



Machine learning of electro-hydraulic motor dynamics

Baronti, Luca; Zhang, Biao; Castellani, Marco; Pham, Duc

DOI:

[10.1007/s42452-019-1889-y](https://doi.org/10.1007/s42452-019-1889-y)

License:

Creative Commons: Attribution (CC BY)

Document Version

Publisher's PDF, also known as Version of record

Citation for published version (Harvard):

Baronti, L, Zhang, B, Castellani, M & Pham, D 2020, 'Machine learning of electro-hydraulic motor dynamics', *SN Applied Sciences*, vol. 2, 130. <https://doi.org/10.1007/s42452-019-1889-y>

[Link to publication on Research at Birmingham portal](#)

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.



Machine learning of electro-hydraulic motor dynamics

Luca Baronti¹ · Biao Zhang² · Marco Castellani¹ · Duc Truong Pham¹

Received: 20 August 2019 / Accepted: 10 December 2019

© The Author(s) 2019

Abstract

In this paper we propose an innovative machine learning approach to the hydraulic motor load balancing problem involving intelligent optimisation and neural networks. Two different nonlinear artificial neural network approaches are investigated, and their accuracy is compared to that of a linearised analytical model. The first neural network approach uses a multi-layer perceptron to reproduce the load simulator dynamics. The multi-layer perceptron is trained using the Rprop algorithm. The second approach uses a hybrid scheme featuring an analytical model to represent the main system behaviour, and a multi-layer perceptron to reproduce unmodelled nonlinear terms. Four techniques are tested for the optimisation of the parameters of the analytical model: random search, an evolutionary algorithm, particle swarm optimisation, and the Bees Algorithm. Experimental tests on 4500 real data samples from an electro-hydraulic load simulator rig reveal that the accuracy of the hybrid and the neural network models is comparable, and significantly superior to the accuracy of the analytical model. The results of the optimisation procedures suggest also that the inferior performance of the analytical model is likely due to the non-negligible magnitude of the unmodelled nonlinearities, rather than suboptimal setting of the parameters. Despite its limitations, the analytical linear model performs comparably to the state-of-the-art in the literature, whilst the neural and hybrid approaches compare favourably.

Keywords Regression · System identification · Load simulator · Electro-hydraulic motor · Analytical modelling · Optimisation · Multi-layer perceptron

1 Introduction

A load simulator [1] is a ground system that reproduces the aerodynamic force on the rudder of a flying aircraft. It is a typical example of passive loading servo system, and finds important application in hardware-in-the-loop simulator devices. It allows designers under strict laboratory conditions to foresee and detect problems related to the rudder mechanics and control policy, cutting down costs and development time.

There are different kinds of load simulators: electric, pneumatic, and electro-hydraulic. The latter excels in terms of durability, power to weight ratio, controllability, accuracy, and reliability [2]. Thanks to these advantages, electro-hydraulic load simulators found wide application

in the development and testing of systems beyond the aviation and aerospace domain, such as ship fin stabilisers [3], power steering for heavy duty vehicles [4], and robotic manipulators [5, 6].

To reproduce aerodynamic flow, the load simulator is typically connected with the actuation system via a rigid shaft. The direct coupling between the two systems produces a strong position disturbance on the load system. This generates an extra torque which needs to be compensated mechanically or via a control system [7–13].

For control and simulation purposes, it is often useful to identify the dynamics of the load simulator. That is, to model the relationship between the device state, control signal, and torque on the load bearing system (including the extra torque).

✉ Luca Baronti, LXB589@student.bham.ac.uk | ¹University of Birmingham, Birmingham B15 2TT, UK. ²School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, Heilongjiang Province, China.



Several studies described the dynamics of electro-hydraulic load simulators via analytical linearised models [14–18]. Although the structure of the linear relationship is to date fairly well understood, its exact parameterization is still uncertain. Some authors addressed the parameterization problem using a standard genetic algorithm [19], neural networks, [20], and a backstepping genetic algorithm [21]. Unfortunately, these methods either suffered from sub-optimal convergence [19], or required lengthy training procedures [20], or required the introduction of extra system parameters [21]. Moreover, the linearised model does not take into account known nonlinearities of electro-hydraulic load simulators [10], and this limits its precision and reliability. Unfortunately, due to its complexity, a full nonlinear analytical model of electro-hydraulic load simulator is at present not available.

Some authors employed machine learning techniques to either learn the electro-hydraulic motor dynamics [22, 23], or design robust controllers able to compensate parameter uncertainties and nonlinearities [24–26].

Artificial neural networks (ANNs) [27] are by far the most common machine learning method used for identification of electro-hydraulic load simulator dynamics. The typical approach is to complement the linear analytical model with an ANN module that reproduces the nonlinear dynamics [10, 28, 29]. Alternatively, some authors used an ANN to identify the full simulator dynamics [30]. Unfortunately, the above studies focused on the accuracy of the whole control system, and did not discuss the precision of the nonlinear model. This paper addresses this gap in the literature, and investigates the use of ANN techniques from a system identification perspective. In addition to the standard control applications, the proposed study is relevant for the general software simulation of electro-hydraulic load simulator systems.

Two system identification approaches are studied, using respectively (a) a full nonlinear ANN model, and (b) a hybrid model comprising an analytical component and an ANN component. The accuracy of these two approaches is compared to that of a standard analytical linearised model, and the differences are analysed. In all instances, the first effort will be to identify the structure (analytical, ANN) of the input-output relationship, and then learn the correct parameterization of the model from a set of data patterns describing the system input, state, and output. That is, to fit the parametric model to a set of data points sampled from an experimental rig at different frequencies.

The analytical linear model was developed by one of the authors (BZ). It will be employed on its own as a term of comparison for the two non-linear models, and as the linear component in the hybrid model. To optimise its parameters, three artificial intelligence techniques were tested: an evolutionary algorithm (EA) [31], particle swarm

optimisation (PSO) [32], and the Bees Algorithm (BA) [33]. The latter algorithm showed great promise on various optimisation benchmarks [33–35], and its application in this context constitutes a novel contribution to its knowledge. As a term of reference, *random guess* (RG) was also tested for the optimisation of the parameters of the linearised model.

