

Parametric Frugal Sensing for Autoregressive and Autoregressive Moving Average Power Spectra

Aritra Konar, *Student Member, IEEE*, and Nicholas D. Sidiropoulos, *Fellow, IEEE*

Abstract—Estimating the power spectrum of a wide-sense stationary random process is an important component of several signal processing tasks. Distributed spectrum sensing problems naturally emerge in cases where measurements of different realizations of a random process are collected at multiple spatial locations. This paper proposes a distributed power spectrum sensing framework for autoregressive (AR) and autoregressive moving-average (ARMA) processes. The sensing model comprises a network of scattered sensors which transmit randomly filtered, sample averaged, one bit quantized power measurements to a fusion center. First, assuming that sample averaging at each sensor is sufficient to converge to the ensemble average, it is shown that AR and ARMA power spectrum estimation from the received bits can be cast as (non-convex) optimization problems with special structure. Next, the sample averaging requirement is relaxed and maximum likelihood formulations are proposed, which take into account errors caused by quantization of inaccurate soft power measurements. Leveraging the block separable structure present in these formulations, we propose Block Coordinate Descent algorithms for obtaining approximate solutions, with favorable convergence properties. Simulation results demonstrate the efficacy of the proposed approaches for reconstructing power spectra from relatively few bits, when the model parametrization is valid.

Index Terms—Distributed spectrum sensing, parametric spectral analysis, auto-regressive moving-average processes, quantization, block coordinate descent.

I. INTRODUCTION

IN several modern signal processing applications (e.g., cognitive radio sensing, and radio astronomy), it is desirable to perform power spectrum (PS) estimation from compressed measurements drawn from the underlying wide-sense stationary (WSS) random process. Non-parametric methods for PS estimation from compressed analog measurements were developed in [2], [3]. These results were extended to the distributed

sensing setting in [4] where a non-parametric approach was adopted to reconstruct power spectra from one-bit compressed measurements. A Maximum-Likelihood (ML) based approach which takes into account errors introduced by quantizing noisy power measurements was also developed in [5]. These methods are well suited for cases when there is little prior knowledge about the structure of the underlying WSS process. However, if it is known *a priori* that the process admits a parametric representation, then this information can and should be exploited for developing parametric PS estimation methods with improved estimation performance—see [6]–[8] for a *moving average* (MA) parametrization in the context of [4], [5]. In this manuscript, we consider autoregressive (AR) and autoregressive-moving-average (ARMA) parametrizations for the underlying WSS process instead.

AR and ARMA power spectrum estimation from analog measurements is a classical problem in signal processing with many applications in geophysics, radar, sonar, radio astronomy, oceanography and speech processing (see [9] and references therein). Traditional AR PS estimation is a two-step process, where a non-parametric estimate of the autocorrelation is used to construct a system of linear equations that determine the AR model parameters. Prevailing methods for ARMA PS estimation also adopt a similar approach, which involves first obtaining an estimate of the autocorrelation sequence, followed by solving a linear system of equations to determine the AR parameters, which are in turn used to solve for the MA parameters. An overview of existing PS estimation methods can be found in [10].

Contributions: In contrast to the classical approaches, we consider the problem of AR and ARMA PS estimation in a distributed sensing scenario, where we employ a network of (possibly very cheap) sensors, each of which draws samples from the underlying WSS process, filters them using a random broadband filter, averages the power at the filter's output and then compresses the result to one bit. The single bit power measurements are then transmitted to a fusion center (FC), which aims to reconstruct the ambient PS. To the best of our knowledge, this is the first time that the problem of AR and ARMA PS estimation from a small number of one-bit power measurements has been considered.

First, we consider the case when sample averaging at each sensor is sufficient to enable accurate power estimation. Exploiting the underlying parametric structure and other pertinent properties of autocorrelation sequences, it is shown that problems of AR and ARMA PS estimation can be formulated as non-convex optimization programs, which appear hard to solve to global optimality in polynomial-time. Instead, the conditionally decomposable structure of the problem formulations are

Manuscript received November 29, 2015; revised April 16, 2016; accepted June 3, 2016. Date of publication June 14, 2016; date of current version August 19, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xavier Mestre. Supported in part by NSF AST 1247885, ECCS 1231504. The conference version of part of this work has appeared at ICASSP 2016 [1].

A. Konar is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455 USA (e-mail: konar006@umn.edu).

N. D. Sidiropoulos holds an ADC Endowed Chair in Digital Technology, Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455 USA (e-mail: nikos@umn.edu).

This paper has supplementary downloadable multimedia material available at <http://ieeexplore.ieee.org> provided by the authors. This includes a discussion of multi-bit transmission schemes (beyond the single-bit case discussed in this paper). This material is 144 KB in size.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2016.2580521

utilized to develop Block Coordinate Descent (BCD) algorithms for obtaining high quality sub-optimal solutions, featuring good convergence properties. Next, we relax the sample averaging requirement at each sensor and reformulate the problems of AR and ARMA PS estimation from possibly erroneous bit measurements obtained by quantizing noisy soft power estimates. It has been shown in [5] that the errors due to insufficient sample averaging can be well approximated by a Gaussian distribution, which is utilized to develop ML formulations for both AR and ARMA cases. Although the ML formulations also turn out to be non-convex, they still possess block separable structure, which is exploited to develop BCD algorithms for obtaining approximate solutions, again with good convergence properties. A comprehensive comparison of the proposed algorithms is carried out under different scenarios and various aspects of their performance are evaluated.

Relative to [1], this journal version adds the ARMA formulation and associated algorithms, along with the ML formulations for both AR and ARMA models and the corresponding algorithms.

The rest of the paper is organized as follows. We begin with some preliminaries in Section II, followed by a description of our system model in Section III. Problem formulations are presented in Section IV, while Section V contains the proposed algorithms. Simulation results are provided in Section VI, which offer insight on the selection of certain design parameters. Conclusion are drawn in Section VII.

We adopt the following notations throughout the article. The superscript $*$ is used to denote conjugation, whereas inline $*$ denotes convolution, as is customary. The superscript H is used to denote the Hermitian (conjugate) transpose of a vector/matrix, while T denotes plain transposition. Capital boldface is reserved for matrices, while vectors are denoted by small boldface. Scalar terms are represented in the normal face. The circularly symmetric complex Gaussian distribution is denoted by $\mathcal{CN}(\cdot, \cdot)$, whereas the $n \times n$ Identity matrix is represented by \mathbf{I}_n .

II. PRELIMINARIES

Consider a discrete-time WSS signal $x(n)$ and let $r_x(l) = \mathbb{E}[x(n)x^*(n-l)]$ denote its autocorrelation sequence, where $r_x(l) = r_x^*(-l)$, $\forall l \in \mathbb{Z}^+$, and \mathbb{Z}^+ is the set of all non-negative integers. In this article, we assume that $x(n)$ can be accurately described by one of the following families of parametric models.

A. AR Models

Assuming that $x(n)$ admits a AR representation of a certain order, we can characterize it as being generated by passing complex, circularly symmetric, uncorrelated, zero mean white Gaussian noise (WGN) of unit variance through a causal, linear shift invariant infinite impulse response (IIR) filter, whose rational transfer function $G(z)$ is given by

$$G(z) = \frac{1}{A_p(z)} = \frac{1}{1 + \sum_{k=1}^p \alpha(k) z^{-k}} \quad (1)$$

where p is the order of the AR process and $\alpha = [\alpha(1), \dots, \alpha(p)]^T \in \mathbb{C}^p$ are the AR parameters. Hence, $x(n)$ can be expressed as

$$x(n) + \sum_{k=1}^p \alpha(k) x(n-k) = v(n) \quad (2)$$

where $v(n) \sim \mathcal{CN}(0, 1)$. Assuming that the filter is stable ($\Leftrightarrow A_p(z)$ is a minimum-phase polynomial, i.e., its roots lie strictly inside the unit circle), the output $x(n)$ will be WSS. Thus, multiplying both sides of the above equation by $x^*(n-l)$ and taking expectation, one obtains the following recursive relation for the autocorrelation sequence

$$r_x(l) + \sum_{k=1}^p \alpha(k) r_x(l-k) = \delta(l), \quad \forall l \in \mathbb{Z}^+ \quad (3)$$

which are the Yule-Walker equations for an AR process. Given the parameters α , the power spectrum of the AR process $S_x(e^{j\omega})$ can be computed as follows

$$S_x(e^{j\omega}) = \frac{1}{|A_p(e^{j\omega})|^2} = \frac{1}{|1 + \sum_{k=1}^p \alpha(k) e^{-j\omega k}|^2} \quad (4)$$

B. ARMA Models

An ARMA(p, q) model is characterized by a causal, linear shift invariant filter with a rational transfer function of the form

$$H(z) = \frac{B_q(z)}{A_p(z)} = \frac{\sum_{k=0}^q \beta(k) z^{-k}}{1 + \sum_{k=1}^p \alpha(k) z^{-k}} \quad (5)$$

where p is the order of the AR process, q is the order of the MA process, $\beta = [\beta(0), \beta(1), \dots, \beta(q)]^T \in \mathbb{C}^{q+1}$ are the MA parameters and $\alpha = [\alpha(1), \dots, \alpha(p)]^T \in \mathbb{C}^p$ are the AR parameters. If the polynomial $A_p(z)$ is minimum-phase, then passing complex, circularly symmetric, uncorrelated, zero mean WGN of unit variance through $H(z)$ generates a WSS random process $x(n)$ at the output, which can be expressed as

$$x(n) + \sum_{k=1}^p \alpha(k) x(n-k) = \sum_{k=0}^q \beta(k) v(n-k) \quad (6)$$

where $v(n) \sim \mathcal{CN}(0, 1)$. Multiplying both sides of the above equation by $x^*(n-l)$ and taking expectation, the following recursive relation is obtained

$$r_x(l) + \sum_{k=1}^p \alpha(k) r_x(l-k) = \begin{cases} c(l) & : 0 \leq l \leq q \\ 0 & : l > q \end{cases} \quad (7)$$

where $c(l) = \beta(l) * h^*(-l) = \sum_{k=0}^q \beta(k) h^*(k-l)$ and $h(l)$ is the impulse response of the filter $H(z)$. Equation (7) represents the Yule-Walker Equations for an ARMA process. Given the parameters β and α , the PS of the ARMA process can be computed as

$$S_x(e^{j\omega}) = \frac{|B_q(e^{j\omega})|^2}{|A_p(e^{j\omega})|^2} = \frac{|\sum_{k=0}^q \beta(k) e^{-j\omega k}|^2}{|1 + \sum_{k=1}^p \alpha(k) e^{-j\omega k}|^2} \quad (8)$$

Alternatively, the PS can also be evaluated without explicit knowledge of the MA parameters β . This approach is outlined in

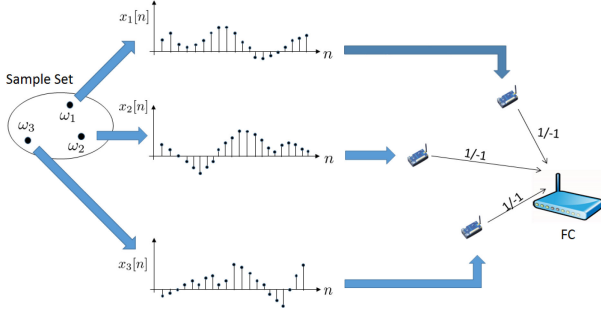


Fig. 1. Frugal Sensing System Model.

