

Article

# Speed Control Based on State Vector Applied for Electrical Drive with Elastic Connection

Mateusz Malarczyk , Mateusz Zychlewicz , Radoslaw Stanislawski  and Marcin Kaminski \* 

Department of Electrical Machines, Drives and Measurements, Faculty of Electrical Engineering, Wrocław University of Science and Technology, 19 Smoluchowskiego St., 50-372 Wrocław, Poland; mateusz.malarczyk@pwr.edu.pl (M.M.); mateusz.zychlewicz@pwr.edu.pl (M.Z.); radoslaw.stanislawski@pwr.edu.pl (R.S.)

\* Correspondence: marcin.kaminski@pwr.edu.pl

**Abstract:** The paper is focused on issues related to the control of electrical drives with oscillations of state variables. The main problem deals with the construction of the mechanical part, which contains elastic elements used as a coupling between the motor machine and the load. In such cases, strict tracking of the reference trajectory is difficult, so damping of the disturbances is necessary. For this purpose, the full state vector of the object is applied as the feedback signal for the speed controller. This method is efficient and relatively easy to implement (including the hardware part). However, the control accuracy is dependent on the quality of the parameters identification and the invariance of the object. Thus, two adaptive structures are proposed for the two-mass system. Moreover, selected coefficients were optimized using metaheuristic algorithms (symbiotic organism search and flower pollination algorithm). After presentation of the preliminaries and mathematical background, tests were conducted, and the numerical simulations are shown. Finally, the experimental verification for the 0.5 kW DC machines was performed. The results confirm the theoretical concept and the initial assumptions: the state controller leads to the precise control of the drive with a long shaft; recalculation of the parameters can improve the work of the drive under changes of time constants; modern design tools are appropriate for this application.

**Keywords:** state controller; gains adaptation; symbiotic organism search; flower pollination algorithm; optimization; electrical drive



**Citation:** Malarczyk, M.; Zychlewicz, M.; Stanislawski, R.; Kaminski, M. Speed Control Based on State Vector Applied for Electrical Drive with Elastic Connection. *Automation* **2022**, *3*, 337–363. <https://doi.org/10.3390/automation3030018>

Academic Editors: Marcel Nicola, Dan Selisteanu and Cosmin Ionete

Received: 31 May 2022

Accepted: 7 July 2022

Published: 13 July 2022

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



**Copyright:** © 2022 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

In industrial applications of electrical drives, the main points of concern are the reliability of the control structure, the dynamics of the system's response and the ability to control the state of the plant. A satisfying level of these properties is difficult to obtain when the mechanical structure of the system is compound [1–3]. When a motor is connected to a load machine through a long, thin gear, the machines may start rotating at varying speeds, often in an oscillatory manner. Highly resonant systems with an elastic shaft and a heavy load machine (such as rolling mills [4], wind turbines [5], and robotic arms [6,7]), often referred to as two-mass systems, can be widely found in many industrial solutions. When the shaft is subjected to torsion, the angular velocity of the motor differs from the angular velocity of the load machine, causing oscillations. As a result, many undesirable effects occur in the control structure: the quality of the industrial process may worsen, the final product may be defective or damaged [8], the torsion may cause damage to the mechanical structure of the drive (shaft rupture), and even the stability could be lost [9]. The most basic method of suppressing oscillations is the use of mechanical dampeners. However, for this approach, the mechanical structure of the drive must be modified.

Besides interfering with the mechanical part of the drive, passive and active control methods can be distinguished. The passive methods used to suppress these oscillations focus on the reduction in the reference signal dynamics through the use of signal filtering [10].

It is, however, very detrimental to the dynamic capabilities of the control structure and complicates the command signal generation. The active methods rely on using advanced control structures to mitigate the influence of the elastic connection on the control quality. The most basic method used to control the speed of such drives is the cascade control with two PI controllers—one of them responsible for controlling the current, while the other controls the speed of the motor. This solution does not use any additional information about the load machine; therefore, damping of the oscillations is not efficient [11]. In the literature, many modifications of the PI control with additional feedback loops from the torsional torque and the angular velocity of the load are presented [12,13]. Very promising results are obtained when a state controller is used [14]. However, all the state space variables (the angular velocity of the motor and the load, and the torsional torque) must be known to apply this method [15]. The mentioned technique is very sensitive to external disturbances (such as measurement noise, friction and additional dynamic load changes), nonlinearities, and inaccurate identification of the plant's parameters [16]. To obtain the information about the state of the plant, appropriate sensors or estimation algorithms should be used. Obviously, the second option is preferable due to the reduction in the overall cost and the increased reliability of the drive. Although mathematical models of the variables can be used, their dependence on parameter uncertainties poses a significant problem [17]. In order to obtain a robust tool for the state variables calculation, state observers [18], the Luenberger observers [19], and the Kalman filters [20] are usually implemented.

The results achieved using the state controller can present efficient damping of the state variables oscillations. Moreover, the hardware implementation of the state controller's topology is simple and feasible. The most problematic issue lies within the structure's robustness against the changes of the object's resonant frequency. To improve the control structure's reaction to the identification imprecision and robustness against external disturbances, structures with adaptive properties seem to be an interesting solution [21,22]. According to an analysis of the structure of the controllers, applications proposed in scientific papers can be generally divided into solutions based on neural networks (NN) [23], fuzzy logic (FL) models [24,25], or classical reconfigurable controllers [26].

Synthesis of control structures with NNs is very challenging because of multiple parameters that must be first adjusted. The main concern is the used learning algorithm. Two approaches can be taken into consideration: offline or online learning. For offline learning, the training data must be first collected and pre-processed before the adaptation of the weights used in the network can be conducted [27]. However, adaptive properties are obtained only if online training is applied, meaning the weights must be changed during the operation of the system [28]. Another parameter to be considered is the network's topology. The simplest solutions present ADALINE NNs [29], and combinations of many linear neurons organized into layers—multiple layer perceptron (MLP) NNs [30]. More sophisticated structures include recurrent connections within the network [31,32]. Another point of concern is the selection of the proper activation function for the neurons. The most often chosen activation functions are sigmoid and radial (Gaussian) activation functions [33]. The last parameter needed to design a neural controller is the selection of the learning algorithm and the value of the learning coefficient. With so many different parameters to adjust, control solutions with neural structures may be very difficult to apply. Moreover, using NNs significantly increases the computational power needed for the structure's real-world application.

Similarly to neural controllers, fuzzy system controllers also need to have multiple parameters adjusted to work correctly [34,35]. The main design problem concerns the proper selection of the rule base. To set the rules correctly, the system designer must have comprehensive knowledge about the controlled process and the fuzzy systems themselves. Additionally, the shape of the membership functions, the method of obtaining conclusions from the system (e.g., Mamdani [36] or Takagi-Sugeno-Kang [37] methods) and the defuzzification method (e.g., the center of gravity method or the mean of maximum method) must be chosen.

The details of the structure and the design process related to neural and fuzzy controllers analyzed above lead to the conclusion that the simple classical solutions and concepts known from the theory of artificial intelligence should be combined. In this paper, a state controller with adaptive parameter vector is presented, and two types of the controller are considered. The first uses only the recalculation of the parameters related to the speed of the drive. The second one presents the full adaptation of the gains. The overall construction of the controller is based on the model reference adaptive control (MRAC) structure.

The idea behind MRAC systems is based on the use of the difference between the output from a reference model and the actual output of the system [38]. The difference signal is used to change the values of the controller's gains with a selected adaptation algorithm. The controller is changed to minimize the difference between the model and the system outputs. The advantage of MRAC systems is that the system designer has a direct way of determining the desired behavior of the system through the reference model selection. Moreover, the calculation of the updates for the adaptation process is rarely described by complex equations; therefore, this method is not as computationally heavy as other adaptive structures [39].

To facilitate the design process of the adaptive state controller, the parameters can be optimized using nature-inspired algorithms. Such methods are based on the minimization of a cost function value. What speaks in their favor is the computational simplicity (the derivative of the cost function does not need to be calculated) and the easy development of the practical implementation. They are also applicable to multiple criteria analysis problems. Various species and natural phenomena have been observed and imitated to create such algorithms: the artificial bee colony (ABC) algorithm [40,41], the cuckoo search (CS) algorithm [42–44], the grey wolf optimizer (GWO) [45,46], or the particle swarm optimization (PSO) [47,48] are some of the commonly used examples of such algorithms. In our research, the focus is shifted toward two other examples—the symbiotic organism search (SOS) [49,50], the goal of which is to search for other specimens that might be capable of entering symbiotic relations with each other, and the flower pollination algorithm (FPA), which is inspired by the process of pollen distribution by flowers in a meadow [51,52].

The paper is organized as follows. Section 2 focuses on the introduction of the mathematical description of the analyzed plant, the derivation of the equations for the state controller, and the explanation of the adaptation using the Widrow-Hoff rule. The next two sections show the preliminary information about the used optimization algorithms. Next, the simulation results with commentary are presented. In the following section, the experimental verification of the obtained simulation results is presented. Finally, the last section summarizes the research results and concludes the paper with final remarks and observations.

## 2. A State Controller Applied for Speed Control of Electrical Drives with Elastic Shaft

A simplified picture of the drive's mechanical part is presented in Figure 1. The control structure analyzed in this article is based on a state controller and a two-mass drive (as presented in Figure 2).

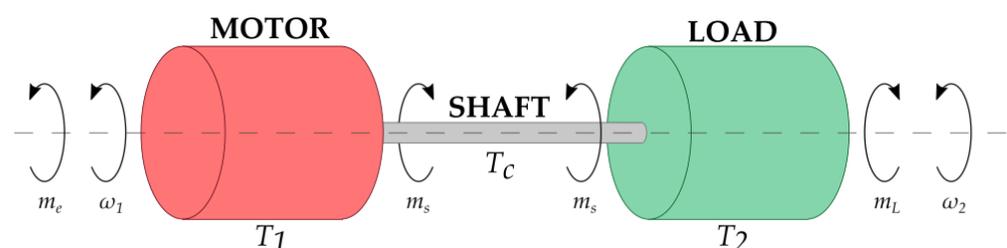


Figure 1. The general view of a two-mass system.

### 2.1. Overall Description of the Control System

The presented model of the mechanical part of the drive can be described using the following system of differential equations [31,53,54]:

$$\frac{d\omega_1}{dt} = \frac{m_e - m_s}{T_1} \quad (1)$$

$$\frac{d\omega_2}{dt} = \frac{m_s - m_L}{T_2} \quad (2)$$

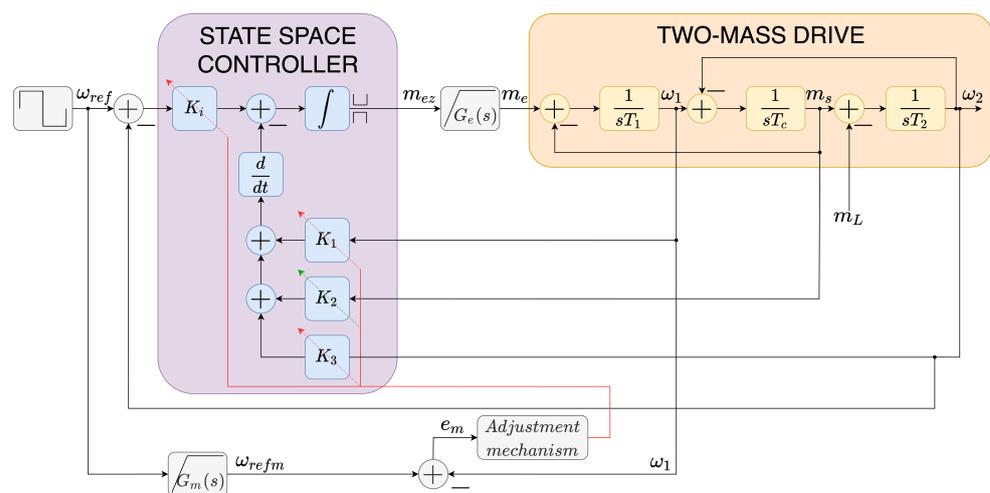
$$\frac{dm_s}{dt} = \frac{\omega_1 - \omega_2}{T_c} \quad (3)$$

where  $\omega_1$  and  $\omega_2$  represent the rotational speed of the motor and the load (respectively),  $m_e$  is the electromagnetic torque,  $m_s$  is the torsional torque and  $m_L$  is the load torque.  $T_1$ ,  $T_2$  and  $T_c$  are the mechanical time constants of the motor, the load and the shaft connecting the two masses. In this paper, the frictional forces are not taken into account.

