AIMS Mathematics, 11(4): 12132–12154.
DOI: 10.3934/math.2026498
Received: 17 February 2026
Revised: 23 March 2026
Accepted: 30 March 2026
Published: 29 April 2026

http://www.aimspress.com/journal/Math

Research article

Hierarchical neural identification approach for Hammerstein large-scale stochastic systems: A simulation study of hydraulic process

Rihab Issaoui[1], **Mourad Elloumi**[2,3], **Imed Bouzida**[4,5,6] **and Omar Naifar**[1,4,*]

[1] Control and Energy Management Laboratory, National School of Engineering, Sfax University, Sfax, Tunisia

[2] Faculty of Sciences of Gafsa, University of Gafsa, Gafsa, Tunisia

[3] Laboratory of Sciences and Technology of Automatic Control and Computer Engineering, National School of Engineering of Sfax, Sfax University, P.O. Box 1173, Sfax 3038, Tunisia

[4] Higher Institute of Applied Science and Technology of Kairouan, University of Kairouan, Kairouan, Tunisia

[5] Department of Mathematics and Statistics, College of Engineering, Abu Dhabi University, Abu Dhabi, United Arab Emirates

[6] Laboratory of Probability and Statistics, Sfax University, Sfax, Tunisia

* **Correspondence:** Email: omar.naifar@enis.tn.

Abstract: This paper proposes an interconnected Hammerstein neural network (IHNN)-based hybrid identification method for large-scale interconnected Hammerstein systems subject to stochastic disturbances. In the proposed method, the static nonlinear blocks are approximated by neural networks, while the linear dynamic parameters are recursively estimated using a recursive least-squares scheme with forgetting and covariance adaptation. The proposed identification framework preserves the block-oriented Hammerstein structure and is designed to handle strong subsystem interconnections and noisy operating conditions. A Lyapunov-based analysis is further developed to establish convergence and stability conditions for the overall learning algorithm, which combines backpropagation for the neural-network parameters and recursive estimation for the linear dynamics. The effectiveness of the proposed IHNN identification method is validated through a benchmark interconnected system and a hydraulic-process case study. The simulation results show consistent improvements over a conventional recursive extended least squares (RELS) baseline, including root mean square error (RMSE) reductions of about 35–38% and prediction-error variance reductions of about 60%, at the expense of increased computational time. These results demonstrate that the proposed IHNN approach provides an accurate and practical solution for identifying noisy large-scale interconnected Hammerstein systems.

Keywords: system identification; Hammerstein model; neural networks; large-scale systems;

stochastic systems; hydraulic process
Mathematics Subject Classification: 68T07, 93A30, 93B30, 93C56

1. Introduction

Identifying nonlinear, large-scale interconnected systems with stochastic disturbances is a long time challenge in control engineering. Block-oriented Hammerstein models (static nonlinearity followed by linearity) provide an interpretable and compressed description for many industrial processes, however, estimation accuracy is reduced in cases with strong interconnections, colored noise, and time delays [1–4]. These difficulties are increased in multi-input/multi-output situations wherein coupling results in bias and variance trade-offs [1, 3].

Neural networks (NNs) have the universal approximation capabilities, which are very appropriate to model the static nonlinear blocks without restricting them to polynomial structures. Recent advancements have shown reliable learning of complex nonlinear mappings in identification and forecasting, which includes deterministic learning artificial neutral network (ANN) [5, 6] identification and physics informed recurrent architectures. For Hammerstein models in particular, even deeper NN parameterizations are able to increase representational power as compared to shallow polynomial expansions [7]. Recent neural-network-based parameter identification methods for Hammerstein systems have also been reported in the literature. For example, Li et al. proposed a parameter identification method for nonlinear Hammerstein models using a stacked sparse autoencoder (SSAE) network to represent the static nonlinear block, while the linear dynamic part is described by an AutoRegressive Moving Average with eXogenous inputs (ARMAX) model and estimated through a multi-innovation recursive extended least squares (RELS) strategy [8]. In addition, Li et al. developed a deep GRU-network-based parameter estimation approach for multi-input multi-output Hammerstein nonlinear systems [9]. These recent studies further confirm the growing interest in combining deep neural architectures with Hammerstein-system identification. Extent advances in graph/hypergraph reasoning as well as task-based architectures have been made to aid yet again model relational dependencies and difficult patterns in connected data [10, 11].

At the same time, vast interconnected systems encourage decentralized and data-driven designs, and decentralized control and adaptive graph reasoning with reinforcement learning to invoke scalable routes to coordination [10, 12]. Complementary threads are also employed in robust operation under uncertainty, including fault-tolerant tracking/estimation methods for linear parameter-varying (LPV) and fuzzy systems, and fractional-order techniques for modelling, observer design, and stability analysis. Specifically, [13] extends Barbalat's lemma to the framework of generalised conformable fractional integrals, providing a theoretical foundation for adaptive observer design in fractional-order systems; investigates the synchronisation of mutually coupled fractional-order one-sided Lipschitz systems, which is relevant for coordinating interconnected nonlinear subsystems; presents observer-based control strategies for fractional-order systems, emphasising the design of convergent state estimators under fractional dynamics; and focuses on observer design for systems based on renewable energy sources, addressing practical challenges such as sensor noise and model uncertainties in energy-conversion processes. Furthermore, gradient-based optimisation can be extended to generalised

fractional-order settings [14]. In safety- and security-critical deployments, advances in large-scale intrusion detection and entropy modelling of complex activity provide insight into the importance of resilience and the information-theoretic structure of shaped and networked activities [15, 16].

Despite these advances, a gap still remains on the principled integration of neural-network-based static nonlinear blocks with recursive estimation of linear dynamics for large-scale interconnected Hammerstein systems operating in stochastic environments. Prior iterative/recursive identification studies, such as the iterative parametric estimation method for Hammerstein large-scale systems proposed by Elloumi and Kamoun [2], effectively handle linear dynamics via RELS but rely on polynomial approximations of the nonlinearities, which limit accuracy for strongly nonlinear or time-varying interconnections. More recent noise-aware Hammerstein identification methods [3] focus on multiple inputs multiple outputs (MIMO) Hammerstein models with process noise, employing correlation-analysis and data-filtering techniques, yet they do not address the simultaneous online learning of nonlinear neural mappings together with recursive linear parameter updates. Similarly, Zhang et al. [4] consider Hammerstein systems with time delay and colored noise, proposing identification schemes that separate the estimation of the nonlinear block and the linear dynamics, but without a unified Lyapunov-based convergence analysis that covers both parts jointly. Decentralised reinforcement-learning-based control strategies [12] are designed for constrained interconnected systems, but they prioritise control performance rather than the hybrid identification problem with convergence guarantees for the combined neural-recursive structure. Consequently, none of these works fully solve the joint neural-network approximation and recursive dynamical parameter estimation with provable convergence in the interconnected, time-varying, and noise-corrupted regime targeted in this paper.

We present an interconnected Hammerstein neural network (IHNN) model of identification which retains the block based structure but takes advantage of neural approximators to capture nonlinearities and recursive updates to capture linear dynamics:

- **Architecture:** An IHNN that explicitly codes intra-/inter-subsystem couplings of Hammerstein systems for the accurate approximation of subsystem nonlinearities and their interactions.
- **Hybrid learning:** Two-stage algorithm, which couples backpropagation for NN weights with recursive (extended) least squares (with forgetting/covariance adaptation) for the dynamic parameters, suitable for time-varying stochastic settings.
- **Convergence guarantees:** Lyapunov-based analysis of the hybrid estimator to establish stability and convergence conditions in the presence of stochastic disturbances, supported by modern estimator-stability tools and fractional-order insights.

The proposed IHNN framework is assessed through numerical simulations and a hydraulic-process case study, where its identification performance is compared with that of a conventional RELS-based method. We also discuss choice on implementation (e.g., Xavier initialization for stable signal propagation). We also give an extension—namely, a dynamic event-triggered update and game theoretic formulation for multi agent identification and control in a networked system that can further reduce the communication/computation or model strategic interactions occur in these networked systems [17–19].

Organization. Section 2 formalizes the model of nonlinear interconnected systems. Section 3 presents the IHNN-based identification method and the hybrid training algorithm with convergence

analysis. Section 4 reports numerical and hydraulic-process simulations and comparisons. Section 5 concludes with perspectives on distributed deployment, fault-aware extensions, and security-conscious identification.

2. Mathematical modelling of nonlinear interconnected systems

The synthesis of an effective control law necessitates a foundational mathematical model. This section therefore develops a discrete-time model to describe the dynamics of large-scale nonlinear systems, specifically those composed of interconnected monovariable subsystems. The model accounts for subsystems with known structural variables—such as order and delay—but with unknown time-varying parameters.

We consider a large-scale nonlinear system S comprised of $\mathcal{N}$ interconnected subsystems $S_1, \ldots, S_{\mathcal{N}}$. These subsystems, governed by unknown time-varying parameters, operate in a stochastic environment. The system structure is illustrated in Figure 1.