[11, p. 191] and is briefly reviewed here for completeness. Since $h(n)$ is assumed to be causal, it follows that $c(l) = 0, \forall l > q$. Hence, knowledge of $c(l)$ for $0 \leq l \leq q$ implies that $c(l)$ is known $\forall l \in \mathbb{Z}^+$. Denote the z-transforms of the causal and anti-causal parts of $c(l)$ as

$$[C(z)]_+ = \sum_{l=0}^{\infty} c(l) z^{-l}, \quad [C(z)]_- = \sum_{l=-\infty}^{-1} c(l) z^l \quad (9)$$

From the definition of $c(l)$, we have that

$$C(z) = B_q(z) H^*(1/z^*) = B_q(z) \frac{B_q^*(1/z^*)}{A_p^*(1/z^*)} \quad (10a)$$

$$\implies B_q(z) B_q^*(1/z^*) = C(z) A_p^*(1/z^*) \quad (10b)$$

$$= [C(z)]_+ A_p^*(1/z^*) + [C(z)]_- A_p^*(1/z^*) \quad (10c)$$

Since $A_p(z)$ is a causal polynomial by assumption, it holds that $A_p^*(1/z^*)$ is anti-causal. Thus the causal part of $B_q(z) B_q^*(1/z^*)$ is given by

$$[B_q(z) B_q^*(1/z^*)]_+ = [C(z)]_+ A_p^*(1/z^*) \quad (11a)$$

$$\implies |B_q(e^{j\omega})|_+^2 = [C(e^{j\omega})]_+ A_p^*(e^{j\omega}) \quad (11b)$$

The full MA spectrum can then be evaluated from the causal part $|B_q(e^{j\omega})|_+^2$ since the sequence corresponding to $B_q(z) B_q^*(1/z^*)$ is conjugate symmetric. Hence, even without explicitly knowing the MA parameters¹, the MA part of the PS can be evaluated from knowledge of the causal part of $c(l)$ and the AR parameters α . We will adopt this method for estimating the ARMA PS.

III. SYSTEM MODEL

A network sensing scenario as described in [4] is considered (see Fig. 1), where a network of M distributed, low-end sensors transmit randomly filtered, single bit power measurements to a fusion center (FC). Sensor m acquires samples of $x(n)$ in the form of the data sample vector $\mathbf{x}_m = [x_m(n), x_m(n-1), \dots, x_m(n-K+1)]^T \in$

¹If one desires, an estimate of the MA parameters β can be obtained by performing a *spectral factorization* step [12] on $|B_q(e^{j\omega})|_+^2$. However, since the factorization step is non-unique, it may not prove useful with regards to parameter identifiability, unless it is known *a priori* that $B_q(z)$ is a minimum phase polynomial.

\mathbb{C}^K , where \mathbf{x}_m can be drawn from a common sample path (e.g., $x_m(n) = x(n - \ell_m)$, which models possibly different acquisition times), or from another realization of the same WSS second-order ergodic process. Each sensor is also equipped with a finite impulse response (FIR) filter possessing a pseudo-random impulse response $\mathbf{g}_m \in \mathbb{C}^K$, with each entry being drawn from the following uniform distribution defined on a set of 4 complex symbols

$$g_m(n) = \begin{cases} \sim \mathcal{U}(\pm 1 \pm j) & : n \in [0, K-1] \\ 0 & : \text{otherwise} \end{cases} \quad (12)$$

where $\mathcal{U}(S)$ denotes the uniform probability mass function defined over the finite set S . Using \mathbf{g}_m , each sensor obtains random linear projections of the sampled data vectors of the form $z_m = \mathbf{g}_m^H \mathbf{x}_m$. Denote the average power of the random linear projections obtained at sensor m by $\rho_m = \mathbb{E}[|z_m|^2] = \mathbb{E}[|\mathbf{g}_m^H \mathbf{x}_m|^2] = \mathbf{g}_m^H \mathbf{R}_x \mathbf{g}_m$, where $\mathbf{R}_x \in \mathbb{C}^{K \times K}$ is the Toeplitz-Hermitian autocorrelation matrix of \mathbf{x}_m , whose first row is $[r_x(0), r_x(1), \dots, r_x(K-1)]$. Each sensor performs a sample averaging operation over N samples to obtain soft estimates of ρ_m of the form

$$\rho_m^{(N)} = \frac{1}{N} \sum_{n=1}^N |z_m(n)|^2 \quad (13)$$

Finally, each power estimate $\rho_m^{(N)}$ is compared to a single, sensor specific threshold t_m . If $\rho_m^{(N)} \geq t_m$, the sensor transmits a bit $b_m = 1$ to the FC, otherwise, it sends $b_m = -1$. Thus, each transmitted power measurement bit can be expressed as

$$b_m = \text{sign}(\rho_m^{(N)} - t_m) \quad (14)$$

where $\text{sign}(u) = 1$ if $u \geq 0$ and -1 otherwise $\forall u \in \mathbb{R}$. Assuming certain ergodic mixing conditions hold [13, p. 171], we have that $\lim_{N \rightarrow \infty} \rho_m^{(N)} = \rho_m$, i.e., sample averages converge to ensemble averages, which in turn corresponds to

$$b_m = \text{sign}(\mathbf{g}_m^H \mathbf{R}_x \mathbf{g}_m - t_m) \quad (15)$$

Hence, on receipt of a bit $b_m = 1$ (or $b_m = -1$) from sensor m , the FC infers that the inequality $\mathbf{g}_m^H \mathbf{R}_x \mathbf{g}_m \geq t_m$ (or $\mathbf{g}_m^H \mathbf{R}_x \mathbf{g}_m < t_m$) is satisfied.

The assumption that sample averaging at each sensor is sufficient to allow accurate estimation of ρ_m can be relaxed and the estimation errors introduced in the power measurements prior to thresholding due to insufficient sample averaging can be explicitly modeled. Define $e_m := \rho_m^{(N)} - \rho_m$ as the error due to insufficient sample averaging at sensor m . By virtue of the central limit theorem, these errors can be approximated as being independent, zero-mean Gaussian random variables with variances $\{\sigma_m^2\}_{m=1}^M$ [5]. Thus, after taking the errors into account, each measurement bit can be expressed as

$$b_m = \text{sign}(\mathbf{g}_m^H \mathbf{R}_x \mathbf{g}_m + e_m - t_m) \quad (16)$$

In both cases, the goal of the FC is to estimate the ambient PS from the binary power measurements $\{b_m\}_{m=1}^M$.

IV. PROBLEM FORMULATION

The bit-measurement model in (14) assumes sufficient sample averaging at each sensor, whereas (15) allows for insufficient sample averaging. We will consider both cases in the sequel. In each case, we seek to cast power spectrum estimation from binary power measurements as an optimization problem that incorporates all pertinent prior information in order to reduce the under-determinacy of our estimation setup (cf. the *binary* measurements).

First, note that since \mathbf{R}_x is Toeplitz-Hermitian, it can be expressed as $\mathbf{R}_x = \sum_{k=-(K-1)}^{K-1} r_x(k) \Theta_k^K$, where $\Theta_k^K \in \mathbb{R}^{K \times K}$ is a elementary Toeplitz matrix with ones on the k th diagonal and zeros elsewhere (by our notation, $k=0$ corresponds to the main diagonal, $k>0$ correspond to the super-diagonals and $k<0$ are the sub-diagonals). Exploiting this structure, the average power $\rho_m = \mathbf{g}_m^H \mathbf{R}_x \mathbf{g}_m$ can be expressed as

$$\mathbf{g}_m^H \mathbf{R}_x \mathbf{g}_m = \mathbf{g}_m^H \left(\sum_{k=-(K-1)}^{K-1} r_x(k) \Theta_k^K \right) \mathbf{g}_m \quad (17a)$$

$$= \sum_{k=-(K-1)}^{K-1} \underbrace{\mathbf{g}_m^H \Theta_k^K \mathbf{g}_m}_{c_m(k)} r_x(k) \quad (17b)$$

$$= c_m(0) r_x(0) + \sum_{k=1}^{K-1} 2\text{Re}\{c_m(k) r_x(k)\} \quad (17c)$$

$$= \mathbf{q}_m^T \mathbf{r}_x \quad (17d)$$

where $c_m(k)$ represents the k th lag of the deterministic autocorrelation sequence of the m th broadband filter with impulse response \mathbf{g}_m , and in the last step we have used the fact that $\text{Re}\{c_m(k) r_x(k)\} = \text{Re}\{c_m(k)\} \text{Re}\{r_x(k)\} + \text{Im}\{c_m(k)\} \text{Im}\{r_x(k)\}$ to define the vectors

$$\begin{aligned} \mathbf{q}_m &:= [c_m(0), 2\text{Re}\{c_m(1)\}, \dots, 2\text{Re}\{c_m(K-1)\}, \\ &\quad 2\text{Im}\{c_m(1)\}, \dots, 2\text{Im}\{c_m(K-1)\}]^T \in \mathbb{R}^{2K-1} \\ \mathbf{r}_x &:= [r_x(0), 2\text{Re}\{r_x(1)\}, \dots, 2\text{Re}\{r_x(K-1)\}, \\ &\quad 2\text{Im}\{r_x(1)\}, \dots, 2\text{Im}\{r_x(K-1)\}]^T \in \mathbb{R}^{2K-1} \end{aligned} \quad (18)$$

An initial feasible region for \mathbf{r}_x can be constructed from the structural properties of autocorrelation sequences. From the Cauchy-Schwarz inequality, we have $|r_x(k)| \leq r_x(0)$ for $k=1, \dots, K-1$. Thus, if an upper bound P_{\max} on the total signal power is known *a priori*, these inequalities define the initial feasible set $\mathbf{r}_x \in \mathcal{P}$, where \mathcal{P} is the bounded polyhedron

$$\begin{aligned} \mathcal{P} &:= \{\mathbf{r}_x \in \mathbb{R}^{2K-1} \mid 0 \leq r_x(0) \leq P_{\max}, \\ &\quad -r_x(0) \leq \text{Re}\{r_x(k)\} \leq r_x(0), \\ &\quad -r_x(0) \leq \text{Im}\{r_x(k)\} \leq r_x(0), k=1, \dots, K-1\} \end{aligned} \quad (19)$$

Additionally, we exploit the fact that the autocorrelation matrix \mathbf{R}_x associated with any autocorrelation vector \mathbf{r}_x of any

order must be positive semi-definite, which also ensures the non-negativity of the PS $\forall \omega \in [0, 2\pi]$. However, since we employ a finite parametrization of the autocorrelation sequence, the windowed PS estimate that is obtained by taking the discrete-time Fourier Transform (DTFT) of \mathbf{r}_x is not necessarily non-negative at all frequencies. In spite of this, incorporating the non-negativity of the windowed PS estimate in the constraint set has been shown [4] to improve the quality of PS estimation by reducing the under-determinacy of our problem setup. This constraint can be represented as $\mathbf{F} \mathbf{r}_x \geq \mathbf{0}$, where $\mathbf{F} \mathbf{r}_x$ is the discrete N_F -point PS estimate, $\mathbf{F} := \mathbf{F} \mathbf{W}$, \mathbf{F} is the $N_F \times (2K-1)$ phase shifted discrete Fourier transform (DFT) matrix, and

$$\mathbf{W} := \begin{bmatrix} \mathbf{0}_{K-1} & \mathbf{J}_{K-1} & -j\mathbf{J}_{K-1} \\ 1 & \mathbf{0}_{K-1}^T & \mathbf{0}_{K-1}^T \\ \mathbf{0}_{K-1} & \mathbf{I}_{K-1} & j\mathbf{I}_{K-1} \end{bmatrix} \quad (20)$$

where $\mathbf{0}_{K-1}$ is a vector of $K-1$ zeros, \mathbf{I}_{K-1} is the $K-1$ identity matrix, and \mathbf{J}_{K-1} is the $K-1$ anti-identity matrix. It has been established in [4, Appendix C] that $\mathbf{F} \mathbf{r}_x \geq \mathbf{0} \implies \mathbf{R}_x \succeq \mathbf{0}$, thus the latter Linear Matrix Inequality (LMI) constraint is redundant when the former constraint is enforced. If we assume sufficient averaging at each sensor m such that $e_m \approx 0$, then from (16) and (17d), we have that each measurement bit b_m corresponds to the following linear inequality in \mathbf{r}_x .

