

## Article

# Sensorless Estimation Based on Neural Networks Trained with the Dynamic Response Points

Omar Rodríguez-Abreo <sup>1,2,\*</sup>, Francisco Antonio Castillo Velásquez <sup>2,3</sup>, Jonny Paul Zavala de Paz <sup>2,3</sup>,  
José Luis Martínez Godoy <sup>1,2</sup> and Crescencio García Guendulain <sup>2,4</sup>

<sup>1</sup> Industrial Technologies Division, Universidad Politecnica de Queretaro, El Marques 76240, Queretaro, Mexico; jose.martinez@upq.mx

<sup>2</sup> Red de Investigación OAC Optimización, Automatización y Control, El Marques 76240, Queretaro, Mexico; francisco.castillo@upq.mx (F.A.C.V.); jonny.zavala@upq.mx (J.P.Z.d.P.); crescencio.garcia@tec.mx (C.G.G.)

<sup>3</sup> Information Technology Division, Universidad Politecnica de Queretaro, El Marques 76240, Queretaro, Mexico

<sup>4</sup> Tecnológico de Monterrey, School of Engineering and Sciences, Altamira 89600, Tamaulipas, Mexico

\* Correspondence: omar.rodriguez@upq.edu.mx

**Abstract:** In the present work, a neuronal dynamic response prediction system is shown to estimate the response of multiple systems remotely without sensors. For this, a set of Neural Networks and the response to the step of a stable system is used. Six basic characteristics of the dynamic response were extracted and used to calculate a Transfer Function equivalent to the dynamic model. A database with 1,500,000 data points was created to train the network system with the basic characteristics of the dynamic response and the Transfer Function that causes it. The contribution of this work lies in the use of Neural Network systems to estimate the behavior of any stable system, which has multiple advantages compared to typical linear regression techniques since, although the training process is offline, the estimation can perform in real time. The results show an average 2% MSE error for the set of networks. In addition, the system was tested with physical systems to observe the performance with practical examples, achieving a precise estimation of the output with an error of less than 1% for simulated systems and high performance in real signals with the typical noise associated due to the acquisition system.

**Keywords:** Neural Network; parameter estimation; dynamic response; sensorless estimation



**Citation:** Rodríguez-Abreo, O.; Castillo Velásquez, F.A.; Zavala de Paz, J.P.; Martínez Godoy, J.L.; García Guendulain, C. Sensorless Estimation Based on Neural Networks Trained with the Dynamic Response Points. *Sensors* **2021**, *21*, 6719.

<https://doi.org/10.3390/s21206719>

Academic Editor: Stefanos Kollias

Received: 29 July 2021

Accepted: 31 August 2021

Published: 10 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The estimation of parameters is a widely studied problem; there are multiple works such as [1–6] in which the estimation of the parameters of the Photovoltaic Models is developed as the main object of study. It is due to the high interest in having a function that describes the dynamic behavior of the systems since its performance can be estimated as in [7], where the authors propose a sensor-less prediction system. Another option to use parameter estimation is precise control design since it is generally done theoretically based on system dynamics, as in [8–12] where the authors base the control design on the analysis of the dynamic model and the parameters precision gives the controller precision.

A recurrent option in the control and analysis of the dynamic system is the use of the Transfer Function. This type of function allows a general analysis of the system. The main advantage is that different systems can be represented in the same way, especially when studying first-order and second-order systems. Several investigations use systems of first-order or second-order [13–15]. The investigation [16] uses the parameter estimation of the Transfer Function instead of working with the dynamic model of the system to solve the inverse heat conduction problem. Another example of the use of Transfer Function is the work [17], where the authors estimate the dynamic behavior of the electrohydraulic servo drive through the Transfer Function.

There are various techniques for parameter estimation. One option is a heuristic method like the one presented in [18], where a parametric estimation system with the vector-type recursive least squares (VRLS) is developed. The investigations [19–22] use a heuristic method to estimate and control the systems based on the dynamic response. Another example for the analysis with a heuristic method is the study [23] where the authors do the Transfer Function parameter estimation with the Vector Fitting technique.

Another option for parametric estimation is the so-called population-based meta-heuristic algorithms, such as the research presented in [24–26]. The authors use the Cuckoo Search Algorithm to estimate different types of physical systems. These types of algorithms have numerous advantages, among them their easy implementation [27], which allows them to be implemented in systems of multiple natures [28,29]. However, its nature is inherently iterative; therefore, its processing time is usually longer, and its implementation online is more complicated.

Some authors have developed adaptive techniques for parametric estimation. For instance, in the research shown in [30], the authors develop a method for parameters that vary in time. Another widely studied option is the use of Neural Networks. Several works exploit the use of networks in different fields [31,32]. Particularly in the field of parameter estimation, Neural Networks have shown excellent results, for example, in motors [33]. Another example is the work exhibited in [34], which showed the advances in parametric estimation through a Neural Network for lithium batteries.

Some authors have developed adaptive techniques for parametric estimation. For instance, in the study [30], the authors develop a method for parameters that vary in time. Another widely studied option is the use of Neural Networks. Several works exploit the use of networks in different fields [31,32]. Particularly in parameter estimation, Neural Networks have shown excellent results, for example, in motors [33]. Another example is the work exhibited in [34], which showed the advances in parametric estimation through a Neural Network for lithium batteries. Despite the different estimation and optimization methods, it is important to note that no one algorithm is better in all cases since, according to the No Free Lunch theorem, the algorithms cannot outperform the others if they are averaged over all possible problems. Much of the performance of the algorithms lies then in the correct parameters selection that governs their behavior and the problem specificity [35].

Unlike previous works, the proposed research is not based on estimating a specific system. Instead, it aims to develop a standard parametric estimation system for any open-loop stable systems using the second-order standard Transfer Function. Therefore, this method is applied to systems that enter a steady state with a finite value different than zero. The main contribution of this work is summarized in the estimation of the equivalent Transfer Function of second-order through Neural Networks for systems of any type of system based on the key characteristics of the dynamic response. This Transfer Function will allow estimating the response of the system and designing controllers effectively sensorless even if the system cannot be measured continuously or its parameters are difficult to calculate.

Additionally, this work presents the analysis of the components of the dynamic response as input of a Neural Network. The time rise and final value in steady state are the main factors of influence in estimating parameters based on the response to the step of a system. On the other hand, time delay and time peak are the variables that have a minor influence. Multiple architectures were analyzed to determine the best performance of the network, demonstrating that it can estimate the equivalent Transfer Function with only the key points of the dynamic response instead of all points. In addition to the vector of tests, the network was tested with two typical engineering systems: an electrical circuit and a DC motor to evaluate this network behavior with a step dynamic response in physical systems.

The rest of the work is described in the following way. Section 2 briefly describes the parts of the Transfer Function and their application in the description of physical systems. Section 3 shows the description of the Neural Network, its use, and the analysis of its performance based on the dynamic response. Section 4 presents the results of the

trained network when implemented with signals from physical systems obtained through simulation. In Section 5 the analysis of the performance of the proposed network in real systems is analyzed. Finally, in Section 6, the conclusions are shown.

