Synchronization and packet separation in wireless ad hoc networks by Known Modulus Algorithms

Relja Djapic, Alle-Jan van der Veen and Lang Tong

Submitted to JSAC Special Issue on Wireless Ad Hoc Networks
(topic: signal processing issues in ad hoc networks)

August 9, 2004

Abstract—**In mobile asynchronous ad hoc networks (MANETs), multiple users may transmit packets at the same time. If a collision occurs, then in current systems both packets are lost and need to be retransmitted, reducing the overall throughput. To mitigate this, we consider to extend the receiver with a small antenna array, so that it can suppress interfering signals. To characterize the signal of interest, we propose to modulate it at the symbol rate by a known amplitude variation. This allows the corresponding multichannel receiver to estimate the beamformer weights that will suppress the interfering sources. We introduce "known modulus algorithms" (KMAs) to achieve this. We also derive synchronization algorithms to estimate the offset of the desired packet in an observation window, among interfering data packets. The algorithms are illustrated via simulations.**

I. INTRODUCTION

A key limiting factor on the throughput of wireless networks is packet collisions among uncoordinated transmitters. Conventionally, medium access control (MAC) protocols are used to schedule transmissions either in a deterministic fashion (e.g., TDMA, FDMA or CDMA) or by random access protocols such as Aloha and CSMA. For ad hoc networks, however, the absence of base stations and the necessity of distributed medium access control requires some form of random access, and avoiding collisions is difficult. Even more challenging is the so-called hidden/exposed terminal problem that severely limits the effectiveness of techniques based on carrier sensing. Although the use of CTS-RTS exchange along with busy-tone [3] can eliminate collisions [4], such protocols are vulnerable to interference from other services.

Recent advances in antenna array processing and space-time coding challenge the fundamental premise of the classical approach to MAC that prohibits the simultaneous transmission of different users. Specifically, various algorithms have been developed in the past decade that allow the separation of multiple signals, sometimes even without prior knowledge of the propagation channel (blind detection, e.g., [5]). The new possibilities call for new approaches in MAC protocols that exploit the ability to separate colliding packets by signal processing [6], by so-called cross-layer design.

R. Djapic and A.J. van der Veen are with Delft University of Technology, Department of Electrical Engineering, Mekelweg 4, 2628 CD Delft, The Netherlands, email: {relja,allejan}@cas.et.tudelft.nl. L. Tong is with Cornell University, Department of Electrical Engineering, 326 Frank H.T. Rhodes Hall, Ithaca, NY 14853, USA, email ltong@ee.cornell.edu. This research was supported in part by the Dutch Min. Econ. Affairs/Min. Education Freeband-impulse project DTC.5961 *Airlink*, and by NWO-STW under the VICI programme (DTC.5893). Parts of this paper were published in [1], [2] .

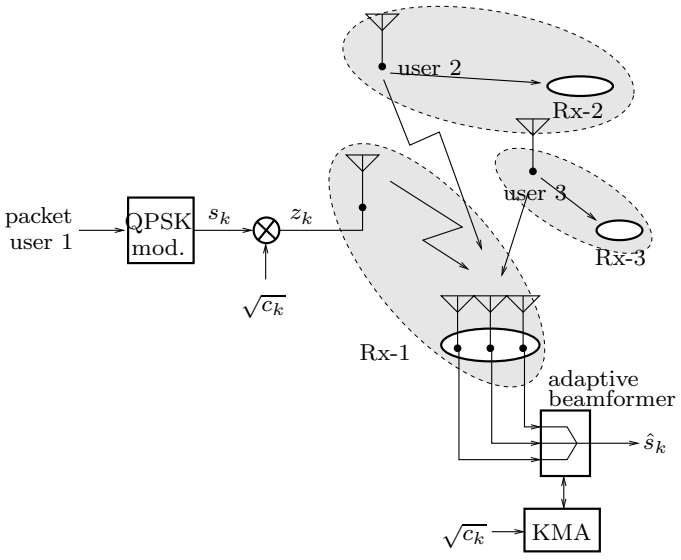

Fig. 1. Wireless ad hoc communication scenario. $\sqrt{c_k}$ is a known modulus variation used to recognize user-1.

Signal separation was first applied to the design of MAC protocols in [7] where an N-fold collision is resolved by a special retransmission protocol. This technique is only applicable in cellular networks. In [8], the problem of packet separation is formulated as one of signal separation in a MIMO system. While this technique is applicable in ad hoc networks, it is restricted to a slot-synchronized network, which means that the network cannot cover a large area and all nodes in the network must be synchronized to the slot structure.

In this paper, we present a technique that allows packet separation in asynchronous ad hoc networks. As illustrated in figure 1, the user of interest transmits a constant modulus signal multiplied by an amplitude modulating code known at the receiver. This unique "color code" allows the antenna array at the receiver to detect and filter out the desired user among the other interfering signals that may or may not have a similar structure. The modulation code can be a random binary sequence determined either by the transmitter or the receiver, or it can be a common pseudo-random long code with different offsets for different users. The separating beamformer is computed using

one of the known modulus algorithms (KMAs) developed in this paper.

The idea of modulus variations to assist capture of the desired user goes back to Treichler and Larimore [9], who also derived the first (iterative) KMA. More recently, the idea was picked up again in [10], [11] for the purpose of multi-user interference cancellation, and, independently, by us in [1], [2] for the separation of finite duration packets.

In general, KMA requires neither slot synchronization nor any coordination among transmitters, which makes its application in an uncontrolled environment such as wireless LAN (WLAN) or mobile ad hoc networks (MANETs) particularly attractive. In the context of WLANs, it is interesting to note that the required amplitude modulations can be so small that they are not perceived by legacy receivers, making the system upward compatible.

From a source separation point of view, amplitude modulation is just one way to mark users of interest, and several other techniques could play a role. Spread spectrum techniques such as CDMA would be possible but reduce the data rates. User-specific training sequences have disadvantages in asynchronous systems. For instantaneous channels, general blind techniques such as ILSP [12] and ACMA [13] are applicable, but not efficient since we are interested in only one user. Several modulation approaches have been proposed in the literature. Stochastic techniques such as "transmitter induced cyclostationarity", initially derived for single user blind equalization [14], [15], [16] have recently been extended to multi-user convolutive channels [17] and OFDM [18]. A deterministic version of such a technique, using phase modulation codes, was proposed for source separation in MANETs [19] (this paper relates to several ideas proposed in [1] and also offers a throughput analysis which, therefore, we omit here). Another example of a deterministic source separation technique is [20], but it needs multiple transmit antennas per source (spatial redundancy).

Our objective here is to derive a system that is simpler than ACMA and other blind techniques, does not reduce the capacity, and only finds the desired user. We consider a scenario where users can transmit packages at arbitrary moments, and assume that the receiver knows the modulation code and packet length of the user of interest. We collect a block of data from an observation window, and aim to detect the presence of a packet from the user of interest, estimate its offset within the window, and estimate the beamformer by which this packet can be received while suppressing the interfering packets. Some of these results were presented by us at conferences [1], [2]; the present paper offers a more in-depth discussion.

The structure of the paper is as follows. In section II, we introduce the communication scenario and resulting data model. Section III derives the basic KMA algorithm, assuming synchronization is available, and corresponding simulations. Subsequently, in section IV, we present algorithms for estimating the packet offset, and corresponding simulations.

II. DATA MODEL

We assume the situation in figure 1 where several users occupy a common wireless channel. For simplicity, the channel is assumed to be narrowband. The potential number of users is unlimited, but the offered network load is fixed. User 1 is the desired user, it is supposed to be received by receiver 1, but there will be interference from the other users. To suppress the interference, the receiver is equipped with an antenna array of M elements.

