IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 14, 2022, accepted May 25, 2022, date of publication May 30, 2022, date of current version June 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3179421

Diffusion Hammerstein Spline Adaptive Filtering Based on Orthogonal Gradient Adaptive Algorithm

SUCHADA SITJONGSATAPORN
Department of Electronic Engineering, Faculty of Engineering and Technology, Mahanakorn Institute of Innovation (MII), Mahanakorn University of Technology, Nong Chok, Bangkok 10530, Thailand
e-mail: ssuchada@mut.ac.th

ABSTRACT In this paper, we propose a class of nonlinear diffusion filtering based on Hammerstein function with the spline adaptive filter (HSAF) implemented by normalised version of orthogonal gradient adaptive (NOGA) algorithm over the distributed network. Diffusion adaptation algorithm approximates a variable vector with the help of a network of agents using a joint optimisation on the sum of cost function. A HSAF comprises of memoryless function during learning by interpolating polynomials with respect to the linear filter. We derive a diffusion adaptation framework on HSAF motivated from NOGA algorithm; called DHSAF-NOGA. There are two types of adaptive diffusion strategies with the *combine-then-adapt (CTA)* algorithm and the *adapt-then-combine (ATC)* algorithm that are considered and implemented by DHSAF-NOGA algorithm. The network stability and performance over mean square error networks is derived. Experiment results depict that proposed CTA-DHSAF-NOGA and ATC-DHSAF-NOGA algorithms can learn robustly underlying the nonlinear Hammerstein model compared with a non-cooperative solution and existing techniques.

INDEX TERMS Spline adaptive filtering, Hammerstein model, diffusion strategy, orthogonal gradient adaptive algorithm.

I. INTRODUCTION

Nonlinear spline-based adaptive filtering (SAF) has been interested for nonlinear system identification and modelling [1]–[5]. SAF structure consists of the adaptive linear finite impulse response (FIR) and the adaptive look-up table (LUT) by spline interpolation based on the least mean square (LMS) algorithm [1] and normalised LMS (NLMS) [2] in the nonlinear structure. In order to model with the gradient method, an adaptive combination of SAF based on LMS algorithm has been demonstrated in a non-stationary environment [3]. In [4], the authors have studied the SAF based on infinite impulse response (IIR) against the impulsive noise with the low computational complexity. As concerned with the fast convergence, an orthogonal gradient adaptive algorithm based on SAF framework has been revealed with the low disadjustment [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

According to the multitask case in the real world application [6]–[8], the authors in [6] has described the adaptation and learning of nonlinear filtering over the distributed network. A distributed with sparsity-aware adaptive algorithm [7] has been proposed to verify the Voterra system that can provide a good performance in the wireless sensor network (WSN). In [8], a censored-regression has been introduced to compensate the bias estimation over the nonlinear WSN.

To deal with the impulsive noise scenario, a diffusion normalised least mean M-estimate algorithm [9] has been equipped for learning capability over the distributed network. In [10], the development of robust diffusion recursive least square algorithm has been designed with a side data from neighbouring nodes in the distributed network. In terms of robust algorithm against the impulsive noise, the robust Geman-McClure SAF has been obtained in [11]. In [12], the authors has been organised the combined nonlinear adaptive filters in a network with two subnetworks by linking the nodes in each subnetwork.

Hammerstein architecture is the block-oriented nonlinear model consisting of a cascade topology with memoryless functions in the linear time invariant (LTI) model called as nonlinear-linear model [13]. Hammerstein system has been used as a kernel method modelled for the echo path in a nonlinear acoustic echo cancellation [14], [15]. Hammerstein SAF (HSAF) has been proposed on the stochastic gradient scheme in the impulsive noise environment [16]–[18]. Stability performance of HSAF has been proved by considering the steepest descent method [17]. In [18], HSAF-based self-interference digital cancellation has been evaluated inband full-duplex system at 2.4GHz.

As stated to standard linear filtering, the diffusion filtering is the extended version of linear filtering in case of distributed optimisation by local upldate as gradient descent method with the global steps, while the data from each node is processed and diffused by its neighbours across the network [19], [20]. Distributed estimation is widely used with the diffusion adaptive network in the field of signal processing and control engineering [21]–[23]. Distributed active noise canceller has been implemented over the realistic scenario of wireless acoustic sensor network with the different nodes and interaction between them [24], [25]. Simulation results proved that the distributed algorithm can achieve good performance in both centralised and distributed practical nonlinear acoustic sensor network. Distributed Hammerstein filters [26] have been developed and performed in robustness for estimation. In [27], a diffused version of recursive least square algorithm based on Cholesky factorisation has been performed in the system identification. Based on the SAF, a diffusion spline adaptive filtering has been introduced using the gradient descent method [28]. The simulation results are shown to outperform robustly the nonlinear model compared with the non-cooperative SAF.

In order to improve the convergence properties, the adaptive orthogonal gradient-based algorithm has been presented, which can provide with the development of simple and robust adaptive filtering across the wide range of input environments [29], [30]. Motivation for using the orthogonal gradient-based algorithm is concerned with the development of simple and robust adaptive filtering by the normalised orthogonal gradient adaptive (NOGA) algorithm.

As a remark on the fast convergence of NOGA algorithm, we introduce a diffused version of SAF framework with Hammerstein model named HSAF. To the best of my knowledge, a proposed diffusion Hammerstein spline-based adaptive filtering based on normalised orthogonal gradient adaptive (DHSAF-NOGA) algorithm employs to achieve the robustness over the distributed network.

The rest of paper is arranged as follows. Section II presents the framework of proposed HSAF based on NOGA algorithm in details. Section V introduces the diffusion model of HSAF architecture based on NOGA algorithm. Section VI shows how to derive the network stability and performance over mean square error networks. Subsequently, Section VII shows the simulation results. Finally, Section VIII concludes this paper.

The notations are used throughout this paper as follows. Normal and boldface lowercase letters denote as scalars and vectors, where boldface uppercase letters are matrices. The operators $E\{\cdot\}$, $(\cdot)^T$ and $\lfloor\cdot\rfloor$ stand for the expectation, transpose and floor operators, respectively.

II. PROPOSED HAMMERSTEIN SPLINE ADAPTIVE FILTERING BASED ON NORMALISED ORTHOGONAL GRADIENT ADAPTIVE ALGORITHM

In this paper, we denote the input x_n at index n to HSAF [13] and $\mathbf{x}_n = [\ x_n, \ldots, x_{n-M+1}\ ]^T$ at M samples. We assume to be dealing with the real inputs and an unknown Wiener model generates the desired response adding with the Gaussian noise d_n as

$$d_n = f(\mathbf{w}_o^T\,\mathbf{x}_n) + \nu_n, \tag{1}$$

where $\mathbf{w}_o \in \mathbb{R}^M$ are the linear weight, $\mathbf{x}_n$ is the input vector. $f(\cdot)$ is a nonlinear function and $\nu_n \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian noise term.

A spline function is a polynomial defined by a set of Q control point. We suppose that the control points coefficient or non-linear weight coefficient $\mathbf{q}_i$ is uniformly distributed as $\mathbf{q}_{i+1,n} = \mathbf{q}_{i,n} + \Delta x$, for a fixed $\Delta x \in \mathbb{R}$.

We refer to the linear filter input $\mathbf{s}_n$, that is a uniform spline interpolation of adaptive control points in a look-up table (LUT). By considering that $\mathbf{s}_n$ is a function of two local parameters u_n and i which depends on x_n as

$$u_n = \frac{x_n}{\Delta x} - \left\lfloor \frac{x_n}{\Delta x} \right\rfloor, \tag{2}$$

$$i = \left\lfloor \frac{x_n}{\Delta x} \right\rfloor + \frac{Q-1}{2}, \tag{3}$$

where Δx is the uniform space between the control points coefficients, Q is the total number of control points and $\lfloor\cdot\rfloor$ is a floor operator.

Similarly, a HSAF computes the output in a two-step process as shown in Figure 1. First, it performs a spline interpolation on the nonlinear weight coefficient $\mathbf{q}_{i,n}$ as [13]

$$\mathbf{s}_n = \mathbf{u}_n^T\,\mathbb{C}\,\mathbf{q}_{i,n}, \tag{4}$$

where the spline basis matrix $\mathbb{C} \in \mathbb{R}^{4\times 4}$ is a pre-computed matrix. The input vector $\mathbf{u}_n \in \mathbb{R}^{4\times 1} = [\ u_n^3\ \ u_n^2\ \ u_n\ \ 1\ ]^T$ and $\mathbf{q}_{i,n} \in \mathbb{R}^{4\times 1} = [\ q_i\ \ q_{i+1}\ \ q_{i+2}\ \ q_{i+3}\ ]^T$.

Then, the final output y_n is calculated through the linear filtering operation $\mathbf{w}_n$ as

$$y_n = \mathbf{w}_n^T\,\mathbf{s}_n. \tag{5}$$

We consider to minimise the expected mean square error (MSE) constraint for both the nonlinear and linear part of HSAF given by [17]

$$\mathbb{J}(\mathbf{q}_n\,\mathbf{w}_n) = \frac{1}{2}\,\mathbb{E}\{\ e_n^2\ \}, \tag{6}$$

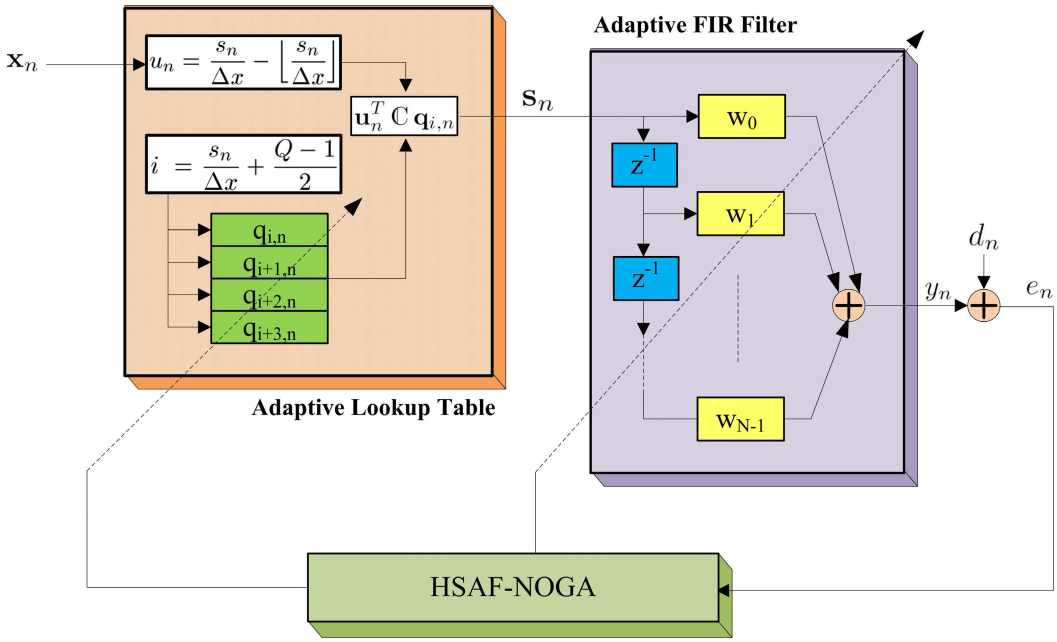

FIGURE 1. Proposed HSAF structure based on NOGA algorithm.

where $\mathbb{E}\{\cdot\}$ denotes an expectation operator and e_n is *a priori* error as

$$e_n = d_n - y_n = d_n - \mathbf{w}_n^T\,\mathbf{s}_n, \tag{7}$$

where $\mathbf{s}_n$ is given in (4).

For the minimisation of the cost function in (6) with respect to (w.r.t) the linear weight coefficient $\mathbf{w}_n$, we apply the stochastic gradient adaptation at index n as follows.

$$\begin{aligned}\nabla_q \mathbb{J}(\mathbf{q}_n, \mathbf{w}_n) &= -e_n \frac{\partial y_n}{\partial \mathbf{s}_n}\frac{\partial \mathbf{s}_n^T}{\partial \mathbf{q}_{i,n}} \\ &= -\mathbb{C}\,\mathbf{u}_n^T\,\mathbf{w}_n\,e_n.\end{aligned} \tag{8}$$

For the derivative of (6) w.r.t the control points $\mathbf{q}_{i,n}$, we apply the chain rule as

$$\nabla_w \mathbb{J}(\mathbf{q}_n, \mathbf{w}_n) = e_n \frac{\partial y_n}{\partial \mathbf{w}_n} = -\mathbf{s}_n\,e_n. \tag{9}$$

Further, an iterative learning of the control points $\mathbf{q}_{i,n}$ can be expressed as

$$\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_q\,\mathbf{d}_{q_n}, \tag{10}$$

where the learning rate μ_q is a small constant and $\mathbf{d}_{q_n}$ is the directional vector for the non-linear weight coefficient $\mathbf{q}_{i,n}$ as

$$\mathbf{d}_{q_{n+1}} = \lambda_q\,\mathbf{d}_{q_n} - \mathbf{g}_{q_n}, \tag{11}$$

where λ_q and $\mathbf{g}_{q_n}$ denote the forgetting-factor and the negative gradient of non-linear weight $\mathbf{q}_{i,n}$ by deriving the cost function in (6) w.r.t $\mathbf{q}_{i,n}$ as

$$\begin{aligned}\mathbf{g}_{q_{n+1}} &= \lambda_q\,\mathbf{g}_{q_n} - \frac{\partial \mathbb{J}(\mathbf{q}_n, \mathbf{w}_n)}{\partial \mathbf{q}_{i,n}} \\ &= \lambda_q\,\mathbf{g}_{q_n} + \mathbb{C}\,\mathbf{u}_n^T\,\mathbf{w}_n\,e_n.\end{aligned} \tag{12}$$

Following [5], the forgetting factor λ_q places on the orthogonal projection of gradient vector $\mathbf{g}_{q_n}$ and the previous directional vector $\mathbf{d}_{q_{n-1}}$ as

$$\lambda_q = \frac{\mathbf{d}_{q_{n-1}}^T\,\mathbf{g}_{q_n}}{\mathbf{d}_{q_n}^T\,\mathbf{d}_{q_{n-1}}}. \tag{13}$$

It is noted that an idea of the orthogonal gradient adaptive algorithm is how to update the forgetting-factor parameter using the orthogonal projection of gradient vector and directional vector shown in (11), (12) and (13), respectively.

