ADMM-Based Distributed Recursive Identification of Wiener Nonlinear Systems Using WSNs

Saurav Gupta*,1, Ajit Kumar Sahoo*,2, Upendra Kumar Sahoo*,3
*Department of Electronics and Communication Engineering, National Institute of Technology, Rourkela, India
Email: [1]greater.saurav@gmail.com, [2]ajitsahoo@nitrkl.ac.in, [3]sahooupen@nitrkl.ac.in

***Abstract*—The distributed estimation over wireless sensor networks (WSNs), as opposed to least-squares and fusion-center based estimations, is proficient to work with real-time applications. In this paper, a block-structured Wiener model is identified in a distributed fashion by minimizing the least-squares cost function on prediction error. As the block-structured Wiener model can approximate a large class of nonlinear systems with a small number of characteristics parameters hence makes it more suitable to work with. The global minimization task is reformed into several constrained subtasks in a manner that each node in WSN can obtain the parameters of interest locally. Each node in the network has the ability to combine its local estimates with the single-hop neighbors' estimates to obtain the global parameters of interest. The optimization of the reformulated cost is accomplished using a powerful distributed method called alternating direction method of multipliers. Simulations are carried on an infinite-order nonlinear system under the impact of observation noise. The obtained results are juxtaposed to the results of non-cooperative algorithm to show the effectiveness and superiority of the proposed algorithm.**

***Keywords*—WSN, nonlinear systems, Distributed, basis functions, ADMM, fusion-center.**

I. Introduction

Data-driven system modeling is an important research area for a variety of real-time applications. This approach firstly selects the pertinent model structure and then estimates the interested parameters using an identification method. The block-oriented nonlinear models have simple structures hence attracted many researchers in the field of system identification, system design and control, and prediction. Further, it requires very less parameter complexity to approximate a large class of nonlinear [1], [2]. The block-oriented nonlinear modeling is classified into two common model structures: 1) Wiener model where the cascade connection of a linear dynamical system is followed by a static nonlinear element, and 2) Hammerstein model where the linear and nonlinear blocks get reversed i.e. a static nonlinear element is followed by a linear dynamical system.

Previously in [3], we have proposed distributed Wiener modeling for distributed parameter systems where parameters are spatially and temporally coupled. There are many engineering applications where parameters are only temporally varying i.e. lumped parameter systems (LPSs). The objective of our proposed work is to identify the Wiener model for LPSs in a distributed recursive manner through wireless sensor network (WSN). A similar approach can be employed to identify the Hammerstein model in a distributed fashion. WSN composed of spatially dispersed battery operated sensors with communication and processing capabilities have gained importance because of their real-time applications including precision agriculture, transportation, habitat monitoring, network re-tasking, and health monitoring [4]–[7]. The online decentralized estimation using WSN is able to retrieve the desired parameters [8]. The distributed estimation using WSNs has gained importance in the field of modeling applications with reduced use of resources.

Some of the important literature on Wiener modeling are: Wiener systems are identified by Wigren in [9] using prediction error based recursive algorithm. Hagenblad et al. in his article [10] utilized maximum likelihood estimation for identifying Wiener model. Adrian Wills et al. in [11] addresses a new maximum likelihood-based method to identify Hammerstein-Wiener model where nonlinearity has been considered to be non-invertible. A renowned least-squares based and gradient-based Wiener model identification algorithms have been designed by Dongqing Wang et al. in [12]. Bai in [13] proposed a blind approach for the identification of Hammerstein-Wiener model. Sharareh Talaie in [14] developed an algorithm by employing Adaptive Weighted Particle Swarm Optimization (AWPSO) to identify nonlinear systems using Hammerstein-Wiener structure. Gupta et al. in [15] described least mean square approach for Wiener system identification. F. Ding et al. in [16] described a recursive least-squares identification algorithm for Wiener nonlinear systems where the unknown terms of the information vector are replaced by their preceding estimates to take care the difficulty in estimating unmeasured variables and unknown terms.