The transmission is modeled by a linear data model of the

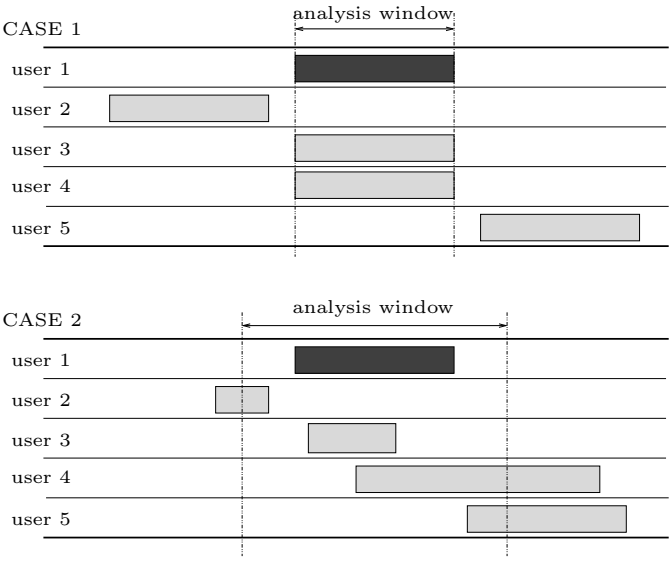

Fig. 2. Slot structure

form

$$\mathbf{x}_k = \sum_{q=1}^{\infty} \mathbf{a}_q s_k^{(q)} \; + \; \mathbf{n}_k \,, \tag{1}$$

where $\mathbf{x}_k \in \mathbb{C}^M$ is the data vector received by the array of M antennas at time k, $\mathbf{a}_q$ is the signature vector of source q and $s_k^{(q)} \in \mathbb{C}$ its transmitted symbol at time k, and $\mathbf{n}_k \in \mathbb{C}^M$ an additive noise vector. In this model, as a useful abstraction each source is assumed to transmit only once a data packet, and for the rest to be silent. Hence each $s^{(q)}$ has finite support. A physical user with several data packets counts as several independent sources, each with independent **a**-vectors, hence the model allows for a slowly changing (fading) channel.

The modulation of source 1 is assumed to be constant modulus (e.g., QPSK), i.e., $|s_k^{(1)}|^2 = 1$. The modulation of the other users is arbitrary.

We will consider two types of transmission scenarios (see figure 2):

1. *Slotted, with fixed slot length L.* The situation in a slot is stationary: the number of active users is constant inside a slot, and their spatial signature vectors are constant.

2. *Unslotted, with fixed or variable packet lengths.* Packets can have arbitrary starting times, hence the number of active users changes throughout the slot. The packet length of user 1 is denoted by L.

Initially, we assume that we are synchronized to the user of interest: the start time and length of his packet is known. We collect N samples in a data matrix $\mathbf{X} = [\mathbf{x}_1 \,, \cdots \,, \mathbf{x}_N] : M \times N$. In case 1, we take $N = L$ and $\mathbf{x}_1$ contains the first sample of the packet. In case 2, we take a slightly larger analysis window to avoid certain edge effects, $N \geq L$ samples. In section IV, we consider the estimation of the packet offset.

Let d be the maximal number of active users in the analysis window, and assume for notational simplicity that these are users 1 to d. Defining $\mathbf{A} = [\mathbf{a}_1 \,, \cdots \,, \mathbf{a}_d] : M \times d$, $\mathbf{S} = [s_{qk}] = [s_k^{(q)}] : d \times N$ and $\mathbf{N} = [\mathbf{n}_1 \,, \cdots \,, \mathbf{n}_N] : M \times N$, we obtain

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{N} \,. \tag{2}$$

$\mathbf{A}$, $\mathbf{S}$ and $\mathbf{N}$ are unknown. The objective is to reconstruct the nonzero part of $\mathbf{s}^{(1)} = [s_1^{(1)}, \cdots, s_N^{(1)}]$ using linear beamforming,

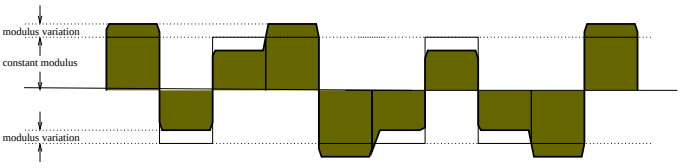

Fig. 3. Constant modulus signal with coded amplitude variations which are used to identify the user of interest

i.e., to find a beamformer $\mathbf{w}$ such that $\hat{s}_k = \mathbf{w}^{\mathrm{H}}\mathbf{x}_k$ approximates $s_k^{(1)},\ k = 1, \cdots, N$. Here $^{\mathrm{H}}$ denotes the Hermitian transpose.

Several algorithms for source separation are applicable at this point (e.g., CMAs), but they all have the problem that they cannot distinguish one user from another. To distinguish the desired source, we give it a "color code", in the form of a known pseudo-random modulus variation (figure 3). Instead of transmitting s_k, we transmit $z_k = s_k\sqrt{c_k}$, where $c_k = 1 \pm \epsilon$ is a real and positive scaling that induces a small modulus variation, without changing the average transmission power. For notational convenience, we assume that $c_k = 0$ outside the support of the packet. The data model (2) is replaced by

$$\mathbf{X} = \mathbf{A}\mathbf{Z} + \mathbf{N}\,. \tag{3}$$

Recall that we assume that $|s_k|^2 = 1$, so that $|z_k|^2 = c_k$. Similar to CMA, the objective of the beamformer will be to recover z_k based on its modulus, i.e., such that

$$|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 = |\hat{z}_k|^2 \;=\; c_k\,, \qquad k = 1, \cdots, N\,.$$

With noise, we try to minimize the difference and can obviously recover the source only approximately.

III. KNOWN MODULUS ALGORITHMS

In this section we consider receiver algorithms, assuming the receiver is synchronized to the user of interest and knows his code.

III-A. Iterative solutions

The usual iterative CMA can easily be adapted for the present case: one only has to define the instantaneous modulus error as $e_k = |\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 - c_k$. This leads to the updating step

$$\mathbf{w} := \mathbf{w} - \mu(\mathbf{x}_k^{\mathrm{H}}\mathbf{w})\mathbf{x}_k(|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 - c_k)\,, \qquad k = 1, 2, \cdots, N\,.$$

This is the Known Modulus Algorithm (KMA) introduced by Treichler and Larimore in [9], and used more recently in [10], [11]. Apart from the usual stability and initialization issues, the resulting algorithm would not be very useful for the current purpose since we don't want to track the beamformer; we require a block solution where also the initial symbols are detected correctly. This is provided by an alternating projection algorithm: iterate until convergence

$$\left[\begin{array}{lcl} \mathbf{r} & := & \mathbf{w}^{\mathrm{H}}\mathbf{X} \\ \hat{z}_k & := & \frac{r_k}{|r_k|}\sqrt{c_k}\,, \qquad k = 1, \cdots, N \\ \mathbf{w} & := & (\hat{\mathbf{z}}\mathbf{X}^{\dagger})^{\mathrm{H}} \end{array}\right. \tag{4}$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse. Note that a candidate solution $\hat{\mathbf{z}}$ is alternatingly projected onto the row span of $\mathbf{X}$ (via the projection $\mathbf{X}^{\dagger}\mathbf{X}$), and entry-wise scaled to fit the modulus condition. With a sufficiently accurate initial point of $\mathbf{w}$, this algorithm is stable and converges usually nicely (similar to the LS-CMA, see [21]).

