algorithms

MDPI

Article

Identification of Mechanical Parameters in Flexible Drive Systems Using Hybrid Particle Swarm Optimization Based on the Quasi-Newton Method

Ishaq Hafez [1,*,†] and Rached Dhaouadi [2,†]

[1] Mechatronics Graduate Program, College of Engineering, American University of Sharjah, Sharjah 26666, United Arab Emirates

[2] Department of Electrical Engineering, College of Engineering, American University of Sharjah, Sharjah 26666, United Arab Emirates; rdhaouadi@aus.edu

* Correspondence: b00084765@aus.edu

† These authors contributed equally to this work.

Abstract: This study presents hybrid particle swarm optimization with quasi-Newton (HPSO-QN), a hybrid optimization method for accurately identifying mechanical parameters in two-mass model (2MM) systems. These systems are commonly used to model and control high-performance electric drive systems with elastic joints, which are prevalent in modern industrial production. The proposed method combines the global exploration capabilities of particle swarm optimization (PSO) with the local exploitation abilities of the quasi-Newton (QN) method to precisely estimate the motor and load inertias, shaft stiffness, and friction coefficients of the 2MM system. By integrating these two optimization techniques, the HPSO-QN method exhibits superior accuracy and performance compared to standard PSO algorithms. Experimental validation using a 2MM system demonstrates the effectiveness of the proposed method in accurately identifying and improving the mechanical parameters of these complex systems. The HPSO-QN method offers significant implications for enhancing the modeling, performance, and stability of 2MM systems and can be extended to other systems with flexible shafts and couplings. This study contributes to the development of accurate and effective parameter identification methods for complex systems, emphasizing the crucial role of precise parameter estimation in achieving optimal control performance and stability.

Keywords: parameter identification; two-mass model; electric drive systems; particle swarm optimization; quasi-Newton method; hybrid optimization; stochastic algorithms; mechanical parameters; optimal control performance; elastic joints

check for updates

Citation: Hafez, I.; Dhaouadi, R. Identification of Mechanical Parameters in Flexible Drive Systems Using Hybrid Particle Swarm Optimization Based on the Quasi-Newton Method. *Algorithms* **2023**, *16*, 371. https://doi.org/10.3390/a16080371

Academic Editor: Frank Werner

Received: 30 June 2023
Revised: 23 July 2023
Accepted: 27 July 2023
Published: 31 July 2023

1. Introduction

Drive systems play a crucial role in modern industries, such as steel mills, wind turbines, and industrial robots. Many of these systems can be represented as a model, where the motor and load inertia are connected by a shaft. In such applications, low mechanical resonance can occur due to the flexible shaft between the actuator and the load inertia. This can cause undesirable problems in the mechanical coupling of the system and decrease its performance [1–4]. To address these issues, the dynamics of these flexible systems must be modeled as a two-mass or multi-mass system when designing the control laws. However, incorrect calculations and assumptions of the parameters in these models can be problematic when relying solely on the information provided in the manufacturer's datasheets [5–7]. Without knowledge of the inertia and shaft stiffness, it is difficult to effectively suppress mechanical resonance. Identifying these parameters can help suppress mechanical resonance, improve system response, and reduce tracking errors. Additionally, these mechanical parameters may not be constant over time in some practical applications.

A mechanical system with inertia disks distributed on the actuating motor and its load side, connected by a flexible shaft between them, can be considered a 2MM system. A typical 2MM drive system exhibits resonant and anti-resonant frequencies when the system's frequency response is measured from the applied electromagnetic torque to the measured speeds of the actuating motor. Over the years, numerous methods and approaches have been developed and applied to extract the modal parameters of the system. The 2MM model has been used to describe the mechanics of various systems, including rolling mills [8], wind turbines [9], industrial robots [10], paper machines [11], and elevators [12]. Furthermore, the 2MM model has been employed in designing control laws to mitigate harmful resonances in mechanical systems and improve dynamic accuracy.

Recent years have seen a rise in the application of artificial intelligence (AI), machine learning (ML), and deep learning (DL) within the engineering and industrial electronics sectors [13–15]. These methodologies are now commonly used for complex tasks such as system identification, control, optimization, and fault diagnostics. For example, in power electronics, AI is utilized in system design, control, and maintenance [13]. Deep learning, a subset of ML, is making strides in fault diagnostics in electrical applications, contributing to improved system reliability [14]. Additionally, deep learning methods, such as sparse autoencoders, have been successfully employed in improving the reconstruction of fingerprint images, further expanding the scope of AI applications [16]. In fact, AI and ML technologies hold promising prospects, but their wider adoption in industry and society faces several challenges from both internal and external perspectives [15]. However, solutions are being sought, and these techniques are being incorporated into traditional engineering methods, such as model-based multivariable control systems [17]. In drive systems, AI and ML offer advanced techniques for parameter identification and system modeling, with applications like the use of dynamic neural networks for estimating nonlinear friction [18]. The intersection of AI, ML, DL, and engineering indicates a new era in industrial electronics, characterized by increased efficiency, robustness, and performance.

Parameter identification methods in the literature can be roughly divided into parametric and non-parametric methods. Parametric methods involve estimating the parameters of the 2MM system's transfer function polynomials in the time domain. On the other hand, non-parametric methods utilize frequency response function (FRF) methods to estimate the resonant frequencies, i.e., the modal parameters of the system. Many approaches have been successfully applied for the parameter identification of 2MM systems. In [19], the authors proposed an identification method that involves solving a weighted least squares problem for an over-determined linear system derived from sampling the dynamic model along a closed-loop tracking trajectory. An experimental study was conducted on a mechanical system to validate the effectiveness of the proposed method. In [20], the authors proposed a non-linear modeling approach based on the Hammerstein structure and used recursive least squares to identify the mechanical parameters of the system, including Coulomb friction and dead zones. They found that the non-linear approach performed better than the linear approach for low-speed operation, where the non-linearities in the system had a significant impact, resulting in a reduction in the identification error.

The following examples highlight some more identification methods, emphasizing the similarities and differences amongst them. In [21], the authors studied closed-loop identification procedures for multi-mass systems by comparing different identification models, sampling periods, and model orders. Simulation and experimental results showed that reducing the sampling time can limit high-frequency noise, improve the ratio of useful signal to noise, and accurately localize the crucial points in the spectrum by limiting the sampling frequency. In [4], Ke et al. proposed an identification method that combines a Luenberger observer and the variable forgetting factor recursive least squares method to accurately estimate the mechanical parameters of a 2MM system. This method was compared with the standard forgetting factor recursive least squares method and found to achieve good results with high identification accuracy in a short amount of time. In a study by [22], a disturbance observer (DOB) method was proposed to identify the mechanical parameters of series elastic actuators (SEAs) modeled as 2MM systems. The identification process was performed with two closed-loop motor position-tracking experiments, one for identifying the motor-side parameters and spring stiffness, and one for the load-side parameters. Experimental results were provided to validate the proposed method, but the authors noted that further investigation is needed due to non-negligible errors in the Coulomb friction coefficient caused by high nonlinear frictions affecting the device. In [23], Villwock and Pacas developed an identification method in the frequency domain that combines the Welch method with the Levenberg–Marquardt algorithm to identify the parameters of 2MM and three-mass (3MM) systems. The results of this method were found to be effective for the parameter identification of 2MM and 3MM systems, as well as more complex mechanical systems. In a study by Dhaouadi and Kubo [7], a technique was presented that involves the repeated integration of data and the use of the recursive least squares method to identify the mechanical parameters of a 2MM system. Numerical results demonstrated that the estimated parameters converged to the actual transfer function parameters with high accuracy.

An intriguing method was introduced by Nowopolski and Wicher [24], where they used the standard particle swarm optimization (PSO) method to determine the mechanical parameters of a 2MM system with backlash. The PSO is an optimization technique inspired by the social behavior of birds flocking or fish schooling. It is a population-based stochastic optimization algorithm designed to find the global optimum in a problem space. In the context of parameter identification, the PSO algorithm is used to optimize a cost function to determine the best system parameters. The authors compared the results obtained using two different cost functions, and the experimental setup was excited using three different excitation signals for the identification process. However, the cost functions used resulted in relatively large errors, and the experimental results did not match the simulated responses perfectly.

There are various hybrid techniques that incorporate PSO with other optimization algorithms, with the aim of improving global search ability and convergence speed. This has been demonstrated in hybrids that involve PSO with genetic algorithms (GAs), ant colony optimization, and differential evolution. Each of these hybrids showed superior performance compared to the standard PSO, finding wide applications in fields such as engineering, finance, and image processing [25,26]. Specific strategies have been adopted to enhance the social and cooperative aspects of PSO. For instance, the integration of GA with PSO has been implemented in a few distinct ways. One approach involves using GA and PSO either sequentially or in parallel, while another strategy involves the exchange of the fittest particles between GA and PSO when run concurrently [27,28]. Some studies have even employed a two-phase mechanism, where PSO accelerates evolution, and GA ensures diversity [29]. Several innovative hybrid approaches have also been proposed. These include a fuzzy approach that utilizes simple rules for decision making [30], a hybrid algorithm known as FAPSO that combines PSO with the firefly algorithm [31], and a heuristic based on PSO and simulated annealing for resolving multi-objective problems in network-based models [32]. A unique hybrid PSO method focuses specifically on estimating the parameters of photovoltaic (PV) cells. This comes in response to the escalating demand for renewable energy and the need for accurate PV cell models [33]. In a similar vein, an adaptive simulated annealing-parallel particle swarm optimization (ASA-PPSO) approach was developed, incorporating PSO with an infix condition that applies simulated annealing (SA). Moreover, hybridizing differential evolution (DE) with PSO has been explored in various ways, one of which involves using DE to maneuver particles and thereby enhance the convergence rate of PSO [26,34].

Hybrid optimization techniques that combine the strengths of gradient-based and evolutionary algorithms have demonstrated excellent performance in finding satisfactory solutions. One such method is the PSO-descent hybrid, developed by Coelho and Mariani [35], which combines the global search ability of particle swarm optimization (PSO) with the local search capability of the quasi-Newton method. The PSO algorithm runs until convergence, and then the quasi-Newton method is applied to the best solution found. This hybrid approach has proven highly effective in solving the economic dispatch problem. Another hybrid approach, GRPSO, developed by Wang et al. [36], merges the quasi-Newton method and PSO to leverage the strengths of both algorithms. The hybrid begins by running the quasi-Newton method from a random starting point until a local minimum is achieved, then initiates a population of random particles, and runs the PSO algorithm until the global best particle surpasses the quasi-Newton method result. The PSO algorithm is then stopped, and the global best particle is used to restart the quasi-Newton method. This hybrid approach benefits from the local search capability of the quasi-Newton method and the global search ability of the PSO algorithm, enabling it to escape local minima and explore different regions of the search space. Results show that GRPSO is more efficient and reliable than other methods compared to it [36].

