

Article

Recursive Identification for MIMO Fractional-Order Hammerstein Model Based on AIAGS

Qibing Jin, Bin Wang * and Zeyu Wang

Institute of Automation, Beijing University of Chemical Technology, Beijing 100020, China; jinqb@mail.buct.edu.cn (Q.J.); wangzeyu@buct.edu.cn (Z.W.)

* Correspondence: 2019210478@buct.edu.cn

Abstract: In this paper, adaptive immune algorithm based on a global search strategy (AIAGS) and auxiliary model recursive least square method (AMRLS) are used to identify the multiple-input multiple-output fractional-order Hammerstein model. The model's nonlinear parameters, linear parameters, and fractional order are unknown. The identification step is to use AIAGS to find the initial values of model coefficients and order at first, then bring the initial values into AMRLS to identify the coefficients and order of the model in turn. The expression of the linear block is the transfer function of the differential equation. By changing the stimulation function of the original algorithm, adopting the global search strategy before the local search strategy in the mutation operation, and adopting the parallel mechanism, AIAGS further strengthens the original algorithm's optimization ability. The experimental results show that the proposed method is effective.

Keywords: adaptive immune algorithm; multiple-input multiple-output; fractional-order model; Hammerstein model; system identification



Citation: Jin, Q.; Wang, B.; Wang, Z. Recursive Identification for MIMO Fractional-Order Hammerstein Model Based on AIAGS. *Mathematics* **2022**, *10*, 212. <https://doi.org/10.3390/math10020212>

Academic Editors: António M. Lopes, Alireza Alfi, Liping Chen and Sergio A. David

Received: 30 November 2021

Accepted: 9 January 2022

Published: 11 January 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 recent years, with the rapid economic and social development, the complexity of industry has been increasing. In order to understand and control these industrial processes more accurately, it is necessary to study system identification. However, in real life, nonlinear processes are inevitable and widespread. Nowadays, there is no definite characterization for nonlinear processes. A block-oriented model is a description of nonlinear model, which is the result of the interaction between the dynamic linear module and static nonlinear module. These model components can be connected in series, parallel, or feedback [1]. Hammerstein model is a typical block-oriented model that consists of a static nonlinear block in cascade with a dynamic linear block [2]. Because the dynamic behavior of the model is only included in the linear block, and the nonlinear block is static, this feature is conducive to identifying and controlling the nonlinear system constructed by the Hammerstein model [3]. Hammerstein model is extensively used to identify nonlinear systems [4–7]. As the model is widely used, the identification methods are also intensively discussed. These methods include neural networks [8,9], piecewise linear model [6], least square method [10], support vector machine [11], combined prior information [12], and so on.

In real life, it is evident that the dynamic linear block based on integer order cannot fully simulate the real model [13]. The fractional-order model extends the order of the model from the integer level to the fractional level. Therefore, the study of the fractional-order nonlinear model is essential [14]. At present, fractional-order models have been discussed in many fields, such as molecular materials [15,16], the voltage and current of the drive end impedance [17], industrial battery [18–20], and so on.

With the wide application of the fractional-order model, the problem of model identification has also been intensively discussed. However, the current methods have some

limitations. The particle swarm optimization algorithm can be used to identify the parameters of the fractional Hammerstein model [21]. This method excessively depends on the optimization ability of the algorithm and does not consider the internal relationship between system parameters. Once the optimization algorithm has problems, it will significantly impact the identification results. The Levenberg–Marquardt algorithm developed by combining the two decomposition principles [22] can only be applied to the theoretical environment. Once the system is affected by noise, the model's parameters will not be identified exactly. Reference [23] also requires an ideal environment. Some scholars pay attention to the fractional-order Hammerstein model with single-input single-output [24–28]. Some pay attention to the fractional-order Hammerstein model with multiple-input multiple-output, but most use the state space equation as the linear block of the model [29,30]. However, fractional-order calculus is a whole concept [31]. Using the transfer function of differential equation to construct the linear block of the Hammerstein model can better integrate the two concepts.

Based on the above background, this paper discusses a new method to identify the nonlinear coefficients, linear coefficients, and fractional order of the MIMO fractional Hammerstein model. In this method, AIAGS greatly improves the optimization ability by improving the immune algorithm's stimulation function and search strategy. Then, the algorithm estimates the initial values of all MIMO fractional Hammerstein model parameters, including fractional order. The estimated result provides relatively accurate initial values for the subsequent algorithm. It solves the problem that the two-step method [28], which identifies coefficient and order, depends on the initial values. Then, using AMRLS, a method for accurate parameter identification of the MIMO fractional-order model is proposed. Finally, the effectiveness of the proposed method is verified by numerical simulation.

The main contribution of this paper is to propose an adaptive immune algorithm with a global search strategy to accurately identify the initial parameters of the fractional Hammerstein system. Secondly, a new recursive identification method for coefficients and fractional order of MIMO fractional-order nonlinear system with differential equation transfer function as linear block model is derived using an auxiliary model. Due to the different ways of selecting the optimal solution, the AIAGS algorithm proposed in this paper has higher reliability than the classical immune algorithm. Based on the auxiliary model, the recursive identification algorithm for the MIMO fractional Hammerstein model is given using the recursive least square method. The method in this paper solves the initial value problem of previous methods and provides more accurate initial values. This initial value cooperates with AMRLS, making the result of parameters identification of multi-input and multi-output fractional Hammerstein model closer to reality.

In this paper, an improved immune algorithm is proposed in Section 2. In Section 3, a new recursive identification method for MIMO fractional-order Hammerstein model with differential equation transfer function as linear block model is derived by using auxiliary model is discussed. In Section 4, numerical simulations show the effectiveness of the proposed method. Finally, Section 5 gives some conclusions.

2. Adaptive Immune Algorithm Based on Global Search Strategy

2.1. Review of Immune Algorithms

The immune algorithm is an adaptive intelligent system inspired by immunology and simulates the functions and principles of the biological immune system to solve complex problems. It retains several characteristics of the biological immune system, including global search capability, diversity maintenance mechanism, strong robustness, and parallel distributed search mechanism. The immune algorithm automatically generates the initial population by uniform probability distribution. After initialization, the population evolves and improves by the following steps: calculation of stimulation, selection, cloning, mutation, clonal inhibition, etc. [32].

2.2. AIAGS

2.2.1. Stimulation Improvement

Individual stimulation is the evaluation result of individual quality, which needs to be comprehensively considered individual affinity and concentration. The individual stimulation can usually be obtained by a simple mathematical calculation based on the evaluation results of individual affinity and concentration. In the traditional immune algorithm [33], the stimulation is expressed as

$$f_{sim}(x_i) = a \cdot f_{aff}(x_i) - b \cdot f_{den}(x_i) \tag{1}$$

where x_i means the i th individual of the population; $f_{aff}(x_i)$ is affinity, which represents the Euclidean distance between the current individual and the optimal individual; $f_{den}(x_i)$ is the concentration, indicating the number of other individuals whose Euclidean distance between the current individual and other individuals is within a certain threshold; $f_{sim}(x_i)$ is the stimulation; a and b is the calculation parameter. The algorithm will sort the individuals according to the stimulation and make the next choice.

This paper made the following changes to the coefficients of affinity and concentration. Firstly, the minus sign of Equation (1) is changed on the plus sign. Because the concentration represents the quality of population diversity, and too high concentration means that there are many very similar individuals in the population, the key point of the immune algorithm is to suppress the individuals with a high concentration to achieve global optimization. However, in both the original algorithm and various improved immune algorithms today, the coefficient b is non-negative, which leads to a minor incentive for individuals with low affinity and high concentration [34–38]. This improvement conforms to the core concept of the algorithm.

Secondly, this paper designs a parameter β related to the current population’s maximum, minimum, and individual affinity values. In the original algorithm, the a and b are constants. In various improved algorithms [34–38], the adaptive coefficients are only related to the number of current iterations. Because the comparison of stimulations between individuals is carried out in the population of the current iteration, these adaptive coefficients are not different from constants. They will not affect the stimulation ranking of the population. In this paper, because β is quadratic when selecting individuals based on stimulations, individuals with low affinity and individuals with high affinity will be considered, increasing the global searchability. The parameter is expressed as

$$\beta = \left(\frac{f_{aff}(x_i) - f_{aff_a}}{f_{aff_{max}} - f_{aff_a}} \right)^2 \tag{2}$$

where f_{aff_a} is the average of $f_{aff_{max}}$ and $f_{aff_{min}}$.

Finally, after a certain number of iterations, the population will move closer to the optimal global individual. If the concentration problem is also considered, it may give up the found optimal range and select the new random individual when selecting the individuals. Therefore, a monotone decreasing adaptive operator is designed in this paper. In the middle and later iteration stages, the concentration effect is negligible.

To sum up, the stimulation for this paper is expressed as

$$f_{sim}(x_i) = (1 - \beta) \cdot f_{aff}(x_i) + \left[1 - \sqrt{\frac{2gen}{G} - \left(\frac{gen}{G}\right)^2} \right] \cdot 0.5\beta \cdot f_{den}(x_i) \tag{3}$$

where gen means the current number of iterations and G is the total number of iterations.

After improvement, the approximate trend of individual stimulations is shown in Figure 1a. The approximate trend of the stimulations of the original or other improved immune algorithm is shown in Figure 1b. The x -axis is 100 individuals sorted from smallest to largest according to affinity, and the y -axis is individual stimulation. It can be seen from Figure 1 that the original algorithm and other improved algorithms generally only

select individuals with low affinity. In contrast, the algorithm in this paper can consider individuals with high affinity.