In a similar fashion, an iterative learning of the linear tap-weight vector $\mathbf{w}_n$ based on NOGA algorithm can be written as

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_w\,\mathbf{d}_{w_n}, \tag{14}$$

where the learning rate μ_w is a small constant and $\mathbf{d}_{w_n}$ is the directional vector for the linear weight $\mathbf{w}_n$ as

$$\mathbf{d}_{w_{n+1}} = \lambda_w\,\mathbf{d}_{w_n} - \mathbf{g}_{w_n}, \tag{15}$$

where λ_w and $\mathbf{g}_{w_n}$ represent the forgetting factor parameter and the negative gradient of linear weight by applying the derivation of cost function in (6) w.r.t $\mathbf{w}_n$ as

$$\begin{aligned}\mathbf{g}_{w_{n+1}} &= \lambda_w\,\mathbf{g}_{w_n} - \frac{\partial \mathbb{J}(\mathbf{q}_n, \mathbf{w}_n)}{\partial \mathbf{w}_n} \\ &= \lambda_w\,\mathbf{g}_{w_n} + \mathbf{s}_n\,e_n.\end{aligned} \tag{16}$$

where the forgetting factor λ_w can be expressed as

$$\lambda_w = \frac{\mathbf{d}_{w_{n-1}}^T\,\mathbf{g}_{w_n}}{\mathbf{d}_{w_n}^T\,\mathbf{d}_{w_{n-1}}}. \tag{17}$$

The overall of HSAF based on NOGA is summarised in Algorithm 1.

Algorithm 1: HSAF-NOGA Algorithm

Initialise: $\mathbf{w}_0 = \delta_n \cdot [1 \ 0 \ \ldots \ 0]^T$, $\mathbf{q}_0 = [1 \ 0 \ \ldots \ 0]^T$
for $n = 0, 1, \ldots$ **do**
$\mathbf{s}_n = \mathbf{u}_n^T \, \mathbb{C} \mathbf{q}_{i,n}$
$\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_q \, \mathbf{d}_{q_n}$
$\mathbf{d}_{q_{n+1}} = \lambda_q \, \mathbf{d}_{q_n} - \mathbf{g}_{q_n}$
$\mathbf{g}_{q_{n+1}} = \lambda_q \, \mathbf{g}_{q_n} + \mathbf{u}_n^T \, \mathbb{C} \, \mathbf{w}_n \, e_n$
$\lambda_q = \dfrac{\mathbf{d}_{q_{n-1}}^T \mathbf{g}_{q_n}}{\mathbf{d}_{q_n}^T \mathbf{d}_{q_{n-1}}}$
$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_w \, \mathbf{d}_{w_n}$
$\mathbf{d}_{w_{n+1}} = \lambda_w \, \mathbf{d}_{w_n} - \mathbf{g}_{w_n}$
$\mathbf{g}_{w_{n+1}} = \lambda_w \, \mathbf{g}_{w_n} + \mathbf{s}_n \, e_n$
$\lambda_w = \dfrac{\mathbf{d}_{w_{n-1}}^T \mathbf{g}_{w_n}}{\mathbf{d}_{w_n}^T \mathbf{d}_{w_{n-1}}}$
$e_n = d_n - \mathbf{w}_n^T \, \mathbf{s}_n$
end
end

III. DIFFUSION ADAPTATION STRATEGY ON HAMMERSTEIN SPLINE ADAPTIVE FILTERING

Following [19], the idea of diffusion adaptation strategies for the distributed network is to allow the multi-agent to diffuse information in the network. It can diminish the effects of stochastic gradient noise through the learning process. The concept of diffusion adaptation can be modified for minimum mean square error (MMSE) estimation over the distributed network.

For the standard network model in [28], [31], we consider a network of L agents connected with $\mathbf{\Omega} \in \mathbb{R}^{L \times L}$, where $\Omega_{kl} \geq 0$. If agents k and l are connected, otherwise is zero. We consider a network consisting of N agents that each agent k has its own neighbour $\mathcal{N}_k$ connected to the agent k including itself as shown in Figure 2, where the symbol $\mathcal{N}_k$ denotes the neighborhood of agent k. Each agent is connected to a neighborhood of other and it updates a local estimate of HSAF model. Each agent in the distributed network operates its own adaptive filters. We assume the $\{\Omega_{kl}\}$ is non-negative scalar that satisfies the following conditions as

$$\sum_{k=1}^{L} \Omega_{kl} = 1 \text{ and } \Omega_{kl} = 0, \text{ if } l \notin \mathcal{N}_k, \tag{18}$$

for $k = 1, 2, \ldots, L$. An $L \times L$ matrix $\mathbf{\Omega}$ consists of the entries $\{\Omega_{kl}\}$ that the l^{th} row of $\mathbf{\Omega}$ is formed of $\{\Omega_{kl}, k = 1, 2, \ldots, L\}$.

So, the output signal $d_{n,k}$ at sample n and agent k is given as

$$d_{n,k} = \mathbf{u}_{n,k}^T \, \mathbf{w}_o + \nu_{n,k}, \tag{19}$$

where the input vector $\mathbf{u}_{n,k} = [\, u_{n,k} \ u_{n-1,k} \ \ldots \ u_{n-M+1,k}]^T$, $\mathbf{w}_o$ denotes as the linear weight and $\nu_{n,k} \sim \mathcal{N}(0, \sigma^2)$ is defined as a Gaussian noise.

For the distributed network, there are two local parameters $u_{n,k}$ and index i in a LUT for each agent are as

$$u_{n,k} = \frac{x_{n,k}}{\Delta x} - \left\lfloor \frac{x_{n,k}}{\Delta x} \right\rfloor \tag{20}$$

$$i = \left\lfloor \frac{x_{n,k}}{\Delta x} \right\rfloor + \frac{Q-1}{2}, \tag{21}$$

where Δx is the distance between the control points coefficient $\mathbf{q}_{i,n,k}$.

Next, the objective of network is to optimise the HSAF non-linear weight coefficient $\mathbf{q}_{i,n,k}$ and linear weight coefficient $\mathbf{w}_{n,k}$ by following the minimised global cost function by aggregating the sum of local cost function as [32]

$$\begin{aligned} \mathbb{J}_{global}(\mathbf{q}_{i,n,k} \, \mathbf{w}_{n,k}) &= \sum_{k=1}^{L} \mathbb{J}_{local,k}(\mathbf{q}_{i,n,k}, \mathbf{w}_{n,k}) \\ &= \sum_{k=1}^{L} \mathbb{E}\{\, e_{n,k}^2 \,\}, \end{aligned} \tag{22}$$

where each expectation is computed w.r.t the local input. An individual local cost function $\mathbb{J}_{local,k}(\mathbf{q}_{i,n,k}, \mathbf{w}_{n,k})$ of each agent is assumed in terms of MMSE, where $e_{n,k}$ is a local error of each agent.

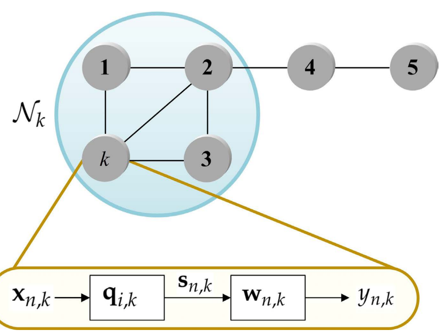

FIGURE 2. Schematic of DHSAF interpolation over a network agents.

IV. PROPOSED DIFFUSION HAMMERSTEIN SPLINE ADAPTIVE FILTERING BASED ON LEAST MEAN SQUARE ALGORITHM

In this section, we employ the diffusion adaptation on HSAF (DHSAF) with the standard stochastic gradient descent method. Proposed DHSAF based on least mean square (DHSAF-LMS) algorithm is followed [28], [31] by consisting of two types diffusion strategies as *combine-then-adapt* (CTA) algorithm and *adapt-then-combine* (ATC) algorithm.

A. COMBINE-THEN-ADAPT DHSAF-LMS ALGORITHM

We present a *combine-then-adapt* (CTA) of diffusion Hammerstein spline filtering based on least mean square (CTA-DHSAF-LMS) algorithm consisting of the combination and adaptation parts.

For the nonlinear part of DHSAF, each agent diffuses its own estimate in the combination part of the nonlinear part $\boldsymbol{\zeta}_{i,n,k}$ in terms of the controls point coefficients $\mathbf{q}_{i,n,k}$ as

$$\boldsymbol{\zeta}_{i,n,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl} \, \mathbf{q}_{i,n,l}, \tag{23}$$

where the combination coefficient $\{\Omega_{kl}\}$ is non-negative scalar. In the adaptation part, the combined of each diffused agent can be obtained adaptively as

$$\mathbf{q}_{i,n+1,k} = \boldsymbol{\zeta}_{i,n,k} + \mu_q \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \mathbf{w}_{n,k} \, e_{n,k}, \tag{24}$$

where $\mathbb{C}$ is a spline basis matrix and $\boldsymbol{\zeta}_{n,k}$ is given in (23).

For the linear part of DHSAF, each agent diffuses its own estimate in the combination part of the linear part $\boldsymbol{\Gamma}_{i,n,k}$ in terms of linear tap-weight coefficients $\mathbf{w}_{n,k}$ as

$$\boldsymbol{\Gamma}_{i,n,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl} \, \mathbf{w}_{n,l}, \tag{25}$$

where the combination coefficient $\{\Omega_{kl}\}$ is non-negative scalar. In the adaptation part, the combined of each diffused agent of $\mathbf{w}_{n,k}$ can be expressed adaptively as

$$\mathbf{w}_{n+1,k} = \boldsymbol{\Gamma}_{n,k} + \mu_w \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \mathbf{q}_{i,n,k} \, e_{n,k}, \tag{26}$$

where $\boldsymbol{\Gamma}_{n,k}$ is given in (25) and the local error $e_{n,k}$ is defined as

$$e_{n,k} = d_{n,k} - \mathbf{w}_{n,k} \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \mathbf{q}_{i,n,k}. \tag{27}$$

The summary of CTA version of DHSAF-LMS is shown in Algorithm 2.

Algorithm 2: CTA-DHSAF-LMS Algorithm

Initialise: $\mathbf{w}_0 = \delta_n \cdot [1 \;\; 0 \; \ldots \; 0]^T$, $\mathbf{q}_0 = [1 \;\; 0 \; \ldots \; 0]^T$
  for $n = 0, 1, \ldots$ **do**
    for $k = 1, \ldots, L$ **do**
      $\boldsymbol{\zeta}_{i,n,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl} \, \mathbf{q}_{i,n,l}$
      $\mathbf{q}_{i,n+1,k} = \boldsymbol{\zeta}_{i,n,k} + \mu_q \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \mathbf{w}_{n,k} \, e_{n,k}$
      $\boldsymbol{\Gamma}_{i,n,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl} \, \mathbf{w}_{n,l}$
      $\mathbf{w}_{n+1,k} = \boldsymbol{\Gamma}_{n,k} + \mu_w \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \mathbf{q}_{i,n,k} \, e_{n,k}$
      $e_{n,k} = d_{n,k} - \mathbf{w}_{n,k} \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \mathbf{q}_{i,n,k}$
    end
  end

B. ADAPT-THEN-COMBINE DHSAF-LMS ALGORITHM

We present a *adapt-then-combine* (ATC) of diffusion Hammerstein spline filtering based on least mean square (ATC-DHSAF-LMS) algorithm consisting of the combination and adaptation parts.

For the nonlinear part of DHSAF, each diffused agent of the controls point coefficients $\tilde{\boldsymbol{\zeta}}_{i,n,k}$ can be obtained adaptively in the adaptation part as

$$\tilde{\boldsymbol{\zeta}}_{i,n,k} = \tilde{\mathbf{q}}_{n+1,k} + \mu_q \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \tilde{\mathbf{w}}_{n,k} \, \tilde{e}_{n,k}, \tag{28}$$

where $\tilde{e}_{n,k}$ is given by

$$\tilde{e}_{n,k} = d_{n,k} - \tilde{\mathbf{w}}_{n,k} \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \tilde{\mathbf{q}}_{i,n,k}. \tag{29}$$

In the combination part of the nonlinear part in terms of the controls point coefficients $\tilde{\mathbf{q}}_{i,n,k}$, each agent diffuses its own estimate as

$$\tilde{\mathbf{q}}_{i,n+1,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl} \, \tilde{\boldsymbol{\zeta}}_{i,n,k}, \tag{30}$$

where the combination coefficient $\{\Omega_{kl}\}$ is nonnegative scalar and $\tilde{\boldsymbol{\zeta}}_{n,k}$ is given in (28).

For the linear part of DHSAF, each diffused agent of the linear coefficients $\tilde{\boldsymbol{\Gamma}}_{n,k}$ can be obtained adaptively in the adaptation part as

$$\tilde{\boldsymbol{\Gamma}}_{n,k} = \tilde{\mathbf{w}}_{n+1,k} + \mu_w \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \tilde{\mathbf{q}}_{i,n,k} \, \tilde{e}_{n,k}, \tag{31}$$

where $e_{n,k}$ is given in (27).

In the combination part of the nonlinear part in terms of the linear coefficients $\tilde{\mathbf{w}}_{n,k}$, each agent diffuses its own estimate

$$\tilde{\mathbf{w}}_{n+1,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl} \, \tilde{\boldsymbol{\Gamma}}_{i,n,k}, \tag{32}$$

where the combination coefficient $\{\Omega_{kl}\}$ is nonnegative scalar and $\tilde{\boldsymbol{\Gamma}}_{n,k}$ is given in (31).

The summary of ATC version of DHSAF-LMS is shown in Algorithm 3.

Algorithm 3: ATC-DHSAF-LMS algorithm

Initialise: $\tilde{\mathbf{w}}_0 = \delta_n \cdot [1 \;\; 0 \; \ldots \; 0]^T$, $\tilde{\mathbf{q}}_0 = [1 \;\; 0 \; \ldots \; 0]^T$
  for $n = 0, 1, \ldots$ **do**
    for $k = 1, \ldots, L$ **do**
      $\tilde{\boldsymbol{\zeta}}_{i,n,k} = \tilde{\mathbf{q}}_{n+1,k} + \mu_q \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \tilde{\mathbf{w}}_{n,k} \, \tilde{e}_{n,k}$
      $\tilde{\mathbf{q}}_{i,n+1,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl} \, \tilde{\boldsymbol{\zeta}}_{i,n,k}$
      $\tilde{\boldsymbol{\Gamma}}_{n,k} = \tilde{\mathbf{w}}_{n+1,k} + \mu_w \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \tilde{\mathbf{q}}_{i,n,k} \, \tilde{e}_{n,k}$
      $\tilde{\mathbf{w}}_{n+1,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl} \, \tilde{\boldsymbol{\Gamma}}_{i,n,k}$
      $\tilde{e}_{n,k} = d_{n,k} - \tilde{\mathbf{w}}_{n,k} \, \mathbf{u}_{n,k}^T \, \mathbb{C} \, \tilde{\mathbf{q}}_{i,n,k}$
    end
  end

V. PROPOSED DIFFUSION HAMMERSTEIN SPLINE ADAPTIVE FILTERING BASED ON NORMALISED ORTHOGONAL GRADIENT ADAPTIVE ALGORITHM

As stated in [31], we inspire the diffusion filtering approach on the parallel adaptation with the diffusion steps, where data on the present estimates are combined locally on the matrix $\boldsymbol{\Omega}$. The structure of proposed DHSAF based on normalised orthogonal gradient adaptive algorithm (DHSAF-NOGA) is depicted in Figure 3.

