In-Network Distributed Identification of Wiener and Volterra-Laguerre Models for Nonlinear Systems

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*—Distributed estimation over wireless sensor networks (WSNs) has been used to obtain the parameters of interest with reduced resource consumption, hence gained importance in system modeling and control applications. Unlike least-squares and fusion-center based approaches, distributed signal processing is competent in real-time applications. In this article, Volterra-Laguerre model and Wiener model are identified in a distributed manner through WSNs for modeling of nonlinear systems. A block-structured Wiener model has been widely used as it is characterized by a small number of parameters, but can only model specific nonlinearities. A generalized Volterra model over Wiener model can approximate any nonlinear system to a desired precision but has increased parameter complexity. By expanding nonlinear Volterra kernels with orthogonal Laguerre functions, the parameter complexity is reduced significantly. A distributed recursive algorithm for the identification of above-mentioned nonlinear models is designed by minimizing the quadratic prediction error. The algorithm reformulates model identification framework into multiple constrained separable subtasks. These subtasks are optimized using a powerful method called alternating direction method of multipliers. Simulation results for an infinite-order and a 2^{nd}-order nonlinear systems are obtained under the influence of process noise and are compared with the results of non-cooperative estimation showing the superiority of the proposed algorithm.**

***Index Terms*—Distributed signal processing, Wiener, Volterra, nonlinear systems, ADMM, fusion-center.**

I. Introduction

Data-based system modeling is a key issue for a wide range of real-time engineering applications such as pH-neutralization process, two tank system control and continuous stirred tank reactor etc. Such an approach first selects the appropriate model structure and then estimate the parameter of interest using some identification methodology. Generally, all the real-time systems have nonlinear nature hence nonlinear modeling is preferred for model-based control, analysis, and design. The nonlinear Volterra model is one that can model any fading memory nonlinear systems with arbitrary accuracy [1]. This model consists of a series of Volterra kernels that are the higher order approximation of the system's impulse response [2]. It has significant modeling capability and simple nonlinear structure [3] but may lead to cumbersome modeling because of its high parameter complexity while modeling highly nonlinear systems. To overcome the parameter complexity, there are two main approaches involved. First, the block-structured models are used for modeling high-order nonlinear systems. Block-oriented Wiener model got much attention in the field of system identification, system control, system design and prediction because of its simple structure and can approximate a wide range of nonlinear systems [4], [5]. Wiener model is a cascade connection of a linear time-invariant (LTI) system followed by a static nonlinear element [5]. The Wiener model allows to take only specific nonlinearities and also its output is nonlinear w.r.t. the parameters [6]. The second approach in reducing the parameter complexity involves the employment of orthogonal Laguerre basis functions to approximate the Volterra kernels. The Laguerre functions are very efficient to approximate the kernels of practical nonlinear systems [7]. The objective of our proposed work is to implement both the above approaches in a distributed recursive manner through ad-hoc wireless sensor network (WSN). The online decentralized estimation using WSN has been proved to retrieve the desired parameters of interest [8].

A. *Previous Works*

Many algorithms in the literature employ least-square (LS), least mean square (LMS) and recursive least square (RLS) to identify Volterra and Wiener models. Wigren in [9] derived a prediction error based recursive identification algorithm using Wiener model. In [10], Hagenblad et al. have identified Wiener model using maximum likelihood estimation. A renowned LS-based and gradient-based Wiener model identification algorithms have been designed by D. Wang et al. in [11]. F. Ding et al. in [12] described RLS algorithm to identify Wiener nonlinear systems where the difficulty of estimating unmeasured variables and unknown terms in the information vector have been taken care.

Literatures of Volterra modeling include: T. Koh et al. in [13] presented a factorization method to iteratively obtain Volterra kernels. In [14], Volterra filter identification for Gaussian inputs is carried out using LS optimization. J. Li et al. in [15] presented an LS identification of Volterra predistorter to compensate nonlinear effects in OFDM transmitters. The article [16] presented a fast RLS algorithm for 2^{nd}-order Volterra filtering. Authors in article [17] presented LMS and RLS based quadratic-Volterra filters for nonlinear speech coding. Article [18] studied an application of adaptive Volterra filter for the identification of parametric loudspeaker system. Z. Sirgit et al. presented an LMS based Volterra modeling in [19].

B. *Motivation and Overview of the Proposed Article*

All the articles in the literature survey resort to either LS or adaptive algorithms for the estimation of Volterra and Wiener model parameters. The LS based approach is not suitable for real-time modeling and control applications due to its offline nature of data processing. Performance improvement and better resilient towards failure as the benefits of co-operation [20] are incorporated with the use of WSN-based distributed nonlinear system modeling. The main contribution of this research pertains to design and analyze a novel algorithm to identify the parameters of Wiener and Volterra-Laguerre model in a distributed recursive way. The Centralized approach that involves online processing of data through fusion-center (FC) can also estimate the models parameters but has following limitations: 1) It requires a large amount of energy due to multi-hop communication [21], 2) It needs powerful central processing unit (CPU) with fast computational capability to process the data in real-time [8], 3) Lacks robustness due to failure of the whole system if FC gets damaged [22] etc. Distributed strategies have been introduced with an intention to overcome the limitations of LS and FC based approaches. As per our knowledge, there is no state of art algorithm available in the literature where the block-oriented model and Volterra model have been identified in a distributed manner.

This article estimates the parameters of the above-mentioned models in a distributed manner using an ad-hoc WSN. Each node in the network obtains the local estimate and then cooperates with its neighboring nodes to reach the global estimate. The least-squares cost function is decomposed into constrained minimization subtasks that are solved using simple yet powerful distributed optimization method called alternating direction method of multipliers (ADMM) [8], [23]. Simulation results are drawn in Section V to show the superiority of the designed algorithm over the non-cooperative method of model estimation.