$$b_m (\mathbf{q}_m^T \mathbf{r}_x - t_m) \geq 0, \forall m \in \mathcal{M} \quad (21)$$

However, if the measurements at each sensor are not accurate enough, i.e., $\rho_m^{(N)} \neq \rho_m$, then the errors e_m may result in $\text{sign}(\mathbf{q}_m^T \mathbf{r}_x - t_m) \neq \text{sign}(\mathbf{q}_m^T \mathbf{r}_x + e_m - t_m)$, which corresponds to bits being flipped. In this case, we utilize the Gaussian distribution of the errors $\{e_m\}_{m=1}^M$ to derive the following error model. First, we introduce the set notation $\mathcal{M}_+ = \{m \in \mathcal{M} \mid b_m = +1\}$ and $\mathcal{M}_- = \{m \in \mathcal{M} \mid b_m = -1\}$. Assuming that errors are independent across sensors, the probability mass function of the bits $\{b_m\}_{m=1}^M$ can be parametrized by \mathbf{r}_x as follows

$$f(b_1, \dots, b_M; \mathbf{r}_x) \quad (22a)$$

$$= \prod_{m \in \mathcal{M}_+} \Pr(\rho_m + e_m \geq t_m) \prod_{m \in \mathcal{M}_-} \Pr(\rho_m + e_m < t_m) \quad (22b)$$

$$= \prod_{m \in \mathcal{M}_+} \Phi\left(\frac{\mathbf{q}_m^T \mathbf{r}_x - t_m}{\sigma_m}\right) \prod_{m \in \mathcal{M}_-} \Phi\left(\frac{-(\mathbf{q}_m^T \mathbf{r}_x - t_m)}{\sigma_m}\right) \quad (22c)$$

$$= \prod_{m \in \mathcal{M}} \Phi\left(\frac{b_m (\mathbf{q}_m^T \mathbf{r}_x - t_m)}{\sigma_m}\right) \quad (22d)$$

where $\Phi(x) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-u^2/2} du$ is the cumulative distribution function (CDF) of the Gaussian distribution. This model can be utilized in developing a Maximum Likelihood (ML) formulation which is robust to flipped bits, as we will soon demonstrate.

Finally, we can utilize the fact that $x(n)$ belongs to a certain class of parametric models in order to impose additional structure on \mathbf{r}_x . Depending upon which modeling assumption is invoked, we obtain the following formulations.

A. AR Models

1) *Error Free Formulation:* We first consider the case when we have accurate power measurements. For an AR process, the first K autocorrelation lags must satisfy (3). If the true autocorrelation of the AR process were known, then one could have formed a square system of $p + 1$ linear equations in α obtained from (3) by taking $l = 0 : p$ (shorthand for $l \in \{0, 1, \dots, p\}$). Since the autocorrelation matrix is positive definite if and only if the AR parameters are minimum-phase [11, p. 228], it follows that this system of linear equations can be uniquely solved for α , which also corresponds to the true minimum-phase solution. In practice, when the true autocorrelations are unknown, the traditional two-step approach uses sample autocorrelation estimates obtained from the WSS process to solve the Yule-Walker equations for α . The sample autocorrelation matrix in that case can also be shown to be positive definite under mild conditions [10, p. 93], and hence the Yule-Walker equations again admit a unique solution. Using sample autocorrelation estimates to solve the Yule-Walker equations can also be interpreted as solving an approximate Maximum-Likelihood estimation problem for α [14, p. 196], which yields the true AR parameters when the sample size is large. Hence, the sample autocorrelation lags in the range $-p : p$ constitute a *sufficient statistic* for estimating α .

Note that sample autocorrelation estimates of \mathbf{r}_x are not available in our setup since our problem involves estimation from a finite number of bits, and not samples of the WSS process. An estimate of the window of $2p - 1$ autocorrelation lags obtained from a few bits is not guaranteed to be a sufficient statistic for estimating α . Thus, we propose to estimate \mathbf{r}_x and α jointly. In order to make our problem less under-determined, we use the information contained in the higher autocorrelation lags. Assuming $K \geq p$, we obtain the following overdetermined system of linear equations

$$\begin{bmatrix} r_x(0) & r_x(-1) & \cdots & r_x(-p) \\ r_x(1) & r_x(0) & \cdots & r_x(-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(p) & r_x(p-1) & \cdots & r_x(0) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(K-1) & r_x(K-2) & \cdots & r_x(K-p-1) \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_1 \\ \vdots \\ \alpha_q \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (23)$$

Denoting $\tilde{\mathbf{R}}_x$ as the $K \times (p + 1)$ matrix defined in (23), $\tilde{\alpha} := [1, \alpha]^T \in \mathbb{C}^{p+1}$ and \mathbf{e}_1^K as the first canonic basis vector in \mathbb{R}^K , we can compactly express (23) as

$$\tilde{\mathbf{R}}_x \tilde{\alpha} = \mathbf{e}_1^K \quad (24)$$

which we refer to as the *extended Yule Walker equations* for an AR process. In order to impose AR structure on \mathbf{r}_x , we consider

the following formulation

$$\min_{\mathbf{r}_x, \alpha} \left\| \tilde{\mathbf{R}}_x \tilde{\alpha} - \mathbf{e}_1^K \right\|_2^2 \quad (25a)$$

$$\text{s.t. } b_m (\mathbf{q}_m^T \mathbf{r}_x - t_m) \geq 0, \forall m \in \mathcal{M} \quad (25b)$$

$$\mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (25c)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (25d)$$

Note that the problem is again non-convex since the cost function is the composition of a convex function with a bilinear function in \mathbf{r}_x and α .

2) *Maximum Likelihood Formulation:* When sample averaging is insufficient to allow accurate estimation of ρ_m , we propose the following ML reconstruction scheme, which is more robust to errors due to flipped bits. Given the bits $\{b_m\}_{m=1}^M$, we define the log-likelihood function $\text{LLN}(\mathbf{r}_x)$ as

$$\begin{aligned} \text{LLN}(\mathbf{r}_x) &:= \log f(b_1, \dots, b_M; \mathbf{r}_x) \\ &:= \sum_{m=1}^M \log \Phi \left(\frac{b_m (\mathbf{q}_m^T \mathbf{r}_x - t_m)}{\sigma_m} \right) \end{aligned} \quad (26)$$

The ML formulation is then given by

$$\min_{\mathbf{r}_x, \alpha} -\text{LLN}(\mathbf{r}_x) + \gamma \left\| \tilde{\mathbf{R}}_x \tilde{\alpha} - \mathbf{e}_1^K \right\|_2^2 \quad (27a)$$

$$\text{s.t. } \mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (27b)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (27c)$$

where the second term in the cost function is a regularization term which imposes AR structure on \mathbf{r}_x and λ is a penalty parameter. Note that $\text{LLN}(\mathbf{r}_x)$ is convex since the CDF of the Gaussian distribution is log-concave [15, p. 104] and by the affine composition rule. However, the overall problem is non-convex since the regularizer is non-convex.

B. ARMA Models

1) *Error Free Formulation:* Under the assumption that $x(n)$ can be described by an ARMA model, the autocorrelation lags should satisfy (7). When the true autocorrelation lags of the ARMA process are known, estimates of α and \mathbf{c} can be determined in a two-step procedure where the AR parameters are estimated from the autocorrelation lags first, and are then used to solve for \mathbf{c} . If the true autocorrelation matrix is unknown (as is usually the case), it can be estimated from the samples of the WSS process. In our setup, we are unable to do so since the FC only receives a finite number of bits, without having any access to the samples of the WSS process. Hence, we will attempt to estimate the parameters \mathbf{r}_x , α and \mathbf{c} jointly. In order to enhance estimation performance, the information in the higher autocorrelation lags should be exploited to reduce under-determinacy. Assuming $K > p + q$, we obtain the following overdetermined system of linear equations involving the first K autocorrelation lags

$$\tilde{\mathbf{R}}_x \tilde{\alpha} = \tilde{\mathbf{c}} \quad (28)$$

where $\tilde{\mathbf{c}} = [\mathbf{c}, 0, \dots, 0]^T \in \mathbb{C}^K$, $\mathbf{c} = [c(0), \dots, c(q)] \in \mathbb{C}^{q+1}$, $\tilde{\boldsymbol{\alpha}} = [1, \boldsymbol{\alpha}]^T \in \mathbb{C}^{p+1}$ and $\tilde{\mathbf{R}}_x \in \mathbb{C}^{K \times (p+1)}$ is defined as

$$\tilde{\mathbf{R}}_x = \begin{bmatrix} r_x(0) & r_x(-1) & \cdots & r_x(-p) \\ r_x(1) & r_x(0) & \cdots & r_x(-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(q) & r_x(q-1) & \cdots & r_x(q-p) \\ r_x(q+1) & r_x(q) & \cdots & r_x(q-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(K-1) & r_x(K-2) & \cdots & r_x(K-p-1) \end{bmatrix} \quad (29)$$

In order to impose ARMA structure, we propose to use the following formulation

$$\min_{\mathbf{r}_x, \boldsymbol{\alpha}, \mathbf{c}} \|\tilde{\mathbf{R}}_x \tilde{\boldsymbol{\alpha}} - \tilde{\mathbf{c}}\|_2^2 \quad (30a)$$

$$\text{s.t. } b_m(\mathbf{q}_m^T \mathbf{r}_x - t_m) \geq 0, \forall m \in \mathcal{M} \quad (30b)$$

$$\mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (30c)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (30d)$$

where $\mathcal{M} = \{1, 2, \dots, M\}$ denotes the set of sensors. Note that the problem is not convex since the cost function (30a) is the composition of a convex function with a bilinear function in \mathbf{r}_x and $\boldsymbol{\alpha}$.

2) *Maximum Likelihood Formulation*: Similar to the AR case, the ML estimation problem for ARMA models can be posed as

$$\begin{aligned} \text{LLN}(\mathbf{r}_x) &:= \log f(b_1, \dots, b_M; \mathbf{r}_x) \\ &:= \sum_{m=1}^M \log \Phi \left(\frac{b_m(\mathbf{q}_m^T \mathbf{r}_x - t_m)}{\sigma_m} \right) \end{aligned} \quad (31)$$

The ML formulation is then given by

$$\min_{\mathbf{r}_x, \boldsymbol{\alpha}, \mathbf{c}} -\text{LLN}(\mathbf{r}_x) + \lambda \|\tilde{\mathbf{R}}_x \tilde{\boldsymbol{\alpha}} - \tilde{\mathbf{c}}\|_2^2 \quad (32a)$$

$$\text{s.t. } \mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (32b)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (32c)$$

which is also non-convex in \mathbf{r}_x and \mathbf{c} .