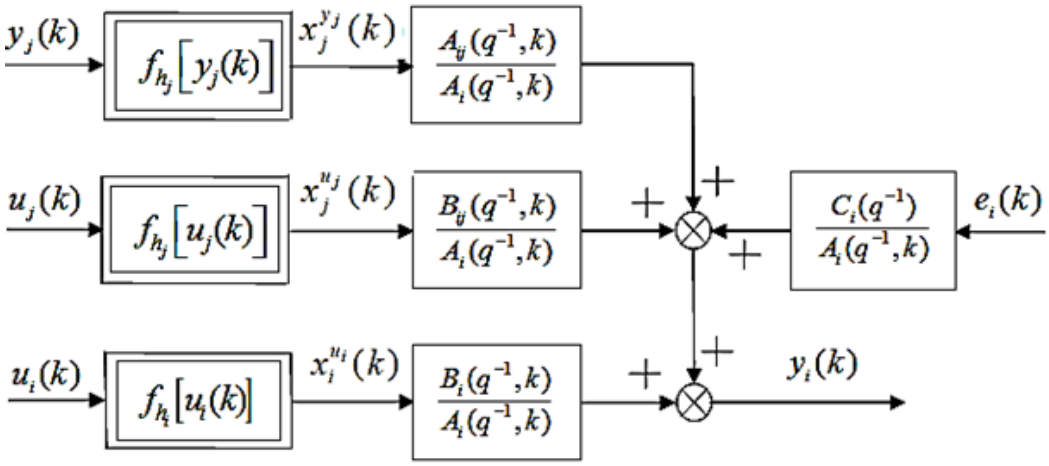

Figure 1. Structure of interconnected Hammerstein stochastic system.

The dynamic linear part of the considered structure is expressed by the following equation:

$$A_i(q^{-1},k)y_i(k) = B_i(q^{-1},k)x_i^{u_i}(k) + \sum_{j=1,j\neq i}^{N} B_{ij}(q^{-1},k)x_j^{u_j}(k) + \sum_{j=1,j\neq i}^{N} A_{ij}(q^{-1},k)x_j^{y_j}(k) + C_i(q^{-1})e_i(k), \quad (2.1)$$

where $x_i^{u_i}(k)$ and $y_i(k)$ represent the input and the output of the dynamic linear block at the discrete-time k, respectively; $u_i(k)$, $u_j(k)$, and $y_j(k)$ denote the inputs of the static nonlinear blocks; $x_j^{u_j}(k)$ and $x_j^{y_j}(k)$ designate the inputs from the other interconnected systems S_j, $i, j = 1, \ldots, N$; $j \neq i$; $\{e_i(k)\}$ is a sequence of independent random variables with zero mean and constant variance σ_i^2; and $A_i(q^{-1},k)$, $B_i(q^{-1},k)$, $A_{ij}(q^{-1},k)$, $B_{ij}(q^{-1},k)$, and $C_i(q^{-1})$ are polynomials with unknown time-varying parameters, which are expressed as:

$$A_i(q^{-1},k) = 1 + a_{i,1}(k)q^{-1} + \ldots + a_{i,n_{A_i}}(k)q^{-n_{A_i}}, \quad (2.2)$$

$$B_i(q^{-1},k) = b_{i,1}(k)q^{-1} + \ldots + b_{i,n_{B_i}}(k)q^{-n_{B_i}}, \quad (2.3)$$

$$A_{ij}(q^{-1},k) = 1 + a_{ij,1}(k)q^{-1} + \ldots + a_{ij,n_{A_{ij}}}(k)q^{-n_{A_{ij}}}, \quad (2.4)$$

$$B_{ij}(q^{-1},k) = b_{ij,1}(k)q^{-1} + \ldots + b_{ij,n_{B_{ij}}}(k)q^{-n_{B_{ij}}}, \tag{2.5}$$

$$C_i(q^{-1}) = 1 + c_{i,1}q^{-1} + \ldots + c_{i,n_{C_i}}q^{-n_{C_i}}, \tag{2.6}$$

where $i, j = 1, \ldots, N$; $j \neq i$, and n_{A_i}, n_{B_i}, $n_{A_{ij}}$, $n_{B_{ij}}$, and n_{C_i} are the orders of the polynomials $A_i(q^{-1},k)$, $B_i(q^{-1},k)$, $A_{ij}(q^{-1},k)$, $B_{ij}(q^{-1},k)$, and $C_i(q^{-1})$, respectively.

The static non-linear blocks are represented by the non-linear equations:

$$x_i^{u_i}(k) = f_{x_i^{u_i}}[u_i(k)], \tag{2.7}$$

$$x_j^{u_j}(k) = f_{x_j^{u_j}}[u_j(k)], \tag{2.8}$$

$$x_j^{y_j}(k) = f_{x_j^{y_j}}[y_j(k)], \tag{2.9}$$

with $f_{x_i^{u_i}}[\cdot]$, $f_{x_j^{u_j}}[\cdot]$ and $f_{x_j^{y_j}}[\cdot]$ representing nonlinear functions.

The behavior of Eqs (2.7)–(2.9) can be captured by the polynomial approximations:

$$x_i^{u_i}(k) = \sum_{r_1=1}^{p_1} \alpha_{i,r_1} u_i^{r_1}(k) + \Delta x_i^{u_i}[u_i(k)], \tag{2.10}$$

$$x_j^{u_j}(k) = \sum_{r_2=1}^{p_2} \beta_{j,r_2} u_j^{r_2}(k) + \Delta x_j^{u_j}[u_j(k)], \tag{2.11}$$

$$x_j^{y_j}(k) = \sum_{r_3=1}^{p_3} \gamma_{j,r_3} y_j^{r_3}(k) + \Delta x_j^{y_j}[y_j(k)]. \tag{2.12}$$

The terms $\Delta x_i^{u_i}[u_i(k)]$, $\Delta x_j^{u_j}[u_j(k)]$, and $\Delta x_j^{y_j}[y_j(k)]$ denote the approximation errors of the nonlinear functions, which can be treated as output noise. The variance of these errors depends on the choice of the nonlinearity degrees p_t ($t = 1, 2, 3$), decreasing as p_t increases. We select p_t such that these approximation errors become negligible.

Combining the dynamics described by Eqs (2.1) and (2.10)–(2.12), we can express the system output as:

$$\begin{aligned} y_i(k) = &- \sum_{h=1}^{n_{A_i}} a_{i,h}(k) y_i(k-h) + \sum_{h=1}^{n_{B_i}} \sum_{r_1=1}^{p_1} b_{i,h}(k)\alpha_{i,r_1} u_i^{r_1}(k-h) \\ &+ \sum_{j=1,j\neq i}^{N} \sum_{h=1}^{n_{B_{ij}}} \sum_{r_2=1}^{p_2} b_{ij,h}(k)\beta_{j,r_2} u_j^{r_2}(k-h) \\ &+ \sum_{j=1,j\neq i}^{N} \sum_{h=1}^{n_{A_{ij}}} \sum_{r_3=1}^{p_3} a_{ij,h}(k)\gamma_{j,r_3} y_j^{r_3}(k-h) + \sum_{h=1}^{n_{C_i}} c_{i,h} e_i(k-h) + e_i(k), \end{aligned} \tag{2.13}$$

with $i, j = 1, \ldots, N$; $j \neq i$.

Without loss of generality, we assume that the polynomials $A_i(q^{-1},k)$, $B_i(q^{-1},k)$, $A_{ij}(q^{-1},k)$, $B_{ij}(q^{-1},k)$, and $C_i(q^{-1})$ share a common order n_i.

3. System identification using IHNN

In this section, we develop an NN-based identification approach for large-scale interconnected Hammerstein systems. The proposed methodology extends the traditional RELS algorithm by incorporating NN structures to model the nonlinear components of the system.

3.1. Neural network architecture for interconnected Hammerstein systems

The identification of Hammerstein large-scale systems is achieved through a specialized NN architecture that preserves the block-oriented structure of the Hammerstein model while leveraging the universal approximation capabilities of NNs. The proposed IHNN architecture is illustrated in Figure 2.

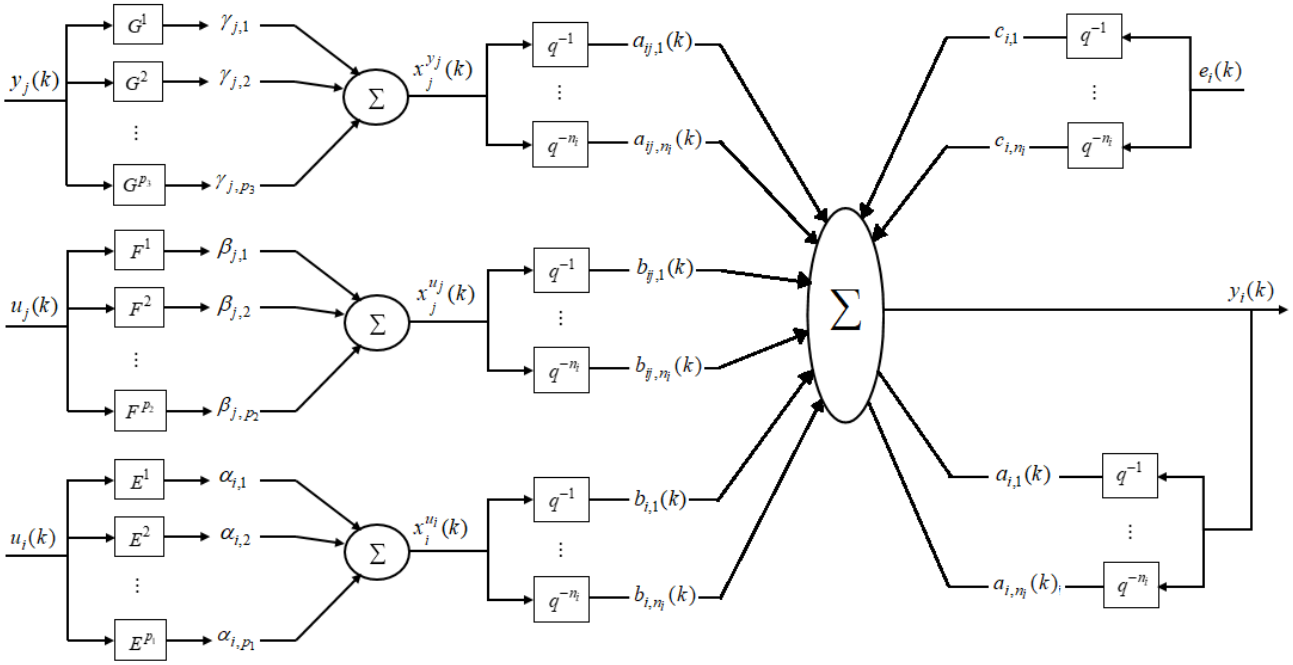

Figure 2. IHNN architecture for large-scale system identification.

