

Aerodynamic Modelling of Saab 340B Development Using Binary Particle Swarm Optimization

Murat Millidere¹, Mushfiqul Alam², Simon Place³, Seyedali Mirjalili⁴,
and James Whidborne⁵

Cranfield University, Cranfield, Bedfordshire, MK43 0AL, United Kingdom
Torrens University, Adelaide, SA 5000, Australia

This paper follows-up previous work on the development of a high-fidelity Saab 340B aerodynamic model using system identification methods. In the prior work, Saab 340B flight tests were carried out using different excitations on the control surfaces. The flight test data was collected at predefined trim points. Thrust forces and moment were obtained using the propeller efficiency map provided by the manufacturer. The equation and output error methods were employed to analyse flight test data to estimate aerodynamic parameters in the time domain. This paper follows-up previous work on the development of a high-fidelity Saab 340B aerodynamic model using system identification methods. In the prior work, Saab 340B flight tests were carried out using different excitations on the control surfaces. The flight test data was collected at predefined trim points. Thrust forces and moment were obtained using the propeller efficiency map provided by the manufacturer. The equation and output error methods were employed to analyse flight test data to estimate aerodynamic parameters in the time domain. The paper extends the work to select independent variables in the equation error method in an optimal way using binary particle swarm to determine the best subset of independent variables. The impact of the hyperparameters of the binary PSO approach such as the transfer function scheme, inertia weight updating strategy, and the value of acceleration coefficients is investigated.

Nomenclature

CG	= Centre of gravity
α, β	= Angle of attack and angle of sideslip
V_{TAS}	= True airspeed
V_{CAS}	= Calibrated airspeed
KCAS	= Calibrated airspeed in knots
T	= Thrust force
D, C, L	= Aerodynamic drag, crosswind, and lift forces in wind coordinates
X, Y, Z	= Total longitudinal, lateral, and vertical forces in body coordinates
X_a, Y_a, Z_a	= Aerodynamic longitudinal, lateral, and vertical forces in body coordinates
X_p, Y_p, Z_p	= Propulsive longitudinal, lateral, and vertical forces in body coordinates

¹Research Fellow in Aircraft Systems Modelling and Simulation, Centre for Aeronautics

²Lecturer in Flight Dynamics, Dynamics, Simulation and Control Group, Centre for Aeronautics

³Senior Lecturer at National Flying Laboratory Centre

⁴Professor at the Centre for Artificial Intelligence Research and Optimization

⁵Professor in Control Engineering, Dynamics, Simulation and Control Group, Centre for Aeronautics

X_g, Y_g, Z_g	= Propulsive longitudinal, lateral, and vertical forces in body coordinates
C_X, C_Y, C_Z	= Nondimensional aerodynamic force coefficients vector resolved into body coordinates
C_D, C_C, C_L	= Nondimensional aerodynamic force coefficients vector resolved into wind coordinates
T_C	= Nondimensional thrust coefficient
C_T	= Nondimensional thrust coefficient
m	= Mass of the air vehicle
$\vec{g}$	= Acceleration due to gravity
p, q, r	= Angular roll, pitch and yaw rates of aircraft resolved in body coordinates
a_y, a_z	= Lateral and normal acceleration
h	= Altitude
$\delta_e, \delta_a, \delta_r$	= Effective angular deflections of elevator, aileron, and rudder
δ_{th}	= Throttle setting
δ_F, δ_{UC}	= Flap deflection angle and undercarriage deflection
$\vec{f}$	= Vector sum of all external forces
$\vec{f}_a, \vec{f}_p, \vec{f}_g$	= The resultant forces due to aerodynamics, propulsion, and gravity.
$\vec{m}_B$	= Vector sum of all external moments referred to as the vehicle CG
$\vec{m}_{B,a}$	= Aerodynamic moments referred to the vehicle CG in the body axis
$\vec{m}_{B,p}$	= Propulsive moments referred to the vehicle CG in the body axis

I. Introduction

The flight simulation is an essential element of modern aircraft design, flight control system design, certification process, and aircraft pilot training; however, the aerodynamic model forms the core of a flight simulator. An error in the aerodynamic model might compromise the qualification process [1]. Therefore, the flight simulator needs a high-fidelity aerodynamic database to replicate the actual aircraft behaviour for the pilots. There are several methods for developing the aerodynamic database. In the increasing order of fidelity, these are typically semi-empirical datasheet methods, linear flow solvers, nonlinear flow solvers, wind tunnel tests, and flight tests [2] [3] [4].

A mathematical model of a dynamical system from a collection of known flight test inputs and outputs is obtained using system identification. There are four key features of system identification: manoeuvre, measurements, methods, and models [5]. The prior work by the authors focused on the system identification method to develop an accurate model of the Saab 340B aircraft [6]. System identification in the time domain can be conducted using several methods such as the equation-error method [3] [5] [7] [8] [9] [10] [11], output error method [5] [7] [8], filter error method [5] [12] and artificial neural networks [3] [13] [14] [15] [16]. The equation-error method is used firstly to identify aerodynamic parameters. Any mismatch of the equation-error model to the (derivative) data will be integrated over time. Unless the modelling errors caused by the equation-error parameter estimation are minimal in amplitude (magnitude) and zero in mean, the result can be a considerable difference between measured outputs and the model outputs calculated using the equations of motion. The estimates obtained with the equation-error method are used as initial guesses, which are bounded at $\pm 10\%$ in the output-error process for fine-tuning (to avoid the drifts in outputs).

However, defining model structure (also known as feature selection) in the equation error method is often a problematic issue because it involves choosing a subset of relevant features from a larger set of available features. The challenge lies in determining which features are the most informative and contribute significantly to the predictive power of the model. There are some reasons for the feature selection can be problematic: 1) model complexity and interpretability 2) correlation among features 3) underfitting and overfitting concerns 4) dynamic nature of data. There are many approaches for defining model structure. Expert-based selection is a common approach whereby statistical criteria and flight mechanic intuition are used to select model terms [17] [18] [19]. However, it requires extensive knowledge of flight mechanics. Subset selection is the another approach, which divides into two parts: best subset selection and stepwise selection [20] [21] [22]. Best subset selection entails choosing the optimal model from among all possible models. When there is a large number of predictors, the best subset selection suffers computationally [22] [23]. Due to computational constraints, the best subset selection cannot be applied with a large predictor number. Stepwise methods, which examine a much smaller set of models than the best subset selection, are appealing alternatives. Stepwise selection may be classified into three types: forward stepwise selection, backward stepwise selection, and hybrid selection. They offer several advantages such as improved computational efficiency, automated feature selection, sequential model building etc. However, the stepwise approach is a kind of local search, and its main disadvantage is that it is susceptible to getting trapped in local optima, hindering the ability to discover the global optimum [22]. In this study, binary Particle Swarm Optimization is proposed to identify the most related independent

variables in the equation error method. There are some potential advantages of using Binary Particle Swarm Optimization over local feature selection methods like stepwise methods: 1) global search capability, 2) Parallel Exploration, 3) adaptability to non-linear relationships 4) robustness to feature dependencies, and 5) flexibility with evaluation metrics.

A mathematical model of the Saab 340B aircraft is used in this study with the flight simulation tool and the algorithms are developed using MATLAB [24]. The rest of the paper is organized as follows. Section II presents a summary of the flight simulation model. Section III describes the flight maneuvers and the data preprocessing. Section IV discusses the methodology. The results and discussions are given in Section V, and conclusions are drawn in Section VI.

II. Air Vehicle Flight Simulation Model

An air vehicle in flight is simulated using several subsystem models, including, but not limited to, the atmosphere, aerodynamics, propulsion, weight, and balance properties, landing gear (ground handling), equations of motion (flight dynamics and kinematics), flight control system (FCS) algorithms, actuators, and sensor models. An illustrative flow diagram is shown in Figure 1. The purpose of the equations of motion for flight dynamics is to provide sufficient knowledge about the motion of the air vehicle. The purpose of the other subsystem models is to produce air-vehicle data for the equations of motion model.