To represent the control structure more clearly, the torque control loop is simplified to a first-order transfer function  $G_e$  with a time constant of  $T_{me}$

$$G_e = \frac{1}{T_{me}s + 1} \quad (4)$$

The whole proposed control scheme is based on a cascade structure. It consists of the inner control loop with a PI controller responsible for establishing the electromagnetic torque, and the outer loop with the proposed adaptive state feedback controller, which controls the angular velocity of the drive. It must be noted that for the purpose of this study, the drive is assumed to be fully observational (all of the state space variables are measurable). Even though the torque control loop is represented as a simple transfer function, in the practical implementation the output of the torque controller is limited. The simplification of the electromagnetic loop  $G_e$  is also related to a small time constant of the current loop. It should be mentioned that with the current advances in power electronics, the delay caused by the power converter's switches does not exceed 5 ms, which is negligible when compared to the mechanical time constant responsible for shaping the output speed. This approximation simplifies the calculation provided in the next sections.



**Figure 2.** The proposed control scheme.

## 2.2. A State Controller with Fixed Parameters Implemented for the Two-Mass System

The proposed adaptive speed controller is based on a classic state controller. The initial gain values for the adaptation process are selected as for the classic version. In the following analysis,  $G_e$  is assumed to be equal 1.

In addition to the system of Equations (1)–(3), the electromagnetic torque can be described as

$$m_e = \int \left( K_i(\omega_{ref} - \omega_2) - \frac{d}{dt}(K_1\omega_1 + K_2m_s + K_3\omega_2) \right) dt \quad (5)$$

which, using the Laplace operator  $s$ , can be rewritten as

$$m_e = \frac{K_i}{s}(\omega_{ref} - \omega_2) - (K_1\omega_1 + K_2m_s + K_3\omega_2) \quad (6)$$

Parameters  $K_i$ ,  $K_1$ ,  $K_2$  and  $K_3$  are the gains of the controller (scalar values, meaning they are dimensionless). Combining Equations (1)–(3) and (6), the closed-loop transfer function can be represented as

$$\frac{\omega_2}{\omega_{ref}} = \frac{K_i}{s^4 T_1 T_2 T_c + s^3 K_1 T_2 T_c + s^2 (T_1 + T_2 + K_2 T_2) + s(K_1 + K_3) + K_i} \quad (7)$$

The characteristic polynomial of the control structure has the following form:

$$H(s) = s^4 + s^3 \frac{K_1}{T_1} + s^2 \left( \frac{1}{T_1 T_c} + \frac{1}{T_2 T_c} + \frac{K_2}{T_1 T_c} \right) + s \frac{K_1 + K_3}{T_1 T_2 T_c} + \frac{K_i}{T_1 T_2 T_c} \quad (8)$$

To obtain the values for the controller's gains, the coefficients of the polynomial  $H(s)$  must be compared with the values of the coefficients in the reference polynomial  $R(s)$ . The polynomial  $R(s)$  is chosen arbitrarily; however, both polynomials must be the same order. For this study, the polynomial  $R(s)$  is assumed as

$$R(s) = (s^2 + 2\zeta_r \omega_0 s + \omega_0^2)^2 \quad (9)$$

where  $\zeta_r$  is the desired damping coefficient, and  $\omega_0$  is the desired resonating angular frequency. As a result of the comparison, the following equations are derived:

$$4\zeta_r \omega_0 = \frac{K_1}{T_1} \quad (10)$$

$$2\omega_0^2 + 4\zeta_r^2 \omega_0^2 = \frac{1}{T_1 T_c} + \frac{1}{T_2 T_c} + \frac{K_2}{T_1 T_c} \quad (11)$$

$$4\zeta_r \omega_0^3 = \frac{K_1 + K_3}{T_1 T_2 T_c} \quad (12)$$

$$\omega_0^4 = \frac{K_i}{T_1 T_2 T_c} \quad (13)$$

and finally, the expressions describing the feedback gains of the state controller are obtained:

$$K_1 = 4\zeta_r \omega_0 T_1 \quad (14)$$

$$K_2 = T_1 T_c \left( 2\omega_0^2 + 4\zeta_r^2 \omega_0^2 - \frac{1}{T_1 T_c} - \frac{1}{T_2 T_c} \right) \quad (15)$$

$$K_3 = T_1 T_2 T_c 4\zeta_r \omega_0^3 - K_1 \quad (16)$$

$$K_i = \omega_0^4 T_1 T_2 T_c \quad (17)$$

The expressions presented above were applied for the tuning of the controller with fixed parameters. The values were also used as the initial points for the adaptive versions of the speed controller.

### 2.3. Scenario 1—Partial Adaptation of the Parameters Used in the State Controller

The adaptation of weights in a simple neural model can be achieved through the use of the delta rule [55]. Here, similar calculations inspired by the fundamentals from the theory of artificial intelligence are proposed. In the state controller, the feedback coefficients are used to recalculate the signal values. Then the signals are combined into one element (a sum of values, similar to a neuron with a linear activation function). In order to reduce the state error, an additional integral element is often implemented (main path with error). In consequence, a form of integral control with state feedback is obtained.

In these considerations, for the updates of the speed controller gains, a simplification is assumed—direct multiplication (without integration) is considered. Additionally, the basic form of the full state feedback controller is analyzed. According to the conditions mentioned above, additional issues related to the limitation of signals are omitted. Then, the output of the state controller (the reference electromagnetic torque value), can be rewritten as follows:

$$m_{ez} = -K_i e - K_1 \omega_1 - K_2 m_s - K_3 \omega_2 \quad (18)$$

where:

$$e = \omega_{ref} - \omega_2 \quad (19)$$

The main goal of coefficient optimization is to reduce the error defined as

$$e_m = \omega_{refm} - \omega_1 \quad (20)$$

where  $\omega_{refm}$  is obtained by passing the reference signal  $\omega_{ref}$  through a model of the desired dynamics. In this paper, the model takes the following form:

$$G_m(s) = \frac{\omega_0^2}{s^2 + 2\zeta_r \omega_0 s + \omega_0^2} \quad (21)$$

The parameters of the state controller are updated using the following formulas:

$$K_i(k+1) = K_i(k) + \Delta K_i(k) \quad (22)$$

$$K_1(k+1) = K_1(k) + \Delta K_1(k) \quad (23)$$

$$K_3(k+1) = K_3(k) + \Delta K_3(k) \quad (24)$$

where  $k$  is the number of the current iteration.

The novelty of the presented research lies within the adaptation of the variable  $K_2$ . The proposed adaptation algorithm for  $K_2$  will be described in the subsequent section. Before that, the updates of the  $K_i$ ,  $K_1$ , and  $K_3$  are analyzed. First, Equations (22)–(24) are redefined:

$$K_i(k+1) = K_i(k) - \alpha g_i(k) \quad (25)$$

$$K_1(k+1) = K_1(k) - \alpha g_1(k) \quad (26)$$

$$K_3(k+1) = K_3(k) - \alpha g_3(k) \quad (27)$$

where  $\alpha$  is the adaptation coefficient, and  $g_i$ ,  $g_1$  and  $g_3$  are the partial derivatives of the error function with respect to their corresponding variables:

$$g_i(k) = \frac{\partial E(k)}{\partial K_i} \quad (28)$$

$$g_1(k) = \frac{\partial E(k)}{\partial K_1} \quad (29)$$

$$g_3(k) = \frac{\partial E(k)}{\partial K_3} \quad (30)$$

The mentioned error function  $E$  can be taken from the neural network theory. It can be defined as

$$E = \frac{1}{2} (d(k) - y(k))^2 = \frac{1}{2} (e_m(k))^2 \quad (31)$$

where  $d$  is the desired value from the model, and  $y$  is the output value. For this application, the output of the neural model  $y$  can be defined using (18). According to the issues considered above, the adjustments of the gains in the analyzed controller are described with the following equations [56]:

$$\Delta K_i(k+1) = -\alpha e_m(k) e(k) \quad (32)$$

$$\Delta K_1(k+1) = -\alpha e_m(k) \omega_1(k) \quad (33)$$

$$\Delta K_3(k+1) = -\alpha e_m(k) \omega_2(k) \quad (34)$$

The definition of the correct value is hard to describe using mathematical formulas. Thus, in this work, the selected metaheuristic algorithms were applied for this purpose.

The stability of the presented structure can be analyzed in two main ways. The experimental and the theoretical approaches can be adopted. To check the stability of the given system, the trial and error method for different values of the initial weights and the learning coefficients can be applied. On the other hand, there are theoretical methods for ensuring the stability of a given system. One of the most commonly used methods is the Lyapunov's stability theorem. It is applied in a wide range of plants with adaptive controllers. However, this work is focused on the implementation and tests of a modified adaptive state controller.

#### 2.4. Scenario 2—A Novel Approach to the Adaptive State Space Controller—Indirect Adaptation of $K_2$

The first version of the adaptive state controller uses recalculation of the parameters based on the speed error. The gain of the path with the  $m_s$  signal is constant. This assumption is related to problems with the accessibility of the reference signal for the torsional torque. Otherwise, the method (from the previous section) of adaptation should be extended also for the  $K_2$  parameter. However, the speed and the torque have different dynamics and shapes; they eliminate the adaptation of this coefficient similar to the others. Thus, an additional modification is proposed. Using Equations (14)–(17) the formula for  $K_2$  can be rewritten as

$$K_2 = 2T_1 T_c \omega_0^2 (1 + 2\zeta_r^2) - \frac{T_1}{T_2} - 1 \quad (35)$$

The following can also be observed

$$T_1 = \frac{K_1}{4\zeta_r \omega_0} \quad (36)$$

$$T_2 = \frac{K_i}{T_1 T_c \omega_0^4} = \frac{4K_i \zeta_r}{K_1 T_c \omega_0^3} \quad (37)$$

By inserting (36) and (37) into (35)

$$K_2 = \frac{K_1 T_c \omega_0}{2\zeta_r} - \frac{K_1^2 T_c \omega_0^2}{16\zeta_r^2 K_i} + K_1 T_c \zeta_r \omega_0 - 1 \quad (38)$$

After recalculations, the new formula for  $K_2$  is achieved:

$$K_2 = \frac{K_1 T_c \omega_0 (8K_i \zeta_r - K_1 \omega_0)}{16\zeta_r^2 K_i} + K_1 T_c \zeta_r \omega_0 - 1 \quad (39)$$

With this description, the parameter  $K_2$  now becomes indirectly subjected to adaptation. It is dependent on other gains of the controller ( $K_i$  and  $K_1$ ), which are already changing their value, causing  $K_2$  to also change its value alongside them. What is worth

noting is that now, that value is independent from the time constant of the load machine  $T_2$ , which is the most changing part for this expression.

### 3. Nature Inspired Optimization Algorithm—The Symbiotic Organism Search

The symbiotic organism search is a nature-inspired algorithm that simulates the behavior of organisms spotted in their natural habitats [57]. None of the organisms live alone. The relationships between them can be described as symbiosis, which stands for “living together”, where two or more species bring benefits to one or more parties involved. Depending on the result of the symbiosis, different relationships can be formed: commensalism, mutualism, and parasitism. Mutualism is the most common form of symbiosis between different species where two organisms benefit from the relationship. Commensalism describes the two organisms living together, where only one benefits but the other one is not harmed. The third relationship, parasitism, benefits only the parasite, while the other organism, called the host, is harmed. The SOS describes only the three aforementioned relationships from all six possible relationships.

The SOS uses the population of organisms to solve the numerical optimization task over the course of many iterations. The generated population is used to model the three symbiotic behaviors. A member of the population or ecosystem is randomly generated with constraints to the limits of the optimization task:

$$x_i = UB + rand(0,1) \cdot (UB - LB) \quad (40)$$

where  $x$  is the newly generated solution,  $i$  is the number of the specimen, and  $LB$  and  $UB$  are the lower and the upper bounds of the optimization problem, respectively. Then, the fitness function for the whole ecosystem is evaluated, and the best candidate should be found—the one with the lowest value of the fitness function for the minimization problems. In this paper, the following fitness function is used:

$$F_{cost} = \sum_{k=1}^N \frac{|\omega_1(k) - \omega_2(k)|}{N} \quad (41)$$

where  $N$  is the number of samples of the simulation.

After finding the best organism, the mutualism phase takes place. As in all forms of symbiosis, two organisms are needed, so apart from the currently selected individual  $x_i$ , another one ( $x_j$ ) is randomly selected from the population. Both organisms are coupled for a mutualistic relationship, the goal of which is to increase the survival rate of both. To calculate the new solutions, mutual vector ( $MV$ ), as well as beneficial factors ( $BF$ ) should be calculated first [58].

$$MV = \frac{x_i + x_j}{2} \quad (42)$$

$MV$  represents the relationship between both organisms, while  $BF$  models the level of benefit from  $i$  or  $j$  organism.  $BF$  is randomly selected as 1 or 2. New values of  $x_i$  and  $x_j$  are then created using the following formula [50]:

$$x_i^{new} = x_i + rand(0,1) \cdot (x_{best} - MV \cdot BF_1) \quad (43)$$

$$x_j^{new} = x_j + rand(0,1) \cdot (x_{best} - MV \cdot BF_2) \quad (44)$$

where  $x_{best}$  is the fittest individual.

