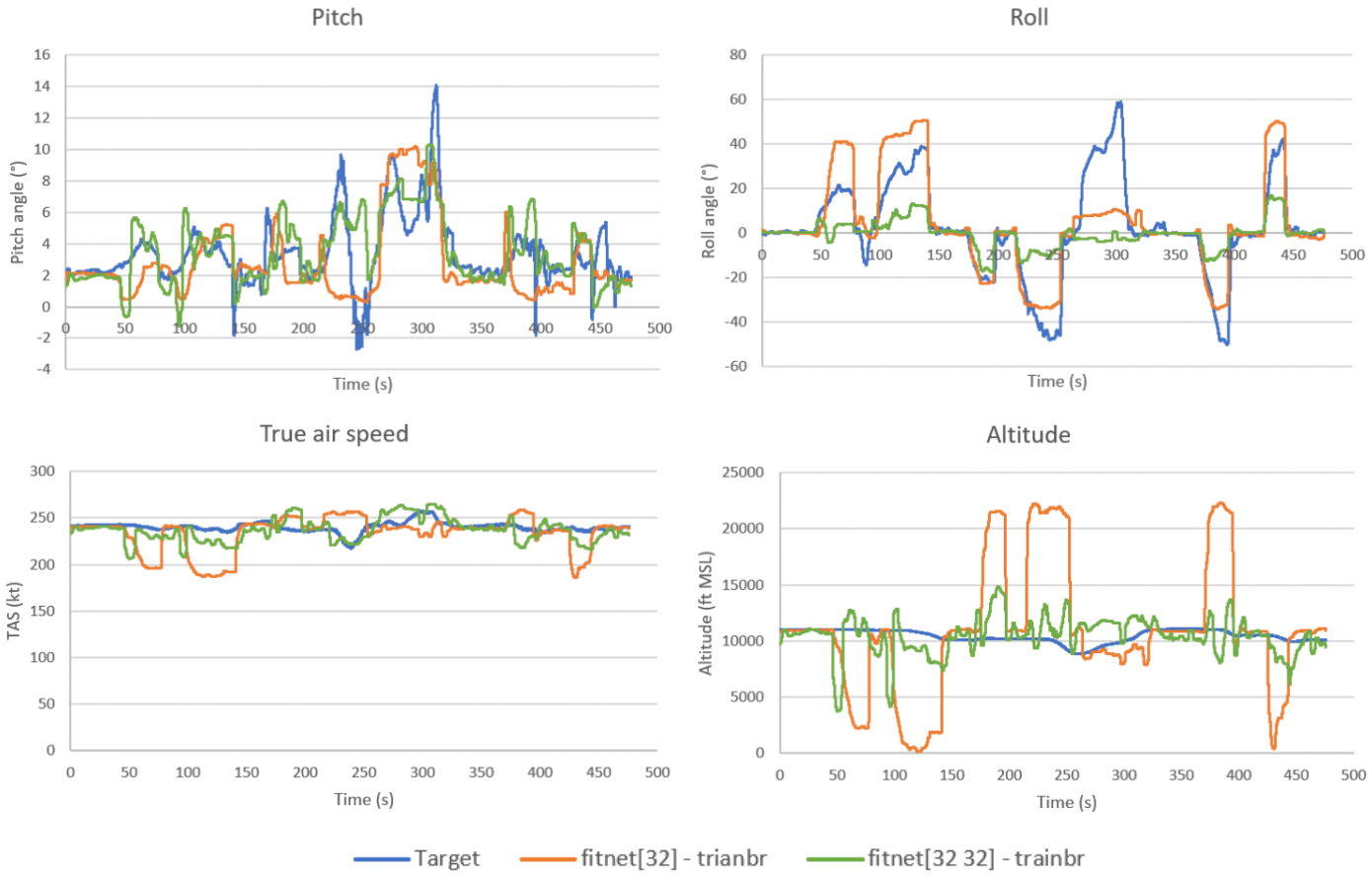

Figure 4: Predictions of two different neural networks

DOI: 10.13009/EUCASS2023-965

5.2 Prediction with Real Flight Data

The final testing of every manoeuvre from real flight data with at least one corresponding similar manoeuvre has been carried out and the resulting accuracies can be found in Table 3. The displayed comparison shows how the prediction accuracy varies strongly depending on the manoeuvre examined, due to their differences in complexity, duration, and extensiveness. The specific comparison of the Rudder Doublet manoeuvres, with one or two trained manoeuvres respectively, is displayed in Figure 5 and Figure 6.