As of now, all the literature on Wiener model identification employs either LS-based method or adaptive way of optimization. The real-time modeling and control applications demand the modeling of nonlinear systems in an online adaptive manner with improved flexibility towards failure. These real-time demands are inefficiently fulfilled by employing LS and single agent based adaptive optimization because LS-based methods are offline in nature and single agent based adaptive methods are prone to noise and failure. The above-mentioned real-time demands can be fulfilled by employing WSN-based cooperative and distributed signal processing that improves the modeling in terms of performance and resilience towards failure [17]. The aim of this research is to propose an algorithm for the identification of Wiener model using distributed adaptive strategy. The job can also be fulfilled by using the centralized approach where data are processed online using fusion-center (FC). The FC-based approach suffers from the limitations of large energy consumption [5], computationally fast central processor requirement [8] and non-robustness due to the failure of the whole system if FC gets corrupted [6]. Distributed estimation over the network is introduced with an intention to rescue from these limitations.

In distributed estimation, the nodes in the network locally estimate the parameter vector and then with the means of cooperation between nodes, all the nodes tend towards global estimate. The ADMM-based consensus strategy is used as an optimization method where the least-square cost function of Wiener model is decomposed into several subtasks. These subtasks are individually allocated to each node, finding their desired estimates parallelly in a cooperative way [18], [19]. The competence to estimate the parameters of nonlinear Wiener model is presented in the simulation result section.

Notation: The boldface letters and bar-head letters are used to denote the matrix and vector quantities respectively. Scalar quantities are represented by simple letters. Moreover, $(\cdot)^T$ and $\|\cdot\|$ denote the transpose and the Frobenius norm of any quantity respectively. Other used notations are defined wherever needed.

II. Traditional Wiener Modeling

Wiener nonlinear model was firstly studied by N. Wiener in 1958 to significantly model sensor nonlinearities. Wiener nonlinear system consists of a series blocks of LTI subsystem followed by memoryless nonlinearity function as shown in Fig. 1. The input $b(t)$ is first passed through LTI subsystem having

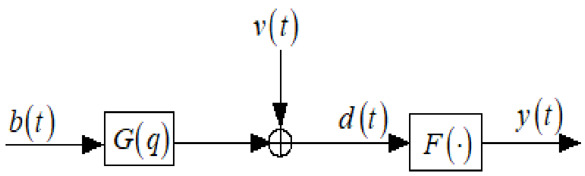

Fig. 1: Wiener nonlinear system [20]

transfer function $G(q)$ then the outcome is passed through static nonlinear function $F(\cdot) : \mathbb{R}$ to get the corresponding output $y(t)$ as

$$y(t) = F(d(t)) = F(G(q)\,b(t) + v(t)) . \tag{1}$$

$G(q)$ is assumed to be expanded into the orthonormal basis functions as

$$G(q) = \sum_{i=1}^{n_\alpha} \alpha_i g_i(q), \tag{2}$$

where $g_i(q)$ $(i = 1, ..., n_\alpha)$ can take any of the known orthonormal basis functions and the parameters $\alpha_i \in \mathbb{R}$ $(i = 1, .., n_\alpha)$ has to be estimated. Finite impulse response, Laguerre and Kautz functions etc. can be used as known basis functions $g_i(q)$ $(i = 1, ..., n_\alpha)$.

The Wiener system intermediate variable $d(t)$ can be expressed using Fig. 1 and Eq. (2) as

$$d(t) = G(q)\,b(t) + v(t) = \sum_{i=1}^{n_\alpha} \alpha_i g_i(q) b(t) + v(t) . \tag{3}$$

Let us assume the nonlinear function used is invertible. The inverse of nonlinear function $F(\cdot)$ can be expanded as the nonlinear basis functions in the following form as

$$d(t) = F^{-1}(y(t)) = \sum_{j=1}^{n_\beta} \beta_j f_j(y(t)), \tag{4}$$

where the unknown parameters $\beta_j \in \mathbb{R}\,(j = 1, ..., n_\beta)$ need to be estimated. The basis functions $f_j(\cdot) : \mathbb{R} \rightarrow \mathbb{R}\,(j = 1, ...n_\beta)$ to capture nonlinear dynamics may be any of the following: such as simple polynomials, radial basis functions, splines basis functions etc. The polynomial representation is commonly used because of its simple implementation and easy to analyze. The orders n_α and n_β are supposed to be known formerly. The estimation of the unknown parameters α_i $(i = 1, ...n_\alpha)$ and β_j $(j = 1, ...n_\beta)$ using the data set $\{b(t), y(t)\}_{t=1}^{L}$ is the primal task of Wiener modeling.