$x_{best}$  is incorporated into Equations (43) and (44) as the population tries to evolve according to Darwin’s law—“survival of the fittest”. Later, the fitness function of the evolved organism is evaluated. If the new values are better,  $x^{new}$  takes the place of  $x$ ; in the other case,  $x$  is saved in the ecosystem. The effect of the mutualism phase and the effect of  $MV$  is presented in Figure 3 below.

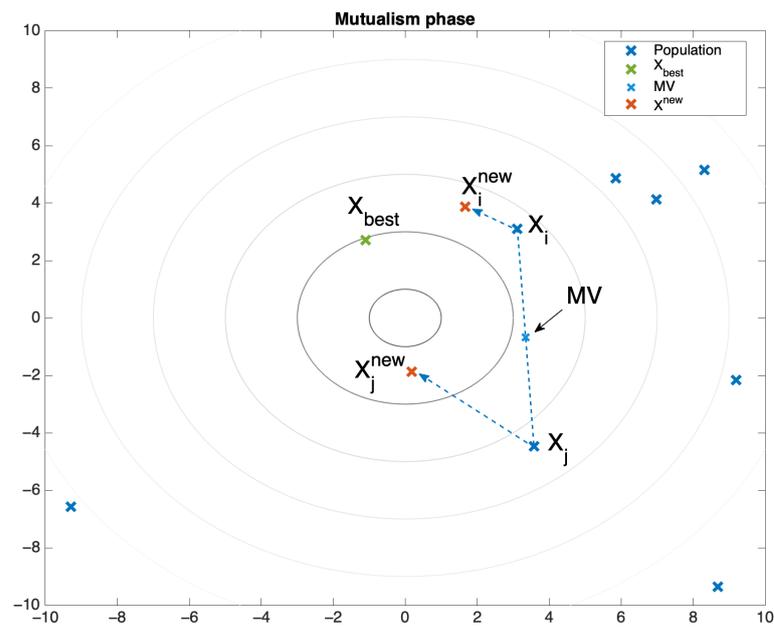


Figure 3. Visualization of the mutualism phase.

The commensalism phase comes later, where only one organism benefits from the relationship. It can be modeled in the same way without the usage of the *MV* and *BF*. The new value of the organism is generated based on the best individual and the randomly selected organism (45) [59].

$$x_i^{new} = x_i + rand(-1, 1) \cdot (x_{best} - x_j) \quad (45)$$

The effect of the commensalism phase is depicted in Figure 4.

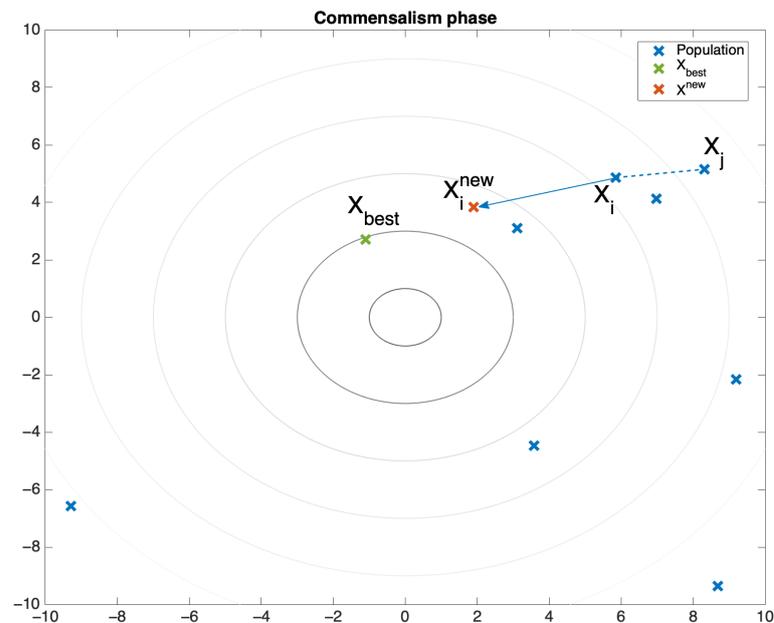


Figure 4. Visualization of the commensalism phase.

The last part of the algorithm consists of the parasitism phase. The parasitism phase utilizes the so-called parasite vector (*PV*) [60]. It is created by duplicating and modifying random dimensions from a selected organism  $x_i$ . Then, another organism  $x_j$  is randomly selected that can be treated as a host to a parasite  $x_i$ . The goal of the parasite is to replace

the host in the ecosystem. The fitness function of both organisms is evaluated, and the fittest one stays in the population. If the PV has a better value of the fitness function, it replaces the host position in the ecosystem. Otherwise, if the host is immune to the parasite, it is saved in the population and the PV is erased. There are many different ways to implement the parasitism phase. In some cases, creating the PV just means creating a new organism [50,61]. In other implementations, some dimensions are changed, and this is dependent on the version of the algorithm [57,59]. This application changes only one dimension of the selected organism. The visualization of the last phase is shown in Figure 5. The basic parameters of the SOS are presented in Table 1 below. The whole algorithm can be summarized in the block diagram depicted in Figure 6.

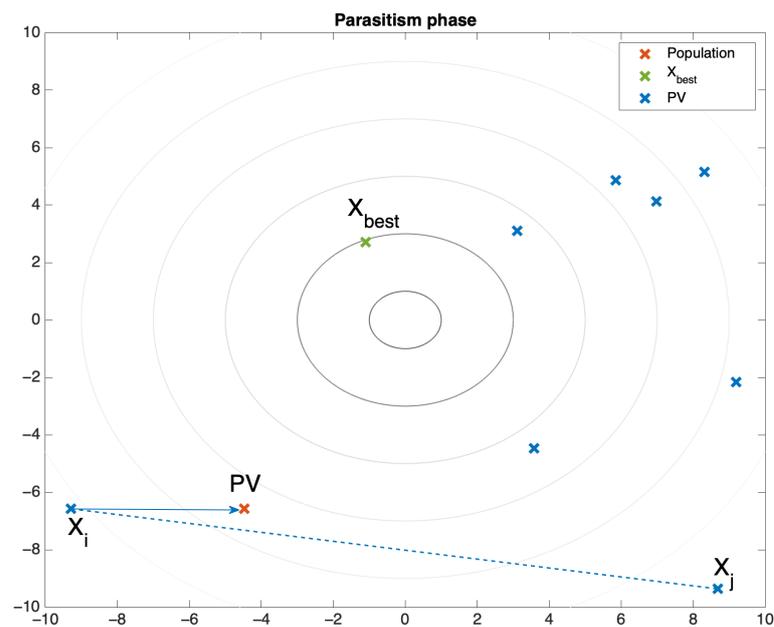


Figure 5. Visualization of the parasitism phase.

Table 1. Parameters of the symbiotic organism search algorithm.

Parameter	Value
Population size	20
Main loop iteration count	20
$[LB, UB]$	$[-0.5, 0]$
Max. time for a main loop	120 min

Changes in the fitness function are presented in Figure 7 below. The minimum value achieved by the algorithm was equal to  $F_{cost} = 1.253 \times 10^{-3}$ , it was also marked with the light blue transient. As presented, even the value achieved in the first iteration was close to the final result, which shows that this task was relatively simple for a metaheuristic algorithm to calculate.

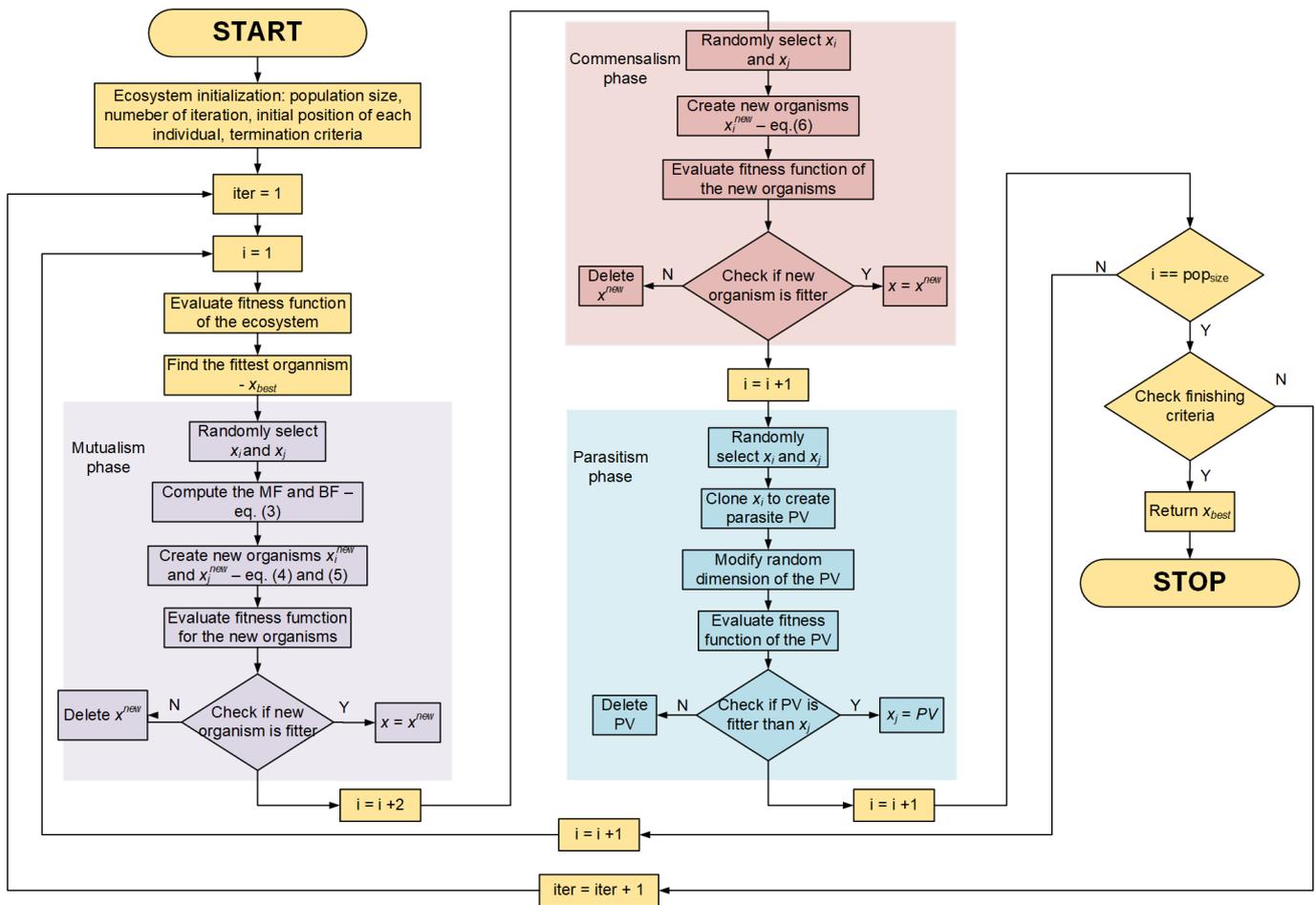


Figure 6. The block diagram of the symbiotic organism search algorithm.

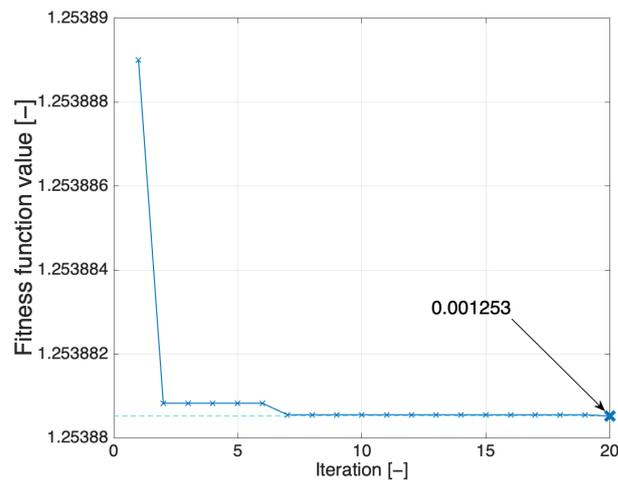


Figure 7. Changes in the fitness function during the optimization process.

#### 4. Nature Inspired Optimization Algorithm—The Flower Pollination Algorithm

Currently, a significant increase in applications of metaheuristic optimization algorithms may be observed, especially for solving non-linear, complex functions. As it was presented in the introduction of the paper, a few parameters of the controller have to be determined in order to ensure the expected dynamics of the plant. The presented method of pole placement may only be used if all parameters of the controlled plant are properly identified [12]. That method is not appropriate for closed-loop control when any elements

of artificial intelligence are employed. The stability issues may be observed in such cases when the controller is tuned with the standard method. This is caused by the fact that during the synthesis, such elements are not taken into account in the mathematical model of the control structure [62]. However, the issue may be eliminated by the approach of the direct tuning of closed-loop systems. The initial controller coefficients may be tuned during the operation of the plant. However, the more efficient and safer solution is to perform the tuning during simulations. There are different algorithms that may be employed for such a task [63]. In this paper, the nature-inspired algorithms were examined. For comparison purposes, the tuning was also conducted with the flower pollination algorithm.