The IHNN consists of three main components for each subsystem S_i.

The input NNs approximate the static nonlinear functions $f_{x_i^{u_i}}[\cdot]$ and $f_{x_j^{u_j}}[\cdot]$ that transform the inputs $u_i(k)$ and $u_j(k)$ to the intermediate variables $x_i^{u_i}(k)$ and $x_j^{u_j}(k)$, respectively:

$$x_i^{u_i}(k) = \mathrm{NN}_{u_i}(\Theta_i^{u_i}; u_i(k)), \tag{3.1}$$

$$x_j^{u_j}(k) = \mathrm{NN}_{u_j}(\Theta_i^{u_j}; u_j(k)), \tag{3.2}$$

where $\mathbf{\Theta}_i^{u_i}$ and $\mathbf{\Theta}_i^{u_j}$ represent the weight parameters of the inputs NNs.

The output NNs approximate the static nonlinear functions $f_{x_j^{y_j}}[\cdot]$ that transform the output $y_j(k)$ from interconnected subsystem S_j to the intermediate variables $x_j^{y_j}(k)$:

$$x_j^{y_j}(k) = \mathrm{NN}_{y_j}(\Theta_i^{y_j}; y_j(k)), \tag{3.3}$$

where $\mathbf{\Theta}_i^{y_j}$ represents the weight parameters of the output NNs.

The complete parameter vector for each subsystem S_i is given by:

$$\mathbf{\Theta}_i(k) = \left[(\boldsymbol{\theta}_i^L(k))^T, (\mathbf{\Theta}_i^{u_i}(k))^T, (\mathbf{\Theta}_i^{u_j}(k))^T, (\mathbf{\Theta}_i^{y_j}(k))^T\right]^T, \tag{3.4}$$

where $\boldsymbol{\theta}_i^L(k)$ contains the linear parameters and $\mathbf{\Theta}_i^{u_i}(k)$, $\mathbf{\Theta}_i^{u_j}(k)$, $\mathbf{\Theta}_i^{y_j}(k)$ contain the NN weights. Hence, the parameter estimation problem of the interconnected Hammerstein model, which corresponds to the weights of the multilayer NN, will be formulated based on the backpropagation learning algorithm.

3.2. Parameter estimation algorithm

The parameter estimation for the IHNN will be performed using a hybrid algorithm that combines backpropagation for the NN components and recursive least squares for the linear components.

3.2.1. Backpropagation-based gradient algorithm (BPG)

The BPG algorithm updates the NN parameters using gradient descent. The cost function for each subsystem is defined as:

$$J_i(w_i, k) = \frac{1}{2}e_i^2(k) = \frac{1}{2}\left(y_i(k) - \hat{y}_i(k)\right)^2, \tag{3.5}$$

where $y_i(k)$ represents the actual output, $\hat{y}_i(k)$ corresponds to the neural predicted one, and w_i represents the weights of the multilayer network or the parameters of the interconnected Hammerstein structure.

The gradients are computed using the backpropagation algorithm through time, taking into account the temporal nature of the dynamic system:

$$\frac{\partial J_i(k)}{\partial \boldsymbol{w_i}} = -e_i(k)\frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}}. \tag{3.6}$$

The update rules for the NNs parameters are given by:

$$\mathbf{\Theta}_i^{u_i}(k+1) = \mathbf{\Theta}_i^{u_i}(k) - \eta_i^{w_i}\frac{\partial J_i(k)}{\partial \mathbf{\Theta}_i^{u_i}}, \tag{3.7}$$

$$\mathbf{\Theta}_j^{u_j}(k+1) = \mathbf{\Theta}_j^{u_j}(k) - \eta_i^{w_i}\frac{\partial J_i(k)}{\partial \mathbf{\Theta}_j^{u_j}}, \tag{3.8}$$

$$\mathbf{\Theta}_j^{y_j}(k+1) = \mathbf{\Theta}_j^{y_j}(k) - \eta_i^{w_i}\frac{\partial J_i(k)}{\partial \mathbf{\Theta}_j^{y_j}}, \tag{3.9}$$

where $\eta_i^{w_i}$ represents the learning rate.

3.2.2. Recursive least squares algorithm

The linear dynamic parameters are updated using the following recursive algorithm:

$$\begin{aligned}
\boldsymbol{\theta}_i^L(k) &= \boldsymbol{\theta}_i^L(k-1) + \mathbf{K}_i(k)\left[y_i(k) - \boldsymbol{\varphi}_i^T(k)\boldsymbol{\theta}_i^L(k-1)\right],\\
\mathbf{K}_i(k) &= \frac{\mathbf{P}_i(k-1)\boldsymbol{\varphi}_i(k)}{\lambda_i(k) + \boldsymbol{\varphi}_i^T(k)\mathbf{P}_i(k-1)\boldsymbol{\varphi}_i(k)},\\
\mathbf{P}_i(k) &= \frac{1}{\lambda_i(k)}\left[\mathbf{P}_i(k-1) - \mathbf{K}_i(k)\boldsymbol{\varphi}_i^T(k)\mathbf{P}_i(k-1)\right],
\end{aligned} \tag{3.10}$$

where $\boldsymbol{\varphi}_i(k)$ is the regression vector containing past inputs, outputs, and prediction errors, $K_i(k)$ represents the Kalman gain, $P_i(k)$ is the covariance matrix, and $\lambda_i(k)$ indicates the forgetting factor.

3.2.3. Hybrid training algorithm

The complete hybrid training algorithm for the IHNN is summarized in Algorithm 1.

Algorithm 1 Hybrid training algorithm for IHNN.

1: Initialize NN weights $\boldsymbol{\Theta}_i^{u_i}(0)$, $\boldsymbol{\Theta}_i^{u_j}(0)$, $\boldsymbol{\Theta}_i^{y_j}(0)$ and linear parameters $\boldsymbol{\theta}_i^L(0)$
2: Initialize covariance matrices $\mathbf{P}_i(0)$ for RELS algorithm
3: **for** $k = 1$ to N **do**
4: **For each subsystem** S_i**:**
5: Compute NN outputs: $x_i^{u_i}(k)$, $x_j^{u_j}(k)$, $x_j^{y_j}(k)$
6: Form regression vector $\boldsymbol{\varphi}_i(k)$
7: Update linear parameters using RELS: $\boldsymbol{\theta}_i^L(k)$
8: Compute prediction error: $e_i(k) = y_i(k) - \boldsymbol{\varphi}_i^T(k)\boldsymbol{\theta}_i^L(k)$
9: Compute gradients: $\frac{\partial J_i(k)}{\partial \boldsymbol{\Theta}_i^{u_i}}$, $\frac{\partial J_i(k)}{\partial \boldsymbol{\Theta}_i^{u_j}}$, $\frac{\partial J_i(k)}{\partial \boldsymbol{\Theta}_i^{y_j}}$
10: Update NN weights using BPG algorithm
11: **end for**

3.3. Convergence analysis

The convergence properties of the proposed algorithm are analyzed using Lyapunov stability theory. Consider the Lyapunov function candidate

$$V_i(k) = \frac{1}{2}e_i(k)^2. \tag{3.11}$$

The choice of the scalar Lyapunov candidate $V_i(k) = \frac{1}{2}e_i^2(k)$ is justified by the fact that the proposed hybrid estimator can be viewed as a single prediction-error-driven adaptation scheme with complete parameter vector $\boldsymbol{\Theta}_i(k) = \left[(\boldsymbol{\theta}_i^L(k))^T, (\boldsymbol{\Theta}_i^{u_i}(k))^T, (\boldsymbol{\Theta}_i^{u_j}(k))^T, (\boldsymbol{\Theta}_i^{y_j}(k))^T\right]^T$. Although this vector contains both the linear dynamic parameters and the NN weights, all parameter updates are driven by the same one-step prediction error $e_i(k) = y_i(k) - \hat{y}_i(k, \boldsymbol{\Theta}_i(k))$. Hence, $V_i(k)$ provides a common scalar energy measure for the whole hybrid estimation system. In addition, since the variation of $e_i(k)$ depends on the joint sensitivity $\partial\hat{y}_i(k)/\partial\boldsymbol{\Theta}_i$, the decrease of $V_i(k)$ reflects the coordinated adaptation of all parameter blocks. Under standard identifiability and persistent-excitation conditions, together with bounded parameter sensitivities, the convergence $e_i(k) \to 0$ implies convergence of the complete parameter vector to the corresponding invariant set and to the true parameters when the model is identifiable [20–22].