III-B. AKMA for case 1

Assume case 1 in figure 2, i.e., all packets in a slot are synchronized. We will derive a closed-form solution to the problem of estimating $\mathbf{w}$, in the style of ACMA [13]. This can be used to obtain an initial point for the iteration (4). Given a block of N samples, we try to minimize

$$\begin{array}{rcl} \hat{\mathbf{w}}_1 & = & \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{k=1}^{N} |\,|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 - c_k|^2 \\ & = & \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{k=1}^{N} |\mathbf{x}_k^{\mathrm{H}}\mathbf{w}\mathbf{w}^{\mathrm{H}}\mathbf{x}_k - c_k|^2 \\ & = & \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{k=1}^{N} |(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^{\mathrm{H}}(\bar{\mathbf{w}} \otimes \mathbf{w}) - c_k|^2 \\ & = & \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{P}(\bar{\mathbf{w}} \otimes \mathbf{w}) - \mathbf{c}\|^2\,, \end{array} \tag{5}$$

where $\otimes$ denotes a Kronecker product, $\bar{}$ stands for complex conjugate, $\mathbf{c} = [c_1, \cdots, c_N]^T$, $\mathbf{P} = (\bar{\mathbf{X}} \circ \mathbf{X})^{\mathrm{H}}$ and $\circ$ denotes a column-wise Kronecker product: $\bar{\mathbf{X}} \circ \mathbf{X} = [\bar{\mathbf{x}}_1 \otimes \mathbf{x}_1\,,\ \cdots,\ \bar{\mathbf{x}}_N \otimes \mathbf{x}_N]$. The size of $\mathbf{P}$ is $N \times M^2$. We follow the strategy of ACMA and split this optimization into two steps (hence suboptimal),

$$\begin{array}{rcl} \hat{\mathbf{y}} & = & \operatorname{argmin} \|\mathbf{P}\mathbf{y} - \mathbf{c}\|^2 \\ \hat{\mathbf{w}}_1 & = & \operatorname{argmin} \|\hat{\mathbf{y}} - \bar{\mathbf{w}} \otimes \mathbf{w}\|^2\,. \end{array}$$

If $\mathbf{P}$ is full column rank, the first problem has a unique solution in terms of the pseudo-inverse $\mathbf{P}^{\dagger}$:

$$\hat{\mathbf{y}} = \mathbf{P}^{\dagger}\mathbf{c}\,.$$

With this solution and setting $\hat{\mathbf{Y}} = \operatorname{unvec}(\hat{\mathbf{y}})$, where "unvec" denotes an unstacking of a vector into a square matrix, we can solve the second problem as

$$\hat{\mathbf{w}}_1 = \operatorname{argmin} \|\hat{\mathbf{y}} - \bar{\mathbf{w}} \otimes \mathbf{w}\|^2 = \operatorname{argmin} \|\hat{\mathbf{Y}} - \mathbf{w}\mathbf{w}^{\mathrm{H}}\|^2\,,$$

the solution of which is given in terms of the dominant eigenvector of $\mathbf{Y}$, scaled by the square root of the corresponding eigenvalue.

We thus see that, if $\mathbf{P}$ is full rank, the algorithm becomes particularly simple, and in the noise-free case will produce the exact separating beamformer to recover the desired packet. If $\mathbf{P}$ is not of full column rank, then there will exist additional solutions $\mathbf{y}_0$ to $\mathbf{P}\mathbf{y}_0 = \mathbf{0}$ (where $\mathbf{0}$ is an all-zero vector) which will add to the desired solution $\mathbf{y} = \bar{\mathbf{w}}_1 \otimes \mathbf{w}_1$, producing a result that cannot be factored. We thus need to study the rank properties of $\mathbf{P}$. This is done in section III-D.

III-C. AKMA for case 2 (known offset)

In case 2 in figure 2, users are not slotted. We first assume for simplicity that the base station is synchronized to user 1. Estimation of the offset is done in section IV.

If the analysis window length N is chosen to be the same as the packet length L, then the algorithm is the same as in section III-B. If N is larger than L, the packet has leading and/or trailing zeros. This can be modeled by defining

$$\mathbf{c} = [0 \cdots 0 \quad c_1 \quad c_2 \cdots c_L \quad 0 \cdots 0]^T\,, \qquad N \times 1\,,$$

where the total number of zeros is $N - L$ and the offset of the code matches the offset of the packet in the analysis window. After this, the algorithm is as in section III-B.

III-D. Rank of $\mathbf{P}$ *for case 1*

We consider the rank of P for case 1 (all users synchronized) in the noise-free case, where $\mathbf{X} = \mathbf{AZ}$. To recover $\mathbf{Z}$ using linear beamforming, we need $\mathbf{A}$ to be tall ($d \leq M$) and full rank. In this case, $\mathbf{X}$ has rank d. $\mathbf{P}$ has size $N \times M^2$, and $\mathbf{P}^{\mathrm{H}} = \bar{\mathbf{X}} \circ \mathbf{X} = (\bar{\mathbf{A}} \otimes \mathbf{A})(\bar{\mathbf{Z}} \circ \mathbf{Z})$, where $\bar{\mathbf{A}} \otimes \mathbf{A}$ has size $M^2 \times d^2$ and is full rank, and $\bar{\mathbf{Z}} \circ \mathbf{Z}$ has size $d^2 \times N$. The rank of $\mathbf{P}$ is equal to the rank of $\bar{\mathbf{Z}} \circ \mathbf{Z}$, therefore it cannot exceed d^2. A necessary condition for $\mathbf{P}$ to have rank d^2 is $d^2 \leq N$.

If $d = M$ and $d^2 \leq N$, then $\mathbf{P}$ can be full rank. If $d < M$, then $\mathbf{P}$ is not full rank, but can be made full rank by a prefiltering step (cf. [13], [22]). Compute the SVD of $\mathbf{X}$, i.e., $\mathbf{X} = \mathbf{U\Sigma V}$, where $\mathbf{U} : M \times d$ is unitary, $\mathbf{\Sigma} : d \times d$ is positive diagonal, and $\mathbf{V} : d \times N$ is unitary, then we can replace $\mathbf{X}$ by

$$\underline{\mathbf{X}} := (\sqrt{N})\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{H}}\mathbf{X} = (\sqrt{N})\mathbf{V}$$

which has d rows and is of full rank. Note that due to the prewhitening, $\underline{\mathbf{X}}$ satisfies a model $\underline{\mathbf{X}} = \underline{\mathbf{A}}\mathbf{Z}$, where $\underline{\mathbf{A}}$ is $d \times d$ and asymptotically *unitary* (for large N). From now on, we assume that the prewhitening has been performed and that, therefore, $d = M$ (we omit the underscore from the notation).

Even after the prefiltering, there are cases where $\mathbf{P}$ is singular, namely when at least two other sources are constant modulus (or equal-modulus). Indeed, if $\mathbf{z}^{(2)} = \mathbf{w}_2^{\mathrm{H}}\mathbf{X}$ and $\mathbf{z}^{(3)} = \mathbf{w}_3^{\mathrm{H}}\mathbf{X}$ are constant-modulus, then $\mathbf{P}(\bar{\mathbf{w}}_2 \otimes \mathbf{w}_2) = \mathbf{1}$, $\mathbf{P}(\bar{\mathbf{w}}_3 \otimes \mathbf{w}_3) = \mathbf{1}$, and

$$\mathbf{P}(\bar{\mathbf{w}}_2 \otimes \mathbf{w}_2 - \bar{\mathbf{w}}_3 \otimes \mathbf{w}_3) = \mathbf{0}\,.$$

Here, $\mathbf{1}$ is a vector with all entries equal to '1'. To avoid this nullspace solution, all sources (except perhaps one) should have amplitude modulations. In case 1, this level of cooperation is reasonable to assume.