While global search methods, such as PSO, are effective at finding the global optimum in large search spaces, they can be less efficient when applied to local search. Gradient-based methods, on the other hand, can be faster and more efficient in finding the optimal solution in a local search space. However, gradient-based methods can struggle when there are multiple local minima or the function being optimized is non-convex. As a result, finding the global optimum using only gradient-based methods can be challenging. To address these issues, a hybrid approach that combines the strengths of both global and local search methods is often employed. In this study, a hybrid approach combining PSO and quasi-Newton was chosen because of the PSO algorithm's ability to explore a large search space and quasi-Newton's efficiency in finding the optimal solution in a local search space. Additionally, quasi-Newton methods are capable of handling non-linear optimization problems and can converge to the global minimum, even when the function being optimized is non-convex. Furthermore, quasi-Newton methods do not require the computation of second-order partial derivatives, making them computationally more efficient than traditional Newton methods. This is particularly important when dealing with optimization problems involving large-scale systems.

Therefore, this paper presents a novel hybrid optimization method called HPSO-QN for identifying the mechanical parameters of a 2MM system. The method combines PSO for global exploration with quasi-Newton for local exploitation and estimates the mechanical parameters, such as motor and load inertias, shaft stiffness, and viscous and Coulomb friction coefficients on both the motor and load sides of the system. This work represents a significant step forward from previous studies [2,3] that used various variants of PSO for the parameter identification of a 2MM simulation. The most effective variant of PSO was carefully selected and incorporated into the hybrid framework. We then explored different versions of local search with the quasi-Newton method to reach the globally optimal solution. The proposed method was validated using experimental results from a 2MM system, and its estimated results were compared with those identified using the frequency response function (FRF) method presented in [1].

The remainder of this paper is organized as follows. Section 2 presents the mathematical model of the 2MM system. Section 3 presents the proposed identification method. Section 4 describes the detailed experimental setup of the study. Section 5 focuses on the analysis and discussion of the results obtained. Finally, conclusions are drawn in Section 6.

2. System Modeling

The 2MM system represents an actuating motor coupled to a mechanical system with two lumped inertias connected by a low-stiffness shaft and couplings [7,18,37–40]. The block diagram in Figure 1 shows the 2MM system with its equivalent current control loop. In this diagram, i_{ref} is the reference current (A), K_t is the torque constant (N·m/A), J_1 is the equivalent motor-side inertia (kg·m^2), and J_2 is the equivalent load-side inertia (kg·m^2). The motor-side and load-side angular velocities are represented by ω_1 and ω_2, respectively (rad/s). The inertias are coupled through a low-stiffness shaft K_s, which is subjected to torsional torque τ_s (N·m). Here, the Coulomb friction on both sides is taken into account, while external forces are neglected, as the system is considered to move freely. The following describes the mechanical dynamics of the 2MM system.

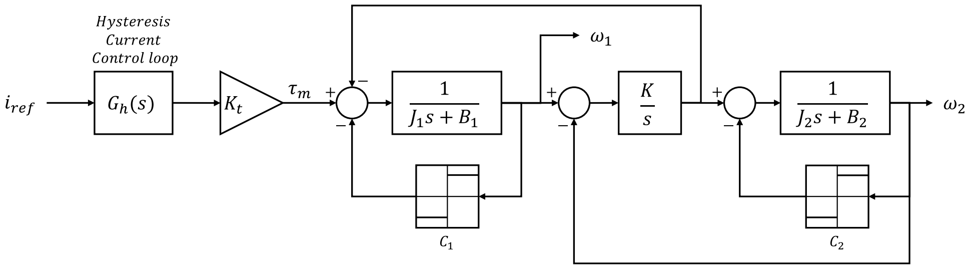


Figure 1. Open-loop control structure of two-mass model (2MM) system.

The mathematical model of the 2MM system is described by the following dynamic equations:

$$J_1 \frac{d\omega_1}{dt} = \tau_m - B_1\omega_1 - \tau_s - C_1 sign(\omega_1) \tag{1}$$

$$J_2 \frac{d\omega_2}{dt} = \tau_s - \tau_l - B_2\omega_2 - C_2 sign(\omega_2) \tag{2}$$

$$\tau_s = K(\theta_1 - \theta_2) \tag{3}$$

To aid in the identification of system parameters using the proposed method, these equations are normalized with respect to the rated speed and torque. Normalizing parameters is crucial in optimization methods, ensuring faster convergence, preventing numerical instability, and enabling the consistent comparison of system performance. It eliminates scale differences, improves interpretability, robustness, and result consistency. Additionally, normalization facilitates uniform distribution in the search space, reducing the likelihood of encountering local optima and enhancing optimization efficiency and robustness [41,42]. The normalized parameters for the motor and load side inertias are calculated as $J_{1n} = J_1(\omega_{rated}/\tau_{rated})$ and $J_{2n} = J_2(\omega_{rated}/\tau_{rated})$, respectively. The normalized shaft stiffness is calculated as $K_n = K_n/\omega_{rated}$. The normalized Coulomb and viscous friction coefficients for both the motor and load sides are calculated as $C_{1n} = C_1/\tau_{rated}$, $C_{2n} = C_2/\tau_{rated}$, $B_{1n} = B_1(\omega_{rated}/\tau_{rated})$, and $B_{2n} = B_2(\omega_{rated}/\tau_{rated})$. The motor and load speeds are normalized using $\omega_{1n} = \omega_1/\omega_{rated}$ and $\omega_{2n} = \omega_2/\omega_{rated}$. The normalized equations are denoted with a subscript 'n' to indicate that they are normalized with respect to the rated speed and torque of the actuator. This includes variables and parameters, such as the normalized angular velocities (Ω_1, Ω_2), normalized angular positions (Θ_1, Θ_2), normalized inertias (J_{1n}, J_{2n}), and normalized viscous damping and Coulomb friction coefficients (B_{1n}, B_{2n}, C_{1n}, C_{2n}). Additionally, the normalized motor and load torques are represented by τ_{mn} and τ_{ln}, respectively, and the normalized shaft torsional torque is represented by τ_{sn}.

To analyze the system behavior, the transfer functions of the 2MM system can be obtained by linearizing the system equations and assuming that the Coulomb friction is small and can be neglected. The transfer functions of the system are given by

$$G_1(s) = \frac{\Omega_1(s)}{\tau_{mn}(s)} = \frac{1}{J_{2n}s}\left[\frac{\frac{B_{2n}}{J_{1n}}s + \frac{K_n}{J_{1n}}}{s^2 + \frac{B_{1n}+B_{2n}}{J_{1n}J_{2n}}s + \frac{K_n}{J_{1n}J_{2n}}}\right] \tag{4}$$

$$G_2(s) = \frac{\Omega_2(s)}{\tau_{mn}(s)} = \frac{1}{J_{1n}s}\left[\frac{\frac{B_{1n}}{J_{2n}}s + \frac{K_n}{J_{2n}}}{s^2 + \frac{B_{1n}+B_{2n}}{J_{1n}J_{2n}}s + \frac{K_n}{J_{1n}J_{2n}}}\right] \tag{5}$$

The parameters to be identified in the 2MM system include the normalized motor and load inertias (J_{1n}, J_{2n}), the normalized torsional stiffness (K_n), the normalized Coulomb friction coefficients (C_{1n}, C_{2n}) and the normalized viscous damping coefficients (B_{1n}, B_{2n}). These parameters can be represented in vector notation as given below:

$$\boldsymbol{\theta} = [J_{1n},\ J_{2n},\ K_n,\ C_{1n},\ C_{2n},\ B_{1n},\ B_{2n}] \tag{6}$$

The goal of parameter identification is to estimate the values of these parameters in order to accurately model the 2MM system.

3. Proposed Identification Methods

This section presents our proposed parameter identification process for the 2MM system, which involves the use of swept-sine signal excitation and error minimization through a cost function. Each step of the process is described in the following subsections.

3.1. Excitation Signal

The 2MM system is excited using a swept-sine signal, also known as a periodic chirp, with a duration T_f. The signal has a starting frequency f_1, a final frequency f_2, and an amplitude A. The equation for the swept sine signal is given by

$$x(t) = A \sin\left(2\pi\left(\frac{1}{2}(f_2 - f_1)\frac{t^2}{T_f} + f_1 t\right)\right) \tag{7}$$

where t ranges from T_0 to T_f.

3.2. Cost Function

The objective of the parameter identification process is to minimize the mean square error between the actual and estimated angular velocities of the 2MM system. This is achieved through a cost function ($Fcost$) defined as follows:

$$Fcost = \frac{1}{N}\sum_{i=1}^{N}\left((\Omega_1 - \hat{\Omega}_1)^2 + (\Omega_2 - \hat{\Omega}_2)^2\right) \tag{8}$$

where $\hat{\Omega}_1$ and $\hat{\Omega}_2$ represent the estimated normalized motor and load angular velocities, respectively. N denotes the number of data samples used for parameter identification.

3.3. Particle Swarm Optimization

Particle swarm optimization (PSO) is a meta-heuristic optimization algorithm that was developed by Kennedy and Eberhart in 1995 [43]. It is a population-based algorithm that simulates the social behavior observed in swarms of biological individuals, such as flocking birds and insects searching for food sources [44]. The PSO algorithm is easy to implement and has been applied to a wide range of scientific and engineering problems [44,45].

In the PSO algorithm, a swarm of particles (N_{PSO}) is initialized randomly within a search space defined by upper (x_{max}) and lower (x_{min}) bound limits. Each particle is evaluated against a pre-defined cost function ($Fcost$) to search for an optimal solution. The particles update their velocity vectors, which control their movement in the next iteration [43–46]. The velocity updates are influenced by two key factors: personal best ($pBest$) and global best ($gBest$). The $pBest$ represents the best solution found by an individual particle, indicating its optimal position within the search space. The $gBest$ refers to the best solution discovered by any particle in the entire swarm, representing the overall optimal position identified by the collective behavior of the swarm. The velocity and position of each particle are updated as follows [45,46]:

$$v_{k+1}(i) = v_k(i) + c_1 r_1 (pBest_k(i) - x_k(i)) + c_2 r_2 (gBest_k - x_k(i)) \tag{9}$$

$$x_{k+1}(i) = x_k(i) + v_{k+1}(i) \tag{10}$$

where i denotes the current particle; c_1 and c_2 are the acceleration coefficients for $pBest$ and $gBest$ terms; and r_1 and r_2 are random numbers generated by uniform distribution in the interval $[0,\ 1]$ [45,46]. The PSO offers several advantages that make it suitable for various optimization problems. It excels at quickly finding solutions, even in high-dimensional search spaces, and is capable of handling complex and noisy optimization problems [45]. Additionally, the PSO has the ability to adapt to changes in the optimization landscape, allowing it to continue searching for better solutions, even when the search space becomes more challenging [47]. However, the PSO also has some limitations. One of the main drawbacks of the PSO is its tendency to get attracted to sub-optimal solutions that are not globally optimal [48]. This can result in premature convergence, where the particles get trapped in a locally optimal solution and continue to search within reduced and limited regions of the search space. Moreover, the performance of the PSO algorithm is sensitive to the selection of control parameters, such as the inertia weight and acceleration constants. Inappropriate parameter choices can lead to poor performance and potential convergence to sub-optimal solutions [49].