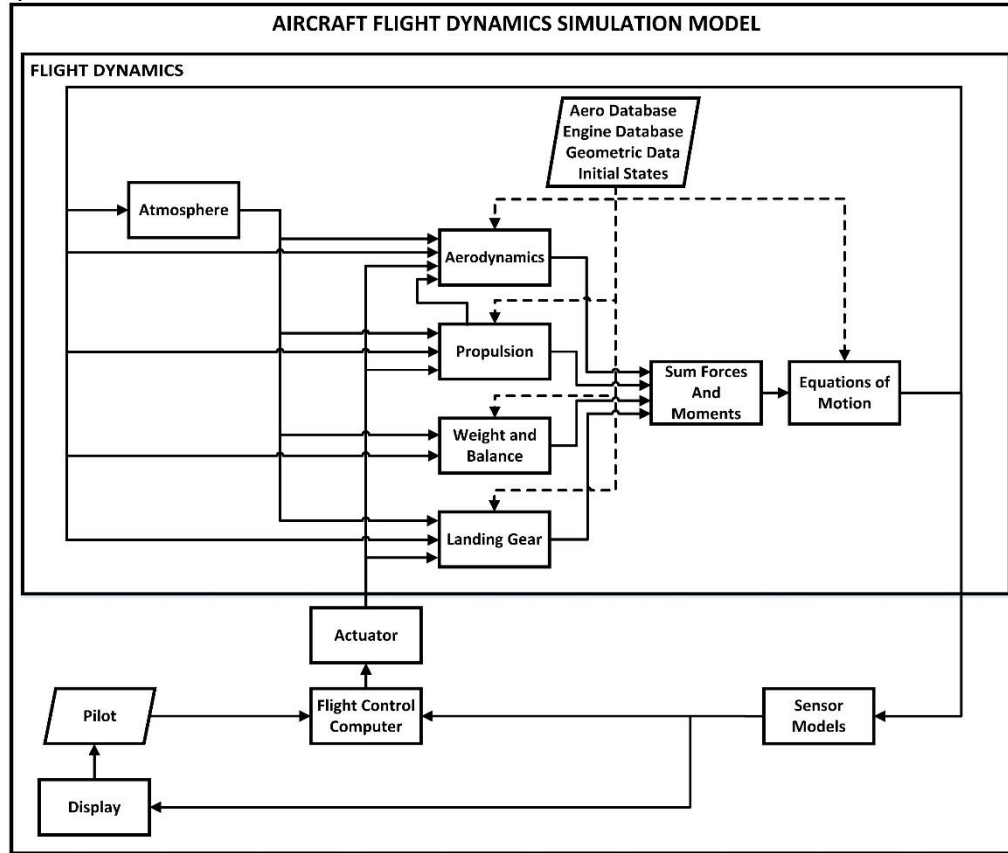


Figure 1 Flight Modelling Block Diagram

The sum of all applied forces and moments in the preceding equations arises from aerodynamics, propulsion, and gravity. Since gravity acts through the centre of gravity (CG) and the gravity field is uniform, no moment due to gravity acts on the air vehicle. The resultant force ($\vec{f}$) and moment ($\vec{m}_B$) are thus expressed as

$$\vec{f} = \vec{f}_a + \vec{f}_p + \vec{f}_g, \quad \vec{m}_B = \vec{m}_{B,a} + \vec{m}_{B,p} \quad (1)$$

where $\vec{f}_a$, $\vec{f}_p$ and $\vec{f}_g$ are the resultant forces due to aerodynamics, propulsion, and thrust, respectively and $\vec{m}_{B,a} + \vec{m}_{B,p}$ are the aerodynamic and propulsion moments referred to the vehicle CG in the body axis

This section comprises two subsections: the first describes the aerodynamic force and moment coefficients, and the second briefly describes the thrust modelling approaches. The reader is referred to [6] for more detail about 1) aerodynamic forces and moments and 2) propulsion forces and moments

III. Flight Maneuvers and Data Preprocessing

Saab 340B aircraft flight test data is used in this paper. A flight test program was carried out at Cranfield University National Flying Laboratory Centre (NFLC) to obtain flight data of a Saab 340B aircraft

Stability and control derivatives are estimated from the dynamic motion of an aircraft to specific control inputs. Many manoeuvres can be performed to excite the dynamic motion around separate axes utilizing individual inputs on each control input channel. It is usually recommended to begin each maneuver from a trimmed flight and allow about 5 to 10 seconds of steady flight before applying specific control inputs and allow sufficient time after the input for the aircraft to oscillate according to the aircraft mode [5] [8]. Maneuvers for system identification purposes were performed at some steady level flight conditions expressed in $145 \leq KCAS \leq 210$, and $5000 \text{ ft} \leq h \leq 10000 \text{ ft}$. Steady level trim data at discrete flight conditions $125 \leq KCAS \leq 220$, and $5000 \text{ ft} \leq h \leq 10000 \text{ ft}$ were also collected for testing the developed model. Specific maneuver were designed using specific control inputs from the pilot for individually/independently exciting short-period, phugoid, dutch-roll, bank-to-bank (roll) and thrust-variation manoeuvres. Figure 2 illustrates the flight test scope. Some test steps are also performed at idle or maximum power settings since the aerodynamic coefficients are affected by the thrust coefficient, also known as, slipstream effect. For more details about flight maneuvers and data recording parameters, the reader is referred to previous work [6].

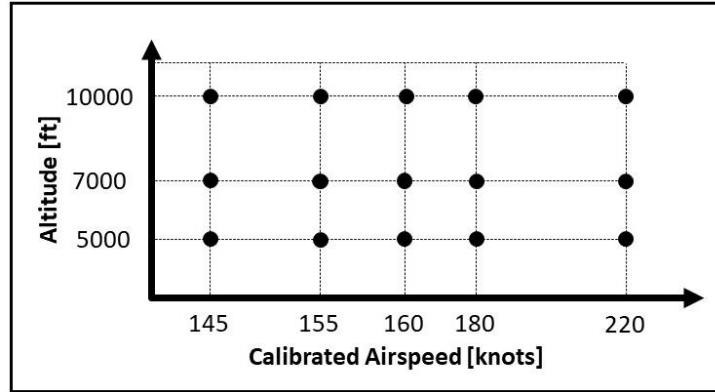


Figure 2 Flight test scope.

IV. Methodology

A. Model Description

The mathematical model of the dynamic system (process) in a state space realization is given by the following general nonlinear system [5] [7] [25]:

$$\begin{aligned} \dot{\vec{x}}(t) &= f[\vec{x}(t), \vec{u}(t), \vec{\beta}] + F(\vec{\lambda})\vec{w}(t), \quad \vec{x}(t_0) = \vec{x}_0 \\ \vec{y}(t) &= g[\vec{x}(t), \vec{u}(t), \vec{\beta}] \end{aligned} \quad (2)$$

where $\vec{x}$ is the $n_x \times 1$ column vector of state variables, $\vec{u}$ is the $n_u \times 1$ control input vector, $\vec{y}$ is the $n_y \times 1$ system output vector, $\vec{\beta}$ is the $n_\beta \times 1$ system parameter vector, n_x , n_u and n_y are the system dimensions functions, and f and g are general nonlinear functions. Besides the deterministic control input u , the system is also excited by process noise $\vec{w}(t)$, a stochastic input, which is usually a non-measurable $n_w \times 1$ column vector. The matrix F represents the additive process (state) noise distribution matrix.

Since it is not possible to measure the system parameters $\vec{\beta}$, they must be estimated from discrete measurements $z(t_k)$ of the model outputs $y(t_k)$. The output equation is formulated as

$$\vec{z}(t_k) = \vec{y}(t_k) + G\vec{v}(t_k) \quad (3)$$

where k is the discrete-time index and $\vec{v}(t_k)$ is the $n_v \times 1$ measurement noise vector. The measurement noise vector is assumed to be characterized by a sequence of independent Gaussian random variables with zero mean and identity covariance. The matrix G represents the additive measurement noise distribution matrix.

It is assumed that besides the system parameters $\vec{\beta}$, the elements $\vec{\lambda}$ of process noise distribution matrix F and the initial conditions $\vec{x}_0$ are also unknown. Accordingly, the unknown parameter vector is given by:

$$\vec{\theta} = [\vec{\beta}^T \quad \vec{\lambda}^T \quad \vec{x}_0^T]^T \quad (4)$$

There is no stochastic input in this study, i.e., process noise is assumed zero. So, the formulation becomes

$$\begin{aligned} \dot{\vec{x}}(t) &= f[\vec{x}(t), \vec{u}(t), \vec{\beta}] + F(\lambda)\vec{w}(t), \quad \vec{x}(t_0) = \vec{x}_0 \\ \vec{y}(t) &= g[\vec{x}(t), \vec{u}(t), \vec{\beta}] \\ \vec{z}(t_k) &= \vec{y}(t_k) + G\vec{v}(t_k) \end{aligned} \quad (5)$$

and the unknown vector is

$$\vec{\theta} = [\vec{\beta}^T \quad \vec{x}_0^T]^T \quad (6)$$

For the aircraft system identification problem, $\vec{\beta}$ corresponds to the stability and control derivatives or other parameters that model the aerodynamic and propulsive forces and moments.

B. Parameter Estimation Methods

The basic concept in parameter estimation methods is that the measured vehicle response is compared with the estimated response, and the difference between these responses is called the response error. A cost function includes this response error. The objective (cost) function J has the following form:

$$J(\theta) = \frac{1}{2} \sum_{k=1}^N (z(t_k) - y(t_k))^T (z(t_k) - y(t_k)) \quad (7)$$

The unknown coefficient values can be determined using any minimization optimization method. Gauss-Newton method is one of the most widely used minimization algorithms and is selected for this study. The unconstrained Gauss-Newton method can be extended to account for simple lower and upper bounds on the estimated parameters. This leads to a linearly constrained optimization problem, formulated as [26]:

$$\min_{\theta} J(\theta) \text{ subject to } \theta_{min} \leq \theta \leq \theta_{max} \quad (8)$$

that can be posed in the standard form,

$$\min_{\vec{\theta}} J(\vec{\theta}) \text{ subject to } c_1: \vec{\theta} - \vec{\theta}_{min} \geq 0, c_2: \vec{\theta}_{max} - \vec{\theta} \geq 0 \quad (9)$$

1. Equation Error Method

The equation error method is used to determine non-state parameters such as force and moment coefficients, which are not integrated during simulation. By using the equation-error form, there is no need to integrate equations of motion to achieve model outputs since the matching is performed in the equations of motion (hence the term "equation-error") [29]. By minimizing differences in the least-square sense between measured and model (estimated) responses, the equation-error approach estimates the unknown parameter vector (aerodynamic stability and control derivatives).