Table 3: Prediction accuracies for real flight data

Trained manoeuvre	Predicted manoeuvre	Pitch	Roll	TAS	Altitude	Overall
AOT 200 kt	AOT 240 kt	66.87 %	15.50 %	97.95 %	99.08 %	69.81 %
Pitch Doublet 220 kt	Pitch Doublet 240 kt	28.02 %	32.03 %	99.43 %	99.29 %	64.79 %
Roll Perf 200 kt	Roll Perf 240 kt	63.64 %	03.56 %	96.24 %	93.59 %	64.26 %
Rudder Doublet 120 kt	Rudder Doublet 200kt	58.14 %	12.94 %	98.10 %	98.17 %	66.84 %
Rudder Doublet 120 kt	Rudder Doublet 240 kt	69.43 %	12.86 %	99.12 %	99. 23 %	70.16 %
Rudder Doublet 120 kt, Rudder Doublet 200 kt	Rudder Doublet 240 kt	71.44 %	23.26 %	99.73 %	99.82 %	73.56 %
SHSS 120 kt	SHSS 200kt	66.86 %	43.21 %	60.76 %	63.89 %	59.00 %
Steady Pulls	Sudden Pulls	11.59 %	11.23 %	93.30 %	91.56 %	51.93 %

Additionally, a Neural Network has been trained with all the available manoeuvres, only leaving out the same manoeuvres from the previous testing series, where a similar manoeuvre exists, to leave flight data remaining to test the network. Table 4 shows the accuracies of the predictions for each manoeuvre, together with the difference to the accuracy of a network that has only been trained specifically with the manoeuvre similar to the predicted manoeuvre from Table 3.

Table 4: Prediction accuracies with fully trained network

Predicted manoeuvre	Pitch	Roll	TAS	Altitude	Overall	Difference
AOT 240 kt	55.21 %	07.97 %	99.38 %	99.07 %	65.41 %	- 04.40 %
Pitch Doublet 240 kt	11.99 %	16.19 %	99.57 %	99.28 %	56.76 %	- 08.03 %
Roll Perf 240 kt	06.24 %	04.56 %	55.91 %	45.39 %	28.03 %	- 36.23 %
Rudder Doublet 200kt	12.83 %	23.13 %	78.37 %	96.51 %	52.71 %	- 14.13 %
Rudder Doublet 240 kt	60.26 %	10.28 %	98.26 %	98.33 %	66.78 %	- 03.38 %
SHSS 200kt	38.77 %	07.62 %	74.59 %	31.61 %	38.15 %	- 20.85 %
Sudden Pulls	05.07 %	07.12 %	30.98 %	13.63 %	14.20 %	- 37.73 %

The comparison shows an average loss in accuracy of 17.82%, proving that a specifically trained network outperforms a general network, with specific training of similar manoeuvres being the superior method of predicting the behaviour of the trainer aircraft in a certain condition.

