A Robust Distributed Notch Filtering Algorithm for Frequency Estimation Over Sensor Networks

Wael Bazzi

Electrical and Computer Engineering Department
American University in Dubai
Dubai, UAE
Email: wbazzi@aud.edu

Amir Rastegarnia, Azam Khalili, Mahtab Bahrami

Department of Electrical Engineering
Malayer University
Malayer, Iran
Email: {rastegarnia,khalili,bahrami}@malayeru.ac.ir

***Abstract*—In this paper, we consider the distributed frequency estimation problem where nodes of a network collaborate with each other to estimate the frequency of a single-frequency signal from measurements corrupted by impulsive noise. In the proposed algorithm, we reduce the impulsive noise effect by using the maximum correntropy criteria (MCC). The MCC is a robust optimality criterion for non-Gaussian signal processing. In the proposed algorithm, each node employs an adaptive notch filter to filter the input noisy measurements. The nodes collaborate with each other to optimize a cost function (given in terms of the MCC) in such a way that the filter output resembles as closely as possible, the desired signal. To derive the algorithm, we first formulate the distributed frequency estimation problem in terms of the MCC. Next, we use the iterative gradient ascent approach in our solution. The developed approach will be referred to as the diffusion notch filter-MCC (dNF-MCC) algorithm. The effectiveness of the proposed algorithm is demonstrated by computer simulations.**

Keywords–Adaptive networks; frequency estimation; diffusion; notch filter.

I. Introduction

The frequency estimation problem appears in many practical applications, such as biomedical engineering, power systems, radar detection, source localization, and speech processing [1]. Several methods have been introduced in the literature for frequency estimation and tracking. In the absence of measurement noise, Pronys method can be applied [2]. For noisy environments, different algorithms such as linear prediction (LP) methods have been developed [3]. When SNR is low and limited (short data length) is available, the principal eigenvector (PE) method is a proper solution [4]. For the mentioned case, the total least squares (TLS) method can provide better frequency estimation performance [5]. Adaptive notch filtering based methods are also developed for the frequency estimation problem to track the time-varying frequencies.

All mentioned methods have been developed for single processing node. However, in many practical applications, such as radar, power systems, sensor networks we need to solve frequency estimation problems in a fully distributed manner. Recently, distributed estimation has become an important topic in signal processing research due to the developments in wireless networking and computer and sensor technologies. Several useful distributed solutions for the estimation problem have been developed, such as consensus strategies [6]–[8], adaptive networks (i.e., incremental strategies and diffusion strategies). It has been shown in [9] that adaptive networks are more stable than consensus networks and they provide better steady-state error performance. In this paper, we focus on an adaptive network based solution.

We adopt the term adaptive networks from [10] to refer to a collection of nodes that interact with each other and function as a single adaptive entity that is able to track statistical variations of data in real-time. The two major classes of adaptive networks are incremental strategy [11]–[14] and diffusion strategy [15]–[19]. Comparing the two, incremental algorithms require less communication among nodes of the networks while diffusion algorithms are scalable and more robust to link and node failure. In general, diffusion based algorithms consist of two steps including the adaptation step, where the node updates the weight estimate using local measurement data, and the combination step where the information from the neighbors are aggregated. Based on the order of these two steps, diffusion algorithms can be categorized into two classes known as the Combine-then-Adapt and Adapt-then-Combine (ATC).

In [20], a diffusion LMS algorithm for frequency estimation over sensor networks have been introduced. Although the algorithm works well in noisy environments, as we will show in this paper, it performs poorly when the data are disturbed by impulsive noises. To address this issue, we need to move beyond mean squared error (MSE) and exploit higher order moments of the error. To this end, we propose a new ATC diffusion algorithm which relies on the maximum correntropy criteria. MCC is a robust optimality criterion for non-Gaussian signal processing and has recently been successfully applied in adaptive filtering [21]–[23]. In the proposed algorithm, each node employs an adaptive notch filter to filter the input noisy measurements and generate the output signal. The nodes collaborate with each other to optimize a cost function (given in terms of the MCC) in such a way that the filter output resembles as closely as possible, the desired signal. To derive the proposed algorithm, we first formulate the distributed frequency estimation problem in terms of the MCC. Then, we resort to iterative gradient ascent approach to solve it and derive the proposed algorithm, which will be referred to as the diffusion notch filter-MCC (dNF-MCC) algorithm. We also present simulation results to show the effectiveness of the new proposed algorithm.