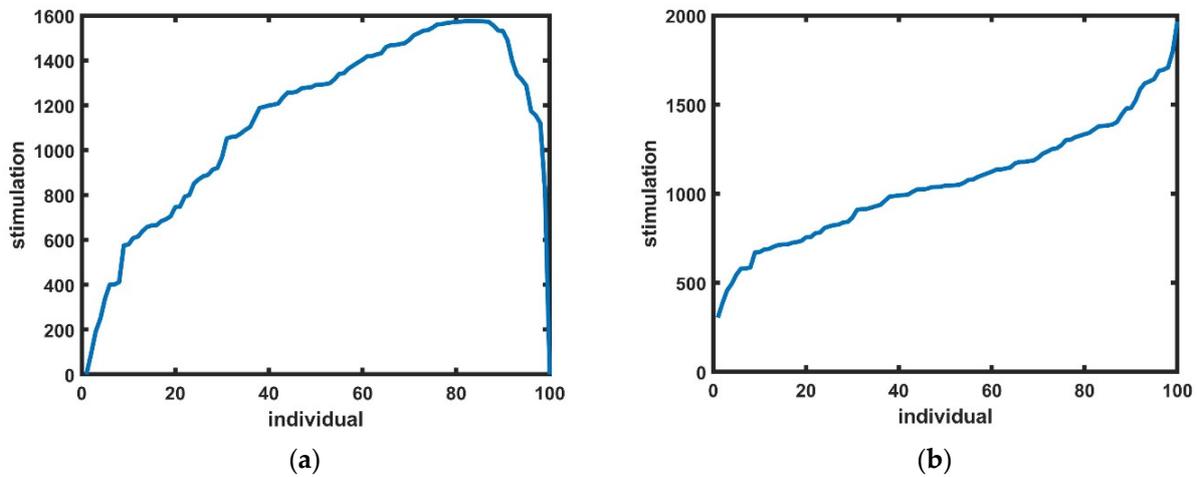


Figure 1. Comparison of stimulations. (a) The stimulations of AIAGS. (b) The stimulations of other algorithms.

2.2.2. Mutation Strategy Improvement

The original algorithm has a single strategy in the mutation stage. The algorithm improved by others will enrich the mutation strategy and improve the probability of all individuals for mutation. However, the mutation strategy is selected only by random numbers, which makes the algorithm not flexible [38].

The algorithm of this paper has two minor changes in the mutation stage. First, an adaptive operator pm that changes from algebra is designed, and its value decreases monotonically between 0 and 0.8. The parameter can be expressed as

$$pm = 0.8 \cdot \left(1 - \frac{gen}{G}\right) \tag{4}$$

Secondly, when setting the global optimization step, a variable sv is added based on adaptation, gradually changing the mutation step. The optimal individual is selected for retention of the individuals after several mutations, which greatly enhances the global search ability.

To sum up, the mutation strategy for this paper can be expressed as

$$x_{i,j} = \begin{cases} x_{best,j} + pm \cdot (x_{r1,j} - x_{r2,j}), & rand > pm \\ x_{r1,j} + (pm + sv) \cdot (x_{r2,j} - x_{r3,j}), & otherwise \end{cases} \tag{5}$$

where i means the sequence of individuals in the population; j denotes the sequence of dimensions in the individual; x_{r1} , x_{r2} , and x_{r3} are different individuals randomly selected from the population except for the x_i .

Obviously, in the early stage of the iteration, the mutation strategy will mostly choose the second mutation strategy, edge mutation strategy, which will enhance the global optimization ability of the algorithm. In the middle and later stages of the iteration, the first mutation strategy, the optimal individual mutation strategy, will be selected for local search.

2.2.3. Simulated Annealing Strategy

The simulated annealing algorithm mimics the annealing process in metallurgy and is classified as a single-based solution method. After comparing the current optimal solution with the previous optimal solution, if the fitness of the current optimal solution is greater than that of the previous one, it may abandon the current result and choose the previous result [39].

At the end of the improved algorithm, simulated annealing is added to avoid the algorithm falling into the local optimum. Some people have done similar work, but both the initial algorithm and others' improved algorithm use stimulus to evaluate the optimal solution [33]. This paper uses affinity to evaluate the optimal solution at the end. However, the stimulation of the optimal individual of the previous generation may be slightly big, resulting in not being selected during mutation selection, so the affinity of the optimal individual of the current generation may be greater than that of the previous generation. At this time, the effect of simulated annealing will likely jump back to the result to optimize further. The replacement for such a case depends on the probability p as defined as

$$p = e^{-\Delta F} \Delta F = \frac{f_{aff}(x_i')}{f_{aff}(x_i)} - 1 \tag{6}$$

where x_i' is the current optimal solution; x_i denotes the previous optimal solution. This part will replace the solution if $p < rand(0, 1)$.

2.2.4. Pseudo Code of AIAGS

To sum up, there are some innovations of this paper on the existing immune algorithms. The pseudo code of AIAGS is explained in detail in Algorithm 1. The flowchart of AIAGS is explained in detail in Figure 2.

Algorithm 1: AIAGS

- Step.1** Define the objective function $F(x)$;
 - Step.2** Initialize population X ;
 - Step.3** Evaluate all the individuals x_i by the objective function $F(x)$;
 - Step.4** Calculate the affinity $f_{aff}(x_i)$ and concentration $f_{den}(x_i)$ of each individual;
 - Step.5** Initialize the number of iteration $m = 1$;
 - Step.6** While $m < \text{max number of iterations } M$;
 - Step.7** Calculate the stimulation $f_{sim}(x_i)$ of each individual by the Equation (3);
 - Step.8** Select the individuals in the population by stimulation and clone the individuals;
 - Step.9** Mutate the cloned individuals by the Equation (5);
 - Step.10** If the generated mutation vector exceeds the boundary, a new mutation vector is generated randomly until it is within the boundary;
 - Step.11** Inhibit cloning and calculate the affinity of each new individual;
 - Step.12** Generate optimal individual by Simulated Annealing by the Equation (6);
 - Step.13** End;
 - Step.14** $m = m + 1$;
 - Step.15** End while;
 - Step.16** Return the best solution.
-

2.3. Benchmark Function

Due to the limitations of intelligent optimization algorithms, unlike the traditional algorithm, which has a mathematical theoretical basis, it is not strict. After improving the optimization algorithm, most people use the classical benchmark function to test the algorithm's effectiveness. This article uses eight classical and four CEC2017 benchmark functions to evaluate AIAGS. The $u()$ of F6 and F7 is expressed as

$$u(x_i, a, k, m) = \begin{cases} K(x_i - a)^m, & \text{if } x_i > a \\ 0, & -a \leq x_i \leq a \\ K(-x_i - a)^m, & -a \leq x_i \end{cases} \tag{7}$$

These classical functions are divided into three groups: unimodal (F1–F4), multimodal (F5–F7), and fixed-dimension multimodal (F8). The unimodal benchmark function has only one optimal solution, which can verify the development and convergence. The multimodal benchmark function has many optimal solutions. However, there is only one global optimal

solution, and the rest are local optimal solutions. The fixed dimensional multimodal functions can define the desired number of design variables and could provide a different search space. Therefore, the multimodal functions are responsible for testing exploration and avoiding the entrapment in the optimal local solution. Hybrid and composition functions can reflect some problems that are closer to reality [40]. In Table 1, the corresponding properties of these functions are listed, where dim represents the dimensions of the functions and range indicates the scope of the search space.

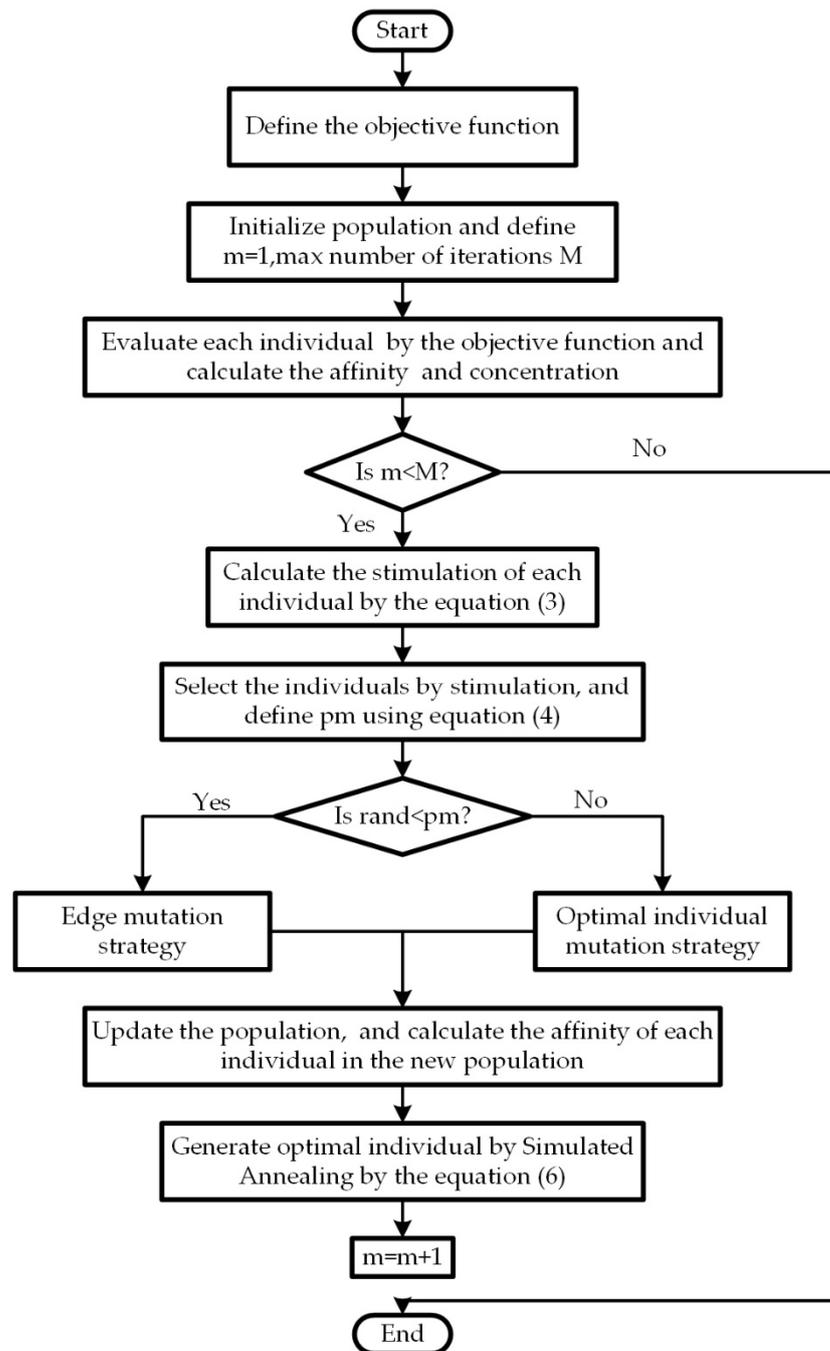


Figure 2. AIAGS.

Table 1. Benchmark functions.

