SYSTEM IDENTIFICATION OF MULTI-ROTOR UAV'S USING ECHO STATE NETWORKS

Aldo Vargas,[*] Murray Ireland,[†] and David Anderson[‡]

Controller design for aircraft with unusual configurations presents unique challenges, particularly in extracting valid mathematical models of the MRUAVs behaviour. System Identification is a collection of techniques for extracting an accurate mathematical model of a dynamic system from experimental input-output data. This can entail parameter identification only (known as grey-box modelling) or more generally full parameter/structural identification of the non-linear mapping (known as black-box). In this paper we propose a new method for black-box identification of the non-linear dynamic model of a small MRUAV using Echo State Networks (ESN), a novel approach to train Recurrent Neural Networks (RNN).

INTRODUCTION

There is a continuous growing interest in developing unmanned aerial vehicles (UAV) due to their ability to perform complex functions and assist human professionals in carrying out dangerous missions with on-board autonomous capabilities. Trends for developing UAV with advanced autonomous capabilities will continue for the foreseeable future. They offer major advantages when used for aerial surveillance, reconnaissance and inspection in complex and dangerous environments. Multirotor UAVs (MRUAV) are better suited for many civilian applications than manned aircraft and fixed-wing UAVs, particularly those that require vertical take-off and landing (VTOL) and hover capability. However, MRUAV are complex, non-linear and dynamically unstable systems that are difficult to control. The model of the MRUAV system, as in other engineering problems[1], is a crucial part of the analysis and design of controllers. These models are usually not well defined, because of existing uncertainties and non-modelled dynamics.

With the power of today's embedded systems processors, hybrid platform concepts (tilt-multirotor, stop rotor etc.) are now also being considered. Controller design for aircraft with unusual configurations presents unique challenges, particularly in extracting valid mathematical models of the MRUAV behaviour.

One of the most popular MRUAV platforms is the quadrotor. It has four fixed-pitch rotors with electric motors, arranged in a cross configuration. Dynamic models of quadrotors can be obtained through several techniques. Grey-box modelling involves measuring system properties and dynamic relationships through experimentation. In this way, it can be used to derive non-

[*] a.vargas.1@research.gla.ac.uk Aerospace Sciences Division, University of Glasgow, United Kingdom.
[†] murray.ireland@glasgow.ac.uk Aerospace Sciences Division, University of Glasgow, United Kingdom.
[‡] dave.anderson@glasgow.ac.uk Aerospace Sciences Division, University of Glasgow, United Kingdom.

linear models, however obtaining such parameters can be difficult and expensive with the required level of accuracy and precision. This is just one example of system identification.

System identification is a collection of techniques for extracting an accurate mathematical model of a dynamic system from experimental input-output data. This can range from parameter identification only (light-grey-box modelling) or to full parameter/structural identification of the non-linear mapping (known as black-box).

In this paper we propose a new method for black-box identification of the non-linear dynamic model of a small MRUAV using Echo State Networks (ESN), a novel approach to train Recurrent Neural Networks (RNN).

RNN have been an important focus of research and development since the 1990s. They are designed to learn sequential or time-varying patterns. A recurrent network is a neural network with feedback (closed-loop) connections[2]. Training a RNN is inherently difficult[3]. RNNs however, represent a very powerful generic tool, integrating both large dynamical memory and highly adaptable computational capabilities. They are the Machine Learning (ML) models most closely resembling biological brains, the substrate of natural intelligence.

When flying, for example, a quadrotor is known to be a strongly non-linear, time-varying and coupled system[4]. With this point of view, a dynamic reservoir (RNN) with echo states is proposed and used to obtain good system identification.

The paper is organized as follows. Section II provides an introduction to the quadrotor MRUAV and presents the data acquisition system. Section III describes the ESNs. Section IV provides the experimental methodology and results. Conclusions are presented in Section V.

II. QUADROTOR MRUAV

The quadrotor is a six degree-of-freedom (three translational and three rotational) system controlled by four independent inputs. Quadrotors are under-actuated because the system has a lower number of actuators than degrees of freedom. The resulting dynamics are highly non-linear, especially after accounting for complicated aerodynamic effects.