The remainder of this paper is organized as follows. Section

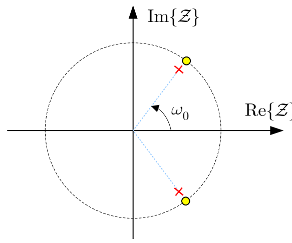

Figure 1. Pole-zero plot for the IIR notch filter with transfer function (2).

II briefly reviews the notch filter and the maximum correntropy criteria. In Section III, the proposed algorithm is introduced. In Section IV, we present simulation results to verify our theoretical analysis, and we conclude in Section V.

II. Preliminary Knowledge

To make the paper self-contained, in this section, we introduce the notch filter and maximum correntropy criteria.

A. Notch Filter

The transfer function for an M order IIR can be expressed as

$$H(z) = \frac{\sum_{i=0}^{M} a_i z^{-i}}{\sum_{i=0}^{P} b_i z^{-i}} = \frac{\prod_{i=1}^{M} (z - z_i)}{\prod_{i=1}^{P} (z - p_i)} \tag{1}$$

where in (1) $\{z_i\},\ i = 1, 2, \cdots, M$ and $\{p_i\},\ i = 1, 2, \cdots, P$ denote the zeros and poles of $H(z)$ respectively. As $H(z)$ reaches zero at $\{z_i\}$ and infinity at $\{p_i\}$, then we can obtain the transfer function of a notch filter with desired properties, by the appropriate placement of poles and zeros. In [24], the transfer function for a notch filter has been introduced as follows

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + \theta z^{-1} + z^{-2}}{1 + \rho\theta z^{-1} + \rho^2 z^{-2}} \tag{2}$$

where $\theta = -2\cos(\omega_0 T)$ (T is the sampling period used to generate a discrete-time sinusoidal signal from a continuous time signal) and $0 \ll \rho < 1$. The idea is to place constrained pole-zero pairs with their angles equal to ω_0 relative to the horizontal axis on the pole-zero plot [24] (See Figure 1). Taking the inverse $\mathcal{Z}$ transform of (2), we can obtain the input-output relation for the notch filter as

$$y(n) = -\rho\theta y(n-1) - \rho^2 y(n-2]) + x(n) + \theta x(n-1) + x(n-2) \tag{3}$$

In this paper, we consider the given notch filter model in (2) to develop our proposed algorithm.

B. Maximum Correntropy Criteria

For two scalar random variables X and Y the Correntropy is defined by [21]

$$\begin{aligned} C_\sigma(X, Y) &\triangleq \mathbb{E}\left[\kappa_\sigma(X - Y)\right] \\ &= \kappa_\sigma(x - y) f_{X,Y}(x, y) dx dy \end{aligned} \tag{4}$$

where $\kappa_\sigma(\cdot)$ is a shift-invariant Mercer kernel, with the kernel width $\sigma > 0$ and $f_{X,Y}(x, y)$ denotes the joint probability distribution function of X and Y. The most widely used kernel in correntropy is the complex Gaussian kernel which is given by

$$\kappa_\sigma(\zeta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-|\zeta|^2}{2\sigma^2}\right) \tag{5}$$

Comparing correntropy with MSE, we note that Correntropy is a local similarity measure, whereas MSE is global; meaning that all the samples in the joint space contribute appreciably to the value of the similarity metric, while the locality of correntropy means that the value is primarily dictated by the kernel function along the $x = y$ line. Thus, we can use the localization provided by the kernel width to reduce the effects of outliers in the measured data. Note that the metrics, such as MSE that rely only on the second order moment can easily get biased in such conditions.