Name	Formula	Range	f_{min}
Sphere	$F1(x) = \sum_{i=1}^D x_i^2$	[-20, 20]	0
Schwefel 1.2	$F2(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j\right)^2$	[-100, 100]	0
Rosenbrock	$F3(x) = \sum_{i=1}^{D-1} [100 \cdot (x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	[-30, 30]	0
Step	$F4(x) = \sum_{i=1}^D (x_i + 0.5)^2$	[-100, 100]	0
Ackley	$F5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left[\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right] + 20 + e$	[-40, 40]	0
Generalized penalized 1	$F6(x) = \frac{\pi}{n} [10 \sin(\pi y_1)] + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + \sum_{i=1}^D u(x_i, 10, 100, 4), y_i = 1 + \frac{x_i + 1}{4}$	[-50, 50]	0
Generalized penalized 2	$F7(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^D (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	[-50, 50]	0
Shekel's Foxholes	$F8(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})} \right]^{-1}$	[-70, 70]	1
Hybrid function 4 (N = 4)	F9(x)	[-100, 100]	1400
Hybrid function 7 (N = 5)	F10(x)	[-100, 100]	1700
Composition function 1 (N = 3)	F11(x)	[-100, 100]	2100
Composition function 4 (N = 4)	F12(x)	[-100, 100]	2400

2.3.1. Comparison of AIAGS with Other Algorithms

In order to reflect the improvement effect of the immune algorithm in this paper, this section compares AIAGS with the original immune algorithm two improved immune algorithms: improved artificial immune algorithm (IAIA) [28] and modified artificial immune algorithm (MAIA) [29], and two new algorithms: Harris hawks optimization (HHO) [41] and Aquila optimizer (AO) [42]. The parameter settings of the counterparts' algorithms are given in Table 2. The comparison results are shown in Table 3. However, intelligent algorithms are highly accidental. After several tests, this paper calculates the average value and standard deviation of each test result to avoid misleading the experimental results and the practical application of the algorithm.

Table 2. Parameter settings.

Algorithm	Parameter Settings
AIAGS	$\delta = 0.1, sv = 0.2$
AO	$\alpha = 0.5, \delta = 0.5$
IA	$\alpha = 2, \beta = 1, \delta = 0.2, pm = 0.7$
IAIA	$\alpha = 2, \beta = 1, \delta = 0.613, pm = 0.7$
MAIA	$\delta = 0.8, pm = 0.8, cr = 0.8$
HHO	$\alpha = 0.5, \delta = 0.5$

Table 3. Comparison of results obtained for the benchmark functions.

	AIAGS	AO	IA	IAIA	MAIA	HHO
F1						
worst	0	2.86×10^{-71}	0.000124	0.000145	0.030882	1.98×10^{-46}
best	0	7.37×10^{-76}	7.65×10^{-5}	3.54×10^{-5}	0.001683	2.62×10^{-58}
Avg	0	5.74×10^{-72}	9.86×10^{-5}	7.71×10^{-5}	0.012509	1.99×10^{-47}
Std	0	1×10^{-71}	1.46×10^{-5}	3.28×10^{-5}	0.009914	5.95×10^{-47}
F2						
worst	0	2.82×10^{-56}	0.006578	0.022761	16.07011	1.71×10^{-42}
best	0	1.72×10^{-73}	0.002606	0.013182	0.812125	1.15×10^{-51}

Table 3. Cont.

	AIAGS	AO	IA	IAIA	MAIA	HHO
Avg	0	2.82×10^{-57}	0.003962	0.017273	4.565824	3.78×10^{-43}
Std	0	8.93×10^{-57}	0.001401	0.003087	4.947259	6.69×10^{-43}
F3						
worst	6.39×10^{-7}	0.001305	433.5283	696.2436	83.41411	0.008889
Best	5.5×10^{-9}	5×10^{-6}	0.99727	0.762353	4.4702	2.1×10^{-5}
Avg	9.99×10^{-8}	0.000319	80.76008	143.8289	29.75245	0.002238
Std	1.83×10^{-7}	0.000424	143.8194	240.8117	30.68641	0.002581
F4						
worst	0	6.97×10^{-5}	0.004139	0.00329	0.00329	9.33×10^{-5}
Best	0	2.3×10^{-7}	0.001612	0.00174	0.00174	7.93×10^{-10}
Avg	0	1.87×10^{-5}	0.003066	0.002567	0.002567	2.05×10^{-5}
Std	0	2.32×10^{-5}	0.00077	0.00053	0.00053	2.64×10^{-5}
F5						
worst	8.88×10^{-16}	8.88×10^{-16}	4.663342	3.223428	1.019824	8.88×10^{-16}
Best	8.88×10^{-16}	8.88×10^{-16}	0.017455	0.019081	0.137416	8.88×10^{-16}
Avg	8.88×10^{-16}	8.88×10^{-16}	1.139553	0.342006	0.437464	8.88×10^{-16}
Std	0	0	1.617355	1.012431	0.323219	0
F6						
worst	4.71×10^{-32}	3.84×10^{-5}	4.772913	6.250579	0.005788	2.07×10^{-5}
Best	4.71×10^{-32}	7.83×10^{-8}	1.16×10^{-5}	0.335882	0.000107	1.56×10^{-7}
Avg	4.71×10^{-32}	7.48×10^{-6}	1.984778	3.781554	0.001743	6.34×10^{-6}
Std	0	1.16×10^{-5}	1.830602	2.512286	0.002048	6.86×10^{-6}
F7						
worst	1.35×10^{-32}	0.000281	0.000101	8.19×10^{-5}	0.039677	0.000501
best	1.35×10^{-32}	1.31×10^{-6}	5.21×10^{-5}	3.87×10^{-5}	0.002672	1.18×10^{-7}
Avg	1.35×10^{-32}	4.25×10^{-5}	8.01×10^{-5}	5.89×10^{-5}	0.017996	8.5×10^{-5}
Std	2.88×10^{-48}	8.69×10^{-5}	1.55×10^{-5}	1.58×10^{-5}	0.01293	0.000143
F8						
worst	0.998004	2.982105	1.992031	0.998004	0.999027	1.992031
best	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004
Avg	0.998004	1.593234	1.166875	0.998004	0.998107	1.196819
Std	2.34×10^{-16}	0.958412	0.362935	2.01×10^{-15}	0.000323	0.397606
F9						
worst	1528.366	5142.015	2215.496	2302.871	5755.439	4349.2
best	1472.889	1557.776	1443.205	1428.962	1488.148	1450.039
Avg	1503.786	2462.484	1580.193	1655.434	2510.73	1833.8
Std	19.26844	978.4552	223.2629	287.2931	1264.939	843.7423
F10						
worst	1794.68	1838.131	1763.443	1782.14	2200.955	1840.59
Best	1744.138	1731.296	1722.813	1725.397	1766.414	1744.772
Avg	1774.579	1781.842	1738.947	1748.674	1898.936	1781.998
Std	17.10128	32.03933	10.8574	22.15373	122.3724	30.2191
F11						
worst	2260.104	2338.993	2264.487	2288.434	2319.733	2388.341
Best	2209.787	2204.09	2200.005	2200.003	2201.822	2205.34
Avg	2236.802	2272.26	2211.444	2211.249	2265.511	2272.888
Std	18.17673	56.04231	18.0651	25.8142	44.29724	71.44948
F12						
worst	2717.367	2778.692	2772.984	2762.261	2824.593	2857.503
Best	2521.748	2746.416	2500.074	2500.073	2505.906	2770.847
Avg	2626.946	2767.838	2669.676	2629.372	2710.07	2799.953
Std	61.88762	9.524766	114.739	111.2591	104.6337	28.32715

2.3.2. Convergence

Convergence is the ability of the algorithm to search and converge to an acceptable solution in a certain time. Convergence is an important index to evaluate the performance of the algorithm. An algorithm has high convergence, which means fast optimization speed and high precision. Generally, the convergence speed can be measured by the number of iterations, and the convergence value can measure the accuracy.

The convergence curves of AIAGS and the other five algorithms in 12 benchmark functions are shown in Figure 3. It can be seen from Table 2 and Figure 3 that the convergence speed and optimization ability of AIAGS are not the strongest in individual benchmark functions. On the whole, AIAGS is far better than other immune algorithms in terms of convergence speed and optimization ability, and it is also better than the other two algorithms.

2.4. Summary

In this chapter, the immune algorithm’s stimulation function and mutation strategy are improved, and simulated annealing is added to the final step to select the optimal solution. The core idea of these improvements is to avoid finding the optimal local solution. After improving the algorithm, 12 different types of benchmark functions are used to evaluate the algorithm’s performance. Experiments show that the development and exploration ability of AIAGS is significantly improved compared with the previous immune algorithm. These conclusions provide substantial proof for the following system identification work.

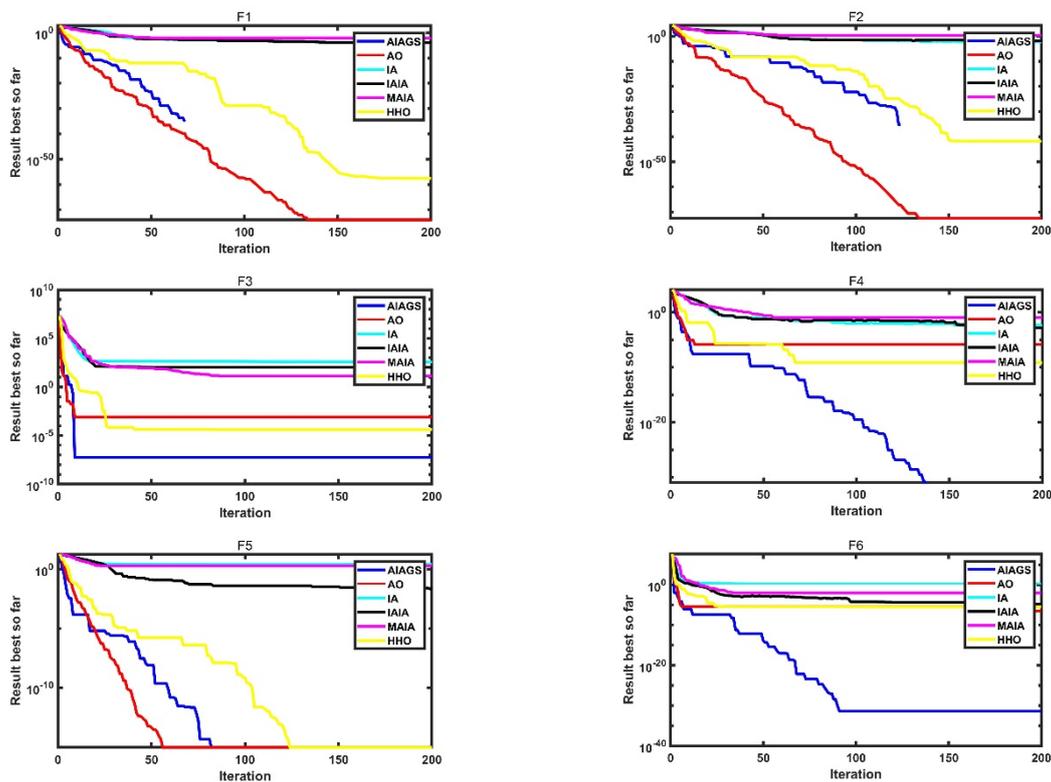


Figure 3. Cont.