DOI: 10.13009/EUCASS2023-965

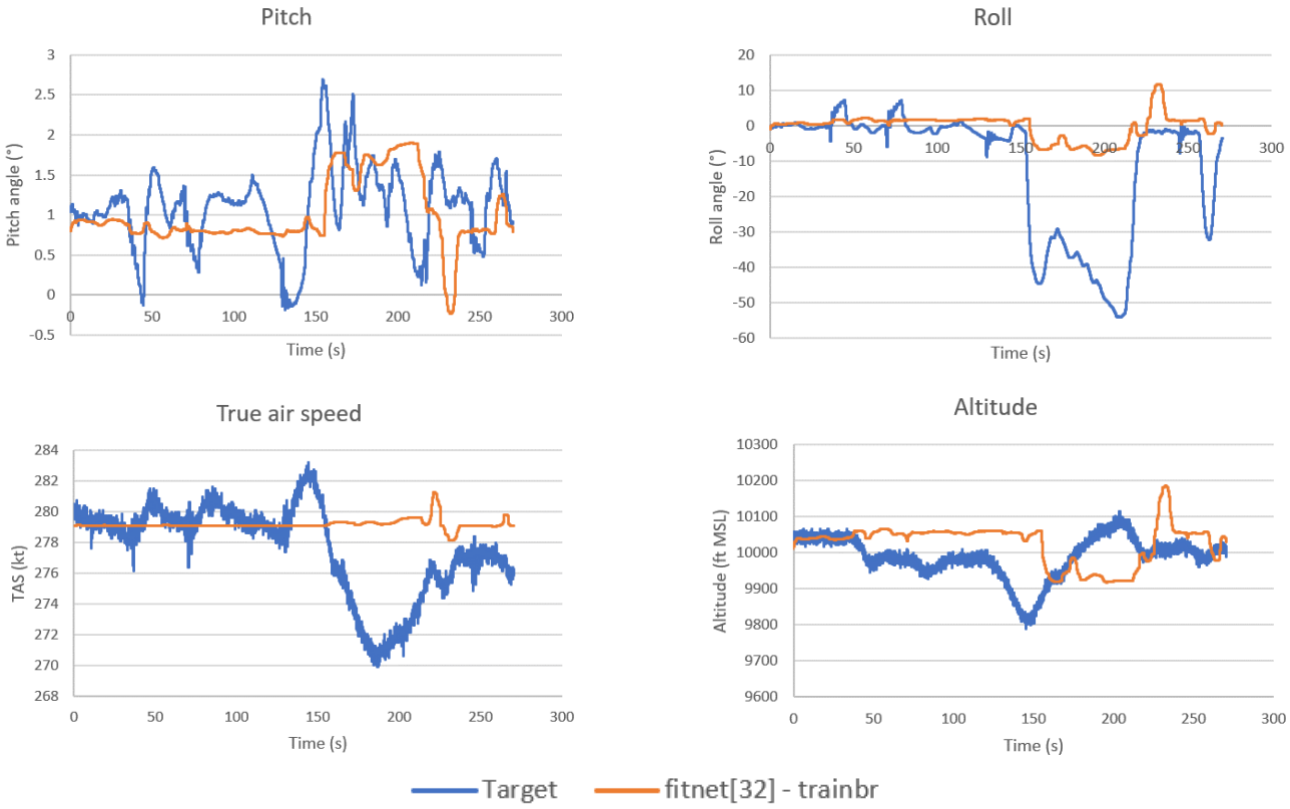

Figure 5: Predictions for Rudder Doublet 240 kt manoeuvre with one trained manoeuvre

