Stachiw *et al. Advanced Modeling and Simulation in Engineering Sciences(2022)9:13*
https://doi.org/10.1186/s40323-022-00227-7

Advanced Modeling and Simulation in Engineering Sciences

RESEARCH ARTICLE Open Access

A physics-based neural network for flight dynamics modelling and simulation

Terrin Stachiw, Alexander Crain and Joseph Ricciardi

*Correspondence: alexander.crain@nrc-cnrc.gc.ca
Flight Research Laboratory National Research Council Canada, 1200 Montreal Rd, Bldg U-61, K1A 0R6 Ottawa, Canada

Abstract

The authors have developed a novel physics-based nonlinear autoregressive exogeneous neural network model architecture for flight modelling across the entire flight envelope, called *FlyNet*. When using traditional parameter estimation and output-error methods, aircraft models are captured about a single point in the flight envelope using a first-order Taylor series to approximate forces and moments. To enable analysis throughout the entire operational envelope, the traditional models can be extended by interpolating or stitching between a number of these single-condition models. This paper completes the evolutionary next step in aircraft modelling to consider all second-order Taylor series terms instead of a subset of those and by exploiting the ability of neural networks to capture more complex and nonlinear behaviour for the efficient development of a continuous flight simulation model valid across the entire envelope. This method is valid for fixed- and rotary-wing aircraft. The behaviour of a conventional model is compared to FlyNet using flight test data collected from the National Research Council of Canada's Bell 412HP in forward flight.

Keywords: Flight simulation, Modelling and simulation, Aerospace

Introduction

Flight modelling and simulation plays a major role in the life of an aircraft. From the aircraft development to assess the aircraft stability and develop controls, to pilot training, flight models must provide a realistic representation of aircraft behaviour throughout the entire flight envelope. This includes the aircraft's possible range of speeds, altitudes, angles of attack and sideslip, and aircraft configurations, such as the flap position and centre of gravity (CG) location. A flight simulator model that spans the operational space of an aircraft is known as a *global model* or a *stitch model*.

Neural networks (NN) are known for their ability to accurately capture the complex and nonlinear behaviour of systems by exploiting efficient optimization algorithms. Accordingly, NNs have been studied for applications in flight modelling, both about a single flight condition and across the entire envelope. Since the development of a flight simulation model is typically a lengthy process requiring manual interference and expert physical insight, this paper describes a method that employs neural networks for the efficient development of an aircraft global model. This paper seeks to take an evolutionary step from the

traditional methods by using an approach that is still rooted in physics but instead uses a neural network to capture the complex behaviour of the forces and moments through the flight envelope. This approach would significantly reduce the difficulty of developing a global model, thereby opening it up to more end-users, as well as decreasing model development time. This approach is applicable to both fixed- and rotary-wing aircraft and is demonstrated using flight test data collected from the National Research Council of Canada's (NRC) Bell 412HP.

Flight simulation and global modelling

In a typical flight modelling process, several aerodynamic models are determined across a range of speeds, altitudes, and aircraft configurations by assuming small perturbations about the trim condition. The model is linear and called a *point model* or an *anchor point* [1], which is linearized about the trim condition of the flight test point. The point model may be estimated using time-domain or frequency-domain system identification techniques, which are presented in the popular textbooks by Klein and Morelli [2] or Tischler and Remple [3].

A global model is a connection of the point models through some regression technique. The regressors typically include some measure of the forward speed, air density, and other parameters that vary between fixed- and rotary-wing aircraft. These regressors may be selected using stepwise regression, expert insight, or another method for model structure determination. A global model is a combination of linear models that vary throughout the flight envelope so such a model may be referred to as *quasi-non-linear*. When simulating using a global model, the parameter values at the instantaneous state are used [1], essentially turning the global model simulation into a non-linear one; because the point model uses the trim condition instead of the instantaneous one, this creates a disconnect between the point model and global model simulation methods.

Brandon and Morelli [4] indicate that a variety of techniques may be used to develop global models, including stepwise regression with splines or model stitching [2,5–18] or multivariate B-splines [19,20], stepwise regression with polynomial regression [21–26], multivariate orthogonal functions [2,27–31], fuzzy logic [32,33], data partitioning with simplified local models [2,34,35], and combining local models [36,37]. These techniques fall into two general approaches: model structure determination with curve fitting and model partitioning.

Model structure determination with curve fitting

The approach of model structure determination followed by curve fitting is the one most often seen in practice. In this approach, the structure of the model is dictated by a set of independent variables that span the flight envelope, called the *regressors*. The regressors may be selected from the aircraft states and controls using convention, expert insight, stepwise regression, or orthogonal functions. Since several aircraft states and controls are correlated, such as the angle of attack and the elevator position, a set of orthogonal functions removes any correlation between regressors [27].

A curve fitting method may be applied for regression of the stability and control derivatives. The use of polynomial regression allows for a smooth function that may be a function of several variables. This may serve to smooth out the effects of over-specialized point

models given a sufficiently low-ordered polynomial or a sufficiently high number of point models. This approach begins to break down for highly non-linear behaviour, such as that which may occur near stall or during aircraft upsets. This may be addressed by adding in more terms or higher-ordered terms at the risk of generating a high variance model. Alternatively, splines may be used or the model may be partitioned, for example, to low angle of attack and high angle of attack models.

Splines allow one to fit a lower-ordered polynomial through a subset of points. Tobias and Tischler [1] provide a review of the spline method, which is core to the *stitching* technique. This model stitching approach was first proposed by Aiken [5] and Tischler [6], and is commonly seen in practice for frequency-domain system identification models with a small number of anchor points [9–17]. Depending on the order of the chosen interpolating polynomial, various levels of continuity may be enforced. A linear interpolating function permits C^0 continuity, and a cubic interpolating function provides C^1 continuity with clamped endpoints for extrapolation. A drawback of this method is the computational complexity when considering more than two regressors [27]. Further, there is no basis for determining the model structure so the spline may match noise and is unsuitable for a scattered dataset [27]. Accordingly, the spline method is typically applied to models identified in the frequency-domain since spectral smoothing techniques allow the combination of several points about a flight condition to form a single frequency response for identification. A more recent publication from Millidere et al. [38] proposes the Lasso technique as yet another means of determining the global model regressors. The downside of this method when compared to the proposed method is that it depends on first determining local linear models through Jacobian Linearisation. The Lasso technique is then used to stitch the individual models.

Model partitioning

The model partitioning approach is less commonly seen in literature. This approach separates the data into bins and identifies a model for each bin. In the model partitioning with simplified local models approach [2,34,35], there is a discrete number of models and only one model is considered at a time. This is essentially a nearest-neighbour interpolation and may result in discontinuities when transitioning between bins. These discontinuities may be addressed through combining local models where models are combined using model superposition and Gaussian weighting [36,37].

The model partitioning approach has several limitations compared to other methods. Each partition must have a sufficient number of data points to fit a model. This requirement generally makes it infeasible to consider more than one regressor. Although the models in each bin are generally linear and each bin is relatively small, the model partitioning approach can be combined with one of the previous curve-fitting approaches. This technique generally uses larger bin sizes and may, for example, separate the model to subsonic, transonic, and supersonic bins; high angle of attack and low angle of attack bins for fixed-wing aircraft; or hover and forward flight bins for rotary-wing aircraft. The simplicity and computational efficiency of the model partitioning approach have attractiveness for real-time global model identification [39].

Aircraft trim

Not only must the aircraft model match the aircraft dynamics, but it must also have similar static behaviour, which is represented by the trim solution. A trim solution refers to the control settings and initial conditions that result in zero net forces and moments applied to the aircraft such that there is no rectilinear or angular acceleration in any axis. The establishment of a trim solution is important for cases such as evaluating performance in steady flight and for evaluating the handling qualities where a linear model is established about the initial point. For most analyses, it is desirable to begin at a specific flight condition instead of beginning on the ground, taking off, then flying to the desired condition [40].