In this section, there are two types of adaptive diffusion strategies with the CTA algorithm and ATC algorithm [31] implemented on the proposed DHSAF-NOGA algorithm as follows.

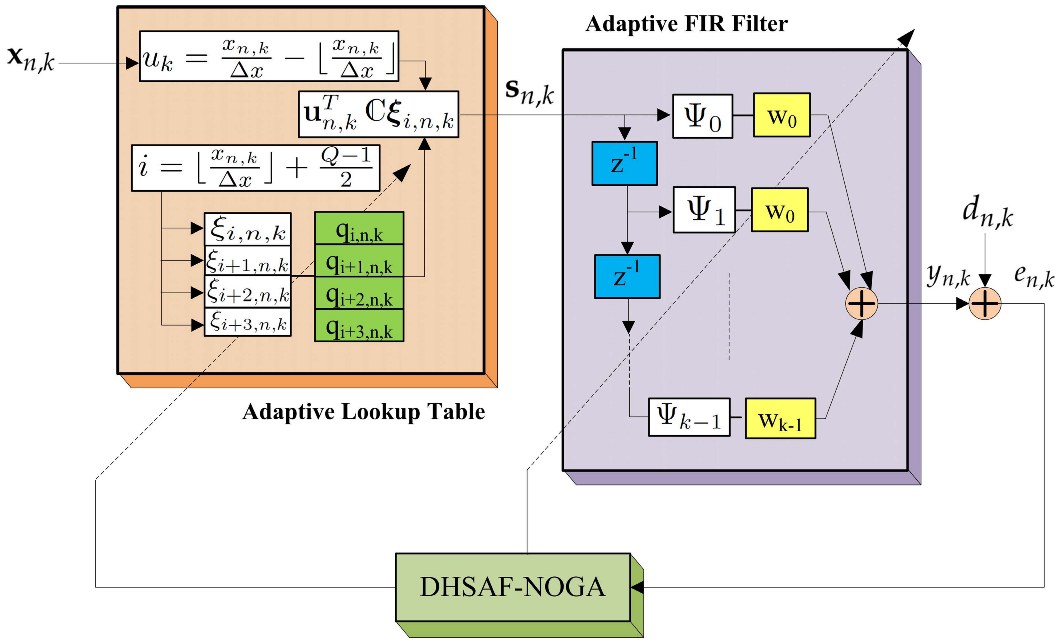

FIGURE 3. Proposed diffusion hammerstein spline adaptive filtering (DHSAF) structure based on NOGA algorithm.

A. DHSAF-NOGA WITH COMBINE-THEN-ADAPT STRATEGY

We propose a CTA version of DHSAF based on NOGA (CTA-DHSAF-NOGA) algorithm. Each agent diffuses its own estimate of the nonlinear part of DHSAF as

$$\boldsymbol{\xi}_{i,n,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl}\, \hat{\mathbf{q}}_{i,n,l}, \tag{33}$$

where the combination coefficient $\{\Omega_{kl}\}$ is nonnegative scalar shown in (18).

In particular, the combination weight $\boldsymbol{\xi}_{i,n,k}$ is computed in the output of the nonlinear filtering part as

$$\hat{\mathbf{s}}_{n,k} = \mathbf{u}_{n,k}^T\, \mathbb{C}\, \boldsymbol{\xi}_{i,n,k}, \tag{34}$$

where $\boldsymbol{\xi}_{i,n,k}$ is defined in (33).

For the linear filtering part, a second diffusion step of the weight $\boldsymbol{\Psi}_{n,k}$ is combined as

$$\boldsymbol{\Psi}_{n,k} = \sum_{l \in \mathcal{N}_k} \Omega_{kl}\, \hat{\mathbf{w}}_{n,l}. \tag{35}$$

Since, the local error $\hat{e}_{n,k}$ at each agent is given as

$$\hat{e}_{n,k} = d_{n,k} - \boldsymbol{\Psi}_{n,k}^T\, \hat{\mathbf{s}}_{n,k}, \tag{36}$$

where $y_{n,k}$ denotes as the output of the liner filtering part. The vectors $\mathbf{s}_{n,k}$ and $\boldsymbol{\Psi}_{n,k}$ are defined in (34) and (35), respectively.

Then, the control points coefficient $\hat{\mathbf{q}}_{i,n,k}$ for nonlinear part of DHSAF-NOGA is adapted recursively with the CTA version as

$$\hat{\mathbf{q}}_{i,n+1,k} = \boldsymbol{\xi}_{i,n,k} + \mu_q\, \hat{\mathbf{d}}_{q_{n,k}}, \tag{37}$$

where μ_q is the step-size parameter, $\boldsymbol{\xi}_{i,n,k}$ is given in (33) and the directional vector $\mathbf{d}_{q_{n,k}}$ is defined as

$$\hat{\mathbf{d}}_{q_{n+1,k}} = \hat{\lambda}_{q_{n,k}}\, \hat{\mathbf{d}}_{q_{n,k}} - \hat{\mathbf{g}}_{q_{n,k}}, \tag{38}$$

and the gradient vector $\hat{\mathbf{g}}_{q_{n,k}}$ is given as

$$\hat{\mathbf{g}}_{q_{n+1,k}} = \hat{\lambda}_{q_{n,k}}\, \hat{\mathbf{g}}_{q_{n,k}} - \nabla_q \mathbb{J}(\boldsymbol{\xi}_{i,n,k}), \tag{39}$$

and the forgetting-factor $\hat{\lambda}_{q_{n,k}}$ is given as

$$\hat{\lambda}_{q_{n,k}} = \frac{\hat{\mathbf{d}}_{q_{n-1,k}}^T\, \hat{\mathbf{g}}_{q_{n,k}}}{\hat{\mathbf{d}}_{q_{n,k}}^T\, \hat{\mathbf{d}}_{q_{n-1,k}}}. \tag{40}$$

The derivative of cost function $\nabla_q \mathbb{J}(\boldsymbol{\xi}_{i,n,k})$ in (22) w.r.t $\hat{\mathbf{q}}_{i,n,k}$ is applied with the chain rules [13] as

$$\begin{aligned} \nabla_q \mathbb{J}(\boldsymbol{\xi}_{i,n,k}) &= -\hat{e}_{n,k}\, \frac{\partial y_{n,k}}{\partial \hat{\mathbf{s}}_{n,k}}\, \frac{\partial \mathbf{s}_{n,k}}{\partial \boldsymbol{\xi}_{i,n,k}} \\ &= -\mathbf{u}_{n,k}^T \mathbb{C} \boldsymbol{\Psi}_{n,k}\, \hat{e}_{n,k}, \end{aligned} \tag{41}$$

where $\boldsymbol{\Psi}_{n,k}$ is defined in (35).

Moreover, the linear tap-weight coefficient $\hat{\mathbf{w}}_{n,k}$ for linear part of DHSAF adaptation is updated recursively with the CTA version as

$$\hat{\mathbf{w}}_{n+1,k} = \boldsymbol{\Psi}_{n,k} + \mu_w\, \hat{\mathbf{d}}_{w_{n,k}}, \tag{42}$$

where μ_w is the step-size parameter and the directional vector $\hat{\mathbf{d}}_{w_{n,k}}$ can be expressed as

$$\hat{\mathbf{d}}_{w_{n+1,k}} = \hat{\lambda}_{w_{n,k}}\, \hat{\mathbf{d}}_{w_{n,k}} - \hat{\mathbf{g}}_{w_{n,k}}. \tag{43}$$

Following [5], the forgetting-factor $\hat{\lambda}_{w_{n,k}}$ is given as

$$\hat{\lambda}_{w_{n,k}} = \frac{\hat{\mathbf{d}}^T_{w_{n-1,k}} \hat{\mathbf{g}}_{w_{n,k}}}{\hat{\mathbf{d}}^T_{w_{n,k}} \hat{\mathbf{d}}_{w_{n-1,k}}}, \tag{44}$$

where $\hat{\mathbf{g}}_{w_{n,k}}$ denotes as the gradient vector of linear coefficient $\hat{\mathbf{w}}_{n+1,k}$ that can be obtained as

$$\hat{\mathbf{g}}_{w_{n+1,k}} = \hat{\lambda}_{w_{n,k}} \hat{\mathbf{g}}_{w_{n,k}} - \nabla_w \mathbb{J}(\boldsymbol{\Psi}_{i,n,k}), \tag{45}$$

and the derivative of cost function $\nabla_w \mathbb{J}(\boldsymbol{\Psi}_{i,n,k})$ in (22) w.r.t $\hat{\mathbf{w}}_{n,k}$ is defined as

$$\nabla_w \mathbb{J}(\boldsymbol{\Psi}_{i,n,k}) = -\hat{\mathbf{s}}_{n,k}\, \hat{e}_{n,k}, \tag{46}$$

where $\hat{\mathbf{s}}_{n,k}$ is given in (34).

Therefore, the overall DHSAF with CTA version of NOGA is summarised in Algorithm 4.

Algorithm 4: Proposed DHSAF-NOGA Algorithm With CTA Strategy

1: **Initialise:** $\hat{\mathbf{w}}_0 = \delta_n \cdot [1\ \ 0\ \dots\ 0]^T$, $\hat{\mathbf{q}}_0 = [1\ \ 0\ \dots\ 0]^T$ $\mathbf{d}_{q_0} = \mathbf{d}_{w_0} = [1\ \ 0\ \dots\ 0]^T$, $\mathbf{g}_{q_0} = \mathbf{g}_{w_0} = [1\ \ 0\ \dots\ 0]^T$
2: **for** $n = 0, 1, \dots$ **do**
3: **for** $k = 1, \dots, L$ **do**
4: $\hat{\mathbf{s}}_{n,k} = \mathbf{u}^T_{n,k}\, \mathbb{C} \boldsymbol{\xi}_{i,n,k}$
5: $\hat{e}_{n,k} = d_{n,k} - \boldsymbol{\Psi}^T_{n,k}\, \hat{\mathbf{s}}_{n,k}$
6: $\boldsymbol{\xi}_{i,n,k} = \sum_{l\in\mathcal{N}_k} \Omega_{kl}\, \hat{\mathbf{q}}_{i,n,l}$
7: $\hat{\mathbf{q}}_{i,n+1,k} = \boldsymbol{\xi}_{i,n,k} + \mu_q\, \hat{\mathbf{d}}_{q_{n,k}}$
8: $\hat{\mathbf{d}}_{q_{n+1,k}} = \hat{\lambda}_{q_{n,k}}\, \hat{\mathbf{d}}_{q_{n,k}} - \hat{\mathbf{g}}_{q_{n,k}}$
9: $\hat{\mathbf{g}}_{q_{n+1,k}} = \hat{\lambda}_{q_{n,k}}\, \hat{\mathbf{g}}_{q_{n,k}} + \mathbf{u}^T_{n,k}\, \mathbb{C}\, \boldsymbol{\Psi}_{n,k}\, \hat{e}_{n,k}$
10: $\hat{\lambda}_{q_{n,k}} = \frac{\hat{\mathbf{d}}^T_{q_{n-1,k}} \hat{\mathbf{g}}_{q_{n,k}}}{\hat{\mathbf{d}}^T_{q_{n,k}} \hat{\mathbf{d}}_{q_{n-1,k}}}$
11: $\boldsymbol{\Psi}_{n,k} = \sum_{l\in\mathcal{N}_k} \Omega_{kl}\, \hat{\mathbf{w}}_{n,l}$
12: $\hat{\mathbf{w}}_{n+1,k} = \boldsymbol{\Psi}_{n,k} + \mu_w\, \hat{\mathbf{d}}_{w_{n,k}}$
13: $\hat{\mathbf{d}}_{w_{n+1,k}} = \hat{\lambda}_{w_{n,k}}\, \hat{\mathbf{d}}_{w_{n,k}} - \hat{\mathbf{g}}_{w_{n,k}}$
14: $\hat{\mathbf{g}}_{w_{n+1,k}} = \hat{\lambda}_{w_{n,k}}\, \hat{\mathbf{g}}_{w_{n,k}} + \hat{\mathbf{s}}_{n,k}\, \hat{e}_{n,k}$
15: $\hat{\lambda}_{w_{n,k}} = \frac{\hat{\mathbf{d}}^T_{w_{n-1,k}} \hat{\mathbf{g}}_{w_{n,k}}}{\hat{\mathbf{d}}^T_{w_{n,k}} \hat{\mathbf{d}}_{w_{n-1,k}}}$
16: **end**
17: **end**

B. DHSAF-NOGA WITH ADAPT-THEN- COMBINE STRATEGY

We propose an ATC version of DHSAF based on NOGA (ATC-DHSAF-NOGA) algorithm. The output of nonlinear filtering part of DHSAF $\tilde{\mathbf{s}}_{n,k}$ is defined as

$$\tilde{\mathbf{s}}_{n,k} = \mathbf{u}^T_{n,k}\, \mathbb{C}\, \check{\mathbf{q}}_{i,n,k}. \tag{47}$$

Hence, the local error $\check{e}_{n,k}$ at each agent can be expressed as

$$\check{e}_{n,k} = d_{n,k} - \check{\mathbf{w}}^T_{n,k}\, \tilde{\mathbf{s}}_{n,k}, \tag{48}$$

where $\tilde{\mathbf{s}}_{n,k}$ is given in (47).