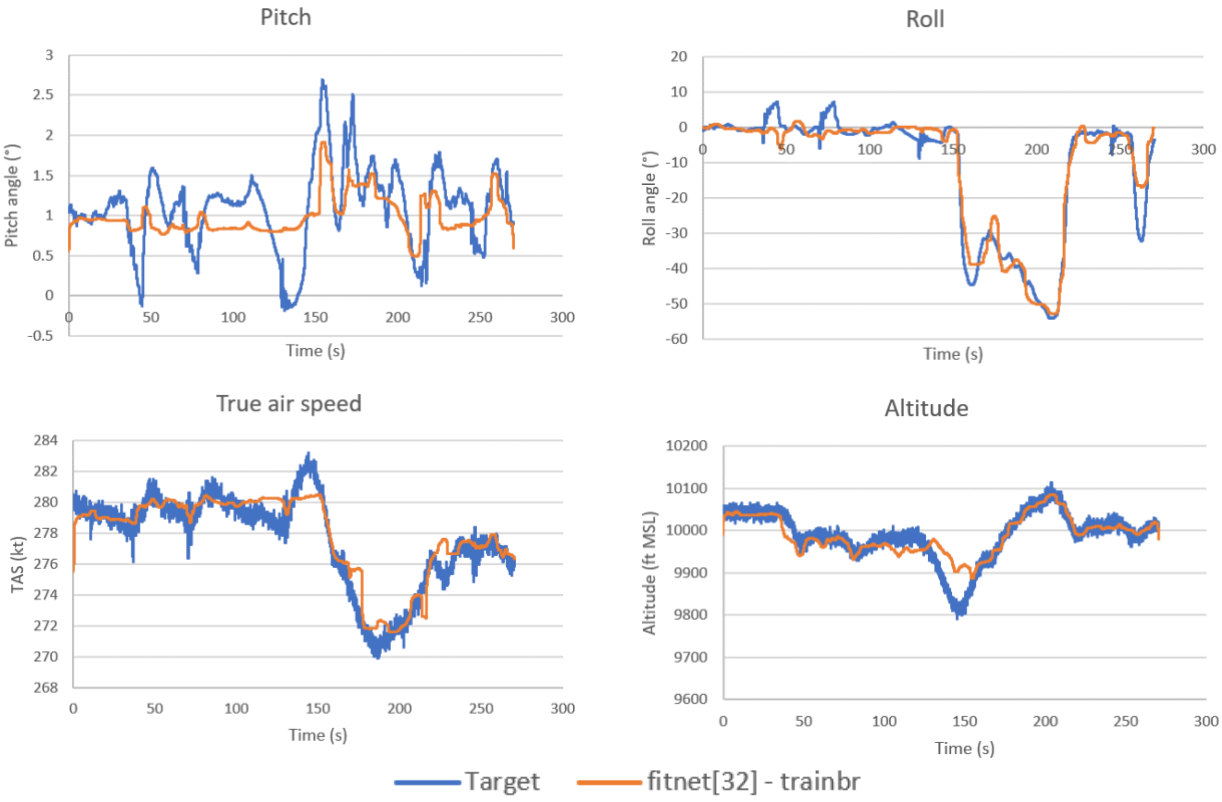

Figure 6: Predictions for Rudder Doublet 240 kt manoeuvre with two trained manoeuvres

DOI: 10.13009/EUCASS2023-965

5.3 Prediction with Simulated Flight Data

In addition to the testing of Neural Networks with real flight data, there has also been conducted testing with simulated flight data from X-Plane, accessed via UDP connection through Simulink. The purpose of the test displayed was to prove the feasibility of simulated flight data, expanding the usability of the DT framework, but also to test the accuracy of the predictions with a Neural Network solely trained with simulated flight data. For the testing 3 manoeuvres from the previously utilised real flight data have been executed in X-Plane, the selection of the manoeuvres was based on the feasibility of the execution of the manoeuvres in the flight simulator.

Table 5: Prediction accuracies for simulated flight data in X-Plane

Trained manoeuvre	Predicted manoeuvre	Pitch	Roll	TAS	Altitude	Overall	Difference
Pitch Doublet 220 kt	Pitch Doublet 240 kt	14.21 %	24.83 %	89.77 %	97.86 %	56.67 %	- 08.12 %
Steady Pulls	Sudden Pulls	18.08 %	12.27 %	92.09 %	75.11 %	49.39 %	- 02.54 %
Roll Perf 200 kt	Roll Perf 240 kt	06.26 %	04.26 %	64.08 %	11.11 %	21.43 %	- 42.83 %

The accuracies of the predictions by the Neural Network that has been trained with simulated flight data of the same manoeuvres show that the predictions based on simulated flight data performs worse than predictions based on real flight data, with an average loss in accuracy of 17.83 %. The comparison of the Sudden Pulls manoeuvre is displayed in Figure 7.