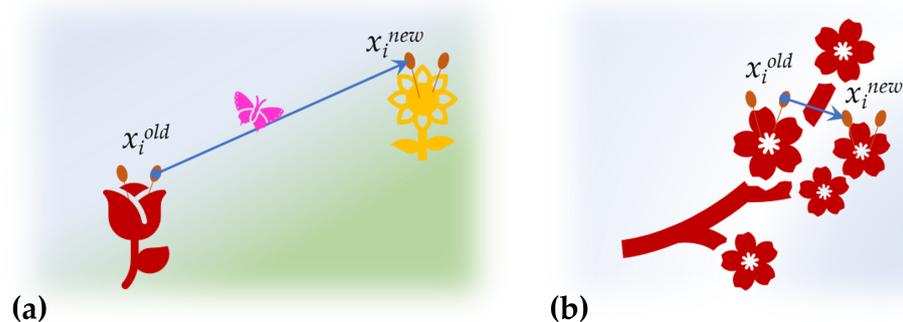
The flower pollination algorithm is a metaheuristic algorithm that mimics the natural process of pollen transfer. The algorithm was presented by Xin-She Yang in 2012 [64]. The algorithm is a universal tool for optimization in different areas of science, economics and also social behavior studies [65–68]. It belongs to swarm-based metaheuristic algorithms. This means that in every iteration, the initial population is improved because only the strongest specimens are able to survive. The strength of every specimen equals the value of the fitness function. Because the tuning process is a minimization task, the smaller the obtained value of the cost function, the better the solution. In the considered application, the same cost function that was employed with the SOS algorithm was proposed. The difference between the angular velocity of the electrical motor and the load machine is calculated for every sample. Then the mean of the absolute values is calculated.

The flower pollination algorithm is based on the observations of pollen transfer. Two processes of this phenomenon may be differentiated in nature. The first one is called cross-pollination, and it requires the specimens of two different plants to take part. Because of the fact that the pollen has to be moved over longer distances in cross-pollination, pollinators (insects) are required to move pollen between distant flowers (Figure 8a). The movement of the insects is considered a global, random process. This leads to the suggested mathematical model of such movement, called Lévy flight (46). The random movement is obtained by the  $\Gamma(\lambda)$  parameter [69].

$$L = \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi s^{1+\lambda}} \quad (46)$$

where  $\Gamma(\lambda)$  is the standard gamma function for the index  $\lambda$ ,  $s$  is a random step size from the Lévy distribution.

The second mechanism of pollen transfer is independent from external creatures and may be driven only by wind. It employs flowers of the same plant, and is called self-pollination (Figure 8b). This process allows the plant to survive; however, the features of the new specimen are very similar to its predecessors. Because of that, it is considered local and depends only on specimens from the current population.



**Figure 8.** Pollen transfer mechanisms: (a) cross-pollination, (b) self-pollination.

Global and local optimization is required to obtain satisfactory results. Global optimization ensures that the calculations do not stop at the local extremum and the global minimum is obtained, while local optimization is required to obtain even better approx-

imation of the fittest solution. These two mechanisms are switched randomly in every iteration. The drawn value  $r$  is compared to the probability switch parameter  $P$ . Depending on the value of  $r$ , the current iteration may be calculated either as global optimization (Equation (47)) or local optimization (Equation (48)). After the new solution is calculated, the cost function value is checked. If the improvement is noted, the  $i$ -th specimen in the population  $x_i^{old}$  is replaced with a new solution  $x_i^{new}$ . After all the iterations are computed, the global best result is returned. The flow of data during the optimization with the FPA is shown in Figure 9.

$$x_i^{new} = x_i^{old} + \gamma L(\lambda)(g^* - x_i^{old}), \tag{47}$$

$$x_i^{new} = x_i^{old} + \epsilon(x_j^{old} - x_k^{old}), \tag{48}$$

where  $x_i^{new}$  is the new solution,  $x_i^{old}$  is the value of the  $i$ -th specimen in the current population,  $\gamma$  is the step size for the global pollination,  $L(\lambda)$  is the calculated Lévy flight step,  $g^*$  is the current best solution,  $\epsilon$  is a random step size for the local pollination from Gaussian distribution, and  $x_j^{old}, x_k^{old}$  are values of two specimens chosen from the current population.

For research purposes, in order to receive comparable data, the population size was set to 20 specimens, and only 20 iterations were considered. The optimized variable was the  $\alpha$  coefficient, which is responsible for the adaptation speed. The bounds of the search range were defined basing on the knowledge about  $\alpha$  presented in [70]. All parameters are gathered in Table 2.

Table 2. Parameters of the flower pollination algorithm.

Parameter	Value
Population size	20
Main loop iteration count	20
Probability switch $P$	0.8
$[L_b, U_b]$	$[-0.5, 0]$

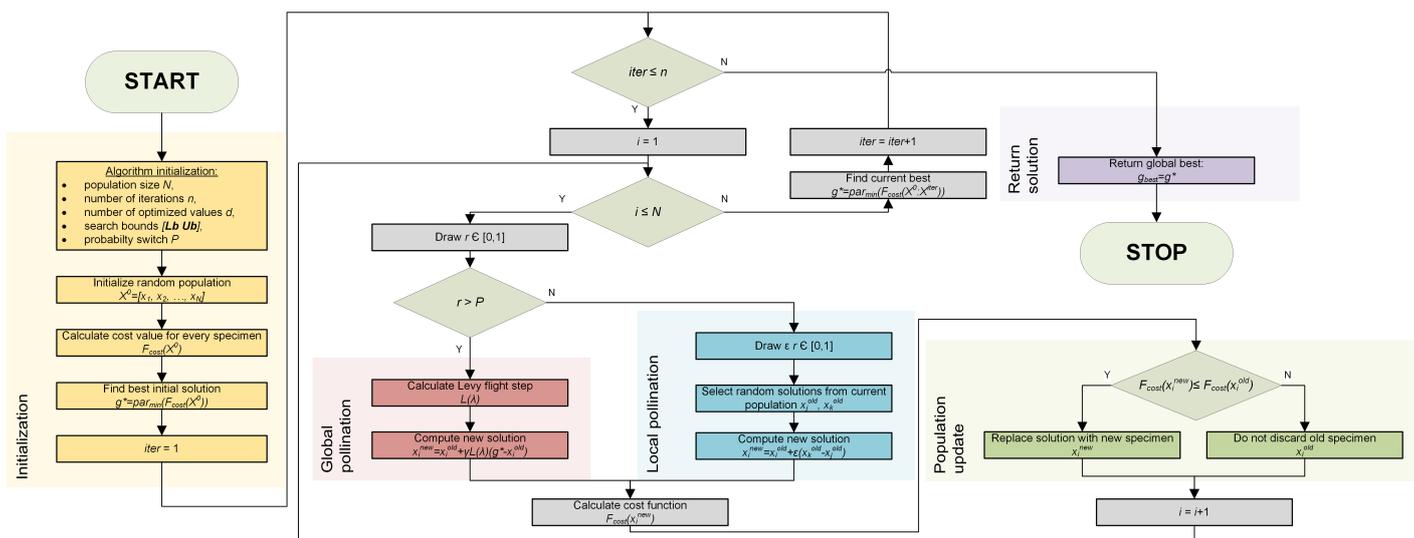
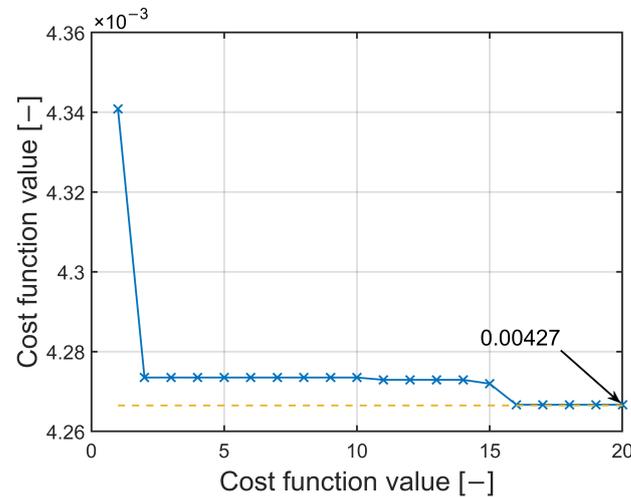


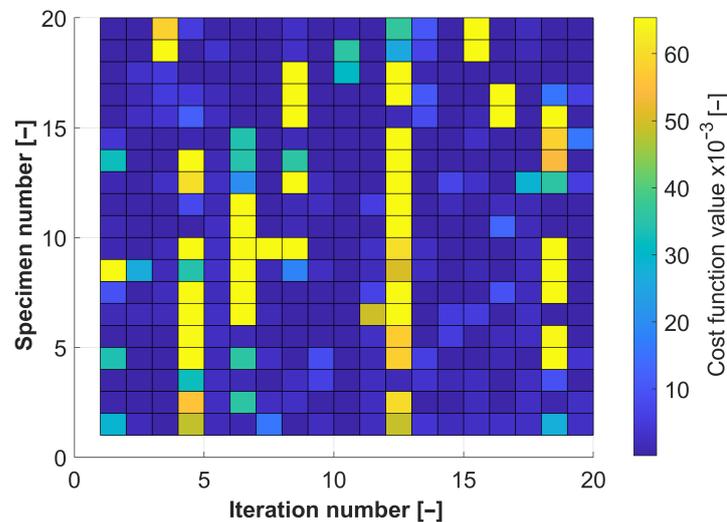
Figure 9. Block diagram of the flower pollination algorithm.

Even a small number of computed main loop iterations leads to improvement. The values of the fitness function in consecutive iterations are shown in Figure 10. In the given example, the obtained minimum value of the fitness function is equal to  $F_{cost} = 4.27 \times 10^{-3}$ . As it was already noted in the SOS description, the very first iteration quickly improves the solution close to the global best.



**Figure 10.** Best value of cost function during optimization process.

Because the process is based on the development of the populations - the improvement of every specimen in following iterations can be observed (Figure 11). This inheritance-based approach improves the optimization efficiency.



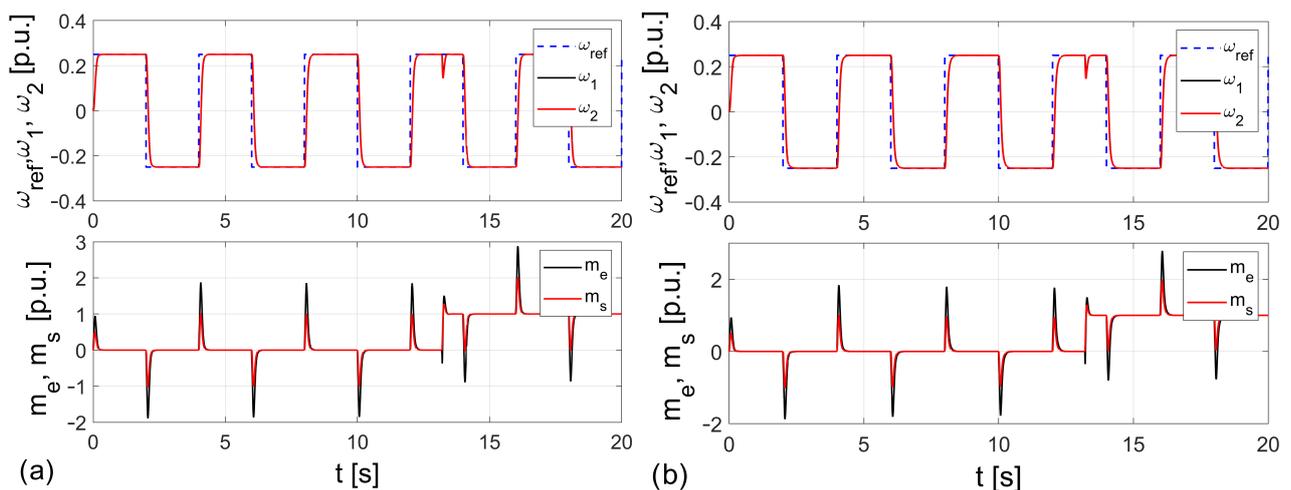
**Figure 11.** Optimization process—changes of population composition measured with cost function value.

Both presented algorithms use a pseudo-random number generator, which is a common practice in the meta-heuristic approach. The goal of the optimization process is to find a point close to the global minimum of the fitness function. Finding a point that is a global minimum is not always guaranteed when nature-inspired algorithms are used. The optimization process gives only the starting point to the gradient-based algorithm presented in the previous section. Then, the goal of the learning process of the controller is to obtain the global minimum point.