The time difference of the Lyapunov function is given by

$$\begin{aligned} \Delta V_i(k) &= V_i(k+1) - V_i(k) \\ &= \frac{1}{2}\left[e_i^2(k+1) - e_i^2(k)\right] \\ &= \Delta e_i(k)\left[e_i(k) + \frac{\Delta e_i(k)}{2}\right]. \end{aligned} \tag{3.12}$$

The error difference can be expressed as:

$$\begin{aligned} e_i(k+1) &= e_i(k) + \Delta e_i(k) \\ &= e_i(k) + \left[\frac{\partial \hat{e}_i(k)}{\partial \boldsymbol{w_i}}\right]^T \Delta w_i. \end{aligned} \tag{3.13}$$

Based on the update rule, we obtain:

$$\begin{aligned} \Delta w_i &= -\eta_i e_i(k) \frac{\partial e_i(k)}{\partial \boldsymbol{w_i}} \\ &= \eta_i e_i(k) \frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}}. \end{aligned} \tag{3.14}$$

The assumptions used in Theorem 3.1 follow standard conditions commonly adopted in recursive least-squares identification and Lyapunov-based NN learning analyses [20–22].

Theorem 3.1. *Under the following conditions:*

(1) The system satisfies the persistent excitation condition [20, 21];
(2) The component vectors $\boldsymbol{\theta}_i^L(0)$, $\boldsymbol{\theta}_i^{u_i}(0)$, $\boldsymbol{\theta}_j^{u_j}(0)$, and $\boldsymbol{\theta}_j^{y_j}(0)$ contain finite numerical values;
(3) The covariance matrix $\mathbf{P}_i(k)$ remains bounded and converges to a steady-state value [21];
(4) The noise sequence $\boldsymbol{e}_i(k)$ is independent, uncorrelated with the regressor $\boldsymbol{\varphi}_i(k)$, and follows a Gaussian distribution with zero mean and constant variance [2];
(5) The forgetting factor satisfies $0 < \lambda_i(k) < 1$ [21];
(6) The learning rate $\eta_i^{w_i}$ for the weights of the IHNN is chosen such that

$$0 < \eta_i^{w_i} < \frac{2}{(\zeta_{i,max}^{w_i})^2}, \tag{3.15}$$

which is consistent with Lyapunov-based convergence conditions for NN learning rates [22, 23], where $\zeta_{i,max}^{w_i} = \max_k \left\| \frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w}_i} \right\|$.

Then, the convergence of the hybrid training algorithm is guaranteed.

Proof. The convergence of the hybrid training algorithm is analyzed using Lyapunov stability theory. Consider the Lyapunov function candidate

$$V_i(k) = \frac{1}{2} e_i^2(k).$$

The first difference of the Lyapunov function is given by

$$\Delta V_i(k) = V_i(k+1) - V_i(k) = \frac{1}{2}\left[e_i^2(k+1) - e_i^2(k)\right] = \Delta e_i(k)\left[e_i(k) + \frac{\Delta e_i(k)}{2}\right].$$

The error difference can be expressed in terms of the weight update

$$e_i(k+1) = e_i(k) + \Delta e_i(k) = e_i(k) + \left[\frac{\partial e_i(k)}{\partial \boldsymbol{w}_i}\right]^T \Delta \boldsymbol{w_i}.$$

Using the weight update rule from the BPG algorithm,

$$\Delta \boldsymbol{w_i} = -\eta_i^{w_i} e_i(k) \frac{\partial e_i(k)}{\partial \boldsymbol{w_i}} = \eta_i^{w_i} e_i(k) \frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}}.$$

Substituting into the expression for $\Delta V_i(k)$,

$$\Delta V_i(k) = \left[\frac{\partial e_i(k)}{\partial \boldsymbol{w_i}}\right]^T \eta_i^{w_i} e_i(k) \frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}} \left[e_i(k) + \frac{1}{2}\left(\left[\frac{\partial e_i(k)}{\partial \boldsymbol{w_i}}\right]^T \eta_i^{w_i} e_i(k) \frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}}\right)\right].$$

Noting that

$$\frac{\partial e_i(k)}{\partial \boldsymbol{w_i}} = -\frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}},$$

we simplify to obtain:

$$\Delta V_i(k) = -\eta_i^{w_i} e_i^2(k) \left\|\frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}}\right\|^2 + \frac{1}{2}\left(\eta_i^{w_i}\right)^2 e_i^2(k) \left\|\frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}}\right\|^4.$$

Let $\zeta_{i,\max}^{w_i} = \max_k \left\|\frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}}\right\|$. Then

$$\Delta V_i(k) \leq -\frac{1}{2}\left\|\frac{\partial \hat{y}_i(k)}{\partial \boldsymbol{w_i}}\right\|^2 \eta_i^{w_i}\left[2 - \eta_i^{w_i}(\zeta_{i,\max}^{w_i})^2\right] e_i^2(k).$$

Since $V_i(k) > 0$ for all k, convergence requires $\Delta V_i(k) < 0$. This condition is satisfied when

$$0 < \eta_i^{w_i} < \frac{2}{(\zeta_{i,\max}^{w_i})^2}.$$

Thus, under the stated conditions, the hybrid training algorithm converges. □

3.4. Implementation considerations

This section describes practical aspects of implementation of the proposed estimator, i.e., IHNN, which includes initialization, adaptive step-size/forgetting policies, and mechanisms for reliably handling subsystem interconnections and delays.

3.4.1. Initialization

To ensure rapid and steady convergence of the hybrid algorithm (NN backpropagation + recursive least squares), proper initialization is necessary.

- **NN weights.** All NN weights $\boldsymbol{\Theta}_i^{u_i}$, $\boldsymbol{\Theta}_i^{u_j}$, $\boldsymbol{\Theta}_i^{y_j}$ are initialized with Xavier initialization [17]. For a layer with n_{in} inputs, weights are sampled from a zero-mean uniform law with variance $1/n_{in}$ to preserve signal variance at start-up. Biases are set to zero. Inputs/outputs are linearly scaled to $[-1, 1]$ to match the initialization regime.
- **Linear-parameter warm start.** The dynamic parameters $\boldsymbol{\theta}_i^L(0)$ are obtained by ordinary least squares on a short window (50–100 samples) to seed the recursive estimator. If the initial regressor matrix is ill-conditioned, we use a ridge-regularized fit with Tikhonov parameter 10^{-4} (units consistent with the scaled data).

- **Covariance and numerical safety.** The covariance is set to $\mathbf{P}_i(0) = \delta^{-1}\mathbf{I}$ with $\delta \in [10^{-2}, 10^{-1}]$. Before each covariance update, we add a small jitter $\varepsilon\mathbf{I}$ ($\varepsilon \approx 10^{-9}$) to avoid numerical singularity during early transients.
- **Forgetting factor (initialization).** We set $\lambda_i(0) \in [0.99, 0.995]$ to favor stability until the estimates enter a low-error regime, after which $\lambda_i(k)$ is adapted online (see below). A conventional RELS configuration is recovered as a special case [2].

3.4.2. Architecture selection

The NN architectures used in this work were selected through preliminary empirical tuning aimed at balancing approximation capability, convergence behavior, and computational cost. Starting from compact feedforward structures, the number of hidden layers and neurons was gradually increased until the prediction accuracy improved satisfactorily and the training behavior became stable, while avoiding unnecessarily large networks that would increase computation and the risk of over-parameterization. In this sense, the retained architecture in each case corresponds to the smallest structure that provided a satisfactory trade-off between modeling accuracy and numerical efficiency for the considered identification task. An exhaustive hyperparameter optimization (e.g., full grid search) was not the main objective of this study and is left for future work.

3.4.3. Learning-rate and forgetting-factor adaptation

We combine a monotone decay for the NN learning rates with a local, error-aware correction and pair it with an error-adaptive forgetting factor for the recursive linear update.

$$\eta_i^{w_i}(k) = \frac{\eta_0}{1+\epsilon k}, \qquad \eta_0 \in [10^{-2}, 5\cdot 10^{-2}],\ \epsilon \in [10^{-3}, 5\cdot 10^{-3}], \tag{3.16}$$

$$\eta_i^{w_i}(k+1) = \begin{cases} 1.05\,\eta_i^{w_i}(k) & \text{if } \|\nabla J_i(k)\| < \|\nabla J_i(k-1)\|, \\ 0.7\,\eta_i^{w_i}(k) & \text{if } \|\nabla J_i(k)\| > 1.05\,\|\nabla J_i(k-1)\|, \\ \eta_i^{w_i}(k) & \text{otherwise,} \end{cases} \tag{3.17}$$

$$\eta_i^{w_i}(k) \leftarrow \min\Big\{\eta_i^{w_i}(k),\ \tfrac{2}{(\zeta_{i,\max}^{w_i})^2}\Big\} \quad \text{(step-size cap from the convergence analysis).} \tag{3.18}$$

To improve robustness, gradients are clipped as $\nabla J_i(k) \leftarrow \nabla J_i(k)\cdot \min\{1, c/\|\nabla J_i(k)\|\}$ with $c \in [1,5]$. The forgetting factor is adapted to the instantaneous prediction-error energy

$$\lambda_i(k) \;=\; \lambda_{\max} - (\lambda_{\max} - \lambda_{\min})\,\frac{\|e_i(k)\|^2}{\|e_i(k)\|^2 + \rho}, \lambda_{\max} \in [0.99, 0.995],\ \lambda_{\min} \in [0.95, 0.98],\ \rho > 0, \tag{3.19}$$

so that large errors induce faster tracking ($\lambda_i \downarrow$) while small errors favor variance reduction ($\lambda_i \uparrow$). When persistent excitation weakens, we temporarily raise $\lambda_i(k) \uparrow \lambda_{\max}$ and inflate $\mathbf{P}_i(k)$ diagonals by a small factor to preserve numerical conditioning [2].