C. Notations

The notations used in this article are: any alphanumeric with bar-head is termed to be a vector quantity, any alphanumeric with bold-case represents a matrix, and scalar quantity is represented by a simple alphanumeric term. $(\cdot)^T$ denotes the transpose of any quantity and $\|\cdot\|^2$ denotes the Frobenius norm. Other notations are defined wherever they are used.

II. Formulation of Wiener model identification

Consider a Wiener nonlinear system that has cascaded blocks of a LTI subsystem with $G(q)$ as the transfer function followed by a static nonlinear function $F(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ as shown in Fig. 1. Here $a(t)$ and $\gamma(t)$ are the input and corresponding

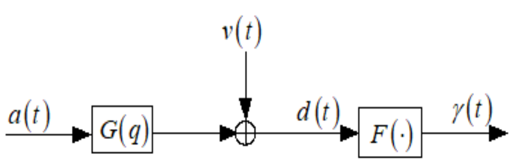


Fig. 1: Wiener nonlinear system [24]

measured output respectively, obtained for a time period of length L. $v(t)$ denotes the process noise and q represents the forward shift operator. The transfer function $G(q)$ of the linear dynamical subsystem can be assumed as the expansion of orthonormal basis functions as

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

where $\alpha_i \in \mathbb{R}\,(i = 1, 2, ..n_\alpha)$ are the parameters to be estimated and $g_i(q)$ $(i = 1, ..., n_\alpha)$ are the known basis functions. These known basis functions can take any of the generalized basis functions such as finite impulse response, Laguerre and Kautz functions etc. Referring to Fig. 1 and Eq. (1), the intermediate variable $d(t)$ can be expressed as

$$d(t) = G(q)\,a(t) + v(t) = \sum_{i=1}^{n_\alpha} \alpha_i g_i(q)a(t) + v(t). \tag{2}$$

Assuming nonlinearity is invertible, the inverse of $F(\cdot)$ can be approximated as the expansion of nonlinear basis functions as

$$d(t) = F^{-1}(\gamma(t)) = \sum_{j=1}^{n_\beta} \beta_j f_j(\gamma(t)), \tag{3}$$

where $\beta_j \in \mathbb{R}\,(j = 1, ...n_\beta)$ are the unknown parameters associated to the nonlinear basis functions $f_j(\cdot) : \mathbb{R} \rightarrow \mathbb{R}\,(j = 1, ...n_\beta)$. The nonlinear basis functions can be simple polynomials, radial basis functions, splines basis functions, wavelets, etc. The polynomial representation is commonly used as its implementation is simple and can be easily analyzed. The orders of n_α and n_β are assumed to be known beforehand. We apply the general assumption that $\beta_1 = 1$, $f_1(\gamma(t)) = \gamma(t)$ then equating Eq. (2) and Eq. (3) will lead to

$$\gamma(t) = \sum_{i=1}^{n_\alpha} \alpha_i g_i(q)a(t) - \sum_{j=2}^{n_\beta} \beta_j f_j(\gamma(t)) + v(t). \tag{4}$$

The above expression is in the form of linear regression and further can be rewritten as

$$\gamma(t) = \bar{s}^T \bar{\zeta}(t) + v(t), \tag{5}$$

where

$$\bar{s} = \left[\alpha_1, ...\alpha_{n_\alpha}, \beta_2, ...\beta_{n_\beta}\right]^T \in \mathbb{R}^{(n_\alpha + n_\beta - 1)}, \tag{6}$$

$$\bar{\zeta}(t) = \begin{bmatrix} (g_1(q)\,a(t))^T, ..., (g_{n_\alpha}(q)\,a(t))^T, \\ -f_2^T(\gamma(t)), ..., -f_{n_\beta}^T(\gamma(t)) \end{bmatrix}^T \in \mathbb{R}^{n_\alpha + n_\beta - 1}. \tag{7}$$

The quadratic cost function of prediction error can be given as

$$\bar{\hat{s}} = \arg\min_{\bar{s}} \left\{ \frac{1}{L} \sum_{t=1}^{L} \left\| \gamma(t) - \bar{s}^T \bar{\zeta}(t) \right\|^2 \right\}. \tag{8}$$

III. Formulation of Volterra-Laguerre Model identification

Consider a causal nonlinear system with fading memory characteristics

$$\gamma(t) = F_d\{a(\tau)\} + v(t), \tag{9}$$

where F_d is the nonlinear operator with fading memory (say M), $\{a(\tau)|\,\tau = (t - M + 1), .., t\}$ is the input, $\gamma(t)$ is the instantaneous output and $v(t)$ is the process noise of the system. The nonlinear system in (9) can be approximated by using finite order (say R) discrete-time Volterra model [1] as

$$\gamma(t) = \sum_{n=1}^{R} \sum_{\tau_1=0}^{M-1} \cdots \sum_{\tau_n=0}^{M-1} h_n(\tau_1, \ldots, \tau_n) \prod_{i=1}^{n} a(t - \tau_i) + v(t), \tag{10}$$

where h_n is the n^{th}-order temporal Volterra kernel.

