Aerospace Science and Technology 49 (2016) 52–62

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Modeling the fuel flow-rate of transport aircraft during flight phases using genetic algorithm-optimized neural networks

Tolga Baklacioglu [*]

Department of Aircraft Airframe and Engine Maintenance, Faculty of Aeronautics and Astronautics, Anadolu University, 26470 Eskisehir, Turkey

ARTICLE INFO

Article history:
Received 30 September 2014
Received in revised form 13 January 2015
Accepted 23 November 2015
Available online 30 November 2015

Keywords:
Aircraft
Fuel flow-rate prediction
Neural networks
Genetic algorithms
Learning algorithms

ABSTRACT

Predicting the fuel consumption of transport aircraft is vital for minimizing the detrimental effects of fuel emissions on the environment, saving fuel energy sources, reducing flight costs, achieving more accurate aircraft trajectory prediction, and providing effective and seamless management of air traffic. In this study, a genetic algorithm-optimized neural network topology is designed to predict the fuel flow-rate of a transport aircraft using real flight data. This model incorporates the cruise flight phase and the fuel consumption dependency with respect to both the variation of true airspeed and altitude. Feed-forward backpropagation and Levenberg–Marquardt algorithms are applied, and a genetic algorithm is utilized to design the optimum network architecture regarding time and effort. The predicted fuel flow-rates closely match the real data for both neural network training algorithms. Backpropagation gives the best accuracy for the climb and cruise phases, whereas the Levenberg–Marquardt algorithm is optimal for the descent phase.

1. Introduction

Because fuel rates constitute the largest portion of operational costs, airlines, aircraft manufacturers, and air traffic authorities attempt to implement procedures to reduce fuel consumption, such as continuous descent and tailored arrivals, and employ modeling approaches to predict the fuel flow-rate [1]. Conservation of fuel energy sources, as well as the reduction of detrimental effects of fuel emissions on the environment, can only be fulfilled in this way. Furthermore, the accomplishment of more accurate aircraft trajectory predictions, which will provide more effective management of air traffic, demands the development of better propulsive and fuel flow-rate prediction models [2].

Early attempts to estimate fuel consumption use Collins' model [3], which was also used in the FAA's Airport and Airspace Simulation Model, SIMMOD. Later, Trani et al. [4] developed a neural network (NN) model for the prediction of fuel consumption using the Fokker F-100 aircraft performance flight manual data. As noted by Senzig et al. [1], this model requires detailed aerodynamic information or a large database of airplane operations and associated airplane state data, meaning its acceptance was limited. While Trani et al.'s [4] NN model provided some accurate results; it requires improvement in terms of the input parameters and optimization of the modeling architecture. Trani et al. introduced the effect of a variable Mach number as an input parameter, and considered the initial and final altitudes, but not the effect of a variation in altitude. Additionally, the initial aircraft weight and temperature were also defined as input parameters. It should be noted that the specific air range, but not the fuel flow-rate, was given as an output parameter in their modeling of the cruise flight phase. They tested eight candidate topologies via a sensitivity analysis, and selected a three-layer model with eight neurons in the first two layers and one neuron in the output (third) layer. A trial-and-error method was used to select the number of layers and neurons, rather than an optimization for the NN model architecture [4]. Bartel and Young [5] developed a thrust-specific fuel consumption model for the cruise flight phase. Their model considered fuel consumption with respect to the Mach number, whereas the effect of the temperature ratio was excluded. The temperature term was replaced by the ratio of any cruise thrust to the cruise thrust in some reference condition. Senzig et al. [1] investigated a fuel consumption model with regards to the terminal area. Their proposed arrival thrust-specific fuel consumption algorithm was based on that of Hill and Peterson with modifications by Yoder. The temperature ratio effect, variation of Mach number, and net corrected thrust with respect to the pressure ratio were included in their model for the terminal area. Turgut and Rosen [6] recently proposed a fuel consumption model based on a ge-

* Tel.: +902222261457; fax: +902223221619.
E-mail address: tbaklacioglu@anadolu.edu.tr.

http://dx.doi.org/10.1016/j.ast.2015.11.031
1270-9638/© 2015 Elsevier Masson SAS. All rights reserved.

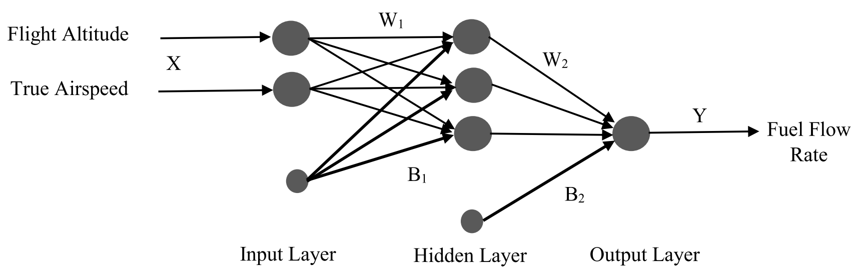

Fig. 1. Topology of a three-layer fuel flow rate FNN.

netic algorithm (GA) for the descent flight phase by considering the fuel flow-rate with respect to altitude. According to its user-manual, Eurocontrol's BADA (current version 3.12) [7] calculates the nominal fuel flow using the thrust and the thrust-specific fuel consumption, which is specified as a function of the true airspeed for all flight phases except the idle descent and cruise. The cruise fuel flow expression differs from the nominal fuel flow by an additional correction coefficient, whereas for the idle thrust descent condition, the fuel flow is expressed as a function of pressure, altitude, and descent fuel flow coefficients.