We can show that if the sources are statistically independent constant modulus sources, all modulated by binary random power modulations $1 \pm \epsilon$, then as N becomes large, $\frac{1}{N}\mathbf{P}^{\mathrm{H}}\mathbf{P}$ converges to its expected value

$$\mathbf{C}_x := \mathrm{E}\{(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^{\mathrm{H}}\} = (\bar{\mathbf{A}} \otimes \mathbf{A})\mathbf{C}_z(\bar{\mathbf{A}} \otimes \mathbf{A})^{\mathrm{H}}\,, \quad (6)$$

where

$$\begin{aligned}\mathbf{C}_z &:= \mathrm{E}\{(\bar{\mathbf{z}}_k \otimes \mathbf{z}_k)(\bar{\mathbf{z}}_k \otimes \mathbf{z}_k)^{\mathrm{H}}\} \\ &= \mathbf{I} + \mathrm{vec}(\mathbf{I})\mathrm{vec}(\mathbf{I})^{\mathrm{H}} - (\mathbf{I} \circ \mathbf{I})(\mathbf{I} \circ \mathbf{I})^{\mathrm{H}} + \epsilon^2(\mathbf{I} \circ \mathbf{I})(\mathbf{I} \circ \mathbf{I})^{\mathrm{H}}\,,\end{aligned}$$

where $\mathbf{I}$ is the identity matrix. For the proof, see the appendix. The eigenvalues of $\mathbf{C}_z$ are

$$\mathrm{eig}(\mathbf{C}_z) = \{d + \epsilon^2,\ \underbrace{1, \cdots, 1}_{d^2-d},\ \underbrace{\epsilon^2, \cdots, \epsilon^2}_{d-1}\}\,. \quad (7)$$

For the proof, see again the appendix. These are also the eigenvalues of $\mathbf{C}_x$ since $\mathbf{A}$ is asymptotically unitary after prewhitening. Thus, the singular values of $\mathbf{P}$ converge to

$$\mathrm{svd}(\mathbf{P}) = \{\sqrt{N(d+\epsilon^2)},\ \underbrace{\sqrt{N}, \cdots, \sqrt{N}}_{d^2-d},\ \underbrace{\epsilon\sqrt{N}, \cdots, \epsilon\sqrt{N}}_{d-1}\}\,. \quad (8)$$

The smallest singular value of $\mathbf{P}$ is raised by the modulation from 0 to $\epsilon\sqrt{N}$. If ϵ is not too small, $\mathbf{P}$ will be left invertible, so that $\mathbf{y} = \mathbf{P}^{\dagger}\mathbf{c}$ will lead to the correct solution. It can also be verified that these singular values carry important information: truncating them to zero leads to the wrong result.

III-E. Rank of $\mathbf{P}$ *for case 2*

In case 2 there are additional situations where $\mathbf{P}$ becomes singular, namely when two sources are non-overlapping in time. Indeed, suppose $\mathbf{z}^{(2)} = \mathbf{w}_2^{\mathrm{H}}\mathbf{X}$, $\mathbf{z}^{(3)} = \mathbf{w}_3^{\mathrm{H}}\mathbf{X}$ are such that $z_k^{(2)} z_k^{(3)} = 0$, $\forall k$. Then $\mathbf{w}_2^{\mathrm{H}}\mathbf{x}_k\mathbf{x}_k^{\mathrm{H}}\mathbf{w}_3^{\mathrm{H}} = 0$, $\forall k$, hence

$$\mathbf{P}(\bar{\mathbf{w}}_2 \otimes \mathbf{w}_3) = \mathbf{0}\,, \qquad \mathbf{P}(\bar{\mathbf{w}}_3 \otimes \mathbf{w}_2) = \mathbf{0}\,.$$

Thus, solutions $\mathbf{y}$ to $\mathbf{Py} = \mathbf{c}$ can be written as

$$\mathbf{y} = \bar{\mathbf{w}}_1 \otimes \mathbf{w}_1 + \lambda_{23}(\bar{\mathbf{w}}_2 \otimes \mathbf{w}_3) + \lambda_{32}(\bar{\mathbf{w}}_3 \otimes \mathbf{w}_2)$$

for arbitrary scalars $\lambda_{23}, \lambda_{32}$, and an arbitrary $\mathbf{y}$ selected from the solution space cannot be factored into $\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1$. We see two solutions for this problem. Firstly, we can write $\mathbf{Y} = \mathrm{unvec}(\mathbf{y})$ as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 \end{bmatrix} \begin{bmatrix} 1 & & \\ & \lambda_{32} & \\ & & \lambda_{23} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1^{\mathrm{H}} \\ \mathbf{w}_3^{\mathrm{H}} \\ \mathbf{w}_2^{\mathrm{H}} \end{bmatrix} = \mathbf{W}\mathbf{\Lambda}_1\mathbf{M}^{\mathrm{H}}\,,$$

where $\mathbf{M}$ is a permutation of $\mathbf{W}$. Similarly, if we take a basis $\{\mathbf{y}_2, \mathbf{y}_3\}$ of the null space of $\mathbf{P}$, it can be written as

$$\mathbf{Y}_2 = \mathbf{W}\mathbf{\Lambda}_2\mathbf{M}^{\mathrm{H}}\,, \qquad \mathbf{Y}_3 = \mathbf{W}\mathbf{\Lambda}_3\mathbf{M}^{\mathrm{H}}\,,$$

where $\mathbf{\Lambda}_2$, $\mathbf{\Lambda}_3$ are diagonal matrices (with their first entry equal to 0). The problem boils down to a joint diagonalization of unsymmetric matrices, or a joint Schur decomposition, which can be solved using Jacobi iterations [13].

Alternatively, we try to avoid the joint diagonalization step. If we have N sufficiently large and do prewhitening, then $\mathbf{A}$ is approximately unitary, and the $\mathbf{w}_i$ are approximately orthogonal to each other. Hence, the desired solution $\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1$ is orthogonal to the null space of $\mathbf{P}$. In this case, we can simply set

$$\mathbf{y} = \mathbf{P}^{\dagger}\mathbf{c} \approx \bar{\mathbf{w}}_1 \otimes \mathbf{w}_1\,.$$

With noise, $\mathbf{P}$ will not be exactly singular, and we will have to set a threshold on the pseudo-inverse: compute the SVD of $\mathbf{P}$ as $\mathbf{P} = \mathbf{U}_P\mathbf{\Sigma}_P\mathbf{V}_P^{\mathrm{H}}$, let $\hat{\mathbf{\Sigma}}_P$ be the submatrix containing the singular values of $\mathbf{P}$ larger than a threshold, and let $\hat{\mathbf{U}}_P$, $\hat{\mathbf{V}}_P$ be the corresponding left and right singular vectors. The solution to $\mathbf{Py} = \mathbf{c}$ which is orthogonal to the (approximate) null space of $\mathbf{P}$ is then given by $\mathbf{y} = \mathbf{V}_P\hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{U}}_P^{\mathrm{H}}\mathbf{c}$. As is clear from equation (8), the threshold on the singular values of $\mathbf{P}$ should be smaller than $\epsilon\sqrt{N}$.

In case 2, it may also happen that only a few samples of the desired packet are disturbed by the head or tail of another source, but with insufficient samples present to estimate that source reliably. A convenient solution that avoids this problem is to increase the analysis window N to be larger than the packet length L, with the desired packet located in the center of the analysis window. In that case, if a disturbing source overlaps the desired packet, it will have at least $(N - L)/2$ samples in the analysis window, sufficient to detect it.

Figure 4 lists the algorithm as used in the simulations.

III-F. Simulations–known timing

We test the algorithm on simulated data. In these simulations, the receiver knows the code of the desired user, and it knows the timing of this user.

1. SVD: $\mathbf{X} =: \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ (M^2N)
   Estimate rank and truncate to $\mathbf{U}_s\mathbf{\Sigma}_s\mathbf{V}_s$
   Prefiltering: $\underline{\mathbf{X}} := \sqrt{L}\cdot\mathbf{\Sigma}_s^{-1}\mathbf{U}_s^{\mathrm{H}}\mathbf{X} = \sqrt{L}\cdot\mathbf{V}_s$
2. $\mathbf{P} = (\underline{\bar{\mathbf{X}}} \circ \underline{\mathbf{X}})^{\mathrm{H}}$ (d^2N)
   $\mathbf{y} = \mathbf{P}^{\dagger}\mathbf{c}$, with threshold $\frac{1}{2}\epsilon\sqrt{L}$ (d^4N)
3. $\mathbf{Y} = \mathrm{unvec}(\mathbf{y})$
   $\underline{\mathbf{w}}$ = dominant eigenvector of $\mathbf{Y}$ (d^3)
4. $\mathbf{w} = \sqrt{L}\cdot\mathbf{U}_s\mathbf{\Sigma}_s^{-1}\underline{\mathbf{w}}$
   $\hat{\mathbf{z}} = \mathbf{w}^{\mathrm{H}}\mathbf{X}$ (dN)
5. *optional:* alternating projection iterations

$\mathcal{O}(M^2N + d^4N)$

Fig. 4. Summary of AKMA and the complexity of the algorithm. L is the packet length.