Traditional Volterra model is not suitable for model-based control applications as it suffers from ill-conditioned estimation due to high parameter complexity. Hence it is desirable to reduce the number of parameters to be estimated while retaining the adequate approximation of Volterra kernels. Orthogonal Laguerre functions are used to approximate the nonlinear Volterra kernels to reduce parameter complexity [7]. The n^{th}-order nonlinear Volterra kernel can be approximated using r-dimensional Laguerre function as

$$h_n(\cdot) = \sum_{k_1=1}^{r} \cdots \sum_{k_n=1}^{r} L_{k_1 \ldots k_n}^{(n)} \prod_{i=1}^{n} \phi_{k_i}(\tau_i), \tag{11}$$

where $\{\phi_{k_i}(\tau_i)\}_{k_i=1,..,r|i=1,..,n}$ are the set of orthogonal Laguerre basis functions and $L_{k_1 \ldots k_n}^{(n)}$ are the Laguerre coefficients. Now, substitution of Eq. (11) in Eq. (10) will give the approximated Volterra-Laguerre model as

$$\begin{aligned}\gamma(t) = &\sum_{n=1}^{R} \sum_{\tau_1=0}^{M-1} \cdots \sum_{\tau_n=0}^{M-1} \sum_{k_1=1}^{r} \cdots \sum_{k_n=1}^{r} L_{k_1 \ldots k_n}^{(n)} \\ &\times \prod_{i=1}^{n} \phi_{k_i}(\tau_i)\, a(t-\tau_i) + v(t).\end{aligned} \tag{12}$$

By defining the parameter $l_k(t) = \sum_{\tau=0}^{M-1} \phi_k(\tau)\, a(t-\tau)$, Eq. (12) can be rewritten as

$$\gamma(t) = \sum_{n=1}^{R} \sum_{k_1=1}^{r} \cdots \sum_{k_n=1}^{r} L_{k_1 \ldots k_n}^{(n)} \prod_{i=1}^{n} l_{k_i}(t) + v(t). \tag{13}$$

Eq. (13) can be seen to fit in the form of linear regression as

$$\gamma(t) = \bar{S}^T \bar{\Phi}(t) + v(t), \tag{14}$$

where

$$\bar{S} = \left[\bar{S}^{(1)}, ..., \bar{S}^{(R)}\right]^T \in \mathbb{R}^{(r+\ldots+r^R)\times 1}, \tag{15}$$

$$\bar{\Phi}(t) = \left[\bar{\Phi}^{(1)}(t), ..., \bar{\Phi}^{(R)}(t)\right]^T \in \mathbb{R}^{(r+\ldots+r^R)}, \tag{16}$$

with

$$\begin{aligned}&\bar{S}^{(n)} = \left[L_{1\ldots 1}^{(n)} \cdots L_{r\ldots r}^{(n)}\right] \in \mathbb{R}^{1\times r^n},\ l_{k_1\ldots k_n}^{(n)}(t) = \prod_{i=1}^{n} l_{k_i}(t) \\ &\bar{\Phi}^{(n)}(t) = \left[l_{1\ldots 1}^{(n)}(t) \cdots l_{r\ldots r}^{(n)}(t)\right] \in \mathbb{R}^{1\times r^n}\end{aligned}.$$

The objective is to estimate the parameter vector $\bar{s}$ and $\bar{S}$ by minimizing the quadratic cost functions of (5) and (14) in a distributed manner using an ad-hoc WSN.

IV. Distributed Volterra-Laguerre Modeling and Distributed Wiener modeling

Let us consider an ad-hoc WSN with P number of spatially dispersed sensors where each sensor has the ability to measure the output for any input. Further, each node j has the processing capability to estimate the desired parameters locally and can exchange the estimate with its neighboring nodes $\mathcal{N}_j$. At any time t, node j measures the scalar output $\gamma_j(t)$ corresponding to input $\{a_j(\tau)|\, \tau = (t-M+1), .., t\}$. The scalar measurements of all the nodes are stacked in global vector $\bar{\Gamma}(t) = [\gamma_1(t), ..., \gamma_P(t)]^T \in \mathbb{R}^{P\times 1}$ with their corresponding regressors stacked in global matrix $\boldsymbol{U}(t) = \left[\bar{\Phi}_1(t), ..., \bar{\Phi}_P(t)\right] \in \mathbb{R}^{(r+\ldots+r^R)\times P}$ and then estimate the $\left(r+\ldots+r^R\right)\times 1$ vector $\bar{S}$ by minimization of

$$\begin{aligned}\hat{\bar{S}} &= \arg\min_{\bar{S}} E\left\|\bar{\Gamma}(t) - \boldsymbol{U}^T(t)\,\bar{S}\right\|^2 \\ &= \arg\min_{\bar{S}} \sum_{j=1}^{P} E\left[\left(\gamma_j(t) - \bar{\Phi}_j^T(t)\,\bar{S}\right)^2\right],\end{aligned} \tag{17}$$

where E is the expectation operator. Eq. (17) represents the global cost function. In order to obtain the optimal estimate of $\bar{S}$, all the sensor nodes need to send their data to FC for further processing. This centralized approach requires a powerful CPU and a large amount of communication resources. To overcome these limitations, an adaptive distributed algorithm is designed.

To facilitate the distributed estimation of $\bar{S}$, auxiliary variables $\{\bar{S}_j\}_{j=1}^{P}$ are introduced to represent the local estimates at each sensor nodes. The convex optimization problem in (17) can be re-expressed as

$$\begin{aligned}\left\{\hat{\bar{S}}_j(t)\right\}_{j=1}^{P} = &\arg\min_{\{\bar{S}_j\}_{j=1}^{P}} \sum_{m=0}^{t} \sum_{j=1}^{P} \lambda^{t-m}\left[\gamma_j(m) - \bar{\Phi}_j^T(m)\bar{S}_j\right]^2 \\ &+ P^{-1}\lambda^t \sum_{j=1}^{P} \bar{S}_j^T \boldsymbol{\Psi}_0 \bar{S}_j, \\ &\text{s.t. } \bar{S}_j = \bar{S}_{j'},\ j \in [1, .., P],\ j' \in \mathcal{N}_j,\end{aligned} \tag{18}$$

where λ is the forgetting factor and $\boldsymbol{\Psi}_0$ is the positive definite matrix used for regularization. Since WSN is connected, Eq. (17) and (18) are equivalent because $\{\bar{S}_j = \bar{S}\}_{j=1}^{P}$.