V. ALGORITHMS

Although the problem formulations for both AR and ARMA models are non-convex, they exhibit block separable structure which makes them well suited for Block Coordinate Descent (BCD) methods [16], [17] for obtaining approximate solutions. BCD methods are a well known class of optimization tools that are applicable to problems whose cost function and constraints admit a conditionally decomposable structure. Since all problem formulations in this article possess bilinear cost functions subject to separable convex constraints, they naturally fall within the BCD framework, which yields the following simple algorithms.

A. AR Models

1) *Error Free Formulation*: Although problem (25) is non-convex, if either of the variables \mathbf{r}_x or $\boldsymbol{\alpha}$ is fixed, the resulting

subproblem is convex in the other variable. This suggests that \mathbf{r}_x and $\boldsymbol{\alpha}$ can be updated in an alternating fashion, which results in a simple 2-block BCD algorithm consisting of the following steps.

- *Initialization*: First, we solve the feasibility problem

$$\text{find } \mathbf{r}_x \quad (33a)$$

$$\text{s.t. } b_m(\mathbf{q}_m^T \mathbf{r}_x - t_m) \geq 0, \forall m \in \mathcal{M} \quad (33b)$$

$$\mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (33c)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (33d)$$

in order to obtain a truncated K -lag autocorrelation sequence that is consistent with the inequalities corresponding to the bit measurements $\{b_m\}_{m=1}^M$, which is then used to initialize the algorithm.

- *$\boldsymbol{\alpha}$ Update*: If \mathbf{r}_x is fixed, $\boldsymbol{\alpha}$ can be updated by solving (25a) with respect to $\boldsymbol{\alpha}$. In terms of $\boldsymbol{\alpha}$, (25a) can be represented as

$$g(\boldsymbol{\alpha}) := \|\bar{\mathbf{R}}_x \boldsymbol{\alpha} + \boldsymbol{\rho} - \mathbf{e}_1^K\|_2^2 = \|\bar{\mathbf{R}}_x \boldsymbol{\alpha} + \tilde{\boldsymbol{\rho}}\|_2^2 \quad (34)$$

where $\boldsymbol{\rho} \in \mathbb{C}^K$ is the first column of $\tilde{\mathbf{R}}_x$, $\bar{\mathbf{R}}_x \in \mathbb{C}^{K \times p}$ is obtained by deleting $\boldsymbol{\rho}$ from $\tilde{\mathbf{R}}_x$ and $\tilde{\boldsymbol{\rho}} = \boldsymbol{\rho} - \mathbf{e}_1^K$. Equation (34) is a least squares problem in $\boldsymbol{\alpha}$, whose closed form solution is given by

$$\boldsymbol{\alpha} = -(\bar{\mathbf{R}}_x^H \bar{\mathbf{R}}_x + \epsilon \mathbf{I}_K)^{-1} (\bar{\mathbf{R}}_x^H \tilde{\boldsymbol{\rho}}) \quad (35)$$

where $\epsilon > 0$ is a regularization parameter which guards against ill-conditioning of $\bar{\mathbf{R}}_x$. In the event that the estimated K -lag autocorrelation sequence is slowly decaying, the columns of $\bar{\mathbf{R}}_x$ will be nearly linearly dependent. The regularization term alleviates this problem, at the cost of introducing a bias into our estimate.

- *\mathbf{r}_x Update*: Given an estimate of $\boldsymbol{\alpha}$, the update for \mathbf{r}_x can be cast as a Quadratic Programming (QP) problem. To show this, we first express the cost function (25a) in terms of \mathbf{r}_x . Define the matrix $\mathbf{E}_{p+1}^K := [\mathbf{e}_1^K, \dots, \mathbf{e}_{p+1}^K]$, whose columns are the first $(p+1)$ canonical basis vectors in \mathbb{R}^K . Then, we have

$$\tilde{\mathbf{R}}_x \tilde{\boldsymbol{\alpha}} = \mathbf{R}_x^T \mathbf{E}_{p+1}^K \tilde{\boldsymbol{\alpha}} \quad (36a)$$

$$= \left(\sum_{k=-(K-1)}^{K-1} r_x(k) \boldsymbol{\Theta}_{-k}^K \right) \mathbf{E}_{p+1}^K \tilde{\boldsymbol{\alpha}} \quad (36b)$$

$$= \left(r_x(0) \boldsymbol{\Theta}_0 + \sum_{k=1}^{K-1} \text{Re}\{r_x(k)\} \underbrace{(\boldsymbol{\Theta}_k^K + \boldsymbol{\Theta}_{-k}^K)}_{\boldsymbol{\Phi}_k^K} \right) \mathbf{E}_{p+1}^K \tilde{\boldsymbol{\alpha}} \quad (36c)$$

$$+ \sum_{k=1}^{K-1} \text{Im}\{r_x(k)\} \underbrace{(j\boldsymbol{\Theta}_{-k}^K - j\boldsymbol{\Theta}_k^K)}_{\boldsymbol{\Psi}_k^K} \right) \underbrace{\mathbf{E}_{p+1}^K \tilde{\boldsymbol{\alpha}}}_{\boldsymbol{\gamma}} \quad (36d)$$

$$= r_x(0) \gamma + \sum_{k=1}^{K-1} \operatorname{Re} \{r_x(k)\} \Phi_k^K \gamma \quad (36e)$$

$$+ \sum_{k=1}^{K-1} \operatorname{Im} \{r_x(k)\} \Psi_k^K \gamma \quad (36f)$$

$$= \underbrace{[\gamma, \Phi_1^K \gamma, \dots, \Phi_{K-1}^K \gamma, \Psi_1^K \gamma, \dots, \Psi_{K-1}^K \gamma]}_{\mathbf{\Pi}} \mathbf{r}_x \quad (36g)$$

$$= \mathbf{\Pi} \mathbf{r}_x \quad (36h)$$

where $\mathbf{\Pi} \in \mathbb{C}^{K \times 2K-1}$. Thus, in order to update \mathbf{r}_x , we are required to solve a problem of the form

$$\min_{\mathbf{r}_x} \|\mathbf{\Pi} \mathbf{r}_x - \mathbf{e}_1^K\|_2^2 \quad (37a)$$

$$\text{s.t. } b_m(\mathbf{q}_m^T \mathbf{r}_x - t_m) \geq 0, \forall m \in \mathcal{M} \quad (37b)$$

$$\mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (37c)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (37d)$$

which is a QP problem in \mathbf{r}_x , and can be efficiently solved via convex programming.

The algorithm can be summarized as follows

Algorithm 1: 2-block BCD for AR Estimation.

Initialization: Solve the problem (33) to obtain a feasible truncated K -lag autocorrelation vector \mathbf{r}_x . Set $k := 0$.

Repeat

- Fix $\mathbf{r}_x^{(k)}$. Update $\alpha^{(k+1)}$ according to (35).
- Fix $\alpha^{(k+1)}$. Update $\mathbf{r}_x^{(k+1)}$ by solving the QP problem (37).
- Compute cost value $v^{(k+1)} = \|\tilde{\mathbf{R}}_x^{(k+1)} \tilde{\alpha}^{(k+1)} - \mathbf{e}_1^K\|_2^2$
- Set $k := k + 1$.

Until Improvement in cost function $<$ *tolerance factor* in the last 10 iterations OR specified no. of iterations exceeded.

Since the update step for each variable is conditionally optimal given the other variable, the algorithm generates a sequence of iterates with monotonically non-increasing cost. In addition, we have the following result.

Proposition 1: If we impose loose upper and lower bounds on the elements of α , as is implicitly done for $\epsilon > 0$ through norm regularization, then every limit point of Algorithm 1 is a stationary point of (25).

Proof: In [18, Theorem 3.1], it is shown that for any sequence of iterates generated by BCD, there exists a convergent subsequence whose limit point corresponds to a stationary point, provided the constraint set of each block is compact and the block which yields the best improvement is selected for update in each iteration. The feasible set of the block \mathbf{r}_x can be represented as $\mathbf{r}_x \in \mathcal{P} \cap \{\mathbf{r}_x \mid \mathbf{F} \mathbf{r}_x \geq \mathbf{0}\} \cap \{\mathbf{r}_x \mid b_m(\mathbf{q}_m^T \mathbf{r}_x - t_m) \geq 0, \forall m \in \mathcal{M}\}$. The second set represents a polyhedral cone, which is closed and convex, while the last constraint defines a convex polyhedron, which is also closed. It thus follows that

the intersection of these two sets is closed also. The first set is a convex polyhedron, which is closed and bounded, and hence, by the Heine-Borel Theorem [21, p. 41], is also compact. As the intersection of a compact set and a closed set is compact [21, p. 38], it follows that the feasible set of \mathbf{r}_x is compact. Since $\alpha \in \mathbb{C}^p$, which is not compact, we can impose a loose upper and lower bound on the elements of α in order to make it compact. Moreover, since there are only two blocks, after optimally updating one block, the other block is guaranteed to yield the best improvement in the next iteration. Thus, the update rule proposed in [18] corresponds to the alternating update for the case of BCD with two blocks. ■

2) *ML Formulation:* The ML formulation (27) is also bilinear in the variables \mathbf{r}_x and α , thus allowing us to adopt the same approach outlined in the previous section. The algorithm is composed of the following steps.

- *Initialization:* First, we solve the problem

$$\min_{\mathbf{r}_x, \alpha, \mathbf{c}} -\text{LLN}(\mathbf{r}_x) \quad (38a)$$

$$\text{s.t. } \mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (38b)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (38c)$$

in order to obtain an ML estimate of \mathbf{r}_x which is then used to initialize the algorithm.

- α *Update:* Fixing \mathbf{r}_x , the update for α is given by (35).
- \mathbf{r}_x *Update:* For a fixed α , the update for \mathbf{r}_x is the solution of the problem

$$\min_{\mathbf{r}_x} -\text{LLN}(\mathbf{r}_x) + \|\mathbf{\Pi} \mathbf{r}_x - \mathbf{e}_1^K\|_2^2 \quad (39a)$$

$$\text{s.t. } \mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (39b)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (39c)$$

which is a convex optimization problem in \mathbf{r}_x . The overall algorithm is given by

Algorithm 2: 2-block BCD for ML AR Estimation.

Initialization: Solve the problem (38) to obtain an ML estimate of the truncated K -lag autocorrelation vector \mathbf{r}_x . Set $k := 0$.

Repeat

- Fix $\mathbf{r}_x^{(k)}$. Update $\alpha^{(k+1)}$ according to (35).
- Fix $\alpha^{(k+1)}$. Update $\mathbf{r}_x^{(k+1)}$ by solving the problem (39).
- Compute cost value $v^{(k+1)} = -\text{LLN}(\mathbf{r}_x^{(k+1)}) + \|\tilde{\mathbf{R}}_x^{(k+1)} \tilde{\alpha}^{(k+1)} - \mathbf{e}_1^K\|_2^2$
- Set $k := k + 1$.

Until Improvement in cost function $<$ *tolerance factor* in the last 10 iterations OR specified no. of iterations exceeded.

Proposition 2: If we impose loose upper and lower bounds on the elements of α , every limit point of Algorithm 2 is a stationary point of (25).