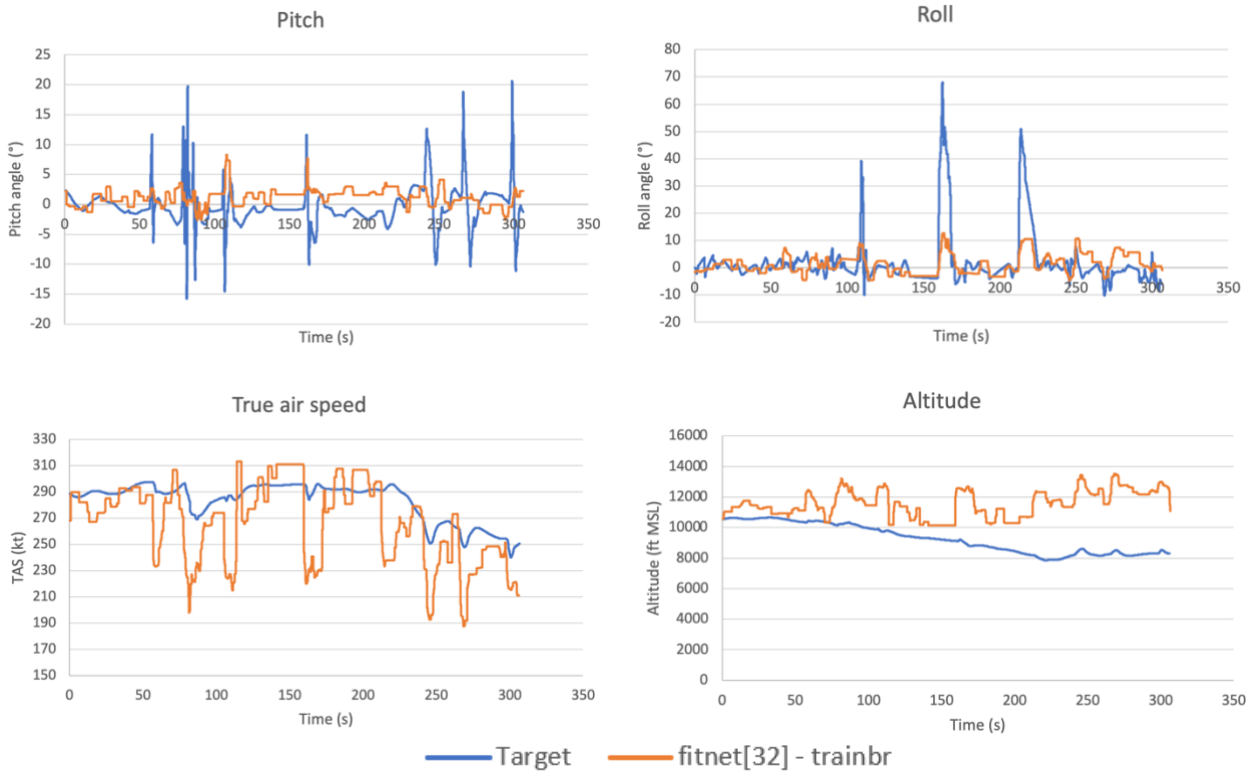

Figure 7: Predictions for Sudden Pulls manoeuvre with simulated flight data (X-Plane)

DOI: 10.13009/EUCASS2023-965

6. Conclusion

A simplified framework for a DT of a trainer aircraft has been proposed and validated in this paper. The testing of the DT framework with real flight data has shown satisfactory results in predictions for real flight manoeuvres, with an overall average accuracy of 65%, depending on the tested manoeuvre, and visually recognisable comprehensions of aircraft reactions by the Neural Networks with partially matching amplitudes in the predictions of the aircraft behaviour. The testing of simulated flight data, although not extensive, proved the feasibility of the DT framework with multiple data sources, which meets the aim proposed for the paper. The degradation in the accuracy of the predictions based on simulated flight data (from X-Plane) could be due to many factors, like less detailed timesteps in the data with a lower frequency in measurements in X-Plane, flown manoeuvres in the flight simulator not performed perfectly, or loss of data detail during data transfer when building the DT. Overall, the aims of the paper have been satisfied, with the inclusion of all manoeuvres provided by ETPS/QinetiQ and an extensive testing of predictions for similar manoeuvres. The DT framework can be trained with data from multiple sources, either formatted real flight data, or simulated flight data from X-Plane using the provided UDP connection.

The testing of the Rudder Doublet manoeuvres has shown an increase in accuracy of predictions with more training of similar manoeuvres. On the other hand, the testing of the fully trained network has shown that more training increases the accuracy of predictions only provided the manoeuvres are related to another, while training of different kinds of manoeuvres decreases the accuracy of predictions. This follows from the high dissimilarity of the tested manoeuvres, which brings disconnections to the neural network between the underlaying patterns, and the high complexity of flight dynamics in general. While the application of the trainbr training function in MATLAB reduces the risk of the network overfitting the training data, some of the lower accuracies in predictions could still result from data overfitting.

The proposed DT framework proved to be a promising contribution to the aerospace industry, providing the opportunity to train and predict different kinds of flight manoeuvres instead of specific conditions like the landing approach in previous research. The paper demonstrates the feasibility of using several data sources with Neural Network structures and provides a framework to expand on for further projects.

Future work

To expand on the DT framework proposed in this paper, several approaches could be taken. One option to expand the Neural Network structure would be to include additional signals from both real and simulated flight data, with their transfer already prepared in the DT framework, to widen the Neural Network structure and enhance the detail of the aircraft behaviour predicted. Additionally, the Neural Network structure itself could be altered with more extensive testing and more complex networks to determine an optimal Neural Network setting. Also, the Neural Network structure could be tested with much larger sets of flight data from different sources, expanding over several different manoeuvres and a varied difference or similarity between the manoeuvres.