ADMM is employed to optimize (18) in a distributed fashion which gives the online recursive estimate of $\bar{S}$. Further, this adaptive algorithm can track the time-varying behavior of the nonlinear systems. To apply ADMM, auxiliary variables $\left\{\bar{z}_{j'}^{j}\right\}$ for $j' \in \mathcal{N}_j$ are considered with consensus constraints that are equivalent to the constraints specified in (18) as

$$\bar{S}_j = \bar{z}_j^{j'}, \bar{S}_j = \bar{z}_{j'}^{j}\ for\ j \in [1, P],\ j' \in \mathcal{N}_j,\ j \neq j'. \tag{19}$$

Now, a decomposable structure of the quadratically augmented Lagrangian function is formed with consensus constraints specified in (19) as) as

$$\begin{aligned}&L_a(\bar{S},\ \bar{z},\ \bar{\omega},\ \bar{\mu}) = \\ &\sum_{j=1}^{P} \sum_{m=0}^{t} \lambda^{t-m}[\gamma_j(m) - \bar{\Phi}_j^T(m)\bar{S}_j]^2 + \frac{\lambda^t}{P} \sum_{j=1}^{P} \bar{S}_j^T \boldsymbol{\Psi}_0 \bar{S}_j \\ &+ \sum_{j=1}^{P} \sum_{j' \in \mathcal{N}_j} \left[(\bar{\omega}_j^{j'})^T (\bar{S}_j - \bar{z}_j^{j'}) + (\bar{\mu}_j^{j'})^T (\bar{S}_j - \bar{z}_{j'}^{j})\right] \\ &+ \frac{c}{2} \sum_{j=1}^{P} \sum_{j' \in \mathcal{N}_j} \left[\left\|\bar{S}_j - \bar{z}_j^{j'}\right\|^2 + \left\|\bar{S}_j - \bar{z}_{j'}^{j}\right\|^2\right],\end{aligned} \tag{20}$$

where c represents positive penalty coefficient, $\bar{\omega}$ and $\bar{\mu}$ are the Lagrangian multipliers, $\bar{S} = \{\bar{S}_j\}_{j=1}^{P}$, $\bar{z} = \left\{\bar{z}_j^{j'}\right\}_{j\in[1,..,P]}^{j'\in\mathcal{N}_j}$ and $[\bar{\omega}, \bar{\mu}] = \left\{\bar{\omega}_j^{j'}, \bar{\mu}_j^{j'}\right\}_{j\in[1,..,P]}^{j'\in\mathcal{N}_j}$. At time instant $t+1$ and ADMM iteration k, Eq. (20) is minimized to get the optimum parameters in a distributed recursive fashion. The first step of ADMM updates the Lagrangian multipliers using gradient ascent method of optimization as

$$\bar{\omega}_j^{j'}(t+1;k) = \bar{\omega}_j^{j'}(t+1;k-1) + c\left[\bar{S}_j(t+1;k) - \bar{z}_j^{j'}(t+1;k)\right] \tag{21}$$

$$\bar{\mu}_j^{j'}(t+1;k) = \bar{\mu}_j^{j'}(t+1;k-1) + c\left[\bar{S}_j(t+1;k) - \bar{z}_{j'}^{j}(t+1;k)\right]. \tag{22}$$

Second step in ADMM minimizes the expression (20) w.r.t. $\bar{S}_j$ with all other variables fixed to their most updated values. The third step in ADMM involve the minimization of (20) w.r.t. $\bar{z}_j^{j'}$ assuming all other variables kept to their warm-start values. Then the variables $\bar{S}_j$ and $\bar{z}_j^{j'}$ can be respectively updated in the recursive form as

$$\bar{S}_j(t+1;k+1) = \left[2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\Phi}_j(m)\bar{\Phi}_j^T(m) + \frac{\lambda^{t+1}}{P}\mathbf{\Psi}_0\right) + 2c\left|\mathcal{N}_j\right| I\right]^{-1} \times \begin{bmatrix} 2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\Phi}_j(m)\gamma_j(m)\right) \\ -\sum_{j'\in\mathcal{N}_j}\left(\bar{\omega}_j^{j'}(t+1,k) + \bar{\mu}_j^{j'}(t+1,k)\right) \\ +c\sum_{j'\in\mathcal{N}_j}\left(\bar{z}_j^{j'}(t+1;k) + \bar{z}_{j'}^{j}(t+1;k)\right) \end{bmatrix} \tag{23}$$

$$\bar{z}_j^{j'}(t+1;k+1) = 0.5\left[\bar{S}_j(t+1;k+1) + \bar{S}_{j'}(t+1;k+1)\right] + 0.5c^{-1}\left[\bar{\omega}_j^{j'}(t+1;k) + \bar{\mu}_{j'}^{j}(t+1;k)\right]. \tag{24}$$

Using Eq. (24) into (21) and (22), and if Lagrange multipliers are initialized as $\bar{\omega}_j^{j'}(t+1;0) = -\bar{\mu}_{j'}^{j}(t+1;0)$, it follows that $\bar{\omega}_j^{j'}(t+1;k) = -\bar{\mu}_{j'}^{j}(t+1;k)$, $\forall t$, k which turns out to be

$$\bar{\omega}_j^{j'}(t+1;k) = \bar{\omega}_j^{j'}(t+1;k-1) + \frac{c}{2}\left[\bar{S}_j(t+1;k) - \bar{S}_{j'}(t+1;k)\right]. \tag{25}$$

Now, substituting Eq. (24) into Eq. (23) and following $\bar{\omega}_j^{j'}(t+1;k) = -\bar{\mu}_{j'}^{j}(t+1;k)$, $\forall t$, k, Eq. (23) can be rewritten as

$$\bar{S}_j(t+1;k+1) = \left[2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\Phi}_j(m)\bar{\Phi}_j^T(m) + \frac{\lambda^{t+1}}{P}\mathbf{\Psi}_0\right) + 2c\left|\mathcal{N}_j\right| I\right]^{-1} \times \begin{bmatrix} 2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\Phi}_j(m)\gamma_j(m)\right) \\ -\sum_{j'\in\mathcal{N}_j}\left(\bar{\omega}_j^{j'}(t+1,k) - \bar{\omega}_{j'}^{j}(t+1,k)\right) \\ +c\sum_{j'\in\mathcal{N}_j}\left(\left[\bar{S}_j(t+1;k) + \bar{S}_{j'}(t+1;k)\right]\right) \end{bmatrix}. \tag{26}$$

Recursions (25) and (26) are repeated for K number of ADMM iterations i.e. $k = 1, ..., K$ to enforce the consensus constraints given in (19).