Proof: This follows from the fact that when loose upper and lower bounds on α are applied, the conditions stated in [18] are satisfied. ■

B. ARMA Models

For ARMA models, we propose a 2-block and a 3-block BCD algorithm for obtaining approximate solutions to (30) and (32).

2-block approach:

1) *Error Free Formulation*: Note that if either of the variable blocks $\{\alpha, \mathbf{c}\}$ or \mathbf{r}_x is fixed, then (30) becomes convex in the other variable block. Hence, the variable blocks \mathbf{r}_x and $\{\alpha, \mathbf{c}\}$ can be updated in an alternating fashion. Our algorithm consists of the following steps.

- *Initialization*: An initial estimate of \mathbf{r}_x obtained from the solution of an instance of the feasibility problem (33) serves as the initialization point.
- $\{\alpha, \mathbf{c}\}$ *Update*: Since the constraints of (30) do not involve α, \mathbf{c} , given a feasible \mathbf{r}_x , we can obtain an estimate of the variables $\delta := [\alpha^T, \mathbf{c}^T]^T \in \mathbb{C}^{p+q+1}$ by simply solving the Yule-Walker equations with respect to δ . Performing some algebraic manipulations allows us to express the cost function (30a) in terms of δ as

$$f(\delta) := \|\tilde{\mathbf{R}}_x \tilde{\alpha} - \tilde{\mathbf{c}}\|_2^2 = \|\mathbf{Q}\delta + \rho\|_2^2 \quad (40)$$

where $\rho \in \mathbb{C}^K$ corresponds to the first column of $\tilde{\mathbf{R}}_x$ and the matrix $\mathbf{Q} \in \mathbb{C}^{K \times (p+q+1)}$ is defined as

$$\mathbf{Q} = [\tilde{\mathbf{R}}_x - \mathbf{E}_{q+1}^K] \quad (41)$$

Here, $\tilde{\mathbf{R}}_x \in \mathbb{C}^{K \times p}$ is obtained by deleting ρ from $\tilde{\mathbf{R}}_x$ and $\mathbf{E}_{q+1}^K \in \mathbb{R}^{K \times (q+1)}$ is a matrix whose columns comprise the first $(q+1)$ canonical basis vectors in \mathbb{R}^K . Thus, we are required to solve a least squares (LS) problem of the form

$$\min_{\delta} \|\mathbf{Q}\delta + \rho\|_2^2 \quad (42)$$

for which the closed form solution is given by

$$\delta = -(\mathbf{Q}^H \mathbf{Q} + \epsilon \mathbf{I}_K)^{-1} (\mathbf{Q}^H \rho) \quad (43)$$

where $\epsilon > 0$ is a regularization parameter. From (43), the variables α and \mathbf{c} are readily obtained.

- \mathbf{r}_x *Update*: Fixing δ , the update for \mathbf{r}_x reduces to a QP problem. Following the steps outlined in (36), the cost function (30a) can be expressed in terms of \mathbf{r}_x as

$$\|\tilde{\mathbf{R}}_x \tilde{\alpha} - \tilde{\mathbf{c}}\|_2^2 = \|\mathbf{\Pi} \mathbf{r}_x - \tilde{\mathbf{c}}\|_2^2 \quad (44)$$

Thus, in order to update \mathbf{r}_x , we are required to solve a problem of the form

$$\min_{\mathbf{r}_x} \|\mathbf{\Pi} \mathbf{r}_x - \tilde{\mathbf{c}}\|_2^2 \quad (45a)$$

$$\text{s.t. } b_m (\mathbf{q}_m^T \mathbf{r}_x - t_m) \geq 0, \forall m \in \mathcal{M} \quad (45b)$$

$$\mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (45c)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (45d)$$

which is a QP problem in \mathbf{r}_x .

The algorithm can be summarized as follows

The algorithm generates a sequence of iterates with monotonically non-increasing cost. In addition, we have the following result.

Algorithm 3: 2-block BCD for ARMA Estimation.

Initialization: Solve the problem (33) to obtain a feasible truncated K -lag autocorrelation vector $\mathbf{r}_x^{(0)}$. Set $k := 0$.

Repeat

- Fix $\mathbf{r}_x^{(k)}$. Update $\delta^{(k+1)}$ according to (43).
- Fix $\delta^{(k+1)}$. Update $\mathbf{r}_x^{(k+1)}$ by solving the QP problem (45).
- Compute cost value $v^{(k+1)} = \|\tilde{\mathbf{R}}_x^{(k+1)} \tilde{\alpha}^{(k+1)} - \tilde{\mathbf{c}}^{(k+1)}\|_2^2$
- Set $k := k + 1$.

Until Improvement in cost function $<$ *tolerance factor* in the last 10 iterations OR specified no. of iterations exceeded.

Proposition 3: Every limit point generated by Algorithm 3 is a stationary point of problem (30).

Proof: This follows from the conditions laid out in [18]. It has already been established that the feasible set of \mathbf{r}_x is compact, while $\delta \in \mathbb{C}^{p+q+1}$ can be made compact by the imposition of loose upper and lower bounds on its elements, which is implicitly accomplished via norm regularization. ■

2) *ML Formulation*: A 2-block BCD algorithm can also be derived for the ML formulation (32).

- *Initialization*: The algorithm is initialized with an initial ML estimate of \mathbf{r}_x obtained by solving an instance of (38).
- $\{\alpha, \mathbf{c}\}$ *Update*: The update for $\delta = [\alpha^T, \mathbf{c}^T]^T$ is exactly the same as in the error free case.
- \mathbf{r}_x *Update*: Given an estimate of $\delta = [\alpha^T, \mathbf{c}^T]^T$, the update for \mathbf{r}_x is given by

$$\min_{\mathbf{r}_x} -\text{LLN}(\mathbf{r}_x) + \|\mathbf{\Pi} \mathbf{r}_x - \tilde{\mathbf{c}}\|_2^2 \quad (46a)$$

$$\text{s.t. } \mathbf{F} \mathbf{r}_x \geq \mathbf{0}, \quad (46b)$$

$$\mathbf{r}_x \in \mathcal{P} \quad (46c)$$

which is a convex optimization problem in \mathbf{r}_x .

The overall algorithm can be summarized as

Algorithm 4: 2-block BCD for ML ARMA Estimation.

Initialization: Solve the problem (38) to obtain a ML estimate of the truncated K -lag autocorrelation vector $\mathbf{r}_x^{(0)}$. Set $k := 0$.

Repeat

- Fix $\mathbf{r}_x^{(k)}$. Update $\delta^{(k+1)}$ according to (43).
- Fix $\delta^{(k+1)}$. Update $\mathbf{r}_x^{(k+1)}$ by solving the problem (46).
- Compute cost value $v^{(k+1)} = -\text{LLN}(\mathbf{r}_x^{(k+1)}) + \|\tilde{\mathbf{R}}_x^{(k+1)} \tilde{\alpha}^{(k+1)} - \tilde{\mathbf{c}}^{(k+1)}\|_2^2$
- Set $k := k + 1$.

Until Improvement in cost function $<$ *tolerance factor* in the last 10 iterations OR specified no. of iterations exceeded.

Proposition 4: Every limit point generated by Algorithm 4 is a stationary point of problem (32).

Proof: When the elements of δ are loosely bounded, the proof is exactly the same as in the error free case. ■

3-block approach:

3) *Error Free Formulation:* Although $\{\alpha, \mathbf{c}\}$ is the maximal set for which problem (30) is convex, instead of defining the variables $\{\alpha, \mathbf{c}\}$ as a single block, we may instead opt to define each variable as a block, in which case we have 3 blocks of variables. The BCD algorithm in this case consists of the following steps.

- *Initialization:* We require initial estimates $\mathbf{r}_x^{(0)}, \alpha^{(0)}, \mathbf{c}^{(0)}$ of each of the variables in order to initialize the algorithm. An estimate of $\mathbf{r}_x^{(0)}$ can be obtained by solving the feasibility problem (33). In order to obtain estimates of $\alpha^{(0)}$ and $\mathbf{c}^{(0)}$, we refer to the Yule-Walker equations (28). From the last $K - 1 - q$ equations of (28), we have

$$\hat{\mathbf{R}}_x^{(0)} \alpha = \hat{\rho}^{(0)} \quad (47)$$

where $\hat{\rho} = -[r_x(q+1), \dots, r_x(K-1)]^T \in \mathbb{C}^{(K-q-1) \times 1}$, $\hat{\mathbf{R}}_x \in \mathbb{C}^{(K-q-1) \times p}$ is defined as

$$\hat{\mathbf{R}}_x := \begin{bmatrix} r_x(q) & r_x(q-1) & \cdots & r_x(q-p+1) \\ r_x(q+1) & r_x(q) & \cdots & r_x(q-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(K-2) & r_x(K-3) & \cdots & r_x(K-p-1) \end{bmatrix} \quad (48)$$

and the superscript denotes the dependence on $\mathbf{r}_x^{(0)}$. The initial estimate of α is then obtained from the LS solution of (48), which is given by

$$\alpha^{(0)} = \left(\hat{\mathbf{R}}_x^{(0)H} \hat{\mathbf{R}}_x^{(0)} + \epsilon \mathbf{I}_K \right)^{-1} \hat{\mathbf{R}}_x^{(0)H} \hat{\rho}^{(0)} \quad (49)$$

where $\epsilon > 0$ is a regularization parameter. Meanwhile, from the first $q+1$ equations of (28), we have

$$\mathbf{c}^{(0)} = \check{\mathbf{R}}_x^{(0)} \check{\alpha}^{(0)} \quad (50)$$

where $\check{\mathbf{R}}_x \in \mathbb{C}^{(q+1) \times (p+1)}$ corresponds to

$$\check{\mathbf{R}}_x := \begin{bmatrix} r_x(0) & r_x(-1) & \cdots & r_x(-p) \\ r_x(1) & r_x(0) & \cdots & r_x(-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_x(q) & r_x(q-1) & \cdots & r_x(q-p) \end{bmatrix} \quad (51)$$

and the superscript again denotes the dependence on $\mathbf{r}_x^{(0)}$ and $\alpha^{(0)}$.

- *Update Rule:* Once initial estimates of all variables are available, we update the variable blocks according to the *Maximum Block Improvement* (MBI) rule² proposed

²We choose this update rule over the well known *Gauss-Seidel* (GS) update rule [16, p. 268] since the convergence of the BCD method for non-convex problems using the latter rule can only be established under certain restrictive conditions (see [16], [19] and references therein). In contrast, the MBI update rule guarantees convergence under more relaxed conditions. Although convergence of BCD using the GS update rule for general non-convex problems can be established with a *proximal point modification* of the GS method [20], experimental results indicated the sensitivity of this modified method to the choice of the coefficients of the added proximal term. Hence, we omitted this method from our manuscript.

in [18]. According to this rule, at each iteration, we determine the conditionally optimal update for each block, but finally we only choose to update the block which results in the maximum reduction of the cost function. At iteration k , if $\mathbf{r}_x^{(k)}, \alpha^{(k)}$ and $\mathbf{c}^{(k)}$ represent the current block variables, then the conditionally optimal updates for each block can be determined as follows.