## 5. Simulation Tests

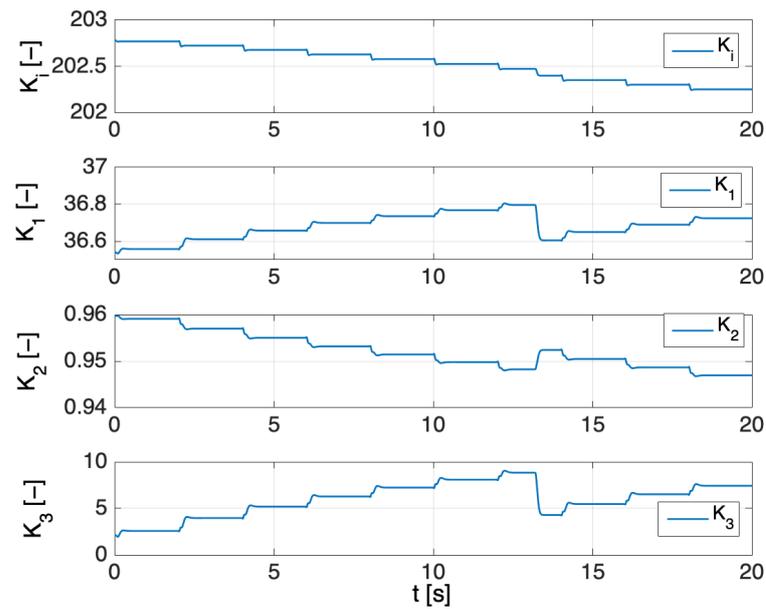
Simulation tests were carried out using the MATLAB/Simulink environment. The control structure was implemented in Simulink, and the data were fed through the MATLAB workspace. During the simulation time of  $t_{sim} = 20$  s, the drive performs reversions with the amplitude of  $\omega_{ref} = 0.25$  p.u. The load torque is applied in  $t_{ml} = 13.2$  s of simulation with a nominal value. All the state space variables were saved during the simulations.

The main point of this paper is to show that the adaptive state space controller can be implemented in many ways. One of them is to set the initial values of the adaptive gains as calculated in Equations (14)–(17), and optimize the value of the adaptation parameter  $\alpha$ . Transients for the nominal case for the SOS and the FPA optimization are shown in Figure 12. There is no overshoot or oscillations present in the speed transients. The drive precisely follows the reference speed. A high level of control accuracy is observed. The reaction of the state variables after the load switching is dynamic. Even though both figures look similar, the optimization parameter  $\alpha$  slightly differs between the algorithms. The SOS optimization yields  $\alpha_{SOS} = 0.0341$ , and the FPA yields  $\alpha_{FPA} = 0.0312$ . Correct adaptation can be noticed in the changes of the gains (Figure 13). Values of the adaptive parameters are changed after every reversion of the drive. It can also be seen that the load torque also has an impact on the values of the weights.

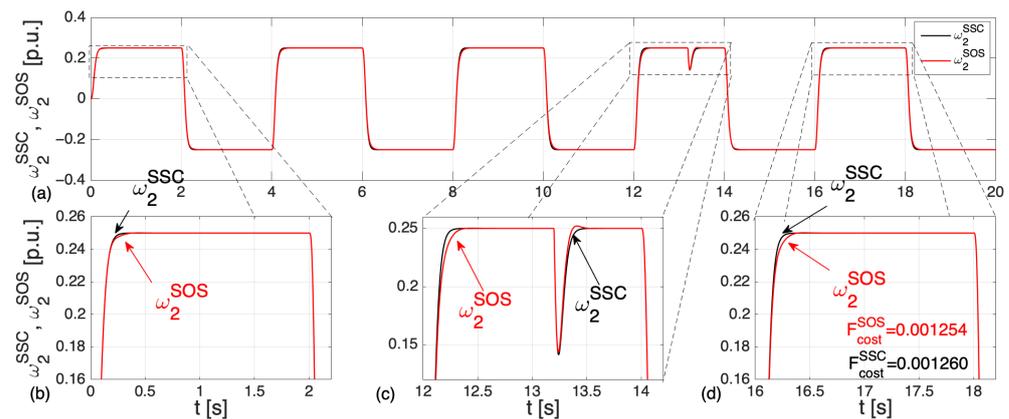


**Figure 12.** Changes in the state variables of the drive after the optimization process for the nominal parameters of the drive: (a) the SOS optimization, (b) the FPA optimization.  $\omega_1$  and  $\omega_2$ —motor and load speeds.  $m_e$  and  $m_s$ —electromagnetic and shaft torques.

To show the improvement over the classic state space controller, a comparison of the optimized adaptive structure (by the SOS) and the state space controller is also presented. The difference between drive performance with the SSC and the adaptive optimized controller (with only the value of the adaptive coefficient optimized) is shown in Figure 14a, and the zooms in Figure 14b–d. A minimal overshoot is visible after the load torque is applied. The speeds of both structures are similar during the course of the simulation, but the fitness function  $F_{cost}$  is slightly smaller in the adaptive scenario, which proves that the presented structure has higher dynamics.

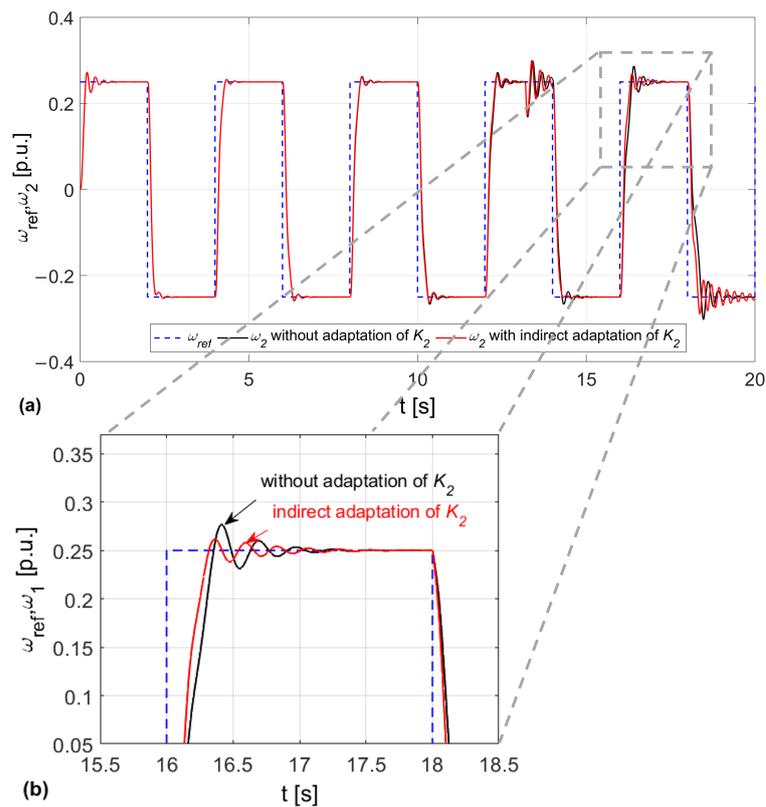


**Figure 13.** The state space controller gains changes in the drive with optimized learning parameter  $\alpha$  optimized by the SOS.



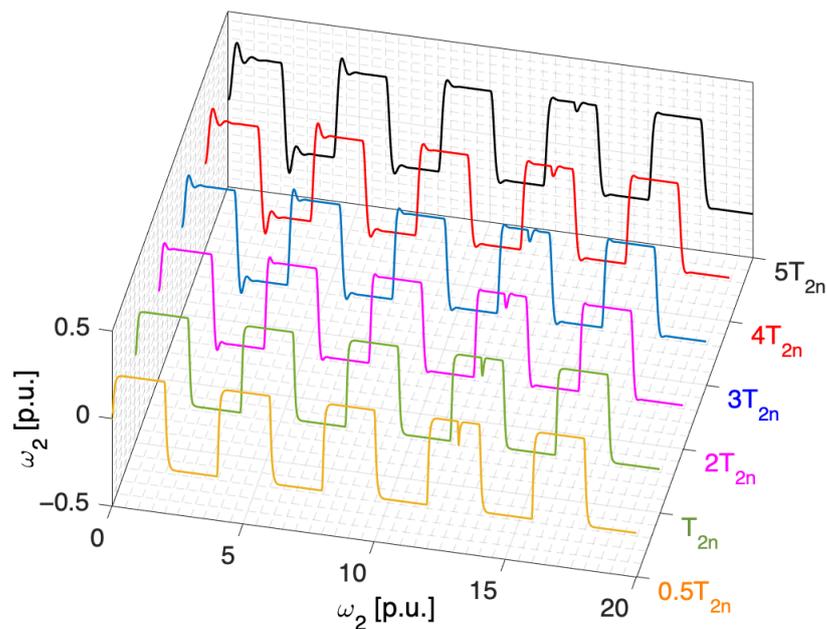
**Figure 14.** Comparison between the optimized control structure and the state space controller with constant gains: (a) the entire transient of the load motor speed, (b–d) zooms of transients for different parts of the simulation.  $\omega_2^{SSC}$ —load speed with the state space controller.  $\omega_2^{SOS}$ —load speed of the optimized state space controller.

The novelty of the presented structures lies in the indirect adaptation of the gains of the controller; it shows significant improvement for the dynamics of the drive. Comparison of classic adaptive state controller and the novel structure is shown in Figure 15. As it can be seen in the figures, a significant increase in adaptation of speed is observed when all coefficients are adaptable. Not only is the response quicker, but also the overshoot is clearly smaller (Figure 15b).



**Figure 15.** Comparison of changes in the state space variables in the drive without adaptation of  $K_2$  and the novel indirect adaptation of  $K_2$ : (a) the entire transient, (b) the zoom of the speed transient.  $\omega_2$ —the speed of the load.

Robustness against changes of the load time constant was also tested (Figure 16).

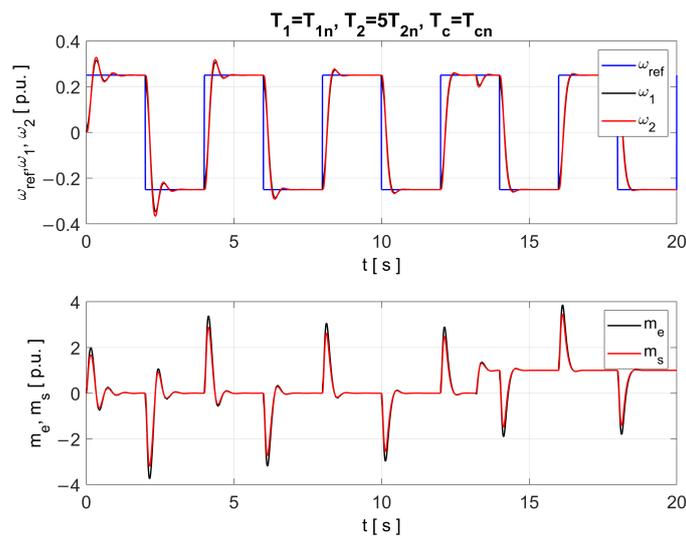


**Figure 16.** Influence of the changes in the load motor time constant in the system with optimized learning parameter  $\alpha$  by the SOS.

As the load time constant increases, the overshoot is higher in the startup process of the drive. Even though the time constant rises, overshoot is diminished due to the

changes in the gains of the controller. The maximum overshoot for the drive with the load time constant increased five times is about 36%. Lowering the load machine time constant increases the reaction to the load torque; other than that, the drive works in the same manner.

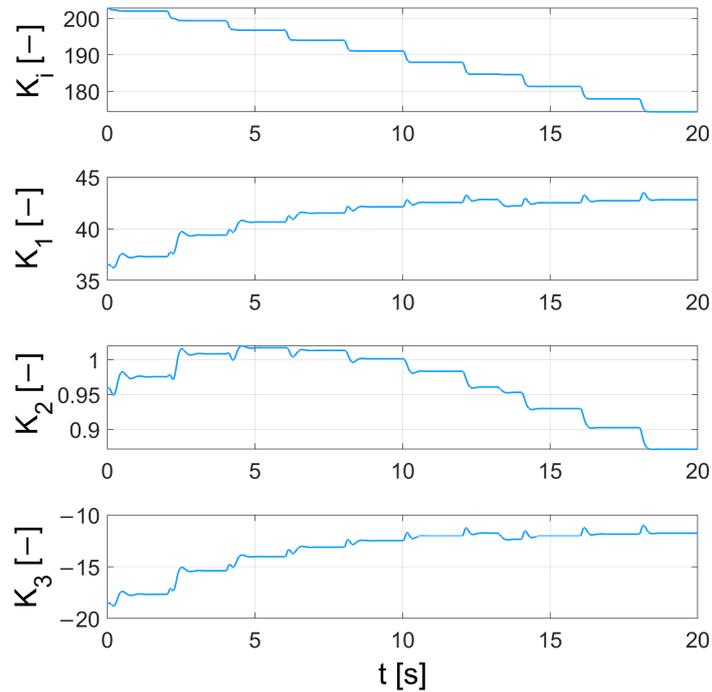
Similar tests were conducted for the FPA algorithm (optimization of all parameters of the controller). Figure 17 shows transients of the speeds and torques obtained for the changed load time constant. Only the worst case— $T_2 = 5T_{2n}$ —is presented, as the previously shown results (Figure 12) confirm that both algorithms yield similar solutions.