The non-linear model of the the load simulator dynamics was built using a multi-layer perceptron (MLP) [27] ANN. In this case, the MLP architecture defines the structure of the model, and the connection weights are the parameters to be optimised. The strong points of the MLP approach are the possibility of reproducing arbitrarily complex non-linear mappings [36], and the existence of well-known in-built algorithms to optimise the weights inductively from data points. In this study, a fast weight learning algorithm [37] was used to reduce the ANN training time. The weak point of the MLP approach is the complexity of the ANN, which makes the model non-transparent to the user. That is, the MLP model is a black box.

In the hybrid approach, the analytical model was used to reproduce the main behaviour of the system, and the MLP to fit unmodelled nonlinearities.

The three modelling approaches are presented in Sect. 2, whilst Sect. 3 describes their optimisation. The experimental results are presented in Sect. 4 and discussed in Sect. 5. Section 6 concludes the paper.

2 Modelling approaches

The overall experimental system is described in Fig. 1. It consists of two subsystems connected by a rigid shaft and a torque sensor: the loading system (on the left in the figure) and load-bearing system (on the right). The overall

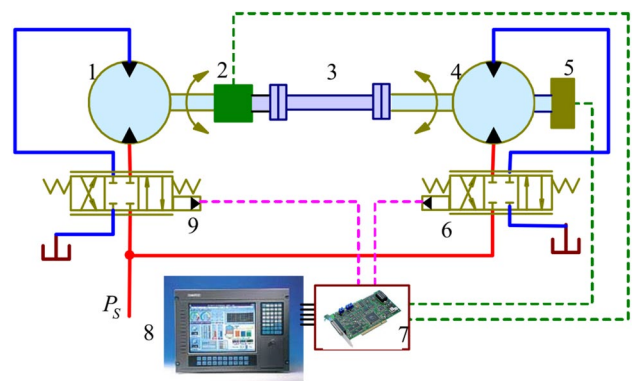


Fig. 1 Schematic diagram of the experimental rig. The components are: 1 load motor; 2 torque sensor; 3 coupling axis; 4 load bearing motor; 5 angle encoder; 6 load bearing valve; 7 AD/DA card; 8 computer; 9 load valve

stiffness of the connecting link includes the stiffness of the torque sensor.

The two subsystems are controlled using closed loop schemes. The angle of the load-bearing system is monitored using an encoder. The signal from the encoder is used by the load bearing controller to achieve servo position control. The load system controller uses angle and torque information to regulate the torque on the load bearing system. The remaining of this section describes the linear and non-linear models of the electro-hydraulic load simulator used in this study.

2.1 Linearised analytical model

This section briefly outlines the linear model that approximates the relationship at time k between the current plant output $\tilde{T}_g(k)$ (output torque of the load-on motor), its previous values, and the current and past values of the two plant inputs: $\theta_d(k)$ (turn angle of the motor drive shaft in radians), and $u_m(k)$ (load input from the controller). The derivation of the model is fully detailed in “Appendix 1”.

The linearised model consists of the following equation:

$$\begin{aligned} \tilde{T}_g(k) = & -f_1 T_g(k-1) + f_2 T_g(k-2) + f_3 T_g(k-3) \\ & - f_4 T_g(k-4) \\ & + d_0 u_m(k) + 4d_0 u_m(k-1) + 6d_0 u_m(k-2) \\ & + 4d_0 u_m(k-3) + d_0 u_m(k-4) \\ & - e_0 \theta_d(k) - e_1 \theta_d(k-1) + e_2 \theta_d(k-2) \\ & + e_3 \theta_d(k-3) - e_4 \theta_d(k-4) \end{aligned} \quad (1)$$

Equation (1) includes the following 14 independent variables: the torques at time steps $k-1, \dots, k-4$ (4 variables), load inputs u_m at time steps $k, \dots, k-4$ (5 variables), and angles θ_d at time steps $k, \dots, k-4$ (5 variables). The contribution of each of the 14 independent variables is weighted by a number of constant parameters that are detailed in “Appendix 1”, and depend on the 9 primary parameters listed in Table 1. The optimal value of these 9 parameters is not known. The boundaries showed in the

table were set based on expertise and the physical properties of the system.

2.2 Multi-layer perceptron model

The analytical model described in Sect. 2.1 approximates linearly the non-linear relationship between the output torque and plant inputs. Given the difficulty of representing such relationship, the linearised model offers a reasonable trade-off between modelling accuracy and mathematical complexity. An alternative approach to circumvent the difficulty of identifying the load simulator dynamics is to employ a non-linear neural network model.

Artificial neural networks (ANNs) [27] are widely used learning systems composed of an *input* layer of units (*neurons*), a variable number of *hidden* layers, and an *output* layer. The number of units per layer and their connectivity defines the ANN *topology*.

The multi-layer perceptron (MLP) [38] is arguably the best known ANN model. It is composed of three or four layers of units. Each element is fully connected to all the units in the immediately previous and following layer, with no inter-layer connections between units. Each connection is associated to a weight, which allows modulating the signal passed from one unit to the other. The plant inputs are collected by the first (input) layer and processed in feedforward manner, one layer after the other (*feed-forward* model). The input layer is composed of as many units as the plant inputs, and acts as a buffer. The units of the hidden layer(s) perform a non-linear mapping of the weighted sum of their inputs (i.e. the outputs of the previous layer). Depending on the implementation, the units of the last (output) layer perform a linear or non-linear mapping of the weighted sum of their inputs (i.e. the outputs of the last hidden layer). The output of the units of the last layer constitutes the ANN response to a given input pattern. In the specific case of the hydraulic motor plant, there is only one output unit giving the motor output torque. The transfer function of each unit, which determine the

Table 1 Analytical model parameters with boundaries

Parameter	Meaning	Unit	Lower bound	Upper bound
V_m	Motor volume	m^3	1.6×10^{-4}	1.9×10^{-4}
J	Gen. equivalent motor inertia	kg m^2	1×10^{-2}	1×10^{-1}
K_p	Valve coefficient	–	5×10^{-3}	3×10^{-2}
K_q	Valve flow gain	m^2/s	2×10^{-1}	2.5
G	Gen. coupled stiffness	N m/rad	7×10^3	1.5×10^4
K_{ce}	Flow pressure coefficient	$\text{m}^5/\text{N s}$	1×10^{-12}	1×10^{-11}
B_m	Viscosity damp coefficient	rad/s	10	1×10^2
K_{sv}	Valve coefficient	–	3×10^{-5}	2×10^{-4}
β_e	Bulk modulus of oil	N/m^2	6.5×10^8	7.5×10^8

unit response at a weighted input, is the hyperbolic tangent function (*tanh*).