## 2. Second-Order Transfer Function Characteristics and Its Dynamic Response

The Transfer Function is highly used in control systems; it allows us to relate the input with the output of the system as expressed in Equation (1).

$$G(s) = \frac{O(s)}{I(s)} \quad (1)$$

where  $G(s)$  is the Transfer Function,  $O(s)$  is the output of the system, and  $I(s)$  is the input. All variables are in the Laplace domain and are considered null initial values.

The function can be expressed in terms of the damping ratio ( $\zeta$ ), the natural frequency ( $\omega_n$ ), and a DC gain ( $k$ ), as seen in Equation (2).

$$G(s) = \frac{O(s)}{I(s)} = k \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2)$$

The above equation represents a large number of physical linear and single-input single-output systems. Although these limitations in the representation of systems through a Transfer Function, motors, pistons, damping systems, among many other types of systems, can be represented with this expression. The behavior of the system output is calculated in a standard way if the system is stable in an open-loop for its analysis using a step type input of magnitude  $a$ . Therefore, the output of the system can be obtained by Equation (3).

$$O(s) = k \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \frac{a}{s} \quad (3)$$

A step input generates a typical, expected dynamic response, as shown in Figure 1, where the key characteristics of the dynamic response are observed.

The function returns the characteristics in a structure containing the fields:

- Final value ( $V_f$ )—Value taken in the steady state;
- Rise Time ( $t_r$ )—Time it takes for the output to reach from 10% to 90% to the end value in steady-state response;
- Overshoot ( $M_p$ )—maximum peak value in percentage;
- Peak Time ( $t_p$ )—Time at which the overshoot happens;
- Time delay ( $t_d$ )—Time for reaching 50% of the final value the first time.
- Settling Time ( $t_s$ )—Time it takes for the error between the output and final value to fall within 2% of final value.

Each of these variables constitutes a part of the dynamic response for any system. However, depending on the type of response, some of these characteristics may not appear within the dynamic response. For example, overshoot only appears when the system is underdamped. In any other case, its value will always be zero. These characteristics are extracted and stored in a database as described in the next section. Finally, if the Final Theorem Value is considered, the value of  $k$  is determined by Equation (4).

$$k = \frac{V_f}{a} \quad (4)$$

Equation (4) is valid only when time tends to infinity and the system is in the steady state. The value of  $a$  and  $V_f$  are typically known. Therefore, the dynamic response of the system depends on  $\zeta$  and  $\omega_n$ .

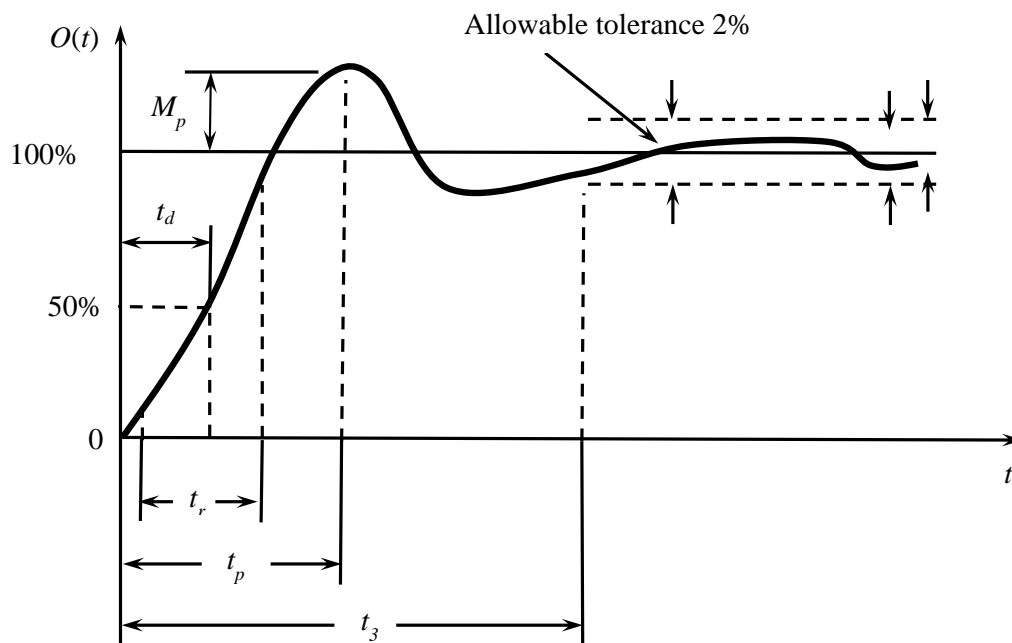


Figure 1. Key characteristics of the dynamic response to step input.

### 3. Neural Network as Parameter Estimator

Artificial Neural Networks (ANN) are widely used in any type of engineering problem, including control problems. The ANNs are mainly used for the resolution of non-linear systems. However, it can also be used in linear systems. However, they have not been tested as parametric estimators for Transfer Functions. This section explains the development of the Neural Network used for estimation tests.

The general process to use the network as an estimator is observed in Figure 2 and consists of training the Neural Network, subsequently using the step response to extract the key points of the dynamic response. With these points as input, the network estimates the parameters of the transfer function.

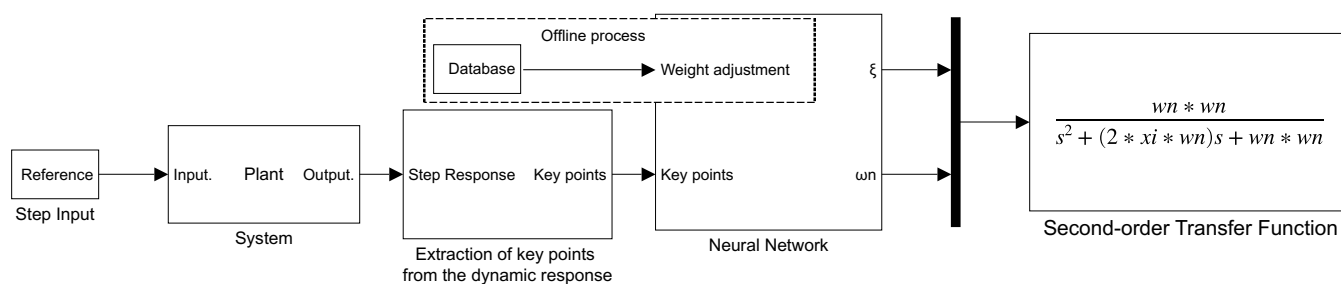


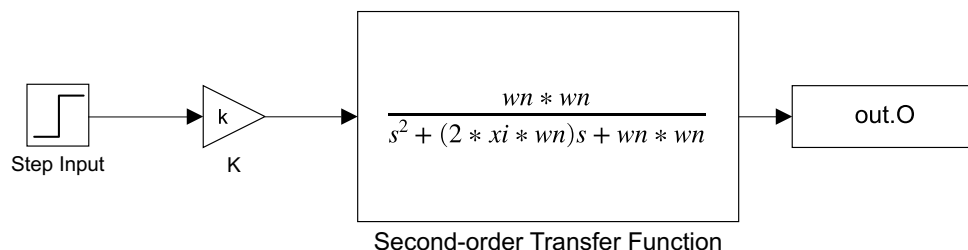
Figure 2. General process to implement the BP-ANN as parameter estimator of a Transfer Function.