The ability of the simulator to find a trim solution to begin the simulation may become a challenge for global modelling. In general, point models are tabulated for the trim solution and a line is fit through each of the point models. A simulator will then refer to the saved trim point for a particular test when beginning a simulation. However, the inevitable errors that arise in the regression process result in discrepancies between global and local model parameters that will likely result in the test beginning out of a trimmed state. The ease in which an optimization method can find small offsets to the desired initial states and control positions such that the simulation begins in trim is referred to herein as the *trimmability* of the model. The Code of Federal Regulations Title 14, Part 60 outlines the trim requirements for flight simulators. For example, Appendix C to Part 60 states that a helicopter full flight simulation model must be in a trim condition when the pitch angle is within $\pm$1.5deg, the sideslip angle is within $\pm$2deg and all control positions are within 5% of the actual values from flight testing to be compliant with the regulations [41].

Neural networks

NNs are known for their potential to capture the nonlinear behaviour of systems accurately and efficiently. Since aircraft exhibit nonlinear behaviour across their flight envelope, NNs have seen successful application for aircraft flight modelling. A NN for this application is typically in the form of a nonlinear autoregressive exogeneous (NARX) model [42–47]. Such a model is represented in the following algebraic form:

$$y_{\langle t\rangle} = F\left(y_{\langle t-1\rangle}, y_{\langle t-2\rangle}, \ldots, u_{\langle t\rangle}, u_{\langle t-1\rangle}, u_{\langle t-2\rangle}, \ldots\right) + \varepsilon \tag{1}$$

where $y_{\langle t\rangle}$ is the output state vector at timestep t, $u_{\langle t\rangle}$ is the input control vector at timestep t, and ε is an error term. Thus, the output at a certain time step is conditioned by the inputs and states at previous timesteps. Instead of directly passing the inputs and states from previous timesteps, the model may also be captured in the form of a recurrent neural network (RNN) [48,49]. An RNN takes the activation values at the previous timestep and combines these with the inputs at the current timestep.

The NNs in literature often model output state derivatives such that they are analogous to a linearized state-space representation at a single flight condition. An alternate approach is to model the force and moment coefficients at a given timestep [50,51]. When the NN outputs force and moment coefficients, the corresponding derivatives at a given condition can be estimated by a method such as the "modified delta" approach used by Chauhan and Singh [50].

The NN architectures in literature typically consist of three total layers (input layer, one hidden layer, and output layer), where the hidden layer may use a rectified linear

unit (ReLU), as in Punjani [44], or a smooth nonlinear activation function, such as a sigmoid function, as in Norouzi et al. [47], or a hyperbolic tangent, as in Fekih et al. [43]. The number of neurons in the hidden layer ranges from 10, as in Norouzi et al. [47], to 2500, as in Punjani [44]. Since flight simulation systems must operate in real-time or faster, simpler networks are preferred, thus explaining use of a single hidden layer and a preference toward a small number of neurons.

Roudbari and Saghafi demonstrated the use of a neural network with a multidimensional output for modelling across the complete flight envelope [48]. This neural network architecture simultaneously outputs the states in multiple dimensions for a range of altitudes and Mach numbers from which the value at the current state can be interpolated. Since the model results are output in a multidimensional matrix, the model is computationally intensive and concurrently predicts the outputs at discrete values in the flight envelope, which may be non-smooth between states. Further, the model grows exponentially in complexity if additional parameters are considered. This motivates the work presented herein where the NN model outputs results in a single dimension while maintaining smooth behaviour throughout the flight envelope.

The approach described in this paper assumes a sufficiently deep neural network can adequately model the forces and moments acting on the aircraft. However, transitioning between flight regimes, such as hover, forward flight, or autorotation, may result in a non-smooth force or moment function that may require a complex neural network to adequately model. Alternative approaches as described by Jacquemin et al. [52], such as finite difference or point colocationmay, may simplify the potentially non-smooth function. Future work will investigate full flight envelope modelling including hover and autorotation.

There are also several studies in literature not directly related to flight modelling that are of interest. Deshpande et al. [53] proposed a deep neural network architecture trained with finite element method generated force-displacement data. The main benefit of their method is that it is able to accurately predict large deformations in real-time, which is not possible when using classical finite element methods. This method clearly demonstrates how deep neural networks can be used to develop computationally simpler models that retain the predictive capabilities of the full model.

Physics-based neural networks

NNs, though powerful, are understood to be purely interpolative. The potential complexity of NNs with one or more layers consisting of many neurons requires that the function's domain be restricted to that defined by the training data since the behaviour outside of that domain is undefined. Recently, the push in machine learning and modelling with NNs is toward physics-based or physics-informed models. Such a model may have any of the following three properties:

1. Uses a physics-based dataset;
2. Uses physics-informed training constraints; or
3. Uses a physics-guided algorithm.

In the context of aircraft flight simulation, the dataset may consist of outputting forces and moments rather than state derivatives. Physics-informed training constraints can be

used by enforcing rigid body equations of motion, for example, and a physics-guided algorithm may use multiple smaller NNs to represent force generating components, such as the engine, skin friction, and wings, instead of one large neural network to capture all forces and moments acting on the aircraft. The concept of building physics into the model is hypothesized to output a model with better generalizability and may be extrapolative.

Paper overview

This paper develops and compares two global aerodynamic models using flight data collected from the National Research Council of Canada's Bell 412HP aircraft in forward flight. The first approach, herein the *Classical* approach, is a conventional one using polynomial regression of point models. The second approach uses a physics-based neural network, called *FlyNet*. The structure of the neural network is determined by tuning hyperparameters and the output models are compared in terms of their time history matches, generalizability, and trimmability.

Global modelling architecture

Background theory

A linear state-space approximation is typically used for point modelling and the determination of the frequency-response or stability characteristics. Consider first the conventional linear state space approximation of a helicopter model

$$\dot{\vec{x}} = \mathbf{A}\vec{x} + \mathbf{B}\vec{u} \tag{2}$$

where the states, $\vec{x}$, and controls, $\vec{u}$, are

$$\vec{x} = \begin{bmatrix} u & v & w & p & q & r & \theta & \phi \end{bmatrix}^T \tag{3}$$

$$\vec{u} = \begin{bmatrix} \Delta_{\text{lon}} & \Delta_{\text{lat}} & \Delta_{\text{ped}} & \Delta_{\text{col}} \end{bmatrix}^T \tag{4}$$

A helicopter uses deflection of the longitudinal cyclic stick, Δ_{lon}, lateral cyclic stick, Δ_{lat}, pedals, Δ_{ped}, and the collective, Δ_{col}, for control. The body velocities, u, v, and w, and the angular velocities, p, q, and r, are fixed to the aircraft's centre of gravity and given relative to the inertial frame. Neglecting the effect of the spinning rotors, the rigid body equations of motion are

$$m\dot{u} = X - mg\sin\theta - mqw + mrv \tag{5}$$

$$m\dot{v} = Y - mg\cos\theta\sin\phi + mpw - mru \tag{6}$$

$$m\dot{w} = Z + mg\cos\theta\cos\phi - mpv + mqu \tag{7}$$

$$I_{xx}\dot{p} - I_{xz}\dot{r} = L + I_{xz}pq + qr\left(I_{yy} - I_{zz}\right) \tag{8}$$

$$I_{yy}\dot{q} = M + rp\left(I_{zz} - I_{xx}\right) + \left(r^2 - p^2\right)I_{xz} \tag{9}$$

$$I_{zz}\dot{r} - I_{xz}\dot{p} = N + I_{xz}qr + pq\left(I_{xx} - I_{yy}\right) \tag{10}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{11}$$

$$\dot{\phi} = p + q\tan\theta\sin\phi + r\tan\theta\cos\phi \tag{12}$$

where X, Y, Z, L, M, and N, are the forces and moments given in the body-fixed coordinate system, θ is the Euler pitch angle, and ϕ is the Euler roll angle. The time propagation of the numerical solution can be completed with a second-order Adams-Bashforth integration

scheme, which was found to be sufficient for the helicopter dynamic simulations by Crain et al. [54].