To enhance the performance of PSO and address its limitations, various PSO variants have been developed, such as constriction PSO, adaptive inertia weight PSO, and hybrid PSO [50,51]. These variants incorporate additional strategies like dynamic control of the inertia weight, self-adaptive acceleration constants, and hybridization with other optimization methods. Some prior studies have explored variants such as constricted PSO with linearly decreasing inertia weight, adaptive weight factor PSO, and chaos initialized PSO for the identification of 2MM systems [2,3]. Among these, constricted PSO with a linearly decreasing inertia weight showed particularly promising performance, making it our selected base PSO variant for the hybrid framework. For more detailed information on these PSO variants and their application in 2MM system identification, readers are encouraged to refer to [2,3].

3.3.1. Linearly Decreasing Inertia Weight

The inertia weight plays a crucial role in guiding the exploration and exploitation processes in the PSO algorithm. One method for dynamically adjusting the inertia weight is the linearly decreasing time-varying inertia weight [52]. This method iteratively adjusts the inertia weight using Equation (11) as follows:

$$w_k = \frac{w_{max} - w_{min}}{k_{max}}(k_{max} - k) + w_{max} \tag{11}$$

Here, the suggested values for w_{max} and w_{min} are 0.9 and 0.4, respectively. Empirical studies have shown that this method can produce good results when the inertia weight is decreased from a high initial value (e.g., 0.9) to a lower value (e.g., 0.4) as the optimization progresses [52]. However, it is important to note that selecting an unsuitable value for the inertia weight can impede convergence or result in sub-optimal solutions.

3.3.2. Constriction Factor

The constriction factor method, also known as the constriction coefficient method, was introduced by Clerc et al. [53] to improve the convergence and stability of the PSO algorithm. It involves the use of factor $\aleph$, which can be calculated using Equation (12) as follows:

$$\aleph = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \tag{12}$$

where $\varphi = c_1 + c_2$, $\varphi > 4$. The constriction factor is inversely proportional to φ, so when φ is set to 4.1, for example, the constriction factor becomes 0.73. This value is then multiplied by the three terms in Equation (9): the velocity, $pBest$, and $gBest$. Empirical studies have shown that the constriction factor method can improve the performance of PSO compared to the original algorithm. However, it does not offer a mechanism for preventing premature convergence, which can occur when the optimization process reaches a satisfactory solution before fully exploring the search space.

The updated velocity equation incorporating the inertia weight and the constriction factor is as follows:

$$v_{k+1}(i) = \aleph(w_k v_k(i) + c_1 r_1 (pBest_k(i) - x_k(i)) + c_2 r_2 (gBest_k - x_k(i))) \tag{13}$$

In this research, the selected variant of PSO is referred to as the standard PSO (SPSO) for comparison with the hybrid methods. The SPSO will be integrated with the QN method to create a hybrid approach. The pseudo-code of the SPSO algorithm is presented in Algorithm 1. The algorithm will terminate when the maximum number of iterations (k_{max}) is reached.

Algorithm 1 SPSO.

```
Require: $N$, $k_{max}$, $x_{min}$, $x_{max}$, $w_{min}$, $w_{max}$, $\varphi$, $c_1$, and $c_2$
Set $k = 0$, randomly initialize particles' positions $x_i(k)$ and velocities $v_i(k)$
for each particle $P_i(k)$, $i = 1, 2, \ldots, N$ do
    Set $pBest_i(0) = x_i(0)$, $gBest(0) = pBest_1(0)$
end for
repeat
    set $k = k + 1$
    for each particle $P_i(k)$, $i = 1, 2, \ldots, N$ do
        Evaluate $Fcost(x_i(k))$ using Equation (8)
        if $Fcost(x_i(k)) < Fcost(pBest_i(k))$ then
            $pBest_i(k) = x_i(k)$
        end if
        if $Fcost(pBest_i(k)) < Fcost(gBest(k))$ then
            $gBest(k) = pBest_i(k)$
        end if
        Update velocity $v_i(k+1)$ and position $x_i(k+1)$ using Equations (13) and (10)
    end for
until SPSO termination criterion is satisfied
```

3.4. Quasi-Newton Method

The quasi-Newton (QN) method is a powerful gradient-based optimization algorithm that leverages information about the curvature of a cost function to iteratively find its minimum value. It approximates the Hessian matrix at each iteration and performs a cubic line search procedure to enhance the solution obtained. The QN method is particularly useful when the direct calculation of the Hessian matrix is difficult or impractical [54–57]. By incorporating the QN method into the optimization process, the algorithm benefits from improved local optimization capabilities, increasing the likelihood of finding an optimal solution.

At each iteration, the QN method constructs a quadratic model using curvature information. The optimal solution of this quadratic model denoted as x^* can be represented as follows:

$$x^* = -H^{-1} \cdot \nabla f_k \tag{14}$$

where H is a positive definite symmetric matrix (the Hessian), and ∇f_k is the gradient of f at the point x_k [55,56]. To update the Hessian matrix H, the QN method uses the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method, which gives the following formula [55]:

$$H_{k+1} = H_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{H_k s_k s_k^T H_k^T}{s_k^T H_k s_k} \tag{15}$$

where s_k and y_k are calculated as follows:

$$
\begin{aligned}
s_k &= x_{k+1} - x_k \\
y_k &= \nabla f_{k+1} - \nabla f_k
\end{aligned} \tag{16}
$$

Here, s_k represents the change in the independent variable x, and y_k represents the change in the gradient. The Hessian matrix H can be initialized as any symmetric positive definite matrix, such as the identity matrix I. To avoid inverting H, the BFGS method and the Davidon–Fletcher–Powell (DFP) method use formulas that approximate the inverse Hessian at each update rather than inverting H directly. The BFGS method uses the updating formula shown above to approximate the Hessian matrix H, while the DFP method uses the same updating formula but substitutes y_k for s_k to calculate the approximate inverse Hessian H^{-1} [55]. The inverse Hessian matrix H_{k+1}^{-1} can be calculated as follows:

$$
H_{k+1}^{-1} = H_k^{-1} + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k^{-1} y_k y_k^T H_k^{-1}}{y_k^T H_k^{-1} y_k} \tag{17}
$$

Equation (17) approximates the inverse Hessian matrix in the DFP method. The BFGS method uses a similar updating formula but substitutes y_k for s_k in the equation. Both the BFGS and DFP methods avoid directly inverting the Hessian matrix H by employing these formulas to approximate the inverse Hessian at each iteration [55,56].

At each k-th iteration, a line search is performed in the direction d_k, which is calculated as follows [55,56]:

$$
d_k = -H_k^{-1} \cdot \nabla f(x_k) \tag{18}
$$

The next solution x_{k+1} is calculated as follows:

$$
x_{k+1} = x_k + \alpha_k d_k \tag{19}
$$

where x_k represents the current iterate, d_k is the search direction, and α_k is a scalar step length parameter chosen to satisfy the Wolfe conditions [55,56].

The pseudo-code of the QN (BFGS) method is presented in the method in Algorithm 2. It utilizes the BFGS updating formula to approximate the Hessian matrix H, and the convergence tolerance ϵ is used to check the stopping criteria. The line search procedure determines the step size α_k at each iteration. It is worth noting that QN methods exhibit a fast convergence rate and are relatively easy to implement. However, they require a good initial guess and are limited to finding the local minima in multimodal problems [55,56].

Algorithm 2 QN (BFGS).

```
Require: Given starting point $x_0$, convergence tolerance $\epsilon > 0$
set $k = 0$, $H_0 = I$
while $||\nabla f_k|| > \epsilon$ do
    Compute search direction, $d_k = -H_k \cdot \nabla f_k$
    Compute next solution, $x_{k+1} = x_k + \alpha_k d_k$ using line search
    Define $s_k = x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$
    Compute $H_{k+1}$ using Equation (15)
    set $k = k + 1$.
end while
```

The built-in MATLAB function 'fminunc' is commonly utilized for solving optimization problems [58,59]. It utilizes the QN method as the default optimization algorithm and is designed to be robust and efficient. 'fminunc' can handle various optimization problems, including non-smooth and non-convex functions, providing flexibility for solving different types of optimization problems [58,59].

3.5. Proposed Hybrid Methods

To overcome the limitations of the PSO method and enhance its solution refinement capabilities, we introduce a new hybrid approach, the HPSO-QN method. This method merges the global search capability of PSO with the fast convergence and local search proficiency of the quasi-Newton method. While PSO is known for its global search capability, it can be slow to converge and prone to getting trapped in locally optimal solutions. Conversely, the quasi-Newton method offers fast convergence but is limited to the local minima on multi-modal problems and is sensitive to initial conditions.

Drawing inspiration from the successes of previous hybrid techniques [35,36,60,61], our proposed hybrid approach employs both the PSO and QN algorithms to identify the mechanical parameters of the 2MM. Unlike standard PSO algorithms, this hybrid method leverages the PSO algorithm's strength for global search and the QN method's efficiency in local search. Initially, the PSO identifies a potential global optimal region in the search space, after which the algorithm switches to the QN method for more rapid convergence and solution refinement. This approach combines the benefits of both global and local search methods, incorporating a stopping criterion to avoid being trapped in local minima.

Diversifying the approach further, we introduce three variations of the HPSO-QN method. Each variant begins with the PSO to find an initial *gBest* solution, which then serves as the starting point for the QN method. The key difference between these methods lies in their strategies to select particles for a second run of the PSO algorithm after the QN method has identified a minimum. These strategies provide new approaches to hybridizing the PSO and QN methods, leveraging the strengths of each, while mitigating their limitations. The following bullet points provide a brief outline of these proposed methods:

1. HPSO-QN Sequential Method: PSO finds the *gBest* solution, which is then improved by QN.
2. HPSO-QN Single Local Search Method: PSO uses a stopping criterion to find the *gBest* solution, which is then refined by the QN method.
3. HPSO-QN Multi Local Search Method: SPSO is used until a stopping criterion is met, then a percentage of the best particles are selected and improved using the QN method.

3.5.1. HPSO-QN Sequential Method

In the HPSO-QN sequential method (SM), PSO is utilized to find the *gBest* solution. The pseudo-code of the HPSO-QN SM algorithm is presented in Algorithm 3. In the same iteration, this solution serves as the starting point for the QN method. If the QN method's solution is better than that of the PSO, the *gBest* solution from PSO is replaced with the solution from the QN method. This method improves the *gBest* solution obtained from the PSO algorithm with each iteration by balancing the exploration of the search space by PSO and the exploitation of the local minima by the QN method, thus enhancing both the local and global searches in each iteration.