A wide range of information is needed to derive total aerodynamic force and moment coefficients from flight data. Angular rates and accelerations on the body axis, linear accelerations on the body axis, dynamic pressure, thrust-

induced forces and moments on the body axis, and aircraft mass and inertia data are among the data. The location of the accelerometer and air data system is often used to transfer air data and accelerations to the Centre of Mass. In addition, within each maneuver, the Centre of gravity location must be determined to transfer total moments to the moment centre (MC), around which an aerodynamic model can be constructed. The coefficients of nondimensional force and moment are obtained by substituting measured and known quantities. The equation error approach is used to model the functional dependence of aerodynamic forces and moments on aircraft motion and control variables since propulsive forces and moments are generally obtained from ground tests and Figure 3 illustrate this situation.

A preprocessing step is required in the case of aerodynamic parameter identification because the aerodynamic forces and moments are not directly measured but can be obtained from the measurements of the related variables, linear accelerations, angular rates, mass properties, and other external forces and moments as seen in Figure 3.

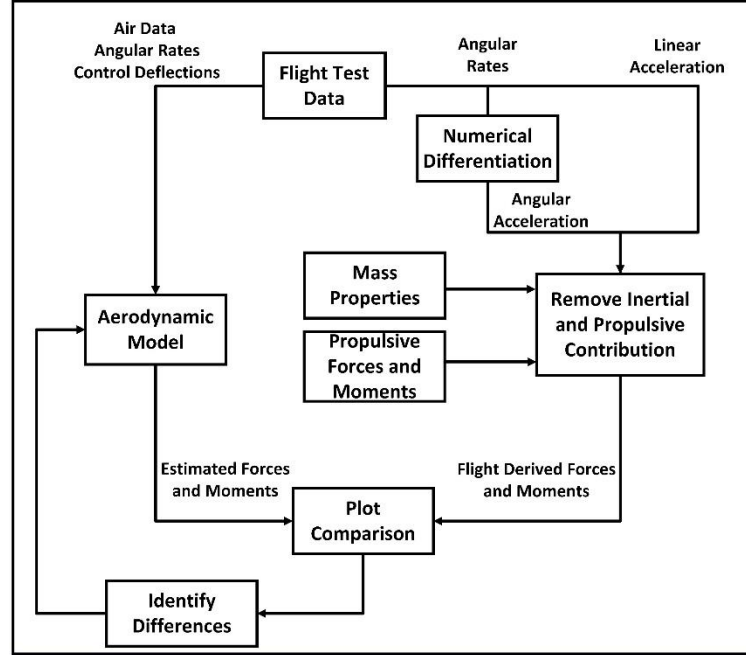


Figure 3 Flowchart of the Equation Error Method

For the equation error estimation, the control input, state variables, measured/model output vectors, and unknown (estimated) variables are shown in Table 1.

Table 1 Input, State, and Output Vectors and Unknown Variables for the Equation Error Method

Symbol	Description	Variables
$\vec{u}$	Control Input Vector	$[V, M, \alpha, \beta, p, q, r, \delta_e, \delta_a, \delta_r, T_C, \delta_F, \delta_{UC}]$
$\vec{x}$	State Variable Vector	$[-]$
$\vec{y}$	Model Observation Vector	$[C_D, C_Y, C_L, C_l, C_m, C_n]$
$\vec{\theta}$	Unknown Variables	$[C_D, C_Y, C_L, C_l, C_m, C_n]$ Model Variables

C. Model Structure Determination and Validation

This section covers the means of determining the accuracy, reliability, and usefulness of the specified model. For each coefficient, each model structure represents a multi-input, single-output subspace. This strategy enables more adaptability and simplifies the training process. The primary goal of aerodynamic model development is to optimize the model structure. Simplicity is preferable but yet possesses a sufficient level of complexity. Although restricting the number of terms used simplifies computing, it is essential to use the correct terms to ensure accuracy. Choosing

which terms to include in each coefficient model is a tedious process and named feature selection. There are numerous well-known methods for selecting appropriate terms (feature selection), including Stepwise Regression, Multivariate Orthogonal Functions (MOFs), and Generalized Additive Models (GAMs). The aerodynamic model structure used in this study is based on the one developed by Grauer and Morelli [27]. Grauer and Morelli used multivariate orthogonal function modelling to investigate measured wind tunnel aerodynamic datasets for eight aircraft over a wide variety of flight regimes. The most critical modelling terms (the number of selected instances is high) for each coefficient are selected as frozen model terms. The remaining model terms for longitudinal and lateral coefficients are selected using expert-based selection and a correlation-based binary particle swarm optimization method, respectively. Certain additions and rejections to the structure of the underlying aerodynamic model are made using quantitative and qualitative graphical findings analyzed by Jategaonkar [28] and resampling methods analyzed by Millidere et al. [29].

1. The Correlation Based Particle Swarm Optimization

In this study, only longitudinal coefficients are identified using the correlation-based particle swarm optimization. Initial model terms pool is generated based on a set of base regressors and a maximum term order. And then, pairwise correlation among the model terms is compared, and certain terms are removed. Finally, binary Particle Swarm Optimization is utilized to generate the best subset of the remaining terms. The flowchart of this process is shown in Figure 6.

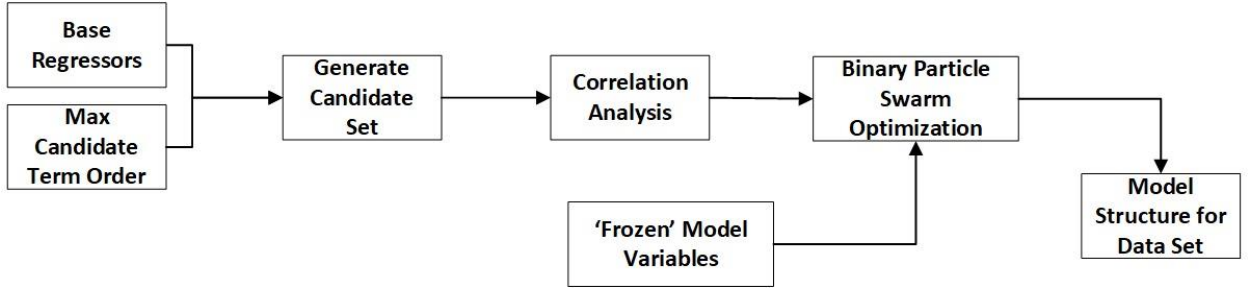


Figure 4 Correlated-Based Binary Particle Swarm Optimization Flowchart

Generating Candidate Pool

For longitudinal aerodynamic coefficients, the candidate model terms are as follows:

$$\begin{aligned}
 C_a = & \sum_{i=0}^3 (C_{a\alpha^i}) \alpha^i + \sum_{i=1}^2 \sum_{j=0}^2 (C_{a\delta_e^i \alpha^j}) \delta_e^i \alpha^j + \sum_{i=1}^2 \sum_{j=0}^2 (C_{aT_c^i \alpha^j}) T_c^i \alpha^j + \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=0}^2 (C_{a\delta_e^i T_c^j \alpha^k}) \delta_e^i T_c^j \alpha^k \\
 & + \sum_{i=1}^1 \sum_{j=0}^2 (C_{a\hat{q}^i \alpha^j}) \hat{q}^i \alpha^j
 \end{aligned} \tag{10}$$

where $a = L, D, m$.

The simplest way to generate a regressor pool is to utilize conventional regressors such as α , δ_e , T_c , and $\hat{q}$ for longitudinal coefficients. If there are no useful terms in the regressor pool, the quality of the model never improves.

Some model terms defined by [27] are so critical that these terms are frozen (default included) for the model structure. After deciding on frozen terms, to avoid correlations of model terms, the correlation between the model terms is checked before implementing binary particle swarm optimization. That is why this procedure is called the correlation-based binary particle swarm optimization technique.

Pearson's Correlation Coefficient

Pairwise correlation is a statistic that indicates the linear relationship between two or more variables. Correlation is used to select variables since the appropriate variables are strongly associated with the output. Additionally, variables should be associated with the output but not with one another. When two terms are correlated, the model requires just one of them, since the second provides no new information.

Given paired data $\{(x_1, y_1), \dots, (x_n, y_n)\}$ consisting of n pairs, Pearson's correlation coefficient, r_{xy} , is defined as [30]

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (11)$$

where n is the sample size, x_i, y_i are the individual sample points indexed with i , and $\bar{x}, \bar{y}$ are the mean of the x and y samples.

The value of Pearson's correlation ranges from -1 to +1. When the coefficient approaches to +1 it means that the variables are directly correlated, and coefficient approaching -1 means the variables are inversely correlated. A value of 0 indicates that variables are not linearly dependent on one another. If the variables are correlated, we can eliminate the one with a lower correlation coefficient value than the output variable. Additionally, we may calculate multiple correlation coefficients to determine if more than two variables are correlated to each other. This is referred to as multi-collinearity.

Binary Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic optimization technique [31]. This method is very popular because of its simplicity and low computing cost [32]. The PSO offers many advantages over other global optimization methods, including ease of implementation, and the minimal number of hyperparameters to adjust. PSO has been effectively used in a wide range of fields, including parameter estimation as well as artificial neural network training, and feature selection [33] [34] [35] [36] [37] [38] [39].