The flight dynamics problem is characterized by the control positions, states, and inertia, all of which are non-constant throughout the manoeuvre or flight envelope. The model is conventionally parameterized by the stability and control derivatives that are in the Taylor series of the forces and moments. The forces and moments can be represented by a Taylor series in multiple dimensions, such as

$$T(\chi_1, \ldots, \chi_d) = \sum_{n_1=0}^{\infty} \cdots \sum_{n_d=0}^{\infty} \left[\frac{(\chi_1 - a_1)^{n_1} \ldots (\chi_d - a_d)^{n_d}}{n_1! \ldots n_d!} \left(\frac{\partial^{n_1+\ldots+n_d}}{\partial x_1^{n_1} \ldots \partial x_d^{n_d}} \right) f(a_1, \ldots, a_d) \right] \tag{13}$$

where the vector $\vec{\chi}$ is a concatenation of the states and controls and $\vec{a}$ is the set of points about which the Taylor series is taken. The forces and moments are conventionally captured by a first-order Taylor series about the initial state and control values for a manoeuvre to form the state space approximation. That is

$$\left(n_i = 1 \mid n_j = 0,\ j \in [1,\ n_d],\ j \neq i, a_i = \chi_i(t = 0)\right) \tag{14}$$

For demonstration, this expansion simplifies to the following expression in the case of the longitudinal body force, X

$$\begin{aligned} X = {} & X_u \delta u + X_v \delta v + X_w \delta w + X_p \delta p + X_q \delta q + X_r \delta r \\ & + X_{\Delta_{\text{lon}}} \delta \Delta_{\text{lon}} + X_{\Delta_{\text{lat}}} \delta \Delta_{\text{lat}} + X_{\Delta_{\text{ped}}} \delta \Delta_{\text{ped}} + X_{\Delta_{\text{col}}} \delta \Delta_{\text{col}} + X_0 \end{aligned} \tag{15}$$

where the subscript notation denotes the derivative with respect to the variable. For example

$$X_u \equiv \frac{\partial}{\partial u} X \tag{16}$$

The X_0 term represents the constant portion of the Taylor series

$$X_0 = f(a_1, \ldots, a_d) \tag{17}$$

The set of derivatives with respect to the states are called *stability derivatives*, and the derivatives with respect to the controls are called *control derivatives*. The stability and control derivatives may be estimated using two main approaches. In the first approach, a linear state space model is formed by linearizing the rigid body equations of motion about the initial condition and combining with the first-order Taylor series of the forces and moments taken about the same point. The linear model can then be identified using time- or frequency-domain methods. In the second approach, the non-linear equations of motion are used and combined with the first-order Taylor series of the forces and moments taken about 0 to form a grey-box system identification problem performed in the time-domain using the output-error (OE) method–this approach was used by Crain et al. [54]. The dynamics of a rotor-craft in a any given flight regime (such as forward flight) are well established continuous functions, as are the forces and moments, but the collected flight data, as described in Sect. , contains noise and is discrete. The function used to fit

this data is nevertheless assumed to be smooth and continuous, as the objective of this paper is not to capture the noise inherently present in flight data. Rather, the objective is to capture the underlying smooth dynamics without the noise.

Point model system identification: the output-error method

Aircraft stability and control derivatives that are valid about a point in the flight envelope can either be identified using time-domain or frequency-domain system identification techniques. This paper uses the OE method for time-domain aircraft system identification. OE optimization for aircraft parameter estimation was first introduced by Main and Iliff [55,56] and remains a popular method [2,57]. Aircraft dynamics are a continuous-time system, but measurements are assumed to be taken at discrete time intervals for processing. The OE method minimizes the difference between the predicted model response corresponding to parameter vector $\vec{\lambda}$, $\hat{\vec{x}}\left(\vec{\lambda}\right)$, and the measured flight data, $\vec{x}$, as in

$$\vec{J}\left(\vec{\lambda}\right) = \frac{1}{\sqrt{N}}\left[\sum_{i=1}^{N}\left(x_{\langle t+i\rangle} - \hat{x}_{\langle t+i\rangle}\left(\vec{\lambda}\right)\right)^2\right]^{\frac{1}{2}} \tag{18}$$

where N is the number of discrete samples.

When simultaneously considering multiple channels with various unit bases and magnitudes, the channel may be scalarized through *mean normalization*, which is necessary to ensure that each channel is considered equally, as in

$$x_{\langle t\rangle}^{\text{norm}} = \frac{x_{\langle t\rangle} - \text{mean}(\vec{x})}{\max\left(\vec{x}\right) - \min\left(\vec{x}\right)} \tag{19}$$

This improves the performance and accuracy of most optimization routines [2].

The cost for a time history is the root mean square error (RMSE) between $\vec{x}^{\text{norm}}$ and its simulated estimate, $\hat{\vec{x}}^{\text{norm}}$, when evaluated using parameter vector $\vec{\lambda}$. For a single input test time-history, j, the overall cost is the average cost across all N_o output channels

$$J_j\left(\vec{\lambda}\right) = \frac{1}{N_o}\sum_{i=1}^{N_o} J_i\left(\vec{\lambda}\right) \tag{20}$$

Finally, the overall cost for a set of inputs at a single flight condition is the average of $J_j\left(\vec{\lambda}\right)$ for all m time-histories

$$J\left(\vec{\lambda}\right) = \frac{1}{m}\sum_{j=1}^{m} J_j\left(\vec{\lambda}\right) \tag{21}$$

Global modelling

Classical approach: point modelling with polynomial regression

The development of this global model using point modelling with polynomial regression is documented in References [54]. In this approach, the Taylor series is taken about 0 for all point models. Further, the nonlinear equations of motion are used instead of a state space model, which was found to improve the quality of global modelling matches by reducing the errors resulting from linearisation.

All stability and control derivatives are first determined using the output-error method. Next, all stability and control derivatives are assumed as a function of the dynamic pressure, P_d, and stepwise regression determines the appropriate order of the fit up to a maximum order of 2. For example, assuming X_u is a second-order function of P_d, the following line is fit using least-square regression:

$$\hat{X}_u = f\left(P_d\right) = b_1 P_d^2 + b_2 P_d + b_3 \tag{22}$$

where b_i are the fit parameters. A similar analysis is performed for each stability and control derivative.

FlyNet: physics-based neural network

The proposed global modelling approach closes the gap in the piecemeal approach of identifying linear models, connecting the points through regression, then validating using a quasi-non-linear simulation. Consider the example of a polynomial fit approach presented in Eq. (22). Substituting this into Eq. (15) expands to

$$X_{uu} = \left(b_1 P_d^2 + b_2 P_d + b_3\right) u \tag{23a}$$

$$X_{uu} = b_1 P_d^2 u + b_2 P_d u + b_3 u \tag{23b}$$

This expanded form can be considered part of the Taylor expansion with

$$\sum_{i=1}^{d} n_i \leq \begin{cases} 2 & n_{P_d} \leq 1 \\ 3 & n_{P_d} = 2 \end{cases} \tag{24}$$

where many of the other terms in the Taylor series are discarded. This knowledge of the conventional global modelling approach indicates that a limited Taylor expansion about zero to the second- or third-order has traditionally been acceptable for global aircraft modelling. Instead of using a higher-ordered Taylor series, this paper postulates the quality of the global model matches will be improved with a neural network. The neural network uses the states, control positions, dynamic pressure, altitude, and longitudinal centre of gravity position as inputs and outputs the forces and moments. The hidden layer uses a hyperbolic tangent activation function and the output layer uses a linear activation function. Thus, the model structure is as follows

$$\vec{y}_{\langle t \rangle} = \mathbf{W}_2 \tanh\left(\mathbf{W}_1 \chi_{\langle t \rangle} + \vec{b}_1\right) + \vec{b}_2 \tag{25}$$

The parameter identification process for a model of this form utilizes the output-error method. Unlike the previous approaches, a single model can consider all test points simultaneously. Thus, for a flight data set consisting of N_t test points, the cost function is the average across all time histories

$$J\left(\vec{\lambda}\right) = \frac{1}{N_t} \sum_{j=1}^{N_t} J_j\left(\vec{\lambda}\right) \tag{26}$$

If all higher-ordered terms are used as inputs, this will include many more inputs than conventional methods, many of which will have a weak influence on the model response.