NOTE 1: *The distributed algorithm for the identification of Wiener model can be obtained in a similar manner by mimicking the steps from (17) to (26). One can easily find that the formulation will be similar but the dimension of the parameter vectors will differ. The update equations for Lagrangian multiplier and unknown parameter vector can be given as follows:*

$$\bar{l}_j^{j'}(t+1;k) = \bar{l}_j^{j'}(t+1;k-1) + \frac{c}{2}\left[\bar{s}_j(t+1;k) - \bar{s}_{j'}(t+1;k)\right]. \tag{27}$$

$$\bar{s}_j(t+1;k+1) = \left[2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\zeta}_j(m)\bar{\zeta}_j^T(m) + \frac{\lambda^{t+1}}{P}\mathbf{\Omega}_0\right) + 2c\left|\mathcal{N}_j\right| I\right]^{-1} \times \begin{bmatrix} 2\sum_{m=0}^{t+1}\left(\lambda^{t+1-m}\bar{\zeta}_j(m)\gamma_j(m)\right) \\ -\sum_{j'\in\mathcal{N}_j}\left(\bar{l}_j^{j'}(t+1,k) - \bar{l}_{j'}^{j}(t+1,k)\right) \\ +c\sum_{j'\in\mathcal{N}_j}\left(\left[\bar{s}_j(t+1;k) + \bar{s}_{j'}(t+1;k)\right]\right) \end{bmatrix} \tag{28}$$

where $\mathbf{\Omega}_0$ is a positive definite matrix used for regularization, $\left\{\bar{l}_j^{j'}\right\}_{j\in[1,..,P]}^{j'\in\mathcal{N}_j}$ are the Lagrange multipliers associated for distributed Wiener modeling.

NOTE 2: *At each time instant, recursions (25), (26), (27) and (28) are capable to attain their optimal solution as ADMM iteration increases. A large number of ADMM iterations i.e $k > 1$ will not be a problem for time-invariant system. But when the characteristics of nonlinear dynamical systems are time-varying then one ADMM iteration per time instant t is used to track the process i.e $(k = t)$.*

A. Communication and Computational Complexity of Distributed Volterra Modeling

The transmission cost per sensor node at each iteration is $\left\{\left(r + ... + r^R\right)\left(\left|\mathcal{N}_j\right| + 1\right)\right\}$ which corresponds to the Lagrange multipliers $\left\{\bar{\omega}_j^{j'}\right\}_{j'\in\mathcal{N}_j}$ and local estimate $\bar{S}_j$. The reception cost per sensor node at each iteration is $\left\{2\left|\mathcal{N}_j\right|\left(r + ... + r^R\right)\right\}$ which corresponds to the Lagrange multipliers $\left\{\bar{\omega}_{j'}^{j}\right\}_{j'\in\mathcal{N}_j}$ and local estimates of the neighboring nodes $\left\{\bar{S}_{j'}\right\}_{j'\in\mathcal{N}_j}$.

The computational complexity to update Lagrange multipliers at sensor j through (25) requires $O\left(\left|\mathcal{N}_j\right|\left(r + ... + r^R\right)\right)$. While updating parameter matrix $\bar{S}_j$ using (26), the significant computations arise from the inverting term. The inversion can be computed using the matrix inversion lemma [25, pg. 571] hence requires $O\left(\left(r + ... + r^R\right)^2\right)$ computations when $\lambda = 1$. For $\lambda < 1$, inversion term requires $O\left(\left(r + ... + r^R\right)^3\right)$ computations per iteration.

B. Communication and Computational Complexity of Distributed Wiener Modeling

The transmission cost per sensor node at each iteration is $\left\{\left(n_\alpha + n_\beta - 1\right)\left(\left|\mathcal{N}_j\right| + 1\right)\right\}$ which corresponds to the Lagrange multipliers $\left\{\bar{l}_j^{j'}\right\}_{j'\in\mathcal{N}_j}$ and local estimate $\bar{s}_j$. The reception cost per sensor node at each iteration is $\left\{2\left|\mathcal{N}_j\right|\left(n_\alpha + n_\beta - 1\right)\right\}$ which corresponds to the Lagrange multipliers $\left\{\bar{l}_{j'}^{j}\right\}_{j'\in\mathcal{N}_j}$ and local estimates of the neighboring nodes $\left\{\bar{s}_{j'}\right\}_{j'\in\mathcal{N}_j}$.

The computational complexity to update Lagrange multipliers at sensor j through (27) requires $O\left(\left|\mathcal{N}_j\right|\left(n_\alpha + n_\beta - 1\right)\right)$. The inversion term involved while updating of parameter matrix $\bar{s}_j$ in (28) can be computed using matrix inversion lemma hence requires $O\left(\left(n_\alpha + n_\beta - 1\right)^2\right)$ computations for $\lambda = 1$. For $\lambda < 1$, inversion term requires $O\left(\left(n_\alpha + n_\beta - 1\right)^3\right)$ computations per iteration.