PSO begins with a collection of random particles (solutions) and then seeks optimal solutions. Each particle is updated by using personal best and global best at each iteration. Personal best refers to the best solution (fitness) of the particle. Global best refers to the best value attained by any particle in the population [31].

Following the determination of the particle's personal and global best values, the particle velocity and positions are updated using the following equations [31] :

$$\begin{aligned} \vec{v}_i^{k+1} &= w\vec{v}_i^k + c_1 r_1 (\vec{p}_i^k - \vec{x}_i^k) + c_2 r_2 (\vec{g}^k - \vec{x}_i^k) \\ \vec{x}_i^{k+1} &= \vec{x}_i^k + \vec{v}_i^{k+1} \end{aligned} \quad (12)$$

where $\vec{v}_i^k$ is the particle velocity, $\vec{x}_i^k$ is the current particle solution, r_1 and r_2 are random numbers between 0 and 1 and c_1 and c_2 acceleration coefficients. The search mechanism of the PSO is illustrated in Figure 8.

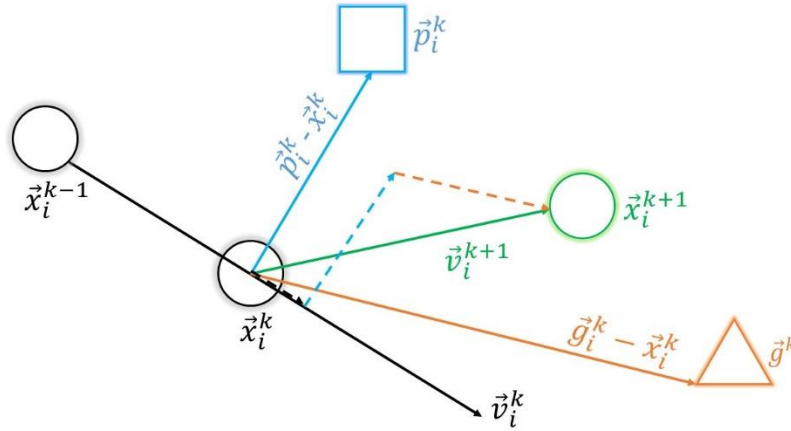


Figure 5 Search Mechanism of the PSO

The first term in the equation is referred to as the inertia component. It maintains the particle's original direction. The second term is called the cognitive component, which serves as the particle's memory, leading it to return to its optimal regions of the search space on an individual basis. The last term is referred to the social component. It leads the particle to gravitate toward the best global solution the swarm has discovered so far.

The binary variant of this method has been developed to address binary problems. A transfer function which is the essential component of the binary version maps search space from continuous to discrete.

Two distinct components differentiate the continuous and binary versions of PSO: a new transfer function and a different position update method. The transfer function is used to map search space from continuous to binary, and the updating process changes the positions of particles in binary search spaces between 0 and 1.

Particles can move around the search space in the continuous version of PSO by employing position vectors. As a result, the notion of position updating for particles may be simply realized by adding velocities to positions. However, position updating is different in the binary space [26]. Due to the fact that binary space contains just two integers ("0" and "1"), the operation of position updating cannot be accomplished. As a result, a method should be developed for altering agents' positions from "0" to "1" or vice versa. In other words, a relationship between velocity and position must be established [40].

Essentially, changing the position in discrete binary spaces entails flipping between "0" and "1" values. This flipping should be dependent on the agent velocities. The issue here is how to update the positions in a binary space using the idea of velocity in real space. The notion is to adjust an agent location based on the likelihood of its velocity. To do this, a transfer function is required to convert velocity values to probability values for position updates.

Two concerns must be addressed when recasting the feature selection as an optimization problem. The first concern is with the solution's representation. Given the binary nature of feature selection, we chose to express the solution as a binary vector, with 1 value indicating that the associated feature is selected and 0 values indicating that it is not. The size of the solution corresponds to the number of features in each dataset. The second concern is with the objective function's design. In the binary PSO feature selection task, the predicted squared error (PSE) metric is used. [27] [41] is used as the objective function to choose the model without overparameterizing the model.

$$PSE = \frac{1}{N} \sum_{i=1}^N \left(\tilde{z}_j - \tilde{y}_j(\vec{\theta}) \right)^2 + \sigma^2 \frac{n}{N} \quad (13)$$

The first component in the PSE reduces monotonically with each additional modelling term and quantifies the model fit error which is referred to as the mean squared fit error. The second component is the overfit penalty, which rises monotonically with each additional model term and prevents the model from being over-parameterized, resulting in poor prediction outcomes. With each additional model term, one component increases continually whereas the other term decreases continually, ensuring that PSE is always minimum at a single point. By selecting this point as the number of modelling terms, a model with low mean squared fit error and high prediction capacity is developed. The variance σ^2 is formulated as follows: [27]

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (\tilde{z}_i - \bar{\tilde{z}})^2 \quad (14)$$

The general steps of the BPSO algorithm are as follows:

- a. Define the objective function to be minimized.
- b. Each particle is assigned a random value.
- c. Initialize the hyperparameters of PSO: w , c_1 and c_2 .
- d. Repeat the following steps e to g until meeting the end condition.
- e. The velocities of each particle are updated.
- f. Determine the probability of changing the components of the position vector.
- g. Update the position vectors.

Hyperparameters of Binary Particle Swarm Optimization

The BPSO algorithm is an extremely efficient approach for solving feature selection problems. However, certain components of the feature selection process, called hyperparameters, are not updated autonomously. The hyperparameter is not the variable that is optimized through the feature selection process, but rather the variable specified by previous information, e.g., transfer function scheme, inertia weight updating strategy, and the value of acceleration coefficients [40]. The setting for these hyperparameters has a considerable impact on the performance of BPSO, and there are no universally optimal default values for various problems and data sets. The fundamental issue in designing a BPSO is determining the optimal hyperparameters. The brute force technique is applied to find the best hyperparameters. The numbers of population and generation number are taken to be 20 and 30, respectively. The dimension number is the number of terms in the generated pool. Each configuration is run 30 times, and the min, max, mean, and standard deviation of the PSE are recorded. The mean values are compared, and the one giving the minimum mean value is selected as the best configuration.

Binary PSO depends on a transfer function to convert a continuous search space to a discrete search space for its core component. S-shaped and v-shaped transfer functions are the two main families of transfer functions described in the literature. A transfer function's range should be restricted to the range [0,1]. All real values of velocity are transformed to probability values in the range [0,1] using an equation below that uses a sigmoid function [40].

$$T(v_i^{k+1}) = \frac{1}{1 + e^{-v_i^{k+1}}} \quad (15)$$

Position vectors may be updated as follows with the likelihood of their velocities once velocities have been converted to probability values:

$$x_i^{k+1} = \begin{cases} 1 & \text{if } rand < T(v_i^{k+1}) \\ 0 & \text{if } rand \geq T(v_i^{k+1}) \end{cases} \quad (16)$$

The other s-shaped transfer functions are listed in Table 2.

Table 2 S-Shaped Family of Transfer Functions

Name	Transfer Function
S1	$T(x) = \frac{1}{1 + e^{-2x}}$
S2	$T(x) = \frac{1}{1 + e^{-x}}$
S3	$T(x) = \frac{1}{1 + e^{-x/2}}$
S4	$T(x) = \frac{1}{1 + e^{-x/3}}$

Table 3 lists the most common V-shaped transfer functions.

Table 3 V-Shaped Family of Transfer Functions

Name	Transfer Function
V1	$T(x) = \left \operatorname{erf}\left(\frac{\sqrt{\pi}}{2}x\right) \right $
V2	$T(x) = \tanh(x) $
V3	$T(x) = \left \frac{x}{\sqrt{1+x^2}} \right $
V4	$T(x) = \left \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x\right) \right $

Because V-shaped functions are distinct from the s-shaped ones, additional rules for changing their positions are required. Position vectors should be updated using a formula based on velocities when using these types of transfer functions.

$$x_i^{k+1} = \begin{cases} (x_i^k)^{-1} & \text{if } rand < T(v_i^{k+1}) \\ x_i^k & \text{if } rand \geq T(v_i^{k+1}) \end{cases} \quad (17)$$

where $(x_i^k)^{-1}$ is the complement of x_i^k .

There are a total of eight different transfer functions and these are tabulated in Table 4.

Table 4 Different Transfer Functions

No	1	2	3	4	5	6	7	8
Name	S1	S2	S3	S4	V1	V2	V3	V4

Each population-based algorithm undergoes two main phases of optimization: exploration and exploitation. By maintaining a favorable balance between those phases, the algorithm may converge to the global optima in an acceptable amount of time. The ideal solution for any algorithm is to begin the search process by using the exploration operator more than the exploitation operator. One of the hyperparameters that control this balance is the inertia weight. There are many strategies for the inertia weight in the literature [39] [42] [43] [44]. In this study, only ten different inertia weight update strategies, which are listed in Table 5, are tested as follows:

Table 5 Different Inertia Weight Update Strategies

No	Name of the Inertia Weight	Formula of the Inertia Weight
1	Constant Inertia Weight	$w = 0.2$
2	Constant Inertia Weight	$w = 0.5$
3	Constant Inertia Weight	$w = 0.7$
4	Constant Inertia Weight	$w = 1.0$
5	Linear Decreasing Inertia Weight	$w^k = w_{max} - (w_{max} - w_{min}) \frac{k}{k_{max}}$ $w_{min} = 0.2, w_{max} = 0.8$
6	Nonlinear Decreasing Inertia Weight	$\left(\frac{2}{k}\right)^{0.3}$
7	Linear Increasing Inertia Weight	$w^k = w_{min} + (w_{max} - w_{min}) \frac{k}{k_{max}}$ $w_{min} = 0.2, w_{max} = 0.8$
8	Simulated Annealing Inertia Weight	$w^k = w_{min} + (w_{max} - w_{min}) 0.99^k$ $w_{min} = 0.2, w_{max} = 0.8$
9	Random Inertia Weight	$w^k = 0.5 + \frac{rand}{2}$
10	Global-Local Best Inertia Weight	$w_i^k = 1.1 - \frac{\vec{g}^k}{\vec{p}_i^k}$

The last hyperparameter that controls this balance is the acceleration coefficients. To have a good balance, the sum of c_1 and c_2 must be equal to 4 [31]

$$c_1 + c_2 = 4 \quad (18)$$

There are many options for c_1 and c_2 selection, but just two most common options are evaluated for this study as listed in Table 6. In the first, both are set to 2. And in the second option, c_1 decreases from 3.5 to 0.5 and c_2 grows linearly from 0.5 to 3.5 to balance exploration and exploitation.