The first step is to select the type of architecture network, although, in the literature, there are multiple network architectures. The backpropagation (BP) Neural Network was chosen because it is the most widely used Neural Network [36]. The BP-ANN has been used in multiple types of systems, and their advantages and disadvantages are well known.

This type of network requires a training stage, and a broad database allows the network to know how the system responds to known inputs. For the trained and performance evaluation, an artificial database was created. It is observed in Equations (3) and (4) that two parameters must be adjusted to obtain different dynamic responses, the damping factor, and the natural frequency. The value of the amplitude and the gain  $k$  are directly

related to the final output value of the system. Therefore, it is unnecessary to consider them as long as the value of the amplitude of the input step used is available.

A total of 1,500,000 combinations were performed in which each parameter was varied within the range shown in Table 1 to cover a vast number of values and different dynamic response types. The Simulink model shown in Figure 3 was used to simulate the 1,500,000 combinations.



**Figure 3.** Simulation used for obtained the variables of the step response.

**Table 1.** Ranges for variables used for generating the database.

| $\omega_n$                    | $\zeta$                       |
|-------------------------------|-------------------------------|
| $[1 \times 10^{-6} \ 10,000]$ | $[1 \times 10^{-6} \ 10,000]$ |

The parameter  $\omega_n$  was varied uniformly to obtain 1000 values within the range. The case of  $\zeta$  was treated with 1500 values since it has special cases. The underdamped case has  $\zeta$  between 0 and 1, and 500 values uniformly distributed in that range were used. For the critically damped case, the value of  $\zeta$  must be 1. The remaining values were uniformly distributed for the overdamped and high-overdamped cases.

The simulations were run, and the key points of the dynamic response and the values of  $\zeta$  and  $\omega_n$  were saved.

Once the database is completed, its data are filtered and all storage data without a finite response or with an establishing time longer than 100 s (22,234 data vectors) are removed. Subsequently, the architecture of the ANN was defined. The architecture depends mainly on four main parameters: the number of inputs and outputs, the number of hidden layers, the number of neurons, and the activation functions. Therefore, the next step is to select the inputs and outputs. The built database has several variables; the use in the ANN is described in Table 2.

**Table 2.** Variables of the database and its use in the ANN.

| $t_r$  | $M_p$  | $t_p$  | $t_d$  | $t_s$  | $\zeta$ | $\omega_n$ |
|--------|--------|--------|--------|--------|---------|------------|
| Input1 | Input2 | Input3 | Input4 | Input5 | Output1 | Output2    |

Based on the tests and the equations of the dynamic response, it was decided to use independent networks for each of the outputs, that is, one BP-ANN for  $\zeta$  estimation and one BP-ANN for  $\omega_n$ . Additionally, a pair of networks was used for critical damped, overdamped, and heavily overdamped systems and a different pair of networks for underdamped systems.

The division was carried out in this way because the overshoot is an important variable in underdamped systems. However, it will always be zero in other types of systems. Thus it cannot be considered a valid input for the critical damped, overdamped, and heavily overdamped network. Therefore, in this network, entry overshoot is removed from the network. There are several options for choosing the appropriate values for the remaining parameters of the architecture. However, it is necessary to carry out tests to analyze the performance of each proposed architecture.

Offline training was carried out using the filtered database. For this stage, the data were randomly divided to use 60% for training (886,670), 25% for validation (369,441), and 15% to carry out the performance tests (221,655). The training stage is crucial to determine the performance of the network. Training a Neural Network consists of adjusting each of the weights of the inputs of all neurons that are part of the Neural Network. Therefore, the responses of the output layer fit as closely as possible to the data we know. Typically, it is done by constructive methods, which allow the selection of the number of hidden neurons within the training process by evaluating the convenience of adding or not a new parameter to the network, depending on whether it increases the performance of the network.

The initial value of neurons was taken using the geometric pyramid rule. Later, the near values were evaluated according to the performance of the networks. The performance of the ANNs for this work was measured using the mean square error (MSE). The stop training conditions are 10,000 epochs, 30 validation checks, or gradient minor to  $1 \times 10^{-7}$ .

Several architectures were tested during the hidden layers, the numbers of neurons per layer, and the activation functions. Transfer functions tested were Hyperbolic Tangent Sigmoid (Tansig), the Log-sigmoid (Logsig), and the Linear transfer function (Purelin). The most relevant tests for the architecture selection are shown in Table 3 for the system without overshoot. The overall performance of the pair of the networks is the average of MSE of both networks.

**Table 3.** Relevant tests for determining the architecture of ANNs for critical damped, overdamped, and heavily overdamped systems.

| Network | Neurons in HL | Activation Functions  | MSE $\omega_n$ | MSE $\zeta$ | MSE Average |
|---------|---------------|-----------------------|----------------|-------------|-------------|
| N1a     | 3             | Tansig-Purelin        | 4.529%         | 1.990%      | 3.260%      |
| N2a     | 5             | Tansig-Purelin        | 4.138%         | 1.968%      | 3.280%      |
| N3a     | 8             | Tansig-Purelin        | 4.421%         | 1.987%      | 3.204%      |
| N4a     | 16            | Tansig-Purelin        | 4.506%         | 1.942%      | 3.224%      |
| N6a     | 16            | Tansig-Tansig         | 4.465%         | 1.990%      | 3.227%      |
| N7a     | 16            | Tansig-Logsig         | 6.176%         | 15.871%     | 11.023%     |
| N8a     | 5-3           | Tansig-Tansig-Purelin | 4.612%         | 1.987%      | 3.229%      |
| N9a     | 12-8          | Tansig-Tansig-Purelin | 3.672%         | 1.987%      | 3.053%      |
| N10a    | 8-12          | Tansig-Tansig-Purelin | 3.832%         | 1.994%      | 5.494%      |
| N11a    | 10-16         | Tansig-Tansig-Purelin | 3.374%         | 1.989%      | 2.681%      |
| N12a    | 10-16         | Logsig-Logsig-Purelin | 4.227%         | 1.994%      | 2.810%      |
| N13a    | 10-16         | Tansig-Tansig-Logsig  | 6.667%         | 15.79%      | 11.22%      |
| N14a    | 14-20         | Tansig-Tansig-Purelin | 3.923%         | 1.980%      | 2.952%      |

As seen in the previous Table, the N11a network has the architecture with the best performance in average. Therefore, the network used for systems without overshoot consists of two hidden layers with 10 and 16 neurons, respectively, with Tansig-Tansig transfer functions for the hidden layers and Purelin for the output layer. Although architectures such as N12a offer similar performance with the same number of layers and neurons, the combination with these transfer functions offers the most considerable reduction in MSE, obtaining a 2.81% error in this architecture. On the other hand, the same tests were repeated for systems with overshoot. The results are summarized in Table 4.