Figure 1. 3D printer quadrotor in flight

Figure 1 shows the vehicle being identified in this instance – a 3D-printed quadrotor built in-house in the Micro Air Systems Technologies (MAST) Laboratory. Although there are several platforms on the market, one of the advantages of building our own platform is economical: a MRUAV quadrotor similar to our own is the *Asctec Pelican*, which is 13 times as expensive as than the one produced in the MAST Lab.

The frame is 3D printed. This approach, also called additive manufacturing, is the process of making a three-dimensional solid object from a digital model. For this, a Makerbot Replicator 2 3D printer was used.

To track the corresponding position (X, Y, Z) and attitude (roll, pitch, yaw) of the vehicle we used the MAST Lab's motion capture (MoCap) system. This employs 18 MoCap cameras (Figure 2) for tracking a vehicle within a designated flight volume. The system uses passive markers, which are coated with a retro-reflective material to reflect infrared light that is emitted from the cameras.

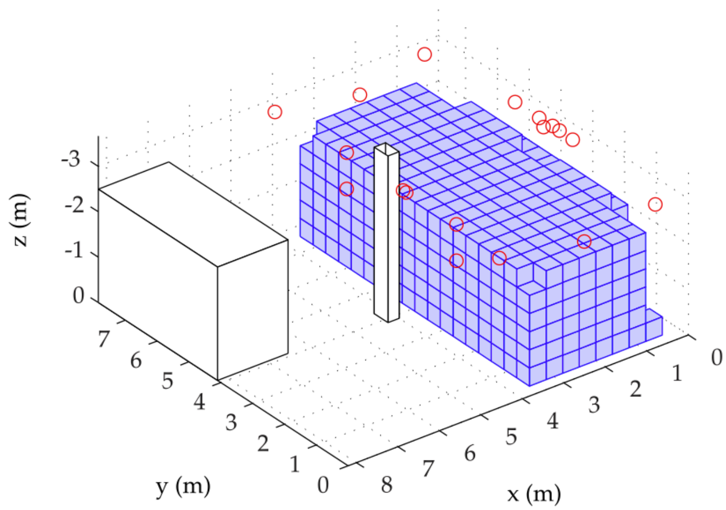

Figure 2. Distribution of cameras in the MAST Lab

The quadrotor is equipped with a flight computer (FC), which includes sensors such as accelerometers, gyroscopes, magnetometers and a barometric pressure sensor, which are used to provide the ability to control and stabilize the MRUAV. This computer calculates the attitude of the vehicle and computes the necessary pulse width modulation (PWM) values for each of the rotors to keep the vehicle level as well as receiving four desired pilot commands (throttle, roll, pitch and yaw). We are using the popular *MultiWii* platform. Due to the lack of processing power (CPU) of this FC, an extra computer is needed to record the inputs of the pilot, the position of the vehicle via the MoCap system and the outputs of the flight computer. This extra computer is called the Aircraft On-board Intelligence (AOI). Our current AOI is a Raspberry Pi.

The data recording flow is as follows: the AOI requests information (pilot commands and attitude) from the FC. At the same time, it receives the position and attitude transmitted from the ground station connected to the MoCap system (Figure 3).

The raw data must be pre-processed before being used to train the ESN due to the fact that the data acquisition process has many disturbances. These wild values have a great effect on the accuracy of the identification. Figure 4 shows the trajectory history of one of the flights in the MAST Lab.