Table 6 Different Acceleration Update Strategies

No	Name of the Inertia Weight	The Formula of the Inertia Weight
1	Constant Accelerations	$c_1 = 2$ $c_2 = 2$
2	Adaptive Accelerations	$c_1 = 3.5 - 3 \frac{k}{k_{max}}$ $c_2 = 0.5 + 3 \frac{k}{k_{max}}$

2. Statistical Accuracy

Statistical Accuracy: First, it is checked if the expected value is reasonable. The magnitude of the order must be close to the real value. And then, the standard deviation, relative standard deviation, and correlation matrix results are checked. The relative standard deviation of any term should not exceed 50, and none of the correlations between the two independent terms should exceed 0.9.

Statistical Accuracy is determined by first checking if the predicted value is rational. The order of magnitude must be similar to its actual value. After that, the effects of the standard deviation, relative standard deviation, and correlation matrix are checked. Any term's relative standard deviation should not exceed 50% since a large uncertainty level indicates low information in the derivative estimate, and a small uncertainty level indicates high information [45]. Data collinearity means that there is a linear or nearly linear relationship among model variables—the existence of data collinearity compromises parameter estimation results. Hence no correlation between two independent terms should be greater than 0.95.

Qualitative graphical results are shown in three distinct layouts: Time History Plots, Cross Plots, and Cross Plots of Residuals.

Comparing the time histories of flight-measured and model-estimated responses is a widespread method for qualitatively evaluating the model's fidelity. Some inconsistencies in matching the two responses often offer critical information for improving the model fidelity.

In addition to plotting time histories of measured and predicted flight responses, cross plots of arbitrary two variables are sometimes needed, for example, lift coefficient versus angle of attack, rolling and yawing moment coefficients versus angle of sideslip, and drag coefficient versus lift coefficient. These cross-plots are advantageous for gaining more insight into the modelling issues.

The residuals test (the difference between the predicted and measured response), also known as cross plots of residuals, is a powerful indicator of the reliability of the assumptions made about measurement noise on the system. The ideal case is for the residual distribution to be flat and centered near zero.

Resampling Methods: rather than evaluating the model on the training data, the trained model is evaluated for the data in the testing dataset not used in the training model, are assessed. The k-Fold cross-validation method is used as a resampling method.

V. Results and Discussion

A. Binary Particle Swarm Optimization

The longitudinal aerodynamic coefficients are determined using the hybrid BPSO approach and results are presented in the next subsections. The algorithm starts by generating 31 candidate model variables (terms). And then, pairwise correlation of the output variable with model variables and pairwise correlation among the model variables are calculated. If the variables are correlated among themselves, i.e., the correlation is higher than 0.95, the variables with a lower correlation coefficient value with the output variable are dropped. Almost half of the variables are dropped after correlation analysis, and finally the binary Particle Swarm Optimization algorithm is run to decide the best subset.

In Table 7, the correlated variables are shown. The total number of dropped variables is 11 for C_L coefficient.

Table 7 The Dropped Variables of C_L model after Correlation Analysis

Variable	Dropped Variables
$C_{L\alpha^2}$	$\{C_{L\alpha^3}\}$
$C_{LT_c\alpha^2}$	$\{C_{LT_c\alpha}\}$
$C_{L\delta_e\alpha^2}$	$\{C_{L\delta_e\alpha}\}$
$C_{LT_c^2\alpha^2}$	$\{C_{LT_c^2\alpha}\}$
$C_{L\delta_eT_c\alpha^2}$	$\{C_{L\delta_eT_c\alpha}\}$
$C_{L\delta_e^2\alpha^2}$	$\{C_{L\delta_e^2\alpha}\}$
$C_{L\delta_eT_c^2\alpha^2}$	$\{C_{L\delta_eT_c^2\alpha}\}$

$C_{L_{T_c}}$	$\{C_{L_{T_c^2}}\}$
$C_{L_{\delta_e^2 T_c \alpha^2}}$	$\{C_{L_{\delta_e^2 T_c \alpha}}\}$
$C_{L_{\delta_e T_c^2 \alpha^2}}$	$\{C_{L_{\delta_e T_c^2 \alpha}}\}$
$C_{L_{q\alpha}}$	$\{C_{L_{q\alpha^2}}\}$

The dropped variables in Table 7 are removed from the candidate list. And some variables as listed in Table 8 are kept fixed in this study because these variables are considered the most influential variables for a generic aircraft in Ref. [27]. So, after dropping 11 correlated variables and fixing the 5 most influential variables, we have total 15 float variables to optimize.

Table 8 The Fixed Variables of C_L Model

#	Variable
1	C_{L_0}
2	C_{L_α}
3	$C_{L_{\alpha^2}}$
4	C_{L_q}
5	$C_{L_{\delta_e}}$

There is total 160 configurations for different acceleration schemes (AS), transfer functions (TF) and inertia weight schemes (IWS). Table 9 gives the best fitness, worst fitness, mean fitness and standard deviations for the best 10 BPSO configurations with respect to mean fitness. It can be seen that mean fitness produced by Acceleration Scheme (AS) 1, Transfer Function (TF) 1, and Inertia Weight Scheme (IWS) 2 is the least with the lowest standard deviation as compared to other configurations. All configurations achieve the same best fitness, but Conf. 1 has better consistency.

Table 9 The Results of the Best 10 BPSO Configurations for C_L Model

Conf #	AS	TF	IWS	Best Fitness	Worst Fitness	Mean Fitness	Std. Dev
1	1	1	2	2.56E-06	2.66E-06	2.57E-06	3.3E-08
2	1	1	3	2.56E-06	2.69E-06	2.58E-06	3.8E-08
3	1	1	7	2.56E-06	2.83E-06	2.58E-06	6.3E-08
4	1	2	2	2.56E-06	2.66E-06	2.58E-06	4.2E-08
5	1	1	8	2.56E-06	2.72E-06	2.58E-06	4.7E-08
6	1	3	9	2.56E-06	2.81E-06	2.59E-06	6.9E-08
7	2	1	8	2.56E-06	2.73E-06	2.59E-06	4.8E-08
8	1	3	4	2.56E-06	2.72E-06	2.59E-06	5.0E-08
9	1	1	4	2.56E-06	2.72E-06	2.59E-06	5.5E-08
10	2	8	6	2.56E-06	2.74E-06	2.59E-06	4.9E-08

For all configurations, The best solutions are obtained for 133 instances out of 160 configurations. The modeling terms which appeared in more than half of the configurations are exactly the same with the first being the best configuration.

B. Equation-Error Method

The resulting model structure for lift coefficient, C_L , is as follows:

$$\begin{aligned}
C_L = \frac{1}{\sqrt{1-M^2}} & \left([C_{L_0} + C_{L_\alpha} \cdot \alpha + C_{L_{\alpha^2}} \cdot \alpha^2] \right. \\
& + [C_{L_{\delta_e}} \cdot \delta_e + C_{L_{\delta_e \alpha^2}} \cdot \delta_e \alpha^2 + C_{L_{\delta_e^2}} \cdot \delta_e^2 + C_{L_{\delta_e^2 \alpha}} \cdot \delta_e^2 \alpha] + [C_{L_q} \cdot \frac{qc}{2V}] \\
& + [C_{L_{T_c}} \cdot T_c + C_{L_{T_c \alpha^2}} \cdot T_c \cdot \alpha^2 + C_{L_{T_c^2}} \cdot T_c^2 + C_{L_{T_c^2 \alpha^2}} \cdot T_c^2 \cdot \alpha^2] \\
& \left. + [C_{L_{T_c \delta_e}} \cdot T_c \cdot \delta_e + C_{L_{T_c^2 \delta_e}} \cdot T_c^2 \cdot \delta_e + C_{L_{T_c \delta_e \alpha}} \cdot T_c \cdot \delta_e \cdot \alpha] \right)
\end{aligned} \tag{19}$$

The time history results for C_L in the training dataset is given in Figure 6. As you seen, the first 3 maneuvers are set idle power settings, the maneuvers 3-8 are set at