**Table 4.** Relevant tests for determining the architecture of ANN in underdamped systems.

| Network | Neurons in HL | Activation Functions    | MSE $\omega_n$         | MSE $\zeta$             | MSE Average            |
|---------|---------------|-------------------------|------------------------|-------------------------|------------------------|
| N1b     | 3             | Tangsig-Purelin         | 1.001%                 | $4.1 \times 10^{-3}\%$  | 0.502%                 |
| N2b     | 5             | Tangsig-Purelin         | 0.105%                 | $2.4 \times 10^{-3}\%$  | 0.054%                 |
| N3b     | 8             | Tangsig-Purelin         | $2 \times 10^{-3}\%$   | $2.1 \times 10^{-3}\%$  | 0.002%                 |
| N4b     | 16            | Tangsig-Purelin         | 0.070%                 | $1.8 \times 10^{-3}\%$  | 0.036%                 |
| N6b     | 16            | Tangsig-Tangsig         | 0.304%                 | $1.9 \times 10^{-3}\%$  | 0.153%                 |
| N7b     | 16            | Tangsig-Logsig          | 6.250%                 | 4.180%                  | 5.125%                 |
| N8b     | 5–3           | Tangsig-Tangsig-Purelin | $9 \times 10^{-3}\%$   | $1.9 \times 10^{-3}\%$  | 0.005%                 |
| N9b     | 12–8          | Tangsig-Tangsig-Purelin | $3 \times 10^{-3}\%$   | $1.9 \times 10^{-3}\%$  | 0.002%                 |
| N10b    | 8–12          | Tangsig-Tangsig-Purelin | $1.7 \times 10^{-3}\%$ | $3.25 \times 10^{-4}\%$ | $8.4 \times 10^{-4}\%$ |
| N11b    | 10–16         | Tangsig-Tangsig-Purelin | $2 \times 10^{-3}\%$   | $1.9 \times 10^{-3}\%$  | 0.002%                 |
| N12b    | 10–16         | Logsig-Logsig-Purelin   | $1.4 \times 10^{-3}\%$ | $8.8 \times 10^{-4}\%$  | 0.001%                 |
| N13b    | 10–16         | Tangsig-Tangsig-Logsig  | 4.209%                 | 4.915%                  | 4.562%                 |
| N14b    | 14–20         | Tangsig-Tangsig-Purelin | 4.182%                 | 4.150%                  | 4.166%                 |

The network N10b has the best performance considering the previous table. As in the previous case, the architecture with the lowest MSE was selected, in this case,  $8.4 \times 10^{-4}\%$ ; for this they used two hidden capable with 8 and 12 neurons and Tansig transfer functions-Tansig between the hidden layers and Purelin for the output layer. A lower MSE can also be observed in general in networks used for systems with overshoot. Therefore, it was used for all subsequent tests when the system was underdamped. The previous results show an adequate performance of the network in estimating the Transfer Function parameters with reduced MSE values for  $\omega_n$  and  $\zeta$  parameters. However, it is desired to validate the inputs and their influence on the ANN to discard unnecessary or redundant entries since not necessarily all the listed characteristics impact the calculation of the parameters of the Transfer Function. Therefore, the network is retrained by deleting one input at a time to discover the effect of the inputs in the ANNs performance. The results of these tests in both sets of ANNs are shown in Table 5.

**Table 5.** Test for validation of the inputs of the ANN.

| Input Removed | MSE Average for N11a | MSE Change in N11a | MSE Average for N10b    | MSE Change in N10b       |
|---------------|----------------------|--------------------|-------------------------|--------------------------|
| $t_s$         | 3.071%               | +0.39%             | $1.6 \times 10^{-3}\%$  | $+7.6 \times 10^{-4}\%$  |
| $M_p$         | -                    | -                  | $2.39 \times 10^{-2}\%$ | $+2.3 \times 10^{-2}\%$  |
| $t_p$         | 2.994%               | +0.313%            | $9.59 \times 10^{-4}\%$ | $+1.19 \times 10^{-4}\%$ |
| $t_r$         | 3.563%               | +0.090%            | $3.5 \times 10^{-3}\%$  | $+2.7 \times 10^{-3}\%$  |
| $t_d$         | 3.081%               | +0.400%            | $1.4 \times 10^{-3}\%$  | $+5.83 \times 10^{-4}\%$ |
| $V_f$         | 3.393%               | +0.019%            | $3.1 \times 10^{-3}\%$  | $+2.2 \times 10^{-3}\%$  |

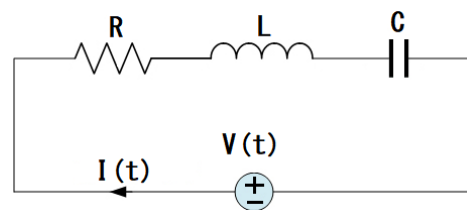
According to the results, the entry that contributes the most to MSE reduction is  $t_s$  for overdamped systems and  $M_p$  in underdamped systems. On the other hand, the most negligible impact on the network is  $V_f$  with only 0.19% reduction to an error in overdamped systems and the  $t_d$  in underdamped systems. However, its use is maintained since, in general terms, it benefits the ANNs and provides a positive effect even if it is small.

In concordance with the tests, the whole system used was the ANN N11a for critical damped, overdamped, and heavily overdamped systems with five inputs and two independent outputs and the ANN N10b for underdamped systems with six inputs and two separate outputs.

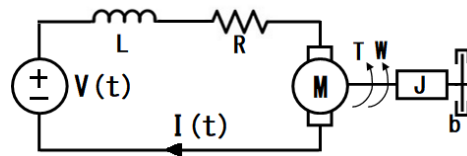
#### 4. ANNs Evaluating Performance with Simulated Systems

The previous section only analyzes the error in the magnitude of the estimation of the parameters. However, it is necessary to examine how these errors in the Transfer Function parameters estimation affect the dynamic response from the point of view of the signals. Therefore, this section has explored the differences between signals produced by the set of ANNs and signals of simulated systems.

Two typical and well-known engineering systems were selected to test the ANNs signals. The Transfer Functions of an overdamped and underdamped system were taken into account. The overdamped system is the second-order electrical circuit shown in Figure 4a. The underdamped system is DC motor which is a widely used second-order electromechanical system, and its diagram is depicted in Figure 4b. The selected examples can be represented with their respective dynamic model. However, it is common to see them represented with their corresponding Transfer Function.



(a) Overdamped system represented by an RLC circuit.



(b) Underdamped system represented by a DC motor.

**Figure 4.** Systems used for the ANNs tests with real systems.

Figure 4a represents a second-order system and its Transfer Function, with the output as the voltage in the capacitor and the input as the voltage, is described by Equation (5).

$$V_c(s) = \frac{1}{LCs^2 + CRs + 1} \frac{a}{s} \quad (5)$$

where  $V_c$  is the voltage in the capacitor,  $L$  is the value of the inductor, and  $R$  is the value of the resistor. For the electromechanical system shown in Figure 4b, the output was considered as the angular velocity and the input is the voltage applied to the motor, and the Transfer Function is expressed in Equation (6).