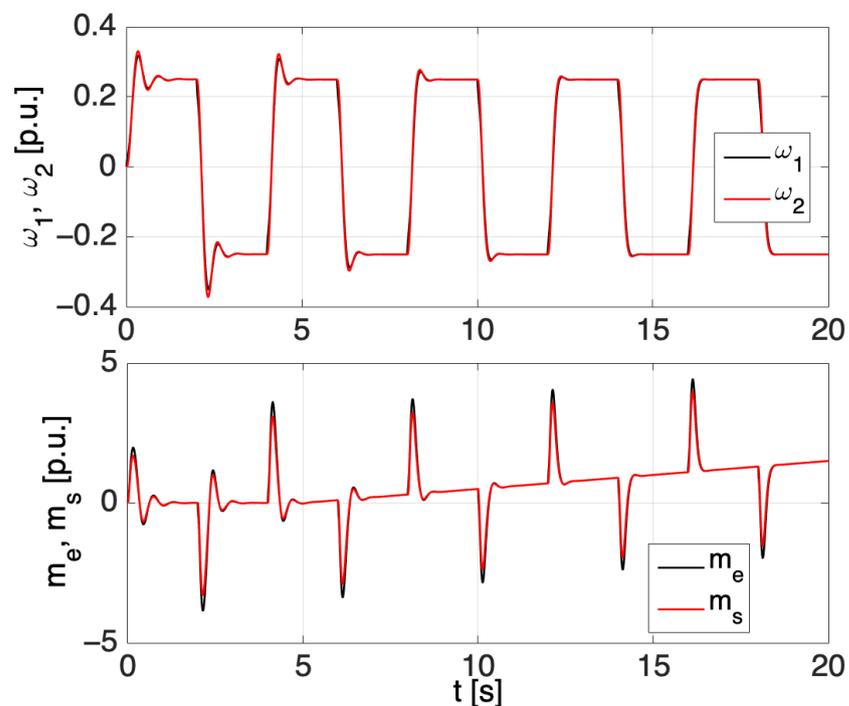
**Figure 17.** Changes in the state space variables in the drive with fully optimized structure,  $\alpha$  optimized by the FPA.  $\omega_1$  and  $\omega_2$ —the speed of the motor and the load.  $m_e$ ,  $m_s$ —the electromagnetic and the shaft torque.

The adaptation process is undoubtedly visible—the initial 20% overshoot is well damped after four reversions. The change (adaptation) of the state controller gains is presented in Figure 18. It is clear that the initial values of gains are not optimal for the plant (when the load time constant is increased). In comparison to the gains presented in Figure 13, the range of change is greater. During the first reversions, rapid change of the coefficients is observed after about 10 s.  $K_1$  and  $K_3$  settle at a stable level. It is worth indicating that the indirect adaptation of  $K_2$  is also visible. At the beginning, the value increases and then starts to decrease after every reversion.

The load torque (disturbance) in the previous tests was applied as a sudden appearance of the nominal value to test the robustness of the system. Such an example is rare in real applications. Thus, other tests were performed. The gradual changes (slope) of the load were also analyzed. After the initial phase (adaptation of the weights), the control system works stably. The load does not influence the precise tracking of the reference trajectory. However, the current values become higher in each reverse of the drive (Figure 19).



**Figure 18.** The adaptive state space controller gains changes in the drive with optimized learning parameter  $\alpha$  optimized by the FPA.

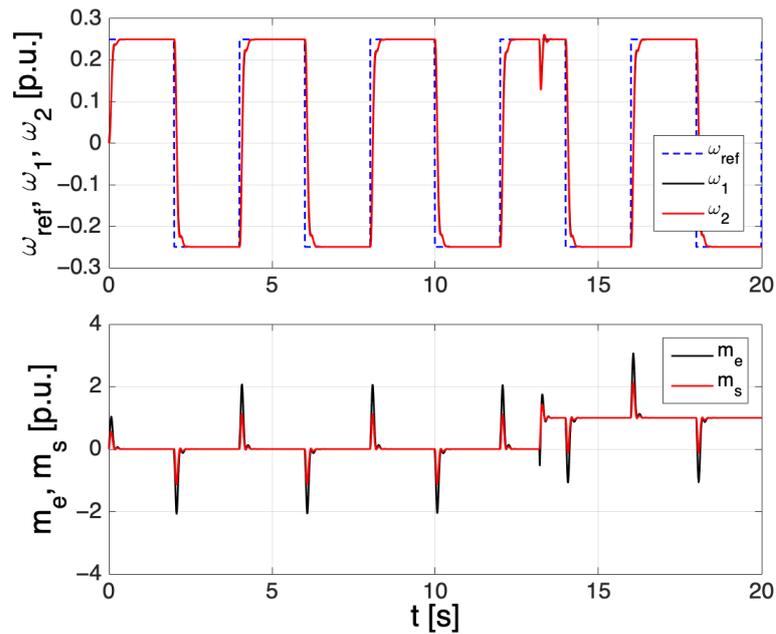


**Figure 19.** Changes in the state space variables in the drive with increased load time constant  $T_2 = 5T_{2n}$ .  $\omega_1$  and  $\omega_2$ —the speed of the motor and the load.  $m_e$ ,  $m_s$ —the electromagnetic and the torsional torque, with a continuous load applied after 5 seconds of the simulation.

Another way of implementing the adaptive state space controller is to optimize the initial values of the controller ( $K_i$ ,  $K_1$ ,  $K_2$ ,  $K_3$ ) as well as the learning coefficient. In this approach, all the values were optimized by a metaheuristic algorithm, which means that

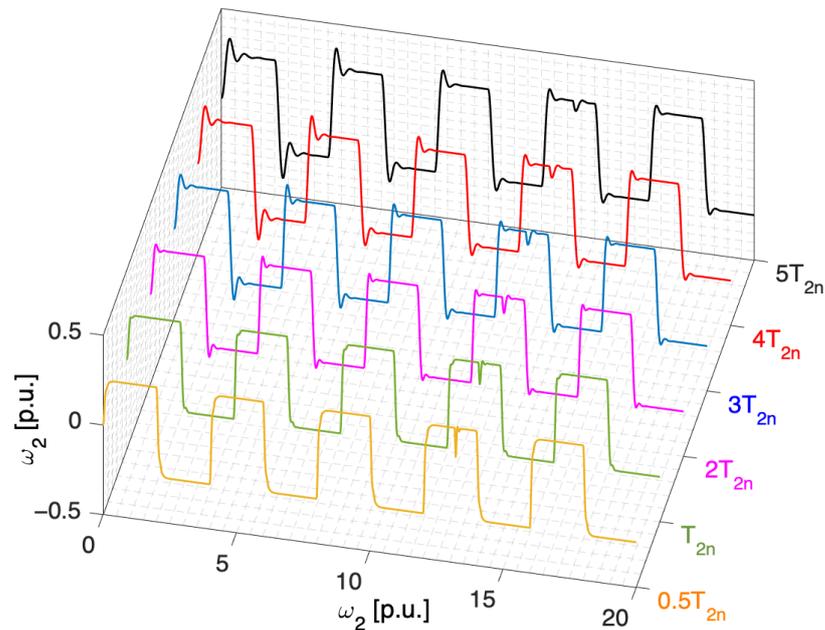
such parameters as  $\zeta_r$  and  $\omega_0$  cannot be determined. Thus, it means that the dynamics of the structure are entirely selected by the algorithm and the fitness function.

The fully optimized structure by the SOS algorithm is presented in Figure 20. There is a minimal overshoot as the speed recovers from the application of the load torque. The dynamics of the whole structure are similar to the one calculated using Equations (14)–(17), which proves that the SOS (or any other metaheuristic algorithm) can find the solution near the optimal point (depending on the fitness function and the difficulty of the problem). The torque is generated in a fast manner, and it is not saturated by the limits of the speed controller.



**Figure 20.** Changes in the state space variables in the drive with fully optimized structure,  $\alpha$  optimized by the SOS.  $\omega_1$  and  $\omega_2$ —the speed of the motor and the load.  $m_e$ ,  $m_s$ —the electromagnetic and the torsional torque.

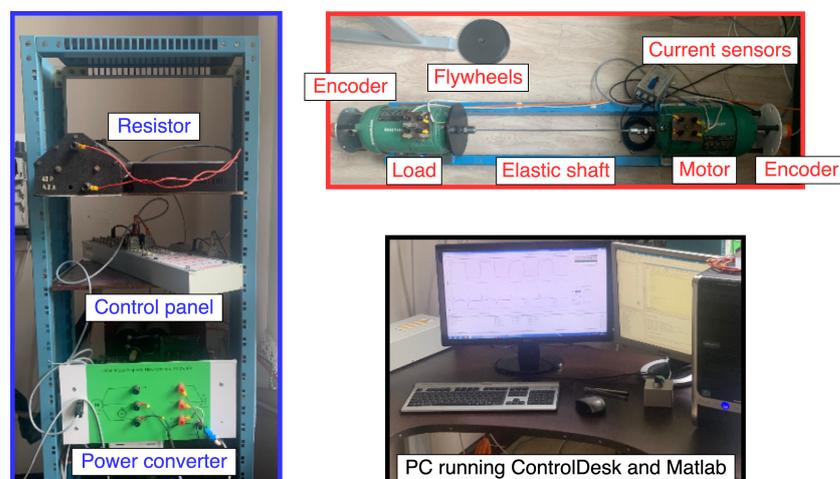
The influence of the load time constant  $T_2$  was also checked and is presented in Figure 21. The overshoot of the load speed, in the startup operation of the drive, increases alongside the increase in  $T_2$ , though it is always damped as the adaptation progresses. Although the time constant of the load machine is high, the drive can follow the reference speed precisely or with a minimal overshoot.



**Figure 21.** Influence of the changes in the load motor time constant in the fully optimized system.

## 6. The Experimental Verification

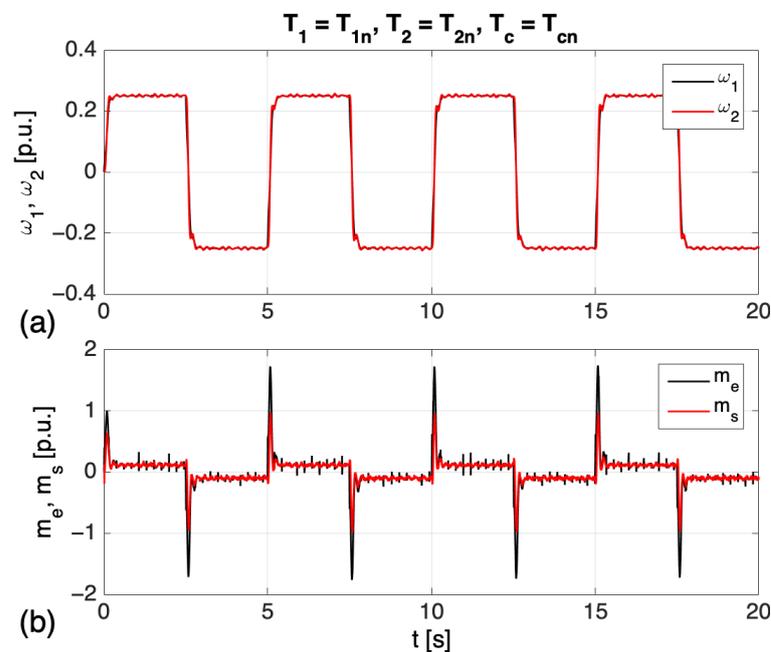
In order to confirm the satisfactory dynamics obtained with simulations, the results were also verified with laboratory equipment. In this part of the paper, the hardware implementation and the initial results are described. The laboratory stand consists of two electrical motors (with the nominal power of 0.5 kW) connected with a flexible shaft. While the first motor is the controlled plant, the second one acts as the load. Time constants of the system may be changed by adding flywheels and replacing the shaft. The control structure is implemented on dSPACE 1103 card. The current feedback is realized with LEM sensors, and the speed feedbacks are achieved with encoders attached to each of the motors. The laboratory stand is shown in Figure 22. The first tests were conducted for the nominal parameters of the drive and the standard state space controller in order to check if the plant was correctly identified for simulations. The results proved that the response of the plant is similar to the modeled one.



**Figure 22.** The laboratory equipment used for the experimental verification (blue—power electronics; red—the drive; black—PC, software).