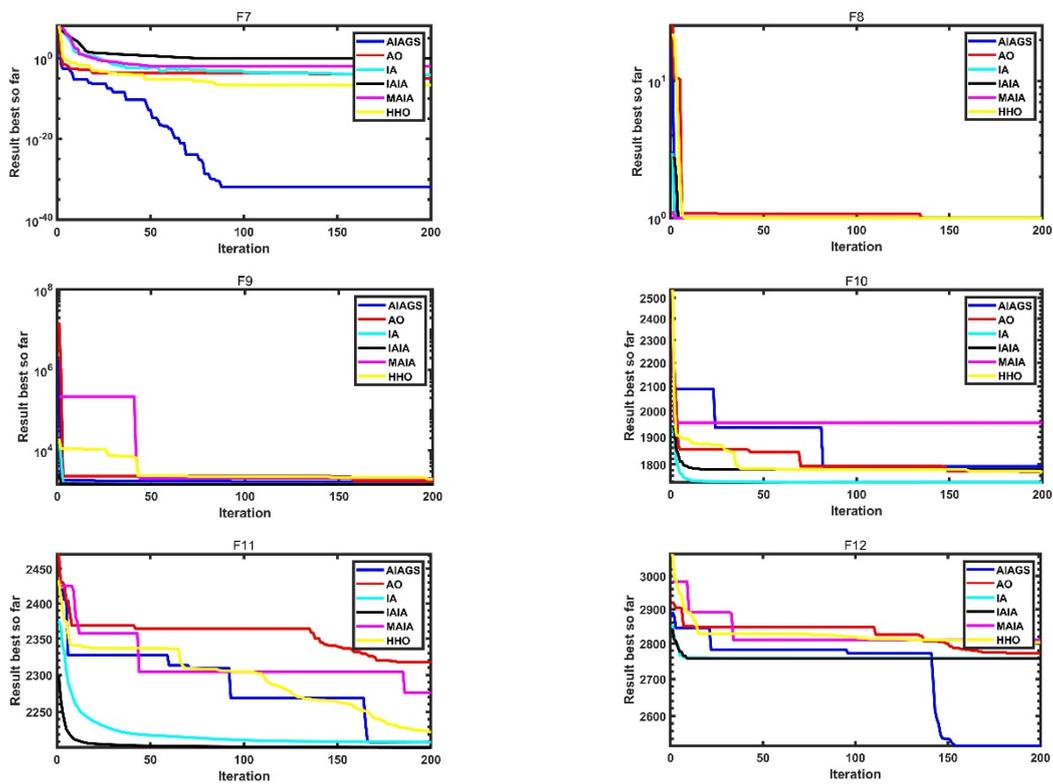


Figure 3. The convergence curves of AIAGS and other six algorithms.

3. Identification Method of MIMO Fractional Order Hammerstein Model

3.1. MIMO Fractional Order Hammerstein Model

3.1.1. Fractional Order Differentiation

At present, there are three definitions widely used in the field of fractional calculus: Grünwald–Letnikov (GL), Riemann–Liouville (RL), and Caputo definitions. Because the GL is easy to program [43], this paper considers it the research object. The definition of fractional order calculus can be expressed as

$$D_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{\lceil \frac{t-t_0}{h} \rceil} (-1)^j \binom{\alpha}{j} f(t - jh) \tag{8}$$

where α is the fractional order. Because this paper explores differential equations, $\alpha > 0$. h is the sampling time; $\lceil \cdot \rceil$ means that the integer part is reserved; $(-1)^j \binom{\alpha}{j}$ is the binomials of $(1 - z)^\alpha$. By denoting w_j^α to replace the binomials, so w_j^α can be expressed as

$$w_j^\alpha = (-1)^j \binom{\alpha}{j} = \frac{(-1)^j \Gamma(\alpha + 1)}{\Gamma(j + 1) \Gamma(\alpha - j + 1)} \tag{9}$$

Finally, when $t_0 = 0$, the definition of fractional order calculus can be expressed as

$$D_t^\alpha f(t) = \frac{1}{h^\alpha} \sum_{j=0}^{\lceil \frac{t-a}{h} \rceil} w_j^\alpha f(t - jh) \tag{10}$$

3.1.2. MIMO Fractional-Order Hammerstein System

The MIMO Hammerstein model of this paper can be schematically represented in Figure 4. Hammerstein model is a typical nonlinear model composed of static nonlinear block and dynamic linear block. the dynamic linear block can be expressed as

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_N(t) \end{bmatrix} = \begin{bmatrix} G_{1,1} & G_{1,2} & \dots & G_{1,M} \\ G_{2,1} & G_{2,2} & \dots & G_{2,M} \\ \vdots & & & \\ G_{N,1} & G_{N,2} & \dots & G_{N,M} \end{bmatrix} \begin{bmatrix} u'_1(t) \\ u'_2(t) \\ \vdots \\ u'_M(t) \end{bmatrix} \tag{11}$$

where $y_k(t)$ is the k th system output; $u'_l(t)$ is generated by the l th system input $u_l(t)$ through the nonlinear block, which can be expressed as

$$\begin{aligned} u'_l(t) &= c_{l,1} \cdot f_{l,1}(u_l(t)) + c_{l,2} \cdot f_{l,2}(u_l(t)) + \dots + c_{l,n_{lc}} \cdot f_{l,n_{lc}}(u_l(t)) \\ &= \sum_{m=1}^{n_{lc}} c_{l,m} \cdot f_{l,m}(u_l(t)) \end{aligned} \tag{12}$$

where $c_{l, \cdot}$ are coefficients to be identified; $f_{l, \cdot}()$ are a series of basic functions. $G_{k,l}$ is a fractional-order transfer function, which can reflect the relationship between $u'_l(t)$ and $y_k(t)$; it is defined as

$$G_{k,l}(s) = \frac{b_{k,l,m} s^{m\alpha} + b_{k,l,m-1} s^{(m-1)\alpha} + \dots + b_{k,l,0}}{a_{k,l,n} s^{n\alpha} + a_{k,l,n-1} s^{(n-1)\alpha} + \dots + a_{k,l,0}} \tag{13}$$

where $a_{k,l, \cdot}$ and $b_{k,l, \cdot}$ are coefficients to be identified; α is the fractional order to be identified. For the convenience of calculation and programming, in this paper $a_{k,l,0}$ is assumed to be 1. According to Equations (11) and (13), the k th system output can be expressed as

$$\begin{aligned} y_k &= G_{k,1}u'_1 + G_{k,2}u'_2 + \dots + G_{k,M}u'_M \\ &= \frac{b_{k,1,m}S^{m\alpha} + b_{k,1,m-1}S^{(m-1)\alpha} + \dots + b_{k,1,0}}{a_{k,1,n}S^{n\alpha} + a_{k,1,n-1}S^{(n-1)\alpha} + \dots + a_{k,1,1}S^{\alpha} + 1} u'_1 \\ &\quad + \frac{b_{k,2,m}S^{m\alpha} + b_{k,2,m-1}S^{(m-1)\alpha} + \dots + b_{k,2,0}}{a_{k,2,n}S^{n\alpha} + a_{k,2,n-1}S^{(n-1)\alpha} + \dots + a_{k,2,1}S^{\alpha} + 1} u'_2 \\ &\quad + \dots + \frac{b_{k,M,m}S^{m\alpha} + b_{k,M,m-1}S^{(m-1)\alpha} + \dots + b_{k,M,0}}{a_{k,M,n}S^{n\alpha} + a_{k,M,n-1}S^{(n-1)\alpha} + \dots + a_{k,M,1}S^{\alpha} + 1} u'_M \end{aligned} \tag{14}$$

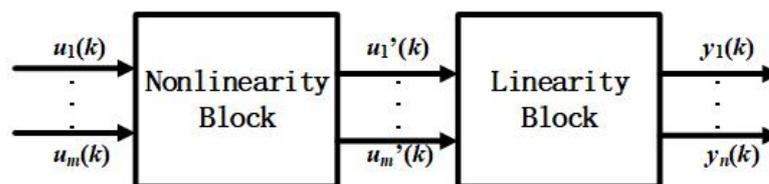


Figure 4. MIMO Hammerstein model.