3.5.2. HPSO-QN Single Local Search Method

In the second method, the HPSO-QN single local search method (SLSM), a stopping criterion is introduced to enhance the exploration of the search space by the PSO algorithm. This criterion allows the PSO algorithm to discover the *gBest* solution. Once this criterion is met, the *gBest* solution serves as the starting point for the QN method before moving on to the next PSO iteration. The QN method then performs a local search on this solution to further improve it. The stopping criterion is defined as follows:

$$\Delta Fcost = \left| \frac{(Fcost(gBest_k) - Fcost(gBest_{k-1})}{Fcost(gBest_{k-1})} \right| \times 100, \quad \Delta Fcost < F_{th} \tag{20}$$

Here, F_{th} represents the user-defined threshold limit. This stopping criterion ensures a balance between exploration and exploitation, facilitating faster convergence and more precise solutions. By refining the PSO-found solution with QN, the chances of discovering the true global minimum are increased, resulting in more efficient search space exploration. The pseudo-code of the HPSO-QN SLSM algorithm is presented in Algorithm 4.

Algorithm 3 HPSO-QN Sequential Method.

```
Require: $N_{PSO}, k_{max}, x_{min}, x_{max}, w_{min}, w_{max}, \varphi, c_1$, and $c_2$
Set $k = 0$, randomly initialize particles' positions $x_i(k)$ and velocities $v_i(k)$
repeat
    set $k = k + 1$.
    for each particle $P_k(i),\ i = 1, 2, ..N_{PSO}$ do
        Evaluate $Fcost(x_k(i))$ using Equation (8)
        Update $pBest_k(i)$ and $gBest_k$ as shown in Algorithm 1
        Calculate $v_{k+1}(i)$ and $x_{k+1}(i)$ using Equations (13) and (10)
    end for
    $x_0 = gBest_k$
    $QN = fminunc(Fcost(x), x_0,$ 'Algorithm', 'quasi-newton')
    if $Fcost(QN) < Fcost(gBest_k)$ then $gBest_k = QN$
    end if
until the termination criterion is met
```

Algorithm 4 HPSO-QN single local search method.

```
Require: $N_{PSO}, k_{max}, x_{min}, x_{max}, w_{min}, w_{max}, \varphi, c_1, c_2$ and $F_{th}$
Set $k = 0$, randomly initialize particles' positions $x_i(k)$ and velocities $v_i(k)$
repeat
    set $k = k + 1$.
    for each particle $P_k(i),\ i = 1, 2, ..N_P$ do
        Evaluate $Fcost(x_k(i))$ using Equation (8)
        Update $pBest_k(i)$ and $gBest_k$ as shown in Algorithm 1
        Calculate $v_{k+1}(i)$ and $x_{k+1}(i)$ using Equations (13) and (10)
    end for
    if $\Delta Fcost < F_{th}$ then
        $x_0 = gBest_k$
        $QN = fminunc(Fcost(x), x_0,$ 'Algorithm', 'quasi-newton')
        if $Fcost(QN) < Fcost(gBest_k)$ then $gBest_k = QN$
        end if
    end if
until the termination criterion is met
```

3.5.3. HPSO-QN Multi Local Search Method

The third method, the HPSO-QN multi local search method (MLSM), improves upon the previous HPSO-QN SLSM by incorporating a new approach for balancing global and local search. The algorithm begins with the SPSO algorithm until a predefined stopping criterion is reached. Then, a specified percentage of the best particles is selected for the QN method, allowing the algorithm to focus on multiple promising solutions. The selected particles are determined by

$$N_{QN} = N_{PSO} \times B \tag{21}$$

where B represents the percentage of selected best particles and N_{PSO} denotes the total number of particles in the PSO algorithm. The QN method is then applied to each selected particle, and the *gBest* solution is replaced by the best solution found among the selected particles. This hybrid approach achieves a balance between global and local search by utilizing PSO to explore the search space and identify promising solutions, while QN fine tunes these solutions, increasing the likelihood of finding the global minimum. The pseudo-code of the HPSO-QN MLSM algorithm is presented in Algorithm 5.

Algorithm 5 HPSO-QN Multi Local Search Method.

```
Require: $N_{PSO}$, $N_{QN}$, $k_{max}$, $x_{min}$, $x_{max}$, $w_{min}$, $w_{max}$, $\varphi$, $c_1$, $c_2$ and $F_{th}$
Set $k = 0$, randomly initialize particles' positions $x_i(k)$ and velocities $v_i(k)$
repeat
    set $k = k + 1$.
    for each particle $P_k(i),\ i = 1, 2, ..N_{PSO}$ do
        Evaluate $Fcost(x_k(i))$ using Equation (8)
        Update $pBest_k(i)$ and $gBest_k$ as shown in Algorithm 1
        Calculate $v_{k+1}(i)$ and $x_{k+1}(i)$ using Equations (13) and (10)
    end for
    Select $N_QN$ particles for QN local search
    for each particle $Q(j),\ j = 1, 2, ..N_{QN}$ do
        $x_0(j) = pBest_k(j)$
        $QN(j) = fminunc(Fcost(x), x_0(j),$ 'Algorithm', 'quasi-newton'$)$
        if $Fcost(QN(j)) < Fcost(gBest_k)$ then $gBest_k = QN(j)$
        end if
    end for
until the termination criterion is met
```

The proposed methods were implemented and evaluated on a computer system with the following specifications: 16 GB of RAM, a 64-bit operating system, and an Intel(R) Xeon(R) E-2124 CPU @ 3.30GHz processor. The simulations were conducted using Matlab 2023a software [58].

4. Experimental Setup

The experimental setup for the 2MM Flexible Drive System is illustrated in Figure 2. It includes two incremental encoders for measuring the motor and load speeds, as well as a current sensor for measuring the armature current. Real-time control is achieved using the dSPACE DS1401/1512/1513 MicroAutobox II controller, which interfaces with MATLAB/Simulink and Control-Desk.

The system comprises an actuating motor on the left side, connected to an inertia disk, a flexible coupling, an encoder, and another inertia disk, forming the first mass. On the right side, there is the second mass, an inertia load connected to a load motor. The actuating motor used in this research is a Buhler DC Motor of model 1.13.075.214. High-resolution encoders with 40,000 PPR are used to measure the angular speeds of the masses. The inertias are supported by ball bearings and interconnected by a torsional shaft.

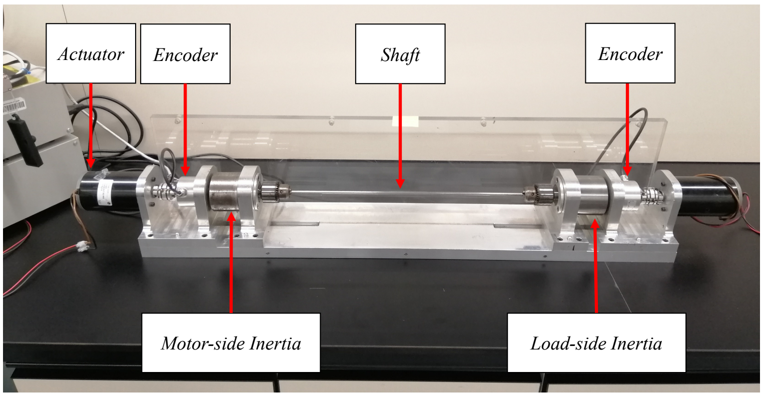


Figure 2. The two-mass model (2MM) experimental platform.

To facilitate the identification process, an open-loop current controller with hysteresis current control was implemented in MATLAB/Simulink and later deployed to the dSPACE controller. The hysteresis current controller is sampled at a rate of 20 µs, while the speed measurements are sampled every 1 ms. Figure 3 shows the overall system block diagram, including the hysteresis current controller implementation. In current control mode, the dynamic equation does not explicitly include electrical components like resistance R_a and inductance L_a. The current controller effectively handles these parameters to account for their impact on system dynamics.

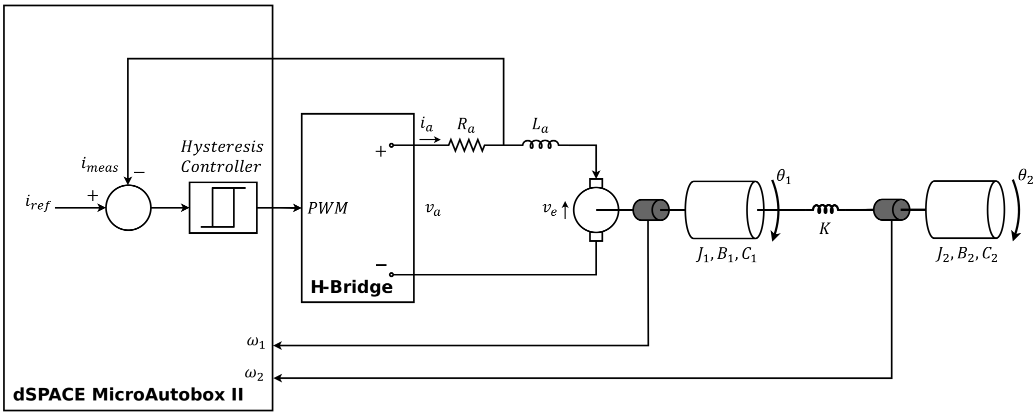


Figure 3. Block diagram illustrating the overall system configuration of the two-mass model (2MM).

The 2MM system was excited by periodic chirp input current signals with frequencies ranging from $f_1 = 1$ Hz to $f_2 = 40$ Hz, an amplitude current of $A = 3$ A, and a time duration ranging from $T_0 = 0$ s to $T_f = 3$ s. Previous research was conducted to identify the parameters of the system. These analyses involved identifying the inertias and stiffness of the system through frequency domain analysis, as well as determining the friction coefficients through time domain analysis [1,2]. In the time domain identification, the friction coefficients for both the motor and load sides were estimated individually. The approach described in [62] was utilized, which involved using a ramp function as the torque input and performing numerical analysis on speed and torque data to estimate the viscous friction coefficient and Coulomb friction coefficient. For more detailed information on the identification techniques used, interested readers are encouraged to refer to the mentioned papers [1,2,62]. The parameters identified in our previous research serve as benchmarks for comparing the results obtained from the proposed methods. In this study, we normalized these parameters using the rated values of the DC motor provided in the datasheet. This normalization improves the interpretability, robustness, and consistency of the results, as well as the numerical stability of the simulation models. The identified benchmark parameters of the 2MM system are presented in Table 1, both in their SI units and normalized units.

Table 1. Benchmark parameters of the two-mass model (2MM).

Parameter	SI Units	Normalized Units
Motor Inertia (J_1)	1.5602×10^{-3} kg·m^2	0.8713 s
Load Inertia (J_2)	1.3966×10^{-3} kg·m^2	0.7799 s
Shaft stiffness (K)	19.4840 N·m/rad	10,881.9233 p.u.
Motor Coulomb Friction (C_1)	0.0182 N·m	0.0304 p.u.
Load Coulomb Friction (C_2)	0.0162 N·m	0.0271 p.u.
Motor Viscous Friction (B_1)	3.7167×10^{-3} N·m/rad·s	2.0759 p.u.
Load Viscous Friction (B_2)	3.5530×10^{-3} N·m/rad·s	1.9844 p.u.