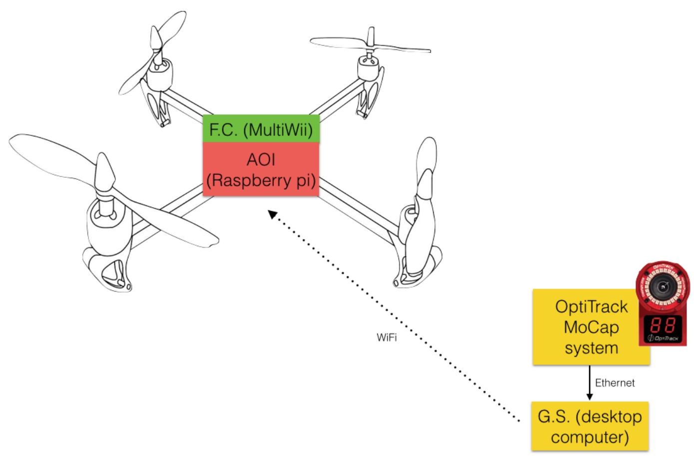

Figure 3. Data acquisition flow diagram

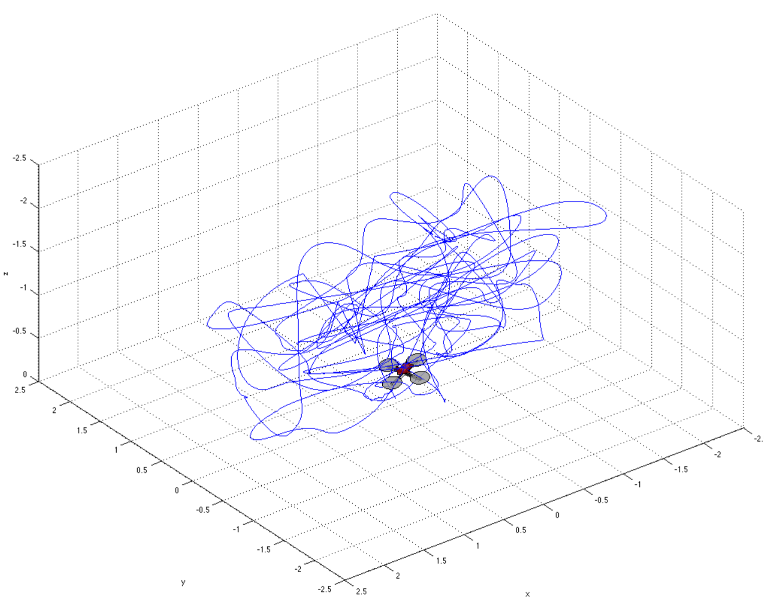

Figure 4. 3D plot of flight performed in the MAST Lab

III. ECHO STATE NETWORKS

Recurrent neural networks have been an important focus of research and development since the 1990s. They are designed to learn sequential or time-varying patterns. Training RNNs is inherently difficult. Nevertheless, they represent a very powerful generic tool, integrating both large dynamical memory and highly adaptable computational capabilities.

In order to overcome the downsides of traditional RNN training such as Back Propagation Through Time (BPTT) and Real Time Recurrent Learning (RTRL), a novel paradigm of computation with dynamical systems, namely Reservoir Computing (RC) has been proposed[5] which can be utilized to achieve efficient training of RNNs. Reservoir computing has emerged as an alternative to gradient descent methods for training recurrent neural networks. RC-based systems possess two parts: a recurrent non-linear layer, called the reservoir, and a linear readout output layer which is shown in Figure 5. The dashed connections are trainable while solid connections are fixed.

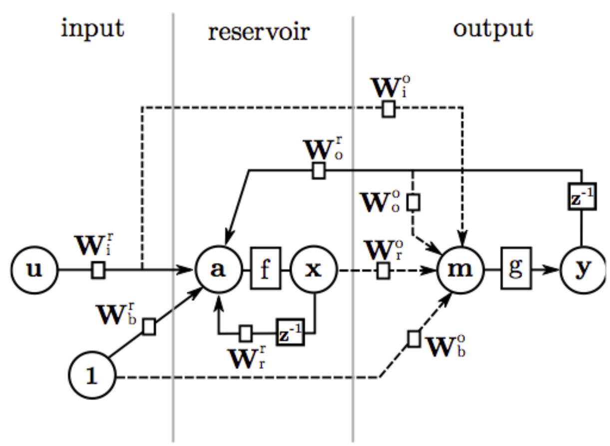

Figure 5. Reservoir computing mapping scheme

The main aspect of an RC network is that the recurrent connections of the reservoir are fixed, while the readout output weights only are trained. This characteristic simplifies much of the training of recurrent networks, as any standard classification or regression method can be used to train the output layer.

Echo State Networks are a “flavour” of RC[6]. The main idea comes from a continuous neural hardware micro-circuitry. ESNs have the advantage of overcoming the difficulties of traditional dynamic RNN in large-scale training. They can also approximate non-linear systems precisely producing excellent results in their predictions. The ESN is practical and conceptually simple, but requires some experience and insight to achieve good performance. An ESN is composed of a discrete hyperbolic-tangent RNN, the dynamic reservoir, and of a linear readout output layer, which maps the reservoir states to the actual output (Figure 5).