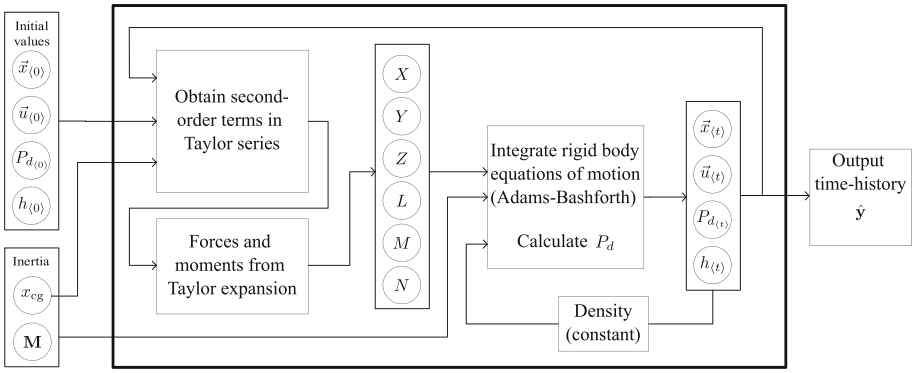

Fig. 1 Flow chart of FlyNet

The use of $L2$ regularization will pull these insensitive parameters toward 0 and may help prevent model overfitting. In this case, the cost function becomes

$$J\left(\vec{\lambda}\right) = \frac{1}{n_t}\sum_{j=1}^{n_t} J_j\left(\vec{\lambda}\right) + \alpha \sum \lambda_k^2 \tag{27}$$

where α is the regularization parameter.

A flow chart of the FlyNet model is shown in Fig. 1. The piecemeal approach of developing point models then using some method to connect the stability and control derivatives makes no guarantee on the performance of the global model since this is not considered in the curve fitting process. The approach of considering all points simultaneously with a neural network consisting of higher-ordered input terms ensures positive global model matches since these are directly considered in the cost function. The structure of the neural network is informed by conventional physics-based flight simulation models and ensures C^1 continuity. The simplicity of this method does not require expert insight or manual intervention and opens it up to a larger range of non-expert users. In Fig. 1, $\vec{x}$ is the vector of states, $\vec{u}$ is the vector of controls, P_d is the dynamic pressure, h is the altitude, $\mathbf{M}$ is the mass and inertia matrix, and x_{cg} is the longitudinal center of gravity.

Case study: comparison of global modelling methods

Aircraft and flight data collection

Flight test data collection was completed using the Advanced Systems Research Aircraft Bell 412HP (S/N 36034), which is operated by the National Research Council of Canada (NRC). It is powered by a PT6T-3BE TwinPac power plant. The maximum gross weight of the aircraft is 11900lbs. Flight testing was performed using the standard mechanical flight control system, which incorporates dual limited-authority automatic flight control systems. These are the standard helipilots furnished with the Bell 412HP and feature both rate damping and attitude retention modes. The yaw axis incorporates rate damping in either mode. The data acquisition system onboard the aircraft collects data at 128 Hz. The data was collected for a Level D full flight simulator to the same standard as required by FAA Title 14 CFR Part 60 [41]; additional details on the aircraft and the instrumentation suite can be found in References [58]. The aircraft is shown in Fig. 2.

This report considers only forward flight for modelling. Each point model is developed from a set of 2-3-1-1 inputs and control response tests. A 2-3-1-1 input is a series of

Fig. 2 Bell 412HP advanced systems research aircraft

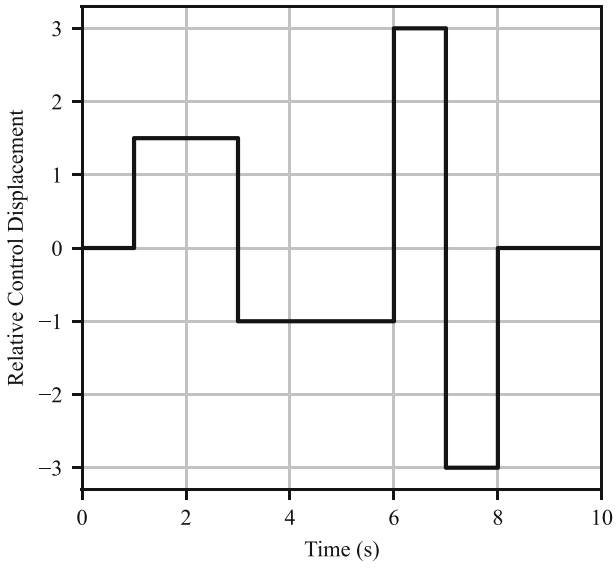

Fig. 3 Typical 2-3-1-1 control input

4 consecutive and alternative step inputs of 2, 3, 1, and 1 seconds in duration in a given control axis. A set of four 2-3-1-1 inputs consisting of one each of inputs in the longitudinal cyclic stick, lateral cyclic stick, pedal, and collective channels shall be referred to hereafter as an input *quartet*. A typical 2-3-1-1 input is shown in Fig. 3. The quartet test points were generally conducted around 45 Knots Indicated Airspeed (KIAS), 60KIAS, 75KIAS, 90KIAS, and 105KIAS.

The data for modelling was separated into two subsets: an *optimization set* and a *test set*. The optimization set is used for model identification and consists of fifty 2-3-1-1 input quartets ranging in speed from 30 Knots True Airspeed (KTAS) to 120KTAS and control response tests for a total of 252 individual time-histories. The test set is not used in the model identification and is used to test the ability of a model to generalize to new cases. The test set consists of 45 time histories including control response tests and two 2-3-1-1 input quartets randomly selected from each of the aforementioned speed bins for a total of ten quartets. It should be noted that separating the data into training and test sets is not required by the regulations for full flight simulators [41] but this is a best practice in the machine learning community.

Global modelling

Classical approach

Parameter identification was performed for each 2-3-1-1 quartet for a total of 50 identified point models. The parameters were identified using a constrained interior point optimization algorithm [59] with the cost function given by Eq. (21). Parameter values were left unbounded. All parameters were initialized using a linear least-squares parameter estimate. Since dynamic stability tests from flight testing demonstrated stability in forward flight, stability was enforced for each point model. Mathematically, a linear flight dynamics model is stable when the real part of the eigenvalues of the state matrix, **A**, are all less than zero. Thus, for stability

$$\max\left(\Re(\text{eig}(\mathbf{A}))\right) < 0 \tag{28}$$

The stability constraint given by Eq. (28) was imposed as a non-linear constraint. The identification of the point model derivatives used non-linear time-varying equations of motion with constant parameters. A full description of the modelling approach is available in References [54].

Once all flight dynamics models were identified, the global model was created using P_d as the regressor. P_d was selected as the regressor since it is a metric that considers both the forward speed, which is conventional [1], as well as the altitude, which is represented by the density in the dynamic pressure expression. All stability and control derivatives, as well as the constant values in the Taylor expansion, were fit using stepwise regression up to the second-order of P_d.

FlyNet

Model training

The model training consists of two parts: first is a pre-training using a feed-forward system, and second is training the system to optimize simulation matches. In the first step, parameters were initialized using Glorot uniform initialization [60], which randomly initializes each parameter in the range $[-\varepsilon, \varepsilon]$ where

$$\varepsilon = \sqrt{\frac{6}{N_i + N_o}} \tag{29}$$

where N_i is the number of input terms and N_o is the number of output terms. The parameters were subsequently refined as a feed-forward system where tabulated states, controls, and the second-order terms are the inputs, and the forces and moments calculated from flight data are the outputs. Optimization of the parameters uses the Adam optimizer [61] with a learning rate of 10^{-3} for 500 epochs. Feed-forward training is faster than the closed-loop system and minimizes the prediction error but makes no guarantee of the simulation error. The prediction error refers to the error one time step ahead whereas the simulation error is the error of the entire time history match. This approach also does not risk simulation model divergence due to initial model instabilities.