C. Convergence Analysis

Proposition 1: *At any time instant t, for the arbitrary initialized values of $\bar{\omega}_j^{j'}(t;0)$ and $\bar{S}_j(t;0)$ in recursions (25) and (26) respectively with $c > 0$, the local estimates $\bar{S}_j(t;k)$ at any node j meet the consensus constraints as $k \to \infty$:*

$$\lim_{k\to\infty}\bar{S}_j(t,k) = \lim_{k\to\infty}\bar{S}_{j'}(t,k) = \bar{S}_{Copt}(t)\ \forall j \in [1, P], \tag{29}$$

where $\bar{S}_{Copt}(t)$ is the centralized optimal solution when the observation data from all the sensors at any time instant t are collected at FC.

Proof: The proof can be easily obtained by mimicking the steps involved while proving [26, Proposition 4.2, pg. 256]. This involves reformulating (17) to a similar form as [26, Eq. 4.76]. If we closely look (18), it has the same form of optimization problem as [26, Eq. 4.76]. It can also be observe that the constraints in problem (18) follow the [26, Assumption 4.1, pg. 255] hence [26, Proposition 4.2, pg. 256] is applicable which completes the proof.

Similar Proposition can be used to obtain the convergence analysis of distributed Wiener modeling.

V. Simulation Results

Consider an infinite-order, finite length ($M = 5$) Wiener type nonlinear dynamical system as shown in Fig. 1:

$$d(t) = 2.5a(t) + 2a(t-1) + 0.5a(t-2) + 0.1a(t-3) + 0.05a(t-4)$$
$$\gamma(t) = \frac{1}{1 + e^{-d(t)}}. \tag{30}$$

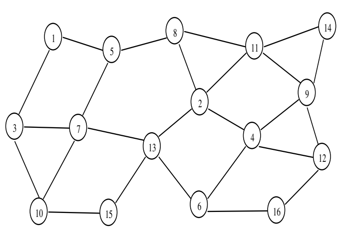
Let us employ the proposed algorithm to obtain the approximate 2^{nd}-order Volterra-Laguerre model and 2^{nd}-order nonlinear polynomial based Wiener model of the above nonlinear system. The results obtained are compared with the non-cooperative way of data processing.

A. Simulation Results of Distributed Volterra-Laguerre Model:

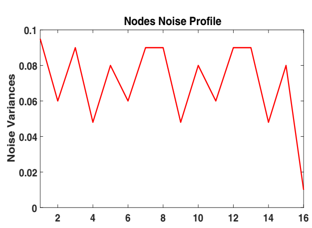
Second order Volterra-Laguerre model with $r = 4$ is used to model the above mentioned nonlinear system which requires 20 coefficients to be estimated. Whereas, traditional Volterra model needs to estimate $\left(M + M^2\right)$ i.e. 30 coefficients for the same. The discrete Laguerre functions [27]

$$\phi_k(t) = \sqrt{1-\rho^2}\rho^t(-\rho)^k \sum_{j=0}^{\min(k,t)} C_j^k C_j^t \left(\frac{\rho^2-1}{\rho^2}\right)^j,$$
$$\text{for} \quad k = 1, .., \infty, \; |\rho| < 1$$

are used with real valued Laguerre pole $\rho = 0.2$. In order to model the aforementioned nonlinear system in a distributed fashion, an ad-hoc WSN with sixteen-nodes is considered as shown in Fig. 2a. Each node in the WSN has the capability to capture the system dynamics. The system is excited with persistent random signal sequence $a(t)$ with zero mean and unit variance. The simulations are carried out by considering observation noise as zero-mean Gaussian with driving noise variances at different nodes as shown in Fig. 2b. We apply the proposed algorithm with $\lambda = 1$ and $c = 1$ to identify the model parameter matrix $\bar{S}$.



(a) Sixteen node ad-hoc WSN



(b) Noise-variance at different nodes

Fig. 2

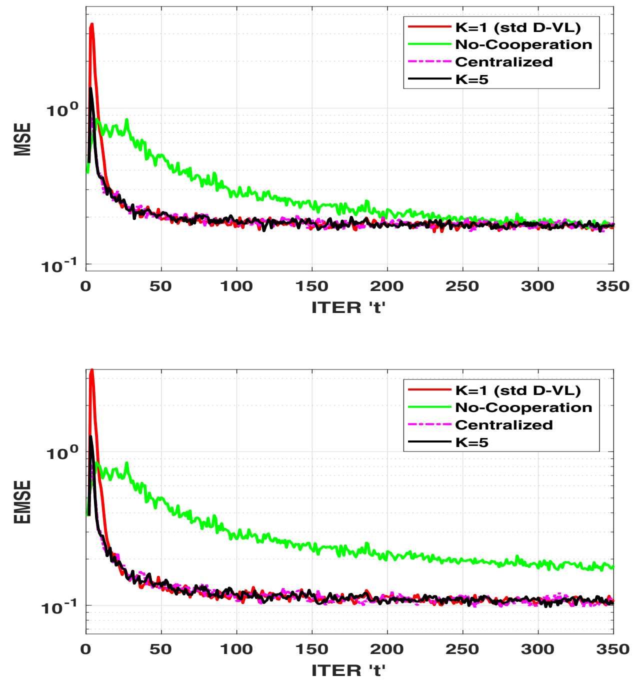


Fig. 3: Network performance of the proposed distributed Volterra-Laguerre model for the nonlinear system (30) with increasing values of K to reach centralized solution. Comparison with non-cooperative data processing algorithm is shown.