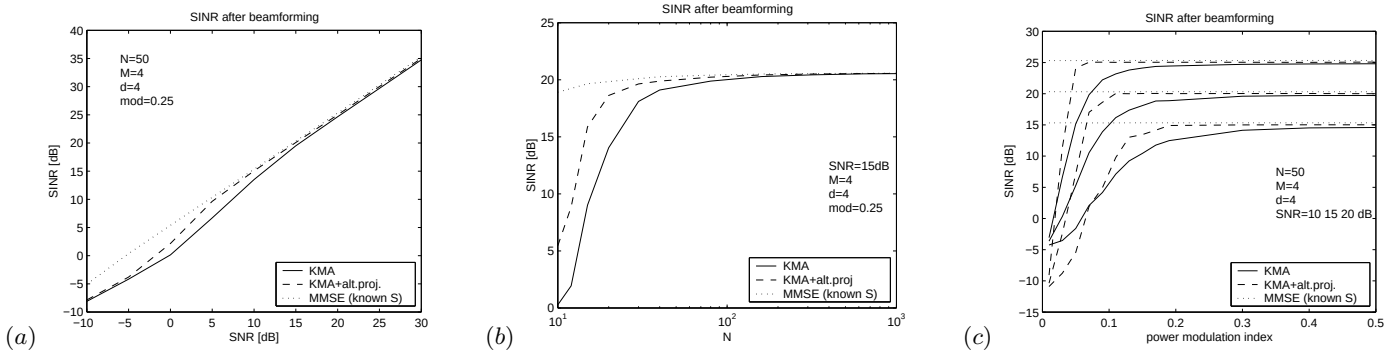

Fig. 5. Case 1 beamformer performance: SINR of user 1 after beamforming.

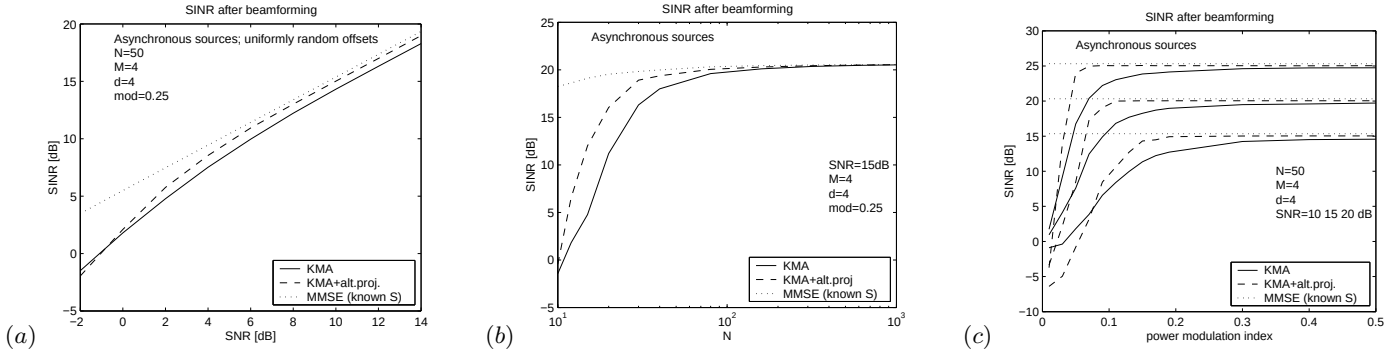

Fig. 6. Case 2 beamformer performance: SINR of user 1 after beamforming. Asynchronous sources with equal-length packets.

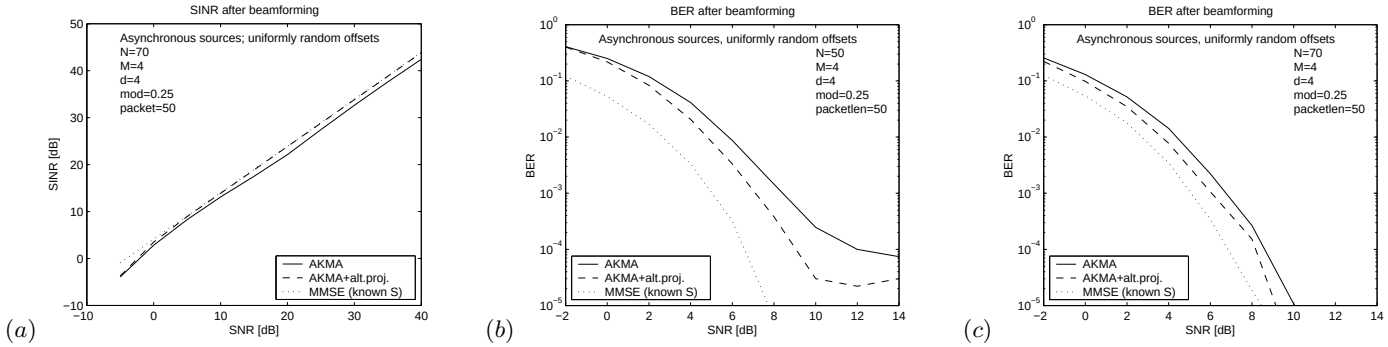

Fig. 7. Case 2 beamformer performance: SINR and BER of user 1 after beamforming. Asynchronous sources with equal-length packets $L = 50$, and an analysis window (b) $N = 50$ or (a, c) $N = 70$.

Case 1

Figure 5 shows SINR performance plots of the beamformer of the first source for a simulation with $d = 4$ sources, $M = 4$ antennas in a uniform linear array, equal source powers, and source angles $-20^\circ, 20^\circ, 40^\circ, -40^\circ$, for varying SNR, packet length $L(= N)$, and power modulation index ϵ. All sources are QPSK-modulated constant modulus sources with code-modulated amplitudes. The reference line is the performance of the MMSE receiver with knowledge of $\mathbf{z}^{(1)}$, namely $\mathbf{w} = (\mathbf{z}^{(1)}\mathbf{X}^\dagger)^{\mathrm{H}}$. The solid line is the performance of AKMA, the dashed line the performance of 15 iterations of the alternating projection algorithm, initialized by the AKMA. It is seen that the performance of the AKMA is generally quite good, but that it can be improved for small modulation indices and small N (i.e., $N < 2d^2$). This is due to the squaring involved in the construction of $\mathbf{P}$, and the presence of additional kernel solutions for small modulations.

The convergence of the AKMA to the MMSE is expected: similar to ACMA it is caused by the prewhitening and the fact that the algorithm is unbiased in the noise-free case [22].

Case 2

Figure 6 shows similar performance curves, but for case 2 (unsynchronized users). In these plots, the analysis window N is equal to the packet length L; the packet of user 1 falls completely in the analysis window whereas the three other users have arbitrary arrival times. The performance is virtually identical to that in case 1.

In figure 7, we investigate the choice $N > L$. This is motivated by the bit-error rate curve in fig. 7(b), which (for $N = L = 50$) shows a saturation of the performance for large SNRs. This is because the head and tail of the desired packet can be disturbed by the tail of another source, but with insufficient samples present to estimate that source reliably. Figure 7(c) shows that increasing the analysis window N to $N = 70$ improves the bit error rate (BER) for large SNRs. Figure 7(a) shows the SINR performance for $N = 70$, which can be compared to Figure 6(a) where $N = L = 50$.

IV. JOINT OFFSET AND BEAMFORMER ESTIMATION

In the previous section, we assumed that the receiver was synchronized to the user of interest. Now, we consider the situation where the receiver is not synchronized, e.g., the users transmit packets at random moments. We assume that the receiver collects a batch of N samples, where $N > L$, and introduce an algorithm to estimate the offset of the packet of the desired user within this analysis window, as well as a beamformer to cancel the interference. Introduce τ as the unknown packet offset. Similar to equation (5), we now have to compute a beamformer $\mathbf{w}$ and offset τ such that

$$|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 = |\hat{z}_k|^2 \;=\; c_{k-\tau}\,, \qquad k = 1, \cdots, N\,. \tag{9}$$

For simplicity, we first assume that τ is an integer, and derive corresponding algorithms in section IV-A. In section IV-D we extend this to arbitrary noninteger delays, which can be estimated if oversampling is considered.