After the model is pre-trained, the output-error method was applied to the closed-loop system using the Adam optimizer to minimize the cost function given by Eq. (27) with regularization parameter $\alpha = 10^{-4}$. The loss function also included the initial trim error

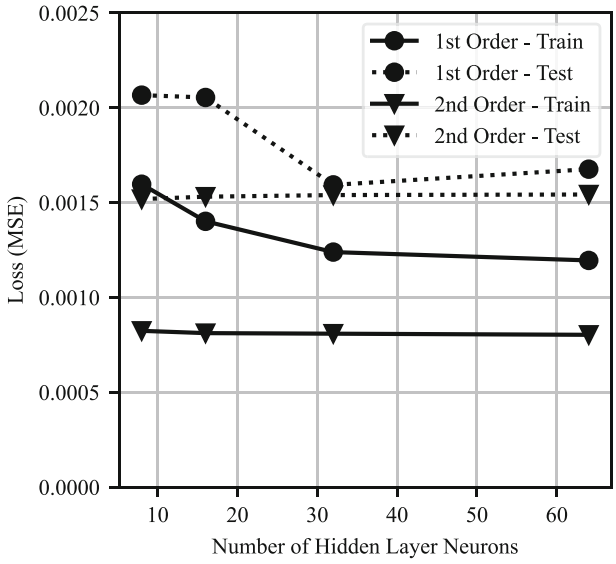

Fig. 4 Losses of models using 1st and 2nd order inputs

as given by the normalized force and moment error in each degree of freedom. This allows the model to simultaneously optimize for time history matches and trimmability.

Model structure determination

Hyperparameter tuning was conducted to determine the optimal input order and the number of neurons in the hidden layer. Using first-order inputs only is intended to test if the neural network alone is able to capture the nonlinearities of the global model instead of using second-order input terms. The losses of models trained with first- and second-order inputs with the number of nodes in the hidden layer $\in \{8, 16, 32, 64, 128\}$ were determined and are plotted in Fig. 4. The models were all initially pre-trained in a feed-forward manner for 500 epochs then as a closed-loop for 1000 epochs. The final MSEs of the time history matches are plotted in Fig. 4.

This figure demonstrates that there are small gains in the performance of the model with first-order inputs using 32 neurons in the hidden layer while the model with second-order inputs had marginal gains with more than 16 hidden layer neurons. Models with 2nd order inputs had better performance on the training set with similar performance on the test set. Accordingly, the model structure will use second-order input terms and 16 nodes in the hidden layer, which has a total of 2278 trainable parameters. This model was subsequently trained for 2500 epochs using the Adam optimizer with a learning rate of 10^{-3}. The training and test set losses are plotted in Fig. 5, which shows a stable and asymptotic minimum was attained. Further, both training and test set losses remained stable, thus indicating that the solution is not one of high variance.

Results

The average RMSEs for each channel between the two approaches with the 95% confidence interval (CI) are given in Tables 1 and 2. The tables also present the one-tailed p-value using a paired-sample t-test of the null hypothesis. Since the datasets are dependent, a paired-sample t-test was performed on the null hypothesis, H_0, which is that mean value

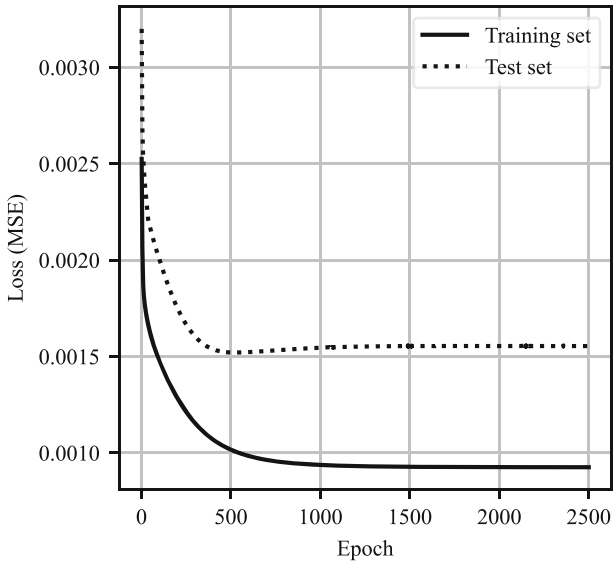

Fig. 5 Training and test set losses versus epoch during model training

Table 1 Average RMSEs in training set ($N = 252$) for each channel with 95% CI and *p*-value

	FlyNet	Classical	*p*-value
u [fts^{-1}]	2.222 [0.327, 6.392]	3.623 [0.753, 22.473]	0.000
w [fts^{-1}]	1.288 [0.339, 3.723]	1.626 [0.455, 8.903]	0.000
q [rads^{-1}]	0.012 [0.004, 0.029]	0.018 [0.005, 0.087]	0.000
θ [rad]	0.023 [0.006, 0.064]	0.038 [0.009, 0.213]	0.000
v [fts^{-1}]	2.159 [0.502, 6.169]	3.112 [0.653, 14.319]	0.000
p [rads^{-1}]	0.024 [0.006, 0.057]	0.035 [0.010, 0.200]	0.000
r [rads^{-1}]	0.021 [0.006, 0.053]	0.032 [0.008, 0.136]	0.000
ϕ [rad]	0.044 [0.009, 0.138]	0.074[0.021, 0.648]	0.000

Table 2 Average RMSEs in test set ($N = 45$) for each channel with 95% CI and *p*-value

	FlyNet	Classical	*p*-value
u [fts^{-1}]	2.038 [0.468, 6.082]	3.171 [0.780, 15.539]	0.081
w [fts^{-1}]	1.765 [0.657, 6.394]	1.728 [0.507, 6.174]	0.207
q [rads^{-1}]	0.015 [0.006, 0.043]	0.018 [0.005, 0.071]	0.020
θ [rad]	0.028 [0.007, 0.096]	0.042 [0.009, 0.225]	0.022
v [fts^{-1}]	2.698 [0.824, 6.879]	3.508 [1.129, 14.636]	0.016
p [rads^{-1}]	0.030 [0.010, 0.078]	0.037 [0.015, 0.207]	0.065
r [rads^{-1}]	0.026 [0.008, 0.076]	0.032 [0.010, 0.110]	0.014
ϕ [rad]	0.050 [0.013, 0.164]	0.072[0.021, 0.295]	0.071