The effective performance of the proposed algorithm is justified by plotting the mean-square error (MSE) and excess mean-square error (EMSE) curves. These metrics are defined at any node j at any time instant t as

$$MSE_j(t) = E\left|\left(\gamma_j(t) + v(t)\right) - \hat{\bar{S}}_j^T(t)\,\bar{\Phi}_j(t)\right|^2,$$
$$EMSE_j(t) = E\left|\left(\gamma_j(t)\right) - \hat{\bar{S}}_j^T(t)\,\bar{\Phi}_j(t)\right|^2.$$

The overall network performance can be obtained by averaging the MSE's and EMSE's of all the nodes

$$MSE^{network} \triangleq \frac{1}{P}\sum_{j=1}^{P} MSE_j, \;\; EMSE^{network} \triangleq \frac{1}{P}\sum_{j=1}^{P} EMSE_j.$$

The MSE and EMSE curves are plotted against the number of time iterations as shown in Fig. 3. The simulation results are obtained by averaging 100 Monte-Carlo runs over 350 time iterations. It can be seen that as the number of ADMM iterations increases the performance in terms of convergence improves. The solution approaches centralized solution once K is increased to a certain value that depends on the topology of the network and the number of sensor nodes in the network. The standard distributed Volterra-Laguerre (std D-VL) model ($k = t$ or $K = 1$) when compared to no-cooperation strategy has improved error convergence rate as can be seen from the performance plots.

B. Simulation Results of Distributed Wiener Model:

The nonlinear system considered in (30) is approximated using Wiener model with invertible 2^{nd}-order polynomial nonlinearity hence $n_\beta = 2$, n_α should be equal to the number of terms in $d(t)$ i.e. $n_\alpha = 5$. Different type of basis functions

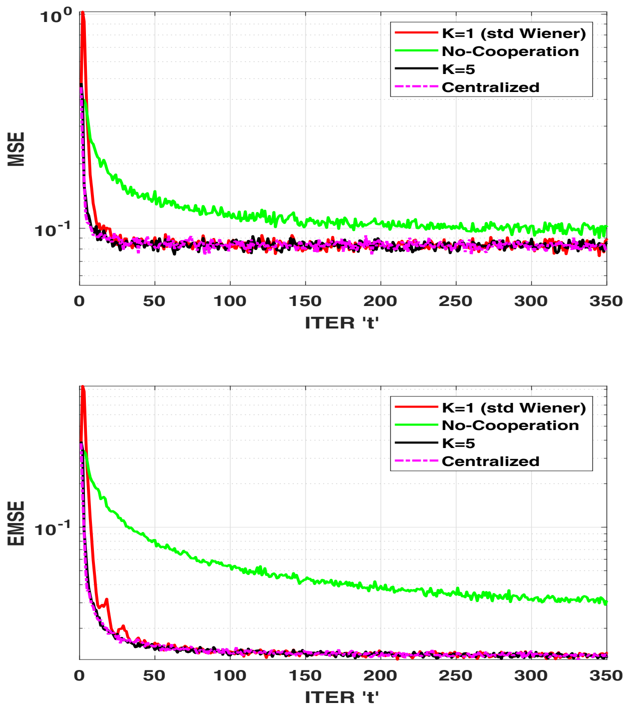


Fig. 4: Network performance of the proposed distributed Wiener model for the nonlinear system (30) with increasing values of K to reach centralized solution. Comparison with non-cooperative data processing algorithm is shown as well.

may also be used to approximate the system nonlinearity that may give different performance. Due to simple and easy in implementation, polynomial nonlinearity is used here. Performance plots are obtained with $\lambda = 1$ and $c = 1$ in a similar way as obtained for distributed Volterra-Laguerre model i.e. averaging the results of 100 Monte-Carlo runs over 350 time iterations. For distributed Wiener modeling, performance metrics at any node j at any time t can be expressed as

$$MSE_j(t) = E\left|(\gamma_j(t) + v(t)) - \hat{\bar{s}}_j^T(t)\,\bar{\zeta}_j(t)\right|^2,$$

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

The network performance curves are plotted in Fig. 4. Steady-state values of the performance curves are significantly less hence the proposed distributed Wiener modeling can be used effectively to model the system under consideration. The standard distributed Wiener (std Wiener) model ($k = t$ or $K = 1$) when compared to no-cooperation strategy have improved error convergence rate that can be seen from the performance plots. One can easily see that the steady-state error for Wiener nonlinear model is less than the Volterra-Laguerre model. One reason could be the assumptions made for approximating nonlinear Volterra kernels. The penalty coefficient c used here is chosen using hit and trial manner and is different for both the models to get the best results. One can choose the value of c in many another way too. The discussion of selecting penalty coefficient c is beyond the scope of this article.

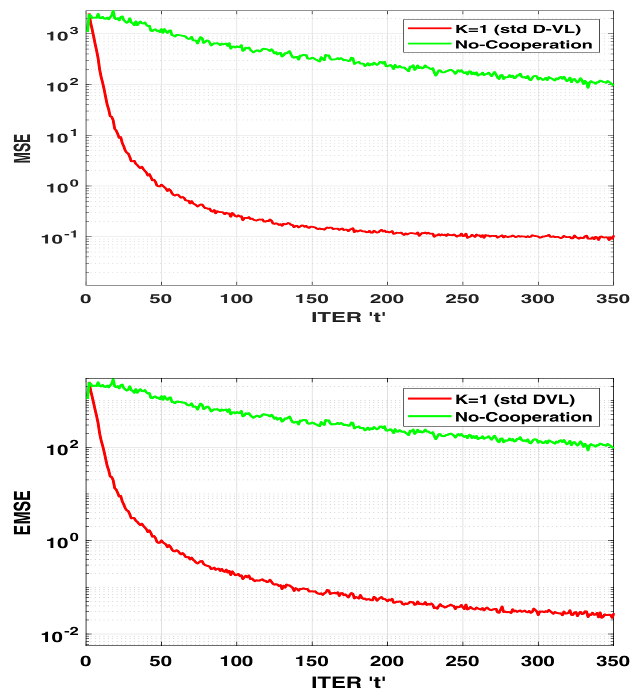


Fig. 5: Network performance of the 2^{nd}-order distributed Volterra-Laguerre model for the nonlinear system (31), comparison with non-cooperative data processing algorithm.