IV-A. Integer offset estimation

Let user 1 with code $\{c_k,\ k = 1, \cdots, L\}$ be the user of interest. We consider the case where the packet falls completely within the analysis window, $N > L$, and the packet offset τ is an integer. Our aim is to compute

$$\begin{aligned}\{\hat{\mathbf{w}}, \hat{\tau}\} &= \operatorname*{argmin}_{\mathbf{w},\tau} \sum_{k=1}^{N} (|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 - c_{k-\tau})^2 \\ &= \operatorname*{argmin}_{\mathbf{w},\tau} \|\mathbf{P}(\bar{\mathbf{w}} \otimes \mathbf{w}) - \mathbf{c}_\tau\|^2\,,\end{aligned}$$

Here,

$$\mathbf{c}_\tau = [\underbrace{0, \cdots, 0}_{\tau},\ c_1, \cdots, c_L,\ \underbrace{0, \cdots, 0}_{N-L-\tau}]^T$$

is a vector of length N, and the matrix $\mathbf{P}$ is constructed from the received data in the same way as in section III-B. As before, we continue with a two-step optimization problem

$$\begin{aligned}\{\hat{\mathbf{y}}, \hat{\tau}\} &= \operatorname*{argmin}_{\mathbf{w},\tau} \|\mathbf{P}\mathbf{y} - \mathbf{c}_\tau\|^2 \\ \hat{\mathbf{w}} &= \operatorname*{argmin}_{\mathbf{w}} \|\hat{\mathbf{y}} - \bar{\mathbf{w}} \otimes \mathbf{w}\|^2\,,\end{aligned}$$

and solve the first problem, which asks for joint estimation of $\mathbf{y}$ and τ. Similar to the derivation of the SI-JADE algorithm [23], we exploit the fact that a delay in time domain corresponds to a phase progression in frequency domain. This can be expressed as

$$\mathbf{F}\mathbf{c}_\tau = \mathbf{F}\mathbf{c}_0 \odot \boldsymbol{\phi}_\tau$$

where $\mathbf{F}$ is the $N \times N$ DFT matrix, $\odot$ represents an entrywise multiplication (Schur-Hadamard product),

$$\boldsymbol{\phi}_\tau := [1 \quad \varphi \quad \varphi^2 \ \cdots \ \varphi^{N-1}]^{\mathrm{T}}$$

and $\varphi = e^{-j\frac{2\pi\tau}{N}}$. The vector $\mathbf{c}_0$ is the unshifted code vector followed by $N - L$ zeros. Our objective will be to estimate φ based on the shift-invariance structure exhibited by the vector $\boldsymbol{\phi}_\tau$. This then immediately determines the offset τ. A similar approach was considered in the SI-JADE algorithm [23] for joint angle-delay estimation.

Thus, apply $\mathbf{F}$ to the equation $\mathbf{P}\mathbf{y} = \mathbf{c}_\tau$ to obtain

$$\mathbf{F}\mathbf{P}\mathbf{y} = \mathbf{F}\mathbf{c}_0 \odot [1\ \varphi\ \varphi^2\ \ldots\ \varphi^{N-1}]^{\mathrm{T}} \quad \Leftrightarrow \quad \mathbf{P}_f\mathbf{y} = \mathbf{g} \odot \boldsymbol{\phi}_\tau \tag{10}$$

where $\mathbf{P}_f := \mathbf{F}\mathbf{P}$ and $\mathbf{g} := \mathbf{F}\mathbf{c}_0$. Dividing the rows of $\mathbf{P}_f$ with the corresponding entries of the vector $\mathbf{g}$ we arrive at

$$\tilde{\mathbf{P}}\mathbf{y} = \boldsymbol{\phi}_\tau \tag{11}$$

where $\tilde{\mathbf{P}} = (\mathrm{diag}(\mathbf{g}))^{-1}\mathbf{P}_f$ is known and the vector $\boldsymbol{\phi}_\tau$ is a known function of the unknown delay τ. Here, 'diag' maps a vector into a diagonal matrix. The above pointwise division puts a design constraint on the code: it should be chosen such that (after zero padding to length N) it does not contain small values in the DFT domain.

Equation (11) can be treated in several different ways. Essentially we have to search for a vector in the column span of $\tilde{\mathbf{P}}$ that has the structure exhibited by $\boldsymbol{\phi}_\tau$, i.e., a shift invariance structure. Obviously a MUSIC-type search is applicable: if $\mathbf{U}_s$ is a basis for the dominant column span of $\tilde{\mathbf{P}}$, then

$$\hat{\tau} = \operatorname*{argmax}_{\tau}\ \|\boldsymbol{\phi}_\tau^{\mathrm{H}}\mathbf{U}_s\|^2\,. \tag{12}$$

The optimum is found by searching over a coarse grid, selecting the best interval, and subsequently refining if a higher resolution is required. For integer values of τ, the rows $\boldsymbol{\phi}_\tau^{\mathrm{H}}$ are actually rows of the inverse FFT matrix $\mathbf{F}^{\mathrm{H}}$, and hence we can implement the coarse search over integer values of τ by applying an inverse FFT to $\mathbf{U}_s$. Hence, we simply have to find the row of $\mathbf{F}^{\mathrm{H}}\mathbf{U}_s$ with maximal norm. Essentially, the algorithm has at

this point performed a deconvolution with the desired code, implemented in frequency domain. The MUSIC-type algorithm shows good performance in simulations. Besides the processing steps described in figure 4, the additional complexity due to the synchronization step is given by: the FFT of $\mathbf{P}$ and IFFT of $\mathbf{U}_s$ ($2d^2N\log_2 N$), division by the user's code (d^2N) and computation of the norm of the first L rows (Ld^2). The complexity of the synchronization part is therefore of order $\mathcal{O}(2d^2N\log_2 N)$.

IV-B. ESPRIT-like algorithms

To avoid the search, we can also implement an ESPRIT-like algorithm, where the difference is that, here, we expect only a single column in the column span of $\tilde{\mathbf{P}}$ with shift-invariance structure, whereas in ESPRIT all columns have such a structure.

To this end, split $\tilde{\mathbf{P}}$ into two matrices $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ by taking its first and last $N-1$ rows, respectively. We thus obtain

$$\begin{array}{rcl}\tilde{\mathbf{P}}_x\mathbf{y} & = & [1 \quad \varphi \quad \cdots \quad \varphi^{N-2}]^{\mathrm{T}} \\ \tilde{\mathbf{P}}_y\mathbf{y} & = & [\varphi \quad \varphi^2 \quad \cdots \quad \varphi^{N-1}]^{\mathrm{T}}\,.\end{array} \tag{13}$$

This can also be written as

$$\tilde{\mathbf{P}}_x\mathbf{y} = \bar{\varphi}\tilde{\mathbf{P}}_y\mathbf{y}\,, \tag{14}$$

which (because $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ are tall) is recognized as a matrix pencil problem. To solve it, we must first find the common column span of $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$. Equivalently, we can look at the null space of $[\tilde{\mathbf{P}}_x \quad \tilde{\mathbf{P}}_y]$.