Fixing $\alpha^{(k)}$ and $\mathbf{c}^{(k)}$, the optimal update of \mathbf{r}_x , which we denote as \mathbf{r}_x^{opt} is given by the QP problem (45). For α and \mathbf{c} , the conditionally optimal updates are again obtained from the Yule-Walker (28), which as a function of α , can be expressed as

$$\check{\mathbf{R}}_x^{(k)} \alpha = \check{\mathbf{c}}^{(k)} \quad (52)$$

where $\check{\mathbf{c}} \in \mathbb{C}^K$ is obtained by subtracting the first column of $\check{\mathbf{R}}_x$ from $\check{\mathbf{c}}$ and $\check{\mathbf{R}}_x \in \mathbb{C}^{K \times p}$ is the matrix obtained by deleting the first column of $\check{\mathbf{R}}_x$. The conditionally optimal update of α is then given by the LS solution of (52)

$$\alpha^{opt} = \left(\check{\mathbf{R}}_x^{(k)H} \check{\mathbf{R}}_x^{(k)} + \epsilon \mathbf{I}_K \right)^{-1} \check{\mathbf{R}}_x^{(k)H} \check{\mathbf{c}}^{(k)} \quad (53)$$

where $\epsilon > 0$ is a regularization parameter and the superscript denotes the dependence on the respective block variables at the k th iteration. For \mathbf{c} , the conditionally optimal update is again given by

$$\mathbf{c}^{opt} = \check{\mathbf{R}}_x^{(k)} \check{\alpha}^{(k)} \quad (54)$$

where $\check{\mathbf{R}}_x$ was defined in the previous section, and the superscript k again indicates the dependence on the respective variable block at the k th iteration.

After determining the update for each block, it is checked to see which of the conditional updates achieves the maximum reduction in the cost function (30a), i.e., we evaluate the functions

$$f_1(\mathbf{r}_x^{opt}) = \left\| \mathbf{\Pi}^{(k)} \mathbf{r}_x^{opt} - \check{\mathbf{c}}^{(k)} \right\|_2^2 \quad (55a)$$

$$f_2(\alpha^{opt}) = \left\| \check{\mathbf{R}}_x^{(k)} \check{\alpha}^{opt} - \check{\mathbf{c}}^{(k)} \right\|_2^2 \quad (55b)$$

$$f_3(\mathbf{c}^{opt}) = \left\| \check{\mathbf{R}}_x^{(k)} \check{\alpha}^{(k)} - \check{\mathbf{c}}^{opt} \right\|_2^2 \quad (55c)$$

following which only the block that yields the minimum cost amongst the functions defined in (55) is chosen to be updated, while the other blocks remain unchanged. Thus, if we define $\mathbf{y}_1 := \mathbf{r}_x^{opt}, \mathbf{y}_2 := \alpha^{opt}, \mathbf{y}_3 := \mathbf{c}^{opt}$, the block which yields the maximum improvement is chosen as

$$i^{opt} = \arg \min_{1 \leq i \leq 3} \{f_i(\mathbf{y}_i)\} \quad (56)$$

and the update rule is given by

$$\mathbf{z}_{i^{opt}}^{(k+1)} = \mathbf{y}_{i^{opt}} \quad (57a)$$

$$\mathbf{z}_i^{(k+1)} = \mathbf{z}_i^{(k)}, \forall i \in \{1, 2, 3\} \setminus i^{opt} \quad (57b)$$

where $\mathbf{z}_1^{(k)} := \mathbf{r}_x^{(k)}, \mathbf{z}_2^{(k)} := \alpha^{(k)}$ and $\mathbf{z}_3^{(k)} := \mathbf{c}^{(k)}$.

Overall, the algorithm can be summarized as

It is obvious that the algorithm produces a monotonically decreasing cost sequence. In addition, the following holds.

Algorithm 5: 3-block BCD with MBI for ARMA Estimation.

Initialization: Obtain initial estimates $\mathbf{r}_x^{(0)}$, $\boldsymbol{\alpha}^{(0)}$, $\mathbf{c}^{(0)}$ according to (33), (49) and (50) respectively. Compute initial cost value $v^{(0)} = \|\tilde{\mathbf{R}}_x^{(0)}\tilde{\boldsymbol{\alpha}}^{(0)} - \tilde{\mathbf{c}}^{(0)}\|_2^2$. Set $k := 0$.

Repeat

- Fix $\boldsymbol{\alpha}^{(k)}$ and $\mathbf{c}^{(k)}$. Determine \mathbf{r}_x^{opt} according to (45).
- Fix $\mathbf{r}_x^{(k)}$ and $\mathbf{c}^{(k)}$. Determine $\boldsymbol{\alpha}^{opt}$ according to (53).
- Fix $\mathbf{r}_x^{(k)}$ and $\boldsymbol{\alpha}^{(k)}$. Determine \mathbf{c}^{opt} according to (54).
- Choose the block which yields maximum improvement according to (56).
- Update the variables according to (57).
- Set $v^{(k+1)} = f(\mathbf{y}_{i^{opt}})$
- Set $k := k + 1$.

Until Improvement in cost function $<$ *tolerance factor* in the last 10 iterations OR specified no. of iterations exceeded

Proposition 5: If we impose loose upper and lower bounds on the elements of $\boldsymbol{\alpha}$ and \mathbf{c} , then every limit point of Algorithm 5 is a stationary point of problem (30).

Proof: This follows from the conditions in [18] being satisfied. ■

Remark 1: The proof of convergence of the MBI method does not require the subproblem associated with each block variable to have a unique solution; a condition which is necessary for the convergence of BCD with three or more blocks of variables using the GS update rule. The downside is the higher computational cost incurred in solving all subproblems in each iteration in order to determine the best block to update.

4) *ML Formulation:* The BCD algorithm with MBI update rule for the ML formulation is similar to the error free case, and proceeds as follows.

- 1) *Initialization:* An initial ML estimate of $\mathbf{r}_x^{(0)}$ can be obtained by solving (38), while initial estimates of $\boldsymbol{\alpha}^{(0)}$ and $\mathbf{c}^{(0)}$ are obtained from (49) and (50) respectively.
- 2) *Update Rule:* If $\mathbf{r}_x^{(k)}$, $\boldsymbol{\alpha}^{(k)}$ and $\mathbf{c}^{(k)}$ represent the iterates at the k th iteration, then \mathbf{r}_x^{opt} (the conditional update of $\mathbf{r}_x^{(k)}$) is obtained by fixing $\boldsymbol{\alpha}^{(k)}$ and $\mathbf{c}^{(k)}$ and solving the problem (46). The updates of $\boldsymbol{\alpha}^{(k)}$ and $\mathbf{c}^{(k)}$ ($\boldsymbol{\alpha}^{opt}$ and \mathbf{c}^{opt} respectively) are the same as (53) and (54). Following the updates, we evaluate the functions 1

$$f_1(\mathbf{r}_x^{opt}) = -\text{LLN}(\mathbf{r}_x^{opt}) + \left\| \boldsymbol{\Pi}^{(k)} \mathbf{r}_x^{opt} - \tilde{\mathbf{c}}^{(k)} \right\|_2^2 \quad (58a)$$

$$f_2(\boldsymbol{\alpha}^{opt}) = -\text{LLN}(\mathbf{r}_x^{(k)}) + \left\| \tilde{\mathbf{R}}_x^{(k)} \tilde{\boldsymbol{\alpha}}^{opt} - \tilde{\mathbf{c}}^{(k)} \right\|_2^2 \quad (58b)$$

$$f_3(\mathbf{c}^{opt}) = -\text{LLN}(\mathbf{r}_x^{(k)}) + \left\| \tilde{\mathbf{R}}_x^{(k)} \tilde{\boldsymbol{\alpha}}^{(k)} - \tilde{\mathbf{c}}^{opt} \right\|_2^2 \quad (58c)$$

and choose the block which yields the maximum improvement in cost according to (56). The update rule is defined in (57).

The overall algorithm can be summarized as

Algorithm 6: 3-block BCD with MBI for ML ARMA Estimation.

Initialization: Obtain initial estimates $\mathbf{r}_x^{(0)}$, $\boldsymbol{\alpha}^{(0)}$, $\mathbf{c}^{(0)}$ according to (38), (49) and (50) respectively. Compute initial cost value $v^{(0)} = -\text{LLN}(\mathbf{r}_x^{(0)}) +$

$\|\tilde{\mathbf{R}}_x^{(0)}\tilde{\boldsymbol{\alpha}}^{(0)} - \tilde{\mathbf{c}}^{(0)}\|_2^2$. Set $k := 0$.

Repeat

- Fix $\boldsymbol{\alpha}^{(k)}$ and $\mathbf{c}^{(k)}$. Determine \mathbf{r}_x^{opt} according to (46).
- Fix $\mathbf{r}_x^{(k)}$ and $\mathbf{c}^{(k)}$. Determine $\boldsymbol{\alpha}^{opt}$ according to (53).
- Fix $\mathbf{r}_x^{(k)}$ and $\boldsymbol{\alpha}^{(k)}$. Determine \mathbf{c}^{opt} according to (54).
- Choose the block which yields maximum improvement according to (56).
- Update the variables according to (57).
- Set $v^{(k+1)} = f(\mathbf{y}_{i^{opt}})$
- Set $k := k + 1$.

Until Improvement in cost function $<$ *tolerance factor* in the last 10 iterations OR specified no. of iterations exceeded

Proposition 6: If we impose loose upper and lower bounds on the elements of $\boldsymbol{\alpha}$ and \mathbf{c} , then every limit point of Algorithm 6 is a stationary point of problem (32).

Proof: The proof is exactly the same as in the error free case. ■

Remark 2: Note that there is no guarantee that the AR parameter estimates obtained will be minimum-phase, since it is known [10] that solving an overdetermined system of equations to determine AR parameters does not generally yield a minimum-phase solution. Nevertheless, our aim here is to estimate the power spectrum; we use the ARMA parametrization only as an intermediate vehicle.

VI. SIMULATION RESULTS

A. AR Models

We first present simulations for the AR case with accurate power measurements. The natural benchmark here is a two-step approach: solving the nonparametric problem in (33), followed by one-time fitting of the AR model parameters according to (35). The LP feasibility problem (33) and the QP problem (35) were modeled using YALMIP [22] and solved using the solver SeDuMi [23]. In all experiments, the maximum iteration counter of the BCD algorithm was set to 60 iterations (each consisting of 2 alternating updates) and the exit tolerance was set to 10^{-5} . We obtained PS estimates from the output of the BCD algorithm by taking the DFT of the autocorrelation estimates and also by plugging the model parameter estimates into (4). We used the Normalized Mean Square Error (NMSE) as a performance criterion, which is defined as $\text{NMSE} = \mathbb{E} \left[\frac{\|S_x - \hat{S}_x\|_2^2}{\|S_x\|_2^2} \right]$ where S_x is the true PS and \hat{S}_x is the estimated PS, with both spectra normalized by their peak values. The expectation is taken with respect to the randomness of the signal and the broadband filters.