Given a sufficient number of non-linear hidden units, an MLP is able to approximate any continuous mapping to an arbitrary degree of precision [39]. The ANN response is fitted to the input-output data patterns by adjusting the weights of the unit connections. The supervised learning approach changes (*trains*) the weights based on the error between the ANN and the actual plant output. In the dataset, the values of the plant output have been divided by a factor 200. This way, all the actual output values present in the dataset are in the $[-1, 1]$ range, and they can be compared with the MLP output.

The MLP is fed to the same 14 inputs used by the analytical model (Eq. 1), and is required to approximate the torque $T_g(k)$ at time step k .

2.3 Hybrid model

A common divide-and-conquer strategy is to combine the two models described in Sects. 2.1 and 2.2 into one hybrid model (HM) [40, 41]. In detail, the analytical linearised model described in Eq. (1) is first optimised. In absence of noise, the error $\varepsilon(t)$ of the optimised analytical model consists of the unmodelled non-linear dynamics of the system. That is, for each pattern p , $\varepsilon(p)$ is equal to the difference between the desired value $T_g(p)$ and the analytical model output $\tilde{T}_g(p)$:

$$\varepsilon(p) = T_g(p) - \tilde{T}_g(p) \quad (2)$$

Rearranging (2), the desired output $T_g(p)$ can be expressed as:

$$T_g(p) = \tilde{T}_g(p) + \varepsilon(p) \quad (3)$$

An MLP is then trained to reproduce the analytical model error ($MLP(p) = \varepsilon(p)$, where $MLP(p)$ is the MLP output). Once the MLP has learned to reproduce the error, the plant dynamics are obtained substituting the MLP output in (3).

$$T_g(p) = \tilde{T}_g(p) + MLP(p) \quad (4)$$

3 Optimisation of models

The experimental rig shown in Fig. 1 was used to generate three sets (series) of data, each containing 1500 samples recorded over a 3 s span (2 ms sampling frequency). Each data sample consists of two input features and the corresponding model output, respectively the load input from the controller at time step $t - 1$ ($u_m(t - 1)$), the load

bearing position (angle) at time step $t - 1$ ($\theta_d(t - 1)$), and the output torque of the load-on motor at time step t ($T_g(t)$). The three sets differ by the frequency of the load input u_m generated by the controller, which is a sinusoidal signal of respectively 1, 7 and 9 Hz. The measured output torque is always within the range $T_g \in (-150, 150)$ N m, except for one outlier at $T_g = 194$ N m.

The test set is composed of 1500 data patterns. It was formed concatenating the first 500 instances of each data series (1, 7 and 9 Hz). The training set contains the remaining 3000 data patterns (1000 patterns from each of the three sets). The difficulty of the data series is uneven, as some high-frequency disturbances are present in a small section of the training set.

The three models predict the output torque using information (θ_d , u_m , $T_g(t)$) from the current and the previous four time steps (Eq. 1). For this reason, the first five time steps of each data set are used only as inputs to the model. That is, for each data set, 995 and 495 data instances are used respectively for training and testing the models.

3.1 Optimisation of the analytical model

For this task, three different global optimisation techniques were tested: EA, BA and PSO. These techniques use different nature-inspired population-based metaheuristics. All three algorithms use the same representation scheme, fitness evaluation function, initialisation procedure, and are parameterized to sample an equal number of solutions in the search space.

Namely, candidate solutions are encoded as real-valued numerical strings composed of the 9 parameters listed in Table 1. The fitness F of a candidate solution is calculated as the *Root Mean Square* (RMS) error of the model, that is:

$$F = \sqrt{\sum_{i=1}^N \frac{(T_g(i) - M(i))^2}{N}} \quad (5)$$

where N is the number of data samples (995 for training, 495 for final testing), $T_g(i)$ is the desired (plant) output for data pattern i , and $M(i)$ is the model output for data pattern i . The initialisation procedure randomly draws the parameter values of the candidate solutions with uniform probability from the range of values defined in Table 1.

In order to obtain the same sampling of the search space, the EA, PSO, and BA used the same number of individuals and were run for the same number of learning cycles. Preliminary experiments showed fast convergence of the three optimisation procedures. For this reason, each run was limited to 100 generations, and the population size set to 100 individuals. These figures correspond to a

total of 104 candidate solutions tested per optimisation procedure.

The above conditions ensure that the three algorithms search the same solution landscape (encoding and fitness function), start from comparable states (initialisation procedure), and are given the same search opportunities (equal sampling opportunities). To define a baseline of performance for the optimisation algorithms, RG was also implemented. RG uses the same encoding scheme, fitness evaluation function, and initialisation procedure as the EA, BA, and PSO algorithms, and is allowed the same sampling opportunities. RG randomly picks candidate solutions from the search space with uniform probability, and retains the best solution encountered. A summary of the parameters used in the optimization algorithms is present in Table 2.

3.1.1 Evolutionary algorithm

The EA [31] used in this study employed the *Roulette Wheel* [31] selection procedure to allocate the reproduction opportunities proportionally to the fitness of the solutions, and generational replacement with elitism [31] to renew the population. Genetic mutations were simulated by adding some 'noise' to each gene. This noise was randomly sampled from a normal distribution centred in zero. That is:

$$og_i(j) = pg_i(j) + K_i \cdot N(0, \sigma_j) \quad (6)$$

where $og_i(j)$ is the i th gene of offspring j , $pg_i(j)$ is the i th gene of parent j , and $N(0, \sigma_j)$ is a random number sampled from a normal probability distribution of mean 0 and variance σ_j . The variance σ_j is individual specific; it is initialised at 1 and undergoes adaptation (via mutation) like the other genes. The parameter K_i rescales the magnitude of the mutation to the range of parameter i according to the following formula:

$$K_i = 0.1 \cdot (par_i^{max} - par_i^{min}) \quad (7)$$

where par_j^{max} and par_j^{min} are respectively the upper and lower bound of the i th parameter (see Table 1).