So, each agent adapts its own estimate of the nonlinear part of the DHSAF as

$$\tilde{\boldsymbol{\xi}}_{i,n,k} = \check{\mathbf{q}}_{i,n,k} + \mu_q\, \tilde{\mathbf{d}}_{q_{n,k}}, \tag{49}$$

where μ_q is the step-size parameter and the directional vector $\tilde{\mathbf{d}}_{q_{n,k}}$ of controls point coefficient $\check{\mathbf{q}}_{i,n,k}$ is defined as

$$\tilde{\mathbf{d}}_{q_{n,k}} = \tilde{\lambda}_{q_{n,k}}\, \tilde{\mathbf{d}}_{q_{n,k}} - \tilde{\mathbf{g}}_{q_{n,k}}, \tag{50}$$

and the gradient vector $\tilde{\mathbf{g}}_{q_{n,k}}$ is given as

$$\tilde{\mathbf{g}}_{q_{n+1,k}} = \tilde{\lambda}_{q_{n,k}}\, \tilde{\mathbf{g}}_{q_{n,k}} - \nabla_q \mathbb{J}(\check{\mathbf{q}}_{i,n,k}), \tag{51}$$

and the forgetting-factor $\tilde{\lambda}_{q_{n,k}}$ is defined by [5]

$$\tilde{\lambda}_{q_{n,k}} = \frac{\tilde{\mathbf{d}}^T_{q_{n-1,k}}\, \tilde{\mathbf{g}}_{q_{n,k}}}{\tilde{\mathbf{d}}^T_{q_{n,k}}\, \tilde{\mathbf{d}}_{q_{n-1,k}}}, \tag{52}$$

and the derivative of cost function $\nabla_q \mathbb{J}(\check{\mathbf{q}}_{i,n,k})$ w.r.t $\check{\mathbf{q}}_{i,n,k}$ is defined as

$$\begin{aligned} \nabla_q \mathbb{J}(\check{\mathbf{q}}_{i,n,k}) &= -\check{e}_{n,k} \frac{\partial y_{n,k}}{\partial \tilde{\mathbf{s}}_{n,k}} \frac{\partial \tilde{\mathbf{s}}_{n,k}}{\partial \check{\mathbf{q}}_{i,n,k}} \\ &= -\mathbf{u}^T_{n,k} \mathbb{C} \check{\mathbf{w}}_{n,k}\, \check{e}_{n,k}, \end{aligned} \tag{53}$$

where $\check{e}_{n,k}$ is given in (48).

Then, the control points coefficient $\check{\mathbf{q}}_{i,n,k}$ is combined as

$$\check{\mathbf{q}}_{i,n,k} = \sum_{l\in\mathcal{N}_k} \Omega_{kl}\, \tilde{\boldsymbol{\xi}}_{i,n,l}, \tag{54}$$

where $\tilde{\boldsymbol{\xi}}_{i,n,l}$ is given in (49).

For the linear filtering part, the weight $\tilde{\boldsymbol{\Psi}}_{n,k}$ can be updated adaptively as

$$\tilde{\boldsymbol{\Psi}}_{n,k} = \check{\mathbf{w}}_{n,k} + \mu_w\, \tilde{\mathbf{d}}_{w_{n,k}}, \tag{55}$$

where μ_w is the step-size parameter and the directional vector $\tilde{\mathbf{d}}_{w_{n,k}}$ is given as

$$\tilde{\mathbf{d}}_{w_{n+1,k}} = \tilde{\lambda}_{w_{n,k}}\, \tilde{\mathbf{d}}_{w_{n,k}} - \tilde{\mathbf{g}}_{w_{n,k}}, \tag{56}$$

and the gradient vector $\tilde{\mathbf{g}}_{w_{n,k}}$ can be calculated as

$$\tilde{\mathbf{g}}_{w_{n+1,k}} = \tilde{\lambda}_{w_{n,k}}\, \tilde{\mathbf{g}}_{w_{n,k}} - \nabla_w \mathbb{J}(\check{\mathbf{w}}_{n,k}). \tag{57}$$

Following [5], the forgetting-factor $\tilde{\lambda}_{w_{n,k}}$ is defined by

$$\tilde{\lambda}_{w_{n,k}} = \frac{\tilde{\mathbf{d}}^T_{w_{n-1,k}}\, \tilde{\mathbf{g}}_{w_{n,k}}}{\tilde{\mathbf{d}}^T_{w_{n,k}}\, \tilde{\mathbf{d}}_{w_{n-1,k}}}, \tag{58}$$

and the derivative of cost function $\nabla_w \mathbb{J}(\check{\mathbf{w}}_{n,k})$ in (22) w.r.t $\check{\mathbf{w}}_{n,k}$ is defined as

$$\nabla_w \mathbb{J}(\check{\mathbf{w}}_{n,k}) = -\tilde{\mathbf{s}}_{n,k}\, \check{e}_{n,k}, \tag{59}$$

where $\tilde{\mathbf{s}}_{n,k}$ is given in (47).

Therefore, the linear tap-weight coefficient $\check{\mathbf{w}}_{n,k}$ can be expressed as

$$\check{\mathbf{w}}_{n,k} = \sum_{l\in\mathcal{N}_k} \Omega_{kl}\, \tilde{\boldsymbol{\Psi}}_{n,l}. \tag{60}$$

The overall DHSAF with ATC version of NOGA is summarised in Algorithm 5.

Algorithm 5: Proposed DHSAF-NOGA Algorithm With ATC Strategy

Initialise: $\check{\mathbf{w}}_0 = \delta_n \cdot [1\ \ 0\ \dots\ 0]^T$, $\check{\mathbf{q}}_0 = [1\ \ 0\ \dots\ 0]^T$, $\tilde{\mathbf{d}}_{q_0} = \tilde{\mathbf{d}}_{w_0} = [1\ \ 0\ \dots\ 0]^T$, $\tilde{\mathbf{g}}_{q_0} = \tilde{\mathbf{g}}_{w_0} = [1\ \ 0\ \dots\ 0]^T$
for $n = 0, 1, \dots$ **do**
  for $k = 1, \dots, L$ **do**
    $\check{e}_{n,k} = d_{n,k} - \check{\mathbf{w}}_{n,k}^T\, \tilde{\mathbf{s}}_{n,k}$
    $\tilde{\boldsymbol{\xi}}_{i,n,k} = \check{\mathbf{q}}_{i,n-1,k} + \mu_q\, \tilde{\mathbf{d}}_{q_{n,k}}$
    $\tilde{\mathbf{d}}_{q_{n+1,k}} = \tilde{\lambda}_{q_{n,k}}\, \tilde{\mathbf{d}}_{q_{n,k}} - \tilde{\mathbf{g}}_{q_{n,k}}$
    $\tilde{\mathbf{g}}_{q_{n+1,k}} = \tilde{\lambda}_{q_{n,k}}\, \tilde{\mathbf{g}}_{q_{n,k}} + \mathbf{u}_{n,k}^T\, \mathbb{C}\, \check{\mathbf{w}}_{n,k}\, \check{e}_{n,k}$
    $\tilde{\lambda}_{q_{n,k}} = \dfrac{\tilde{\mathbf{d}}_{q_{n-1,k}}^T \tilde{\mathbf{g}}_{q_{n,k}}}{\tilde{\mathbf{d}}_{q_{n,k}}^T\, \tilde{\mathbf{d}}_{q_{n-1,k}}}$
    $\check{\mathbf{q}}_{i,n,k} = \sum_{l\in\mathcal{N}_k} \Omega_{kl}\, \tilde{\boldsymbol{\xi}}_{i,n,l}$
    $\tilde{\mathbf{s}}_{n,k} = \mathbf{u}_{n,k}^T\, \mathbb{C}\check{\mathbf{q}}_{i,n,k}$
    $\tilde{\boldsymbol{\Psi}}_{n,k} = \check{\mathbf{w}}_{n-1,k} + \mu_w\, \tilde{\mathbf{d}}_{w_{n,k}}$
    $\tilde{\mathbf{d}}_{w_{n+1,k}} = \tilde{\lambda}_{w_{n,k}}\, \tilde{\mathbf{d}}_{w_{n,k}} - \tilde{\mathbf{g}}_{w_{n,k}}$
    $\tilde{\mathbf{g}}_{w_{n+1,k}} = \tilde{\lambda}_{w_{n,k}}\, \tilde{\mathbf{g}}_{w_{n,k}} + \tilde{\mathbf{s}}_{n,k}\, \check{e}_{n,k}$
    $\tilde{\lambda}_{w_{n,k}} = \dfrac{\tilde{\mathbf{d}}_{w_{n-1,k}}^T \tilde{\mathbf{g}}_{w_{n,k}}}{\tilde{\mathbf{d}}_{w_{n,k}}^T\, \tilde{\mathbf{d}}_{w_{n-1,k}}}$
    $\check{\mathbf{w}}_{n,k} = \sum_{l\in\mathcal{N}_k} \Omega_{kl}\, \tilde{\boldsymbol{\Psi}}_{n,l}$
  end
end

The comparison of computational complexities for the proposed algorithms is shown in Table 1. Following [16], [33], calculation of output of nonlinear filtering part is used $4P$ multiplications and $4P$ additions with data reusing of previous computations, where P is the spline order.

VI. NETWORK STABILITY AND PERFORMANCE FOR DISTRIBUTED STRATEGIES

In this section, we investigate the network stability and performance analysis that are referred to the stochastic gradient approximation for the diffusion strategies.

A. NETWORK STABILITY

Following [31], we analyse by proposing from the gradient descent algorithm dealing with the MSE networks. We consider the MSE network that involves with the common minimisation (w_0, q_0) for the non-cooperative and diffusion strategies in terms of three sets of combination coefficients $\{\Omega_{0,kl}, \Omega_{1,kl}, \Omega_{2,kl}\}$.

TABLE 1. Number of required arithmetic operations of proposed algorithms.

Algorithm	Multiplication	Addition	Division
HSAF-LMS	$2M + 4P + 1$	$2M + 4P$	0
HSAF-NOGA	$4M + 8P + 8$	$4M + 8P + 7$	2
DHSAF-LMS	$2M + 4P + 2$	$2M + 4P + 2N_k$	0
DHSAF-NOGA	$4M + 8P + 9$	$4M + 8P + 7 + 2N_k$	2

Let us consider firstly that involves with the nonlinear part of DHSAF-NOGA algorithm as

$$\begin{cases} \boldsymbol{\zeta}_{i,n,k-1} &= \sum_{l\in\mathcal{N}_k} \Omega_{1,kl}\, \mathbf{q}_{i,l,k-1}, \\ \mathbf{q}_{i,n,k} &= \sum_{l\in\mathcal{N}_k} \Omega_{2,kl} \boldsymbol{\xi}_{i,l,k}, \\ \boldsymbol{\xi}_{i,n,k} &= \sum_{l\in\mathcal{N}_k} \Omega_{0,kl}\, \boldsymbol{\zeta}_{i,l,k-1} + \mu_q\, \mathbf{d}_{q_{n,k}}, \\ \mathbf{d}_{q_{n,k}} &= \lambda_{q_{n,k}}\, \mathbf{d}_{q_{n,k-1}} - \mathbf{g}_{q_{n,k}}, \\ \mathbf{g}_{q_{n,k}} &= \lambda_{q_{n,k}}\, \mathbf{g}_{q_{n,k-1}} - \mu_q\, \nabla_q\, \mathbb{J}(\mathbf{q}, \mathbf{w}), \\ &= \lambda_{q_{n,k}}\, \mathbf{g}_{q_{n,k-1}} + \mu_q \mathbf{u}_{n,k}^T\, \mathbb{C}\, \mathbf{w}_{n,k}, e_{n,k}, \end{cases}$$

and $\{\boldsymbol{\zeta}_{i,n,k-1}, \boldsymbol{\xi}_{i,n,k}\}$ represent the $M \times 1$ vector. While the combination coefficients $\{\Omega_{0,kl}, \Omega_{1,kl}, \Omega_{2,kl}\}$ are the nonnegative $L \times L$ matrices followed the conditions in (18).

Then, we associate with each agent k following three errors as

$$\tilde{\mathbf{q}}_{i,n,k} \triangleq q_0 - \mathbf{q}_{i,n,k}, \tag{61}$$
$$\tilde{\boldsymbol{\xi}}_{i,n,k} \triangleq q_0 - \boldsymbol{\xi}_{i,n,k}, \tag{62}$$
$$\tilde{\boldsymbol{\zeta}}_{i,n,k} \triangleq q_0 - \boldsymbol{\zeta}_{i,n,k}, \tag{63}$$

where q_0 denotes the global minimisation.

Next, we measure the deviation from q_0 by subtracting q_0 from both sides of (61) and substituting the local error $e_{n,k}$ and $\mathbf{s}_{n,k}$ as

$$e_{n,k} = d_{n,k} - \mathbf{w}_{n,k}^T\, \mathbf{s}_{n,k}, \tag{64}$$
$$\mathbf{s}_{n,k} = \mathbf{u}_{n,k}^T\, \mathbb{C}\, \tilde{\boldsymbol{\zeta}}_{i,n,k}. \tag{65}$$

So, we can reorganise the nonlinear part of DHSAF-NOGA algorithm for $\tilde{\mathbf{q}}_{i,n,k}$ as

$$\begin{cases} \tilde{\boldsymbol{\zeta}}_{i,n,k-1} &= \sum_{l\in\mathcal{N}_k} \Omega_{1,kl}\, \tilde{\mathbf{q}}_{i,l,k-1}, \\ \tilde{\mathbf{q}}_{i,n,k} &= \sum_{l\in\mathcal{N}_k} \Omega_{2,kl}\, \tilde{\boldsymbol{\xi}}_{i,l,k}, \\ \tilde{\boldsymbol{\xi}}_{i,n,k} &= \sum_{l\in\mathcal{N}_k} \Omega_{0,kl} \tilde{\boldsymbol{\zeta}}_{i,l,k-1} + \mu_q, \tilde{\mathbf{d}}_{q_{n,k}}, \\ \tilde{\mathbf{d}}_{q_{n,k}} &= \lambda_{q_{n,k}}\, \tilde{\mathbf{d}}_{q_{n,k-1}} - \tilde{\mathbf{g}}_{q_{n,k}}, \\ \tilde{\mathbf{g}}_{q_{n,k}} &= \lambda_{q_{n,k}}\, \tilde{\mathbf{g}}_{q_{n,k-1}} \\ &\quad + \mu_q \mathbf{u}_{n,k}^T \mathbb{C} \tilde{\mathbf{w}}_{n,k} [d_{n,k} - \mathbf{w}_{n,k}^T \mathbf{s}_{n,k}] \\ &= \lambda_{q_{n,k}}\, \tilde{\mathbf{g}}_{q_{n,k-1}} + \mu_q \mathbf{u}_{n,k}^T\, \mathbb{C}\, \tilde{\mathbf{w}}_{n,k}, \nu_{n,k} \\ &\quad - \mu_q \mathbf{u}_{n,k}^T \mathbb{C}\, \tilde{\mathbf{w}}_{n,k}^T \mathbf{w}_{n,k} \mathbf{u}_{n,k}^T \mathbb{C}\, \tilde{\boldsymbol{\zeta}}_{i,n,k}, \end{cases}$$

where $\nu_{n,k}$ denotes as a Gaussian noise term given in (19).