Remark 1. *It must be noted that in practice, the joint pdf $f_{X,Y}(x, y)$ is unknown and only finite number of samples $\{x_t, y_t\},\ t = 1, 2, \ldots, L$ from X and Y are available. Thus, a sample estimator for correntropy can be defined as*

$$\hat{C}_\sigma(X, Y) = \frac{1}{L} \sum_{t=1}^{L} \kappa_\sigma(x_t - y_t)^2 \tag{6}$$

Remark 2. *In general, a larger kernel size makes a kernel-based algorithm less robust to the outliers, while a smaller kernel size makes the algorithm stall. Note that as $\sigma \to \infty$ the MCC approximately becomes equivalent to the MSE criterion.*

III. Proposed Algorithm

We consider a connected sensor network with N sensors (nodes) and denote it by a set $\mathcal{N} = \{1, 2, \cdots, N\}$. We denote by $\mathcal{N}_k$ the neighborhood nodes of node k where, by definition, we have $k \in \mathcal{N}_k$. The network is deployed to estimate the frequency of a sinusoidal signal $s(t) = A\sin(\omega_0 t + \phi)$ through the collected measurements by its nodes. We can assume that at any discrete time instant n, the observed discrete measurement by the node can be expressed by

$$x_k(n) = A_k \sin(\omega_0 nT + \phi_k) + \epsilon_k(n) \tag{7}$$

where A_k and ϕ_k are the amplitude and initial phase respectively and $\epsilon_k(n)$ denotes the observation noise term which is modelled as zeros mean Gaussian with variance $\sigma^2_{\varepsilon,k}$. Note that the input-output relation for the notch filter embedded in node k is given by

$$\begin{aligned} y_k(n) = &-\rho\theta_k(n) y_k(n-1) - \rho^2 y_k(n-2) \\ &+ x_k(n) + \theta_k(n) x_k(n-1) + x_k(n-2) \end{aligned} \tag{8}$$

where $\theta_k(n)$ denotes the local estimate of θ at time instant n at node k. We can estimate θ at every node by an adaptive filter algorithm as follows: at time instant n, every node k uses $x_k(n)$ as the filter input and updates $\theta_k(n)$ to generate the output $y_k(n)$ such that as time evolves, $\theta_k(n)$ converges to $-2\cos(\omega_0 T)$. To this end, we need to consider a suitable cost function. Using the MCC, we can formulate the estimation of parameter θ as the following optimization problem:

$$\underset{\theta_k(n)}{\operatorname{argmax}}\ J(\theta_k(n)) \tag{9}$$

with

$$J(\theta_k(n)) = J_0 \sum_{k=1}^{N} \sum_{m=n-L+1}^{n} \exp\left(\frac{-(d_k(n) - y_k(n))^2}{2\sigma^2}\right) \tag{10}$$

where $J_0 = \frac{1}{L\sigma\sqrt{2\pi}}$. Note that once $\theta_k(n) \to -2\cos(\omega_0 T)$, the notch filter will reject the single-frequency signal $\sin(\omega_0 nT + \phi_k)$, so the desired output is $d_k(n) = 0$. Hence, the cost function in (10) will change to

$$J(\theta_k(n)) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=1}^{N} \exp\left(\frac{-y_k^2(n)}{2\sigma^2}\right) \tag{11}$$

Obviously the cost function in (11) can be expressed by the following equivalent form

$$J(\theta_k(n)) = \sum_{k=1}^{N} J_k(\theta_k(n)) \tag{12}$$

where

$$J_k(\theta_k(n)) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-y_k^2(n)}{2\sigma^2}\right) \tag{13}$$