ESNs are called in this way because the dynamic reservoir contains *echo states*, a property of the network prior to training. ESNs are composed of a discrete hyperbolic-tangent RNN, the dynamic reservoir, and a linear readout output layer, which maps the reservoir states to the actual output.

ESNs are applied to supervised temporal machine learning tasks where, for a given training input signal $u(n)$ of n dimensions, $x(n)$ is the n-dimensional reservoir activation state and $y(n)$ is the n-dimensional output vector, or desired target output signal. ESNs use an RNN type with leaky-integrated discrete-time continuous-value units. The typical update equations are

$$x(n+1) = \tanh(W_r^r x(n) + W_i^r u(n) + W_o^r y(n) + W_b^r) \quad (1)$$

where tanh is the hyperbolic tangent activation function, commonly used for ESNs. The output is computed as

$$\mathrm{y(n+1) = g(W_r^o x(n+1) + W_i^o u(n) + W_o^o y(n) + W_b^r)} \quad (2)$$

$$y \ = g(W^{out}\mathrm{z(n+1)})$$

where g is a post-processing activation function, W^{out} is the concatenation of W_r^o, W_i^o, W_o^o and W_b^o and z is the previous input / output and a bias term.

All weight matrices representing the connections to the reservoir, denoted as $\mathrm{W^r}$, are initialised randomly (represented inside input part in Figure 5) while all connections to the output layer, denoted as $\mathrm{W^o}$ are trained (represented by dashed arrows in Figure 2). Figure 2 is a schematic, which shows the connections and the respective mappings given by the matrices W in equation 1 and 2.

Training of an ESN

There are two basic classes of learning, supervised and unsupervised training we focus on supervised training. In supervised training, we start with *teacher data* (or training data), which represents examples of the desired model behaviour.

What is desired is a trained ESN (W^{in}, W, W^{back}, W^{out}) whose output $y(n)$ approximates the teacher output $y_{\mathrm{target}}(n)$, when the ESN is driven by the training input $u(n)$.

The original method[7] for training ESN is to:

I. Generate a large random dynamic reservoir RNN (W^{out}, W, α)

II. Normalize W^{out} to a matrix with unit spectral radius α

III. Run it using the training input $W_i^o u(n)$ and collect the corresponding reservoir activation states $x(n)$

IV. Compute the linear readout weights W^{out} from the reservoir using linear regression, optimizing the mean square error of the network output w.r.t. the training target signal $y_{\mathrm{target}}(n)$

V. Use the trained network on new input data $u(n)$ by computing $y(n)$ employing the trained output weights W^{out}

For this approach to work is important that the dynamic reservoir (DR) possesses the echo state property, that is, for every internal signal $x_i(n)$ there exists an echo function which maps the input and the output histories to the current state. Usually to ensure the echo state property the recurrent connection weights W must be appropriately scaled. The spectral radius α of the DR is of crucial importance for the eventual success of the ESN. A small spectral radius makes the DR behave quickly, and a larger spectral radius means the DR will be slow.

The task of black-box modelling for a system, like the MRUAV, involves finding a good approximation to the system function. The network output of the trained network is a linear combination of the network states, which are ruled by the echo states, so the approximation of the system function can be interpreted as a linear combination of echo functions. In order words, the black-box model of the highly complex non-linear and dynamically unstable MRUAV is a linear combination of the echo state functions.

IV. EXPERIMENTAL PROCEDURE

Several flights were performed in the MAST Lab to ensure we had sufficient data to feed into the network. As usual in machine learning methodology we obtain several data sets and leave them aside while building our models, thus we are left with training data sets and testing data sets from a number of different flights. The black-box model derived employs the pilot commands as input and position/attitude as outputs, similar to the model used in our previous work[8]. As output position and the attitude of the vehicle were used.

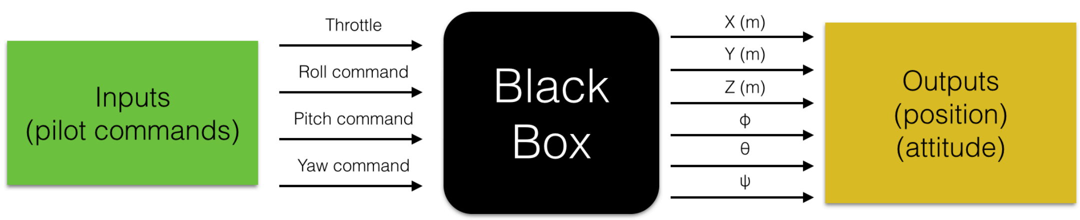

Figure 6. Black Box system model

The pilot commands were designed to keep the vehicle in motion, avoiding hovering in the same position as much as possible and not following any particular trajectory. This was due to the fact that we required sufficient excitation in the dynamics for the reservoir and thus we needed fast target signals. This was to ensure the success of the modelling task[4].