With the use of general assumption i.e. $\beta_1 = 1$, $f_1(y(t)) = y(t)$ then equalizing Eq. (3) and Eq. (4) leads to

$$y(t) = \sum_{i=1}^{n_\alpha} \alpha_i g_i(q) b(t) - \sum_{j=2}^{n_\beta} \beta_j f_j(y(t)) + v(t) . \tag{5}$$

The expression (5) can be specifically rewritten in the linear regression form as

$$y(t) = \bar{\zeta}^T \bar{\varkappa}(t) + v(t) , \tag{6}$$

where

$$\bar{\zeta} = \left[\alpha_1, ...\alpha_{n_\alpha}, \beta_2, ...\beta_{n_\beta}\right]^T \in \mathbb{R}^{(n_\alpha + n_\beta - 1)}, \tag{7}$$

$$\bar{\varkappa}(t) = \begin{bmatrix} (g_1(q)\,b(t))^T, ..., (g_{n_\alpha}(q)\,b(t))^T, \\ -f_2^T(y(t)), ..., -f_{n_\beta}^T(y(t)) \end{bmatrix}^T \in \mathbb{R}^{n_\alpha + n_\beta - 1}. \tag{8}$$

The main objective is to estimate the parameter vector $\bar{\zeta}$ by minimizing the quadratic cost function of prediction error in a distributed manner using WSN. The estimate $\hat{\bar{\zeta}}$ can be obtained by minimizing the quadratic cost function of prediction error

$$\hat{\bar{\zeta}} = \arg\min_{\bar{\zeta}} \left\{ \frac{1}{L} \sum_{t=1}^{L} \left\| y(t) - \bar{\zeta}^T \bar{\varkappa}(t) \right\|^2 \right\}. \tag{9}$$

III. Distributed Wiener Model Identification

A WSN with a N number of sensing units (nodes) is considered for in-network data processing. These sensing units are dispersed into an area under monitoring to get the measured output for any input. At each node, the local estimates can be obtained using these measurements. Any node j can only communicate with its immediate neighbors having cardinality $|\mathcal{N}_j|$ i.e. single-hop communications are allowed for exchanging information. Inter-sensor links are assumed to be symmetric hence the WSN is designed to be the undirected connected graph. For any input $\{ b_j(\tau) | \tau = (t-Q+1),..,t \}$ at any node j, the corresponding measured output is $y_j(t)$. Here Q is the fading memory of the nonlinear system under monitor. The global cost function of any WSN for the linear regression problem (9) can be obtained by stacking all the nodes' measured output into a global vector as $\bar{\Gamma}(t) = [y_1(t), ..., y_N(t)]^T \in \mathbb{R}^{N\times 1}$ and their corresponding regressor vectors are stacked to form a global matrix $\boldsymbol{U}(t) = [\bar{\varkappa}_1(t), ..., \bar{\varkappa}_N(t)] \in \mathbb{R}^{(n_\alpha+n_\beta-1)\times N}$ and then obtain the estimate vector $\bar{\zeta}$ of dimension $(n_\alpha + n_\beta - 1)\times 1$ by the minimization of

$$\begin{aligned} \hat{\bar{\zeta}}(t) &= \arg\min_{\bar{\zeta}} E\left\| \bar{\Gamma}(t) - \boldsymbol{U}^T(t)\bar{\zeta} \right\|^2 \\ &= \arg\min_{\bar{\zeta}} \sum_{j=1}^{N} E\left[\left(y_j(t) - \bar{\varkappa}_j^T(t)\bar{\zeta} \right)^2 \right], \end{aligned} \tag{10}$$

here symbol E represents the expectation operator. The optimal parameters of the Wiener model that are stacked as the entries of $\bar{\zeta}$, can be estimated by the minimization of global cost function (10). For minimizing the cost in (10) w.r.t. $\bar{\zeta}$, a fusion-center is required to store and process all the data from each node. This centralized way of data processing requires a powerful central processing unit (CPU) and the enormous amount of energy as a communication resource. In order to rescue from these drawbacks, in-network data processing based distributed algorithm is developed that allows to recursively estimate the desired parameters of Wiener model.