Algorithm 1

The simplest technique to intersect the column span of $\tilde{\mathbf{P}}_x$ and that of $\tilde{\mathbf{P}}_y$ is to compute the SVD of $[\tilde{\mathbf{P}}_x \quad \tilde{\mathbf{P}}_y]$. Indeed, from equation (13), we see that

$$[\tilde{\mathbf{P}}_x \quad \tilde{\mathbf{P}}_y]\begin{bmatrix}\mathbf{y}\\ -\bar{\varphi}\mathbf{y}\end{bmatrix} = \mathbf{0}\,. \tag{15}$$

Now it is clear that after an "economy size" SVD is performed of $[\tilde{\mathbf{P}}_x \quad \tilde{\mathbf{P}}_y]$, at least one singular value will be zero. The corresponding basis for the null space specifies a set of candidate solutions to (14). If $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ each have full column rank, then we can simplify immediately, since we expect only a single solution $\mathbf{v}_n$ in the null space, which then will have the form

$$\mathbf{v}_n =: \begin{bmatrix}\mathbf{v}_x\\ \mathbf{v}_y\end{bmatrix} = \begin{bmatrix}\mathbf{y}\\ -\bar{\varphi}\mathbf{y}\end{bmatrix}\,. \tag{16}$$

After finding $\mathbf{v}_n$, we can estimate the phaseshift φ as $\hat{\varphi} = -\overline{(\mathbf{v}_x^{\dagger}\mathbf{v}_y)}$, which directly specifies a delay estimate $\hat{\tau}$. The estimate can be improved by performing a MUSIC-type search (12) in the vicinity of this estimate. This will be referred to as "Algorithm 1" in the simulations.

At the same time we can set $\hat{\mathbf{y}} := \mathbf{v}_x$, and since $\mathbf{y} = \bar{\mathbf{w}}\otimes\mathbf{w}$ we can estimate the separating beamformer $\mathbf{w}$ as indicated before: set $\hat{\mathbf{Y}} = \mathrm{unvec}(\hat{\mathbf{y}})$, and let $\hat{\mathbf{w}}$ be the dominant eigenvector of $\mathbf{Y}$, scaled by the square root of the corresponding eigenvalue. This is the estimated beamformer for user 1.

The above algorithm assumed that $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ are full rank. Alternatively we can work with a basis of these subspaces, obtained e.g., after the "economy-size" SVDs

$$\begin{array}{rcl}\tilde{\mathbf{P}}_x & \approx & \hat{\mathbf{U}}_x\hat{\mathbf{\Sigma}}_x\hat{\mathbf{V}}_x^{\mathrm{H}} \\ \tilde{\mathbf{P}}_y & \approx & \hat{\mathbf{U}}_y\hat{\mathbf{\Sigma}}_y\hat{\mathbf{V}}_y^{\mathrm{H}}\end{array} \tag{17}$$

where we drop the small singular values and corresponding vectors. Similar to (15) we find

$$[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y]\begin{bmatrix}\hat{\mathbf{\Sigma}}_x\hat{\mathbf{V}}_x^{\mathrm{H}}\mathbf{y}\\ -\bar{\varphi}\hat{\mathbf{\Sigma}}_y\hat{\mathbf{V}}_y^{\mathrm{H}}\mathbf{y}\end{bmatrix} = 0\,. \tag{18}$$

We can compute the vector ($\mathbf{v}_n$ say) in the null space of $[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y]$, which will have the following structure:

$$\mathbf{v}_n =: \begin{bmatrix}\mathbf{v}_x\\ \mathbf{v}_y\end{bmatrix} = \begin{bmatrix}\hat{\mathbf{\Sigma}}_x\hat{\mathbf{V}}_x^{\mathrm{H}}\mathbf{y}\\ -\bar{\varphi}\hat{\mathbf{\Sigma}}_y\hat{\mathbf{V}}_y^{\mathrm{H}}\mathbf{y}\end{bmatrix}\,. \tag{19}$$

The vector $\mathbf{y}$ can be computed as $\mathbf{y} = \hat{\mathbf{V}}_x\hat{\mathbf{\Sigma}}_x^{-1}\mathbf{v}_x$, and φ follows from $-\bar{\varphi} = (\hat{\mathbf{\Sigma}}_y\hat{\mathbf{V}}_y^{\mathrm{H}}\mathbf{y})^{\dagger}\mathbf{v}_y$.

Algorithm 2

Another algorithm for subspace intersection is mentioned in [24]: the common vector in the column span of $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ is given by the *largest* left singular vector of $[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y]$, the one corresponding to a singular value $\sqrt{2}$. Interestingly, this vector should have the structure $\boldsymbol{\phi} = [1 \quad \varphi \quad \varphi^2\cdots\varphi^{N-2}]^{\mathrm{T}}$. By computing the vector in the intersection and matching it to this shift-invariance structure, we have another way to compute φ, and hence the offset delay.

Let $\mathbf{u}$ be the largest left singular vector of $[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y]$. Under noise-free conditions, we have $\mathbf{u} = [1 \quad \varphi \quad \varphi^2\cdots\varphi^{N-2}]^{\mathrm{T}}$. We can estimate φ as in ESPRIT, by constructing $\mathbf{u}_x$ and $\mathbf{u}_y$ consisting of the first and last $N-2$ elements of $\mathbf{u}$, respectively, so that $\hat{\varphi} = \mathbf{u}_x^{\dagger}\mathbf{u}_y$. It is possible to obtain a better estimate of φ by the additional limited MUSIC search using the complete known structure of $\boldsymbol{\phi}$.

In each of the algorithms, the estimated beamformer can be improved by a few iterations of an alternating projection algorithm as already proposed in section III-A.

IV-C. Simulations–integer timing estimation

In the following simulations we consider $d = 4$ users and $M = 4$ antennas in a uniform linear array with half-wavelength spacing. Signals are arriving at the array with angles $[-10°,\ 20°,\ 40°,\ -40°]$ with respect to the array broadside, and with delays [50, 34, 65, 95] samples. The packet length is $L = 128$ while the analysis window size is $N = 256$. All sources are transmitting unit amplitude constant modulus signals modulated by a power modulation with index $\epsilon = 0.5$. The amplitude codes are Gold sequences in order to minimize the cross correlation between the codes of different users. 1000 Monte-Carlo runs for each value of the input SNR were performed.

The power of the i-th user is defined in the region where the signal exists *i.e.*, $P_i = (\sum_{k=\tau_i+1}^{\tau_i+L}|s_k^{(i)}|^2)/L$. P_1 is the power of user 1 - the user of interest. In the simulations presented in figure 8(a,b,c) all users have the same transmitting power whereas in figure 8(d) the power of the user of interest is varied with respect to the power of the interfering sources. The input SNR in dB is defined as $SNR = 10\log(P_1/P_n)$ where $P_n = \sigma^2$ is the power of the AWGN per receiving antenna. The SINR after beamforming is computed as $SINR_{out} = 10\log(P_{1,out}/P_{interf+n})$ with $P_{1,out} = \mathbf{w}^{\mathrm{H}}\mathbf{a}_1P_1\mathbf{a}_1^{\mathrm{H}}\mathbf{w}$, $P_{interf+n} = \mathbf{w}^{\mathrm{H}}(\mathbf{A}\mathrm{diag}([P_1\cdots P_d])\mathbf{A}^{\mathrm{H}} - \mathbf{a}_1P_1\mathbf{a}_1^{\mathrm{H}} + \sigma^2\mathbf{I})\mathbf{w}$, where $\mathbf{a}_1$ is the first column of the array manifold $\mathbf{A}$.

Figure 8(a) presents the root mean square error (RMSE) of the estimated delay for user 1 for each of the proposed algorithms. Similarly, figure 8(b) shows the percentage of cases where the delay offset was not estimated correctly. An estimate is labeled as failure if its rounded value is not equal to the true (integer) delay offset.