Cycles of selection and mutation are repeated until a given number of iterations has elapsed. Due to the lack

of crossover and the adaptation of the mutation width, the EA employed in this study is close to the Evolutionary Programming paradigm [31]

3.1.2 Particle swarm optimisation

A standard Particle swarm optimiser [32] was employed in this study. At each iteration, the position (\mathbf{x}) and velocity (\mathbf{v}) of each particle i were updated according to the following equations:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \quad (8)$$

$$\mathbf{v}_i(t+1) = \alpha \cdot \mathbf{v}_i(t) + \beta \cdot \rho_1 \cdot [\mathbf{PB}_i(t) - \mathbf{x}_i(t)] + \gamma \cdot \rho_2 \cdot [\mathbf{NB}_i(t) - \mathbf{x}_i(t)] \quad (9)$$

where α , β and γ are fixed parameters, ρ_1 and ρ_2 are random numbers drawn with uniform probability in the interval [0,1], \mathbf{PB} is the best solution so far encountered by the particle, \mathbf{NB} is the best solution so far encountered by the social neighbours of the particle, and t is the current iteration of the optimisation procedure. In this work the whole population was considered connected. That is, the number of social neighbours of an individual corresponded with the population size (100 individuals). The standard values $\alpha = 1$, $\beta = 2$, $\gamma = 2$ were used.

3.1.3 Bees Algorithm

The standard version [33] of the Bees Algorithm was used in this study. The experiments were performed setting the number of elite sites $ne = 2$, the number of best sites $nb = 6$, and allocating $nre = 20$ and $nrb = 10$ foragers to perform local exploitative search respectively in the elite and best sites. The parameter for site abandonment was set to $slim = 10$.

3.2 Optimisation of neural network model

The MLP was trained using the Rprop [37] algorithm. Rprop is a fast gradient descent technique that uses only the sign of the gradient, ignoring its module. Rprop increases the step size of the adjustment of a given weight if the partial

Table 2 Parameters used in the optimization algorithms of the analytical model

EA		PSO		BA	
Population size	100	Population size	100	Elite sites	2
Mutation rate	1	Connectivity	100	Best sites	6
Mutation σ	Adaptive	α	1	Recruiting elite	20
Initial mutation σ	1	β	2	Recruiting best	10
–	–	γ	2	Stagnation limit	10
Learning cycles	100	Learning cycles	100	Learning cycles	100

error derivative along that weight keeps the same sign from the previous iteration, and reduces it otherwise.

The model inputs (θ_d and u_m) were normalised using the mean–variance procedure:

$$\theta_d(t) = \frac{\theta_d(t) - \mu_{\theta_d}}{3 \cdot \sigma_{\theta_d}} \quad u_m(t) = \frac{u_m(t) - \mu_{u_m}}{3 \cdot \sigma_{u_m}} \quad (10)$$

where μ_v and σ_v are respectively the mean and standard deviation of the input $v = \theta_d, u_m$.

The MLP structure was experimentally optimised, varying the number of hidden layers (1 or 2) and units per hidden layer (from 1 to 100). The weight decay parameter of the Rprop algorithm was also experimentally optimised. The following procedure was used to find the best ANN parameters. The training set (3000 patterns) was split into one training subset containing 2000 patterns, and one validation subset containing the remaining 1000 patterns. Different instances of MLP structures were trained on the training subset, and their performance was evaluated on the validation subset. For each parameter setting, the training procedure was repeated 20 times, and the results averaged. At the end, the parameter setting that provided the best results on the validation set was retained as the 'optimal' one. If the difference in the results obtained by two different settings was not statistically significant (Mann–Whitney test), the setting that minimised the MLP structure was preferred.

Using the above procedure, the MLP structure of the non-linear model was optimised to 2 hidden layers of 77 units each. The weight decay parameter was set to 5×10^{-4} .

Once the MLP structure had been optimised, 20 different instances of the ANN were trained on the whole training set, and the final performance evaluated on the test set. Since the ANN was optimised and trained using only data from the training set, the results of the final evaluation on the test set are guaranteed to be unbiased.

Due to the relatively fast convergence shown in the preliminary experiments, the training procedure was stopped after 500 epochs.

3.3 Optimisation of the hybrid model

The hybrid model used the best analytical linear model obtained in Sect. 3.1. An MLP was optimised and trained to compensate (i.e. reproduce) the error $\epsilon(t)$ (Eq. 2) of the analytical model.

The procedure described in Sect. 3.2 was used to optimise and train the ANN. The optimal MLP structure for the hybrid model was found to be one hidden layer of 80 hidden units, with no weight decay in the Rprop procedure.

A summary of the parameters used in the MLP and the Hybrid Model can be found in Table 3.

4 Experimental results

The results presented in Tables 4 and 5 show the five number summary of the RMS error (5) attained by the different models on the training and test set. For each optimisation procedure, the tables summarise the results of 20 independent runs. The learning curves are plotted in Fig. 2. Pairwise two-sided Mann–Whitney tests were performed to assess the statistical significance of the differences among the results obtained. The significance level was set to 5%, and the p values are shown in Tables 6 and 7.

Overall, all algorithms gave very consistent results in the optimisation of the analytical model. The BA attained the smallest RMS error on the test set, whilst the PSO gave the most consistent performances. The pairwise Mann–Whitney tests (Tables 6, 7) revealed that the performance of the BA is significantly better than the performance of the other algorithms. In all cases, the RMS error on the test set is smaller than the RMS error on the training set. This result is due to the fact that the analytical model was not able to reproduce the high-frequency disturbances present in a small section of the training set.

The MLP achieved significantly higher accuracy results than the analytical model. The measures of the RMS error obtained on the training and test sets are comparable (Fig. 3), indicating that the ANN model was able to reproduce the high-frequency disturbances in the training set.

According to the results of pairwise Mann–Whitney tests (Tables 6, 7), the modelling accuracies obtained using the Hybrid Model are statistically comparable to those obtained by the MLP model.

5 Discussion

Three different approaches were tested to reproduce the dynamics of an electro-hydraulic load simulator. These approaches fit the model parameters to a set of experimental data points, measured at three different frequencies of the load input.