5. Results and Discussion

To evaluate the effectiveness of the proposed hybrid PSO methods, we conducted a series of simulation runs in this study. Four different algorithms were employed: SPSO, HPSO-QN SM, HPSO-QN SLSM, and HPSO-QN MLSM. Each algorithm was tested independently five times, with randomized initial positions. The proposed algorithms utilized a particle count (N_{PSO}) of 50 and a maximum iteration count (k_{max}) of 300. The choice of

300 iterations as the maximum value was determined by monitoring the convergence of the algorithm, specifically through the *Fcost* plot analysis.

To define the search space bounds, the following upper and lower bounds for the position and velocity vectors were assigned:

$$\begin{aligned}\vec{x}_{\min}(k) &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\\ \vec{x}_{\max}(k) &= \begin{bmatrix}4 & 4 & 40000 & 0.5 & 0.5 & 5 & 5\end{bmatrix}\end{aligned} \tag{22}$$

These bounds correspond to the following parameters:

$$\vec{x}(k) = \begin{bmatrix}J_{1n} & J_{2n} & K_n & C_{1n} & C_{2n} & B_{1n} & B_{2n}\end{bmatrix} \tag{23}$$

To assess the effectiveness of the switching criteria for the local search methods (HPSO-QN SLSM and HPSO-QN MLSM), we evaluated different values of F_{th}: specifically, 5%, 10%, and 20%. Our findings showed that HPSO-QN SLSM achieved the lowest cost function and standard deviation of the cost function when F_{th} was set to 5%. Furthermore, we investigated the impact of different values of B in the HPSO-QN MLSM method. We tested two scenarios: selecting either the 5 or 10 best particles for the QN local search with B set to 5% and 10%, respectively. After evaluating the results, we observed that the lowest standard deviation of the cost function and parameters, as well as the lowest absolute percentage error (APE) of the parameters, were achieved when F_{th} was set to 5% and B was set to 5%. The results of the lowest cost function (*Fcost*) using $F_{th} = 5\%$ and $BP = 5\%$ are presented in Table 2.

Table 2. Selected identified parameters of the 2MM system using the proposed methods with the lowest cost function.

PSO Methods	**Parameters**							$\boldsymbol{Fcost}$ $(\times 10^{-6})$
	J_{1n} **(s)**	J_{2n} **(s)**	K_n (s^{-1})	C_{1n} **(p.u)**	C_{2n} **(p.u)**	B_{1n} **(p.u)**	B_{2n} **(p.u)**	
SPSO	0.9211	1.1142	13,323.9963	0.0093	0.0353	1.6971	4.5508	3.3859
HPSO-QN SM	0.8642	1.0453	12,497.3021	0.0299	0.0261	0.7666	3.9063	2.9167
HPSO-QN SLSM	0.7718	0.8310	10,532.9554	0.0197	0.0303	3.6192	1.4733	2.4787
HPSO-QN MLSM	0.7987	0.8983	11,169.0221	0.0272	0.0325	0.3083	3.7556	2.3200

Based on these results and compared with the benchmark parameters in Table 1, the proposed HPSO-QN methods demonstrated their effectiveness in estimating the mechanical parameters of the 2MM system. The HPSO-QN MLSM method achieved the lowest cost function value of 2.32×10^{-6} with a high degree of speed response accuracy. Table 3 compares the standard deviation values of the identified parameters using four different PSO methods. The HPSO-QN MLSM method consistently exhibits the lowest standard deviation values, highlighting its robustness. In contrast, the SPSO method demonstrates relatively higher standard deviation values compared to the HPSO-QN methods.

Table 3. A comparison of the standard deviation of the identified parameters across five independent runs using the proposed methods.

PSO Methods	**Parameters**							$\boldsymbol{Fcost}$ $(\times 10^{-6})$
	J_{1n} **(s)**	J_{2n} **(s)**	K_n (s^{-1})	C_{1n} **(p.u)**	C_{2n} **(p.u)**	B_{1n} **(p.u)**	B_{2n} **(p.u)**	
SPSO	0.4199	0.3198	4356.4730	0.0142	0.0152	1.3070	1.6753	3.7320
HPSO-QN SM	0.1714	0.2817	2137.6343	0.0119	0.0146	0.9651	1.0390	2.0512
HPSO-QN SLSM	0.1927	0.3636	3548.1141	0.0171	0.0082	1.2920	1.2398	1.2274
HPSO-QN MLSM	0.0218	0.0512	461.0188	0.0052	0.0028	1.2008	1.4837	0.0787

The improvement ratios of the HPSO-QN MLSM method over other methods for each identified parameter are summarized in Table 4. The HPSO-QN MLSM method consistently yields substantially lower standard deviation values across all parameters compared to the other methods. For example, the standard deviation of J_{1n} using the HPSO-QN MLSM is reduced by a factor of 19.26 compared to SPSO, 7.86 compared to HPSO-QN SM, and 8.84 compared to HPSO-QN SLSM. Other parameters also experience notable improvements. The standard deviation of J_{2n} is reduced by factors of 6.25, 5.50, and 7.10 compared to SPSO, HPSO-QN SM, and HPSO-QN SLSM, respectively, when using the HPSO-QN MLSM method. Similarly, the K_n parameter sees reduction factors of 9.45, 4.63, and 7.70. Similar improvements are observed for C_{1n}, C_{2n}, B_{1n}, and B_{2n}. It is important to note that for B_{1n} and B_{2n}, HPSO-QN SM demonstrates slightly lower standard deviation values compared to HPSO-QN MLSM. However, HPSO-QN MLSM still outperforms the other PSO methods in terms of identifying parameters with lower standard deviation values for the majority of parameters. While the HPSO-QN SM method also shows lower standard deviation values for B_{1n} and B_{2n}, HPSO-QN MLSM surpasses it for most parameters. Consequently, the HPSO-QN MLSM method proves to be a promising approach for parameter identification problems in similar systems.

Table 4. Improvement ratio of the standard deviation of the HPSO-QN MLSM method compared to other methods.

PSO Methods	Parameters							$Fcost$ $(\times 10^{-6})$
	J_{1n} (s)	J_{2n} (s)	K_n (s^{-1})	C_{1n} (p.u)	C_{2n} (p.u)	B_{1n} (p.u)	B_{2n} (p.u)	
SPSO	19.26	6.25	9.45	2.73	5.43	1.09	1.13	48.06
HPSO-QN SM	7.86	5.50	4.63	2.29	5.21	0.80	0.70	26.06
HPSO-QN SLSM	8.84	7.10	7.70	3.29	2.92	1.06	0.83	15.60

The results in Table 5 offer a comparison of the proposed methods based on the average identified parameters across five independent runs. The HPSO-QN MLSM method achieved the lowest $Fcost$ value of 2.4341 × 10^{-6}, indicating its effectiveness in accurately estimating the mechanical parameters of the 2MM system. When comparing the average parameters obtained from HPSO-QN SM with those obtained from SPSO, the percentage difference for J_{1n} is around 15.77%, while for J_{2n}, it is around 15.10%. The percentage difference obtained for K_n is around 15.22%, and for C_{1n} and C_{2n}, they are 33.54% and 5.86%, respectively. As for B_{1n} and B_{2n}, the percentage differences are 18.33% and 9.97%, respectively. Regarding $Fcost$, the percentage difference obtained is 31.41% lower compared to SPSO.

Table 5. Performance comparison of the proposed method: average of identified parameters across five independent runs.

PSO Methods	Parameters							$Fcost$ $(\times 10^{-6})$
	J_{1n} (s)	J_{2n} (s)	K_n (s^{-1})	C_{1n} (p.u)	C_{2n} (p.u)	B_{1n} (p.u)	B_{2n} (p.u)	
SPSO	1.2457	1.3193	16,554.7629	0.0158	0.0256	2.5953	3.0403	7.1430
HPSO-QN SM	1.0492	1.1201	14,035.9131	0.0211	0.0271	2.1197	2.7372	4.8995
HPSO-QN SLSM	0.8393	0.9980	11,961.5283	0.0289	0.0316	2.5762	1.4984	3.9044
HPSO-QN MLSM	0.8151	0.8974	11,265.7222	0.0287	0.0332	1.6844	1.7908	2.4341

When comparing HPSO-QN SLSM with HPSO-QN SM, percentage differences were around 20.01% for J_{1n}, 10.90% for J_{2n}, and 14.78% for K_n. The friction coefficients (C_{1n}, C_{2n}, B_{1n}, and B_{2n}) estimated by HPSO-QN SLSM showed percentage differences ranging from 20% to 45% compared to HPSO-QN SM. The percentage difference in $Fcost$ between HPSO-QN SM and HPSO-QN SLSM was around 20.31%, with HPSO-QN SLSM providing a lower value. Comparing HPSO-QN MLSM with HPSO-QN SLSM, the percentage differences obtained for J_{1n}, J_{2n}, and K_n were 2.88%, 10.08%, and 5.82%, respectively. For the friction coefficients (C_{1n} and C_{2n}), the average parameter differences between HPSO-QN MLSM and HPSO-QN SLSM were 0.69% and 5.06%, respectively. The average differences for B_{1n} and B_{2n} were 34.62% and 19.51%, respectively. The average *Fcost* estimated by HPSO-QN MLSM was 37.66% lower than that estimated by HPSO-QN SLSM. Finally, comparing the average parameters obtained from HPSO-QN MLSM to the SPSO method, HPSO-QN MLSM achieved a significant improvement in *Fcost*, with a decrease of 63.92%. The percentage differences obtained for J_{1n} and J_{2n} were around 34.57% and 31.98%, respectively. For K_n, the percentage difference obtained was approximately 31.95%. The percentage differences obtained for C_{1n} and C_{2n} were 81.65% and 29.69%, respectively. For B_{1n} and B_{2n}, the percentage differences obtained were 35.10% and 41.10%, respectively. These results suggest that HPSO-QN MLSM outperforms SPSO in identifying the 2MM parameters and minimizing *Fcost*.

Table 6 presents the accuracy of the estimated parameters compared to the benchmark parameters, measured by the absolute percentage error (APE). The APE is used as a measure of the deviation between the estimated and benchmark parameters, and it is defined as

$$APE = \frac{|Estimated - Benchmark|}{Benchmark} \times 100 \tag{24}$$

Table 6. Performance comparison of the proposed method: absolute percentage error comparison of estimated and actual parameters.

PSO Methods	J_{1n} (%)	J_{2n} (%)	K_n (%)	C_{1n} (%)	C_{2n} (%)	B_{1n} (%)	B_{2n} (%)
SPSO	42.95	69.13	52.13	48.03	5.73	25.02	53.21
HPSO-QN SM	20.40	43.62	28.98	30.59	0.18	2.11	37.93
HPSO-QN SLSM	3.68	27.95	9.92	4.93	16.82	24.10	24.50
HPSO-QN MLSM	6.46	15.05	3.53	5.59	22.74	18.86	9.76

The HPSO-QN SM method shows the lowest error for C_{2n} (APE of 0.18%) and performs well for B_{1n} (APE of 2.11%). The HPSO-QN SLSM method exhibits the lowest error for J_{1n} (APE of 3.68%) and performs well for C_{1n} (APE of 4.93%). In terms of motor-side and load-side inertias (J_{1n} and J_{2n}) and shaft stiffness (K_n), the HPSO-QN MLSM method achieves the lowest error, with APE values ranging from 3.53% to 15.05%. Additionally, it performs well for B_{2n} with an APE of 9.76%.