In (66), we define the non-negative entries of $N \times N$ matrices as

$$A_0 = [\Omega_{0,kl}], A_1 = [\Omega_{1,kl}], A_2 = [\Omega_{2,kl}]. \tag{66}$$

Since, we examine the error dynamic evolution across the network, we collect the error vector into $N \times 1$ block vectors, while the individual entries are of $M \times 1$ vectors as

$$\begin{cases} \tilde{\mathbf{q}}_{i,n,k} & \triangleq [\ \tilde{\mathbf{q}}_{i,1,k}\ \ \tilde{\mathbf{q}}_{i,2,k}\ \ \ldots\ \ \tilde{\mathbf{q}}_{i,N,k}\ ]^T, \\ \tilde{\boldsymbol{\xi}}_{i,n,k} & \triangleq [\ \tilde{\boldsymbol{\xi}}_{i,1,k}\ \ \tilde{\boldsymbol{\xi}}_{i,2,k}\ \ \ldots\ \ \tilde{\boldsymbol{\xi}}_{i,N,k}\ ]^T, \\ \tilde{\boldsymbol{\zeta}}_{i,n,k} & \triangleq [\ \tilde{\boldsymbol{\zeta}}_{i,1,k}\ \ \tilde{\boldsymbol{\zeta}}_{i,2,k}\ \ \ldots\ \ \tilde{\boldsymbol{\zeta}}_{i,N,k}\ ]^T. \end{cases}$$

Furthermore, we use the Kronecker products

$$\mathcal{A}_0 \triangleq A_0 \otimes \mathbb{I},\ \mathcal{A}_1 \triangleq A_1 \otimes \mathbb{I}, \mathcal{A}_2 \triangleq A_2 \otimes \mathbb{I}, \quad (67)$$

where $\mathbb{I}$ is the identity matrix.

And, the following $N \times N$ block diagonal matrices and individual entries are of $M \times M$ are as follows

$$\mathcal{M} \triangleq \mathrm{diag}\{\mathbf{u}_1^T\mathbb{C},\ \mathbf{u}_2^T\mathbb{C}, \ldots,\ \mathbf{u}_N^T\mathbb{C}\}, \quad (68)$$

$$\mathcal{R}_w \triangleq \mathrm{diag}\{\tilde{\mathbf{w}}_{1,j}^T\mathbf{w}_{1,j},\ \tilde{\mathbf{w}}_{2,j}^T\mathbf{w}_{2,j}, \ldots,\ \tilde{\mathbf{w}}_{N,j}^T\mathbf{w}_{N,j}\}. \quad (69)$$

Let us define the gradient noise process $\boldsymbol{\eta}_{q_{n,k}}$ at each k as

$$\boldsymbol{\eta}_{q_{n,k}} = \widehat{\nabla_q \mathbb{J}}(\mathbf{q}, \mathbf{w}) - \nabla_q \mathbb{J}(\mathbf{q}, \mathbf{w}) = \mathbf{u}\,\mathbb{C}^T \tilde{\mathbf{w}}_{n,k-1}, \nu_{n,k}, \quad (70)$$

where $\widehat{\nabla_q \mathbb{J}}(\mathbf{q}, \mathbf{w})$ denotes as an approximate gradient vector and $\nu_{n,k}$ denotes as a Gaussian noise term.

From (66), we can verify that the block network variables satisfy the recursion as

$$\begin{cases} \tilde{\boldsymbol{\zeta}}_{i,n,k-1} = \mathcal{A}_1\, \tilde{\mathbf{q}}_{i,n,k-1} \\ \tilde{\mathbf{q}}_{i,n,k} = \mathcal{A}_2\, \tilde{\boldsymbol{\xi}}_{i,n,k}, \\ \tilde{\boldsymbol{\xi}}_{i,n,k} = \left[\mathcal{A}_0 - \mu_q \mathcal{M}\mathcal{R}_w^T\mathcal{M}\right], \tilde{\boldsymbol{\zeta}}_{i,n,k-1} + \mu_q, \tilde{\mathbf{d}}_{q_{n,k}} \\ \tilde{\mathbf{d}}_{q_{n,k}} = \lambda_{q_{n,k}}\, \tilde{\mathbf{d}}_{q_{n,k-1}} - \tilde{\mathbf{g}}_{q_{n,k}}, \\ \tilde{\mathbf{g}}_{q_{n,}} = \lambda_{q_{n,k}}\, \tilde{\mathbf{g}}_{q_{n,k-1}} + \mu_q\, \boldsymbol{\eta}_{q_{n,k}}. \end{cases}$$

So that, the network weight error vector $\tilde{\mathbf{q}}_{i,n,k}$ evolves as

$$\tilde{\mathbf{q}}_{i,n,k} = \mathcal{A}_2\big(\mathcal{A}_0 - \mu_q \mathcal{M}\mathcal{R}_w^T\mathcal{M}\big)\mathcal{A}_1 \tilde{\mathbf{q}}_{i,n,k-1} + \mu_q \mathcal{A}_2\, \tilde{\mathbf{d}}_{q_{n,k}}. \quad (71)$$

For a straightforward case, we can rewritten equivalently in terms of the gradient noise vector $\boldsymbol{\eta}_q$ as

$$\tilde{\mathbf{q}}_{i,n,k} = \mathbb{D}\, \tilde{\mathbf{q}}_{i,n,k-1} + \mu_q\, \mathcal{A}_2\, \tilde{\mathbf{d}}_{q_{n,k}} \quad (72)$$

$$\tilde{\mathbf{d}}_{q_{n,k}} = \lambda_{q_{n,k}}\, \tilde{\mathbf{d}}_{q_{n,k-1}} - \tilde{\mathbf{g}}_{q_{n,k}} \quad (73)$$

$$\tilde{\mathbf{g}}_{q_{n,k}} = \lambda_{q_{n,k}}\, \tilde{\mathbf{g}}_{q_{n,k-1}} + \mu_q\, \boldsymbol{\eta}_{q_{n,k}}, \quad (74)$$

where the constant matrix $\mathbb{D}$ is given as

$$\mathbb{D} \triangleq \mathcal{A}_2, \left(\mathcal{A}_0 - \mu_q \mathcal{M}\mathcal{R}_w^T\mathcal{M}\right), \mathcal{A}_1. \quad (75)$$

In a manner way, we consider the common minimisation w_0 for the non-operative and diffusion strategies in terms of three sets of combine coefficients $\{\Omega_{0,kl}, \Omega_{1,kl}, \Omega_{2,kl}\}$. We consider the linear part of DHSAF-NOGA algorithm that involves with

$$\begin{cases} \boldsymbol{\Phi}_{n,k-1} = \sum_{l\in\mathcal{N}_k} \Omega_{1,kl}\, \mathbf{w}_{l,k-1}, \\ \mathbf{w}_{n,k} = \sum_{l\in\mathcal{N}_k} \Omega_{2,kl} \boldsymbol{\Psi}_{l,k}, \\ \boldsymbol{\Psi}_{n,k} = \sum_{l\in\mathcal{N}_k} \Omega_{0,kl}\, \boldsymbol{\Phi}_{l,k-1} + \mu_w, \mathbf{d}_{w_{n,k}}, \\ \mathbf{d}_{w_{n,k}} = \lambda_{w_{n,k}}\, \mathbf{d}_{w_{n,k-1}} - \mathbf{g}_{w_{n,k}}, \\ \mathbf{g}_{w_{n,k}} = \lambda_{w_{n,k}}\, \mathbf{g}_{w_{n,k-1}} + \mu_w, \nabla_w \mathbb{J}(\mathbf{q}, \mathbf{w}) \\ = \lambda_{w_{n,k}}\, \mathbf{g}_{w_{n,k-1}} + \mu_w\, \mathbf{s}_{n,k}\, e_{n,k}, \end{cases}$$

where $\{\boldsymbol{\Phi}_{n,k-1}, \boldsymbol{\Psi}_{n,k}\}$ represent the $M \times 1$ vector.

Then, we orchestrate with each agent k following three errors as

$$\tilde{\mathbf{w}}_{n,k} \triangleq w_0 - \mathbf{w}_{n,k}, \quad (76)$$

$$\tilde{\boldsymbol{\Psi}}_{n,k} \triangleq w_0 - \boldsymbol{\Psi}_{n,k}, \quad (77)$$

$$\tilde{\boldsymbol{\Phi}}_{n,k} \triangleq w_0 - \boldsymbol{\Phi}_{n,k}, \quad (78)$$

where w_0 denotes the global minimisation.

Since, the deviation from w_0 can be measured by subtracting w_0 from both sides of (76) and substituting the local error $e_{n,k}$ and $\mathbf{s}_{n,k}$ as

$$e_{n,k} = d_{n,k} - \tilde{\boldsymbol{\Phi}}_{n,k-1}^T, \mathbf{s}_{n,k} \quad (79)$$

$$\mathbf{s}_{n,k} = \mathbf{u}_{n,k}^T\, \mathbb{C}\, \mathbf{q}_{i,n,k}. \quad (80)$$

From (76), we can reorganise the linear part of DHSAF-NOGA algorithm for $\tilde{\mathbf{w}}_{n,k}$ as

$$\begin{cases} \tilde{\boldsymbol{\Phi}}_{n,k-1} & = \sum_{l\in\mathcal{N}_k} \Omega_{1,kl}\, \tilde{\mathbf{w}}_{l,k-1} \\ \tilde{\mathbf{w}}_{n,k} & = \sum_{l\in\mathcal{N}_k} \Omega_{2,kl}\, \tilde{\boldsymbol{\Psi}}_{l,k}, \\ \tilde{\boldsymbol{\Psi}}_{n,k} & = \sum_{l\in\mathcal{N}_k} \Omega_{0,kl} \tilde{\boldsymbol{\Phi}}_{l,k-1} + \mu_w, \tilde{\mathbf{d}}_{w_{n,k}}, \\ \tilde{\mathbf{d}}_{w_{n,k}} & = \lambda_{w_{n,k}}\, \tilde{\mathbf{d}}_{w_{n,k}} - \tilde{\mathbf{g}}_{w_{n,k}} \\ \tilde{\mathbf{g}}_{w_{n,k}} & = \lambda_{w_{n,k}}\, \tilde{\mathbf{g}}_{w_{n,k}} \\ & \quad + \mu_w \mathbf{u}_{n,k}^T \mathbb{C}\tilde{\mathbf{q}}_{i,n,k}\big[d_{n,k} - \tilde{\boldsymbol{\Phi}}_{n,k-1}\mathbf{u}_{n,k}^T \mathbb{C}\tilde{\mathbf{q}}_{i,n,k}\big] \\ & = \lambda_{w_{n,k}}\, \tilde{\mathbf{g}}_{w_{n,k}} + \mu_w \mathbf{u}_{n,k}^T \mathbb{C}\, \tilde{\mathbf{q}}_{i,n,k}\, \nu_{n,k} \\ & \qquad - \mu_w \mathbf{u}_{n,k}^T\, \mathbb{C}\, \tilde{\mathbf{q}}_{i,n,k}^T\, \tilde{\boldsymbol{\Phi}}_{n,k-1}\mathbf{u}_{n,k}^T \mathbb{C}\tilde{\mathbf{q}}_{i,n,k}. \end{cases}$$

We examine the error vector into $N \times 1$ block vectors, whose individual entries are size of $M \times 1$ vectors as

$$\begin{cases} \tilde{\mathbf{w}}_{n,k} & \triangleq [\ \tilde{\mathbf{w}}_{1,k}\ \ \tilde{\mathbf{w}}_{2,k}\ \ \ldots\ \ \tilde{\mathbf{w}}_{N,k}\ ]^T, \\ \tilde{\boldsymbol{\Psi}}_{n,k} & \triangleq [\ \tilde{\boldsymbol{\Psi}}_{1,k}\ \ \tilde{\boldsymbol{\Psi}}_{2,k}\ \ \ldots\ \ \tilde{\boldsymbol{\Psi}}_{N,k}\ ]^T, \\ \tilde{\boldsymbol{\Phi}}_{n,k} & \triangleq [\ \tilde{\boldsymbol{\Phi}}_{1,k}\ \ \tilde{\boldsymbol{\Phi}}_{2,k}\ \ \ldots\ \ \tilde{\boldsymbol{\Phi}}_{N,k}\ ]^T. \end{cases}$$

Let us denote the gradient noise process $\boldsymbol{\eta}_{w_{n,k}}$ based on the linear part of DHSAF at each k as

$$\boldsymbol{\eta}_{w_{n,k}} = \widehat{\nabla_w \mathbb{J}}(\mathbf{q}, \mathbf{w}) - \nabla_w \mathbb{J}(\mathbf{q}, \mathbf{w}) = \mathbf{u}_{n,k}^T\, \mathbb{C}\tilde{\mathbf{q}}_{i,n,k-1}\, \nu_{n,k}, \quad (81)$$

where $\widehat{\nabla_w \mathbb{J}}(\mathbf{q}, \mathbf{w})$ is an approximate gradient vector.