of the Classical approach, μ_1, equals that of FlyNet, μ_0, as in

$$H_{0_1} : \mu_0 = \mu_1, H_{0_1} : (\mu_0 - \mu_1) = 0 \tag{30}$$

Thus, the alternative hypothesis, H_A, is

$$H_A : \mu_1 > \mu_0, H_A : (\mu_1 - \mu_0) > 0 \tag{31}$$

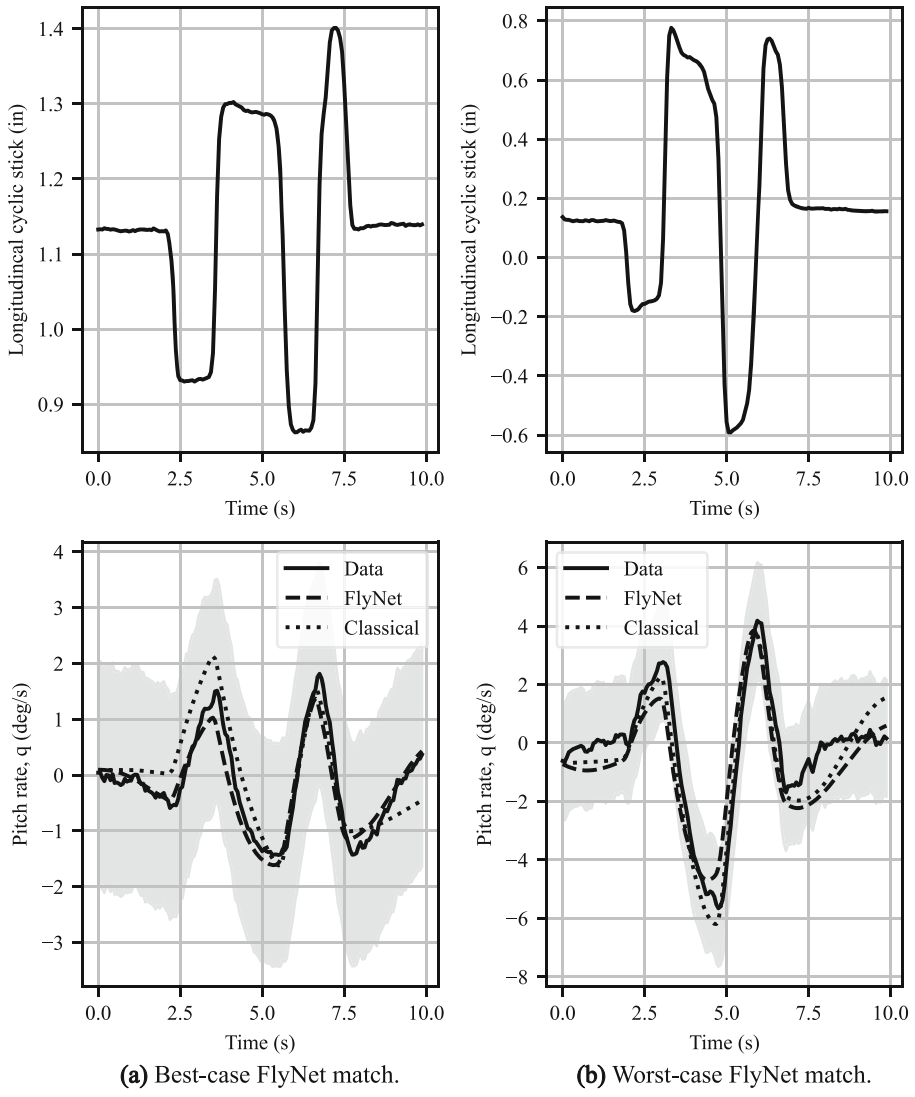

Fig. 6 Example pitch rate time history matches for FlyNet and classical models for a longitudinal stick 2-3-1-1 input in the training set. The gray area is the $\pm 2 \text{deg.s}^{-1}$ bounds set in [41] for full flight simulators

Table 1 demonstrates that FlyNet has a lower average RMSE than the Classical approach in all cases for the training set with a p-value less than 5% in all cases. FlyNet also has a lower average RMSE than the Classical approach for the test set in most cases cases except for the w matches ($p = 0.207$) and the p-value approaches significance for u, p, and ϕ. Example time history matches for the best- and worst-case matches of FlyNet to the pitch rate for a longitudinal cyclic stick input 2-3-1-1 is presented in Fig. 6.

The average of the absolute initial trim offsets with the 95% confidence interval and the one-tailed p-value from the paired-sample t-test are given in Tables 3 and 4. The tables show that the FlyNet model has superior trim performance compared to the Classical model with greater than 95% confidence.

The difference between the overall training set losses of the two models, $J_{\text{FlyNet}} - J_{\text{Classical}}$, are plotted in Fig. 7. This figure shows that in nearly all cases, the loss of the classical model is greater than that of FlyNet. A one-tailed paired-sample t-test results in a p-value

Table 3 Average absolute initial trim offsets for training set ($N = 252$) with 95% CI and p-value

	FlyNet	Classical	p- value
$\dot{u}_0$ [fts^{-2}]	0.861 [0.122, 4.720]	2.159 [0.164, 5.627]	0.000
$\dot{v}_0$ [fts^{-2}]	0.824 [0.121, 4.179]	1.010 [0.110, 4.148]	0.001
$\dot{w}_0$ [fts^{-2}]	1.259 [0.157, 6.071]	2.780 [0.230, 8.550]	0.000
$\dot{p}_0$ [rads^{-2}]	0.027 [0.002, 0.080]	0.165 [0.014, 0.551]	0.000
$\dot{q}_0$ [rads^{-2}]	0.010 [0.001, 0.033]	0.080 [0.009, 0.314]	0.000
$\dot{r}_0$ [rads^{-2}]	0.015 [0.001, 0.047]	0.048 [0.004, 0.132]	0.000

Table 4 Average absolute initial trim offsets for test set ($N = 45$) with 95% CI and p-value

	FlyNet	Classical	p- value
$\dot{u}_0$ [fts^{-2}]	1.115 [0.151, 4.212]	2.826 [0.759, 5.236]	0.000
$\dot{v}_0$ [fts^{-2}]	0.764 [0.067, 2.603]	1.297 [0.158, 6.154]	0.006
$\dot{w}_0$ [fts^{-2}]	1.640 [0.348, 5.404]	3.822 [0.376, 10.227]	0.000
$\dot{p}_0$ [rads^{-2}]	0.025 [0.003, 0.061]	0.226 [0.024, 0.812]	0.000
$\dot{q}_0$ [rads^{-2}]	0.013 [0.003, 0.056]	0.123 [0.008, 0.311]	0.000
$\dot{r}_0$ [rads^{-2}]	0.017 [0.002, 0.074]	0.064 [0.006, 0.140]	0.000

of 0.0% for the training set and 2.9% for the test set, thus confirming with greater than 97% confidence that FlyNet outperforms the classical model on average and better generalizes to new data.

Observations and discussion

The reference flight test data included sixty 2-3-1-1 input quartets. The test points were generally conducted around 45KIAS, 60KIAS, 75KIAS, 90KIAS, and 105KIAS. The test set randomly selected two quartets from each test condition. Two points at each speed is a small sample size and thus the average loss has a large associated standard deviation. It is assumed that the difference between the training and test set losses is a result of the small sample size rather than resulting from a model with high variance since Fig. 4 shows that the difference between the losses remains similar for simpler models.

Perhaps one of the greatest limitations of the Classical piecemeal approach of first identifying a linear model then performing regression is due to the large variances in the point model derivatives. This approach lacks a method of implementing meaningful bounds on derivatives for point modelling to limit the variances such that a function with both low bias and variance can be fit to the data. One way of addressing this is using an orthogonal function approach to fit the derivatives, such as that of Morelli [27]. However, a good fit to stability and control derivatives does not necessarily translate to good time-history matches. The identified stability and control derivatives are specialized for that one point and thus there is random spread in the derivatives resulting from unmodelled dynamics, manoeuvre execution, beginning the test out of trim, atmospheric disturbances and other sources of random error. Obtaining a good fit to these randomly distributed parameters will result in a high variance fit that does not generalize well. Additionally, the disconnect between the linear model assumption when identifying point models then subsequently using a quasi-non-linear model for global simulation, which has varying stability and control derivatives, will further amplify the effect of a high variance fit.

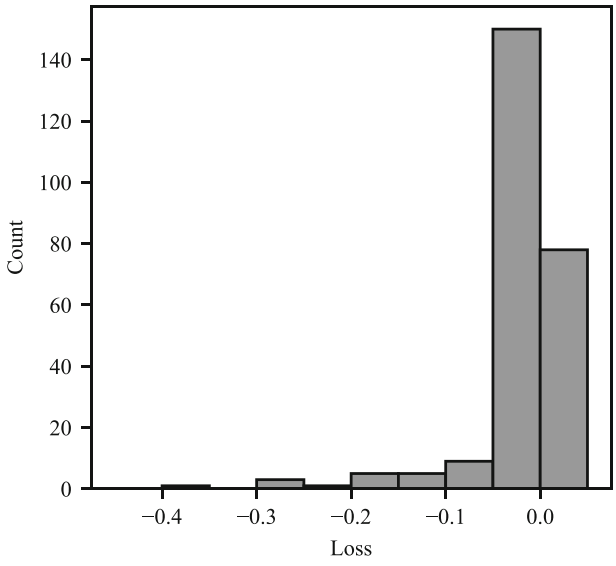

Fig. 7 Histogram of null hypothesis losses, $J_{\text{FlyNet}} - J_{\text{Classical}}$

FlyNet does not require fitting a function to point models. Since it fits all point tests simultaneously, this has a regularization effect to prevent the over-fitting that may occur when considering a single point at a time. Further, by considering many test points, higher-ordered terms can be introduced without as high of a risk of obtaining a high variance model. With the large number of terms in this second-order model, several of these terms may not have a significant influence on the model and thus $L2$ regularization is used to aid in convergence and further prevent an over-specialized model. It was shown with over 95% confidence that FlyNet generalizes better to test set data so it can be concluded that the model does not exhibit over-fitting behaviour.