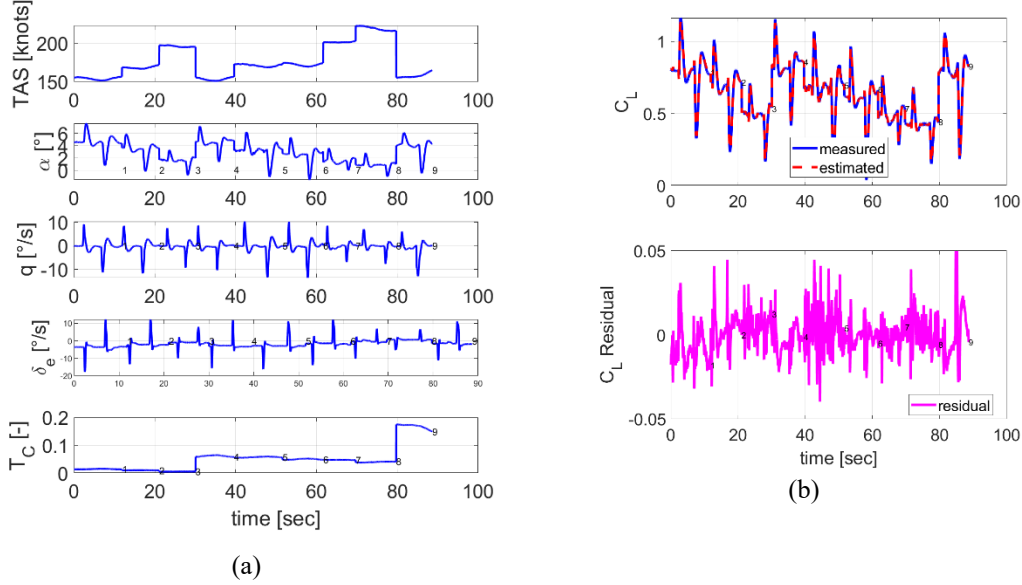


Figure 6 Training Dataset Results of C_L Coefficient a) Time Histories of the Input Variables b) Time Histories of the Output variable

The zoomed time history results for C_L in the training dataset is given in Figure 7. The residual centered around zero, which is a good sign that the model is predicted with good accuracy.

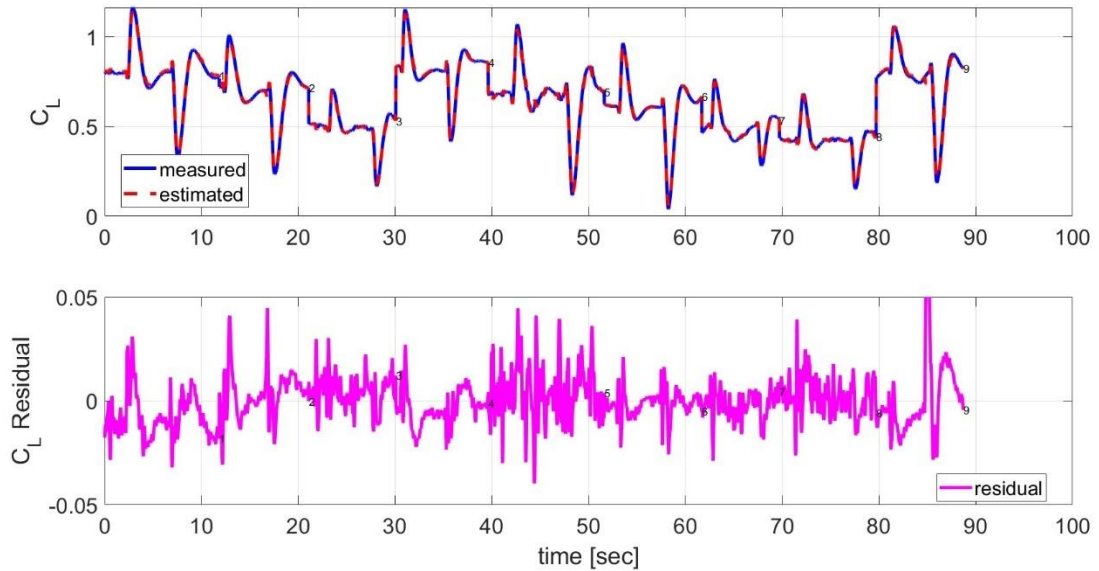


Figure 7 Training Dataset Results of C_L Coefficient a) Time Histories of the Input Variables b) Time Histories of the Output variable

Residual cross plot for C_L with respect to air data, angular rates, and control surface deflections and cross plot for C_L with respect to the angle of attack in the training dataset are given in Figure 8.

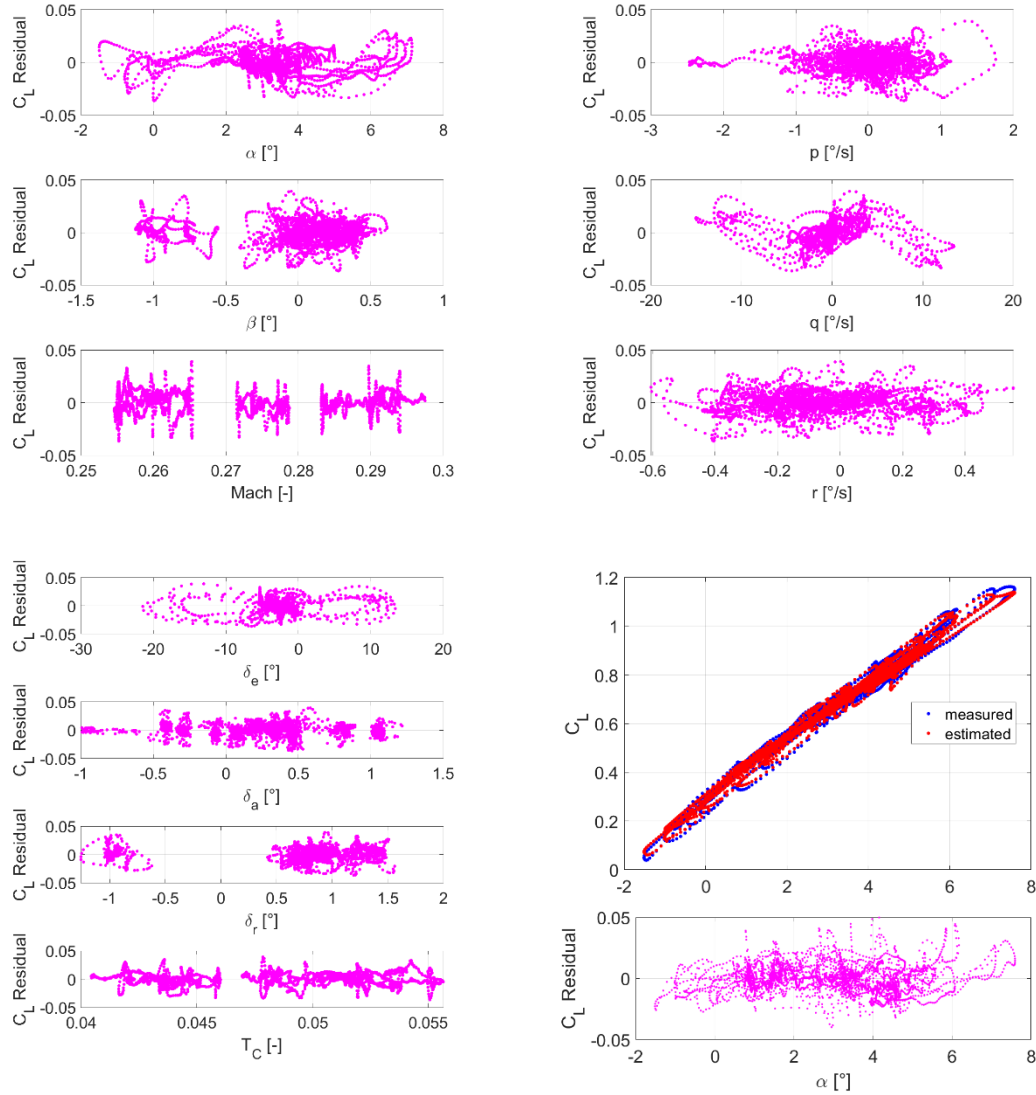


Figure 8 Residual Cross Plots and Cross Plot for C_L in Training Dataset.

The resulting model structure for drag coefficient, C_D , coefficient is as follows: For drag coefficient, we implemented a different technique than lift coefficient by setting the first two variables frozen ($C_{D0}, \frac{C_L^2}{\pi \cdot e \cdot AR}$), and finding an optimal structure for other independent variables.

$$C_D = C_{D0} + \frac{C_L^2}{\pi \cdot e \cdot AR} + [C_{D\delta_e} \cdot \delta_e] + [C_{DT_c} \cdot T_c + C_{DT_{c\alpha}} \cdot T_c \cdot \alpha + C_{DT_c^2} \cdot T_c^2 + C_{DT_c^2\alpha} \cdot T_c^2 \cdot \alpha] \quad (20)$$

where e is Oswald efficient factor and is generally between 0.6 and 1.

In addition to generated lift coefficient regressor pool, some new terms related to elevator trim tab is included for pitch moment coefficient. The resulting model structure for C_m coefficient is as follows.

$$C_m = [C_{m_0} + C_{m_\alpha} \cdot \alpha] + [C_{m_{\delta_e}} \cdot \delta_e + C_{m_{\delta_e^3}} \cdot \delta_e^3] + [C_{m_q} \cdot \frac{qc}{2V} + C_{m_{q\alpha}} \cdot \frac{qc}{2V} \cdot \alpha] + [C_{m_{T_c}} \cdot T_c] + [C_{m_{\delta_{eTT}}} \cdot \delta_{eTT}] \quad (21)$$

The time history results for longitudinal coefficients in the testing dataset is given in Figure 8.