These results highlight the effectiveness of the HPSO-QN method in accurately identifying mechanical parameters, providing more reliable results compared to other methods. However, it is important to note that there is still a significant discrepancy between the estimated and actual values for the friction coefficients. This discrepancy may be attributed to factors such as measurement noise and the potential variability of these coefficients over time. Despite this limitation, the HPSO-QN method yields lower errors for the friction coefficients compared to the SPSO method.

The convergence of the proposed methods is demonstrated in Figure 4, showing the *Fcost* versus the number of iterations for each algorithm. The plot highlights the effectiveness of the implemented switching criteria and particle selection in the hybrid method. Compared to the SPSO algorithm, which often gets trapped in the local optima, the HPSO-QN methods demonstrate faster convergence to a low-cost function in the initial stages and continue to search within that region. Moreover, the SPSO algorithm demonstrates a slower convergence speed when compared to the HPSO-QN methods.

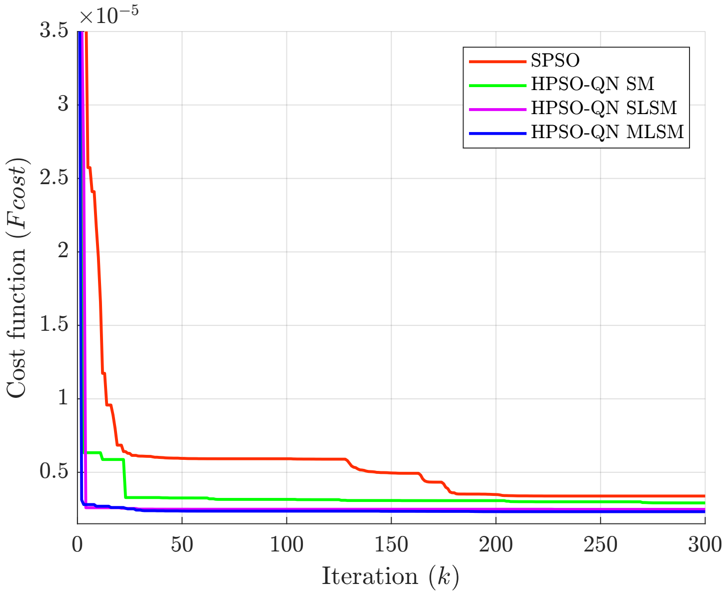
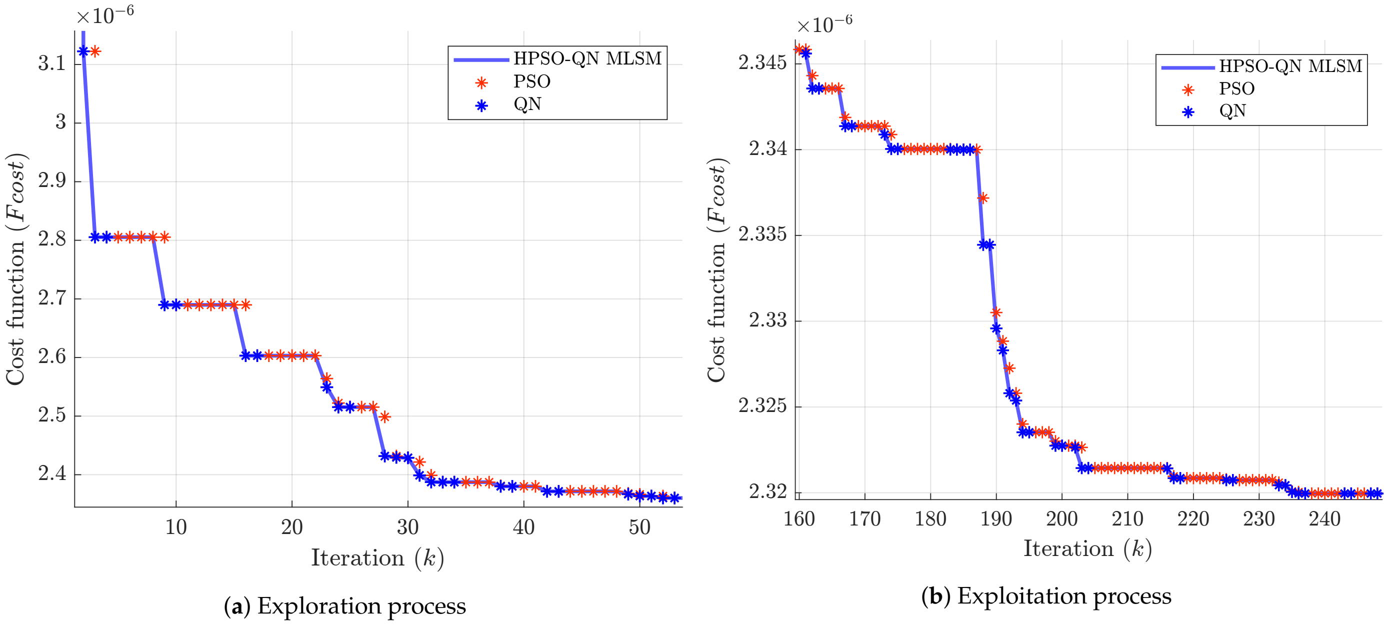


Figure 4. Cost function ($Fcost$) analysis of the proposed methods with the best performance.

Figure 5 compares the $Fcost$ of solutions obtained using the Hybrid PSO-QN MLSM optimization process for both exploration and exploitation processes. This graph demonstrates the effect of both methods at each iteration. It is evident that the QN method aids in escaping from the local optima and finding superior solutions during both exploration and exploitation phases, while the PSO method tends to remain stable in the optimum for a period of iterations.



(**a**) Exploration process

(**b**) Exploitation process

Figure 5. Comparison of $Fcost$ values obtained using the HPSO-QN MLSM optimization process.

The accuracy of the estimated speeds was evaluated using performance measures, such as the IAE, ISE, and $ITAE$ metrics as shown in Figure 6. These metrics provide a quantitative measure of the difference between simulated and experimental speed responses. The IAE, ISE, and $ITAE$ are defined as

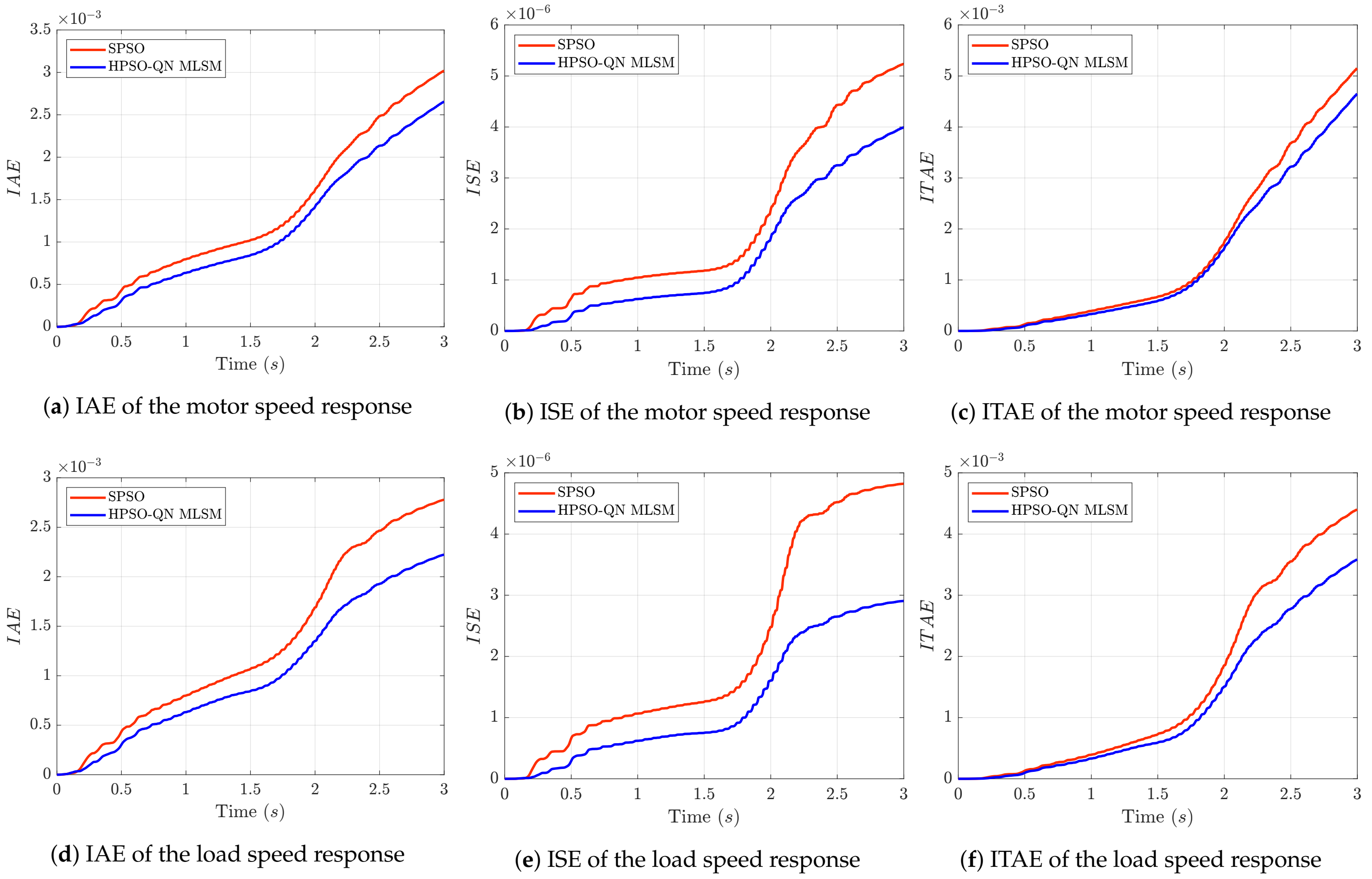
$$IAE = \int_{T_0}^{T_f} |e(t)|dt \tag{25}$$

$$ISE = \int_{T_0}^{T_f} e^2(t)dt \tag{26}$$

$$ITAE = \int_{T_0}^{T_f} t|e(t)|dt \tag{27}$$

The error $e(t)$ between the estimated and measured motor speeds is given by $\Omega_1(t) - \hat{\Omega}_2(t)$, and the error between the estimated and measured load speeds is $\Omega_2(t) - \hat{\Omega}_2(t)$. The performance of the estimated model in accurately capturing the speed responses is indicated by smaller values of these measures.