Table 3 Parameters used in the MLP and hybrid model

	MLP	HM
Layers	2	2
Hidden units per layer	77	80
Weight decay par.	5×10^{-4}	0
Epochs	500	500

Table 4 Results of the different optimization techniques and alternative models on the training set, based on 20 runs

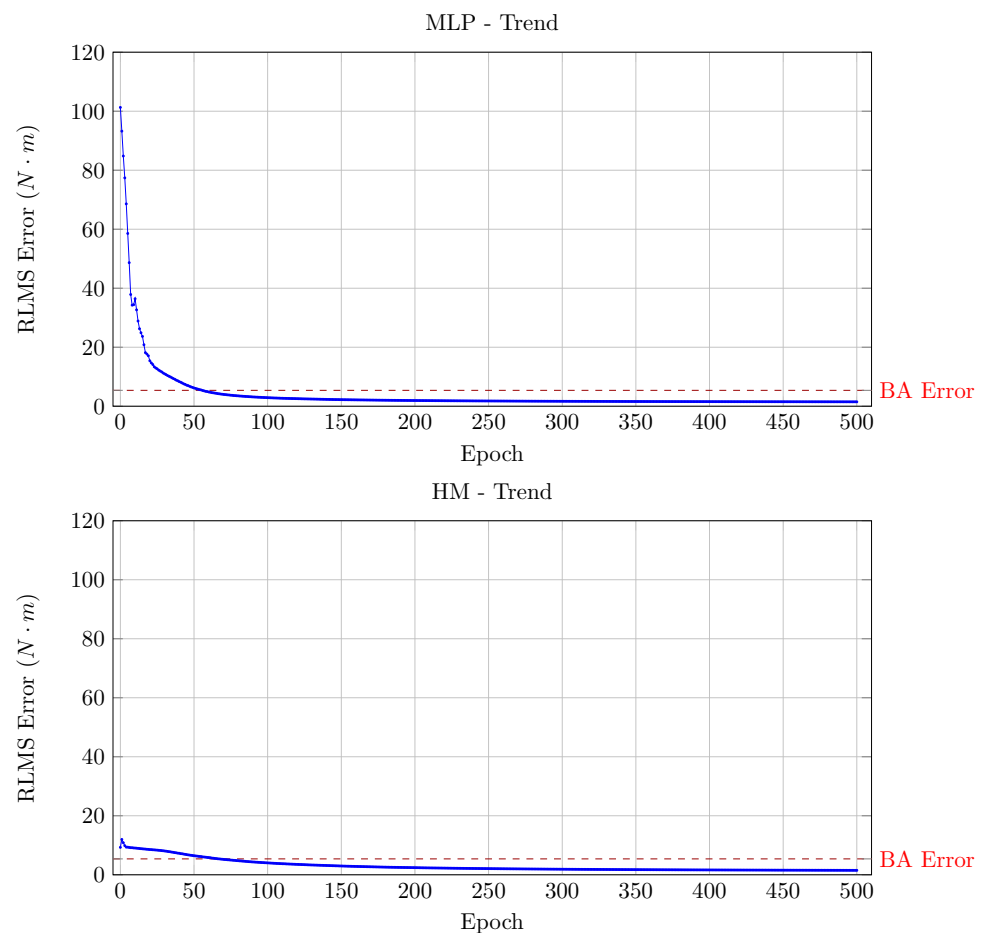
	RG	EA	BA	PSO	MLP	HM
Min	9.47868	9.30164	9.32114	9.30119	1.15892	1.2632
1th q.	9.57782	9.303415	9.3295325	9.30119	1.3777125	1.435455
Median	9.68556	9.30649	9.346555	9.30119	1.45774	1.482695
3th q.	9.7535625	9.3091775	9.3608475	9.30119	1.5018875	1.5398075
Max	9.87132	9.31923	9.42562	9.31689	1.63432	1.89633

The best result is highlighted in bold

Table 5 Results of the different optimization techniques and alternative models on the test set, based on 20 runs

	RG	EA	BA	PSO	MLP	HM
Min	5.44198	5.41832	5.3362	5.44812	1.27542	1.26246
25 perc	5.516595	5.436605	5.373755	5.44812	1.449975	1.434335
Median	5.573365	5.44923	5.38558	5.44812	1.50684	1.48187
75 perc	5.6261375	5.466785	5.4152225	5.44812	1.5319175	1.5379825
Max	6.26452	5.52698	5.55534	5.52491	1.62572	1.89453

The best result is highlighted in bold

Fig. 2 Learning curves for the MLP and HM, averaged on 20 runs. The RMS error is monitored on the test set. The best result achieved with the analytical model (by the BA) is shown for comparison

The MLP-based non-linear model outperformed the baseline linearised analytical model. The RMS error of the non-linear model on the test data corresponds to about

0.5% of the output variable range, and is about 3.5 times smaller than the error of the linearised model. This result confirms that the non-linear terms in the load simulator

Table 6 Mann–Whitney values computed on the training results

	RG	EA	BA	PSO	MLP	HM
RG	1	6.30×10^{-8}	6.30×10^{-8}	6.30×10^{-8}	6.30×10^{-8}	6.30×10^{-8}
EA	–	1	6.30×10^{-8}	5.35×10^{-4}	6.30×10^{-8}	6.30×10^{-8}
BA	–	–	1	6.30×10^{-8}	6.30×10^{-8}	6.30×10^{-8}
PSO	–	–	–	1	6.30×10^{-8}	6.30×10^{-8}
NN	–	–	–	–	1	2.13×10^{-1}
HM	–	–	–	–	–	1

Table 7 Mann–Whitney values computed on the test results

	RG	EA	BA	PSO	MLP	HM
RG	1	6.30×10^{-8}	6.30×10^{-8}	6.30×10^{-8}	6.30×10^{-8}	6.30×10^{-8}
EA	–	1	2.34×10^{-4}	5.88×10^{-1}	6.30×10^{-8}	6.30×10^{-8}
BA	–	–	1	3.56×10^{-4}	6.30×10^{-8}	6.30×10^{-8}
PSO	–	–	–	1	6.30×10^{-8}	6.30×10^{-8}
NN	–	–	–	–	1	4.81×10^{-1}
HM	–	–	–	–	–	1

dynamics have a significant impact on the accuracy of the model.

The four algorithms (RG, EA, PSO, and BA) used to optimise the baseline analytical model gave similar results: the difference between the most (BA) and least (RG) accurate model obtained is extremely small, amounting to about 3% of the RMS error magnitude. This result indicates that the current error is due to the limits of the linearised approach, rather than inadequacies of the optimisation procedures tested.