From (81), we satisfy that the block network variables verify the recursions as

$$\begin{cases} \tilde{\boldsymbol{\Phi}}_{n,k-1} & = \mathcal{A}_1\, \tilde{\mathbf{w}}_{n,k-1} \\ \tilde{\mathbf{w}}_{n,k} & = \mathcal{A}_2\, \tilde{\boldsymbol{\Psi}}_{n,k}, \\ \tilde{\boldsymbol{\Psi}}_{n,k} & = \left[\mathcal{A}_0 - \mu_w \mathcal{M}\mathcal{R}_q^T\mathcal{M}\right] \tilde{\boldsymbol{\Phi}}_{n,k-1} + \mu_w, \tilde{\mathbf{d}}_{w_{n,k}} \\ \tilde{\mathbf{d}}_{w_{n,k}} & = \lambda_{w_{n,k}}\, \tilde{\mathbf{d}}_{w_{n,k-1}} - \tilde{\mathbf{g}}_{w_{n,k}} \\ \tilde{\mathbf{g}}_{w_{n,k}} & = \lambda_{w_{n,k}}\, \tilde{\mathbf{g}}_{w_{n,k-1}} + \mu_w, \boldsymbol{\eta}_{w_{n,k}}. \end{cases}$$

where $\mathcal{M}$ is given in (68) and $\mathcal{R}_q$ is defined by

$$\mathcal{R}_q \triangleq \mathrm{diag}\{\tilde{\mathbf{q}}_{i,1,k}^T \tilde{\mathbf{q}}_{i,1,k},\ \tilde{\mathbf{q}}_{i,2,k}^T \tilde{\mathbf{q}}_{i,2,k}, \ldots,\ \tilde{\mathbf{q}}_{i,N,k}^T \tilde{\mathbf{q}}_{i,N,k}\}. \quad (82)$$

Since, the network weight error vector $\tilde{\mathbf{w}}_{n,k}$ can be expressed as

$$\tilde{\mathbf{w}}_{n,k} = \mathcal{A}_2\big(\mathcal{A}_0 - \mu_w \mathcal{M}\mathcal{R}_q^T \mathcal{M}\big)\mathcal{A}_1 \tilde{\mathbf{w}}_{n,k-1} + \mu_w \mathcal{A}_2\, \tilde{\mathbf{d}}_{w_{n,k}}. \quad (83)$$

Therefore, we can rewritten equivalently in terms of the gradient noise vector $\boldsymbol{\eta}_{w_{n,k}}$ as

$$\tilde{\mathbf{w}}_{n,k} = \mathbb{G}\,\tilde{\mathbf{w}}_{n,k-1} + \mu_w\,\mathcal{A}_2\,\tilde{\mathbf{d}}_{w_{n,k}}, \tag{84}$$

$$\tilde{\mathbf{d}}_{w_{n,k}} = \lambda_{w_{n,k}}\,\tilde{\mathbf{d}}_{w_{n,k-1}} - \tilde{\mathbf{g}}_{w_{n,k}} \tag{85}$$

$$\tilde{\mathbf{g}}_{w_{n,k}} = \lambda_{w_{n,k}}\,\tilde{\mathbf{g}}_{w_{n,k-1}} + \mu_w, \boldsymbol{\eta}_{w_{n,k}}, \tag{86}$$

where the constant matrix $\mathbb{G}$ is defined as

$$\mathbb{G} \triangleq \mathcal{A}_2\left(\mathcal{A}_0 - \mu_w \mathcal{M}\mathcal{R}_q^T\mathcal{M}\right)\mathcal{A}_1. \tag{87}$$

B. PERFORMANCE ANALYSIS

We examine the performance analysis of proposed CTA-DHSAF-NOGA and ATC-DHSAF-NOGA algorithms based on the mean square value of linear tap-weight estimation.

Firstly, Let us consider the estimated error vectors $\boldsymbol{\varepsilon}_{n,k}$ and $\tilde{\boldsymbol{\varepsilon}}_{n,k}$ at sample n and agent k from CTA-DHSAF-NOGA algorithm by [34]

$$\boldsymbol{\varepsilon}_{n,k} = \hat{\mathbf{w}}_{n,k} - \mathbf{w}_{\text{opt}}, \tag{88}$$

$$\tilde{\boldsymbol{\varepsilon}}_{n,k} = \boldsymbol{\Psi}_{n,k} - \mathbf{w}_{\text{opt}}, \tag{89}$$

where $\mathbf{w}_{\text{opt}}$ defines as the optimum linear tap-weight coefficient, $\boldsymbol{\Psi}_{n,k}$ is the tap-weight vector at the combination step and $\hat{\mathbf{w}}_{n,k}$ is the estimated coefficient at the adaptation step.

Following [30], [35], we rewrite the linear tap-weight vector $\hat{\mathbf{w}}_{n,k}$ in the recursive form as

$$\hat{\mathbf{w}}_{n,k} = \boldsymbol{\Psi}_{n,k} + \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\,\hat{\mathbf{s}}_{n,r}\,\hat{e}_{n,r}, \tag{90}$$

where $\hat{e}_{n,k}$ is *a priori* estimated error as

$$\hat{e}_{n,k} = d_{n,k} - \boldsymbol{\Psi}_{n,k}^T\,\hat{\mathbf{s}}_{n,k}, \tag{91}$$

where $\mathbf{s}_{n,k}$ is the output of nonlinear filtering part of CTA-DHSAF-NOGA structure.

Subtracting both sides of (90) with $\mathbf{w}_{\text{opt}}$ and then to eradicate $\hat{\mathbf{w}}_{n,k}$ by (91), we get

$$\begin{aligned}\left(\hat{\mathbf{w}}_{n,k} - \mathbf{w}_{\text{opt}}\right) = \left(\boldsymbol{\Psi}_{n,k} - \mathbf{w}_{\text{opt}}\right) &+ \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\hat{\mathbf{s}}_{n,r}\left\{d_{n,r} - \boldsymbol{\Psi}_{n,r}^T\,\hat{\mathbf{s}}_{n,r}\right\} \\ &+ \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\,\hat{\mathbf{s}}_{n,r}\left(\mathbf{w}_{\text{opt}}^T\,\hat{\mathbf{s}}_{n,r}\right) \\ &- \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\,\hat{\mathbf{s}}_{n,r}\left(\mathbf{w}_{\text{opt}}^T\,\hat{\mathbf{s}}_{n,r}\right).\end{aligned} \tag{92}$$

Replacing (88), (89) in (92), we arrive at

$$\begin{aligned}\boldsymbol{\varepsilon}_{n,k} = \tilde{\boldsymbol{\varepsilon}}_{n,k} - \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\,\hat{\mathbf{s}}_{n,r}^T\,\tilde{\boldsymbol{\varepsilon}}_{n,r}, \hat{\mathbf{s}}_{n,r} \\ + \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\,\hat{\mathbf{s}}_{n,r}\left\{d_{n,r} - \mathbf{w}_{\text{opt}}^T\,\hat{\mathbf{s}}_{n,r}\right\}.\end{aligned} \tag{93}$$

So, the estimated error $\boldsymbol{\varepsilon}_{n,k}$ can be reorganised as

$$\begin{aligned}\boldsymbol{\varepsilon}_{n,k} = \left[\mathbf{I} - \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\,\hat{\mathbf{s}}_{n,r}^T\,\hat{\mathbf{s}}_{n,r}\right]\tilde{\boldsymbol{\varepsilon}}_{n,k} \\ + \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\,\hat{\mathbf{s}}_{n,r}\,e_{\text{opt}},\end{aligned} \tag{94}$$

where e_{opt} is the optimum error as

$$e_{\text{opt}} = d_{n,k} - \mathbf{w}_{\text{opt}}^T\,\hat{\mathbf{s}}_{n,k}, \tag{95}$$

where $\mathbf{s}_{n,k}$ is given in (34).

Hence, we clarify (94) by the orthogonality principle as $E\{\mathbf{s}_{n,k}, e_{\text{opt}}\} \simeq 0$, we get

$$\boldsymbol{\varepsilon}_{n,k} = \left[\mathbf{I} - \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\,\hat{\mathbf{s}}_{n,r}^T\,\hat{\mathbf{s}}_{n,r}\right]\tilde{\boldsymbol{\varepsilon}}_{n,k}. \tag{96}$$

It is noticed that the convergence rate depends on the input of the linear filtering part.

***Assumption** 1:* We assume the condition for convergence,

$$\lim_{k\to\infty} E\{\|\boldsymbol{\varepsilon}_{n,k}\|\} = 0,$$

$$\lim_{k\to\infty} E\{\|\tilde{\boldsymbol{\varepsilon}}_{n,k}\|\} = 0,$$

$$\lim_{k\to\infty} E\{\,\hat{\mathbf{w}}_{n,k}\,\} = \lim_{k\to\infty} E\{\,\check{\mathbf{w}}_{n,k}\,\} = \mathbf{w}_{\text{opt}},$$

where $E\{\cdot\}$ is the expectation operator and $\|\cdot\|$ is the Euclidean norm operator.

Substituting $\boldsymbol{\Psi}_{n,k}$ in (89) into (91), we have the estimated error $\hat{e}_{n,k}$ in terms of the optimum error e_{opt} in (95) by

$$\hat{e}_{n,k} = e_{\text{opt}} + \tilde{\boldsymbol{\varepsilon}}_{n,k}^T\,\hat{\mathbf{s}}_{n,k}. \tag{97}$$

Accordingly, we consider the mean square error of (97) as

$$\begin{aligned}\mathbb{J}(n,k) &= E\{|\hat{e}_{n,k}|^2\} \\ &= E\{|e_{\text{opt}}|^2\} + 2\,E\{\tilde{\boldsymbol{\varepsilon}}_{n,k}^T\,\hat{\mathbf{s}}_{n,k}\,e_{\text{opt}}\} \\ &\quad + E\{\tilde{\boldsymbol{\varepsilon}}_{n,k}^T\,\tilde{\boldsymbol{\varepsilon}}_{n,k}\,\hat{\mathbf{s}}_{n,k}^T\,\hat{\mathbf{s}}_{n,k}\}.\end{aligned} \tag{98}$$

Following Assumption 1, we may write

$$\mathbb{J}(n,k) = \mathbb{J}_{\text{min}}(n,k) + \mathbb{J}_{\text{ex}}(n,k), \tag{99}$$

where $\mathbb{J}_{\text{min}}(n,k)$ denotes as the minimum mean square error (MMSE) implemented by the optimum Wiener solution and $\mathbb{J}_{\text{ex}}(n,k)$ is the excess mean square error (EMSE) of proposed CTA-DHSAF-NOGA algorithm as

$$\mathbb{J}_{\text{min}}(n,k) = E\{|e_{\text{opt}}|^2\} + 2\,E\{\tilde{\boldsymbol{\varepsilon}}_{n,k}^T\,\hat{\mathbf{s}}_{n,k}\,e_{\text{opt}}\}, \tag{100}$$

$$\mathbb{J}_{\text{ex}}(n,k) = E\{\tilde{\boldsymbol{\varepsilon}}_{n,k}^T\,\tilde{\boldsymbol{\varepsilon}}_{n,k}\,\hat{\mathbf{s}}_{n,k}^T\,\hat{\mathbf{s}}_{n,k}\}, \tag{101}$$

where $\hat{\mathbf{s}}_{n,k} = \mathbf{u}_{n,k}^T\mathbb{C}, \boldsymbol{\xi}_{i,n,k}$ and $\boldsymbol{\xi}_{i,n,k}$ is the combination step of $\hat{\mathbf{q}}_{i,n,k}$.

We perform an autocorrelation matrix $\hat{\mathbb{R}}_{\mathbf{s}} = E\{\hat{\mathbf{s}}_{n,k}^T, \hat{\mathbf{s}}_{n,k}\}$, where $\hat{\mathbf{s}}_{n,k}$ is the input of linear filtering part. Therefore, $\mathbb{J}_{\text{ex}}(n,k)$ can be obtained by

$$\mathbb{J}_{\text{ex}}(n,k) = E\{\tilde{\boldsymbol{\varepsilon}}_{n,k}^T\,\hat{\mathbb{R}}_{\mathbf{s}}\,\tilde{\boldsymbol{\varepsilon}}_{n,k}\}, \tag{102}$$

where $\tilde{\boldsymbol{\varepsilon}}_{n,k}$ is the estimated weight error in (89).

In a similar fashion, the performance analysis of proposed ATC-DHSAF-NOGA algorithm based on the mean square value of linear tap-weight estimation is considered.

We recognise the estimated error vectors $\boldsymbol{\epsilon}_{n,k}$ and $\tilde{\boldsymbol{\epsilon}}_{n,k}$ at sample n and agent k by [34]

$$\boldsymbol{\epsilon}_{n,k} = \check{\mathbf{w}}_{n,k} - \mathbf{w}_{\text{opt}}, \tag{103}$$

$$\tilde{\boldsymbol{\epsilon}}_{n,k} = \tilde{\boldsymbol{\Psi}}_{n,k} - \mathbf{w}_{\text{opt}}, \tag{104}$$

where $\mathbf{w}_{\text{opt}}$ denotes as the optimum linear tap-weight coefficient, $\tilde{\boldsymbol{\Psi}}_{n,k}$ is the estimated tap-weight vector at the adaptation step and then $\check{\mathbf{w}}_{n,k}$ is linear weight vector at the combination step.

We reorganise the linear tap-weight vector $\tilde{\boldsymbol{\Psi}}_{n,k}$ into a recursive form following [30], [35] as

$$\tilde{\boldsymbol{\Psi}}_{n,k} = \check{\mathbf{w}}_{n,k} + \mu_w \sum_{r=1}^{k} \tilde{\lambda}_w^{k-i}\, \tilde{\mathbf{s}}_{n,r}\, \check{e}_{n,r}, \tag{105}$$

where $\check{e}_{n,k}$ is *a priori* estimated error as

$$\check{e}_{n,k} = d_{n,k} - \check{\mathbf{w}}_{n-1,k}^{T}\, \tilde{\mathbf{s}}_{n,k}, \tag{106}$$

where $\tilde{\mathbf{s}}_{n,k}$ is the output of nonlinear filtering part of ATC-DHSAF-NOGA structure.

Subtracting both sides of (105) by $\mathbf{w}_{\text{opt}}$ and replacing (103), (104), we have

$$\tilde{\boldsymbol{\epsilon}}_{n,k} = \boldsymbol{\epsilon}_{n,k} - \mu_w \sum_{r=1}^{k} \tilde{\lambda}_w^{k-i}\, \tilde{\mathbf{s}}_{n,r}^{T} \tilde{\boldsymbol{\epsilon}}_{n,r}\, \tilde{\mathbf{s}}_{n,r} + \mu_w \sum_{r=1}^{k} \tilde{\lambda}_w^{k-i} \left\{ d_{n,r} - \mathbf{w}_{\text{opt}}^{T}\, \tilde{\mathbf{s}}_{n,r} \right\}. \tag{107}$$

So, the estimated error $\tilde{\boldsymbol{\epsilon}}_{n,k}$ can be rewritten as

$$\tilde{\boldsymbol{\epsilon}}_{n,k} = \Big[\mathbf{I} - \mu_w \sum_{r=1}^{k} \tilde{\lambda}_w^{k-i}\, \tilde{\mathbf{s}}_{n,r}^{T} \tilde{\mathbf{s}}_{n,r}\Big]\, \boldsymbol{\epsilon}_{n,k} - \mu_w, \sum_{r=1}^{k} \tilde{\lambda}_w^{k-i}\, \tilde{\mathbf{s}} + \mu_w \sum_{r=1}^{k} \tilde{\lambda}_w^{k-i}\, \tilde{\mathbf{s}}_{n,r} \tilde{e}_{\text{opt}}, \tag{108}$$

where $\tilde{e}_{\text{opt}}$ is the optimum error as

$$\tilde{e}_{\text{opt}} = d_{n,k} - \mathbf{w}_{\text{opt}}^{T}\, \tilde{\mathbf{s}}_{n,k}, \tag{109}$$

where $\tilde{\mathbf{s}}_{n,k}$ is given in (47).