The plot indicates that the hybrid PSO-QN MLSM method achieved lower error values compared to SPSO. This demonstrates a higher degree of precision and a significant reduction in error using the hybrid PSO-QN MLSM method. The comparison highlights the accuracy and effectiveness of the hybrid PSO algorithm in estimating the mechanical parameters of the 2MM system.



(**a**) IAE of the motor speed response

(**b**) ISE of the motor speed response

(**c**) ITAE of the motor speed response

(**d**) IAE of the load speed response

(**e**) ISE of the load speed response

(**f**) ITAE of the load speed response

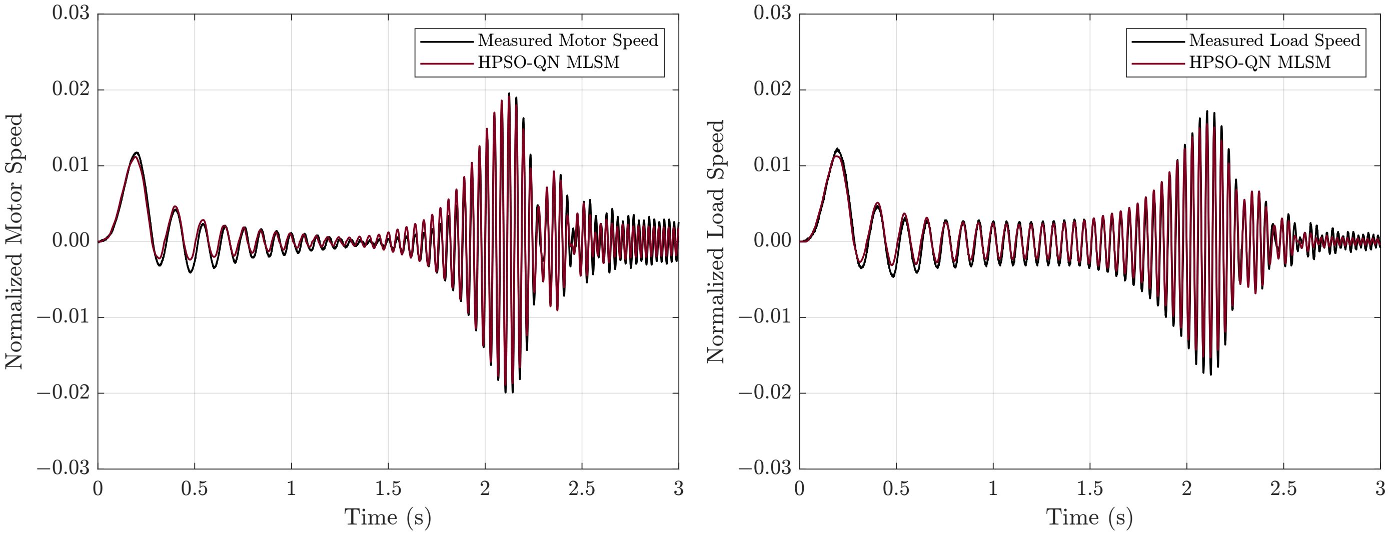
Figure 6. Comparison of integral absolute error (IAE), integral squared error (ISE), and integral time absolute error (ITAE) values between the simulated and experimental motor and load speed responses obtained using HPSO-QN MLSM and SPSO.

Figure 7 compares the simulation results of the motor and load speeds obtained from the algorithm with the lowest *Fcost* value to the corresponding experimental measurements for the same excitation signals. It can be observed that the speed responses of the estimated model match the measured speed.

Finally, Table 7 presents the computation time for the PSO algorithm and its variants. The HPSO-QN SM method has a longer computation time compared to SPSO due to the QN process at each PSO iteration. The HPSO-QN SLSM method introduces a stopping criterion to enhance the search for an optimal region, which increases the computation time but improves the parameter quality and performance compared to SLSM and SPSO. The HPSO-QN MLSM method selects multiple particles for QN local search, resulting in increased computation time. However, it provides optimal and consistent parameters with the lowest *Fcost* values, indicating superior performance. Therefore, HPSO-QN MLSM is a promising approach for accurate and consistent parameter identification.

Table 7. Computation time for the proposed methods.

PSO Methods	Time (minutes)
SPSO	30.22
HPSO-QN SM	161.31
HPSO-QN SLSM	175.65
HPSO-QN MLSM	468.59



(**a**) Motor Speed

(**b**) Load Speed

Figure 7. Simulation response using estimated parameters compared with measured speeds (motor and load side).

6. Conclusions

In this study, a hybrid optimization method called HPSO-QN was proposed to accurately identify the mechanical parameters of flexible 2MM drive systems using PSO methods. The research consisted of the following key aspects:

- Literature review: A comprehensive review of existing methods for parameter identification in two-mass drive systems was conducted to establish the background for the proposed research.
- System modeling and control: An accurate dynamic model of the two-mass drive system was developed, and a hysteresis current controller was implemented to aid in the mechanical identification process.
- Hybrid optimization method: The proposed hybrid PSO method, known as HPSO-QN, was implemented to identify the mechanical parameters of the 2MM drive system. Its effectiveness was evaluated using experimental data, and the HPSO-QN MLSM method exhibited the best performance in terms of accuracy and efficiency.

The following conclusions can be drawn from the results:

- The HPSO-QN MLSM method achieved the lowest cost function value of 2.32×10^{-6} with five independent runs.

- The standard deviation values demonstrated the robustness of the HPSO-QN methods, with the HPSO-QN MLSM method having the lowest values for most parameters. The motor-side Coulomb friction parameter showed the lowest standard deviation of 0.0028 among all identified parameters.
- Comparing the proposed methods, the HPSO-QN MLSM method proved to be the most effective in terms of *Fcost*, indicating its accuracy in parameter estimation. The HPSO-QN SM method achieved the lowest cost function value of 0.18% for load-side Coulomb friction.
- The absolute percentage error (APE) values indicated that the HPSO-QN methods, particularly MLSM, exhibited low errors in estimating mechanical parameters and friction coefficients compared to the standard method.

Possible future directions include the following:

- Exploring more realistic friction models, such as the Dahl and LuGre models, to improve the estimation of viscous friction coefficients with higher APE values.
- Applying the HPSO-QN method for parameter estimation in other dynamic systems, such as battery and supercapacitor models. This could help in improving the accuracy of models used in energy systems and electric vehicles.
- Evaluating the impact of the HPSO-QN methods on the performance of control systems in real-world applications, such as robotics and automotive systems.
- Extending the HPSO-QN method to incorporate global optimization techniques, specifically multi-objective optimization, to optimize multiple objectives simultaneously.
- Extending the HPSO-QN method to estimate parameters in more complex systems, such as three-mass model systems, to capture additional dynamics.
- Utilizing the HPSO-QN method in problems of machine learning and deep learning, where high-dimensional parameter space is prevalent, which could enhance the efficiency of hyperparameter tuning.
- Investigating the potential of integrating deep learning methods with the HPSO-QN approach to enhance parameter estimation accuracy and developing hybrid models that combine data-driven and physics-based approaches.

In conclusion, the proposed HPSO-QN methods offer a promising solution for accurately estimating the mechanical parameters of 2MM systems, providing reliable and consistent results compared to other methods. Moreover, the general optimization capabilities of the HPSO-QN method indicate its potential applicability in a wide range of fields beyond mechanical systems, including energy systems, machine learning, and more. The results of this study demonstrate the effectiveness of this method in improving simulation models and optimizing control systems.

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

Funding: This research was funded by the College of Engineering, American University of Sharjah, Sharjah, UAE.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This paper represents the opinions of the authors and does not mean to represent the position or opinions of the American University of Sharjah.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

2MM	Two Mass Model
APE	Absolute Percentage Error
DC	Direct Current
FRF	Frequency Response Function
HPSO-QN	Hybrid Particle Swarm Optimization Quasi-Newton
IAE	Integral Absolute Error
ITAE	Integral Time Absolute Error
ISE	Integral Squared Error
MLSM	Multi Local Search Method
PSO	Particle Swarm Optimization
QN	Quasi-Newton
SM	Sequential Method
SLSM	Single Local Search Method
SPSO	Standard Particle Swarm Optimization

References

1. Dhaouadi, R.; Hafez, I. Identification of Shaft Stiffness and Inertias in Flexible Drive Systems. *J. Robot. Mechatron.* **2023**, *35*, 212–217. [CrossRef]
2. Hafez, I.; Dhaouadi, R. Application of Particle Swarm Optimization for the Identification of Two-Mass Electric Drive Systems. In Proceedings of the 2022 8th International Conference on Control, Decision and Information Technologies (CoDIT), Istanbul, Turkey, 17–20 May 2022; Volume 1, pp. 758–763. [CrossRef]
3. Hafez, I.; Dhaouadi, R. Parameter Identification of DC Motor Drive Systems using Particle Swarm Optimization. In Proceedings of the 2021 International Conference on Engineering and Emerging Technologies (ICEET), Istanbul, Turkey, 27–28 October 2021; pp. 1–6. [CrossRef]
4. Ke, C.; Wu, A.; Bing, C. Mechanical parameter identification of two-mass drive system based on variable forgetting factor recursive least squares method. *Trans. Inst. Meas. Control* **2019**, *41*, 494–503. [CrossRef]
5. Ozmen Koca, G.; Korkmaz, D. Neural Network Based Control of a Two-Mass Drive System. *Int. J. Intell. Syst. Appl.* **2019**, *7*, 92–98. [CrossRef]
6. Saarakkala, S.E.; Leppinen, T.; Hinkkanen, M.; Luomi, J. Parameter estimation of two-mass mechanical loads in electric drives. In Proceedings of the 2012 12th IEEE International Workshop on Advanced Motion Control (AMC), Sarajevo, Bosnia and Herzegovina, 25–27 March 2012; pp. 1–6. [CrossRef]
7. Dhaouadi, R.; Kubo, K. Transfer function and parameters identification of a motor drive system using adaptive filtering. In Proceedings of the 4th IEEE International Workshop on Advanced Motion Control—AMC '96-MIE, Mie, Japan, 18–21 March 1996; Volume 2, pp. 588–593. [CrossRef]
8. Sadovoy, O.V.; Nazarova, O.S.; Bondarenko, V.I.; Pirozhok, A.V.; Hutsol, T.D.; Nurek, T.; Glowacki, S. *Modeling and Research of Electromechanical Systems of Cold Rolling Mills*; Traicon: Karnataka, India, 2020; 138p.
9. Bajpai, R.S.; Goyal, M.; Gupta, R. Modeling and control of variable speed wind turbine using laboratory simulator. *J. Renew. Sustain. Energy* **2015**, *7*, 053127. [CrossRef]
10. Östring, M.; Gunnarsson, S.; Norrlöf, M. Closed-loop identification of an industrial robot containing flexibilities. *Control Eng. Pract.* **2003**, *11*, 291–300. [CrossRef]
11. Valenzuela, M.; Bentley, J.; Lorenz, R. Evaluation of torsional oscillations in paper machine sections. *IEEE Trans. Ind. Appl.* **2005**, *41*, 493–501. [CrossRef]
12. Ryu, H.M.; Kim, S.J.; Sul, S.K.; Kwon, T.S.; Kim, K.S.; Shim, Y.S.; Seok, K.R. Dynamic load simulator for high-speed elevator system. In Proceedings of the Power Conversion Conference—Osaka 2002 (Cat. No. 02TH8579), Osaka, Japan, 2–5 April 2002; Volume 2, pp. 885–889. [CrossRef]
13. Zhao, S.; Blaabjerg, F.; Wang, H. An Overview of Artificial Intelligence Applications for Power Electronics. *IEEE Trans. Power Electron.* **2021**, *36*, 4633–4658. [CrossRef]
14. Sundaram, K.M.; Hussain, A.; Sanjeevikumar, P.; Holm-Nielsen, J.B.; Kaliappan, V.K.; Santhoshi, B.K. Deep Learning for Fault Diagnostics in Bearings, Insulators, PV Panels, Power Lines, and Electric Vehicle Applications—The State-of-the-Art Approaches. *IEEE Access* **2021**, *9*, 41246–41260. [CrossRef]
15. Mukhamediev, R.I.; Popova, Y.; Kuchin, Y.; Zaitseva, E.; Kalimoldayev, A.; Symagulov, A.; Levashenko, V.; Abdoldina, F.; Gopejenko, V.; Yakunin, K.; et al. Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges. *Mathematics* **2022**, *10*, 2552. [CrossRef]