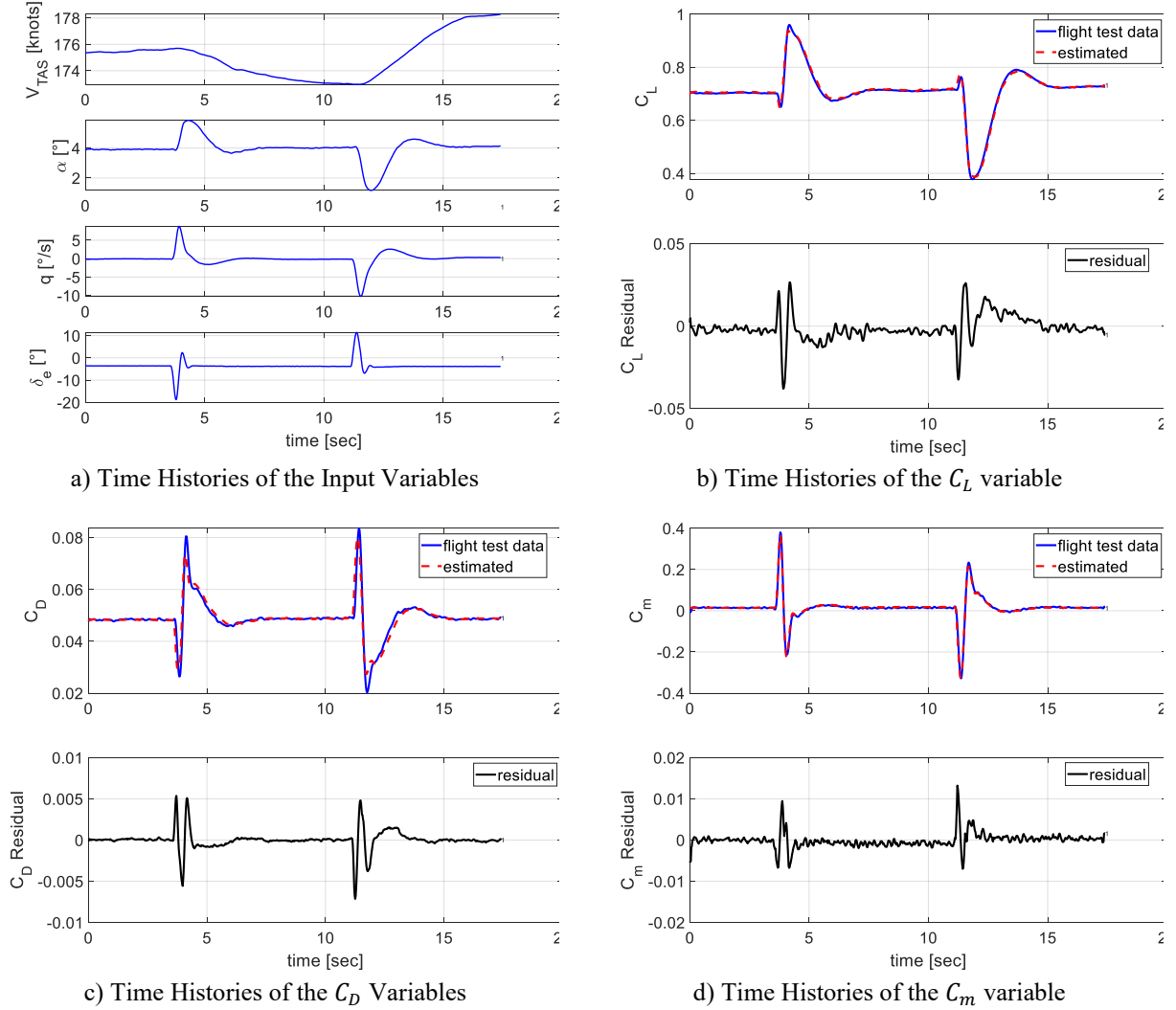


Figure 9 Testing Dataset Results of Longitudinal Coefficients

VI. Conclusion

This study was based on flight test data from the Saab 340B aircraft at a clean configuration. The clean configuration means that no high-lift devices (flaps) or undercarriage are deployed. Control inputs of the pilot have been configured to excite short period, phugoid, dutch roll, and bank-to-bank (roll) manoeuvres for discrete flight conditions where $145 \leq KCAS \leq 180$, and $5000 \text{ ft} \leq h \leq 10000 \text{ ft}$. Steady level trim data at discrete flight conditions $125 \leq KCAS \leq 220$, and $5000 \text{ ft} \leq h \leq 10000 \text{ ft}$ were also collected for testing the developed model. Test steps are performed at the level power setting. The angles of attack and sideslip were covered between $-2^\circ \leq \alpha \leq 8^\circ$ and

$-8^\circ \leq \beta \leq 8^\circ$ in those flight tests. Hence, the obtained aerodynamic model structure and estimated values are valid for this flight envelope, which might not be applicable to outside the flight envelope.

The correlation-based particle swarm optimization is used as a feature selection method. The algorithm starts by generating a set of candidate model terms, based on a set of base regressor and a maximum term order. And then, pairwise correlation among the model terms is compared and some terms are removed. And finally, binary Particle Swarm Optimization is utilized to generate the best subset of the remaining terms.

The BPSO is an efficient algorithm to solve feature selection problems. However, specific components, referred to as hyperparameters, are not autonomously updated in the feature selection process. The hyperparameters of the binary PSO approach are the transfer function scheme, inertia weight updating strategy, and the value of acceleration coefficients. The setting for these hyperparameters have a significant impact on the performance of BPSO, and there are no universal default values that perform well for different problems and data sets. The fundamental issue in designing a BPSO is determining the best configuration of the hyperparameters. To determine the best configuration hyperparameters, 160 configurations are tested over 30 times and their mean fitness are compared. It is observed that choice of acceleration scheme does not affect the result, and different choices are seen to give essentially the same performance. The normalized fitness sum of Acceleration Scheme 1 is slightly better than the second one so it can be used for further studies. The most dominant hyperparameter affecting the binary PSO performance is the choice of transfer function. S1 transfer function is superior to other. Among inertia weight schemes, the most influential ones are 4th, 7th and 9th inertia weight strategies which are Constant Inertia Weight, Linear Increasing Inertia Weight, and Random Inertia Weight. After the analysis, 160 total configuration reduced to 3 total configuration.

Acknowledgements

This research was funded by the UK Research and Innovation under the Powerplant Integration of Novel Engine Systems (PINES) project (Rolls-Royce). The authors of this paper are extremely grateful to Dr Linghai Lu (Centre for Aeronautics) for his invaluable assistance and feedback in defining flight test scope and flight manoeuvres. The authors are also thankful to Peter Beecroft (Rolls-Royce), Dr Kacper Grzedzinski (Centre for Autonomous and Cyberphysical Systems), and Wojciech Korek (Centre for Aeronautics) for their collaboration in the Future System Simulator. The authors would like to extend their sincere thanks to Robert Harrison, Joseph Brown, and Andrew Patrick from National Flying Laboratory Centre for conducting flight tests. The authors would like to acknowledge Jeet Makadia (Postgraduate Student) for his help with providing weight and balance data. They all greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

References

- [1] D. Allerton, Principles of Flight Simulation, West Sussex, UK: A John Wiley and Sons Inc., 2009.
- [2] H. B. Kurt, M. Millidere, F. S. Gömeç and Ö. Uğur, "Multi-fidelity Aerodynamic Dataset Generation of a Fighter Aircraft," in *AIAA Scitech Forum*, Virtual Conference, 2021.
- [3] Q. Liu, P. Zhang, W. Xiao and J. D. Diao, "Correction Methods of Aerodynamic Force and Moment Coefficients Based on the Identification Data," in *Chinese Control Conference*, Guangzhou, China, 2019.
- [4] M. Tyan, M. Kim, V. Pham, C. Choi, L. Nguyen and J. W. Lee, "Development of Advanced Aerodynamic Data Fusion Techniques for Flight Simulation Database Construction," in *AIAA Aviation Forum*, Atlanta, GA, 2018.
- [5] R. V. Jategaonkar, Flight Vehicle System Identification: A-Time Domain Methodology, 2 ed., vol. 245, T. C. Lieuwen, Ed., Reston, VA: American Institute of Astronautics and Aeronautics, Inc., 2015.
- [6] M. Millidere, M. Alam, S. Place and J. Whidborne, "Development of High-Fidelity Saab 340B Aerodynamic Model," in *AIAA SciTech Forum*, National Harbor, MD, 2023.
- [7] E. A. Morelli and V. Klein, Aircraft System Identification: Theory and Practices, 1 ed., Reston, VA: AIAA, 2006.
- [8] M. Millidere, U. Cakin and T. Yigit, "Generating Aerodynamic Database from Closed Loop Simulated Flight Data," in *European Conference for Aeronautics and Space Sciences*, Madrid, Spain, 2019.
- [9] M. Millidere, F. Akgül and K. Leblebicioğlu, "F16 Savaş Uçağının Boylamsal Aerodinamik Veritabanının Tanımlanması ve Doğrulanması," in *Ulusal Havacılık ve Uzay Konferansı*, Virtual Conference, 2021.