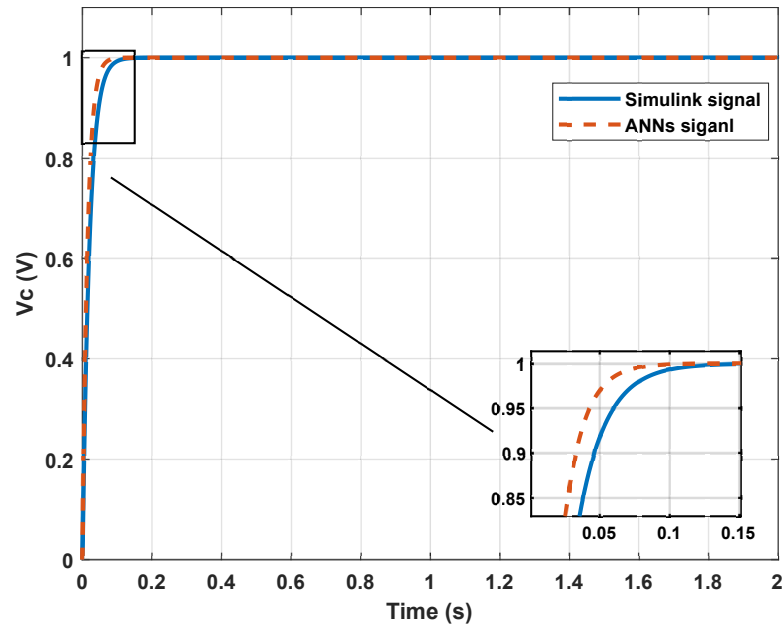
$$\omega(s) = \frac{K_m}{LJs^2 + (RJ + LB)s + (RB + K_m K_a)} \frac{a}{s} \quad (6)$$

where  $V(t)$  is the voltage,  $R$  is the armature resistance,  $I(t)$  is the current in the mesh,  $K_e$  and  $K_m$  are the constant electrical and mechanical, respectively,  $L$  is the armature inductance,  $J$  is the momentum of inertia, and  $B$  is the friction coefficient. In practice, the constants  $K_e$  and  $K_m$  are similar and can be considered an equal constant of magnitude  $K$ . Therefore, Equation (6) can be rewritten as Equation (7):

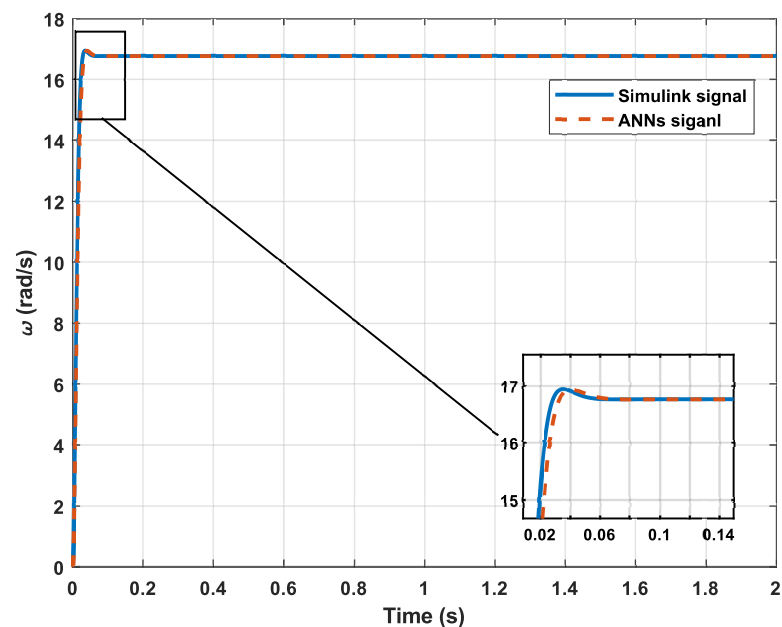
$$\omega(s) = \frac{K}{LJs^2 + (RJ + LB)s + (RB + K^2)} \frac{a}{s} \quad (7)$$



The corresponding simulations were carried out to obtain the step response of each of the plants described above. The two systems were simulated by Simulink using the time and the values expressed in Table 6 and Equations (5) and (7). The outputs of both systems were saved, and the key points of the dynamic responses were extracted from these signals to use as ANNs sets inputs. The comparison between the signal calculated with the coefficients estimated by the ANNs and the simulated signals is shown in Figure 5.



(a) Overdamped system.



(b) Underdamped system.

Figure 5. Systems used for testing the ANN with real systems.

**Table 6.** Values used for simulate the plants.

| Plant       | Parameters Values   | Time Simulation | Step Amplitude |
|-------------|---|-----------------|----------------|
| RLC circuit | $R = 20 \Omega$ ; $L = 0.47 \text{ mH}$<br>$C = 100 \mu\text{F}$  | 2 s             | 1 V            |
| DC motor    | $R = 3.13 \Omega$ ; $L = 13.07 \text{ mH}$<br>$B = 169 \frac{\text{g}\cdot\text{m}^2}{\text{s}^2}$ ; $J = 9.0 \mu\text{Nm}$<br>$K = 0.0487$ | 2 s             | 1 V            |

The tests showed an MSE between signals of 0.45% for the overdamped system and 1.44% for the underdamped system. These results suggested that the parametric estimation can be given with a precision of less than 1.5% despite being of a different nature. The DC motor has a particularity of the  $V_f$  that is greater than the magnitude of input  $a$ , but the use of the Final Value Theorem is helpful to process these types of systems.

Further, the results indicated that the errors in the estimation of the parameters in the Transfer Function are not significant in the response of the proposed system. It is also observed that the set of networks designed to estimate the parameters of systems with overshoots has a slightly better performance than the set of networks used for systems without overshoot. Although, this was an expected effect since in the network design phase this behavior was always shown.

## 5. Experimental Results

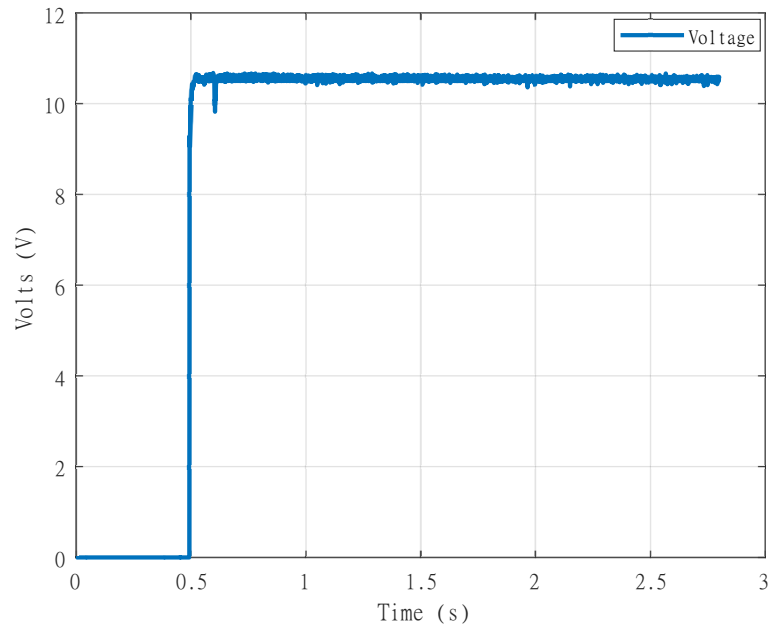
The system has been tested with simulated signals with satisfactory results. However, to verify the efficiency of the proposed method, the ANN performance is analyzed against a real signal. The DC motor shown in Figure 4b was taken as a reference to verify the performance of the proposed method with real systems. It was considered a Mavilor brand motor model CML 050 with the nominal parameters shown in Table 6.