In order to access the in-network processing, each node is allocated with its individual task of calculating their intermediate estimates also called local estimates. From (10), it is unlikely to say how and which node will update the parameter vector $\bar{\zeta}$ as it is spatially non-separable. To solve this uncertainty, auxiliary variables $\{\bar{\zeta}_j\}_{j=1}^N$ at each node has been introduced to take care of updating vector $\bar{\zeta}$. Now the global cost function in (10) can be well expressed in the equality constraint form as

$$\begin{aligned} \left\{ \hat{\bar{\zeta}}_j(t) \right\}_{j=1}^N &= \arg\min_{\{\bar{\zeta}_j\}_{j=1}^N} \sum_{m=0}^{t}\sum_{j=1}^{N} \lambda^{t-m}\left[y_j(m) - \bar{\varkappa}_j^T(m)\bar{\zeta}_j \right]^2 \\ &\quad + N^{-1}\lambda^t \sum_{j=1}^{N} \bar{\zeta}_j^T \boldsymbol{\Psi}_0 \bar{\zeta}_j, \\ &\quad \text{s.t. } \bar{\zeta}_j = \bar{\zeta}_{j'},\ j \in [1,..,N],\ j' \in \mathcal{N}_j, \end{aligned} \tag{11}$$

where λ is the forgetting factor which is used to practically compromise between misadjustment and tracking [21], $\boldsymbol{\Psi}_0$ is the positive definite matrix used for regularization and $\mathcal{N}_j$ is the neighbors of any node j. Eq. (10) and (11) are equivalent in a sense that $\{\bar{\zeta}_j = \bar{\zeta}\}_{j=1}^N$ is followed. Since WSN is connected, Eq. (10) and (11) are equivalent because $\{\bar{\zeta}_j = \bar{\zeta}\}_{j=1}^N$.

Now since each sensor node has been alloted with its individual task, ADMM optimization technique can be employed in (11) to get the estimates of $\bar{\zeta}$ in a decentralized and cooperative manner. The auxiliary variables $\left\{\bar{\zeta}_{j'}^j\right\}$ for $j' \in \mathcal{N}_j$ are also need to be considered with equality consensus constraints equivalent to the constraints specified in (11) as

$$\bar{\zeta}_j = \bar{s}_j^{j'}, \bar{\zeta}_j = \bar{s}_{j'}^{j}, for\ j \in [1,N],\ j' \in \mathcal{N}_j,\ j \neq j'. \tag{12}$$

ADMM algorithm is well suited for decomposition of optimization problem into separable constrained minimization subtasks [19]. To facilitate the ADMM optimization, rewriting (11) with consensus constraints specified in (12) as a quadratically augmented Lagrangian decomposable form

$$\begin{aligned} &L_a(\bar{\zeta}, \bar{z}, \bar{h}, \bar{w}) = \\ &\sum_{j=1}^{N}\sum_{m=0}^{t} \lambda^{t-m}[y_j(m) - \bar{\varkappa}_j^T(m)\bar{\zeta}_j]^2 + \frac{\lambda^t}{P}\sum_{j=1}^{N} \bar{\zeta}_j^T \boldsymbol{\Psi}_0 \bar{\zeta}_j \\ &+ \sum_{j=1}^{N}\sum_{j'\in\mathcal{N}_j} \left[(\bar{h}_j^{j'})^T (\bar{\zeta}_j - \bar{s}_j^{j'}) + (\bar{w}_j^{j'})^T (\bar{\zeta}_j - \bar{s}_{j'}^{j}) \right] \\ &+ \frac{c}{2}\sum_{j=1}^{N}\sum_{j'\in\mathcal{N}_j} \left[\left\| \bar{\zeta}_j - \bar{s}_j^{j'} \right\|^2 + \left\| \bar{\zeta}_j - \bar{s}_{j'}^{j} \right\|^2 \right], \end{aligned} \tag{13}$$

where $\bar{h}$ and $\bar{w}$ are the associated Lagrangian multipliers, c represents positive penalty coefficient, $\bar{\zeta} = \{\bar{\zeta}_j\}_{j=1}^N$, $\bar{z} = \left\{\bar{s}_j^{j'}\right\}_{j\in[1,..,N]}^{j'\in\mathcal{N}_j}$ and $[\bar{h}, \bar{w}] = \left\{\bar{h}_j^{j'}, \bar{w}_j^{j'}\right\}_{j\in[1,..,N]}^{j'\in\mathcal{N}_j}$. At time instant $t+1$ and ADMM iteration k, the first step towards the estimation of optimum parameter vector $\bar{\zeta}$ using ADMM is to get the local updates of Lagrangian multipliers. The updates of Lagrangian multipliers can be obtained using gradient ascent optimization method as

$$\begin{aligned} \bar{h}_j^{j'}(t+1;k) &= \bar{h}_j^{j'}(t+1;k-1) \\ &\quad + c\left[\bar{\zeta}_j(t+1;k) - \bar{s}_j^{j'}(t+1;k) \right] \end{aligned} \tag{14}$$

$$\begin{aligned} \bar{w}_j^{j'}(t+1;k) &= \bar{w}_j^{j'}(t+1;k-1) \\ &\quad + c\left[\bar{\zeta}_j(t+1;k) - \bar{s}_{j'}^{j}(t+1;k) \right]. \end{aligned} \tag{15}$$