The RMS error of the linearised model is higher on the training set of examples, where some short high frequency disturbances are included. The results obtained using the MLP model on the training and test set are comparable. This suggests that the effect of the non-linear terms on the overall system dynamics is larger at high frequencies of the load input signal. In one further test, the training and test set were swapped. That is, the ANN was trained on the test set (not containing the high frequency disturbances) and its performance was validated on the training set (containing the high frequency disturbances). In this case (results not shown), there was a sharp drop in performance between the training accuracy and the test accuracy. The results of this test indicate that the high frequency disturbances contain non-linearities that are not present at low frequency.

Combining the analytical and MLP approach did not bring benefits in terms of modelling accuracy. The difference between the RMS error attained by the hybrid and MLP model is not statistically significant. Overall, the

experimental results indicate that the non-linear MLP model is preferable in terms of simplicity and accuracy.

Comparing the proposed approaches with the existing literature is difficult because of the scarcity of quantitative results published, and the different data sets used. Perhaps the most similar tests were carried out by Zhang and Dong [21], who reported a tracking error within 5% of the load-on motor output torque. However, the present study is the only one where the model is optimised and tested at different load input frequencies. Conservatively, it can be said that the accuracy of the linearised model presented in this study is comparable to the accuracy of the state-of-the-art in the literature. The MLP-based model and the HM appear to obtain error magnitudes one order of magnitude smaller than the state-of-the-art in the literature.

6 Conclusions

The accuracy of analytical linear models of electro-hydraulic load simulators is constrained by the extent of unmodelled nonlinearities, and the difficulty of optimising the numerous parameters that characterise them. A small number of nonlinear models has been reported in the literature, mainly based on ANN approaches. These ANNs were part of larger control systems, and whilst the performance of the whole systems was described, no information was given on the actual accuracy of the models.

This study used experimental data describing the dynamics of an electro-hydraulic motor, and fitted to these data the parameters of three kinds of models:

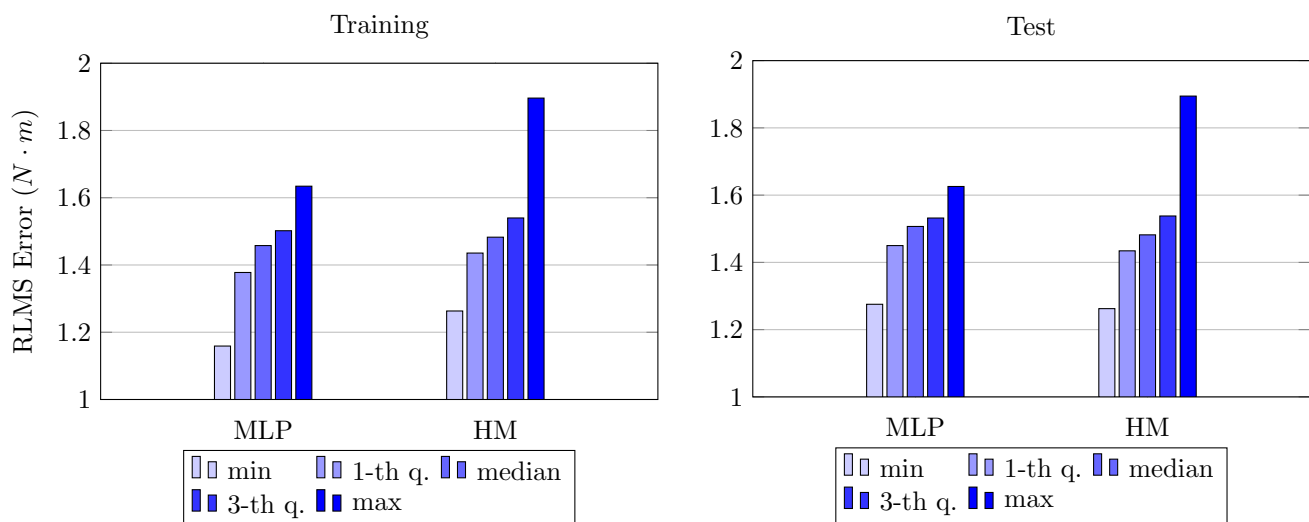


Fig. 3 Training and test error of the MLP and HM, in the form of the five value summary relative of 20 runs

analytical, ANN, and hybrid. The study made the following contributions:

- the upper limit of the accuracy of the analytical linear model was estimated using four intelligent parameter optimisation methods.
- the difference in accuracy between the linear and nonlinear models was quantified and shown to be significant
- it was shown a pure ANN model can be trained with accuracy comparable to a hybrid analytical + ANN model

To the best of the authors' knowledge, this study is to date the most exhaustive attempt at identifying the dynamics of an electro-hydraulic load simulator, involving different kinds of linear and non-linear models and parameter optimisation algorithms. Despite its limitations, the accuracy of the optimised analytical model is comparable with the state-of-the-art linear models in the literature. The MLP and HM outperform the best models reported in the literature. Given the advantages of electro-hydraulic load simulators in terms of reliability, durability, and performances, the results of this study are deemed significant for the understanding and simulation of such systems. Further work should extend the experimental tests to higher frequencies of the load input, and more complex (e.g. recurrent [27]) ANN structures.

Funding The authors received no financial support for the research, authorship, and/or publication of this article.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendices

Linearised analytical model: full description

The load system is driven by a servo valve, which is used to control the output torque. The flow equation of the servo valve of the load system is:

$$Q_L = K_q x_v - K_c p_L \quad (11)$$

where Q_L (m^3/s) is the load flow of the servo valve; K_q (m^3/s) is the flow of the gain of the servo valve; x_v (m) is the opening of the servo valve spool; K_c ($\text{m}^5/\text{N s}$) is the flow-pressure coefficient; and p_L (N/m^2) is the load pressure of the load-on motor;

The flow continuity equation of the load-on motor is:

$$Q_L = D_m \theta_j' + C_{tm} p_L + \frac{V_m}{4\beta_e} p_L' \quad (12)$$

where D_m (m^3/rad) is the theoretical displacement of the load-on motor; θ_j (rad) is the rotational angle of the load-on motor; C_{tm} ($\text{m}^5/\text{N s}$) is the internal leakage coefficient of the load-on motor; V_m (m^3) is the total chamber volume of the load-on motor and pipes; and $[\beta_e]$ (N/m^2) is the effective bulk modulus;

Ignoring friction, the torque balance equation of the load-on motor is:

$$p_L D_m = J \theta_j'' + B_m \theta_j' + T_g \quad (13)$$

where P_L is the load pressure; J is the equivalent moment of inertia; T_g (N m) is the output torque of the load-on motor; and $[B_m]$ ($\text{N m}/\text{rad/s}$) is the viscous damping coefficient;

The output torque equation is:

$$T_g = G(\theta_j - \theta_d) \quad (14)$$

where G ($\text{N m}/\text{rad}$) is the rigidity of the connecting link; θ_d (rad) is the turn angle of the motor drive shaft;

The spool displacement of the servo valve and the input voltage satisfy the following relational expression:

$$K_p x_v' + x_v = K_{sv} u_m \quad (15)$$

where u_m is the load input from the controller, and K_{sv} and K_p are the correlation coefficients of the servo valve.