The acquisition brings with it uncertainty problems due to the measurement hardware. In this case, the acquisition was made with the ADC (Analogue to Digital Converter) of the PIC18F4550 to measure the voltage input and a quadrature encoder to measure the angular velocity of the rotor.

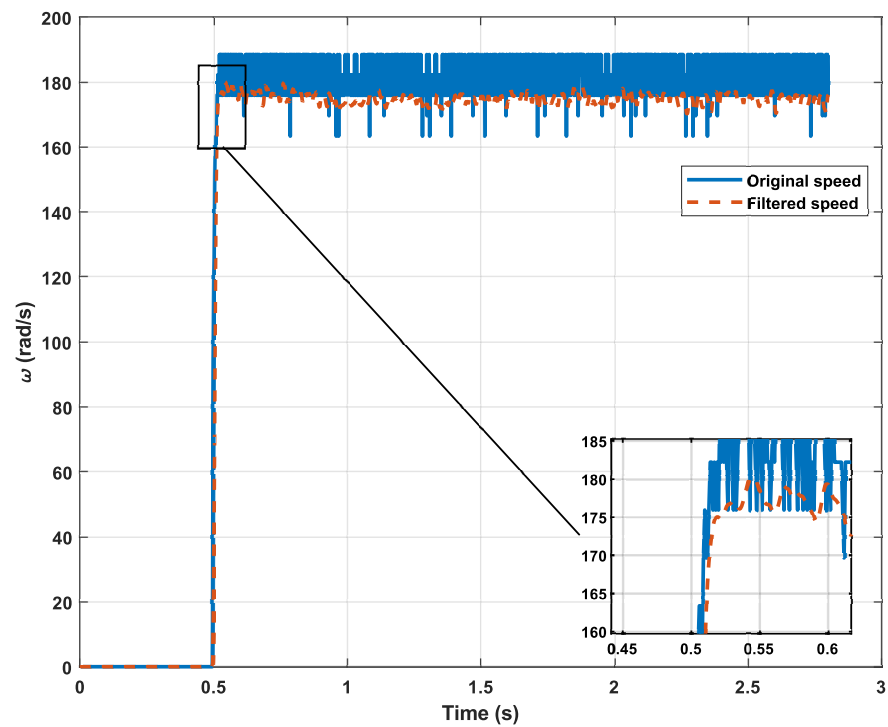
The system is excited with a step input with a magnitude of 10.5 V to obtain the dynamic response and measured with the ADC, getting the signal shown in Figure 6a. For that, its speed data are sampled at a frequency rate of 1 kHz. Due to the ADC operating range, a signal conditioning stage is required since the motor operates with up to 24 V.

Each part of the conditioning and acquisition stage adds noise into the signal measured. The added noise is multifactorial and can be due to a low resolution in the sampling, the precision of the measuring instrument, or personal failures in using the instruments. Although noise can be reduced with more precise and accurate hardware, the real signals will always contain added noise that can also be reduced via software. Therefore, the work with real signals must consider filtering and averaging stages. In this sense, the original speed signal passed through a Chebicheb I filter. The result of the acquired signals is displayed in Figure 6b. On the other hand, to calculate the final or maximum values, the mean values were taken from three contiguous samples of the signal.

The key characteristics of the dynamic response are extracted from the filtered speed signal (Table 7) and are used as inputs in the set of networks N11b. The result of the estimation is shown in Figure 7, where it is observed that the network provides an adequate response regardless of the step input magnitude with which the test is performed.



(a) Real acquired voltage signal applied to a DC motor and sampled by an ADC.



(b) Real speed signal acquired using a quadrature encoder vs. signal with an Chebyshev filter.

Figure 6. Systems used for test the ANN set with real systems.

Table 7. Variables of the dynamic response in real signal.

| $t_r$  | $M_p$   | $t_p$    | $t_d$   | $t_s$   | $V_f$     |
|--------|---------|----------|---------|---------|-----------|
| 0.05 s | 0.0241% | 0.0430 s | 0.010 s | 0.004 s | 178 rad/s |

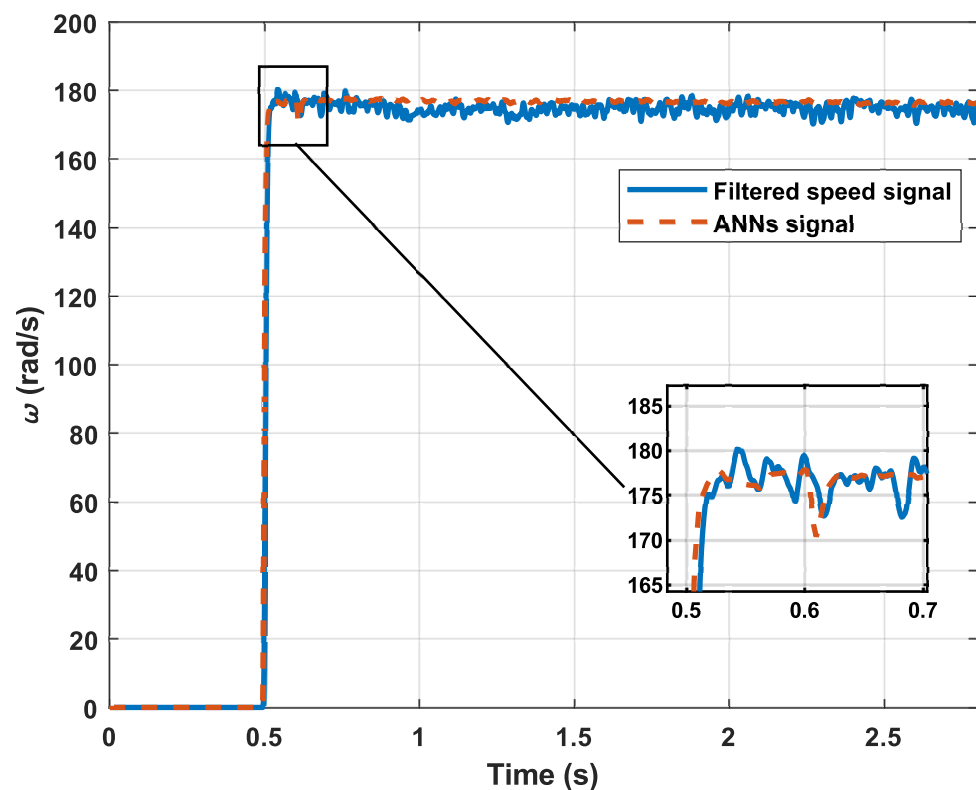


Figure 7. Simulation used for obtaining the variables of the step response.