3.4.4. Handling interconnections

The IHNN explicitly models inter-subsystem effects; the following protocol ensures reliable, scalable operation in distributed settings.

- **Update scheduling.** We use a block Gauss–Seidel sweep (update $S_1 \to S_2 \to \cdots \to S_N$ with freshest measurements) for faster practical convergence; a block Jacobi (fully parallel) variant is also supported when latency is critical. The natural physical flow (e.g., Tank 1→Tank 2→Tank 3) is preferred when it exists.
- **Weighted coupling.** Interconnections are encoded by a (learned) weighting matrix $\mathbf{W} = [w_{ij}]$:

$$x_j^{y_j}(k) \;=\; w_{ij}\,\mathrm{NN}_{y_j}(\Theta_i^{y_j};\, y_j(k-d_{ij})), \tag{3.20}$$

  where $d_{ij} \geq 0$ accounts for delays. During initialization, w_{ij} are fit by least squares on an interconnection probing sequence, then refined jointly with NN/linear parameters. We enforce sparsity by thresholding w_{ij} below τ and monitor $\rho(\mathbf{W}) < 1$ (spectral radius) as a conservative stability heuristic in strongly coupled regimes [10].
- **Delay and packet loss.** All messages carry timestamps; each S_i maintains a first in first out (FIFO) buffer of recent $\{y_j\}$ and aligns samples by $(k - d_{ij})$. Missing data are handled by zero-order hold up to a horizon $\Delta k_{\max}$, after which the corresponding term is skipped to avoid bias. These mechanisms are included to support delayed-interconnection scenarios at the algorithmic level; however, explicit experimental validation under time-delay conditions is beyond the scope of the present simulation study and is left for future work.
- **Communication efficiency.** For bandwidth-limited deployments, we use dynamic event-triggered exchanges—sending only when $|y_j(k) - y_j(k-1)| > \delta_y$ or when a timer elapses—thereby reducing traffic without harming estimation [18].
- **Noise robustness.** In high-noise or outlier-prone channels, we replace the NN quadratic loss with a Huber penalty (threshold $\kappa \in [1.5\sigma, 2.5\sigma]$) for the NN update while keeping the quadratic criterion for the recursive least squares recursion to retain closed-form updates for the linear part.

3.4.5. Regularization and numerical safeguards

To prevent overfitting and ensure stable iterates, we add an ℓ_2 penalty $\lambda_w\|\Theta\|_2^2$ on NN weights with $\lambda_w \in [10^{-6}, 10^{-4}]$, normalize all regressors in $\boldsymbol{\varphi}_i(k)$ to unit variance, and apply early stopping based on a rolling validation window. Stability/consistency monitors include (i) boundedness of $\mathbf{P}_i(k)$, (ii) absence of covariance collapse (eigenvalues $> 10^{-10}$), and (iii) satisfaction of the step-size bound from the Lyapunov analysis. These choices are consistent with the theoretical conditions used in our convergence arguments and standard practice in recursive estimation [2, 13].

4. Simulation study

We assess the proposed IHNN estimator on (i) a numerical example of two interconnected Hammerstein subsystems and (ii) a real-world hydraulic process. In both cases, we compare against a conventional RELS baseline following [2].

It should be noted that the present validation focuses on scenarios with white Gaussian measurement noise and does not yet include dedicated simulation experiments under colored-noise or explicit time-delay conditions. Therefore, the results reported in this section should be interpreted as an evaluation of the proposed IHNN method under stochastic but delay-free white-noise settings, while robustness to colored disturbances and delayed interconnections remains an important direction for future validation.

Overall, the simulation results confirm the effectiveness of the proposed IHNN method in both study cases. In the two-subsystem numerical example, the IHNN reduces the RMSE from 0.031 to 0.018 for subsystem S_1 and from 0.028 to 0.016 for subsystem S_2, corresponding to improvements of about 42% and 43%, respectively. The prediction-error variance is also reduced from 5.8×10^{-4} to 2.3×10^{-4} for S_1 and from 5.1×10^{-4} to 1.9×10^{-4} for S_2, while the parameter convergence time decreases from 142 to 85 samples for S_1 and from 135 to 78 samples for S_2. In the hydraulic-process application, the IHNN achieves RMSE reductions of about 35.7%, 37.5%, and 34.6% for Tanks 1–3, respectively, together with an error-variance reduction of about 60.4%. These accuracy gains are obtained at the expense of a moderate increase in computational time, which remains acceptable given the clear improvement in identification performance.

4.1. Computational complexity and implementation environment

To make the runtime comparison more transparent, we briefly analyze the computational complexity of the proposed IHNN algorithm and the RELS baseline. Let d_i denote the dimension of the regression vector $\boldsymbol{\varphi}_i(k)$ for subsystem S_i, and let W_i^{NN} denote the total number of trainable NN parameters (weights and biases) involved in the nonlinear blocks associated with subsystem S_i.

For the RELS baseline, the dominant cost at each iteration comes from the recursive covariance/gain update and the parameter update. Using the standard matrix–vector implementation of recursive least squares, the per-sample cost for subsystem S_i is of order

$$C_{\mathrm{RELS}}^{(i)}(k) = O(d_i^2), \tag{4.1}$$

since the main operations involve products of the form $\mathbf{P}_i(k-1)\boldsymbol{\varphi}_i(k)$ and rank-one covariance corrections. Hence, for $\mathcal{N}$ interconnected subsystems and K processed samples, the overall complexity scales as

$$C_{\mathrm{RELS}}(K) = O\left(K \sum_{i=1}^{\mathcal{N}} d_i^2\right). \tag{4.2}$$

For the proposed IHNN, the linear recursive update has the same order $O(d_i^2)$, but an additional cost is introduced by the forward and backward propagation through the NN nonlinear blocks. For a feedforward network, the dominant arithmetic cost is proportional to the number of trainable connections; therefore, one training step is of order $O(W_i^{\mathrm{NN}})$. As a result, the per-sample complexity of the hybrid IHNN update for subsystem S_i can be expressed as

$$C_{\mathrm{IHNN}}^{(i)}(k) = O(d_i^2 + W_i^{\mathrm{NN}}), \tag{4.3}$$

and the total complexity over $\mathcal{N}$ subsystems and K samples becomes

$$C_{\mathrm{IHNN}}(K) = O\left(K \sum_{i=1}^{\mathcal{N}} (d_i^2 + W_i^{\mathrm{NN}})\right). \tag{4.4}$$

In the homogeneous case where $d_i = d$ and $W_i^{\mathrm{NN}} = W$ for all subsystems, the above expressions simplify to $O(K\mathcal{N}d^2)$ for RELS and $O(K\mathcal{N}(d^2+W))$ for IHNN. This explains why the proposed method requires additional computation time: The recursive least-squares component has the same asymptotic order in both methods, while the NN approximation adds an extra cost proportional to the size of

the nonlinear model. The measured runtime increase of about 50–65% reported in the experiments is therefore consistent with the additional forward/backward neural computations required by IHNN.

To improve reproducibility, all simulations were executed on an Intel Core Ultra 9 platform equipped with 32 GB RAM and 1 TB SSD under Windows 11, using Python 3.14.3 (latest stable version at the time of writing) and the required scientific computing libraries. The reported runtimes correspond to wall-clock execution times averaged over 10 independent runs under the same implementation settings.

4.2. Numerical example

Setup. We consider a large-scale system with two interconnected nonlinear subsystems modeled by discrete-time Hammerstein systems [2]: Parameters and constants which change with time include

$$
\begin{aligned}
a_{1,1}(k) &= -0.88 + 0.02\sin(0.2k), & a_{1,2}(k) &= 0.45 + 0.03\cos(0.2k),\\
b_{1,2}(k) &= 0.32 + 0.03\sin(0.2k), & b_{12,2}(k) &= 0.33 + 0.04\sin(0.2k),\\
a_{2,1}(k) &= -0.85 + 0.02\sin(0.2k), & a_{2,2}(k) &= 0.40 + 0.03\cos(0.2k),\\
b_{2,2}(k) &= 0.42 + 0.03\sin(0.2k), & b_{21,2}(k) &= 0.44 + 0.04\sin(0.2k),
\end{aligned}
$$

with constants $\alpha_{1,1} = 0.31$, $\alpha_{1,2} = 0.22$, $\beta_{2,1} = 0.32$, $\beta_{2,2} = 0.20$, $c_{1,1} = 0.25$, $\alpha_{2,1} = 0.32$, $\alpha_{2,2} = 0.20$, $\beta_{1,1} = 0.31$, $\beta_{1,2} = 0.22$, $c_{2,1} = 0.28$. Inputs are Pseudo-Random Binary Sequence (PRBS) signals $u_i(k) \in [-1.5, 1.5]$; measurement noise is zero-mean white with variances $\sigma_1^2 = 0.0952$ and $\sigma_2^2 = 0.0791$. According to the architecture-selection procedure described in Section 3.4.2, the IHNN uses one hidden layer with 8 neurons for each input and output NN, together with a second-order linear block. This compact structure was found sufficient for the numerical benchmark, which exhibits moderate nonlinear complexity while still requiring accurate tracking of interconnection effects.

Results. Figure 3 presents the tracking of time-varying parameters for S_1. The IHNN provides less delay and overshoot compared to RELS.