- [10] R. Monstein, P. Capone, M. Dettling and M. Vrdoljak, "Determination of Model Structure from Flight Test with Generalized Additive Models," *Journal of Aircraft*, vol. 56, no. 4, 2019.
- [11] J. B. van Ingen, C. C. de Visser and D. M. Pool, "Stall Model Identification of a Cessna Citation II from Flight Test Data Using Orthogonal Model Structure Selection," in *AIAA Scitech Forum*, Virtual Event, 2021.
- [12] J. A. Grauer and E. A. Morelli, "A New Formulation of the Filter-Error Method for Aerodynamic Parameter Estimation in Turbulence," in *AIAA Atmospheric Flight Mechanics Conference*, Dallas, TX, 2015.
- [13] M. Millidere, H. B. Kurt, H. Ballı and Ö. Uğur, "Kalman Based Neural Network Model Analysis with Resampling Methods for Longitudinal Aerodynamic Coefficient Estimation," in *AIAA Aviation Forum*, Virtual Conference, 2020.
- [14] K. Kirkpatrick, J. May and J. Valasek, "Aircraft System Identification Using Artificial Neural Networks," in *AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, TX, 2013.
- [15] G. Ghazi, M. Bosne, Q. Sammartano and R. M. Botez, "Cessna Citation X Stall Characteristics Identification from Flight Data using Neural Networks," in *AIAA Atmospheric Flight Mechanics Conference*, Grapevine, TX, 2017.
- [16] J. Harris, F. Arthurs, J. V. Henrickson and J. Valasek, "Aircraft System Identification using Artificial Neural Networks with Flight Test Data," in *International Conference on Unmanned Aircraft Systems*, Arlington, VA, 2016.
- [17] R. V. Jategaonkar and W. Mönnich, "Identification of DO-328 Aerodynamic Database for a Level D Flight Simulator," in *AIAA Modeling and Simulation Technologies Conference*, New Orleans, LA, 1997.
- [18] M. Millidere, U. Cakin and T. Yigit, "Generating Aerodynamic Database from Closed Loop Simulated Flight Data," in *European Conference for Aeronautics and Space Sciences*, Madrid, Spain, 2019.
- [19] R. V. Jategaonkar and F. Thielecke, "An Integrated Software Tool for Nonlinear Parameter Estimation," *Aerospace Science and Technology*, vol. 6, no. 8, pp. 565-578, 2002.
- [20] R. V. Jategaonkar, *Flight Vehicle System Identification: A-Time Domain Methodology*, Reston, VA: American Institute of Astronautics and Aeronautics, Inc., 2015.
- [21] E. A. Morelli and V. Klein, *Aircraft System Identification: Theory and Practices*, Reston, VA: AIAA, 2006.
- [22] G. James, D. Witten, T. Hastie and R. Tibshirani, *Introduction to Statistical Learning with Applications in R*, Springer, 2013.
- [23] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning-Data Mining, Inference, and Prediction*, Springer, 2009.
- [24] *MATLAB version 9.5.0.944444 (R2018b)*, Natick, Massachusetts: Mathworks, 2018.
- [25] J. R. Raol and J. Singh, *Flight Mechanics Modeling and Analysis*, CRC Press, 2008.
- [26] R. V. Jategaonkar, D. Fischenberg and W. v. Gruenhagen, "Aerodynamic Modeling and System Identification from Flight Data-Recent Applications at DLR," *Journal of Aircraft*, vol. 41, no. 4, 2004.
- [27] J. A. Grauer and E. A. Morelli, "Generic Global Aerodynamic Model for Aircraft," *Journal of Aircraft*, 2015.
- [28] R. V. Jategaonkar and F. Thielecke, "Aircraft Parameter Estimation - A Tool for Development of Aerodynamic Databases," *Indian Academy of Sciences*, vol. 25, pp. 119-135, 2000.
- [29] M. Millidere, H. B. Kurt, H. Ballı and Ö. Uğur, "Kalman Based Neural Network Model Analysis with Resampling Methods for Longitudinal Aerodynamic Coefficient Estimation," in *AIAA Aviation Forum*, Virtual Conference, 2020.
- [30] J. Grauer, "Aerodynamic Parameter Estimation Using Reconstructed Turbulence Measurements," *Journal of Aircraft*, 2021.
- [31] S. Mirjalili, *Evolutionary Algorithms and Neural Networks*, Springer International Publishing, 2019.
- [32] D. Wang, D. Tan and L. Liu, "Particle Swarm Optimization Algorithm: An Overview," *Soft Computing*, vol. 22, pp. 387-408, 2017.
- [33] T. Jiang, J. Li and K. Huang, "Longitudinal Parameter Identification of a Small Unmanned Aerial Vehicle Based On Modified Particle Swarm Optimization," *Chinese Journal of Aeronautics*, 2015.

- [34] J. Guan, W. Yi, S. Chang and X. Li, "Aerodynamic Parameter Estimation of a Symmetric Projectile Using Adaptive Chaotic Mutation Particle Swarm Optimization," *Mathematical Problems in Engineering*, 2016.
- [35] B. A. Garro and R. A. Vázquez, "Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms," *Computational Intelligence and Neuroscience*, 2015.
- [36] H. Wang, R. Ke, J. Li, Y. An, K. Wang and L. Yu, "A correlation-based binary particle swarm optimization method for feature selection in human activity recognition," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, 2018.
- [37] F. Koumi, M. Aldasht and H. Tamimi, "Efficient Feature Selection using Particle Swarm Optimization: A hybrid filters-wrapper Approach," in *10th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 2019.
- [38] M. Mafarja, R. Jarrar, S. Ahmad and A. A. Abusnaina, "Feature Selection Using Binary Particle Swarm Optimization with Time Varying Inertia Weight Strategies," in *International Conference on Future Networks and Distributed Systems (ICFNDS)*, Amman / Jordan, 2018.
- [39] J. Too, A. R. Abdullah and N. M. Saad, "A New Co-Evolution Binary Particle Swarm Optimization with Multiple Inertia Weight Strategy for Feature Selection," *Informatics*, vol. 6, no. 21, 2019.
- [40] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1-14, 2013.
- [41] A. R. Garcia, R. Vos and C. C. de Visser, "Aerodynamic Model Identification of the Flying V from Wind Tunnel Data," in *AIAA Aviation Forum*, Virtual Event, 2020.
- [42] P. Umapathy, C. Venkateshaiah and M. S. Arumugam, "Particle Swarm Optimization with Various Inertia Weight Variants for Optimal Flow Solution," *Discrete Dynamics in Nature and Society*, 2010.
- [43] J. Liu, Y. Mei and X. Li, "An Analysis of the Inertia Weight Parameter for Binary Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, 2015.
- [44] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon and A. Abraham, "Inertia Weight strategies in Particle Swarm Optimization," in *Third World Congress on Nature and Biologically Inspired Computing*, Salamanca, Spain, 2011.
- [45] K. C. Wang and K. W. Iliff, "Retrospective and Recent Examples of Aircraft Parameter Identification at NASA Dryden Flight Research Center," *Journal of Aircraft*, vol. 41, no. 4, 2004.
- [46] A. C. Paris and M. Bonner, "Nonlinear Model Development from Flight-Test Data for F/A-18E Super Hornet," *Journal of Aircraft*, vol. 41, no. 4, pp. 692-702, 2004.
- [47] E. A. Morelli, "Practical Aspects of the Equation-Error Method for Aircraft Parameter Estimation," in *AIAA Atmospheric Flight Mechanics Conference*, Keystone, CO, 2006.
- [48] L. Nguyen, M. Ogburn, W. P. Gilbert, K. S. Kibler, P. W. Brown and P. L. Deal, "Simulator Study of Stall/PostStall Characteristics of a Fighter Airplane with Relaxed Longitudinal Static Stability," NASA Technical Paper 1538, Hampton, VA, 1979.
- [49] B. Stevens, F. Lewis and E. Johnson, *Aircraft Control and Simulation Dynamics, Controls Design and Autonomous Systems*, Wiley-Blackwell, 2015.
- [50] F. R. Garza and E. A. Morelli, "Collection of Nonlinear Aircraft Simulations in MATLAB," NASA Technical Memorandum 212145, 2003.
- [51] H. B. Kurt, M. Millidere and E. Sezer, "Aerodynamic Database Simulation Implementation Based On Neural Network and Neural Network Parameter Selection Using Genetic Algorithms," in *European Conference for Aeronautics and Space Sciences*, Madrid, Spain, 2019.
- [52] M. Millidere, "Optimal Input Design and System Identification for an Agile Aircraft," Middle East Technical University, Ph.D. Thesis, Ankara, Turkey, 2021.



DSPACE

<https://dspace.org/>

SAAB 340B aerodynamic model development using binary particle swarm optimization: Aerodynamic modelling of Saab 340B development using binary particle swarm optimization

Millidere, Murat; Alam, Mushfiqul; Place, Simon; Whidborne, James F.

2024-01-04

American Institute of Aeronautics and Astronautics (AIAA)

<https://doi.org/10.2514/6.2024-1495>

Millidere M, Alam M, Place S, Whidborne J. (2024) SAAB 340B aerodynamic model development using binary particle swarm optimization. In: AIAA SCITECH 2024 Forum, 8-12 January 2024, Orlando, USA. Paper number AIAA 2024-1495