Problems of the form in (11) can be solved by diffusion adaptive networks. The general adapt-then-combine (ATC) diffusion strategy solution for (11) is given by

$$\begin{aligned} \phi_k(n) &= \theta_k(n-1) + \mu\left(\nabla_\theta J_k(\theta_k(n-1))\right) \\ \theta_k(n) &= \sum_{\ell=1}^{N} c_{\ell k}\phi_\ell(n) \end{aligned} \tag{14}$$

where $\phi_k(n)$ denotes an intermediate estimate at node k, $\mu > 0$ is the step-size parameter and ∇_θ denotes the gradient $J(\theta_k(n-1)$ with respect to $\theta_k(n-1)$. Moreover, nonnegative coefficients $c_{\ell k}$ satisfy the following conditions

$$\begin{cases} c_{\ell k} = 0, & \ell \notin \mathcal{N}_k \\ \sum\limits_{k=1}^{N} c_{\ell k} = 1 & k \in \mathcal{N}_k \end{cases} \tag{15}$$

Substituting $y_k(n)$ in (11) and taking the gradient with respect to $\theta_k(n-1)$ yields

$$\begin{aligned} \nabla_\theta J_k(\theta_k(n-1)) = \frac{-y_k(n)}{\sqrt{2\pi}\sigma^3}\Big(&- \rho y_k(n-1) \\ &+ x_k(n-1)\Big) \exp\left(\frac{-y_k^2(n)}{2\sigma^2}\right) \end{aligned} \tag{16}$$

Replacing (16) in (14) gives the update equation for our proposed algorithm as follows

$$\begin{aligned} \phi_k(n) &= \theta_k(n-1) \\ &\quad - \frac{\mu}{\sqrt{2\pi}\sigma^3} y_k(n)\Big(-\rho y_k(n-1) + x_k(n-1)\Big) \\ &\quad \times \exp\left(\frac{-y_k^2(n)}{2\sigma^2}\right) \\ \theta_k(n) &= \sum_{\ell=1}^{N} c_{\ell k}\phi_\ell(n) \end{aligned} \tag{17}$$

The pseudo code for the proposed algorithm is given in Algorithm 1.

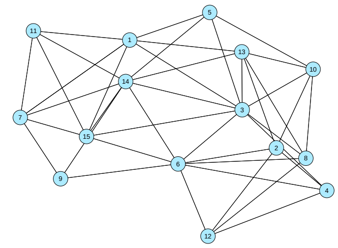

Figure 2. The network topology.

IV. SIMULATION RESULTS

In this section, we present the simulation results to show the effectiveness of the proposed algorithm. To this end, we consider a network with $N = 15$ nodes as shown in Figure 2. The frequency of the sinusoidal signal is $\omega_0 = 100$ and the sampling rate is 600 Hz. So, the observed signal by every node k can be expressed as

$$x_k(n) = \sin(2\pi 100 nT + \phi_k) + \varepsilon_k(n) \tag{18}$$

where ϕ_k is selected randomly for every node. To generate the impulsive noise at node k, we can assume that the measurement noise term is given by

$$\varepsilon_k(n) = g_{k,1}(n) + b_k(n) g_{k,2}(n) \tag{19}$$

where $g_{k,1}(n)$ and $g_{k,2}(n)$ are independent, zero mean Gaussian noise with variances $\sigma_{g,1}^2$ and $\sigma_{g,2}^2$, respectively, and $b_k(n)$ is a switch sequence of ones and zeros which is modeled as an independent and identically distributed Bernoulli random process with occurrence probabilitiesprob$(b_k(n) = 1) = pr$. Note that the variance of $g_{k,2}(n)$ is chosen to be very much larger than that of $g_{k,1}(n)$ so that when $b_k(n) = 1$, a large amplitude impulse is generated. In our simulations we set $\sigma_{g,1}^2 = 0.001$, $\sigma_{g,2}^2 = 5000\sigma_{g,1}^2$and $pr = 0.02$. For the notch filter, we set $\rho = 0.95$. For the given algorithm in [20], we select the step-size as 0.01, while for the proposed algorithm we set $\mu = 0.2$ and kernel size $\sigma = 1.5$. Note that these parameters are selected for the mentioned algorithms such that when the observation noise is Gaussian, their performance is similar. In Figure 3, the learning curves, in terms of the network mean-square deviation (MSD) metric, for both algorithms are presented. Note that the network MSD is defined as

$$\mathrm{MSD} \triangleq \frac{1}{N} \lim_{n\to\infty} \mathbb{E}\left[|\theta_k(n) - \theta|^2\right]$$

From Figure 3, we can see that the proposed algorithm achieves lower steady-state MSD than the dNF-LMS algorithm. The stead-state frequency for node $k = 4$ for both algorithms are plotted in Figure 4, where it is clear that the proposed algorithm provided more robust estimates than those of the dNF-LMS algorithm.