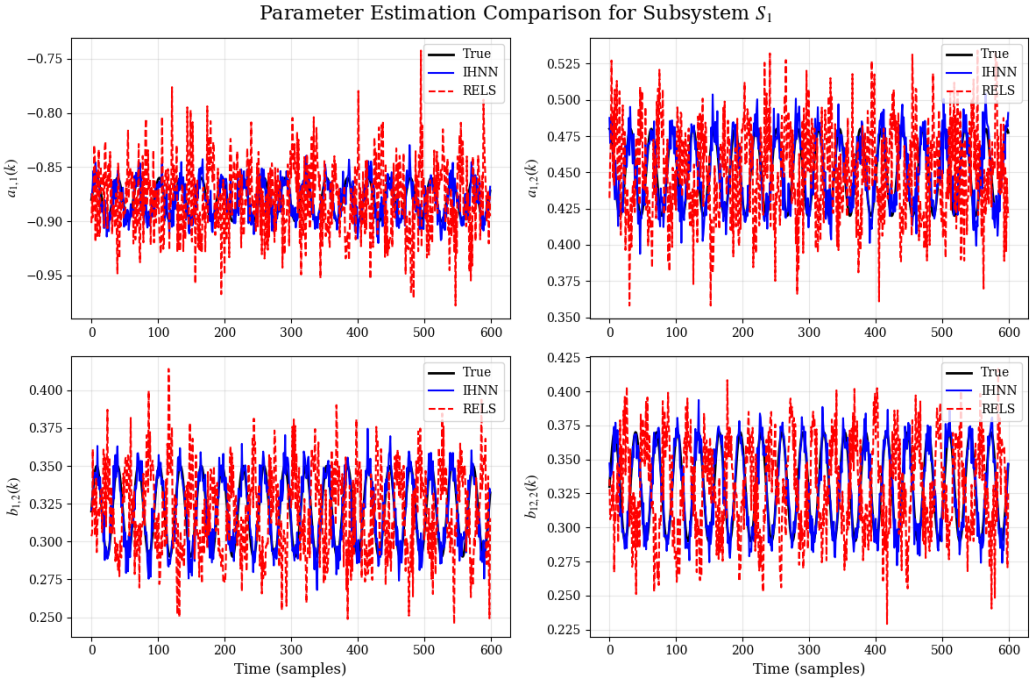

Figure 3. Parameter estimation comparison for subsystem S_1: IHNN vs. RELS.

Figure 4 can be used to point out that improved transient tracking and noise resilience is achieved with S_2.

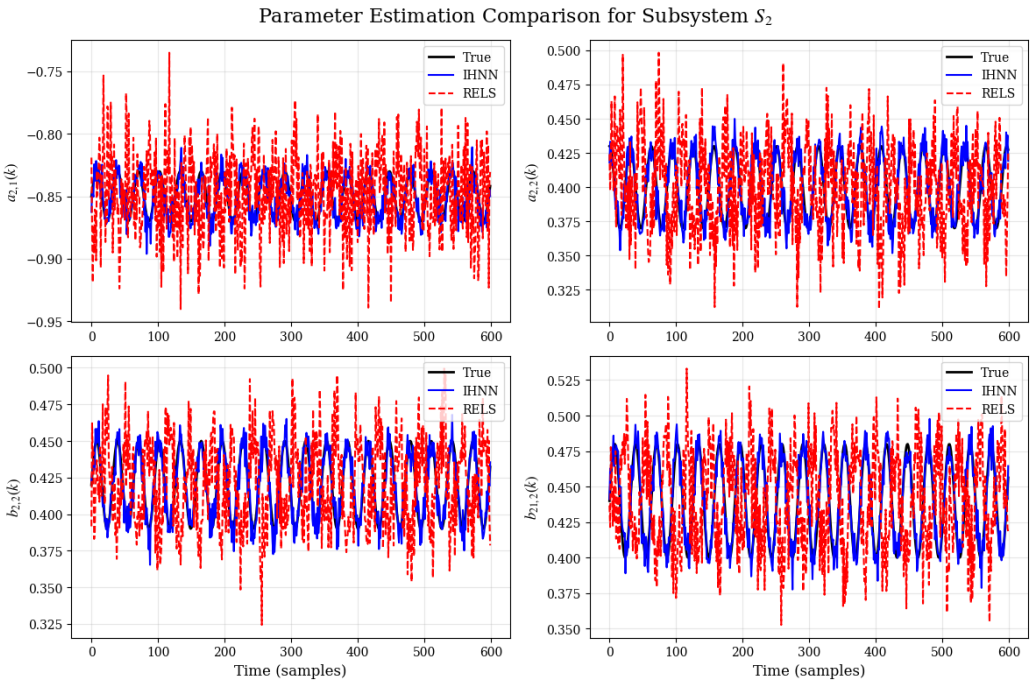

Figure 4. Parameter estimation comparison for subsystem S_2: IHNN vs. RELS.

Prediction errors (Figure 5) remain consistently smaller for IHNN across the horizon.

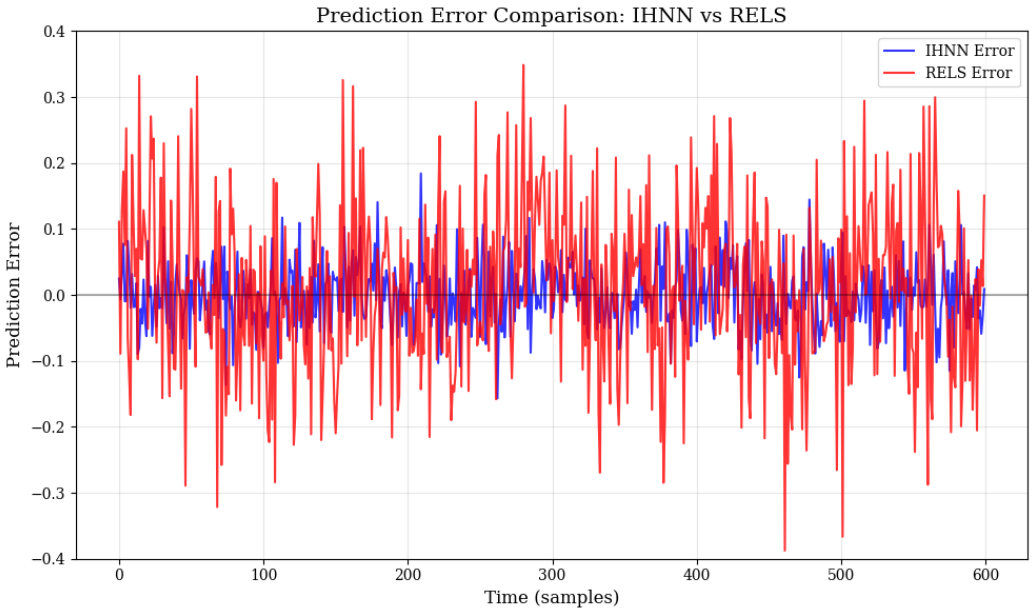

Figure 5. Prediction error comparison: IHNN vs. RELS.

Quantitative comparison. Table 1 summarizes key metrics. The IHNN lowers RMSE by $\approx 42\%$ on S_1 and $\approx 43\%$ on S_2, reduces error variance by $\approx 60\%$ (S_1) and $\approx 63\%$ (S_2), and achieves faster convergence in sample count.

Table 1. Performance metrics comparison between IHNN and RELS.

Metric	Unit	IHNN (S_1)	RELS (S_1)	IHNN (S_2)	RELS (S_2)
RMSE	–	0.018	0.031	0.016	0.028
Maximum deviation	–	0.042	0.067	0.038	0.062
Mean prediction error	–	0.0021	0.0047	0.0018	0.0042
Error variance	$\times 10^{-4}$	2.3	5.8	1.9	5.1
Parameter convergence time	samples	85	142	78	135

Discussion. The gains come due to NN-based approximation the static nonlinearities and recursive estimation of the linear dynamics. In this setting, the IHNN requires approximately 50% more computation time than RELS, which is a reasonable trade-off for the significant improvements in accuracy and robustness achieved for interconnected noisy systems.

4.3. Hydraulic process application

Setup. On this three-tank hydraulic process, with strong nonlinearities and interconnections we evaluate the IHNN. We use 1000 samples with measurement noise of 3% from the operating range. Following the architecture-selection procedure described in Section 3.4.2, the hydraulic-process case uses a richer neural structure, namely an input NN with two hidden layers of 10 and 6 neurons, an output NN with one hidden layer of 8 neurons, and a second-order linear block. This slightly larger architecture was retained because the hydraulic process exhibits stronger nonlinearities and more pronounced subsystem interactions than the numerical benchmark, thus requiring a higher

approximation capacity. Learning takes place with adaptive-rate scheme described in Section 3.4.3 with initial rate $\eta_0 = 0.03$ and decay $\epsilon = 0.002$. The RELS comparator is configured according to the information reported in [2].

Results. Figures 6–8 report oil-level estimates for Tanks 1–3. The IHNN monitors the setpoint changes and transients with less lag and less overshoot than RELS. Figures 9–11 show the respective prediction errors, under the consistent lower amplitudes for the IHNN throughout the time horizon.

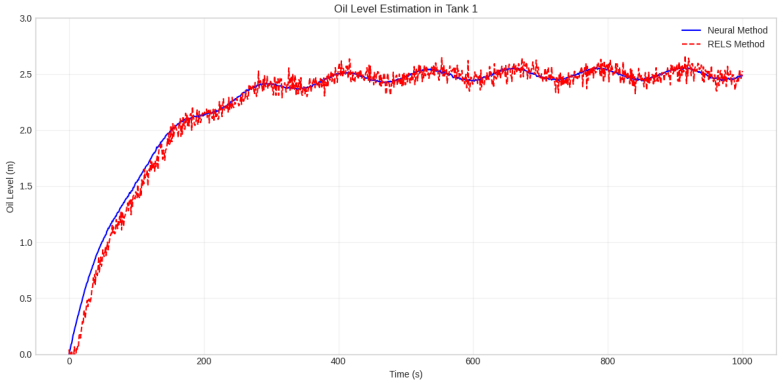

Figure 6. Oil level estimation in Tank 1: IHNN vs. RELS.