Consequently, we simplify (108) by the orthogonality principle as $E\{\tilde{\mathbf{s}}_{n,k}, \tilde{e}_{\text{opt}}\} \simeq 0$, we get

$$\tilde{\boldsymbol{\epsilon}}_{n,k} = \Big[\mathbf{I} - \mu_w \sum_{r=1}^{k} \lambda_w^{k-i}\, \tilde{\mathbf{s}}_{n,r}^{T}\, \tilde{\mathbf{s}}_{n,r}\Big]\, \boldsymbol{\epsilon}_{n,k}. \tag{110}$$

Substituting $\tilde{\boldsymbol{\Psi}}_{n,k}$ in (104) into (105), we can evaluate the estimated error $\check{e}_{n,k}$ in terms of the optimum error $\tilde{e}_{\text{opt}}$ in (109) by

$$\check{e}_{n,k} = \tilde{e}_{\text{opt}} + \tilde{\boldsymbol{\epsilon}}_{n,k}^{T}\, \tilde{\mathbf{s}}_{n,k}. \tag{111}$$

We represent the mean square error of (111) as

$$\begin{aligned} \tilde{\mathbb{J}}(n,k) &= E\{|\check{e}_{n,k}|^2\} \\ &= E\{|e_{\text{opt}}|^2\} + 2, E\{\boldsymbol{\epsilon}_{n,k}^{T}\, \tilde{\mathbf{s}}_{n,k}\, e_{\text{opt}}\} \\ &\quad + E\{\boldsymbol{\epsilon}_{n,k}^{T}\, \tilde{\mathbb{R}}_s\, \boldsymbol{\epsilon}_{n,k}\} \end{aligned} \tag{112}$$

where $\tilde{\mathbb{R}}_s = E\{\tilde{\mathbf{s}}_{n,k}^{T}, \tilde{\mathbf{s}}_{n,k}\}$.

Following Assumption 1, we may write

$$\tilde{\mathbb{J}}_{n,k} = \tilde{\mathbb{J}}_{\min}(n,k) + \tilde{\mathbb{J}}_{\text{ex}}(n,k), \tag{113}$$

where $\tilde{\mathbb{J}}_{\min}(n,k)$ denotes as the MMSE implemented by the optimum Wiener solution and $\tilde{\mathbb{J}}_{\text{ex}}(n,k)$ is the EMSE of proposed ATC-DHSAF-NOGA algorithm as

$$\tilde{\mathbb{J}}_{\min}(n,k) = E\{|e_{\text{opt}}|^2\} + 2, E\{\boldsymbol{\epsilon}_{n,k}^{T}\, \tilde{\mathbf{s}}_{n,k}\, e_{\text{opt}}\}, \tag{114}$$

$$\tilde{\mathbb{J}}_{\text{ex}}(n,k) = E\{\boldsymbol{\epsilon}_{n,k}^{T}\, \tilde{\mathbb{R}}_s\, \boldsymbol{\epsilon}_{n,k}\}, \tag{115}$$

and $\tilde{\mathbf{s}}_{n,k} = \mathbf{u}_{n,k}^{T} \mathbb{C}, \check{\mathbf{q}}_{i,n,k}$ and $\boldsymbol{\epsilon}_{n,k}$ is the estimated weight error in (103).

VII. SIMULATION DESIGN AND RESULTS

In this section, the experiments are conducted with the system identification through computer simulations with the additive white Gaussian noise in order to verify the proposed nonlinear DHSAF adaptive filtering. The input colour signal x_n composes of 5,000 samples by averaging over 20 times can be generated by the following [28]

$$x_n = \omega \cdot x_{n-1} + \sqrt{1-\omega^2}\, \varphi, \tag{116}$$

where φ is a unitary variance of zero mean white Gaussian noise. The local correlation coefficients ω between the adjacent samples [13] is as $0 \le \omega < 1$. The network topology is determined in Figure 4 and the topology weight Ω_{kl} is shown in Table 2, where $N_k = 8$ agents. The desired signal $d_{n,k}$

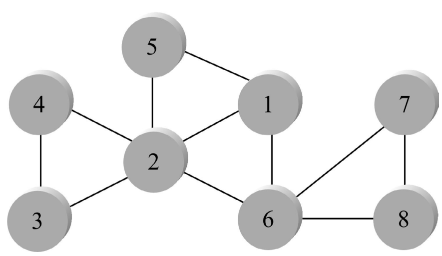

FIGURE 4. Network topology with $N_k = 8$ agents including the noise variance at each agent.

TABLE 2. Topology weight Ω_{kl} of each agents with $N_k = 8$.

Agent	#1	#2	#3	#4	#5	#6	#7	#8
#1	-	0.1	-	-	0.8	0.1	-	-
#2	0.1	-	0.2	0.2	0.2	0.3	-	-
#3	-	0.2	-	0.8	-	-	-	-
#4	-	0.2	0.8	-	-	-	-	-
#5	0.8	0.2	-	-	-	-	-	-
#6	0.1	0.3	-	-	-	-	0.3	0.3
#7	-	-	-	-	-	0.3	-	0.7
#8	-	-	-	-	-	0.3	0.7	-

is defined in (1), where the noise variances at each k agent are assigned in terms of signal to noise ratio (SNR) at each k agent as $SNR_k = \{35, 25, 25, 30, 30, 25, 35, 30\}$dB used for simulation.

The constant matrix $\mathbf{C}_B$ is called a *B-spline* matrix as given by [13]

$$\mathbf{C}_B = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}.$$

For the simulation following [28], a series of the DHSAF-based algorithms are compared with the non-cooperative HSAF-NOGA and based on HSAF-LMS [13] algorithms. The choice of used parameters are of the proposed HSAF-based algorithm are followed in [13], [36] with the learning rate set to $\mu_q = \mu_w = 10^{-3}$, while these parameters of NOGA-based algorithm are referred to [5] with the learning rate set between $\mu_q = \mu_w = 10^{-4}$ and $\mu_q = \mu_w = 10^{-3}$ by randomly trial and error of learning rate to get the best solution for each algorithm.

For the proposed DHSAF-based algorithm, the initial parameters are as follows: $\Delta x = 0.2$, $\delta = 0.001$, number of tap length $M = 7$, number of control points $Q = 23$, $\lambda_w(0) = 1.525 \times 10^{-4}$ and $\lambda_q(0) = 1.425 \times 10^{-4}$. Initial parameters for the DHSAF-LMS are as: $\mu_w = 1.215 \times 10^{-4}$, and $\mu_q = 1.515 \times 10^{-4}$. Other initial parameters for HSAF-NOGA algorithm are as: $\mu_w = 1.525 \times 10^{-2}$ and $\mu_q = 3.125 \times 10^{-3}$, and of HSAF-LMS algorithms are as: $\mu_w = 7.025 \times 10^{-3}$ and $\mu_q = 3.225 \times 10^{-3}$. The local correlation coefficients is set as $\omega = 0.10, 0.70$.

Mean square error (MSE) is used for performance index. Table 3 presents the summary of MSE of each algorithm for simulations. Figure 5 and Figure 6 show the MSE trajectories curves of proposed CTA-DHSAF-NOGA and CTA-DHSAF-LMS algorithms compared with the HSAF-NOGA and HSAF-LMS algorithms at agent #6 using the different local correlation coefficients $\omega = 0.10, 0.70$ in (116). And Figure 7 and Figure 8 show the MSE curves of proposed ATC-DHSAF-NOGA and ATC-DHSAF-LMS algorithms compared with the HSAF-NOGA and HSAF-LMS algorithms at agent #6 using $\omega = 0.10, 0.70$ in (116).

TABLE 3. Summary of MSE for simulations.

Algorithm	Mean square error (MSE)
CTA-DHSAF-NOGA	$\hat{e}^2_{n,k} = \frac{1}{N}\sum_n^N \lvert d_{n,k} - \mathbf{\Psi}^T_{n,k}\mathbf{s}_{n,k}\rvert^2$
ATC-DHSAF-NOGA	$\check{e}^2_{n,k} = \frac{1}{N}\sum_n^N \lvert d_{n,k} - \check{\mathbf{w}}^T_{n,k}\tilde{\mathbf{s}}_{n,k}\rvert^2$
CTA-DHSAF-LMS	$e^2_{n,k} = \frac{1}{N}\sum_n^N \lvert d_{n,k} - \mathbf{w}_{n,k}\mathbf{u}^T_{n,k}\mathbb{C}\mathbf{q}_{i,n,k}\rvert^2$
ATC-DHSAF-LMS	$\tilde{e}^2_{n,k} = \frac{1}{N}\sum_n^N \lvert d_{n,k} - \tilde{\mathbf{w}}_{n,k}\mathbf{u}^T_{n,k}\mathbb{C}\tilde{\mathbf{q}}_{i,n,k}\rvert^2$
HSAF-NOGA	$e^2_n = \frac{1}{N}\sum_n^N \lvert d_n - \mathbf{w}_n\mathbf{s}_n\rvert^2$
HSAF-LMS [13]	$\acute{e}^2_n = \frac{1}{N}\sum_n^N \lvert d_n - \acute{\mathbf{w}}_n\mathbf{s}_n\rvert^2$ $\acute{\mathbf{w}}_n = \acute{\mathbf{w}}_{n-1} + \mu\,\mathbf{s}_n\,e_n$

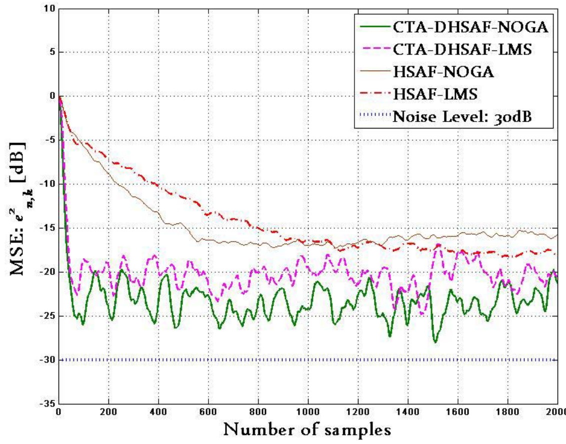

FIGURE 5. MSE curves for the HSAF-NOGA and HSAF-LMS at agent #6 and proposed CTA-DHSAF-NOGA algorithm using $\mu_w = 3.025 \times 10^{-3}$, and $\mu_q = 1.225 \times 10^{-3}$, where $\omega = 0.10$ in (116).

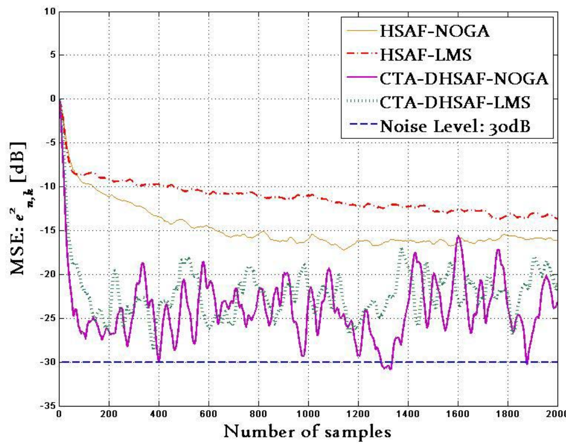

FIGURE 6. MSE curves for the HSAF-NOGA and HSAF-LMS at agent #6 and proposed CTA-DHSAF-NOGA algorithm using $\mu_w = 3.025 \times 10^{-3}$, and $\mu_q = 1.225 \times 10^{-3}$, where $\omega = 0.70$ in (116).

To evaluate the EMSE of proposed CTA-DHAF-NOGA algorithm in (102), we modify the estimated error vector $\tilde{\boldsymbol{\varepsilon}}_{n,k}$ in (89) by the instantaneous weight deviation $\tilde{\boldsymbol{\varepsilon}}^d_{n,k}$ as

$$\tilde{\boldsymbol{\varepsilon}}^d_{n,k} = \lvert \mathbf{\Psi}_{n,k} - \hat{\mathbf{w}}_{n-1,k}\rvert. \tag{117}$$

Therefore, $\mathbb{J}^{\mathrm{CTA}}_{\mathrm{ex}}(n,k)$ can be obtained by

$$\mathbb{J}^{\mathrm{CTA}}_{\mathrm{ex}}(n,k) = \lvert \mathbf{\Psi}_{n,k} - \hat{\mathbf{w}}_{n-1,k}\rvert^T\,\hat{\mathbb{R}}_s\,\lvert \mathbf{\Psi}_{n,k} - \hat{\mathbf{w}}_{n-1,k}\rvert, \tag{118}$$

where $\hat{\mathbb{R}}_{\mathbf{s}} = \hat{\mathbf{s}}^T_{n,k}, \hat{\mathbf{s}}_{n,k}$ and $\hat{\mathbf{s}}_{n,k}$ is given in (34).

Similarly, the EMSE of proposed ATC-DHAF-NOGA algorithm in (115) is calculated by replacing the estimated error vector $\boldsymbol{\varepsilon}_{n,k}$ in (103) with the instantaneous weight deviation $\boldsymbol{\varepsilon}^d_{n,k}$ as

$$\boldsymbol{\varepsilon}^d_{n,k} = \lvert \check{\mathbf{w}}_{n,k} - \check{\mathbf{w}}_{n-1,k}\rvert. \tag{119}$$

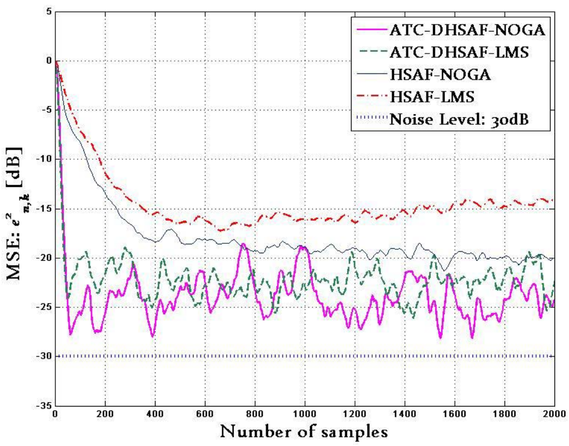

FIGURE 7. MSE curves for the HSAF-NOGA and HSAF-LMS at agent #6 and proposed ATC-DHSAF-NOGA algorithm using $\mu_w = 2.525 \times 10^{-4}$, and $\mu_q = 1.225 \times 10^{-4}$, where $\omega = 0.10$.

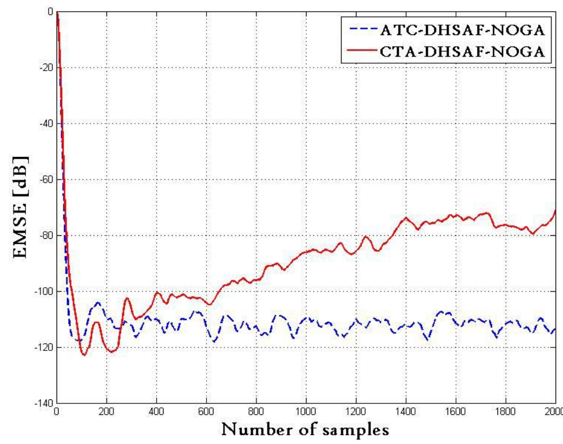

FIGURE 9. EMSE curves of proposed CTA-DHSAF-NOGA and ATC-DHSAF-NOGA algorithms from (118) and (120).