In the second step of ADMM the minimization of (13) is obtained w.r.t. $\{\bar{\zeta}_j\}_{j=1}^N$ with the assumption that all other variables are kept fixed to their recent values. Then $\bar{\zeta}_j(t+1;k+1)$ can be recursively updated as

$$\begin{aligned} &\bar{\zeta}_j(t+1;k+1) = \\ &\left[2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\varkappa}_j(m)\bar{\varkappa}_j^T(m) + \frac{\lambda^{t+1}}{N}\boldsymbol{\Psi}_0 \right) + 2c\left|\mathcal{N}_j\right| I \right]^{-1} \\ &\times \left[\begin{aligned} &2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\varkappa}_j(m) y_j(m) \right) \\ &-\sum_{j'\in\mathcal{N}_j}\left(\bar{h}_j^{j'}(t+1,k) + \bar{w}_j^{j'}(t+1,k) \right) \\ &+c\sum_{j'\in\mathcal{N}_j}\left(\bar{s}_j^{j'}(t+1;k) + \bar{s}_{j'}^{j}(t+1;k) \right) \end{aligned} \right]. \end{aligned} \tag{16}$$

In the third step, (13) is minimized w.r.t. $\bar{s}_j^{j'}$ while keeping other variables to their recent updates. This leaves out the expression

$$\bar{s}_j^{j'}(t+1;k+1) = 0.5\left[\bar{\zeta}_j(t+1;k+1) + \bar{\zeta}_{j'}(t+1;k+1)\right] \\ +0.5c^{-1}\left[\bar{h}_j^{j'}(t+1;k) + \bar{w}_{j'}^{j}(t+1;k)\right]. \tag{17}$$

Let the initialization of Lagrange multipliers is taken as $\bar{h}_j^{j'}(t+1;0) = -\bar{w}_{j'}^{j}(t+1;0)$ then $\bar{h}_j^{j'}(t+1;k) = -\bar{w}_{j'}^{j}(t+1;k)$, $\forall\, t,\, k$ is followed. Now with the usage of (17) into (14) and (15) along with the property $\bar{h}_j^{j'}(t+1;k) = -\bar{w}_{j'}^{j}(t+1;k)$, $\forall\, t,\, k$, the Lagrangian multiplier can be written in the following updated form as

$$\bar{h}_j^{j'}(t+1;k) = \bar{h}_j^{j'}(t+1;k-1) \\ + \frac{c}{2}\left[\bar{\zeta}_j(t+1;k) - \bar{\zeta}_{j'}(t+1;k)\right]. \tag{18}$$

Now, the substitution of Eq. (17) into Eq. (16) along with the above mentioned followed property $\bar{h}_j^{j'}(t+1;k) = -\bar{w}_{j'}^{j}(t+1;k)$, $\forall\, t,\, k$, Eq. (16) can be rewritten as

$$\bar{\zeta}_j(t+1;k+1) = \\ \left[2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\varkappa}_j(m)\bar{\varkappa}_j^T(m) + \frac{\lambda^{t+1}}{N}\mathbf{\Psi}_0\right) + 2c\left|\mathcal{N}_j\right| I\right]^{-1} \\ \times \begin{bmatrix} 2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\varkappa}_j(m)y_j(m)\right) \\ -\sum_{j'\in\mathcal{N}_j}\left(\bar{h}_j^{j'}(t+1,k) - \bar{h}_{j'}^{j}(t+1,k)\right) \\ +c\sum_{j'\in\mathcal{N}_j}\left(\left[\bar{\zeta}_j(t+1;k) + \bar{\zeta}_{j'}(t+1;k)\right]\right) \end{bmatrix}. \tag{19}$$

At any time instant $t+1$, Eq. (18) and Eq. (19) are repeated for K number of ADMM iterations to estimate the desired parameter vector $\bar{\zeta}$ at each node. For tracking of time-varying dynamical systems, one ADMM iteration is used per time instant i.e. ADMM iteration coincides with the time iteration i.e. $k = t$.