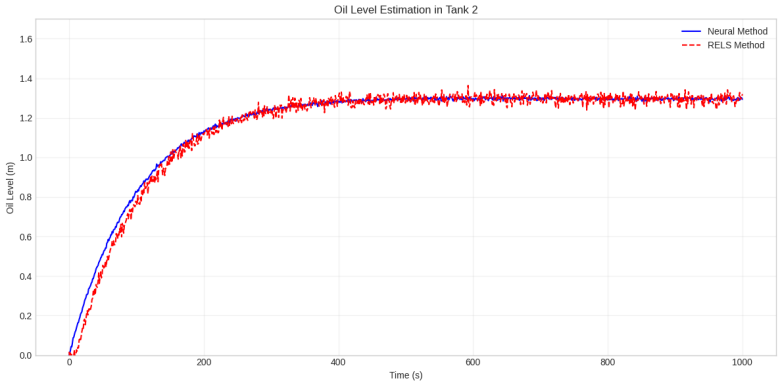

Figure 7. Oil level estimation in Tank 2: IHNN vs. RELS.

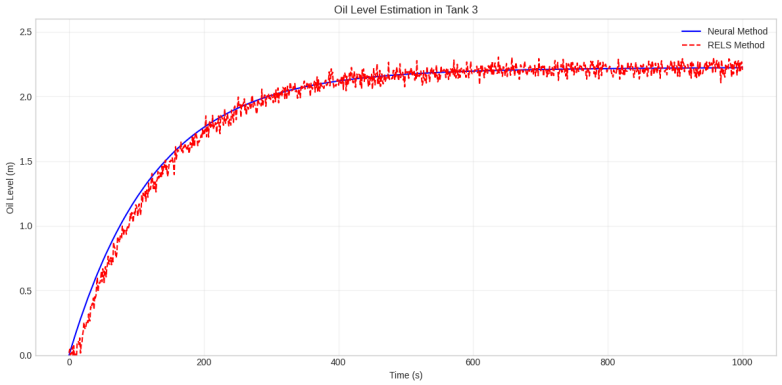

Figure 8. Oil level estimation in Tank 3: IHNN vs. RELS.

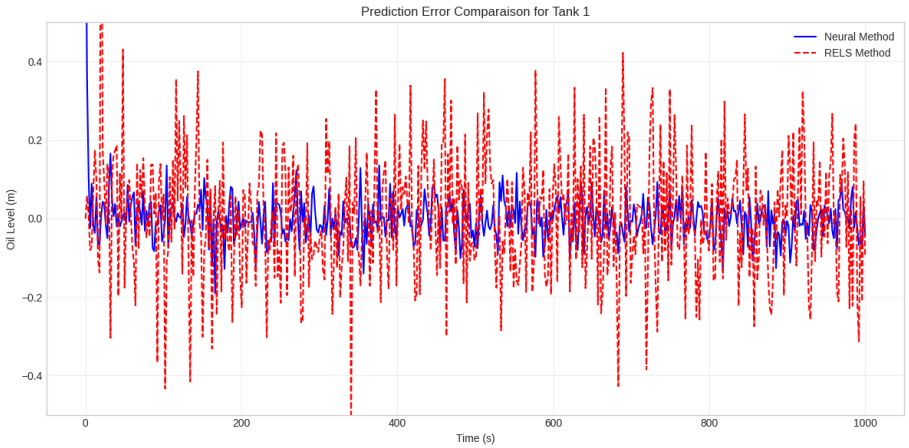

Figure 9. Prediction error comparison for Tank 1.

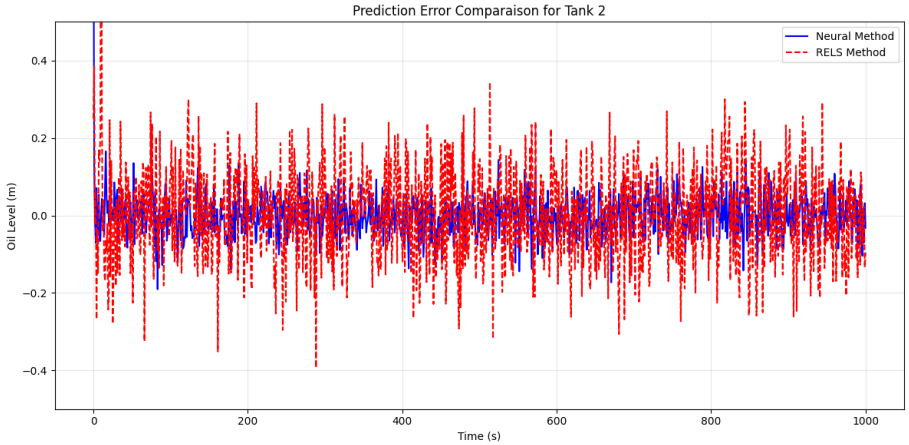

Figure 10. Prediction error comparison for Tank 2.

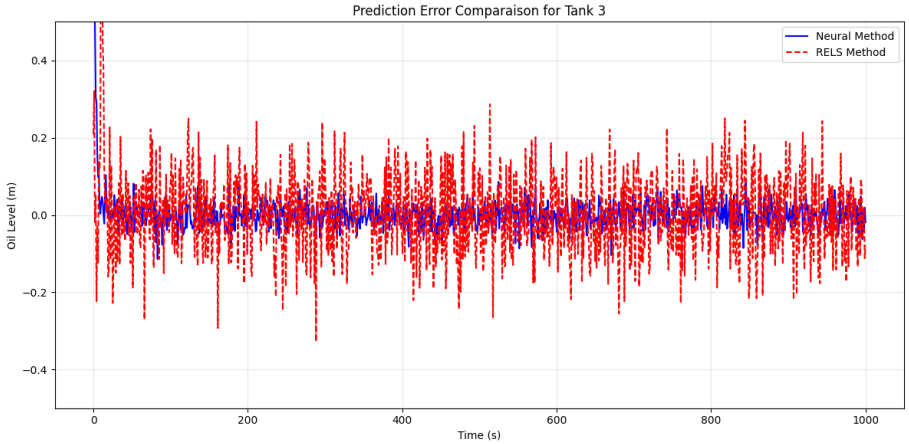

Figure 11. Prediction error comparison for Tank 3.

Quantitative comparison. Across all tanks, Across all tanks, IHNN reduces RMSE by

approximately 35–38% compared to RELS, lowers prediction-error variance by about 60%, and also improves the maximum deviation and mean absolute error.

Discussion. As shown in Table 2, the IHNN is noticeably better than RELS with this nonlinear, interconnected process and has a superior rejection of noise and transient behaviour. The computational overhead (roughly 55–65% longer training) is balanced out by accuracy gains which are particularly valuable when it comes to safety-critical level regulation in hydraulic systems.

Table 2. Hydraulic process: IHNN vs. RELS performance.

Metric	Unit	IHNN	RELS	Improvement
RMSE (Tank 1)	m	0.018	0.028	35.7%
RMSE (Tank 2)	m	0.015	0.024	37.5%
RMSE (Tank 3)	m	0.017	0.026	34.6%
Maximum deviation	m	0.042	0.065	35.4%
Mean absolute error	m	0.014	0.022	36.4%
Error variance	$\times 10^{-4}$	2.1	5.3	60.4%

4.4. Broader comparative analysis with recent Hammerstein identification methods

In addition to the direct numerical comparison with the conventional RELS baseline, it is useful to position the proposed IHNN estimator with respect to other recent Hammerstein identification strategies reported in the literature. Recent NN-based or hybrid identification methods include: (i) a neural-fuzzy-model/ARMAX approach for multi-input multi-output Hammerstein systems with measurement noises using combined signals and data-filtering recursive least squares [24]; (ii) a neural-fuzzy-model/ARX identification approach for Hammerstein systems with noisy output measurements using correlation-function least squares and improved particle swarm optimization [25]; (iii) an SSAE-based Hammerstein identification framework combined with ARMAX modeling and multi-innovation RELS [8]; and (iv) a deep-GRU-network-based parameter estimation method for multi-input multi-output Hammerstein nonlinear systems [9].

Compared with these methods, the proposed IHNN strategy exhibits several distinguishing characteristics. First, unlike the above approaches, which mainly focus on single Hammerstein systems or standard MIMO Hammerstein configurations, the present work explicitly addresses large-scale interconnected Hammerstein systems with stochastic disturbances. Second, the IHNN retains the interpretable Hammerstein block-oriented structure while using NNs for the static nonlinearities and recursive least-squares-type estimation for the linear dynamic part. Third, in contrast to purely algorithmic identification schemes, the proposed method is accompanied by a Lyapunov-based convergence analysis of the hybrid learning procedure. Therefore, the broader comparison highlights that the main advantage of the IHNN is not only the reduction of prediction error with respect to RELS under the adopted benchmarks, but also its suitability for interconnected stochastic settings where structural coupling and recursive online adaptation must be handled simultaneously.

Table 3. Qualitative comparison of the proposed IHNN with representative Hammerstein identification methods.