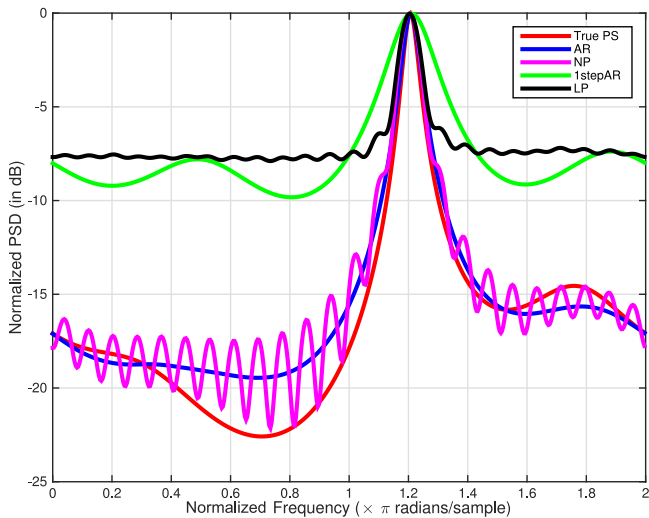


Fig. 2. Mean Normalized spectra for a complex AR(3) model.

We now present an illustrative example showcasing the effectiveness of our approach. A complex AR(3) model was used to generate a WSS stochastic process, and a sensing scenario was considered with $M = 100$ sensors, $K = 25$ and threshold t empirically tuned to select 40 sensors to transmit $b_m = 1$ to the FC, which is assumed to know the true model order in this experiment. Both methods were initialized from the same instance of the LP feasibility problem. The results obtained by averaging over 400 Monte-Carlo trials are depicted in Fig. 2, with each PS estimate normalized by its peak value. The non-parametric LP initial estimate (black), obtained by taking the N_F point DFT correctly estimates the peak of the true AR PS, but the subsequent AR fitting procedure (green) does not improve the estimation performance (in fact the spectral lobe widens). The parametric AR estimate (blue), obtained from the BCD algorithm, exhibits the best estimation performance. Note that the quality of the spectral estimate is very satisfactory considering that only 100 bits (roughly equivalent to 3 floats in IEEE 32-bit precision standard) were used at the FC. Hence, even though the BCD algorithm is incapable of solving (25) exactly, it can generate high quality approximate solutions. Another option for obtaining the final PS estimate, by taking the DFT of the autocorrelation vector returned by BCD, is depicted in magenta. The presence of ripples is due to the fact that we only estimate a finite window of the autocorrelation sequence, which degrades the quality of the spectral estimate.

A more generic quantitative comparison is presented in Fig. 3, where we considered a sensing scenario with $M = 100$ sensors, set $K = 50$ and tuned the threshold t to vary the number of sensors reporting above threshold. The spectral NMSE was computed for 30 randomly drawn complex AR(5) models, with the NMSE for each value of t being averaged out over 200 Monte-Carlo trials for each AR model. Prior knowledge of the true model order was again assumed. Both methods were again initialized from the solution of the same instance of the LP feasibility problem. Again, the superior performance of the BCD algorithm is noted, with the parametric AR estimate (blue) ex-

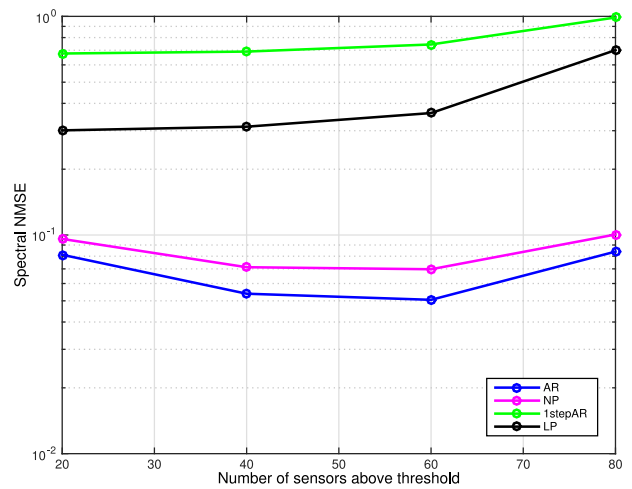


Fig. 3. NMSE vs No. of sensors above threshold for complex AR(5) models with 100 sensors.

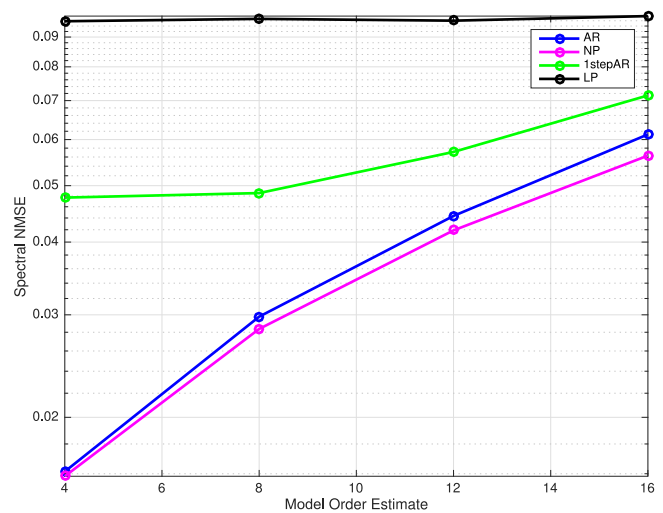


Fig. 4. NMSE vs Model order estimate for complex AR(4) models.

hibiting lowest spectral NMSE. The one step AR fitting method is much worse-off in comparison. Extensive simulations across a range of model orders and sensing scenarios revealed that the BCD algorithm always delivers the best performance, thus providing supporting evidence of its approximation quality.

In many practical scenarios however, the true model order of the underlying AR process is unknown, and one must typically use an estimate of the true model order. Assuming that an upper bound on the true model order is available, we carried out an experiment where we randomly generated 30 AR(4) models and considered a scenario with $M = 200$ sensors, $K = 30$ and selected the threshold to make 40 sensors report above threshold. The spectral NMSE was computed as a function of the model-order overestimate for each AR model over 200 Monte-Carlo trials, with the final result being averaged out across all models and is depicted in Fig. 4. From the figure it is observed that the NMSE obtained from the PS estimates of the BCD algorithm increases very gracefully (notice the logarithmic scaling of the y-axis), even when the upper bound on the true model order

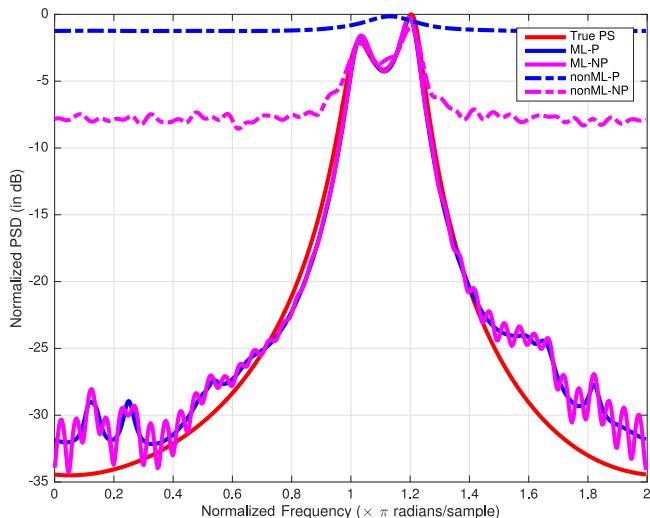


Fig. 5. Mean Normalized Spectra for a complex AR(3) model in the presence of bit errors.

is made very loose, thus demonstrating the robustness of our proposed algorithm to model order over-estimation.

Finally, we considered the case when sample averaging at each sensor is not sufficient to allow accurate estimation of ρ_m , which results in errors in the power estimates prior to thresholding. We considered a sensing scenario for a complex AR(3) model with $M = 200$, $K = 40$ and set the number of sensors above threshold to 80. We used the ML formulation which explicitly models the errors and compared it against the ‘naive’ error-free formulation for solving the PS reconstruction problem. The ML formulation was solved using CVX [24] in MATLAB. We set the penalty parameter $\lambda = 1000$ in order to impose AR structure. The PS estimates averaged over 400 Monte-Carlo trials are depicted in Fig. 5. The random errors in the power measurements resulted in 17% of the power measurement bits being flipped on average. It is evident that the ML PS estimates (solid lines) are of much better quality than their non-ML counterparts (dashed lines). This demonstrates that the ML estimates are robust to bit flips, in contrast to the estimates obtained from the original formulation, which does not take into account the measurement errors.

B. ARMA Models

We now present simulations for the ARMA case. All BCD algorithms for ARMA estimation were run for a maximum of 120 iterations, with an exit tolerance of 10^{-5} . For the ML formulations, we used CVX for solving the optimization subproblems, whereas we used YALMIP for the non-ML formulations, with SeDuMi as the solver. A 3-block BCD algorithm with a Gauss-Seidel update rule was also included in our comparison, in addition to the other two algorithms.

We first illustrate the performance of BCD in the ARMA case with accurate power measurements. A complex ARMA (2, 2) process was generated, and a sensing scenario was considered with $M = 200$ sensors, $K = 20$ and a single threshold t , resulting in 80 sensors reporting above threshold. Knowledge

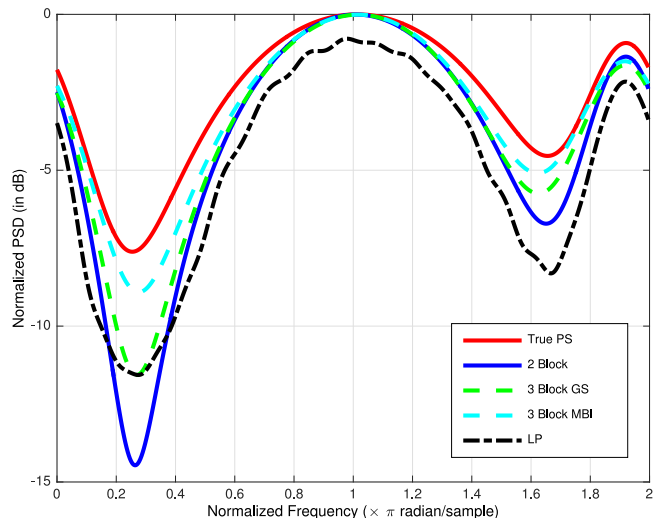


Fig. 6. Mean Normalized Spectra for a complex ARMA(2, 2) model.

of the true model orders of the AR and MA components was assumed at the FC. The results were averaged over 400 Monte-Carlo trials, and are depicted in Fig. 6, with all spectra being normalized by their peak values (before averaging—for the LP method, there is considerable variance in the peak location, which reduces the peak value after averaging). We only plotted the parametric PS estimates (i.e., obtained from estimates of α and c) in this case. The estimation results are again very satisfactory, considering the fact that the data record (i.e., the number of bits) is roughly equivalent to 3 IEEE double precision floats. The BCD algorithms outperform the non-parametric LP method and also exhibit superior performance compared to fitting model parameters from the LP estimate. From the figure, it is observed that the MBI method demonstrates the best performance, however it is the most computationally intensive amongst the proposed BCD algorithms. Another observation is that the 3 block BCD method with GS update rule demonstrates competitive performance in practice compared to the other methods.