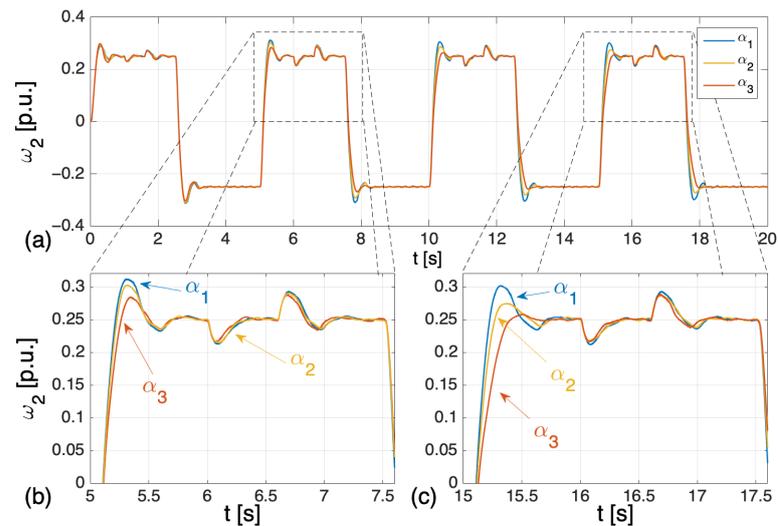
In the next step, the suggested novel adaptive controller with indirect adaptation of  $K_2$  was tested. Figure 23 shows the speed of the motor and the load with adaptive controller (scenario 2 in Section 2.4). These tests were initially performed for the nominal parameters of the drive and the state space controller with optimized value of  $\alpha$ . Stable work of the system and precise tracking of the reference signal as well as the high dynamics of the generated torque are observed.

Next tests were conducted for the scenario where the identification of the plant was performed inaccurately or the load of the drive was changed. This was obtained by the change of  $T_2$  variable through the attachment of additional flywheels to the second motor shaft. The results with a higher time constant of the load machine  $3T_{2n}$  are presented in the figure below. Different values of the adaptation parameter  $\alpha$  were applied in increasing order ( $\alpha_1 < \alpha_2 < \alpha_3$ ). Experimental results prove the results obtained in the simulation part. As the adaptation coefficient increases, the overshoot is reduced for the startup part of the experiment. Overshoot is damped during the process of adaptation in every reversion. It is also observed when an additional flywheel is added to the shaft (for the sum of  $5T_{2n}$ ).



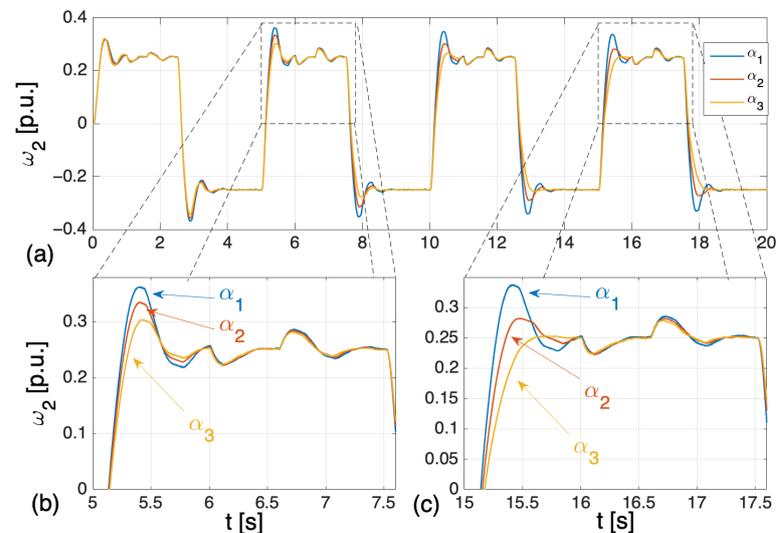
**Figure 23.** Transients of the state variables for nominal time constants, in the control structure with the adaptive state controller: (a) the motor and the load speed, (b) the electromagnetic and the torsional torque.

The second part of the test shows work of the control structure after changes of the mechanical time constant of the load ( $T_2$ ). Results with higher time constant of the load machine ( $3T_{2n}$ ) are presented in Figures 24 and 25 for  $5T_{2n}$ . Different values of the adaptation parameter  $\alpha$  are applied—in increasing order ( $\alpha_1 < \alpha_2 < \alpha_3$ ). The experimental results prove the results obtained in the simulation part. The main trend is similar: as the adaptation coefficient increases, the overshoots are more quickly reduced. The adaptation process leads to damping of the oscillations and correct operation after parametric disturbances.



**Figure 24.** The influence of the  $\alpha$  parameter on the state variables of the control structure with the adaptive speed controller: (a) the entire transient of the load speed, (b,c) zoom used for selected parts of the results. Tests performed for  $T_2 = 3T_{2n}$ .

Even the significant increase of  $T_2$  can be handled by the proposed adaptive controller. The transients presented below show that the unwanted oscillations are not present, even for the situation where  $T_2 = 5T_{2n}$ .



**Figure 25.** The influence of the  $\alpha$  parameter on the state variables of the control structure with the adaptive speed controller: (a) the entire transient of the load speed, (b,c) zoom used for the selected parts of the results—tests performed for  $T_2 = 5T_{2n}$ .

## 7. Concluding Remarks

The paper deals with different approaches to a state controller applied for a two-mass system. The initial structure has a typical form, which uses signals from the mechanical part of the drive. Then, additional adaptation for the parameters of the controller is implemented. Two solutions of the adaptive speed control are considered (novel indirect method of adaptation of one of the coefficient was also considered). Selected optimization techniques are used to optimize the value of the learning rate of the controller as well as the initial values. The use of the metaheuristic algorithms provides an easy and efficient tool for the design stage. The indirect method of adaptation (for the  $K_2$  parameter) improves the dynamics of the drive and its work under changes of the plant parameters. Another law of adaptation implemented for this coefficient is needed due to this parameter's

dependency on the fast changing torsional torque (the remaining gains are based on the angular velocities). The simulation results show high efficiency in damping the oscillations caused by the torsional torque in the electrical drive with an elastic joint. This confirms the possibility of the correct application of the presented structure. The initial experimental results were obtained using the dSPACE card to confirm the correct operation of the proposed control structure. Comparison of the efficiency achieved with the presented adaptive structure and basic controllers such as a PI controller or a state space controller indicates a significant improvement. Not only are the oscillations damped, but also the overshoots are minimized, resulting in a reduction in the cost function value of about 40%. Further research will be focused on the implementation of the algorithms on a low-cost device and the stability analysis.

**Author Contributions:** Conceptualization, M.K.; methodology, M.M., M.Z., R.S. and M.K.; software, M.M., M.Z. and R.S.; validation, M.M., M.Z., R.S. and M.K.; formal analysis, M.M., M.Z., R.S. and M.K.; investigation, M.M., M.Z., R.S. and M.K.; resources, M.K.; data curation, M.M., M.Z. and R.S.; writing—original draft preparation, M.M., M.Z., R.S. and M.K.; writing—review and editing, M.M., M.Z., R.S. and M.K.; visualization, M.M. and M.Z.; supervision, M.K.; project administration, M.K.; funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Dhaouadi, R.; Kubo K.; Tobise, M. Two-Degree-of-Freedom Robust Speed Controller for High Performance Rolling Mill Drives. In Proceedings of the Conference Record of the 1992 IEEE Industry Applications Society Annual Meeting, Houston, TX, USA, 4–9 October 1992. [\[CrossRef\]](#)
2. Brock, S.; Łuczak, D.; Nowopolski, K.; Pajchrowski, T.; Zawirski, K. Two Approaches to Speed Control for Multi-Mass System With Variable Mechanical Parameters. *IEEE Trans. Ind. Electron.* **2017**, *64*, 3338–3347. [\[CrossRef\]](#)
3. Brock, S.; Łuczak, D.; Pajchrowski, T.; Zawirski, K. Selected Methods for a Robust Control of Direct Drive with a Multi-mass Mechanical Load. In *Advanced Control of Electrical Drives and Power Electronic Converters. Studies in Systems, Decision and Control*, 75; Kabziński, J., Ed.; Springer: Cham, Switzerland, 2017; pp. 75–98. [\[CrossRef\]](#)
4. Lukyanov, X.I.; Pishnograev, R.S.; Schvidchenko, N.V. Wide Strip Hot Rolling Mill Runout Roll Table Electric Drive Control System. In Proceedings of the IX International Conference on Power Drives Systems (ICPDS), Perm, Russia, 3–7 October 2016; pp. 1–5. [\[CrossRef\]](#)
5. Neshati, M.; Jersch, T.; Wenske, J. Model Based Active Damping of Drive Train Torsional Oscillations for a Full-Scale Wind Turbine Nacelle Test Rig. In Proceedings of the American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 2283–2288. [\[CrossRef\]](#)
6. Hillsley, K.L.; Yurkovich, S. Vibration Control of a Two-Link Flexible Robot Arm. Proceedings. In Proceedings of the IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; pp. 2121–2126. [\[CrossRef\]](#)
7. Cheng, X.; Zhang, Y.; Liu, H.; Wollherr, D.; Buss, M. Adaptive Neural Backstepping Control for Flexible-Joint Robot Manipulator with Bounded Torque Inputs. *Neurocomputing* **2021**, *458*, 70–86. [\[CrossRef\]](#)
8. Jung, H.; Jeon, K.; Oh, S. Iterative Feedback Tuning of Cascade Control for Position and Velocity of Two-Mass System. *IFAC-PapersOnLine* **2020**, *53*, 8357–8362. [\[CrossRef\]](#)
9. Zhang, R.; Yang, Y.; Chen, Z.; Tong, C. Torsional Vibration Suppression Control in the Main Drive System of Rolling Mill by State Feedback Speed Controller Based on Extended State Observer. In Proceedings of the IEEE International Conference on Control and Automation, Guangzhou, China, 30 May–1 June 2007; pp. 2172–2177. [\[CrossRef\]](#)
10. Lin, M.-H.; Zhang, L.; Zhang, Y.-H.; Li, P.-F.; Wang, X.-H. Command Filter Backstepping Control with Error Compensation for Flexible-Joint Manipulator. In Proceedings of the Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 3696–3701. [\[CrossRef\]](#)
11. Yakub, M.F.H.; Martono, W.; Akmeliawati, R. Vibration control of two-mass rotary system using improved NCTF controller for positioning systems. In Proceedings of the IEEE Control and System Graduate Research Colloquium (ICSGRC 2010), Shah Alam, Malaysia, 22 June 2010; pp. 61–67. [\[CrossRef\]](#)