By reduction of fractions to a common denominator and simplifying Equation (14), we can get an equation described as

$$\begin{aligned} &\left(A_{k,N_A} S^{N_A\alpha} + A_{k,N_A-1} S^{(N_A-1)\alpha} + \dots + A_{k,1} S^{\alpha} + 1 \right) y_k \\ &= \left(B_{k,1,N_B} S^{N_B\alpha} + B_{k,1,N_B-1} S^{(N_B-1)\alpha} + \dots + B_{k,1,0} \right) u'_1 + \\ &\quad \left(B_{k,2,N_B} S^{N_B\alpha} + B_{k,2,N_B-1} S^{(N_B-1)\alpha} + \dots + B_{k,2,0} \right) u'_2 + \dots + \\ &\quad \left(B_{k,M,N_B} S^{N_B\alpha} + B_{k,M,N_B-1} S^{(N_B-1)\alpha} + \dots + B_{k,M,0} \right) u'_M \end{aligned} \tag{15}$$

where A is the polynomial containing a ; B is the polynomial containing a and b ; the coefficient of fractional order $N_A = M * n$, $N_B = (M - 1) * n + m$. To sum up, the MIMO fractional-order Hammerstein system discussed in this paper can be expressed as

$$\begin{cases} y_k(t) + \sum_{i=1}^{n_A} A_{k,i} D^{i\alpha} y_k(t) = \sum_{l=0}^M \sum_{j=0}^{n_B} B_{k,l,j} D^{j\alpha} u_l'(t) \\ y_k'(t) = y_k(t) + v(t) \end{cases} \tag{16}$$

where $v(t)$ is the Gaussian white noise; $y_k'(t)$ is the measured output containing noise. According to Equations (10) and (16), the MIMO fractional-order Hammerstein system can be expressed as

$$y_k'(t) = \frac{1}{(1 + \sum_{i=1}^{n_A} A_i / h^{i\alpha})} \cdot \left[\sum_{l=0}^M \sum_{i=0}^{n_B} \sum_{m=1}^{n_{lc}} \frac{B_{k,l,i}}{h^{i\alpha}} c_{l,m} \cdot \sum_{j=0}^{[t/h]} w_j^{i\alpha} f_{l,m}(u_l(t - jh)) - \sum_{i=1}^{n_A} \frac{A_i}{h^{i\alpha}} \sum_{j=1}^{[t/h]} w_j^{i\alpha} y_k(t - jh) \right] + v(t) \tag{17}$$

3.2. Parameter Identification Based on Auxiliary Model Recursive Least Square Method

In the MIMO fractional-order Hammerstein model, all the coefficients and the fractional order are needed to be identified. Previous articles usually considered only part of coefficients or for the SISO system. The work of this paper is rarely concerned before. The identification work is divided into coefficient identification and order identification. However, the two results affect each other, which cannot identify coefficients precisely without a precise fractional order. This paper will first use a series of input and output data to obtain the initial values of coefficients and the fractional order by the AIAGS algorithm mentioned above. The initial value is a little precise. Then, the initial value will be used to get the parameter identification result of the fractional-order Hammerstein model through the auxiliary model recursive least squares (AMRLS) algorithm.

3.2.1. Coefficient Identification

According to the basic knowledge of system identification, the input–output relations can be expressed as

$$y_k'(t) = y_k(t) + v(t) = \varnothing_k(t) \cdot \theta_k^T + v(t) \tag{18}$$

where $\varnothing_k(t)$ is the variable vector including input–output data, which is expressed as

$$\begin{aligned} \varnothing_k(t) = & \left[\varnothing_{k,A}(t), \varnothing_{B_{k,1,0}}(t), \varnothing_{B_{k,1,1}}(t), \dots, \varnothing_{B_{k,1,n_B}}(t), \dots, \varnothing_{B_{k,M,0}}(t), \varnothing_{B_{k,M,1}}(t), \dots, \varnothing_{B_{k,M,n_B}}(t) \right] \\ \varnothing_{k,A}(t) = & \left[-\sum_{j=1}^{[t/h]} w_j^\alpha y_k(t - jh), \dots, -\sum_{j=1}^{[t/h]} w_j^{n_A \alpha} y_k(t - jh) \right] \\ \varnothing_{B_{k,l,i}}(t) = & \left[\sum_{j=0}^{[t/h]} w_j^{i\alpha} f_{l,1}(u_l(t - jh)), \dots, \sum_{j=0}^{[t/h]} w_j^{i\alpha} f_{l,M}(u_l(t - jh)) \right] \end{aligned} \tag{19}$$

According to Equations (16) and (17), the vector θ_k is found and expressed as

$$\begin{aligned} \theta_k = & \left[\theta_{k,A}, \theta_{B_{k,1,0}}, \dots, \theta_{B_{k,1,n_B}}, \dots, \theta_{B_{k,M,0}}, \dots, \theta_{B_{k,M,n_B}} \right] \\ \theta_{k,A} = & \left[Q_{k,1}, Q_{k,2}, \dots, Q_{k,n_A} \right] \\ \theta_{B_{k,l,i}} = & \left[W_{k,1,i} c_{1,1}, \dots, W_{k,1,i} c_{1,n_{lc}}, \dots, W_{k,M,i} c_{M,1}, \dots, W_{k,M,i} c_{M,n_{lc}} \right] \end{aligned} \tag{20}$$

where

$$\begin{aligned} Q_{k,i} = & \frac{\frac{A_{k,i}}{h^{i\alpha}}}{1 + \sum_{i=1}^{n_A} \frac{A_{k,i}}{h^{i\alpha}}} \\ W_{k,l,j} = & \frac{\frac{B_{k,l,j}}{h^{j\alpha}}}{1 + \sum_{i=1}^{n_A} \frac{A_{k,i}}{h^{i\alpha}}} \end{aligned} \tag{21}$$

It can be clearly seen that θ_k contains coefficients that need to be identified. It is worth mentioning that $y_k(t - jh)$ is unknown so that $\theta_{k,A}$ cannot be identified directly by $\varnothing_{k,A}(t)$. According to references [44], an auxiliary model is used to estimate the unknown variable $y_k(t - jh)$. The auxiliary model of this paper can be schematically represented in Figure 5. The main idea of the auxiliary model is that the real output of the system $y_k'(t)$ is replaced by the output of the auxiliary model $y_{amk}(t)$. Then, the identification problem has changed from the relationship between $y_k'(t)$ and u_l to the relationship between $y_{amk}(t)$ and u_l .

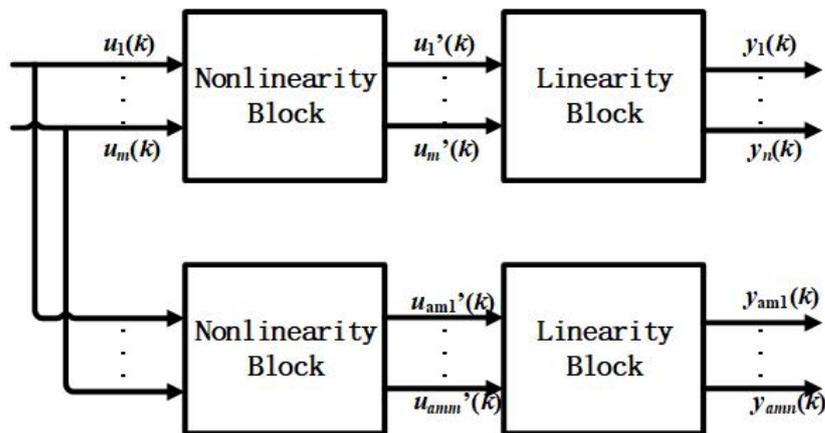


Figure 5. The MIMO Hammerstein model based on the auxiliary model.

According to Figure 5, the input–output relations of the auxiliary model can be written as

$$y_{amk}(t) = \varnothing_{amk}(t) \cdot \theta_{amk}^T \tag{22}$$

where

$$\begin{aligned} \varnothing_{amk}(t) &= [\varnothing_{amk,A}(t), \varnothing_{B_{k,1,0}}(t), \varnothing_{B_{k,1,2}}(t), \dots, \varnothing_{B_{k,1,n_B}}(t), \dots, \varnothing_{B_{k,M,0}}(t), \varnothing_{B_{k,M,2}}(t), \dots, \varnothing_{B_{k,M,n_B}}(t)] \\ \varnothing_{amk,A}(t) &= [-\sum_{j=1}^{\lceil t/h \rceil} w_j^\alpha y_{amk}(t - jh), \dots, -\sum_{j=1}^{\lceil t/h \rceil} w_j^{n_{A\alpha}} y_{amk}(t - jh)] \theta_{amk} = \hat{\theta}_k \end{aligned} \tag{23}$$

The estimate of $\varnothing_k(t)$ can be used as the value of the auxiliary model information vector $\varnothing_{amk}(t)$ and the parameter identification of θ_k can be used as the value of the auxiliary model parameter vector θ_{amk} . Define the criterion function as

$$J(\hat{\theta}_k^T) = \frac{1}{2} \sum_{i=1}^t [y_k'(i) - \varnothing_{amk}(i) \hat{\theta}_k^T]^2 \tag{24}$$

By finding the minimum value of the criterion function, the value of $\varnothing_{amk}(i) \hat{\theta}_k^T$ can approach the value of $y_k'(i)$ to identify $\hat{\theta}_k$. The minimum value can be obtained by the following equation.

$$\frac{\partial J(\hat{\theta}_k^T)}{\partial \hat{\theta}_k^T} = -\sum_{i=1}^t \varnothing_{amk}^T(i) \cdot [y_k'(i) - \varnothing_{amk}(i) \hat{\theta}_k^T] = 0 \tag{25}$$

When $\sum_{i=1}^t \varnothing_{amk}^T(i - 1) \cdot \varnothing_{amk}(i - 1)$ can be inverted, the value of $\hat{\theta}_k$ can be identified by the recursive least squares as follows:

$$\begin{aligned}
 \hat{\theta}_k^T(t) &= [\sum_{i=1}^t \varnothing_{amk}^T(i-1) \cdot \varnothing_{amk}(i-1)]^{-1} \cdot \sum_{i=1}^t \varnothing_{amk}^T(i) y_k'(i) \\
 \hat{\theta}_k^T(t) &= \hat{\theta}_k^T(t-1) + L(t) [y_k'(t) - \varnothing_{amk}(t) \hat{\theta}_k^T(t-1)] \\
 L(t) &= P(t-1) \varnothing_{amk}^T(t) [1 + \varnothing_{amk}(t) P(t-1) \varnothing_{amk}^T(t)]^{-1} \\
 P(t) &= [I - L(t) \varnothing_{amk}(t)] P(t-1)
 \end{aligned}
 \tag{26}$$

where $P(0)$ is a diagonal matrix in which the main diagonal elements are huge and equal.

According to the above equations, the elements of $\hat{\theta}_k$ are all identified. Without losing generality, assuming $c_{l,1}$ as 1 can facilitate calculation and ensure the uniqueness of the final parameters. Then, the unique values of $W_{k,l,j}$ and $c_{l,m}$ are calculated; they can be expressed as

$$\begin{aligned}
 W_{k,l,j} &= \theta_{B_{k,l,i}} [(l-1) * n_{lc} + 1] \\
 c_{l,m} &= \sum_{i=0}^{n_{lc}} \frac{\theta_{B_{k,l,i}}^{(k)}}{W_{k,l,j}}
 \end{aligned}
 \tag{27}$$

So far, the estimates of A , B , and c have been obtained.