A. Communication and Computational Complexity

Since the nodes in the network are transmitting as well as receiving the information, hence there will be transmitting cost and receiving cost for each sensor node. At any time iteration, each node in the network has a transmission cost of $\{(n_\alpha + n_\beta - 1)(|\mathcal{N}_j| + 1)\}$ scalars which include transmission of Lagrange multipliers $\left\{\bar{h}_j^{j'}\right\}_{j'\in\mathcal{N}_j}$ and local estimate $\bar{\zeta}_j$. Similarly, at any time iteration, each node has a reception cost of $\{2|\mathcal{N}_j|(n_\alpha + n_\beta - 1)\}$ scalars that includes the reception of the Lagrange multipliers $\left\{\bar{h}_{j'}^{j}\right\}_{j'\in\mathcal{N}_j}$ and local estimates from the neighboring nodes $\left\{\bar{\zeta}_{j'}\right\}_{j'\in\mathcal{N}_j}$.

At any sensor node j, $O\left(|\mathcal{N}_j|(n_\alpha + n_\beta - 1)\right)$ computations per iteration are required to update Lagrange multipliers using (18). With the usage of matrix inversion lemma, the inversion term involved in updating the parameter matrix $\bar{\zeta}_j$ in (19) requires $O\left((n_\alpha + n_\beta - 1)^2\right)$ computations when $\lambda = 1$. For $\lambda < 1$, $O\left((n_\alpha + n_\beta - 1)^3\right)$ computations per iteration will be required for the inverting term.

B. Convergence Analysis

Proposition 1: *For any time t with positive penalty coefficient $c > 0$ and initializing the values of $\bar{h}_j^{j'}(t;0)$ and $\bar{\zeta}_j(t;0)$ with an arbitrary vector, the local recursive estimate (19) at any node j will satisfy the consensus constraints as $k \to \infty$:*

$$\lim_{k\to\infty}\bar{\zeta}_j(t,k) = \lim_{k\to\infty}\bar{\zeta}_{j'}(t,k) = \bar{\zeta}_{centz}(t)\ \forall\, j \in [1, N], \tag{20}$$

where $\bar{\zeta}_{centz}(t)$ represents the fusion-center based optimal solution (centralized solution) i.e. when measured data of all the nodes along with their corresponding regressors are collected and processed using fusion-center.

Proof: The claim in *Proposition 1* can be proved by going through the same steps as involved in proving the *Proposition 2* of article [22]. The first step is to reformulate the cost function (10) in a similar form as [23, Eq. 4.76]. Now the claim argued in *Proposition 1* can be proven by mimicking the steps involved in ADMM [23, pg. 255].

IV. Simulation Results

A nonlinear Wiener system with infinite-order and finite fading memory ($Q = 10$) as shown in Fig. 1 is taken into consideration. The intermediate variable and output of the system is taken as

$$d(t) = 0.9b(t) + (0.9)^2 b(t-1) + ... \\ + (0.9)^M b(t-(Q-1)) \\ y(t) = \frac{1}{1+e^{-d(t)}}. \tag{21}$$

The approximated block-structured Wiener model consists of invertible 2^{nd}-order polynomial nonlinearity i.e. ($n_\beta = 2$) is used to model the above nonlinear system. The LTI subsystem order n_α should be equal to the number of terms in $d(t)$ i.e. $n_\alpha = Q = 10$. Many types of basis functions other than polynomial functions can also be utilized in order to represent the system nonlinearity. But polynomial functions are simple and easy to analyze, also their implementation is quite easier hence are employed here to represent nonlinear basis functions.

The Wiener model parameters for the above-mentioned system are obtained by applying the proposed algorithm in an eight-node WSN shown in Fig. 2. A Gaussian sequence with $\mathcal{N}(0,1)$ is used as an input to the system and the corresponding output is generated using (21). The observational noise $v_o(t)$ is considered to be i.i.d (independent and identically distributed) with zero mean and noise variances of each node is depicted in Fig. 3. The forgetting factor $\lambda = 1$ and the penalty coefficient $c = 1$ are used to estimate the parameter vector $\bar{\zeta}$. At any time t, the mean-square error (MSE) and