The error increases if it is compared with the data obtained in the simulation stages. This effect is also normal and was expected. However, the increase is due to the noise in the signals. Therefore, the system can be considered valid and capable of adapting to real work signals even with high noise. Nevertheless, it must be considered that the estimation precision depends not only on the network set precision but also on the accuracy of the measurement and adequate filtering of the signal. Table 8 compares the parameters obtained, the MSE between the original signal, and the signal estimated by the network developed to summarize the numerical results.

Table 8. Summary of the results.

|                      | RLC Simulated     | DC Motor Simulated      | DC Motor Real          |
|----------------------|-------------------|-------------------------|------------------------|
| $\zeta$ estimated    | 46.141            | 0.8238 s                | 0.8225                 |
| $\omega_n$ estimated | $5.9 \times 10^3$ | 137.06                  | 160.1                  |
| General MSE in ANNs  | 2.681%            | $8.4 \times 10^{-4}\%$  | $8.4 \times 10^{-4}\%$ |
| MSE between signals  | 0.014%            | $1.44 \times 10^{-4}\%$ | 0.1%                   |

The MSEs for the DC motor shown in the previous table are normalized since their magnitude is different in each case. Hence, it goes through a normalization process to make an adequate comparison between the different MSEs.

## 6. Conclusions

In this work, a neural system was developed to estimate the parameters of a Transfer Function from the step response of the open-loop stable systems. Multiple tests were performed with different architectures to evaluate the best combination. Four BPNNs were used: two for the underdamped system and two for the overdamped systems.

The results indicate that the system can estimate coefficients of Transfer Functions in tests with simulated systems and real signals with an average error of 0.007% for the

simulated signals and 0.1% for the real signal. According to Tables 3, 4, and 8, there is a better performance for the set of networks in systems with overshoot because there is more information on the dynamic response than in systems without overshoot where it is just eliminated. Figures 5 and 7, together with Table 8, show that the network reconstructions through the parameters estimated have an MSE of less than 1% for all cases.

The methodology presented in this work has the advantage of minimizing calculation time when the system is used online. Additionally, it is adaptable to any stable system, and the input is known (the same as in traditional parametric estimation). However, it has the disadvantages of the need for a database and training time for each network.

Unlike heuristic and metaheuristic methods, the parametric estimation was performed using only the characteristics of the respective dynamics of the system, avoiding comparing point to point to make a linear regression, allowing for the estimation of the performance of any system remotely even if its mathematical model is not known. BP-ANNs can correctly estimate underdamped and overdamped functions. This system can be adapted to practically any response signal with a larger database adapted to the expected ranges.

**Author Contributions:** Conceptualization, O.R.-A.; methodology, O.R.-A.; software, O.R.-A. and F.A.C.V.; validation, O.R.-A.; formal analysis, O.R.-A., J.P.Z.d.P., and J.L.M.G.; investigation, O.R.-A. and F.A.C.V.; resources, O.R.-A.; data curation, O.R.-A. and J.P.Z.d.P.; writing—original draft preparation, O.R.-A. and F.A.C.V.; writing—review and editing, O.R.-A. and J.L.M.G.; visualization, O.R.-A.; supervision, O.R.-A., J.L.M.G., C.G.G., and J.P.Z.d.P.; project administration, O.R.-A., C.G.G., and F.A.C.V.; funding acquisition, O.R.-A. and C.G.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Tecnológico de Monterrey.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

**Acknowledgments:** Thanks to Tecnológico de Monterrey for contributing with monetary financing.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kang, T.; Yao, J.; Jin, M.; Yang, S.; Duong, T. A Novel Improved Cuckoo Search Algorithm for Parameter Estimation of Photovoltaic (PV) Models. *Energies* **2018**, *11*, 1060. [[CrossRef](#)]
2. Liao, Q.; Zhou, S.; Shi, H.; Shi, W. Parameter Estimation of Nonlinear Systems by Dynamic Cuckoo Search. *Neural Comput.* **2017**, *29*, 1103–1123. [a\\_00946](#). [[CrossRef](#)]
3. Odry, A.; Kecskes, I.; Sarcevic, P.; Vizvari, Z.; Toth, A.; Odry, P. A novel fuzzy-adaptive extended kalman filter for real-time attitude estimation of mobile robots. *Sensors* **2020**, *20*, 803. [[CrossRef](#)] [[PubMed](#)]
4. Odry, A.; Fuller, R.; Rudas, I.; Odry, P. Fuzzy control of self-balancing robots: A control laboratory project. *Comput. Appl. Eng. Educ.* **2020**, *28*, 512–535. [[CrossRef](#)]
5. Odry, A. An open-source test environment for effective development of marg-based algorithms. *Sensors* **2021**, *21*, 1183. [[CrossRef](#)] [[PubMed](#)]
6. Odry, A.; Tadic, V.; Odry, P. A Stochastic Logic-Based Fuzzy Logic Controller: First Experimental Results of a Novel Architecture. *IEEE Access* **2021**, *9*, 29895–29920. [[CrossRef](#)]
7. Gutierrez-Villalobos, J.M.; Rodriguez-Resendiz, J.; Rivas-Araiza, E.A.; Martínez-Hernández, M.A. Sensorless FOC Performance Improved with On-Line Speed and Rotor Resistance Estimator Based on an Artificial Neural Network for an Induction Motor Drive. *Sensors* **2015**, *15*, 15311–15325. [[CrossRef](#)] [[PubMed](#)]
8. Yang, C.; Jiang, Y.; He, W.; Na, J.; Li, Z.; Xu, B. Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. *IEEE Trans. Ind. Electron.* **2018**, *65*, 8112–8123. [[CrossRef](#)]
9. Macias-Bobadilla, G.; Becerra-Ruiz, J.; Estévez-Bén, A.A.; Rodríguez-Reséndiz, J. Fuzzy control-based system feed-back by OBD-II data acquisition for complementary injection of hydrogen into internal combustion engines. *Int. J. Hydrogen Energy* **2020**, *45*, 26604–26612. [[CrossRef](#)]
10. Thenozhi, S.; Concha, A.; Resendiz, J.R. A Contraction Theory-based Tracking Control Design With Friction Identification and Compensation. *IEEE Trans. Ind. Electron.* **2021**. [[CrossRef](#)]