3.2.2. Order Identification

In the previous section, this paper discusses the identification of coefficients. Substituting the accurate estimated value of the coefficients into Equation (17) can identify the order accurately. Define the criterion function as

$$J(\alpha) = \frac{1}{2} \sum_{i=1}^t [y_k'(i) - \hat{y}_k(i)]^2
 \tag{28}$$

By finding the minimum value of the criterion function, the value of $\hat{y}_k(i)$ can approach the value of $y_k'(i)$. The minimum value can be obtained by the following equation:

$$\frac{\partial J(\alpha)}{\partial \alpha} = - \sum_{i=1}^t \frac{\partial \hat{y}_k(i)}{\partial \alpha} \cdot [y_k'(i) - \hat{y}_k(i)] = 0
 \tag{29}$$

where

$$\begin{aligned}
 \frac{\partial \hat{y}_k(t)}{\partial \alpha} &= - \frac{\partial}{\partial \alpha} (G_{k,1}(s^\alpha) u_1'(t) + G_{k,2}(s^\alpha) u_2'(t) + \dots + G_{k,M}(s^\alpha) u_M'(t)) \\
 &= \sum_{l=0}^M \left[\left(\frac{B_{k,l,N_B} S^{N_B \alpha} + \dots + B_{k,l,0}}{(A_{k,N_A} S^{N_A \alpha} + \dots + 1)^2} \right) \right. \\
 &\quad \left. \cdot (N_A \cdot A_{k,N_A} S^{N_A \alpha} + \dots + A_{k,1} S^\alpha) - \frac{N_B \cdot B_{k,l,N_B} S^{N_B \alpha} + \dots + B_{k,l,1} S^\alpha}{A_{k,N_A} S^{N_A \alpha} + \dots + 1} \right] \cdot \ln(s) \cdot u_l'(t)
 \end{aligned}
 \tag{30}$$

According to references [24], $\ln(s) \cdot u_l'(t)$ can be replaced by $s^\alpha \cdot (\ln(s)/s^\alpha) \cdot u_l'(t)$. The inverse Laplace transform of $\ln(s)/s^\alpha$ is a digamma function can be expressed as

$$L^{-1} \left(\frac{\ln(s)}{s^\alpha} \right) = \frac{t^{\alpha-1}}{\Gamma(\alpha)} \left[\frac{1}{\Gamma(\alpha)} \frac{d\Gamma(\alpha)}{d\alpha} - \ln(t) \right]
 \tag{31}$$

Then, $\ln(s) \cdot u_l'(t)$ can be expressed as

$$D^\alpha \left[\frac{1}{\Gamma(\alpha)} \frac{d\Gamma(\alpha)}{d\alpha} D^\alpha u_l'(t) - \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \ln(t-\tau) u_l'(t) d\tau \right]
 \tag{32}$$

It's easy to see that α can be calculated by Equations (28)–(32).

3.3. Summary

So far, the estimates of A , B , c , and α have been obtained. Because A are polynomials about a , B are polynomials about a and b , it is feasible to estimate the value of $n_A a$ by the value of $n_A A$. Then, it is feasible to estimate the value of b by the value of a and B . To sum up, all estimates work has been completed.

4. Experimental Results

In this section, two numerical examples will demonstrate the validity of the proposed method.

4.1. Example 1

Consider the following model, which is expressed as

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{bmatrix} \begin{bmatrix} u_1'(t) \\ u_2'(t) \end{bmatrix} y'(t) = y(t) + v(t) \tag{33}$$

where

$$\begin{aligned} G_{1,1} &= \frac{4}{5s^{0.3}+1}, & G_{1,2} &= \frac{3}{3s^{0.3}+1}, \\ G_{2,1} &= \frac{4}{6s^{0.3}+1}, & G_{2,2} &= \frac{5}{2s^{0.3}+1}. \end{aligned} \tag{34}$$

$$\begin{aligned} u_1'(t) &= u_1(t) + 0.5u_1^2(t) + 0.3u_1^3(t) + 0.1u_1^4(t) \\ u_2'(t) &= u_2(t) + 0.4u_2^2(t) + 0.2u_2^3(t) + 0.1u_2^4(t) \end{aligned}$$

The inputs u_1 and u_2 are persistent excitation signal sequences with unit variance and zero mean. $v(t)$ is the stochastic Gaussian noise with zero mean and variance is 0.005. Then, the outputs $y(t)$ are generated by their respective transfer functions of the MIMO fractional-order Hammerstein model.

According to the model, the θ to be identified are

$$\begin{aligned} \theta &= [a_{1,1,1}, a_{1,2,1}, b_{1,1,0}, b_{1,2,0}, a_{2,1,1}, a_{2,2,1}, b_{2,1,0}, b_{2,2,0}, c_{1,1}, c_{1,2}, c_{1,3}, c_{2,1}, c_{2,2}, c_{2,3}, \alpha] \\ &= [5, 3, 4, 3, 6, 2, 4, 5, 0.5, 0.3, 0.1, 0.4, 0.2, 0.1, 0.3] \end{aligned} \tag{35}$$

The identification steps are described in Section 3. At first, the intelligent optimization algorithm identifies the initial value of the model. Then, using AMRLS to identify the model coefficients, and at this time regarding the initial value of fractional order as the model's actual value. When the coefficients are estimated, the estimated values of the coefficients are considered to be the true value to identify the fractional order. Finally, identifying coefficients and order is repeated until the iteration's end or satisfactory results are obtained. The pseudo-code of the identification process is explained in detail in Algorithm 2.

Algorithm 2: Identification process

- Step.1** Collect the dates of all inputs, outputs;
 - Step.2** Obtain the initial of unknown parameters by using intelligent optimization algorithm;
 - Step.3** **While** $m < \text{max number of iterations } M$;
 - Step.4** Estimate the value of model coefficients according to Equation (25);
 - Step.5** Estimate the value of fractional order according to Equation (29);
 - Step.6** If the two criterion function values J within the actual accuracy requirements;
 - Step.7** Break;
 - Step.8** End;
 - Step.9** $m = m + 1$;
 - Step.10** **End while**;
 - Step.11** Return the best solution.
-

In order to reflect the importance of the initial value of fractional order, in this section, the initial value is identified by three different optimization algorithms: AIAGS, HHO,

and AO. The next identification work is carried out under four initial values. This section evaluates the final identification results from two aspects: RQE and MSE. They can be expressed as

$$\begin{aligned}
 \text{RQE} &= \sqrt{\frac{(\hat{\theta} - \theta)^2}{\theta^2}} \\
 \text{MSE} &= \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}
 \end{aligned}
 \tag{36}$$

where $\hat{\theta}$ and \hat{y}_i are estimated values; θ and y_i are true values.

The final identification results obtained by Algorithm 2 are shown in Table 4, and the RQE and MSE of the results are shown in Table 5. The outputs of the real model and the outputs of the model obtained through identification are shown in Figures 6 and 7. Figure 8 shows the estimated fractional-order convergence curve.

Table 4. The final identification results.

Method (and AMRLS)	$a_{1,1,1}$	$a_{1,2,1}$	$b_{1,1,0}$	$b_{1,2,0}$	$a_{2,1,1}$	$a_{2,2,1}$	$b_{2,1,0}$	$b_{2,2,0}$	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	α	α_0
AIAGS	5.127	3.120	3.976	2.959	6.118	2.017	3.996	4.970	0.501	0.289	0.095	0.400	0.200	0.100	0.299	0.333
AO	4.619	3.325	3.887	3.465	5.377	1.760	4.196	5.272	0.509	0.297	0.098	0.404	0.198	0.098	0.275	0.391
HHO	4.641	3.329	3.882	3.448	5.289	1.757	4.152	5.283	0.508	0.296	0.097	0.404	0.198	0.099	0.278	0.382

Table 5. The RQE and MSE of the results.

Method (and AMRLS)	AIAGS	AO	HHO
RQE	0.1360	0.2931	0.2987
MSE	0.0144	0.0944	0.1019

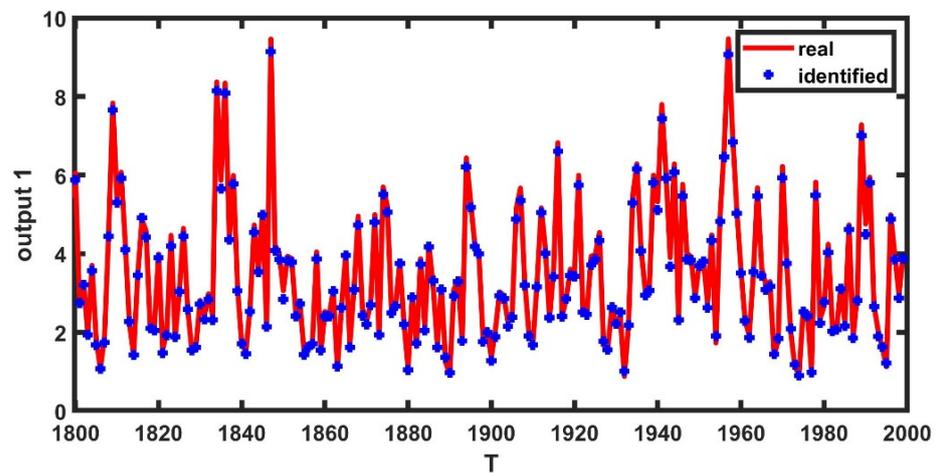


Figure 6. The real output 1 and the identified output 1.

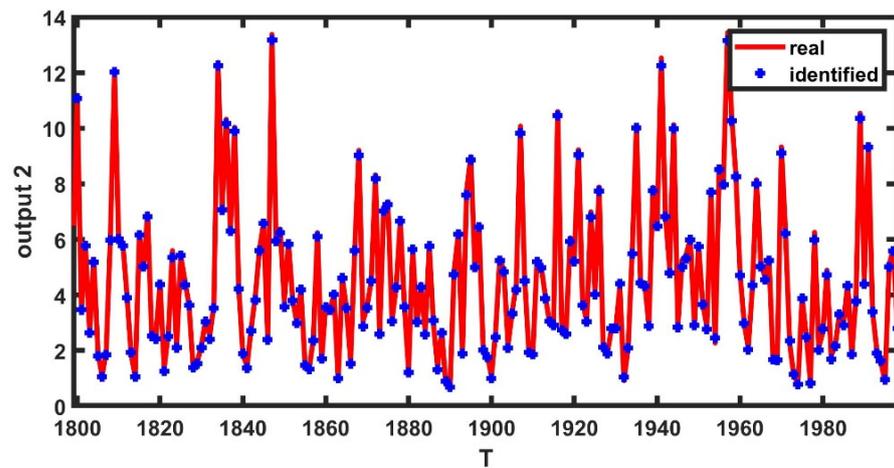


Figure 7. The real output 2 and the identified output 2.