After obtaining the data from several flights, either for training or testing, we proceed with training the ESN. The mean square error (MSE) is calculated to gauge performance of the ESN training and for the purposes of tuning the parameters.

Results

It's important to notice that the training data is different than the testing data. This is to ensure the ESN had captured the dynamics of the model. The MSE of the testing data is used as proof of such an end. The training is done in two stages: sampling and weight computation.

During the sampling stage, the teacher signal (training data) is written into the output units, this technique is often called *teacher forcing*. We now compute W_i^o for the linear output units such that the teacher time series is approximated as a linear combination of the internal activation time series $x_i(n)$.

The first training was made with a dataset of a flight that lasted approximately 5 minutes, the data acquisition system recorded all data at approximately 100 Hz. The parameters used in this experiment where a DR size of 100, random percentage noise of standard deviation 1×10^{-6} and a spectral radius of 0.1. This parameters where manually chosen, one at a time. The error produced at the training stage was 0.00396757 while the testing MSE (using different data) produced a value of 0.010322. The output units of the ESN versus the real flight data are shown in Figure 7.

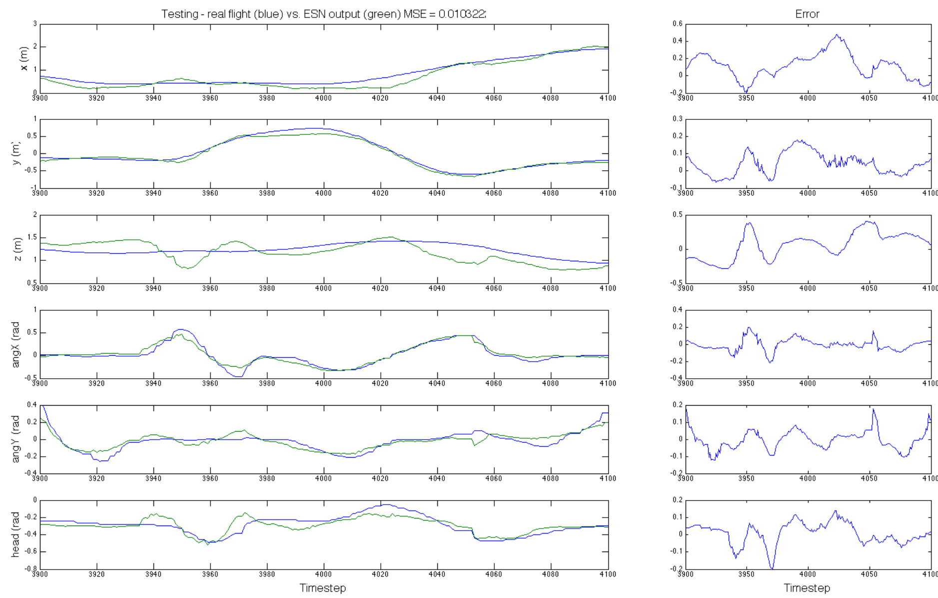

Figure 7. Output of ESN vs. real data

As stated before, the parameters of the first experiment where chosen manually, which in a machine learning approach is unavoidable to some extent. The next step was to optimise some of the parameters of the ESN in order to reduce the MSE produced at the testing stage.

In order to optimise these parameters we chose an evolutionary algorithm called CMA-ES (Covariance Matrix Adaptation Evolution Strategy), which is a state-of-the-art method in evolutionary continuous parameter optimisation[9], it was also proven to be an effective method due to its flexibility[10]. The implantation of the CMA-ES algorithm used in this research closely follows the Hansen[11] algorithm.

The CMA-ES algorithm is considered to be almost parameter-free. Only the number of offspring ensures the success of evolution. The only parameters that we establish to evolve are: spectral radius, DR size and noise added. The performance in our experiment is shown in Figure 8. It is important to note that we had setup the CMA-ES to ensure that it does not break the rules of ESN, that is to preserve the echo states.