For a more generic quantitative experiment, we randomly generated 30 complex ARMA(3, 3) models, considered a sensing scenario with $M = 200$ sensors, $K = 25$, and varied the threshold t in order to change the number of sensors reporting above threshold. Knowledge of the true model order was again assumed. The spectral NMSE as a function of the number of sensors reporting above threshold is depicted in Fig. 7, with each point corresponding to 200 Monte-Carlo trials per ARMA model, averaged out over all models. It is observed that the estimation performance is very poor when the threshold is set high (i.e., few sensors report above threshold), but gradually improves as the threshold is set lower. Initially, the BCD algorithms do not offer marked improvement over the non-parametric LP estimate, but as the threshold is decreased the gap in estimation performance between the LP estimate and the proposed algorithms increases. Below a certain threshold level, the BCD algorithms produce substantial improvement in estimation performance over the initial LP estimate. The 2-block BCD

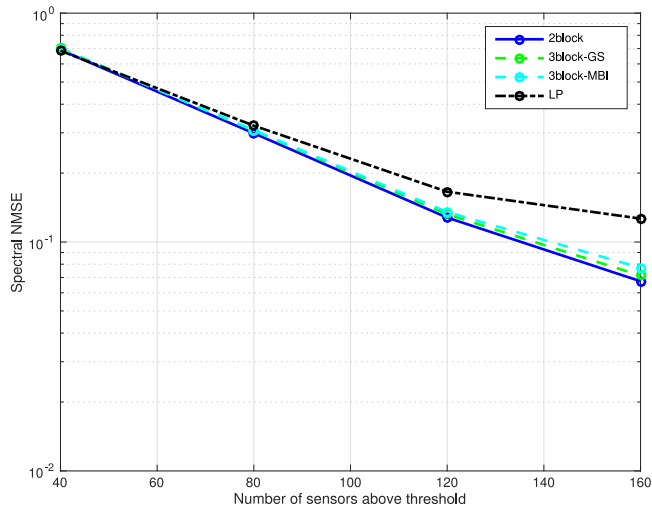


Fig. 7. NMSE vs No. of sensors above threshold for complex ARMA(3, 3) models with 200 sensors.

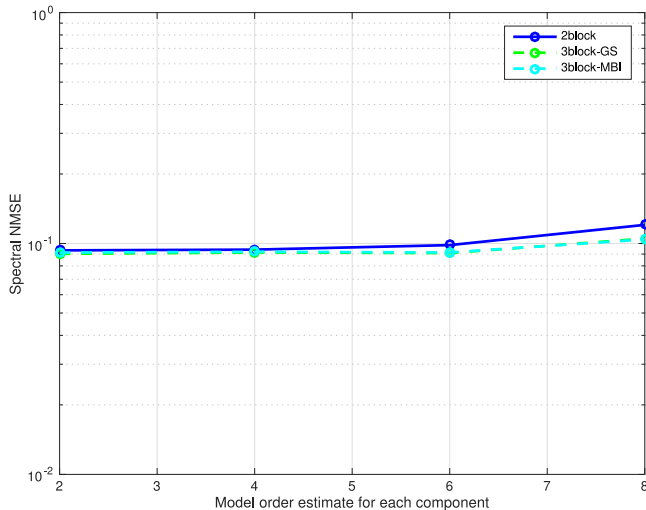


Fig. 8. NMSE vs Model order overestimation for complex ARMA(2, 2) models.

algorithm demonstrated the best overall performance and was also the quickest to achieve convergence amongst the proposed algorithms.

We also considered the case when the true model order of the AR and MA components is not known at the FC; instead only an upper bound on the order of each component is available. In order to investigate the performance of our proposed algorithms, we randomly generated 30 ARMA(2, 2) models, and designed a sensing scenario with $M = 200$ sensors, $K = 30$ and threshold selected to make 160 sensors report above threshold. The spectral NMSE as a function of the order overestimate in each component is depicted in Fig. 8. Each point was obtained by Monte-Carlo averaging over 200 trials per model, followed by averaging across all models. Similar to the AR case, it was observed that the algorithms are robust to model order overestimation, with the NMSE not being appreciably affected

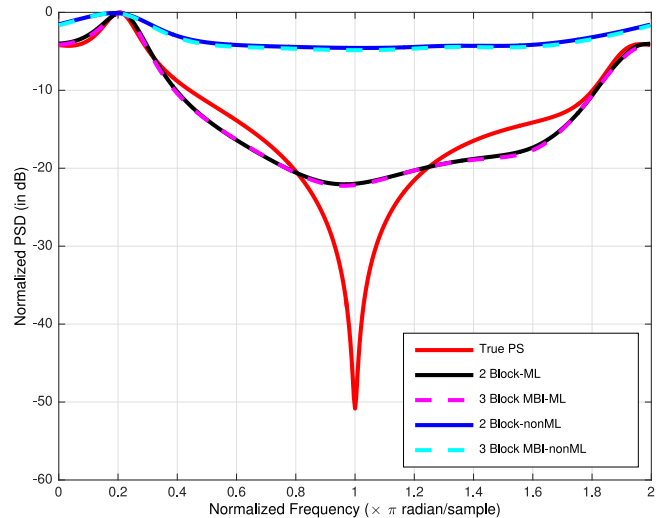


Fig. 9. Mean Normalized Spectra for a complex ARMA(2, 3) model in the presence of bit errors.

as the upper bound on the order of each component is made loose.

Finally, when insufficient sample averaging at each sensor results in random bit-flips, we tested the performance of the ML formulations against the non-ML formulations for a randomly generated complex ARMA(2, 3) model in a scenario comprising $M = 200$ sensors, $K = 20$ and 80 sensors above threshold, with prior knowledge of the true model order. The PS estimates obtained over 400 Monte-Carlo trials are depicted in Fig. 9. The errors in the soft power measurements produced approximately 28 flipped bits on average. We omitted the 3-block GS method in this comparison. From the figure, it is apparent that the ML formulations perform considerably better compared to their non-ML counterparts. Although it may appear from the figure that the quality of the ML PS estimates obtained is not very satisfactory, note that the NMSE is still small. This is because the notch is very sharp, and the error there is under -20 dB, so the integral effect is negligible in terms of NMSE.

VII. CONCLUSION

We considered a network sensing scenario, consisting of scattered low-end sensors transmitting randomly filtered, sample averaged, one-bit quantized power measurements to a FC. Assuming that the underlying WSS process admits an AR or ARMA parametrization, we considered corresponding parametric power spectrum sensing from such binary power measurements. First the case where sensor sample averaging is sufficient to provide accurate bit measurements was considered, and the AR and ARMA power spectrum sensing problems were formulated as block-separable optimization problems. Their block-separable structure was exploited to develop different BCD algorithms with good convergence properties, considering the inherent computational difficulty of the problems considered. Next, the problem was revisited from a statistical estimation viewpoint. Relaxing the sample averaging requirement at each sensor, the errors in the noisy soft power estimates were modeled

as being Gaussian, which was utilized to propose ML formulations for AR and ARMA PS estimation, which are also block-separable. The block separable structure of these problems was again exploited to develop BCD algorithms with good convergence properties. Simulations demonstrated the high quality of the approximate solutions generated by the BCD algorithms under various scenarios, including cases where only an upper bound on the true model order is known. In the presence of bit flips introduced by quantization of noisy soft power measurements, the ML formulations outperformed their non-ML counterparts and demonstrated considerable robustness.

REFERENCES

- [1] A. Konar and N. D. Sidiropoulos, "Parametric frugal sensing of autoregressive power spectra," *Proc. IEEE ICASSP*, Shanghai, Mar. 2016.
- [2] D. D. Ariananda and G. Leus, "Compressive wideband power spectrum estimation," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4775–4789, Sep. 2012.
- [3] D. Romero and G. Leus, "Wideband spectrum sensing from compressed measurements using spectral prior information," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6232–6246, Dec. 2013.
- [4] O. Mehanna and N. D. Sidiropoulos, "Frugal sensing: Wideband power spectrum sensing from few bits," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2693–2703, May 2013.
- [5] O. Mehanna and N. D. Sidiropoulos, "Maximum likelihood passive and active sensing of wideband power spectra from few bits," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1391–1403, Mar. 2015.
- [6] A. Konar, N. D. Sidiropoulos, and O. Mehanna, "Parametric frugal sensing of power spectra for moving average models," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1073–1085, Mar. 2015.
- [7] A. Konar and N. D. Sidiropoulos, "Hidden convexity in QCQP with Toeplitz-Hermitian quadratics," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1623–1627, Oct. 2015.
- [8] A. Konar and N. D. Sidiropoulos, "Distributed compression and maximum likelihood reconstruction of finite autocorrelation sequences," presented at the Asilomar Conf. Signals, Syst., Comput., Pacific Grove, Nov. 2015.
- [9] S. M. Kay and S. L. Marple Jr., "Spectrum analysis—A modern perspective," *Proc. IEEE*, vol. 61, no. 11, pp. 1380–1419, Nov. 1981.
- [10] P. Stoica and R. L. Moses, *Spectral Analysis of Signals*. Upper Saddle River, NJ, USA: Prentice-Hall, 2005.
- [11] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. New York, NY, USA: Wiley, 1996.
- [12] S.-P. Wu, S. Boyd, and L. Vandenberghe, "FIR filter design via semidefinite programming and spectral factorization," in *Proc. IEEE Conf. Decision Control*, 1996, vol. 1, pp. 271–276.
- [13] R. Gray and L. Davidson, *Random Processes: A Mathematical Approach for Engineers*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1986.
- [14] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [16] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.
- [17] S. J. Wright, "Coordinate descent algorithms," *Math. Programm.*, vol. 151, no. 1, pp. 3–34, Jun. 2015.
- [18] B. Chen, S. He, Z. Li, and S. Zhang, "Maximum block improvement and polynomial optimization," *SIAM J. Optim.*, vol. 22, no. 1, pp. 87–107, 2012.
- [19] P. Tseng, "Convergence of a block coordinate descent method for non-differentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.
- [20] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss-Seidel method under convex constraints," *Oper. Res. Lett.*, vol. 26, no. 3, pp. 127–136, Apr. 2000.
- [21] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York, NY, USA: McGraw-Hill, 1976.
- [22] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," presented at the CACSD, Taipei, Sep. 4, 2004.
- [23] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods Softw.*, vol. 11, no. 1, pp. 625–653, 1999.
- [24] M. Grant and S. Boyd, CVX: Matlab software for disciplined convex programming (web page and software), Jun. 2009. [Online]. Available: <http://stanford.edu/Boyd/cvx>



Aritra Konar (S'14) received the B.Tech. degree in electronics and communications engineering from West Bengal University of Technology, West Bengal, India, in 2011, and the M.S. degree in electrical engineering from University of Minnesota, Minneapolis, USA, in 2014. Since then he has been working towards his Ph.D. degree with the Department of Electrical and Computer Engineering, University of Minnesota. His research interests include statistical signal processing, convex optimization and data analytics.



Nicholas D. Sidiropoulos (F'09) received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Greece, and M.S. and Ph.D. degrees in electrical engineering from the University of Maryland—College Park, in 1988, 1990 and 1992, respectively. He served as Assistant Professor at the University of Virginia (1997–1999); Associate Professor at the University of Minnesota—Minneapolis (2000–2002); Professor at the Technical University of Crete, Greece (2002–2011); and Professor at the University of Minnesota—Minneapolis (2011–). His current research focuses primarily on signal and tensor analytics, with applications in cognitive radio, big data, and preference measurement. He received the NSF/CAREER award (1998), the IEEE Signal Processing Society (SPS) Best Paper Award (2001, 2007, 2011), and the IEEE SPS Meritorious Service Award (2010). He has served as IEEE SPS Distinguished Lecturer (2008–2009), and Chair of the IEEE Signal Processing for Communications and Networking Technical Committee (2007–2008). He received the Distinguished Alumni Award of the Department of Electrical and Computer Engineering, University of Maryland, College Park in 2013, and was elected EURASIP Fellow in 2014.