11. Mendoza-Mondragón, F.; Hernández-Guzmán, V.M.; Rodríguez-Reséndiz, J. Robust Speed Control of Permanent Magnet Synchronous Motors Using Two-Degrees-of-Freedom Control. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6099–6108. [[CrossRef](#)]
12. Martínez-Hernández, M.A.; Hernández-Guzmán, V.M.; Carrillo-Serrano, R.V.; Rodríguez-Reséndiz, J. A slightly modified indirect field oriented controller for voltage-fed induction motors with a global asymptotic stability proof. *Eur. J. Control* **2015**, *25*, 60–68. [[CrossRef](#)]
13. Becerra-Ruiz, J.; Gonzalez-Huerta, R.; Gracida, J.; Amaro-Reyes, A.; Macias-Bobadilla, G. Using green-hydrogen and bioethanol fuels in internal combustion engines to reduce emissions. *Int. J. Hydrogen Energy* **2019**, *44*, 12324–12332.
14. Padilla-Garcia, E.; Rodríguez-Angeles, A.; Resendiz, J.; Cruz-Villar, C. Concurrent Optimization for Selection and Control of AC Servomotors on the Powertrain of Industrial Robots. *IEEE Access* **2018**, *6*, 27923–27938. [[CrossRef](#)]
15. Castillo-Zamora, J.; Camarillo-Gomez, K.; Perez-Soto, G.; Rodriguez-Resendiz, J. Comparison of PD, PID and sliding-mode position controllers for v-tail quadcopter stability. *IEEE Access* **2018**, *6*, 38086–38096. [[CrossRef](#)]
16. Fernandes, A.; dos Santos, M.; Guimarães, G. An analytical transfer function method to solve inverse heat conduction problems. *Appl. Math. Model.* **2015**, *39*, 6897–6914. [[CrossRef](#)]
17. Sokolov, V.; Krol, O.; Ivanov, V.; Rong, Y.; Trojanowska, J.; Venus, J.; Liaposhchenko, O.; Zajac, J.; Pavlenko, I.; Edl, M.; et al. Determination of Transfer Functions for Electrohydraulic Servo Drive of Technological Equipment. In *Advances in Design, Simulation and Manufacturing*; Springer International Publishing: Cham, Switzerland, 2019; pp. 364–373.
18. Wei, Z.; Zhao, J.; Ji, D.; Tseng, K. A multi-timescale estimator for battery state of charge and capacity dual estimation based on an online identified model. *Appl. Energy* **2017**, *204*, 1264–1274. [[CrossRef](#)]
19. Xu, L.; Ding, F.; Zhu, Q. Hierarchical Newton and least squares iterative estimation algorithm for dynamic systems by transfer functions based on the impulse responses. *Int. J. Syst. Sci.* **2019**, *50*, 141–151. [[CrossRef](#)]
20. Xu, L.; Chen, L.; Xiong, W. Parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration. *Nonlinear Dyn.* **2015**, *79*, 2155–2163. [[CrossRef](#)]
21. Gómez-Espinosa, A.; Hernández-Guzmán, V.M.; Bandala-Sánchez, M.; Jiménez-Hernández, H.; Rivas-Araiza, E.A.; Rodríguez-Reséndiz, J.; Herrera-Ruiz, G. A New Adaptive Self-Tuning Fourier Coefficients Algorithm for Periodic Torque Ripple Minimization in Permanent Magnet Synchronous Motors (PMSM). *Sensors* **2013**, *13*, 3831–3847. [[CrossRef](#)]
22. Rodriguez, R.; Rodríguez, J.; Augusto Gomez, R. Fast square root calculation for DTC magnetic flux estimator. *IEEE Lat. Am. Trans.* **2014**, *12*, 112–115. [[CrossRef](#)]
23. Arda Ozdemir, A.; Gumussoy, S. Transfer Function Estimation in System Identification Toolbox via Vector Fitting. *IFAC-PapersOnLine* **2017**, *50*, 6232–6237.
24. Rodríguez-Abreo, O.; Hernandez-Paredes, J.M.; Rangel, A.F.; Fuentes-Silva, C.; Velásquez, F.A.C. Parameter Identification of Motors by Cuckoo Search Using Steady-State Relations. *IEEE Access* **2021**, *9*, 72017–72024. [[CrossRef](#)]
25. Wei, J.; Yu, Y. An Effective Hybrid Cuckoo Search Algorithm for Unknown Parameters and Time Delays Estimation of Chaotic Systems. *IEEE Access* **2018**, *6*, 6560–6571. [[CrossRef](#)]
26. Freitas de Andrade, C.; Rocha Veiga, F.; Silveira Macedo, M.V.; Ferreira dos Santos, L.; Costa Rocha, P.A.; Santos Guedes, K. Application Of The Cuco Search In The Adjustment Of Weibull Curves For Wind Energy Using Wind Data Of Petrolina City. *IEEE Lat. Am. Trans.* **2019**, *16*, 2513–2520. [[CrossRef](#)]
27. Feng, Z.K.; Niu, W.J.; Liu, S. Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. *Appl. Soft Comput.* **2021**, *98*, 106734. [[CrossRef](#)]
28. Rojas, M.G.; Olivera, A.C.; Vidal, P.J.; Carballido, J.A. A Memetic Cellular Genetic Algorithm for Cancer Data Microarray Feature Selection. *IEEE Lat. Am. Trans.* **2021**, *18*, 1874–1883. [[CrossRef](#)]
29. Montoya, O.D.; Gil-González, W.J.; López-Lezama, J.M. Vortex Search Algorithm Applied to the Parametric Estimation in PV Cells Considering Manufacturer Datasheet Information. *IEEE Lat. Am. Trans.* **2021**, *19*, 1581–1589. [[CrossRef](#)]
30. Na, J.; Xing, Y.; Costa-Castelló, R. Adaptive Estimation of Time-Varying Parameters With Application to Roto-Magnet Plant. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 731–741. [[CrossRef](#)]
31. Rodriguez-Abreo, O.; Rodriguez-Resendiz, J.; Fuentes-Silva, C.; Hernandez-Alvarado, R.; Falcon, M. Self-Tuning Neural Network PID with Dynamic Response Control. *IEEE Access* **2021**, *9*, 65206–65215. [[CrossRef](#)]
32. Ortiz-Echeverri, C.; Salazar-Colores, S.; Rodríguez-Reséndiz, J.; Gómez-Loenzo, R. A new approach for motor imagery classification based on sorted blind source separation, continuous wavelet transform, and convolutional neural network. *Sensors* **2019**, *19*, 4541. [[CrossRef](#)]
33. Gutierrez-Villalobos, J.; Rodríguez-Resendiz, J.; Rivas-Araiza, E.; Mucino, V. A review of parameter estimators and controllers for induction motors based on artificial neural networks. *Neurocomputing* **2013**, *118*, 87–100. [[CrossRef](#)]
34. Hu, X.; Li, S.; Yang, Y. Advanced Machine Learning Approach for Lithium-Ion Battery State Estimation in Electric Vehicles. *IEEE Trans. Transp. Electr.* **2016**, *2*, 140–149. [[CrossRef](#)]
35. Orosz, T.; Rassölkin, A.; Kallaste, A.; Arsénio, P.; Pánek, D.; Kaska, J.; Karban, P. Robust Design Optimization and Emerging Technologies for Electrical Machines: Challenges and Open Problems. *Appl. Sci.* **2020**, *10*, 6653. [[CrossRef](#)]
36. Ye, Z.; Kim, M.K. Predicting electricity consumption in a building using an optimized back-propagation and Levenberg–Marquardt back-propagation neural network: Case study of a shopping mall in China. *Sustain. Cities Soc.* **2018**, *42*, 176–183. [[CrossRef](#)]