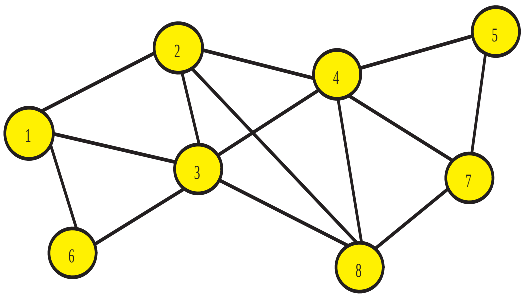

Fig. 2: WSN with eight-node

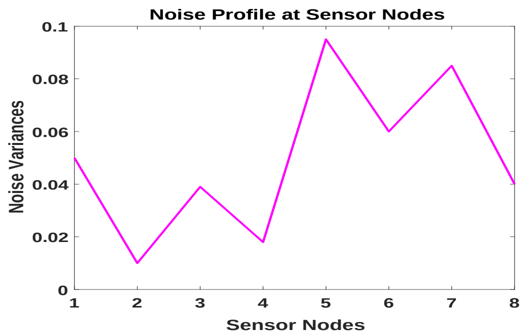

Fig. 3: Noise Variance at different nodes

excess mean-square error (EMSE) are used as the performance metrics that can be defined at any node j as

$$MSE_j(t) = E\left|(y_j(t) + v_o(t)) - \hat{\bar{\zeta}}_j^T(t)\,\bar{\varkappa}_j(t)\right|^2$$

$$EMSE_j(t) = E\left|(y_j(t)) - \hat{\bar{\zeta}}_j^T(t)\,\bar{\varkappa}_j(t)\right|^2.$$

If the average of all the nodes performance is taken, the network performance is defined as

$$MSE^{network} \triangleq \frac{1}{P}\sum_{j=1}^{N} MSE_j$$

$$EMSE^{network} \triangleq \frac{1}{P}\sum_{j=1}^{N} EMSE_j.$$

The average of 100 experiment runs is taken over the time period of 450 iterations. The obtained metrics curves in Fig. 4 and Fig. 5 show that the performance in terms of convergence rate improves if the ADMM iteration number K is increased and can come close to the centralized solution if ADMM iterations K reach to a certain number. The number of ADMM iterations required to reach close to the centralized solution depends on the network anatomy as well as on the number of nodes in the network.

It can also be seen from plots that the non-cooperative way of data processing has poor convergence rate compared to standard distributed Wiener (std. D-Wiener) model ($k = t$ or $K = 1$). The penalty coefficient c used here is chosen using

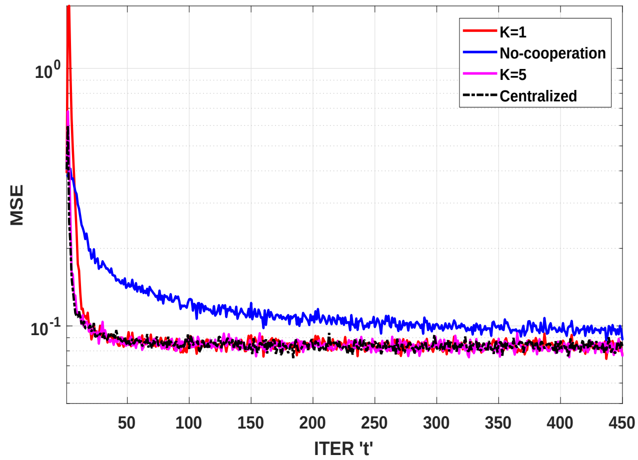

Fig. 4: Overall network MSE with increasing values of K for nonlinear system (21). Comparison with non-cooperation and centralized algorithms.

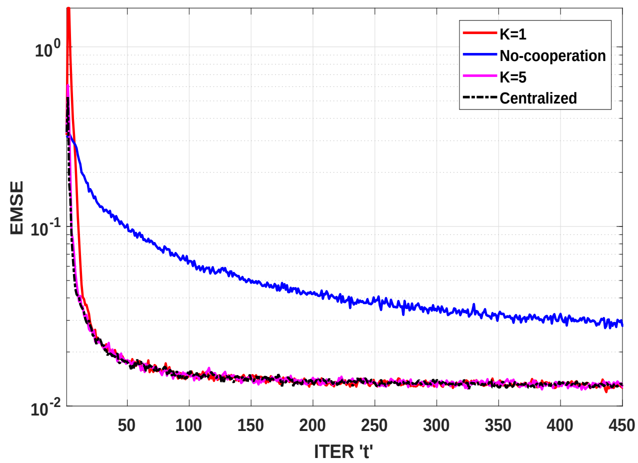

Fig. 5: Overall network EMSE with increasing values of K for nonlinear system (21). Comparison with non-cooperation and centralized algorithms.

hit and trial manner to obtain the optimum results. There can be many other ways of choosing c, which is beyond the scope of this paper.