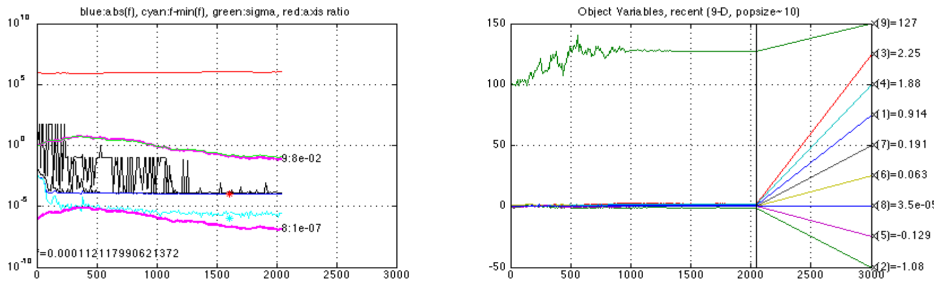

Figure 8. CMA-ES performance

After approximately 2000 runs, the optimized parameters end up being: a DR size of 127, random noise of standard deviation 3.4883×10^{-5} and a spectral radius of 0.9191. This parameter produced a MSE of 1.1185×10^{-4} using unknown data to the ESN (testing). The results can be seen in Figure 9. The difference between signals is almost imperceptible.

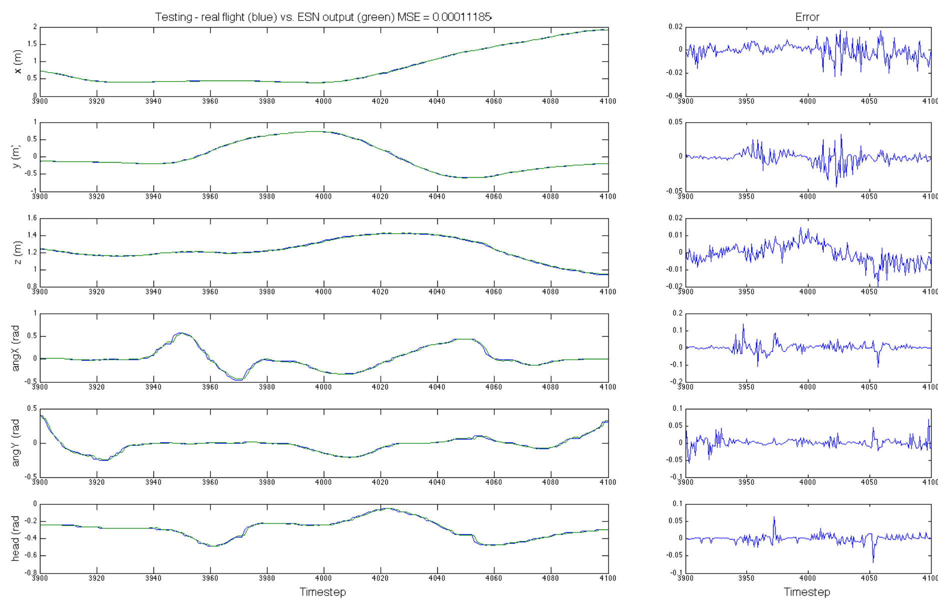

Figure 9. Output of optimized ESN vs. real data

V. CONCLUSIONS

In this paper, we have presented a new way to use ESN in a practical way, with real flight data for performing black-box system identification. Using real flight input and output information, the ESN can identify the system dynamics and produce a black-box model that can be used for creating new state-of-the-art controllers, trajectory tracking algorithms or even optimise current waypoint following position controllers.

Figure 10 shows three quadrotors following the same trajectory (offset in X and Y for discrimination). The blue path corresponds to the actual recorded data from the original flights (the testing data), while the other two trajectories correspond to the ESN output to the pilot inputs of the same flight. If our classifier has succeeded, the trajectories must be equal. The red path corresponds to the first un-optimised ESN proposed in this paper with a MSE error of 0.010322, while the black path is the final optimised ESN with a MSE of 1.1185×10^{-4}. These results show that, although there exist errors between the real flight data and the ESN model output, the identified model using the ESN have an acceptable accuracy and can reflect the trend of the quadrotor.