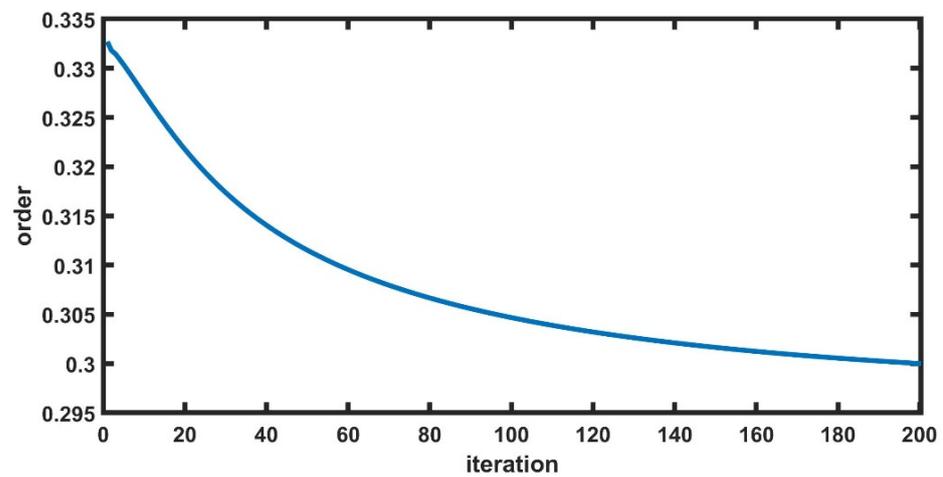


Figure 8. The estimated fractional-order convergence curve.

4.2. Example 2

Consider the following model, which is expressed as

$$\begin{aligned} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} &= \begin{bmatrix} G_{1,1} & G_{1,2} \\ G_{2,1} & G_{2,2} \end{bmatrix} \begin{bmatrix} u'_1(t) \\ u'_2(t) \end{bmatrix} \\ y'(t) &= y(t) + v(t) \end{aligned} \tag{37}$$

where

$$\begin{aligned} G_{1,1} &= \frac{5}{2s^{0.7}+1}, \quad G_{1,2} = \frac{1.7s^{0.7}+1.9}{1.5s^{1.4}+1.3s^{0.7}+1}, \\ G_{2,1} &= \frac{1.8s^\alpha+1.5}{2.2s^{1.4}+2.1s^{0.7}+1}, \quad G_{2,2} = \frac{1}{1.6s^{0.7}+1}. \end{aligned} \tag{38}$$

$$\begin{aligned} u'_1(t) &= u_1(t) + 0.5u_1^2(t) + 0.2u_1^3(t) + 0.1u_1^4(t) \\ u'_2(t) &= u_2(t) + 0.4u_2^2(t) + 0.3u_2^3(t) + 0.1u_2^4(t) \end{aligned}$$

The parameter meanings are similar to that of Example 1, so θ can be expressed as

$$\begin{aligned} \theta &= [a_{1,1,1}, a_{1,2,2}, a_{1,2,1}, b_{1,1,0}, b_{1,2,1}, b_{1,2,0}, a_{2,1,2}, a_{2,1,1}, a_{2,2,1}, b_{2,1,1}, b_{2,1,0}, b_{2,2,0}, c_{1,1}, c_{1,2}, c_{1,3}, c_{2,1}, c_{2,2}, c_{2,3}, \alpha] \\ &= [2, 1.5, 1.3, 5, 1.7, 1.9, 2.2, 2.1, 1.6, 1.8, 1.5, 1, 0.5, 0.2, 0.1, 0.4, 0.3, 0.1, 0.7] \end{aligned} \tag{39}$$

By repeating the identification process similar to Example 1, the final identification results are shown in Table 6, and the RQE and MSE of the results are shown in Table 7. The outputs of the real model and the outputs of the model obtained through identification

are shown in Figures 9 and 10. Figure 11 shows the estimated fractional-order convergence curve.

Table 6. The final identification results.

Method (and AMRLS)	$a_{1,1,1}$	$a_{1,2,2}$	$a_{1,2,1}$	$b_{1,1,0}$	$b_{1,2,1}$	$b_{1,2,0}$	$a_{2,1,2}$	$a_{2,1,1}$	$a_{2,2,1}$	$b_{2,1,1}$	$b_{2,1,0}$	$b_{2,2,0}$	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	α
AIAGS	2.002	1.501	1.297	5.033	1.703	1.915	2.164	2.215	1.766	1.675	1.564	1.029	0.504	0.191	0.095	0.385	0.293	0.101	0.700
AO	2.946	1.453	1.174	5.642	1.127	1.832	2.459	2.544	0.733	4.680	1.626	1.122	0.483	0.188	0.100	0.347	0.290	0.106	0.582
HHO	3.182	1.42	1.197	5.697	1.004	1.808	2.463	2.605	0.691	4.962	1.631	1.132	0.482	0.188	0.100	0.344	0.288	0.106	0.570

Table 7. The RQE and MSE of the results.

Method (and AMRLS)	AIAGS	AO	HHO
RQE	0.1819	0.6579	0.6935
MSE	0.0351	0.5133	0.6626

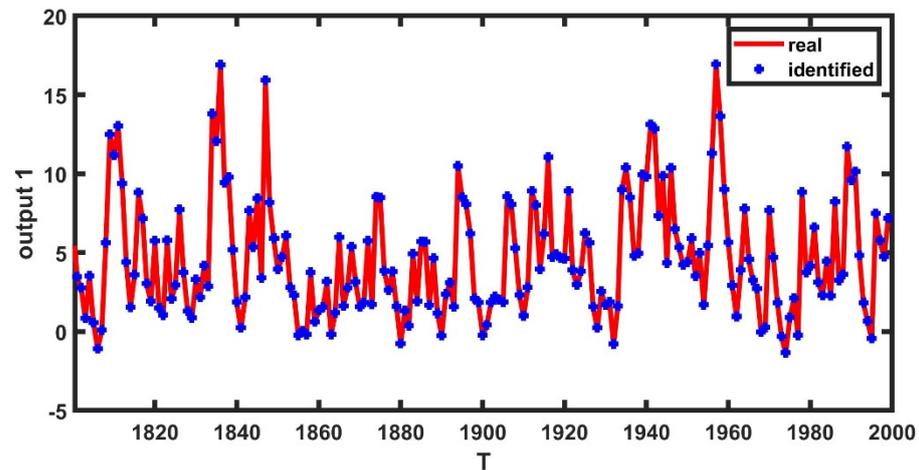


Figure 9. The real output 1 and the identified output 1.

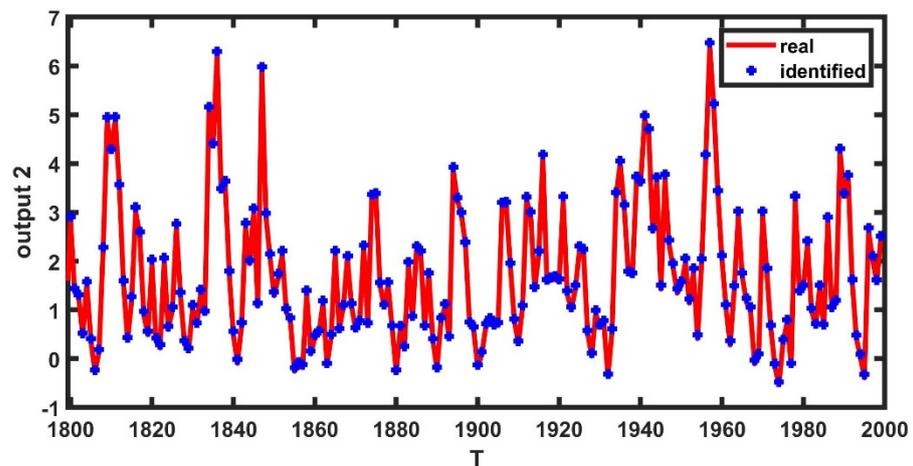


Figure 10. The real output 2 and the identified output 2.

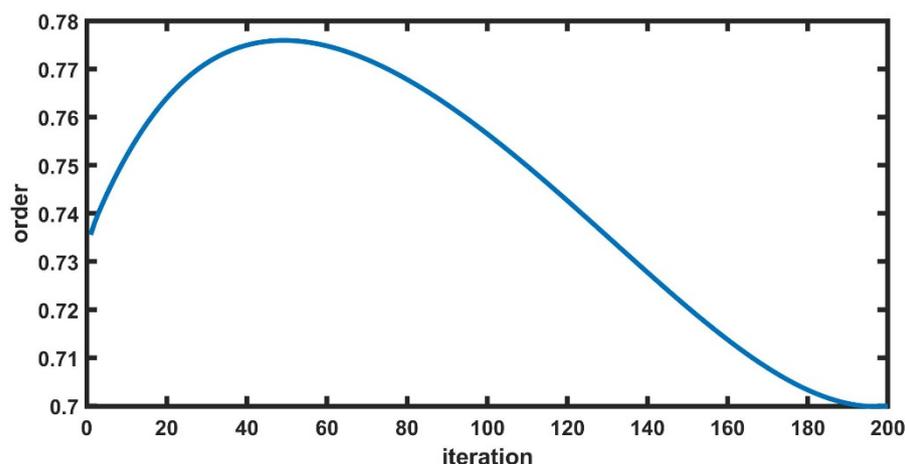


Figure 11. The estimated fractional-order convergence curve.

5. Conclusions

This paper discusses a new identification method for MIMO fractional-order Hammerstein models. In order to improve the accuracy of identification results, the identification process needs a heuristic algorithm to provide the initial value. Because the immune algorithm is prone to premature convergence, this paper improves the immune algorithm and proposes AIAGS. In AIAGS, the immune algorithm's stimulation function and mutation strategy are improved, and simulated annealing is added to the final step to select the optimal solution. The core idea of these improvements is to avoid finding the optimal local solution. Then, through the obtained initial value, the auxiliary model recursive least squares method is used to accurately identify all the MIMO fractional-order Hammerstein model parameters. The experimental results show the effectiveness of the proposed algorithm. The proposed methods in this paper can be applied to other literature [45–47], such as parameter identification problems of different systems, engineering applications, fault diagnosis, and so on.

Author Contributions: Conceptualization, Q.J. and B.W.; methodology, Q.J.; software, B.W.; validation, Q.J., B.W. and Z.W.; formal analysis, B.W.; investigation, B.W.; resources, Q.J.; data curation, Q.J. and B.W.; writing—original draft preparation, B.W.; writing—review and editing, Q.J., B.W. and Z.W.; visualization, Q.J.; supervision, Q.J. 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: In the paper, all the data generation information has been given in detail in the related chapter.