The GA-NN model derived in this study considers the variation of both altitude and true airspeed as input parameters, and provides the fuel flow-rate as an output parameter for climb, cruise, and descent flight phases. This model is developed using real raw flight data records of the medium weight transport aircraft type, Boeing 737-800 operating in Pegasus Airlines of Turkey in 2009. As a modeling approach, flight altitude and true airspeed variations are inserted as inputs to the model to prevent the need for extensive detailed information of airplane operations and data. This enables accurate fuel flow-rate modeling in a simplified manner. With regard to the optimization of the NN modeling architecture, the proposed model utilizes a GA method to determine the network input parameters and number of neurons in the hidden layer(s), thus achieving an optimal or nearly optimal model in a shorter time with less effort.

2. Neural networks

An NN is an information or signal processing system composed of a large number of simple processing elements, called neurons. These are interconnected by weighted links (or weighted connections) that cooperate to perform parallel distributed processing and solve computational tasks. NNs attempt to (at least partially) simulate the structure and functions of the brain and nervous system of living creatures [8]. Because the inputs of the first layer of neurons are connected to external data, this is called the input layer. Similarly, as the outputs of the last layer of neurons are the result of the total NN, this is called the output layer. All neuron layers between the input and output layers are called hidden layers, whose action cannot be seen directly from the outside. If the outputs of one layer are all connected to the input of the next layer, and there are no connections within the same layer or from a later layer back to an earlier layer, this is regarded as a feed-forward network [9]. Collecting the values from all of its input connections, each neuron produces a single output passing through an activation function [10]. In this paper, a multilayer feed-forward NN (FNN) structure is used for the proposed fuel flow-rate prediction model. A number of parameters, such as the number of layers and the type and number of units per layer, must be defined for any FNN. Adjusting the weights of the network to create the desired output constitutes the last design step. This process is called training the NN. A fuel flow rate FNN model with two inputs (flight altitude and true airspeed), three neurons, as an example, in a single hidden layer and one output (fuel flow rate) is shown in Fig. 1. Coefficients associated with the hidden layer (weights and biases) are grouped in matrices W_1 and B_1, whereas coefficients associated with the output layer are grouped in matrices W_2 and B_2. The output of the network can be expressed as

$$Y = f_2\big(W_2 f_1(W_1 X + B_1) + B_2\big) \tag{1}$$

where X is the matrix of the input variables, f_1 and f_2 are the activation functions in the hidden and output layers, respectively, and Y is the matrix of the output variables [11].

2.1. Backpropagation algorithm with momentum

Backpropagation (BP) is a well-known training algorithm. After measuring the output error, the BP algorithm then calculates the gradient of this error and adjusts the NN weights in the direction of descending gradient. As a gradient-descent local search technique, BP is highly accurate, but may also fall into local optima in complex search landscapes.

The squared error of the NN for a set of patterns can be defined as

$$E = \sum_{p=1}^{P} \sum_{i=1}^{N} \left(x_i^p - o_i^p\right)^2 \tag{2}$$

where E is the squared error of the NN, whereas x_i^p and o_i^p are the i^{th} components of the expected vector and the current output vector for the pattern p, respectively, N is the number of output neurons, and P is the number of patterns. The weights of the network define the actual value of the error function. Calculating the gradient of E, the BP algorithm updates the weights by moving them in the gradient descent direction, which can be expressed as

$$w_{ij}(t+1) = w_{ij}(t) - \eta \frac{\partial E}{\partial w_{ij}} \tag{3}$$

where $w_{ij}(t)$ and $w_{ij}(t+1)$ correspond to the network weights at the steps t and $t+1$, respectively. The parameter $\eta > 0$ is the *learning rate* controlling the learning speed. A momentum term is added to Eq. (3) to increase the stability of the search process:

$$w_{ij}(t+1) = w_{ij}(t) + m\Delta w_{ij}(t) - \eta \frac{\partial E}{\partial w_{ij}} \tag{4}$$

where $\Delta w_{ij}(t)$ shows the change in the weight w_{ij} at step t, and m is the *momentum rate*, which takes a value in the interval $[0, 1)$. When the error function does not change, this term accelerates the error minimization [12].

2.2. Levenberg–Marquardt algorithm

Being a variation of the Gauss–Newton (GN) method, the Levenberg–Marquardt (LM) algorithm is designed to minimize functions that are sums of squares of other non-linear expressions [13]. The LM algorithm acts as an intermediate optimization