V. CONCLUSIONS

In this paper, we proposed a diffusion MCC-based notch filtering algorithm for the distributed frequency estimation problem. We resorted to iterative gradient ascent approach to derive the proposed algorithm. Simulation results showed that the proposed algorithm outperforms the available dNF-LMS algorithm when data are corrupted by impulsive noise.

 ISBN: 978-1-61208-572-2

Algorithm 1 Proposed Distributed Stackelberg Algorithm

Initialization
for $n = 3, 4, \cdots$ **do**
  For $k \in \mathcal{N}_k$ initialize $\theta_k(1)$, $\theta_k(2)$,$\theta_k(3)$, μ
  Adaptation
  Every node updates $\phi_k(n)$ as $\phi_k(n) = \theta_k(n-1) - \frac{\mu}{\sqrt{2\pi\sigma^3}} y_k(n) \left(-\rho y_k(n-1) + x_k(n-1)\right) \exp\left(\frac{-y_k^2(n)}{2\sigma^2}\right)$
  Combination
  Every node updates $\theta_k(n)$ as $\theta_k(n) = \sum_{\ell=1}^{N} c_{\ell k} \phi_\ell(n)$
end for

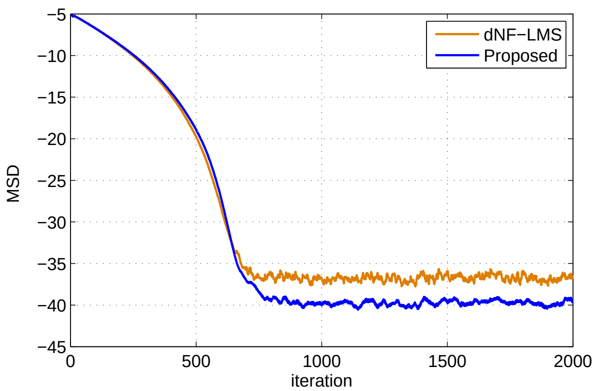

Figure 3. The network MSD learning curves for dNF-LMS and the proposed algorithm.

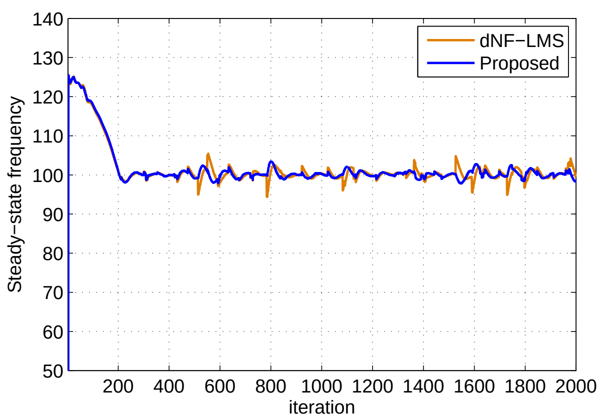

Figure 4. steady-state frequency estimations for dNF-LMS and the proposed algorithm.

References

[1] B. G. Quinn and E. J. Hannan, The Estimation and Tracking of Frequency, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2001.

[2] V. Slivinskas and V. Šimonytė, “Cramér-rao bound for estimates of frequencies and damping factors of real superimposed signals with multiple poles in noise,” Acta Applicandae Mathematica, vol. 38, no. 1, 1995, pp. 55–78.