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

References

1. Billings, S.A.; Fakhouri, S.Y. Identification of systems containing linear dynamic and static nonlinear element. *Automatica* **1982**, *18*, 15–26. [\[CrossRef\]](#)
2. Narendra, K.S.; Gallman, P.G. An Iterative Method for the Identification of Nonlinear Systems using the Hammerstein Model. *IEEE Trans. Autom. Control* **1966**, *AC11*, 546–550. [\[CrossRef\]](#)
3. Moghaddam, M.J.; Mojallali, H.; Teshnehlab, M. Recursive identification of multiple-input single-output fractional-order Hammerstein model with time delay. *Appl. Soft Comput.* **2018**, *70*, 486–500. [\[CrossRef\]](#)
4. Chen, H.F. Strong consistency of recursive identification for Hammerstein systems with discontinuous piecewise-linear memoryless block. *IEEE Trans. Autom. Control* **2005**, *50*, 1612–1617. [\[CrossRef\]](#)

5. Sznaier, M. Computational complexity analysis of set membership identification of Hammerstein and Wiener systems. *Automatica* **2009**, *45*, 701–705. [[CrossRef](#)]
6. Kung, M.C.; Womack, B. Discrete time adaptive control of linear dynamic systems with a two-segment piecewise-linear asymmetric nonlinearity. *IEEE Trans. Autom. Control* **2003**, *29*, 170–172. [[CrossRef](#)]
7. Mccannon, T.E.; Gallagher, N.C.; Minoos-Hamedani, D.; Wise, J.L. On the design of nonlinear discrete-time predictors. *IEEE Trans. Inf. Theory* **1982**, *28*, 366–371. [[CrossRef](#)]
8. Jin, Q.; Wang, H.; Su, Q. A novel optimization algorithm for MIMO Hammerstein model identification under heavy-tailed noise. *ISA Trans.* **2018**, *72*, 77–91. [[CrossRef](#)] [[PubMed](#)]
9. Jin, Q.; Xu, Z.; Cai, W. An Improved Whale Optimization Algorithm with Random Evolution and Special Reinforcement Dual-Operation Strategy Collaboration. *Symmetry* **2021**, *13*, 238. [[CrossRef](#)]
10. Dong, S.; Yu, L.; Zhang, W.A. Robust extended recursive least squares identification algorithm for Hammerstein systems with dynamic disturbances. *Digit. Signal Process.* **2020**, *101*, 102716. [[CrossRef](#)]
11. Dhaifallah, M.A.; Westwick, D.T. Support Vector Machine Identification of Output Error Hammerstein Models. *IFAC Proc. Vol.* **2011**, *44*, 13948–13953. [[CrossRef](#)]
12. Schlegel, M.; Čech, M. Fractal System Identification for Robust Control—The Moment Approach. In Proceedings of the 5th International Carpathian Control Conference, Zakopane, Poland, 25–28 May 2004.
13. Torvik, P.J.; Bagley, R.L. On the Appearance of the Fractional Derivative in the Behavior of Real Materials. *J. Appl. Mech.* **1984**, *51*, 725–728. [[CrossRef](#)]
14. Zhao, C.; Dingy, X. Closed-form solutions to fractional-order linear differential equations. *Front. Electr. Electr. Eng. China* **2008**, *3*, 214–217. [[CrossRef](#)]
15. Chen, S.; Liu, F.; Turner, I. Numerical inversion of the fractional derivative index and surface thermal flux for an anomalous heat conduction model in a multi-layer medium. *Appl. Math. Model.* **2018**, *59*, 514–526. [[CrossRef](#)]
16. Deng, J. Higher-order stochastic averaging for a SDOF fractional viscoelastic system under bounded noise excitation. *J. Frankl. Inst.* **2017**, *354*, 7917–7945. [[CrossRef](#)]
17. Das, S. *Functional Fractional Calculus*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 110–122.
18. Table, M.A.; Béthoux, O.; Godoyb, E. Identification of a PEMFC fractional order model. *Int. J. Hydrogen Energy* **2017**, *42*, 1499–1509.
19. Kumar, S.; Ghosh, A. Identification of fractional order model for a voltammetric E-tongue system. *Measurement* **2019**, *150*, 107064. [[CrossRef](#)]
20. Zhang, Q.; Shang, Y.; Li, Y. A novel fractional variable-order equivalent circuit model and parameter identification of electric vehicle Li-ion batteries. *ISA Trans.* **2019**, *97*, 448–457. [[CrossRef](#)]
21. Hammar, K.; Djamah, T.; Bettayeb, M. Fractional hammerstein system identification using particle swarm optimization. In Proceedings of the 2015 7th International Conference on Modelling, Identification and Control (ICMIC) 2015, Sousse, Tunisia, 18–20 December 2015; pp. 1–6.
22. Hammar, K.; Djamah, T.; Bettayeb, M. Fractional Hammerstein system identification based on two decomposition principles. *IFAC-PapersOnLine* **2019**, *52*, 206–210. [[CrossRef](#)]
23. Chetoui, M.; Aoun, m. Instrumental variables based methods for linear systems identification with fractional models in the EIV context. In Proceedings of the 2019 16th International Multi-Conference on Systems, Signals & Devices (SSD) 2019, Istanbul, Turkey, 21–24 March 2019; pp. 90–95.
24. Wang, J.; Wei, Y.; Liu, T. Fully parametric identification for continuous time fractional order Hammerstein systems. *J. Frankl. Inst.* **2020**, *357*, 651–666. [[CrossRef](#)]
25. Cui, R.; Wei, Y.; Chen, Y. An innovative parameter estimation for fractional-order systems in the presence of outliers. *Nonlinear Dyn.* **2017**, *89*, 453–463. [[CrossRef](#)]
26. Tang, Y.; Liu, H.; Wang, W. Parameter identification of fractional order systems using block pulse functions. *Signal Process.* **2015**, *107*, 272–281. [[CrossRef](#)]
27. Cui, R.; Wei, Y.; Cheng, S. An innovative parameter estimation for fractional order systems with impulse noise. *Isa Trans.* **2017**, 120–129. [[CrossRef](#)] [[PubMed](#)]
28. Zhao, Y.; Yan, L.; Chen, Y. Complete parametric identification of fractional order Hammerstein systems. In Proceedings of the ICFDA'14 International Conference on Fractional Differentiation and Its Applications 2014, Catania, Italy, 23–25 June 2014; pp. 1–6.
29. Zeng, L.; Zhu, Z.; Shu, L. Subspace identification for fractional order Hammerstein systems based on instrumental variables. *Int. J. Control Autom. Syst.* **2012**, *10*, 947–953.
30. Khanra, M.; Pal, J. Reduced Order Approximation of MIMO Fractional Order Systems. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2013**, *3*, 451–458. [[CrossRef](#)]
31. Lakshmikantham, V.; Vatsala, A.S. Basic theory of fractional differential equations. *Nonlinear Anal. Theory Methods Appl.* **2008**, *69*, 2677–2682. [[CrossRef](#)]
32. Chun, J.S.; Jung, H.K.; Hahn, S.Y. A study on comparison of optimization performances between immune algorithm and other heuristic algorithms. *IEEE Trans Magn.* **1998**, *34*, 297222975.
33. Castro, L.N.; Castro, D.L.; Timmis, J. *Artificial Immune Systems: A New Computational Intelligence Approach*; Springer Science & Business Media: London, UK, 2002.

34. Chen, X.L.; Li, J.Q.; Han, Y.Y. Improved artificial immune algorithm for the flexible job shop problem with transportation time. *Meas. Control* **2020**, *53*, 2111–2128. [[CrossRef](#)]
35. Lu, L.; Guo, Z.; Wang, Z. Parameter Estimation for a Capacitive Coupling Communication Channel Within a Metal Cabinet Based on a Modified Artificial Immune Algorithm. *IEEE Access* **2021**, 75683–75698. [[CrossRef](#)]
36. Samigulina, G.; Samigulina, Z. Diagnostics of industrial equipment and faults prediction based on modified algorithms of artificial immune systems. *J. Intell. Manuf.* **2021**. [[CrossRef](#)]
37. Yu, T.; Xie, M.; Li, X. Quantitative method of damage degree of power system network attack based on improved artificial immune algorithm. In Proceedings of the ICAIIS 2021: 2021 2nd International Conference on Artificial Intelligence and Information Systems, Chongqing, China, 28–30 May 2021.
38. Xu, Y.; Zhang, J. Regional Integrated Energy Site Layout Optimization Based on Improved Artificial Immune Algorithm. *Energies* **2020**, *13*, 4381. [[CrossRef](#)]
39. Eker, E.; Kayri, M.; Ekinci, S. A New Fusion of ASO with SA Algorithm and Its Applications to MLP Training and DC Motor Speed Control. *Arab. J. Sci. Eng.* **2021**, *46*, 3889–3911. [[CrossRef](#)]
40. Wu, G.; Mallipeddi, R.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization*; Technical Report; Nanyang Technological University: Singapore, 2016.
41. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
42. Abualigah, L.; Yousri, D.; Elaziz, M.A. Matlab Code of Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [[CrossRef](#)]
43. Dzieliński, A.; Sierociuk, D. Stability of discrete fractional order state-space systems. *IFAC Proc. Vol.* **2006**, *39*, 505–510. [[CrossRef](#)]
44. Ding, F.; Chen, H.; Li, M. Multi-innovation least squares identification methods based on the auxiliary model for MISO systems. *Appl. Math. Comput.* **2007**, *187*, 658–668. [[CrossRef](#)]
45. Zhao, X.; Lin, Z.; Bo, F. Research on Automatic Generation Control with Wind Power Participation Based on Predictive Optimal 2-Degree-of-Freedom PID Strategy for Multi-area Interconnected Power System. *Energies* **2018**, *11*, 3325. [[CrossRef](#)]
46. Ding, J.; Chen, J.; Lin, J. Particle filtering based parameter estimation for systems with output-error type model structures. *J. Frankl. Inst.* **2019**, *356*, 5521–5540. [[CrossRef](#)]
47. Wang, L.; Liu, H.; Dai, L. Novel Method for Identifying Fault Location of Mixed Lines. *Energies* **2018**, *11*, 1529. [[CrossRef](#)]