12. Szabat, K.; Orłowska-Kowalska, T. Vibration Suppression in Two-Mass Drive System Using PI Speed Controller and Additional Feedbacks-Comparative Study. *IEEE Trans. Ind. Electron.* **2007**, *54*, 1193–1206. [[CrossRef](#)]
13. Park, T.; Shin, E.; Oh, W.; Yoo, J. Robust Speed Control for Torsional Vibration Suppression of Rolling Mill Drive System. In Proceedings of the IECON'03, 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No.03CH37468), Roanoke, VA, USA, 2–6 November 2003; Volume 1, pp. 66–71. [[CrossRef](#)]
14. Song, S.; Ji, J.; Sul, S.; Park, M. Torsional Vibration Suppression Control in 2-Mass System by State Feedback Speed Controller. In Proceedings of the IEEE International Conference on Control and Applications, Vancouver, BC, Canada, 13–16 September 1993; Volume 1, pp. 129–134. [[CrossRef](#)]
15. Hara, K.; Hashimoto, S.; Funato, H.; Kamiyama, K. Robust comparison between state feedback-based speed control systems with and without state observers in resonant motor drives. In Proceedings of the Second International Conference on Power Electronics and Drive Systems, Singapore, 26–29 May 1997; Volume 1, pp. 371–376. [[CrossRef](#)]
16. Tang, N.; Yang, M.; Hu, K.; Xu, D. Research on State Feedback of Two-Mass System Based on Weight Coefficient. In Proceedings of the 2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific), Harbin, China, 7–10 August 2017; pp. 1–6. [[CrossRef](#)]
17. Radke, A.; Gao, Z. A Survey of State and Disturbance Observers for Practitioners. In Proceedings of the American Control Conference, Minneapolis, MN, USA, 14–16 June 2006; pp. 5183–5188. [[CrossRef](#)]
18. Sonawane, R.; Apte, A.A. Extended State Observer Based Speed Control Scheme for PMSM Drives. In Proceedings of the International Conference On Advances in Communication and Computing Technology (ICACCT), Sangamner, India, 8–9 February 2018; pp. 524–527. [[CrossRef](#)]
19. Lagrioui, A.; Mahmoudi, H. Speed and Current Control for the PMSM Using a Luenberger Observer. In Proceedings of the 2011 International Conference on Multimedia Computing and Systems, Ouarzazate, Morocco, 7–9 April 2011; pp. 1–6. [[CrossRef](#)]
20. Li, Y.; Yang, M.; Long, J.; Liu, Z.; Xu, D. Current Sensorless Predictive Control Based on Extended Kalman Filter for PMSM Drives. In Proceedings of the 2017 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific), Harbin, China, 7–10 August 2017; pp. 1–6. [[CrossRef](#)]
21. Wei, M.; Liu, T. Design and Implementation of an Online Tuning Adaptive Controller for Synchronous Reluctance Motor Drives. *IEEE Trans. Ind. Electron.* **2013**, *60*, 3644–3657. [[CrossRef](#)]
22. Chiang, H.; Hsu, K.; Li, I. Optimized Adaptive Motion Control Through an SoPC Implementation for Linear Induction Motor Drives. *IEEE/ASME Trans. Mechatronics* **2015**, *20*, 348–360. [[CrossRef](#)]
23. Belov, M.P.; Lanh, N.V.; Khoa, T.D. State Observer based Elman Recurrent Neural Network for Electric Drive of Optical-Mechanical Complexes. In Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), Moscow, Russia, 26–29 January 2021; pp. 802–805. [[CrossRef](#)]
24. Lukichev, D.V.; Demidova, G.L.; Brock, S. Fuzzy adaptive PID control for two-mass servo-drive system with elasticity and friction. In Proceedings of the 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), Gdynia, Poland, 24–26 June 2015; pp. 443–448. [[CrossRef](#)]
25. Orłowska-Kowalska, T.; Szabat, K. Damping of Torsional Vibrations in Two-Mass System Using Adaptive Sliding Neuro-Fuzzy Approach, *IEEE Trans. Ind. Inform.* **2008**, *4*, 47–57. [[CrossRef](#)]
26. Song, G.; Tao, G. A Partial-State Feedback Model Reference Adaptive Control Scheme. *IEEE Trans. Autom. Control* **2020**, *65*, 44–57. [[CrossRef](#)]
27. Al-Waeli K.H.; Ramli R.; Haris, S.M.; Zulkoffli, ZBAmiri, M.S. Offline ANN-PID Controller Tuning on a Multi-Joints Lower Limb Exoskeleton for Gait Rehabilitation. *IEEE Access* **2021**, *9*, 107360–107374. [[CrossRef](#)]
28. Lin, F.-J.; Wai, R.-J.; Duan, R.-Y. Neural-network controller for parallel-resonant ultrasonic motor drive. *IEEE Trans. Control Syst. Technol.* **1999**, *7*, 494–501. [[CrossRef](#)]
29. Kodakkal, A.; Veramalla, R.; Kuthuri, N.R.; Salkuti, S.R. An ALO Optimized Adaline Based Controller for an Isolated Wind Power Harnessing Unit. *Designs* **2021**, *5*, 65. [[CrossRef](#)]
30. Uçak, K. A Runge-Kutta MLP Neural Network Based Control Method for Nonlinear MIMO Systems. In Proceedings of the 2019 6th International Conference on Electrical and Electronics Engineering (ICEEE), Istanbul, Turkey, 16–17 April 2019; pp. 186–192. [[CrossRef](#)]
31. Kaminski, M.; Szabat, K. Adaptive Control Structure with Neural Data Processing Applied for Electrical Drive with Elastic Shaft. *Energies* **2021**, *14*, 3389. [[CrossRef](#)]
32. Lin, F.-J.; Chen, S.-Y.; Teng, L.-T.; Chu, H. Recurrent Functional-Link-Based Fuzzy Neural Network Controller With Improved Particle Swarm Optimization for a Linear Synchronous Motor Drive. *IEEE Trans. Magn.* **2009**, *45*, 3151–3165. [[CrossRef](#)]
33. Wang, J.; Liu, Y.; Cao, G.; Zhao, Y.; Zhang, J. Design of RBF Adaptive Sliding Mode Controller for A Supercavitating Vehicle. *IEEE Access* **2021**, *9*, 39873–39883. [[CrossRef](#)]
34. Chen, S.-G.; Lin, F.-J.; Liang, C.-H.; Liao, C.-H. Intelligent Maximum Power Factor Searching Control Using Recurrent Chebyshev Fuzzy Neural Network Current Angle Controller for SynRM Drive System. *IEEE Trans. Power Electron.* **2021**, *36*, 3496–3511. [[CrossRef](#)]
35. Pajchrowski, T.; Zawirski, K. Application of fuzzy logic techniques to robust speed control of PMSM. In Proceedings of the 2008 13th International Power Electronics and Motion Control Conference, Poznan, Poland, 1–3 September 2008; pp. 1198–1203. [[CrossRef](#)]

36. Dyanamina, G.; Kakodia, S.K. Adaptive neuro fuzzy inference system based decoupled control for neutral point clamped multi level inverter fed induction motor drive. *Chin. J. Electr. Eng.* **2021**, *7*, 70–82. [[CrossRef](#)]
37. Lin, C.; Li, H. TSK Fuzzy CMAC-Based Robust Adaptive Backstepping Control for Uncertain Nonlinear Systems. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 1147–1154. [[CrossRef](#)]
38. Anderson, R.B.; Marshall, J.A.; L’Afflitto, A. Novel Model Reference Adaptive Control Laws for Improved Transient Dynamics and Guaranteed Saturation Constraints. *J. Frankl. Inst.* **2021**, *358*, 6281–6308. [[CrossRef](#)]
39. Szabat, K. Direct and Indirect Adaptive Control of a Two-Mass Drive System—A Comparison. In Proceedings of the 2008 IEEE International Symposium on Industrial Electronics, Cambridge, UK, 30 June–2 July 2008; pp. 564–569. [[CrossRef](#)]
40. Etmianiesfahani, A.; Gu, H.; Salehipour, A. ABFIA: A hybrid algorithm based on artificial bee colony and Fibonacci indicator algorithm. *J. Comput. Sci.* **2022**, *61*, 101651. [[CrossRef](#)]
41. Tarczewski, T.; Grzesiak, L.M. Artificial bee colony based auto-tuning of PMSM state feedback speed controller. In Proceedings of the 2016 IEEE International Power Electronics and Motion Control Conference, Varna, Bulgaria, 5–28 September 2016; pp. 1155–1160. [[CrossRef](#)]
42. Rathod, N.; Wankhade, S. Optimizing neural network based on cuckoo search and invasive weed optimization using extreme learning machine approach. *Neurosci. Inform.* **2022**, *2*, 100075. [[CrossRef](#)]
43. Knypiński, Ł.; Kuroczycki, S.; Márquez, F.P.G. Minimization of Torque Ripple in the Brushless DC Motor Using Constrained Cuckoo Search Algorithm. *Electronics* **2021**, *10*, 2299. [[CrossRef](#)]
44. Zawirski, K.; Nowopolski, K.; Siwek, P. Application of Cuckoo Search Algorithm for Speed Control Optimization in Two-Sided Electrical Drive. In Proceedings of the 2018 IEEE 18th International Power Electronics and Motion Control Conference (PEMC), Budapest, Hungary, 26–30 August 2018; pp. 651–656. [[CrossRef](#)]
45. Banaie-Dezfouli, M.; Nadimi-Shahraki, M.H.; Beheshti, Z. R-GWO: Representative-based grey wolf optimizer for solving engineering problems. *Appl. Soft Comput.* **2021**, *106*, 1–28. [[CrossRef](#)]
46. Mishra, A.K.; DasSR, Ray, P.K.; Mallick, R.K.; Mohanty, A.; Mishra, D.K. PSO-GWO Optimized Fractional Order PID Based Hybrid Shunt Active Power Filter for Power Quality Improvements. *IEEE Access* **2020**, *8*, 74497–74512. [[CrossRef](#)]
47. Du, B.; Huang, S.; Guo, J.; Tang, H.; Wang, L.; Zhou, S. Interval forecasting for urban water demand using PSO optimized KDE distribution and LSTM neural networks. *Appl. Soft Comput.* **2022**, *122*, 108875. [[CrossRef](#)]
48. Lin, X.; Li, K.; Wang, L. A driving-style-oriented adaptive control strategy based PSO-fuzzy expert algorithm for a plug-in hybrid electric vehicle. *Expert Syst. Appl.* **2022**, *201*, 117236. [[CrossRef](#)]
49. Ezugwu, A.E.; Prayogo, D. Symbiotic organisms search algorithm: Theory, recent advances and applications. *Expert Syst. Appl.* **2019**, *119*, 184–209. [[CrossRef](#)]
50. Zhou, Y.; Miao, F.; Luo, Q. Symbiotic organisms search algorithm for optimal evolutionary controller tuning of fractional fuzzy controllers. *Appl. Soft Comput.* **2019**, *77*, 497–508. [[CrossRef](#)]
51. He, H.; Li, J.; Zhao, W.; Li, B.; Li, Y. Reactive power and voltage optimization of new-energy grid based on the improved flower pollination algorithm. *Energies* **2022**, *15*, 3653. [[CrossRef](#)]
52. Tarczewski, T.; Grzesiak, L.M. An application of novel nature-inspired optimization algorithms to auto-tuning state feedback speed controller for PMSM. *IEEE Trans. Ind. Appl.* **2018**, *54*, 2913–2925. [[CrossRef](#)]
53. Serkies, P.; Gorla, A. Implementation of PI and MPC-Based Speed Controllers for a Drive with Elastic Coupling on a PLC Controller. *Electronics* **2021**, *10*, 3139. [[CrossRef](#)]
54. Łuczak, D.; Wójcik, A. The study of neural estimator structure influence on the estimation quality of selected state variables of the complex mechanical part of electrical drive. In Proceedings of the 2017 19th European Conference on Power Electronics and Applications (EPE’17 ECCE Europe), Warsaw, Poland, 11–14 September 2017; pp. P.1–P.10. [[CrossRef](#)]
55. Widrow, B.; Lehr, M.A. Perceptrons, Adalines, and Backpropagation. In *Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1995; pp. 719–724.
56. Kaminski, M., Application of the BAT algorithm in optimization of adaptive state space controller used for two-mass system. *Przegląd Elektrotechniczny* **2017**, *93*, 300–304. [[CrossRef](#)]
57. Cheng, M.-Y.; Prayogo, D. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. *Comput. Struct.* **2014**, *139*, 98–112 [[CrossRef](#)]
58. Zainal, N.A.; Azad, S.; Zamli, K.Z. An Adaptive Fuzzy Symbiotic Organisms Search Algorithm and Its Applications. *IEEE Access* **2020**, *8*, 225384–225406. [[CrossRef](#)]
59. Acharya, D.S.; Mishra, S.K. A multi-agent based symbiotic organisms search algorithm for tuning fractional order PID controller. *Measurement* **2020**, *155*, 107559. [[CrossRef](#)]
60. Singh, S.P.; Prakash, T.; Singh, V.P. Coordinated tuning of controller-parameters using symbiotic organisms search algorithm for frequency regulation of multi-area wind integrated power system. *Eng. Sci. Technol. Int. J.* **2020**, *23*, 240–252. [[CrossRef](#)]
61. Ezugwu, A.E.; Adeleke, O.J. Viriri, S. Symbiotic organisms search algorithm for the unrelated parallel machines scheduling with sequence-dependent setup Times. *PLoS ONE* **2018**, *13*, e0200030. [[CrossRef](#)]
62. Kaminski, M.; Malarczyk, M. Hardware implementation of neural shaft torque estimator using low-cost microcontroller board. In Proceedings of the 2021 25th International Conference on Methods and Models in Automation and Robotics (MMAR), Międzyzdroje, Poland, 23–26 August 2021; Volume 10, pp. 372–377. [[CrossRef](#)]

63. Wang, Z.; Qin, C.; Wan, B.; Song, W.W. A comparative study of common nature-inspired algorithms for continuous function optimization. *Entropy* **2021**, *23*, 874. [[CrossRef](#)]
64. Yang, X.-S. Flower pollination algorithm for global optimization. *arXiv* **2012**, arXiv:1312.5673.
65. Bányai, T.; Illés, B.; Gubán, M.; Gubán, Á.; Schenk, F.; Bányai, Á. Optimization of just-in-sequence supply: A flower pollination algorithm-based approach. *Sustainability* **2019**, *11*, 3850. [[CrossRef](#)]
66. Dubey, H.M.; P; it, M.; Panigrahi, B.K. A biologically inspired modified flower pollination algorithm for solving economic dispatch problems in modern power systems. *Cogn. Comput.* **2015**, *7*, 594–608. [[CrossRef](#)]
67. Bekdaş, G.; Nigdeli, S.M.; Yang, X.-S. Sizing optimization of truss structures using flower pollination algorithm. *Appl. Soft Comput.* **2015**, *37*, 322–331. [[CrossRef](#)]
68. Lalljith, S.; Fleming, P.; Pillay, U.; Naicker, K.; Naidoo, Z.J.; Saha, A.K. Applications of flower pollination algorithm in electrical power systems: A review. *IEEE Access* **2022**, *10*, 8924–8947. [[CrossRef](#)]
69. Vallaey, V.; Tyson, R.C.; Lane, W.D.; Deleersnijder, E.; Hanert, E. A Lévy-flight diffusion model to predict transgenic pollen dispersal. *J. R. Soc. Interface* **2017**, *14*, 126. [[CrossRef](#)] [[PubMed](#)]
70. Kabzinski, J.; Mosiolek, P. Integrated, multi-approach, adaptive control of two-mass drive with nonlinear damping and stiffness. *Energies* **2021**, *14*, 5475. [[CrossRef](#)]