Combining the Laplace transforms of equations Eqs. (11) to (15), the following equation is obtained:

$$\sum_{i=1}^5 a_i T_g s^{i-5} = K_{sv} u_m - G a_1 \theta_d s^4 - G a_2 \theta_d s^3 - b_1 \theta_d s^2 - b_2 \theta_d s \quad (16)$$

where

$$\begin{aligned} a_1 &= \frac{V_m J K_p}{4\beta_e D_m K_q G} \\ a_2 &= \frac{4\beta_e K_p K_{ce} J + V_m (K_p B_m + J)}{4\beta_e D_m K_q G} \\ a_3 &= \frac{4\beta_e (K_p D_m^2 + K_{ce} (K_p B_m + J)) + V_m (K_p G + B_m)}{4\beta_e D_m K_q G} \\ a_4 &= \frac{4\beta_e (G K_p K_{ce} + D_m^2 + K_{ce} B_m) + G V_m}{4\beta_e D_m K_q G} \\ a_5 &= \frac{K_{ce}}{D_m K_q} \end{aligned} \quad (17)$$

$$\begin{aligned} b_1 &= \frac{4\beta_e (K_p D_m^2 + K_{ce} (K_p B_m + J)) + V_m B_m}{4\beta_e D_m K_q} \\ b_2 &= \frac{D_m^2 + K_{ce} B_m}{D_m K_q} \end{aligned} \quad (18)$$

and

$$K_{ce} = K_c + C_m \quad (19)$$

In order to be able to identify the model using the discrete dataset of experimental measurements, Eq. (19) needs to be transformed into discrete form. This is achieved by performing a bilinear transformation of the following equation:

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (20)$$

where T is the sampling interval and s is the complex variable. The final discrete equation is obtained:

$$\begin{aligned} \tilde{T}_g(k) &= -f_1 T_g(k-1) + f_2 T_g(k-2) + f_3 T_g(k-3) \\ &\quad - f_4 T_g(k-4) \\ &\quad + d_0 u_m(k) + 4d_0 u_m(k-1) + 6d_0 u_m(k-2) \\ &\quad + 4d_0 u_m(k-3) + d_0 u_m(k-4) \\ &\quad - e_0 \theta_d(k) - e_1 \theta_d(k-1) + e_2 \theta_d(k-2) \\ &\quad + e_3 \theta_d(k-3) - e_4 \theta_d(k-4) \end{aligned} \quad (21)$$

where

$$\begin{aligned}
 f_0 &= \sum_{i=1}^5 a_i \left(\frac{2}{T}\right)^{5-i} \\
 f_1 &= \frac{1}{f_0} \sum_{i=1, i \neq 3}^5 (-4 + 2(i-1)) a_i \left(\frac{2}{T}\right)^{5-i} \\
 f_2 &= \frac{1}{f_0} \left(-6a_1 \frac{2^4}{T^4} + 2a_3 \frac{2^2}{T^2} - 6a_5 \right) \\
 f_3 &= \frac{1}{f_0} \sum_{i=1, i \neq 3}^5 (-1)^{i-1} (4 - 2(i-1)) a_i \left(\frac{2}{T}\right)^{5-i} \\
 f_4 &= \frac{1}{f_0} \sum_{i=1}^5 (-1)^{i-1} a_i \left(\frac{2}{T}\right)^{5-i}
 \end{aligned} \quad (22)$$

the parameter d_0 is:

$$d_0 = \frac{K_{sv}}{f_0} \quad (23)$$

and the parameters e_i are:

$$\begin{aligned}
 e_0 &= \frac{1}{f_0} G \sum_{i=1}^2 a_i \left(\frac{2}{T}\right)^{5-i} + \frac{1}{f_0} \sum_{i=1}^2 b_i \left(\frac{2}{T}\right)^{3-i} \\
 e_1 &= \frac{4}{f_0 T} b_2 - \frac{1}{f_0} G \sum_{i=1}^2 \frac{4}{i} a_i \left(\frac{2}{T}\right)^{5-i} \\
 e_2 &= -\frac{6}{f_0} G a_1 \left(\frac{2}{T}\right)^4 + \frac{2}{f_0} b_1 \left(\frac{2}{T}\right)^2 \\
 e_3 &= \frac{4}{f_0 T} b_2 + \frac{1}{f_0} G \sum_{i=1}^2 (-1)^{i-1} \frac{4}{i} a_i \left(\frac{2}{T}\right)^{5-i} \\
 e_4 &= \frac{1}{f_0} G \sum_{i=1}^2 (-1)^{i-1} a_i \left(\frac{2}{T}\right)^{5-i} + \frac{1}{f_0} \sum_{i=1}^2 (-1)^{i-1} b_i \left(\frac{2}{T}\right)^{3-i}
 \end{aligned} \quad (24)$$