Improvements in computational power and memory permit the ability to consider all points simultaneously and improved optimization algorithms allow for the consideration of a large number of parameters. It should be noted, however, that FlyNet optimizes fewer parameters overall than the classical approach. For a 6-DOF model using the classical approach, there are 66 parameters for identification at each point. When there are 60 test points, this leaves the optimization of 3960 parameters plus the parameters for the global model fits. The proposed approach requires the determination of a total of 2278 parameters and does require the additional step of determining global model fits.

FlyNet showed an improved ability to trim over the Classical approach. Both modelling approaches take the Taylor expansion about 0 so the simulated time history will result in a high cost using the output-error method if it begins out of trim. When considering only a single point at a time, the constant terms in the Taylor expansion can adequately capture the static behaviour and trim the model for the test. However, the global model fits to the constant terms do not guarantee the quality of the global model trim solution in the same manner that the fits to the stability and control derivatives do not guarantee the quality of the global model dynamic matches. In comparison, FlyNet directly considers the cost of the global model so the simulated time history will have a high cost if it begins out of trim. This forces the optimization to simultaneously consider trimmability both indirectly and by directly including the trim error in the loss function. The improved trimmability using

FlyNet is observed in Tables 3 and 4, which shows that the novel model begins closer to a trim condition in every axis with a one-tailed p-value from a paired-sample t-test of approximately 0%.

The FlyNet architecture is physics informed. The neural network part generates forces, thus allowing the closed-loop system to enforce rigid body equations of motion and use physics-based data, such as the mass and inertia during closed-loop training, and calculated forces and moments for feed-forward training. This architecture is in contrast to more complex neural network models that merely take measured states and controls and output state derivatives or the value one step ahead. Future work will include separate models for the rotors, and bluff-body aerodynamics, thus furthering the physics-guided algorithm.

Future work will also assess the handling qualities and pilot feedback using a model developed using this novel approach. It is possible and trivial to estimate the stability and control derivatives of this model about a given set of initial conditions. The estimation of these derivatives would allow one to form a stability matrix from which the eigenvalues can be determined. This may also be used to estimate the frequency response about a given condition, which can be compared to the estimated frequency response from flight data.

Conclusion

This study compared the performance of a two-step pseudo-non-linear global aircraft modelling architecture to a novel single-step continuous and non-linear global modelling method using a physics-based neural network called FlyNet. It was shown that a model with second-order input terms consisting of states and control positions with a neural network that outputs forces and moments outperformed the conventional global modelling approach with greater than 97% confidence for both training and test sets. Since FlyNet considers all second-order terms in the Taylor expansion, it automatically determines the model structure without the need for expert insight or manual intervention that may be required for legacy multi-step approaches. The novel approach also had improved trimmability compared to the legacy approach. Future work shall assess the handling qualities of a model developed using this novel approach, as well as its performance on a fixed-wing aircraft.

Acknowledgements

The authors would like to acknowledge Greg Craig and Patrick Zdunich from the NRC Flight Research Laboratory for their reviews of this work. The authors would also like to thank Laird McKinnon and Jeremy Smith, DTAES 7-5 Flight Sciences, who have inspired this work by their shared vision of making increasing use of simulation analyses through software engineering tools that are more accessible to non-expert users.

Author contributions

TS developed the FlyNet architecture, trained the model, analysed and interpreted the results, and wrote the manuscript. AC developed the flight data processing software. AC and JR developed the Classical model. JR collected the flight test data. All authors contributed to the conception and theoretical development of FlyNet. All authors read and approved the final manuscript.

Funding

This research was funded by the Defence Technologies and Sustainment Program under the leadership of Prakash Pratnaik as a strategic investment of the National Research Council of Canada and Directorate - Technical Airworthiness and Engineering Support 7-5 Flight Sciences of the Department of National Defence.

Availability of data and materials

The data that support the findings of this study are available from the National Research Council Canada but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly

available. Data are however available from the authors upon reasonable request and with permission of the National Research Council Canada.

Declarations

Competing interests
The authors declare that they have no competing interests

Received: 25 February 2022 Accepted: 14 June 2022
Published online: 04 July 2022

References

1. Tischler MB, Tobias EL. A Model stitching architecture for continuous full flight-envelope simulation of fixed-wing aircraft and rotorcraft from discrete point linear models. U.S. Army Aviation and Missile Research Development and Engineering Center; 2016. http://www.dtic.mil/docs/citations/AD1008448.
2. Klein V, Morelli EA. Aircraft system identification: theory and practice. American Institute of Aeronautics and Astronautics; 2006.
3. Tischler MB, Remple RK. Aircraft and rotorcraft system identification. American Institute of Aeronautics and Astronautics; 2006.
4. Brandon JM, Morelli EA. Real-time onboard global nonlinear aerodynamic modeling from flight data. J Aircr. 2016;53:1261–97.
5. Aiken EW. A Mathematical representation of an advanced helicopter for piloted simulator investigations of control-system and display variations. National Aeronautics and Space Administration; 1980.
6. Tischler MB. Aerodynamic model for piloted V/STOL simulation. Systems Technology. Inc; 1982.
7. Klein V, Batterson JG, Smith PL. On the determination of airplane model structure from flight data. IFAC Proc Vol. 1982;6(15):1163–8.
8. McNally BD. Full-envelope aerodynamic modeling of the harrier aircraft. NASA; 1986.
9. Downs J, Prentice R, Dalzell S, Besachio A, Ivler CM, Tischler MB. et al. Control system development and flight test experience with the MQ-8B fire scout vertical take-off unmanned aerial vehicle (VTUAV). In: American helicopter society 63rd annual forum. 2007 5;.
10. Burnett EL, Atkinson C, Beranek J, Sibbitt B, Holm-Hansen B, Nicolai L. NDOF simulation model for flight control development with flight test correlation. AIAA Model Simul Technol Conf. 2010;2010:1–14.
11. Lawrence B, Malpica CA, Theodore CR. The development of a large civil tiltrotor simulation for hover and low-speed handling qualities investigations. In: 36th European rotorcraft forum, Association Aéronautique et Astronautique de France; 2010. .
12. Zivan L, Tischler MB. Development of a full flight envelope helicopter simulation using system identification. J Am Helicopter Soc. 2010;4:55.
13. Mansur MH, Tischler MB, Bieleeld MD, Bacon JW, Cheung KK, Berrios MG. et al. Full flight envelope inner-loop control law development for the unmanned K-MAX. American Helicopter Society 67th Annual Forum; 2011. p. 5.
14. Greiser S, Seher-Weiss S. A contribution to the development of a full flight envelope quasi-nonlinear helicopter simulation. CEAS Aeronaut J. 2013;10(5):53–66.
15. Spires JM, Horn JF. Multi-input multi-output model-following control design methods for rotorcraft. American Helicopter Society 71st Annual Forum; 2015. p. 5.
16. Tobias E, Tischler M, Berger T, Hagerott SG. Full flight-envelope simulation and piloted fidelity assessment of a business jet using a model stitching architecture. AIAA Modeling and Simulation Technologies Conference; 2015. p. 1.
17. Knapp ME, Berger T, Tischler M, Cotting MC. Development of a full envelope flight identified F-16 simulation model. 2018 AIAA atmospheric flight mechanics conference. 2018. p. 1.
18. Berger T, Tischler MB, Hagerott SG, Cotting MC, Gray WR. Identification of a full-envelope learjet-25 simulation model using a stitching architecture. J Guid Control Dyn. 2020;43:2091–111.
19. de Visser C, Mulder J, Chu Q. Global Aerodynamic Modeling with Multivariate Splines. AIAA Modeling and Simulation Technologies Conference and Exhibit. 2008 8;.
20. de Visser C, Mulder J, Chu Q. A Multidimensional spline-based global nonlinear aerodynamic model for the cessna citation II. AIAA Atmospheric Flight Mechanics Conference; 2010. p. 8.
21. Hui K, Ricciardi J, Ellis K, Tuomey D. Beechjet flight test data gathering and level-D simulator aerodynamic mathematical model development. AIAA Atmospheric Flight Mechanics Conference and Exhibit; 2001.
22. Hui K, Ricciardi J, Srinivasan R, Lambert E, Sarafian A. Assessment of the dynamic stability characteristics of the bell model M427 helicopter using parameter estimation technology. SAE technical paper series. 2002. p. 11.
23. Hui K, Srinivasan R, Auriti L, Ricciardi J, Blair K, Pokhariyal D. et al. King air 350 flight-test data gathering and level-D simulator aerodynamic model development. In: ICAS 2002 congress. ICAS; 2002. p. 1–10.
24. Hui K, Auriti L, Ricciardi J. Advances in real-time aerodynamic model identification. J Aircr. 2005;1(42):73–9.
25. Hui K, Lambert E, Seto J. Bell M427 flight test data gathering and level-D simulator model development. In: 25th international congress of the aeronautical sciences. ICAS; 2006.
26. Hui K, Auriti L, Ricciardi J. Cessna citation CJ1 flight-test data gathering and level-C simulator model development. In: 26th international congress of the aeronautical sciences. ICAS; 2008. .
27. Morelli EA. Global nonlinear aerodynamic modeling using multivariate orthogonal functions. J Aircr. 1995;3(32):270–7.
28. Morelli EA. Global nonlinear parametric modeling with application to F-16 aerodynamics. American Control Conference; 1998. p. 6.
29. Lombaerts TJJ, Oort ERV, Chu QP, Mulder JA, Joosten DA. Online aerodynamic model structure selection and parameter estimation for fault tolerant Control. J Guid Control Dyn. 2010;5(33):707–23.
30. Morelli E. Efficient global aerodynamic modeling from flight data. In: 50th AIAA aerospace sciences meeting including the New Horizons Forum and aerospace exposition; 2012. p. 1.