[3] T. J. Ulrych and R. W. Clayton, “Time series modelling and maximum entropy,” Physics of the Earth and Planetary Interiors, vol. 12, no. 2, 1976, pp. 188 – 200.

[4] D. Tufts and R. Kumaresan, “Singular value decomposition and improved frequency estimation using linear prediction,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 30, no. 4, Aug 1982, pp. 671–675.

[5] M. D. Rahman and K.-B. Yu, “Total least squares approach for frequency estimation using linear prediction,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 35, no. 10, Oct 1987, pp. 1440–1454.

[6] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” IEEE Transactions on Automatic Control, vol. 49, no. 9, Sept 2004, pp. 1520–1533.

[7] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” IEEE Transactions on Automatic Control, vol. 55, no. 4, April 2010, pp. 922–938.

[8] L. Chen and J. Frolik, “Active consensus over sensor networks via selective communication,” in 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), June 2012, pp. 389–397.

[9] S. Y. Tu and A. H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks,” IEEE Transactions on Signal Processing, vol. 60, no. 12, Dec 2012, pp. 6217–6234.

[10] A. H. Sayed, “Adaptive networks,” Proceedings of the IEEE, vol. 102, no. 4, April 2014, pp. 460–497.

[11] C. G. Lopes and A. H. Sayed, “Incremental adaptive strategies over distributed networks,” IEEE Transactions on Signal Processing, vol. 55, no. 8, Aug 2007, pp. 4064–4077.

[12] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, “Distributed estimation over an adaptive incremental network based on the affine projection algorithm,” IEEE Transactions on Signal Processing, vol. 58, no. 1, Jan 2010, pp. 151–164.

[13] C. G. Lopes and A. H. Sayed, “Randomized incremental protocols over adaptive networks,” in 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, March 2010, pp. 3514–3517.

[14] A. Khalili, M. A. Tinati, and A. Rastegarnia, “An incremental block lms algorithm for distributed adaptive estimation,” in 2010 IEEE International Conference on Communication Systems, Nov 2010, pp. 493–496.

[15] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” IEEE Transactions on Signal Processing, vol. 56, no. 7, July 2008, pp. 3122–3136.

[16] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” IEEE Transactions on Signal Processing, vol. 56, no. 5, May 2008, pp. 1865–1877.

[17] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, “Multitask diffusion adaptation over asynchronous networks,” IEEE Transactions on Signal Processing, vol. 64, no. 11, June 2016, pp. 2835–2850.

[18] X. Zhao and A. H. Sayed, “Distributed clustering and learning over networks,” IEEE Transactions on Signal Processing, vol. 63, no. 13, July 2015, pp. 3285–3300.

[19] Z. J. Towfic and A. H. Sayed, “Stability and performance limits of adaptive primal-dual networks,” IEEE Transactions on Signal Processing, vol. 63, no. 11, June 2015, pp. 2888–2903.

[20] C. Li and H. Wang, “Distributed frequency estimation over sensor network,” IEEE Sensors Journal, vol. 15, no. 7, July 2015, pp. 3973–3983.

 ISBN: 978-1-61208-572-2

[21] A. Singh and J. C. Principe, "Using correntropy as a cost function in linear adaptive filters," in 2009 International Joint Conference on Neural Networks, June 2009, pp. 2950–2955.

[22] A. Khalili, A. Rastegarnia, M. K. Islam, and T. Y. Rezaii, "Steady-state tracking analysis of adaptive filter with maximum correntropy criterion," Circuits, Systems, and Signal Processing, vol. 36, no. 4, 2017, pp. 1725–1734.

[23] A. Khalili, A. Rastegarnia, and S. Sanei, "Robust frequency estimation in three-phase power systems using correntropy-based adaptive filter," IET Science, Measurement Technology, vol. 9, no. 8, 2015, pp. 928–935.

[24] L. Tan and J. Jiang, "Novel adaptive iir filter for frequency estimation and tracking [dsp tips tricks]," IEEE Signal Processing Magazine, vol. 26, no. 6, November 2009, pp. 186–189.

 ISBN: 978-1-61208-572-2