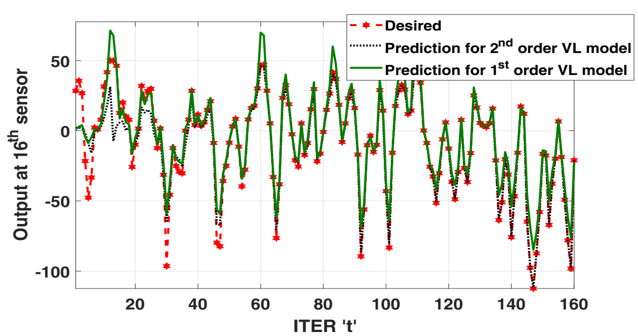


Fig. 6: Prediction performance of the 1^{st} and 2^{nd}-order distributed Volterra-Laguerre model at 16^{th} sensor node for nonlinear system (31).

Next, let us consider a 2^{nd}-order nonlinear system with finite fading memory factor ($M = 5$) as

$$d(t) = 2.5a(t) + 2a(t-1) + 0.5a(t-2) + 0.1a(t-3) + 0.05a(t-4)$$
$$\gamma(t) = 10d(t) - 2(d(t))^2. \quad (31)$$

NOTE 3: *The nonlinear function considered above is non-invertible in the domain of $d(t)$ hence cannot be approximated* *using traditional nonlinear Wiener model. But the considered model can be well approximated using Volterra-Laguerre model.*

Simulation results for the above mentioned 2^{nd}-order nonlinear system are obtained through the same sixteen-node adhoc WSN shown in Fig. 2a. Performance curves in Fig. 5 are obtained in a similar fashion as obtained for the infinite-order, finite length system in (30). The distributed algorithm for nonlinear system modeling gives better performance in terms of error convergence when compared to non-cooperative data processing.

Since the considered system is of 2^{nd}-order hence the 2^{nd} or higher order approximation of Volterra model gives the best result when compared to lower-order model approximations. The prediction performance for 1^{st}-order and 2^{nd}-order distributed Volterra-Laguerre model are depicted in Fig. 6, showing the prediction ability of the proposed distributed modeling algorithm. The predictions are plotted for first 160 observation samples.

VI. Concluding Remarks

In this work, we have proposed a distributed recursive way to identify Volterra-Laguerre model and Wiener model using ad-hoc WSNs. The deduced algorithm can be effectively used in real-time applications. Localized processing, as well as information sharing over WSN, is employed to incorporate the advantages of adaptive as well as distributed techniques which allows more responsive and robust performance. Both the standard models i.e. Volterra and Wiener models are reformulated using simple and powerful distributed convex optimization scheme called ADMM to facilitate the model identification in a distributed manner. The simulation results of modeling the nonlinear systems are plotted under noisy environment and are compared with the non-cooperative estimation to show the efficacy of the proposed algorithm. The designed algorithm can be useful in prediction and control applications for stationary as well as non-stationary dynamical systems with an arbitrary nonlinearity. In future, the work will be continued to identify the spatio-temporal model in a distributed fashion with adaptive penalty coefficient c.

References

[1] S. Boyd and L. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Transactions on circuits and systems*, vol. 32, no. 11, pp. 1150–1161, 1985.

[2] A. Khan and N. Vyas, "Application of Volterra and Wiener theories for nonlinear parameter estimation in a rotor-bearing system," *Nonlinear Dynamics*, vol. 24, no. 3, pp. 285–304, 2001.

[3] H.-X. Li, C. Qi, and Y. Yu, "A spatio-temporal Volterra modeling approach for a class of distributed industrial processes," *Journal of Process Control*, vol. 19, no. 7, pp. 1126–1142, 2009.

[4] 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.

[5] ——, "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.

[6] G. Favier, A. Y. Kibangou, and T. Bouilloc, "Nonlinear system modeling and identification using Volterra-PARAFAC models," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 1, pp. 30–53, 2012.

[7] B. W. Israelsen and D. A. Smith, "Generalized Laguerre reduction of the Volterra kernel for practical identification of nonlinear dynamic systems," *arXiv preprint arXiv:1410.0741*, 2014.

[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] 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.

[12] 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.

[13] T. Koh and E. Powers, "Second-order Volterra filtering and its application to nonlinear system identification," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 33, no. 6, pp. 1445–1455, 1985.

[14] P. Koukoulas and N. Kalouptsidis, "Nonlinear system identification using Gaussian inputs," *IEEE Transactions on Signal Processing*, vol. 43, no. 8, pp. 1831–1841, 1995.

[15] J. Li and J. Ilow, "A least-squares volterra predistorter for compensation of non-linear effects with memory in OFDM transmitters," in *Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual*. IEEE, 2005, pp. 197–202.

[16] J. Lee and V. J. Mathews, "A fast recursive least squares adaptive second order Volterra filter and its performance analysis," *IEEE transactions on signal processing*, vol. 41, no. 3, pp. 1087–1102, 1993.

[17] G. Alipoor and M. Savoji, "Nonlinear speech coding using backward adaptive variable-length quadratic filters," in *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*. IEEE, 2007, pp. 185–189.

[18] W. Ji and W.-S. Gan, "Identification of a parametric loudspeaker system using an adaptive Volterra filter," *Applied Acoustics*, vol. 73, no. 12, pp. 1251–1262, 2012.

[19] Z. Sigrist, E. Grivel, and B. Alcoverro, "Estimating second-order Volterra system parameters from noisy measurements based on an LMS variant or an errors-in-variables method," *Signal Processing*, vol. 92, no. 4, pp. 1010–1020, 2012.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] S. Key, "Fundamentals of statistical signal processing: estimation theory," *Englewood Cliffs*, 1993.

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

[27] A. El Anes, K. Bouzrara, and J. Ragot, "Reduced model of linear systems via Laguerre filters," *Transactions of the Institute of Measurement and Control*, p. 0142331216687020, 2017.