31. Morelli EA, Cunningham K, Hill MA. Global aerodynamic modeling for stall/upset recovery training using efficient piloted flight test techniques. In: AIAA modeling and simulation technologies (MST) conference. 2013. p. 8.
32. Wang Z, Lan C, Brandon J. Fuzzy logic modeling of nonlinear unsteady aerodynamics. In: 23rd atmospheric flight mechanics conference. 1998. p. 8.
33. Brandon JM, Morelli EA. Nonlinear aerodynamic modeling from flight data using advanced piloted maneuvers and fuzzy logic. NASA; 2012.
34. Batterson JG. Estimation of airplane stability and control derivatives from large amplitude longitudinal maneuvers. NASA; 1981.
35. Batterson JG, Klein V. Partitioning of flight data for aerodynamic modeling of aircraft at high angles of attack. J Aircr. 1989;26:334–9.
36. Jategaonkar RV, Mönnich W, Fischenberg D, Krag B. Identification of C-160 simulator data base from flight data. IFAC Proc Vol. 1994;7(27):1031–8.
37. Seher-Weiss S. Identification of nonlinear aerodynamic derivatives using classical and extended local model networks. Aerosp Sci Technol. 2011;1(15):33–44.
38. Millidere M, Yigit T, Ulsu S. Full-envelope stiched simulation model of a fighter aircraft using the lasso technique. In: AIAA SCITECH 2022 forum. American Institute of Aeronautics and Astronautics Inc.; 2022.
39. Morelli EA. Autonomous real-time global aerodynamic modeling in the frequency domain. AIAA Scitech 2020 forum. 2020. p. 1. https://arc.aiaa.org/doi/10.2514/6.2020-0761.
40. Dreier ME. Introduction to helicopter and tiltrotor flight simulation. American Institute of Aeronautics and Astronautics (AIAA); 2018.
41. Federal Aviation Administration. 14 code of federal regulations appendix C to part 60—qualification performance standards for helicopter full flight simulators; 2016.
42. Shaheed MH. Feedforward neural network based non-linear dynamic modelling of a TRMS using RPROP algorithm. Aircr Eng Aerosp Technol 2005 2;77(1):13–22. https://www.emerald.com/insight/content/doi/10.1108/00022660510576000/full/html.
43. Fekih A, Xu H, Chowdhury FN. Neural networks based system identification techniques for model based fault detection of nonlinear systems. Int J Innov Comput Inf Control. 2007;10:3.
44. Punjani A. Machine learning for helicopter dynamics models; 2014.
45. Harris J, Arthurs F, Henrickson JV, Valasek J. Aircraft system identification using artificial neural networks with flight test data. In: 2016 international conference on unmanned aircraft systems (ICUAS). 2016 6;p. 679–688. http://ieeexplore.ieee.org/document/7502624/.
46. Priya SS. Application of neural networks for flight simulation. In: 1st IEEE international conference on power electronics, intelligent control and energy systems, ICPEICES 2016; 2017 2.
47. Norouzi R, Kosari A, Sabour MH. Investigating the generalization capability and performance of neural networks and neuro-fuzzy systems for nonlinear dynamics modeling of impaired aircraft. IEEE Access. 2019;7:21067–93.
48. Roudbari A, Saghafi F. Generalization of ANN-based aircraft dynamics identification techniques into the entire flight envelope. IEEE Trans Aerosp Electron Syst. 2016;52:1866–80.
49. Yu Y, Yao H, Liu Y. Physics-based learning for aircraft dynamics simulation. In: 10th annual conference of the prognostics and health management society. Prognostics and Health Management Society; 2018.
50. Chauhan RK, Singh S. Application of neural networks based method for estimation of aerodynamic derivatives. In: Proceedings of the 7th international conference confluence 2017 on cloud computing, data science and engineering. 2017 6;p. 58–64.
51. Ghazi G, Bosne M, Sammartano Q, Botez RM. Cessna citation X stall characteristics identification from flight data using neural networks. In: AIAA atmospheric flight mechanics conference. American Institute of Aeronautics and Astronautics.
52. Jacquemin T, Tomar S, Agathos K, Mohseni-Mofidi S, Bordas S. Taylor-series expansion based numerical methods: a primer, performance benchmarking and new approaches for problems with non-smooth solutions. Arch Comput Methods Eng. 2019;11(27):1465–513.
53. Deshpande S, Lengiewicz J, Bordas SPA. Probabilistic deep learning for real-time large deformation simulations. arXiv; 2021. https://arxiv.org/abs/2111.01867.
54. Crain A, Ricciardi J, Stachiw T. Bell 412 full flight envelope aircraft simulation model development and evaluation with nonlinear equations of motion. In: Volume 4: advances in aerospace technology. IMECE2021-71173. American Society of Mechanical Engineers.
55. Maine RE, Iliff KW. Identification of dynamic systems, theory and formulation. NASA Ames Research Center; 1985.
56. Maine RE, Iliff KW. Application of parameter estimation to aircraft stability and control: The output-error approach. Dryden Flight Research Cente: NASA Hugh L; 1986.
57. Góes LCS, Hemerly EM, de Oliveira Maciel BC, Neto WR, Mendonca C, Hoff J. Aircraft parameter estimation using output-error methods. Inverse Problems Sci Eng. 2006;14:651–64. https://doi.org/10.1080/17415970600573544.
58. Gubbels A, Carignan S, Ellis K, Dillon J, Bastian M, Swail C, et al. NRC bell 412 aircraft fuselage pressure and rotor state data collection flight test. In: 32nd European rotorcraft forum. Curran Associates; 2008. p. 1064–1086.
59. Byrd RH, Gilbert JC, Nocedal J. A trust region method based on interior point techniques for nonlinear programming. Math Program. 2000;11(89):149–85.
60. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics, vol 9. 2010. p. 249–256. http://proceedings.mlr.press/v9/glorot10a.html.
61. Kingma DP, Ba J. Adam: a method for stochastic optimization. 2014. p. 12.

Publisher's Note