V. Conclusion and Future Scope

This work proposed a distributed algorithm to identify Wiener model using WSNs. The designed algorithm can be implemented for real-time applications. The incorporation of cooperative and distributed advantages allows the performance to be more responsive and robust in real-time. The MATLAB-based simulation results for the identification of Wiener model are obtained under a noisy environment. The obtained results are compared to the non-cooperative simulation results showing the supremacy of the deduced framework. In the future, rather than hit and trial method to find the value of penalty coefficient c, an adaptive method can be incorporated. Also, the algorithm can be made computationally and communicatively more energy efficient. In this article, we have assumed the nonlinearity to be invertible which may not be followed by many practical systems, hence can be considered in the future article.

REFERENCES

[1] J. Vörös, "Parameter identification of wiener systems with multisegment piecewise-linear nonlinearities," *Systems & Control Letters*, vol. 56, no. 2, pp. 99–105, 2007.

[2] H.-X. Li and C. Qi, *Spatio-temporal modeling of nonlinear distributed parameter systems: A time/space separation based approach*. Springer Science & Business Media, 2011, vol. 50.

[3] S. Gupta, A. K. Sahoo, and U. K. Sahoo, "Nonlinear space–time varying parameter estimation using consensus-based in-network distributed strategy," *Digital Signal Processing*, vol. 79, pp. 175–189, 2018.

[4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[5] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 4. IEEE, 2001, pp. 2033–2036.

[6] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[7] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. Acm, 2002, pp. 88–97.

[8] I. D. Schizas and A. Aduroja, "A distributed framework for dimensionality reduction and denoising," *IEEE Transactions on Signal Processing*, vol. 63, no. 23, pp. 6379–6394, 2015.

[9] T. Wigren, "Recursive prediction error identification using the nonlinear wiener model," *Automatica*, vol. 29, no. 4, pp. 1011–1025, 1993.

[10] A. Hagenblad and L. Ljung, "Maximum likelihood estimation of wiener models," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 3. IEEE, 2000, pp. 2417–2418.

[11] A. Wills, T. B. Schön, L. Ljung, and B. Ninness, "Identification of hammerstein–wiener models," *Automatica*, vol. 49, no. 1, pp. 70–81, 2013.

[12] D. Wang and F. Ding, "Least squares based and gradient based iterative identification for wiener nonlinear systems," *Signal Processing*, vol. 91, no. 5, pp. 1182–1189, 2011.

[13] E.-W. Bai, "A blind approach to the hammerstein–wiener model identification," *Automatica*, vol. 38, no. 6, pp. 967–979, 2002.

[14] S. Talaie, M. A. Shoorehdeli, and L. Shahmohamadi, "Nonlinear system identification of hammerstein-wiener model using awpso," in *Intelligent Systems (ICIS), 2014 Iranian Conference on*. IEEE, 2014, pp. 1–4.

[15] S. Gupta, A. K. Sahoo, and U. K. Sahoo, "Parameter estimation of wiener nonlinear model using least mean square (lms) algorithm," in *Region 10 Conference, TENCON 2017-2017 IEEE*. IEEE, 2017, pp. 1399–1403.

[16] F. Ding, X. Liu, and M. Liu, "The recursive least squares identification algorithm for a class of wiener nonlinear systems," *Journal of the Franklin Institute*, vol. 353, no. 7, pp. 1518–1526, 2016.

[17] A. H. Sayed *et al.*, "Adaptation, learning, and optimization over networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[18] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, 2009.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[20] A. Hagenblad, *Aspects of the identification of Wiener models*. Division of Automatic Control, Department of Electrical Engineering, Linköpings universitet, 1999.

[21] C. Paleologu, J. Benesty, and S. Ciochină, "A practical variable forgetting factor recursive least-squares algorithm," in *Electronics and Telecommunications (ISETC), 2014 11th International Symposium on*. IEEE, 2014, pp. 1–4.

[22] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links-part I: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.

[23] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.