The improvement of the optimised ESN output is clearly visible, therefore we can state that the ESN has *understood* the full non-linear dynamics of the quadrotor MRUAV.

The evolutionary strategy CMA-ES helps us improve the parameters of the ESN, decreasing the error by 99.2%, and it also highlights that the optimal spectral radius for our application must be greater than stated at the beginning of the research. The task of identifying a quadrotor MRUAV therefore requires a longer memory of the input when using echo state networks.

Different MRUAV configurations and sizes flight data are needed to further improve the algorithm and make it a more reliable method of black-box identification.

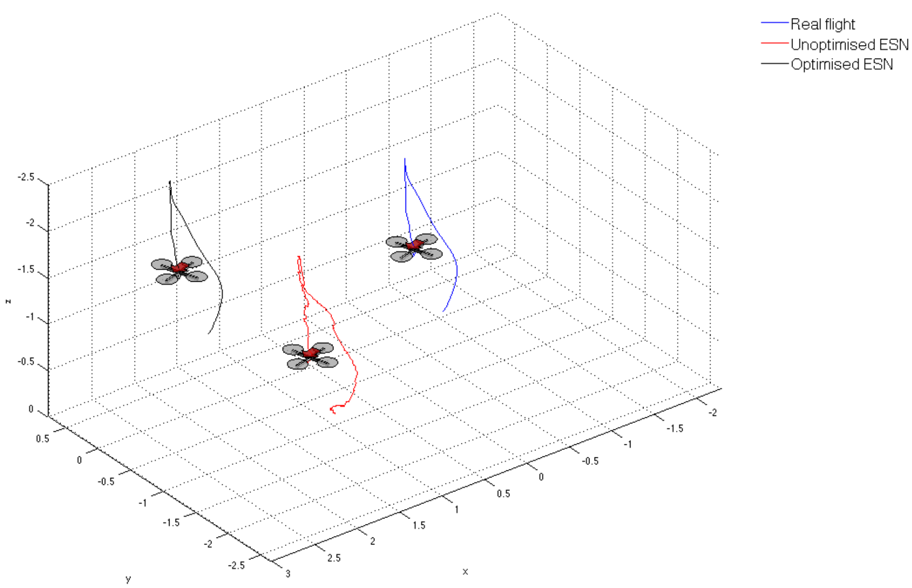

Figure 10. Trajectory comparison

NOMENCLATURE

u	Input signal
y	Output signal
x	Reservoir state
a	Weighted sum for reservoir units
m	Weighted sum for output units
W_i^r	Input to reservoir connection matrix
W_b^r	Bias to reservoir connection matrix
W_r^r	Reservoir connection matrix
W_o^r	Output to reservoir connection matrix
W_i^o	Input to reservoir connection matrix
W_r^o	Reservoir to output connection matrix
W_o^o	Output to output connection matrix
W_b^o	Bias to output connection matrix

REFERENCES

[1] Marcos González-Olvera, Yu Tang. *Black-Box Identification of a Class of Nonlinear Systems by a Recurrent Neurofuzzy Network,* 2010.

[2] Laurene V. Fausett. *Fundamentals of Neural Networks,* 1993.

[3] Mantas Lukosevicius. *A practical Guide to applying Echo State Networks*, 2012.

[4] A. Das, K. Subbarao, F. Lewis. *Dynamic inversion with zero-dynamics stabilisation for quadrotor control*, 2008.

[5] Verstraeten D, B. Schrauwen, M. D'Haene, D. Stroobandt. *An experimental unification of reservoir computing methods,* 2007.

[6] Mantas Lukosevicius, Herbert Jaeger. *Reservoir computing approaches to recurrent neural network training*, 2008.

[7] Herbert Jaeger. *The "echo state" approach to analyzing and training recurrent neural networks,* 2001.

[8] Vargas, Aldo, Ireland, Murray and Anderson David. *Swing-free manoeuvre controller for rotorcraft unmanned aerial vehicle slung-load system using echo state networks,* 2015.

[9] N. Hansen, A. Ostermeir. *Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation.* 1996.

[10] Fei Jiang, Hugues Berry, Marc Schoenauer. *Supervised and Evolutionary Learning of Echo State Networks.* 2008.

[11] Hansen N. The CMA Evolution Strategy: A comparing review. 2006