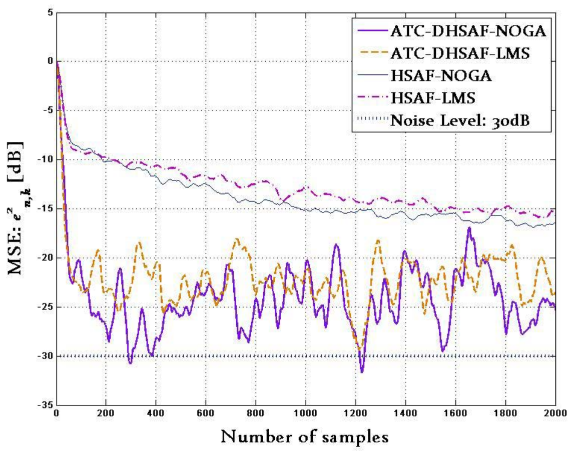

FIGURE 8. MSE curves for the HSAF-NOGA and HSAF-LMS at agent #6 and proposed ATC-DHSAF-NOGA algorithm using $\mu_w = 2.525 \times 10^{-4}$, and $\mu_q = 1.225 \times 10^{-4}$, where $\omega = 0.70$.

Therefore, $\mathbb{J}_{\text{ex}}^{\text{ATC}}(n, k)$ can be expressed by

$$\mathbb{J}_{\text{ex}}^{\text{ATC}}(n, k) = |\check{\mathbf{w}}_{n,k} - \check{\mathbf{w}}_{n-1,k}|^T \, \tilde{\mathbb{R}}_s \, |\check{\mathbf{w}}_{n,k} - \check{\mathbf{w}}_{n-1,k}|, \tag{120}$$

where $\tilde{\mathbb{R}}_{\mathbf{s}} = \tilde{\mathbf{s}}_{n,k}^T, \tilde{\mathbf{s}}_{n,k}$ and $\tilde{\mathbf{s}}_{n,k}$ is given in (47).

Figure 9 shows the EMSE curves of proposed algorithms that can evaluate from (118) and (120) using $\omega = 0.10$ in (116).

This paper presents a novel nonlinear-linear model of DHSAF structure based on NOGA algorithm following the DSAF in [28]. The advantage of Hammerstein model [13] is that can determine properly the high-order nonlinearity with the low order polynomial and low computational cost. As a remark in [13], the behaviour of proposed DHSAF structure is different from DSAF model [1], where the derivative of mean-square error cost function with respect to the coefficients of spline control points in the first nonlinear filtering part of DHSAF model depends on the present input signal concerned with the diffusion cooperative strategies on the combination and adaptation steps.

As mentioned in [26], the NLMS algorithm is applied to protect a large difference of tap-weight coefficients that should cause notably a fluctuation for nonlinear system. In order to decrease the sensitivity model of proposed DHSAF algorithm, the NOGA algorithm is furnished according to the diffusion adaptation. In addition, the proposed DHSAF-NOGA algorithm is similar to HSAF based on the stochastic gradient approach as HSAF-LMS algorithm [13] implemented in the form of diffusion adaptation on DSAF model based on the LMS algorithm as DSAF-LMS [28].

Noted that, the proposed CTA-DHSAF-NOGA and ATC-DHSAF-NOGA algorithms can testify the fast convergence with the non-cooperative HSAF-NOGA algorithm and standard HSAF-LMS algorithm in the multi-agent distributed network, similarly to DSAF-LMS compared with the SAF-LMS algorithm in [28]. According to Table 2, the topology weight Ω_{kl} of each agents with $N_k = 8$ are shown that each agent is influenced for connected agents, which is caused that the MSE curves of the non-cooperative HSAF-NOGA and HSAF-LMS algorithms in the simulation results are shown with slow convergence. While the proposed diffusion-based adaptation algorithms can manage by the information combined with their own neighbour agents in terms of CTA and ATC algorithm for the nonlinear-linear HSAF network.

In comparison with the state of art solution as DSAF-LMS algorithm [28], the proposed DHSAF-NOGA algorithm with a small increasing complexity shown in Table 1 has been orchestrated with two diffusion and two stochastic gradient steps requiring in approximately twice time of computations compared with the standard HSAF-LMS algorithm.

VIII. CONCLUSION

We have introduced a set of DHSAF-NOGA algorithm as the CTA and ATC strategies over the distributed network.

Diffusion-based adaptation on the multi-agent network based on a joint optimisation is considered. Diffusion adaptation framework on memoryless function in terms of adaptive look-up table and linear filter named *HSAF* has been modified from the NOGA algorithm. A set of adaptive diffusion strategies with the CTA and ATC algorithms that have been derived by DHSAF-NOGA algorithm. The network stability and performance over the MSE networks of proposed DHSAF-NOGA algorithms have been derived. Experiment results depict that DHSAF-NOGA algorithm can learn underlying the nonlinear Hammerstein model compared with a non-cooperative solution and existing techniques.

For the real-time dynamic system, diffusion strategy on Hammerstein spline adaptive filtering model is being investigated in the adaptive signal processing for wireless communications and data analysis over the distributed network.

ACKNOWLEDGMENT

The author would like to extend the sincere appreciation to Dr. Theerayod Wiangtong at the King Mongkut's Institute of Technology Ladkrabang and Dr. Sethakarn Prongnuch at Suan Sunandha Rajabhat University for their support and discussion in this research. She would also appreciates the precious and helpful suggestions from all anonymous reviewers to improve the manuscript.

REFERENCES

[1] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Spline adaptive filters: Theory and applications," in *Adaptive Learning Methods for Nonlinear System Modeling*. London, U.K.: Butterworth, 2018, ch. 3, pp. 47–69.

[2] S. Guan and Z. Li, "Normalised spline adaptive filtering algorithm for nonlinear system identification," *Neural Process. Lett.*, vol. 46, no. 2, pp. 595–607, Oct. 2017, doi: 10.1007/s11063-017-9606-6.

[3] M. Scarpiniti, D. Comminiello, and A. Uncini, "Convex combination of spline adaptive filters," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, A Coruña, Spain, Sep. 2019, pp. 1–5, doi: 10.23919/EUSIPCO.2019.8903134.

[4] C. Liu, C. Peng, X. Tang, and X. Liu, "Two variants of the IIR spline adaptive filter for combating impulsive noise," *EURASIP J. Adv. Signal Process.*, vol. 2019, no. 1, pp. 1–15, Feb. 2019, doi: 10.1186/s13634-019-0605-9.

[5] T. Wiangtong and S. Sitjongsataporn, "Analysis of normalized orthogonal gradient adaptive algorithm based on spline adaptive filtering for smart communication technology," *J. Mobile Multimedia*, vol. 17, pp. 657–672, Jun. 2021, doi: 10.13052/jmm1550-4646.1748.

[6] S. Scardapane, J. Chen, and C. Richard, "Adaptation and learning over networks for nonlinear system modeling," in *Adaptive Learning Methods for Nonlinear System Modeling*, D. Comminiello and J.C. Principe, Eds. Amsterdam, The Netherlands: Elsevier, 2018, ch. 10, pp. 223–242.

[7] R. A. D. Prado, F. D. R. Henriques, and D. B. Haddad, "Sparsity-aware distributed adaptive filtering algorithms for nonlinear system identification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8489173.

[8] K.-L. Yin, Y.-F. Pu, and L. Lu, "Censored regression distributed functional link adaptive filtering algorithm over nonlinear networks," *Signal Process.*, vol. 190, pp. 1–15, Jan. 2022, doi: 10.1016/j.sigpro.2021.108318.

[9] Y. Yu, H. He, T. Yang, X. Wang, and R. C. de Lamare, "Diffusion normalized least mean M-estimate algorithms: Design and performance analysis," *IEEE Trans. Signal Process.*, vol. 68, pp. 2022–2199, 2020, doi: 10.1109/TSP.2020.2983905.

[10] Y. Yu, H. Zhao, R. C. de Lamare, Y. Zakharov, and L. Lu, "Robust distributed diffusion recursive least squares algorithms with side information for adaptive networks," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1566–1581, Mar. 2019, doi: 10.1109/TSP.2019.2893846.

[11] Q. Liu and Y. He, "Robust Geman–McClure based nonlinear spline adaptive filter against impulsive noise," *IEEE Access*, vol. 8, pp. 22571–22580, 2020, doi: 10.1109/ACCESS.2020.2969219.

[12] W. Lu, L. Zhang, J. Chen, and J. Chen, "Generalized combined nonlinear adaptive filters: From the perspective of diffusion adaptation over networks," *Signal Process.*, vol. 172, pp. 1–12, Jul. 2020, doi: 10.1016/j.sigpro.2020.107507.

[13] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Hammerstein uniform cubic spline adaptive filters: Learning and convergence properties," *Signal Process.*, vol. 100, pp. 112–123, Jul. 2014, doi: 10.1016/j.sigpro.2014.01.019.

[14] F. Albu and K. Nishikawa, "New iterative kernel algorithms for nonlinear acoustic echo cancellation," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Hong Kong, Dec. 2015, pp. 737–739, doi: 10.1109/APSIPA.2015.7415369.

[15] S. Van Vaerenbergh and L. A. Azpicueta-Ruiz, "Kernel-based identification of Hammerstein systems for nonlinear acoustic echo-cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 3739–3743, doi: 10.1109/ICASSP.2014.6854300.

[16] C. Liu, Z. Zhang, and X. Tang, "Sign normalised Hammerstein spline adaptive filtering algorithm in an impulsive noise environment," *Neural Process. Lett.*, vol. 50, no. 1, pp. 477–496, Aug. 2019, doi: 10.1007/s11063-019-09996-6.

[17] S. Prongnuch and S. Sitjongsataporn, "Stability and steady-state performance of Hammerstein spline adaptive filter based on stochastic gradient algorithm," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 3, pp. 112–123, Jun. 2020, doi: 10.22266/ijies2020.0630.11.

[18] P. P. Campo, D. Korpi, L. Anttila, and M. Valkama, "Nonlinear digital cancellation in full-duplex devices using spline-based Hammerstein model," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–7, doi: 10.1109/GLOCOMW.2018.8644362.

[19] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012, doi: 10.1109/TSP.2012.2198470.

[20] R. Nassif, C. Richard, J. Chen, and A. H. Sayed, "Distributed diffusion adaptation over graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 4129–4133, doi: 10.1109/ICASSP.2018.8461574.

[21] S. Xie and L. Guo, "Analysis of distributed adaptive filters based on diffusion strategies over sensor networks," *IEEE Trans. Autom. Control*, vol. 63, no. 11, pp. 3643–3658, Nov. 2018, doi: 10.1109/TAC.2018.2799567.

[22] A. Rastegarnia, "Reduced-communication diffusion RLS for distributed estimation over multi-agent networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 1, pp. 177–181, Jan. 2020, doi: 10.1109/TCSII.2019.2899194.

[23] J.-H. Seo, S. M. Jung, and P. Park, "Diffusion proportionate affine projection sign algorithm for distributed estimation over network," in *Proc. 14th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Phuket, Thailand, Jun. 2017, pp. 636–639, doi: 10.1109/ECTICon.2017.8096318.

[24] M. Ferrer, M. de Diego, G. Piñero, and A. Gonzalez, "Active noise control over adaptive distributed networks," *Signal Process.*, vol. 107, pp. 82–95, Feb. 2015, doi: 10.1016/j.sigpro.2014.07.026.

[25] F. Albu, "The constrained stability least mean square algorithm for active noise control," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Batumi, Georgia, Jun. 2018, pp. 1–5, doi: 10.1109/BlackSeaCom.2018.8433640.

[26] Y. Liu and C. Li, "Distributed prediction via adaptive Hammerstein filter over networked systems," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 3, pp. 534–548, Sep. 2018, doi: 10.1109/TSIPN.2017.2780620.

[27] S. Sitjongsataporn, S. Prongnuch, and T. Wiangtong, "Diffusion inverse square-root recursive least square-based algorithm with adaptive network," in *Proc. 18th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Chiang Mai, Thailand, May 2021, pp. 709–712, doi: 10.1109/ECTI-CON51831.2021.9454860.

[28] S. Scardapane, M. Scarpiniti, D. Comminiello, and A. Uncini, "Diffusion spline adaptive filtering," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 2016, pp. 1498–1502, doi: 10.1109/EUSIPCO.2016.7760498.

[29] J. A. Apolinario, Jr., R. G. Alves, P. S. R. Diniz, and M. N. Swamy, "Filtered gradient algorithms applied to a subband adaptive filter structure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 6, May 2001, pp. 3705–3708, doi: 10.1109/ICASSP.2001.940647.

[30] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Springer, 2008.

[31] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014, doi: 10.1109/JPROC.2014.2306253.

[32] S. A. Alghunaim and A. H. Sayed, "Distributed coupled multiagent stochastic optimization," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 175–190, Jan. 2020, doi: 10.1109/TAC.2019.2906495.

[33] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Nonlinear spline adaptive filtering," *Signal Process.*, vol. 93, no. 4, pp. 772–783, Apr. 2013, doi: 10.1016/j.sigpro.2012.09.021.

[34] S. Sitjongsataporn, "Widely-linear normalised orthogonal gradient adaptive-based algorithm for OFDM systems," in *Proc. 9th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, Hua Hin, Thailand, May 2012, pp. 1–4, doi: 10.1109/ECTICon.2012.6254161.

[35] S. Haykin, *Adaptive Filter Theory*. London, U.K.: Pearson, 2013.

[36] J. Jeraj and V. J. Mathews, "A stable adaptive Hammerstein filter employing partial orthogonalization of the input signals," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1412–1420, Apr. 2006, doi: 10.1109/TSP.2006.870643.

SUCHADA SITJONGSATAPORN received the B.Eng. and D.Eng. degrees (Hons.) in electronic engineering from the Mahanakorn University of Technology, Bangkok, Thailand, in 2002 and 2009, respectively. Since 2002, she has been working as a Lecturer at the Department of Electronic Engineering, Mahanakorn Institute of Innovation (MII), Mahanakorn University of Technology. since 2002. She is currently an Associate Professor in electronic engineering and the Assistant President of research at the Mahanakorn University of Technology. Her research interests include mathematical and statistical models in the area of adaptive signal processing for communications, networking, embedded systems, and image and video processing.