References

[1] Leskovsky, R., Kucera, E., Haffner, O. & Rosinova, D., 2020. Proposal of Digital Twin Platform Based on 3D Rendering and IIoT Principles Using Virtual / Augmented Reality. *Cybernetics & Informatics (K&I)*, pp. 1-8.

[2] Bittar, A., Figureido, H. V., Guimaraes, P. A. & Mendes, A. C., 2014. Guidance Software-In-the-Loop simulation using X-Plane and Simulink for UAVs. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 993-1002.

[3] Grieves, M. W., 2019. Virtually Intelligent Product Systems: Digital and Physical Twins. In: *Complex Systems Engineering - Theory and Practice.* s.l.:American Institute of Aeronautics and Astronautics (AIAA), pp. 175-200.

[4] Tao, F., Wi, Q., Wang, L. & Nee, A. Y. C., 2019. Digital Twin: a review and prospect from a comprehensive perspective. *Journal of Cleaner Production*, Issue 213, pp. 669-686.

DOI: 10.13009/EUCASS2023-965

[5] Tao, F. et al., 2018. Digital Twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology,* Issue 94, pp. 3563-3576.

[6] Willcox, K. E., Lecerf, M. & Allaire, D., 2015. Methodology for Dynamic Data-Driven Online Flight Capability Estimation. *American Institute of Aeronautics and Astronautics (AIAA),* 53(10), pp. 1-15.

[7] Willcox, K. E. & Singh, V., 2017. Methodology for Path Planning with Dynamic Data-Driven Flight Capability Estimation. *American Institute of Aeronautics and Astronautics (AIAA),* 55(8), pp. 1-12.

[8] Niederer, S. A., Sacks, M. S., Girolami, M. & Willcox, K. E., 2021. Scaling digital twins from the artisanal to the industrial. *Nature Computional Science,* 1(5), pp. 313-320.

[9] Kapteyn, M. G., Knezevic, D. J. & Willcox, K. E., 2020. *Toward predictive digital twins via component-based reduced-order models and interpretable machine learning.* s.l., American Institute of Aeronautics and Astronautics (AIAA).

[10] Wilcox, K. E., Kapteyn, M. G. & Pretorius, J. V. R., 2021. A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale. *Nature Computational Science,* Issue 1, pp. 337-347.

[11] Stelmach, A., 2012. Neural Model of the Aircraft Landing Phase. *Archives of Transport,* 24(2), pp. 249-258.

[12] Entzinger, J. O. & Suzuki, S., 2010. Modeling of the visual approach to landing using neural networks and fuzzy supervisory control. *Aerospace Science and Technology,* 14(2), pp. 118-125.

[13] Yu, Y., Yao, H. & Liu, Y., 2019. Aircraft dynamics simulation using a novel physics-based learning method. *Aerospace Science and Technology,* Volume 87, pp. 254-264.

[14] Harris, J., Arthurs, F., Henrickson, J. V. & Valasek, J., 2016. *Aircraft system identification using artificial neural networks with flight test data.* s.l., International Conference on Unmanned Aircraft Systems (ICUAS), pp. 679-688.

[15] Roudbari, A. & Saghafi, F., 2013. An evolutionary optimizing approach to neural network architecture for improving identification and modeling of aircraft nonlinear dynamics. *Journal of Aerospace Engineering,* 228(12), pp. 2178-2191.

[16] Nagarajan, V. & Kolter, J. Z., 2019. *Generalization in Deep Networks: The Role of Distance from Initialization.* s.l., NeurIPS workshop on Deep Learning: Bridging Theory and Practice.

[17] Cocoma-Ortega, J. & Martinez-Carranza, J., 2019. *A CNN-based Drone Localisation Approach for Autonomous Drone Racing.* Madrid, 11th International Micro Air Vehicle Competition and Conference.

[18] Guo, X. et al., 2014. Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning. *Advances in Neural Information Processing Systems,* Volume 4, pp. 3338-3346.

[19] Wilson, A. G. & Izmailov, P., 2020. Bayesian Deep Learning and a Probabilistic Perspective of Model Construction. *International Conference on Neural Information Processing Systems,* Volume 394, pp. 4697-4708.

[20] Saha, S., 2018. A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way. *Towards Data Science.*

[21] Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning.* s.l.:MIT Press.

[22] Gurney, K., 1997. *An introduction to neural networks.* s.l.:UCL Press.