16. Saponara, S.; Elhanashi, A.; Gagliardi, A. Reconstruct fingerprint images using deep learning and sparse autoencoder algorithms. In *Real-Time Image Processing and Deep Learning 2021*; Kehtarnavaz, N., Carlsohn, M.F., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, DC, USA, 2021; Volume 11736, p. 1173603.
17. Venkatesh, G.S.; Steven Gray, W.; Duffaut Espinosa, L.A. Combining Learning and Model Based Multivariable Control. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 1013–1018. [CrossRef]
18. Jafari, A.H.; Dhaouadi, R.; Jhemi, A. Nonlinear Friction Estimation in Elastic Drive Systems Using a Dynamic Neural Network-Based Observer. *J. Adv. Comput. Intell. Intell. Inform.* **2013**, *17*, 637–646. [CrossRef]
19. Pham, M.; Gautier, M.; Poignet, P. Accelerometer based identification of mechanical systems. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), Washington, DC, USA, 11–15 May 2002; Volume 4, pp. 4293–4298. [CrossRef]
20. Kara, T.; Eker, I. Experimental nonlinear identification of a two mass system. In Proceedings of the 2003 IEEE Conference on Control Applications, Istanbul, Turkey, 25 June 2003; Volume 1, pp. 66–71. [CrossRef]
21. Łuczak, D.; Nowopolski, K. Identification of multi-mass mechanical systems in electrical drives. In Proceedings of the 16th International Conference on Mechatronics—Mechatronika 2014, Brno, Czech Republic, 3–5 December 2014; pp. 275–282. [CrossRef]
22. Petrea, R.A.B.; Oboe, R. A DOB-based Parameter Identification method for Series Elastic Actuators without Load-Side Encoder. In Proceedings of the 2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Sapporo, Japan, 11–15 July 2022; pp. 1755–1762. [CrossRef]
23. Villwock, S.; Pacas, M. Application of the Welch-Method for the Identification of Two- and Three-Mass-Systems. *IEEE Trans. Ind. Electron.* **2008**, *55*, 457–466. [CrossRef]
24. Nowopolski, K.; Wicher, B. Parametric identification of electrical drive with complex mechanical structure utilizing Particle Swarm Optimization method. 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]
25. Sengupta, S.; Basak, S.; Peters, R.A. Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 157–191. [CrossRef]
26. Li, G.; Wang, T.; Chen, Q.; Shao, P.; Xiong, N.; Vasilakos, A. A Survey on Particle Swarm Optimization for Association Rule Mining. *Electronics* **2022**, *11*, 3044. [CrossRef]
27. Robinson, J.; Sinton, S.; Rahmat-Samii, Y. Particle swarm, genetic algorithm, and their hybrids: Optimization of a profiled corrugated horn antenna. In Proceedings of the IEEE Antennas and Propagation Society International Symposium (IEEE Cat. No. 02CH37313), San Antonio, TX, USA, 16–21 June 2002; Volume 1, pp. 314–317. [CrossRef]
28. Shi, X.; Lu, Y.; Zhou, C.; Lee, H.; Lin, W.; Liang, Y. Hybrid evolutionary algorithms based on PSO and GA. In Proceedings of the 2003 Congress on Evolutionary Computation, Canberra, ACT, Australia, 8–12 December 2003; Volume 4, pp. 2393–2399. [CrossRef]
29. Yang, B.; Chen, Y.; Zhao, Z. A Hybrid Evolutionary Algorithm by Combination of PSO and GA for Unconstrained and Constrained Optimization Problems. In Proceedings of the 2007 IEEE International Conference on Control and Automation, Guangzhou, China, 30 May–1 June 2007; pp. 166–170. [CrossRef]
30. Valdez, F.; Melin, P.; Castillo, O. Evolutionary method combining particle swarm optimization and genetic algorithms using fuzzy logic for decision making. In Proceedings of the 2009 IEEE International Conference on Fuzzy Systems, Jeju, Republic of Korea, 20–24 August 2009; pp. 2114–2119. [CrossRef]
31. Xia, X.; Gui, L.; He, G.; Xie, C.; Wei, B.; Xing, Y.; Wu, R.; Tang, Y. A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm. *J. Comput. Sci.* **2018**, *26*, 488–500. [CrossRef]
32. Zhao, F.; Zhang, Q.; Yu, D.; Chen, X.; Yang, Y. A Hybrid Algorithm Based on PSO and Simulated Annealing and Its Applications for Partner Selection in Virtual Enterprise. In *Advances in Intelligent Computing*; Huang, D.S., Zhang, X.P., Huang, G.B., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 380–389.
33. Singh, A.; Sharma, A.; Rajput, S.; Bose, A.; Hu, X. An Investigation on Hybrid Particle Swarm Optimization Algorithms for Parameter Optimization of PV Cells. *Electronics* **2022**, *11*, 909. [CrossRef]
34. Xu, L.; Huang, C.; Li, C.; Wang, J.; Liu, H.; Wang, X. Estimation of tool wear and optimization of cutting parameters based on novel ANFIS-PSO method toward intelligent machining. *J. Intell. Manuf.* **2021**, *32*, 77–90. [CrossRef]
35. dos Santos Coelho, L.; Mariani, V.C. Particle Swarm Optimization with Quasi-Newton Local Search for Solving Economic Dispatch Problem. In Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics, Taipei, Taiwan, 8–11 October 2006; Volume 4, pp. 3109–3113. [CrossRef]
36. Wang, Y.J.; Zhang, J.S.; Zhang, Y.F. A fast hybrid algorithm for global optimization. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 5, pp. 3030–3035. [CrossRef]
37. Dhaouadi, R.; Kubo, K.; Tobise, M. Two-degree-of-freedom robust speed controller for high-performance rolling mill drives. *IEEE Trans. Ind. Appl.* **1993**, *29*, 919–926. [CrossRef]
38. Boukhezzar, B.; Siguerdidjane, H. Nonlinear Control of a Variable-Speed Wind Turbine Using a Two-Mass Model. *IEEE Trans. Energy Convers.* **2011**, *26*, 149–162. [CrossRef]

39. Kabziński, J.; Mosiołek, P. Integrated, Multi-Approach, Adaptive Control of Two-Mass Drive with Nonlinear Damping and Stiffness. *Energies* **2021**, *14*, 5475. [CrossRef]
40. Zhou, F. Drive-train torsional vibration suppression of large scale PMSG-based WECS. *Prot. Control Mod. Power Syst.* **2022**, *7*, 37. [CrossRef]
41. LeCun, Y.A.; Bottou, L.; Orr, G.B.; Müller, K.R. Efficient BackProp. In *Neural Networks: Tricks of the Trade: Second Edition*; Montavon, G., Orr, G.B., Müller, K.R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 9–48. [CrossRef]
42. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer Series in Operations Research and Financial Engineering; Springer: New York, NY, USA, 2006; Volume XXII, 664p. [CrossRef]
43. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
44. Kennedy, J. Swarm intelligence. In *Handbook of Nature-Inspired and Innovative Computing*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 187–219.
45. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 69–73. [CrossRef]
46. Eberhart, S.Y. Particle swarm optimization: Developments, applications and resources. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), Seoul, Republic of Korea, 27–30 May 2001; Volume 1, pp. 81–86. [CrossRef]
47. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
48. Bonyadi, M.R.; Michalewicz, Z. Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. *Evol. Comput.* **2017**, *25*, 1–54. [CrossRef]
49. Yang, X.S.; Deb, S. Cuckoo Search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214. [CrossRef]
50. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 10031–10061. [CrossRef]
51. Yang, X.S.; Deb, S. Multiobjective cuckoo search for design optimization. *Comput. Oper. Res.* **2013**, *40*, 1616–1624. [CrossRef]
52. Shi, Y.; Eberhart, R. Empirical study of particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 3, pp. 1945–1950. [CrossRef]
53. Clerc, M.; Kennedy, J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [CrossRef]
54. Nocedal, J.; Wright, S.J. Fundamentals of unconstrained optimization. In *Numerical Optimization*; Springer International Publishing: Cham, Switzerland, 2006; pp. 10–29.
55. Nocedal, J.; Wright, S.J. Quasi-newton methods. In *Numerical Optimization*; Springer International Publishing: Cham, Switzerland, 2006; pp. 135–163.
56. Arora, J. *Introduction to Optimum Design*; Elsevier: Amsterdam, The Netherlands, 2004.
57. Zhao, S.Z.; Liang, J.J.; Suganthan, P.N.; Tasgetiren, M.F. Dynamic multi-swarm particle swarm optimizer with local search for Large Scale Global Optimization. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 3845–3852. [CrossRef]
58. MathWorks Inc. *MATLAB*; MathWorks Inc.: Natick, MA, USA, 2023.
59. Arora, J.S. 7—More on Linear Programming Methods for Optimum Design. In *Introduction to Optimum Design*, 2nd ed.; Arora, J.S., Ed.; Academic Press: San Diego, CA, USA, 2004; pp. 259–275. [CrossRef]
60. Han, F.; Liu, Q. An Improved Hybrid PSO Based on ARPSO and the Quasi-Newton Method. In *Advances in Swarm and Computational Intelligence*; Tan, Y., Shi, Y., Buarque, F., Gelbukh, A., Das, S., Engelbrecht, A., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 460–467.
61. Noel, M.M. A new gradient based particle swarm optimization algorithm for accurate computation of global minimum. *Appl. Soft Comput.* **2012**, *12*, 353–359. [CrossRef]
62. Kelly, R.; Llamas, J.; Campa, R. A measurement procedure for viscous and coulomb friction. *IEEE Trans. Instrum. Meas.* **2000**, *49*, 857–861. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.