References

- Yalla SK, Kareem A (2007) Dynamic load simulator: actuation strategies and applications. *J Eng Mech* 133.8:855–863
- Wang CJZ, Wu S (2014) A practical nonlinear robust control approach of electro-hydraulic load simulator. *Chin J Aeronaut* 27.3:735–744
- Liang L-h (2009) Research on the performance of electro-hydraulic load simulator in fin stabilizer. MA thesis. Harbin Engineering University, pp 5–12
- Sato Y, Hirano K, Tanaka H (2013) Performance test of an AC servomotor drive valveless hydraulic power steering by using a passive type electro-hydraulic load simulator. *Trans Jpn Fluid Power Syst Soc* 44.4:75–80
- Temeltas H, Gokasan M, Bogosyan OS (2001) A nonlinear load simulator for robot manipulators. In: The 27th annual conference of the IEEE industrial electronics society, 2001. IECON'01, vol 1. IEEE, pp 357–362
- Plummer AR (2007) Robust electrohydraulic force control. *Proc Inst Mech Eng Part I J Syst Control Eng* 221.4:717–731
- Nam Y (2001) QFT force loop design for the aerodynamic load simulator. *IEEE Trans Aerosp Electron Syst* 37.4:1384–1392
- Li G-q et al (2006) Design of robust controller in electrohydraulic load simulator. In: 2006 international conference on machine learning and cybernetics. IEEE, pp 779–784
- Wang X, Liu W (2007) Neuralnetwork internal feedback control for electrohydraulic servo loading. *ACTA Aeronaut Astronaut Sinica Ser A, B* 28.3:690
- Wang M et al (2003) Design of electric dynamic load simulator based on recurrent neural networks. In: Electric machines and drives conference, 2003. IEMDC'03. IEEE International, vol 1. IEEE, pp 207–210
- Truong DQ, Ahn KK (2011) Parallel control for electro-hydraulic load simulator using online self tuning fuzzy PID technique. *Asian J Control* 13.4:522–541
- Li G-Q et al (2012) Research on decoupling control for electro-hydraulic load simulator. In: The 2012 international conference on advanced mechatronic systems. IEEE, pp 473–477
- Han S et al (2014) Compound velocity synchronizing control strategy for electrohydraulic load simulator and its engineering application. *J Dyn Syst Meas Control* 136.5:051002
- Su YQ, Zhao KD, Su DH (1995) Application of fuzzy adaptive control in syncho-compensation for electro-hydraulic loading simulator. In: International symposium on fluid power transmission and control. Harbin, China, pp 269–272
- Jiao Z-X et al (2004) The velocity synchronizing control on the electro-hydraulic load simulator. *Chin J Aeronaut* 17.1:39–46
- Li GQ, Zhao KD, Yuan RB (2005) Study on the effect of generalized coupled stiffness in the electro-hydraulic load simulator. In: Proceedings of the international conference on mechanical engineering and mechanics, Nanjing, China, pp 1137–1141
- Yao J et al (2012) Nonlinear adaptive robust force control of hydraulic load simulator. *Acta Aeronaut Astronaut Sinica* 25.5:766–775
- Jing C, Xu H (2016) A study on the influence of friction on loading performance of electro-hydraulic friction load simulator for actuator test. In: 2016 7th international conference on mechanical and aerospace engineering (ICMAE). IEEE, pp 524–528
- Zhang B et al (2008) Parameters identification of passive force control system based on genetic algorithm. In: 2008 27th Chinese control conference. IEEE, pp 230–233
- Biao Z, Keding Z, Fengying S (2009) Neural network parameter identification of electro-hydraulic load simulator. *Acta Aeronaut Astronaut Sinica* 2:032
- Zhang B, Dong Y (2014) Parameters identification of passive force control system based on backstepping genetic algorithm. In: 2014 11th world congress on intelligent control and automation (WCICA). IEEE, pp 5846–5851
- Yin X, Lin Y, Li W (2016) Predictive pitch control of an electro-hydraulic digital pitch system for wind turbines based on the extreme learning machine. *Trans Inst Meas Control* 38.11:1392–1400
- Jing WX et al (2019) Research on low-speed performance of continuous rotary electro-hydraulic servo motor based on robust control with Adaboost prediction. *J Eng* 2019.13:60–67
- Guan C, Pan S (2008) Nonlinear adaptive robust control of single-rod electrohydraulic actuator with unknown nonlinear parameters. *IEEE Trans Control Syst Technol* 16.3:434–445
- Dashti ZAS et al (2014) Neural—Adaptive control based on backstepping and feedback linearization for electro hydraulic servo system. In: 2014 Iranian conference on intelligent systems (ICIS). IEEE, pp 1–6
- Guo Q et al (2016) Parametric adaptive estimation and backstepping control of electrohydraulic actuator with decayed memory filter. *ISA Trans* 62:202–214

27. Mitchell T (1997) Machine learning. McGraw Hill, New York
28. Hwang C-L (1999) Neural-network-based variable structure control of electrohydraulic servosystems subject to huge uncertainties without persistent excitation. *IEEE/ASME Trans Mechatron* 4.1:50–59
29. Yuan Z-H, Wu J-D, Teng J-H (2003) Hybrid control of load simulator for unmanned aerial vehicle based on wavelet networks. In: 2003 international conference on machine learning and cybernetics, vol 2. IEEE, pp 715–719
30. Huang A et al (1997) Identification and adaptive control for electro-hydraulic servo system using neural networks. In: 1997 IEEE international conference on intelligent processing systems, 1997. ICIPS'97, vol 1. IEEE, pp 688–692
31. Fogel DB (2006) Evolutionary computation: toward a new philosophy of machine intelligence, vol 1. Wiley, New York
32. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the 1995 IEEE international conference on neural networks. IEEE, pp 1942–1948
33. Pham DT, Castellani M (2009) The Bees Algorithm: modelling foraging behaviour to solve continuous optimization problems. *Proc Inst Mech Eng Part C J Mech Eng Sci* 223.12:2919–2938
34. Pham DT, Castellani M (2014) Benchmarking and comparison of natureinspired population-based continuous optimisation algorithms. *Soft Comput* 18.5:871–903
35. Pham DT, Castellani M (2015) A comparative study of the Bees Algorithm as a tool for function optimisation. *Cogent Eng* 2.1:1091540
36. Stinchcombe K, Hornik M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Netw* 2.5:359–366
37. Riedmiller M, Braun H (1993) A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In: IEEE international conference on neural networks
38. Haykin S (1999) Neural networks: a comprehensive foundation. Prentice Hall, Upper Saddle River
39. Hornik K (1991) Approximation capabilities of multilayer feed-forward networks. *Neural Netw* 4.2:251–257
40. Khashei M, Bijari M (2011) Which methodology is better for combining linear and nonlinear models for time series forecasting? *J Ind Syst Eng* 4.4:265–285
41. Terui N, Van Dijk HK (2002) Combined forecasts from linear and nonlinear time series models. *Int J Forecast* 18.3:421–438

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.