Method	Nonlinear block model	Linear/dynamic estimation	Main scope / limitation relative to IHNN
RELS baseline [2]	Polynomial/block-oriented approximation	RELS	Direct benchmark used in this paper; does not exploit neural approximators for nonlinear blocks.
Neural-fuzzy + combined signals [24]	Neural fuzzy model (NFM)	Correlation analysis + data-filtering recursive least squares with ARMAX	Handles MIMO Hammerstein systems with measurement noise, but does not target large-scale interconnected stochastic structures.
Neural-fuzzy + noisy outputs [25]	Neural fuzzy model (NFM)	CF-LS + improved PSO with ARX	Effective for noisy Hammerstein systems, but designed for single-system identification rather than interconnected large-scale settings.
SSAE-based Hammerstein [8]	SSAE	MI-RELS with ARMAX	Strong deep nonlinear representation, but not developed for interconnected stochastic Hammerstein subsystems.
Deep-GRU Hammerstein [9]	Deep GRU network	Deep recurrent parameter estimation	Targets MIMO Hammerstein identification, but the interconnection-aware hybrid recursive structure considered here is not explicitly addressed.
Proposed IHNN	NN approximation preserving Hammerstein structure	Backpropagation + recursive least squares with forgetting/covariance adaptation	Specifically developed for large-scale interconnected Hammerstein stochastic systems, with Lyapunov-based convergence analysis and validation on benchmark and hydraulic-process cases.

Accordingly, the direct RELS-based simulations reported above provide the quantitative benchmark under identical experimental conditions, while Table 3 broadens the evaluation by situating the proposed IHNN method among recent Hammerstein identification strategies in terms of modeling structure, estimation mechanism, and applicability domain.

5. Conclusions

This paper proposed an IHNN framework for the identification of large-scale interconnected Hammerstein systems operating under stochastic disturbances. The proposed approach preserved the block-oriented Hammerstein structure by combining neural-network approximators for the static nonlinear blocks with recursive least-squares-based estimation for the linear dynamic part. A Lyapunov-based analysis was further developed to establish convergence and stability conditions for the hybrid learning algorithm.

The effectiveness of the proposed method was demonstrated through a two-subsystem numerical benchmark and a hydraulic-process case study. In comparison with a conventional RELS baseline, the IHNN approach achieved clear improvements in identification accuracy, including RMSE reductions of about 35–43%, substantial decreases in prediction-error variance, and faster parameter convergence, while requiring a moderate increase in computational time. These results show that the proposed IHNN method provides an effective and practical solution for identifying noisy interconnected Hammerstein systems with strong couplings and time-varying behavior.

Future work will focus on distributed and event-triggered implementations for very large-scale systems, robustness enhancement against colored noise, delays, and outliers, and integration with fault diagnosis and security-aware monitoring for safety-critical applications. In particular, although the proposed framework is designed to accommodate more challenging disturbance conditions, explicit validation under colored-noise and time-delay scenarios will be addressed in future work to further assess its robustness in practical industrial environments.

Author contributions

Rihab Issaoui: Conceptualization, software, validation, writing – original draft; Mourad Elloumi: Methodology, formal analysis, writing – review & editing; Imed Bouzida: Investigation, data curation, visualization; Omar Naifar: Methodology, supervision, project administration, funding acquisition, writing – review & editing. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was supported by Abu Dhabi University. The authors gratefully acknowledge the financial support provided. The authors also express their sincere appreciation to Abu Dhabi University for its continuous support of research activities.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. M. Elloumi, H. Gassara, O. Naifar, An overview on modelling of complex interconnected nonlinear systems, *Math. Probl. Eng.*, **2022** (2022), 4789405. https://doi.org/10.1155/2022/4789405
2. M. Elloumi, S. Kamoun, An iterative parametric estimation method for hammerstein large-scale systems: A simulation study of hydraulic process, *Int. J. Simul. Process Model.*, **11** (2016), 207–219. https://doi.org/10.1504/IJSPM.2016.078501
3. F. Li, X. Sun, T. Wang, R. Liu, Identification methodology for MIMO hammerstein nonlinear model with process noise, *Trans. Inst. Meas. Control*, **47** (2025), 2190–2202. https://doi.org/10.1177/01423312241276072
4. M. Zhang, L. Jia, F. Li, Identification of the hammerstein system with time delay and colored noise, In: *2025 IEEE 14th Data driven control and learning systems (DDCLS)*, IEEE, 2025, 1815–1819. https://doi.org/10.1109/DDCLS66240.2025.11065333
5. W. Wu, J. Hu, Z. Zhu, F. Zhang, J. Xu, C. Wang, Deterministic learning-based neural identification and knowledge fusion, *Neural Netw.*, **169** (2024), 165–180. https://doi.org/10.1016/j.neunet.2023.10.004
6. P. Saha, S. Dash, S. Mukhopadhyay, Physics-incorporated convolutional recurrent neural networks for source identification and forecasting of dynamical systems, *Neural Netw.*, **144** (2021), 359–371. https://doi.org/10.1016/j.neunet.2021.07.004
7. M. F. da Silva, M. T. da Silva, P. R. Barros, G. A. Junior, Enhanced hammerstein models identification using multiple hidden layers neural networks, *Circuits Syst. Signal Process.*, **44** (2025), 4669–4703. https://doi.org/10.1007/s00034-025-03029-5
8. F. Li, L. Song, T. Wang, R. Liu, Parameter identification for nonlinear hammerstein models with stacked sparse autoencoder network, *Eng. Appl. Artif. Intell.*, **163** (2026), 113002. https://doi.org/10.1016/j.engappai.2025.113002
9. F. Li, Y. Zhu, C. Zhao, R. Liu, Parameter estimation of multi-input multi-output hammerstein nonlinear system with deep GRU networks, *Int. J. Robust Nonlinear Control*, **36** (2026), 4792–4810. https://doi.org/10.1002/rnc.70461
10. B. Li, H. Wang, X. Tan, Q. Li, J. Chen, X. Qiu, Adaptive heterogeneous graph reasoning for relational understanding in interconnected systems, *J. Supercomput.*, **81** (2025), 112. https://doi.org/10.1007/s11227-024-06623-7
11. Y. Tang, J. Ding, Y. Chen, Y. Gao, A. Jiang, C. Wang, Anxiety disorder identification with biomarker detection through subspace-enhanced hypergraph neural network, *Neural Netw.*, **187** (2025), 107293. https://doi.org/10.1016/j.neunet.2025.107293
12. G. Yang, X. Yang, Data-based decentralized control of nonlinear-constrained interconnected systems using reinforcement learning, *Neural Netw.*, **191** (2025), 107780. https://doi.org/10.1016/j.neunet.2025.107780

13. A. B. Makhlouf, O. Naifar, On the barbalat lemma extension for the generalized conformable fractional integrals: Application to adaptive observer design, *Asian J. Control*, **25** (2023), 563–569. https://doi.org/10.1002/asjc.2797
14. E. B. Alaia, S. Dhahri, O. Naifar, A gradient-based optimization algorithm for optimal control problems with general conformable fractional derivatives, *IEEE Access*, **13** (2025), 140270–140281. https://doi.org/10.1109/ACCESS.2025.3595958
15. J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, H. El Makhtoum, OMIC: A bagging-based ensemble learning framework for large-scale IoT intrusion detection, *J. Fut. Artif. Intell. Tech.*, **1** (2025), 401–416. https://doi.org/10.62411/faith.3048-3719-63
16. C. W. Lynn, Q. Yu, R. Pang, S. E. Palmer, W. Bialek, Exact minimax entropy models of large-scale neuronal activity, *Phys. Rev. E*, **111** (2025), 054411. https://doi.org/10.1103/PhysRevE.111.054411
17. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, 249–256.
18. R. Meng, C. Hua, K. Li, Q. Li, Dynamic events-based adaptive NN output feedback control of interconnected nonlinear systems under general output constraint, *Neural Netw.*, **188** (2025), 107452. https://doi.org/10.1016/j.neunet.2025.107452
19. A. Sahoo, V. Narayanan, Differential-game for resource aware approximate optimal control of large-scale nonlinear systems with multiple players, *Neural Netw.*, **124** (2020), 95–108. https://doi.org/10.1016/j.neunet.2019.12.031
20. N. T. Nguyen, Least-squares parameter identification, In: *Model-reference adaptive control*, Cham: Springer, 2018, 125–149. https://doi.org/10.1007/978-3-319-56393-0_6
21. B. Lai, D. S. Bernstein, Generalized forgetting recursive least squares: Stability and robustness guarantees, *IEEE Trans. Autom. Control*, **69** (2024), 7646–7661. https://doi.org/10.1109/TAC.2024.3394351
22. L. Behera, S. Kumar, A. Patnaik, On adaptive learning rate that guarantees convergence in feedforward networks, *IEEE Trans. Neural Netw.*, **17** (2006), 1116–1125. https://doi.org/10.1109/TNN.2006.878121
23. Z. Man, H. R. Wu, S. Liu, X. Yu, A new adaptive backpropagation algorithm based on Lyapunov stability theory for neural networks, *IEEE Trans. Neural Netw.*, **17** (2006), 1580–1591. https://doi.org/10.1109/TNN.2006.880360
24. F. Li, X. Sun, Q. Cao, Parameter learning of multi-input multi-output hammerstein system with measurement noises utilizing combined signals, *Int. J. Adapt. Control Signal Process.*, **39** (2024), 1416–1433. https://doi.org/10.1002/acs.3857
25. Q. Zha, F. Li, R. Liu, Identification of the hammerstein nonlinear system with noisy output measurements, *Control Theory Technol.*, **22** (2024), 203–212. https://doi.org/10.1007/s11768-024-00196-9

AIMS Press