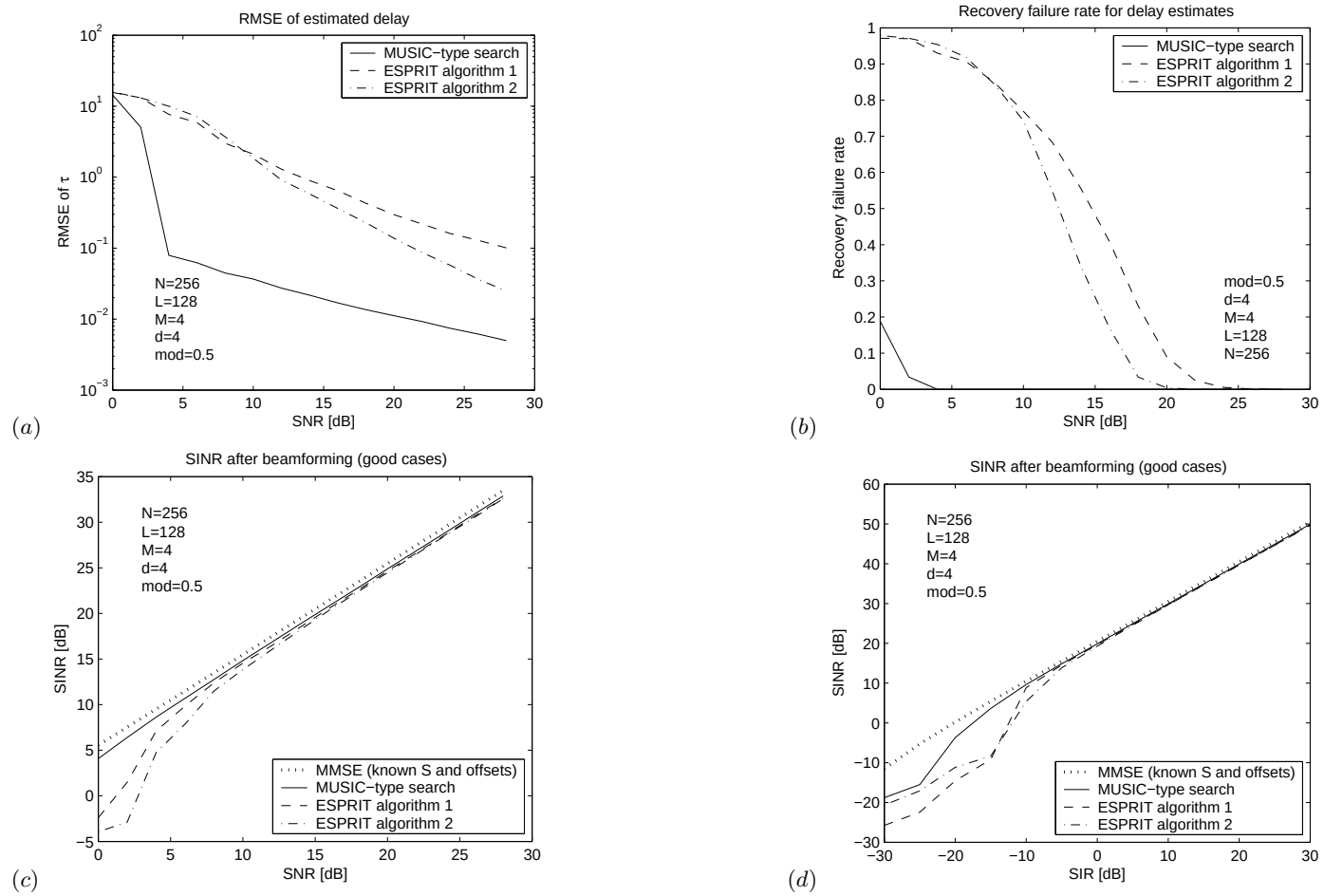

Fig. 8. Case 2 (asynchronous sources): (a) Root Mean Square Error of the delay estimate; (b) percentage of incorrectly estimated delays τ; (c) output SINR of user 1 after beamforming (cases without failure), all users having equal power, (d) output SINR of user 1, where its power is varied with respect to the total power of the interfering users.

From the panels (a) and (b), we see that the MUSIC search performs very reliable, and much better than the closed-form ESPRIT-type algorithms. These algorithms perhaps can be used to obtain coarse estimates of the delays, which can subsequently be refined using a MUSIC-type search over a limited interval. (At the same time, the complexity of the MUSIC-search, when implemented via an inverse FFT, is lower than that of the ESPRIT-type algorithms.)

The "jump" in the performance for low SNR is typical for subspace algorithms: these involve a nonlinear step namely the selection of a subspace which, for low SNR, may exclude the vector of interest.

For the cases without failure, figure 8(c) presents the signal to interference and noise ratio (SINR) at the output, i.e., after beamforming. The dotted line is a theoretical reference line showing the performance of the MMSE beamformer assuming the transmitted signals, codes and offsets of all users are known. The statistics are computed only over the cases without failure. The performance of the algorithms follows that of the MMSE estimator closely, and can be further improved with a few iterations of the alternating projection algorithm.

Figure 8(d) presents the SINR after beamforming as a function of the input signal to interference ratio $SIR = 10\log(P_1/P_{interf})$. All interfering sources have fixed unit power while the power of the user of interest is changed in order to simulate a near/far scenario. The interference to noise ratio is defined as $INR = 10\log(P_{interf}/P_n) = +15$dB where P_{interf} , P_n are the total interfering and noise power respectively. The INR is kept constant for all the SIR values in the simulation.

IV-D. Non-integer offset estimation

There are two cases where the DFT property of mapping a delay to a phase progression is accurate: the delay is a multiple of the sampling period, or the signal is sampled at or above Nyquist rate. If the signal was sampled below the Nyquist rate, aliasing occurs which destroys the shift invariance property (cf. [24]). Up to this point in the paper, we considered sampling at the symbol rate. Since delays are, in general, noninteger, this leads to inaccuracies when realistic data is considered. Therefore we now consider a case-2 scenario with non-integer delays τ, and where we sample at a rate of P times the symbol rate, assuming that this is above Nyquist. Since there is no need to sample much faster than Nyquist, typically $P = 2$ is sufficient.

The signal $s(t)$ is an analog constant modulus signal, e.g., a phase modulated data sequence. The amplitude code sequence is an analog function $c(t)$, and we assume that $z(t) = s(t)\sqrt{c(t)}$ is sampled above Nyquist, hence $c(t)$ is sufficiently smooth. Under these conditions, we are still in context of the proposed KMAs, hence the joint delay-beamformer estimation algorithms proposed in section IV-A are applicable.

Because of oversampling, the spectrum of the 'zero delay code' vector $\mathbf{g} = \mathbf{F}\mathbf{c}_0$ introduced in IV has most of the power concen-

trated around the zero frequency, and the spectrum decays fast for higher frequencies. Only the N/P samples centered around the zero frequency are expected to have significant amplitude. In the computation of $\tilde{\mathbf{P}}$, we avoid divisions by small values by taking only the central N/P rows of $\tilde{\mathbf{P}}$ corresponding to the significant part of the vector $\mathbf{g}$, and discarding the other rows.

The estimated delay will be a rational number. For improved accuracy, we propose not to round it to the nearest sample instant, but rather to resample the signal after shifting it over $\hat{\tau}$. Using Shannon's resampling theorem, we can obtain new samples at exactly $\hat{\tau} + k\frac{T}{P}$, $k = 0, \cdots, LP-1$, where T is the symbol period, namely

$$r_k = \hat{z}_{intp}(\hat{\tau} + k\frac{T}{P}) = \sum_{n=0}^{N-1} (\hat{\mathbf{w}}^{\mathrm{H}}\mathbf{x}_n) \cdot \mathrm{sinc}(\frac{\hat{\tau} + (k-n)(T/P)}{T/P}) \,,$$

where $k = 0, \ldots, LP-1$. These samples are used for detection.

IV-E. Simulations–noninteger timing estimation

In the following simulations, we used phase-modulated sources that employ Minimum Shift Keying (MSK) modulation. Even if phase modulation is a non-linear operation in general, an MSK-modulated signal can also be represented as a linearly modulated signal[1]

$$s(t) = d(t) * p(t) = \sum_{k=1}^{L} d_k \cdot p(t - kT) \;\; , \qquad (20)$$

where the symbols d_k satisfy $d_k \in \{-1, 1\}$ for even k, $d_k \in \{-j, j\}$ for odd k, and the pulse shape is

$$p(t) = \begin{cases} \cos\frac{\pi}{2}\frac{t}{T} & -T \le t < T \,, \\ 0 & \text{otherwise} \,. \end{cases}$$

Known modulus variations (the code) are inserted over the signal $s(t)$ as

$$z(t) \quad = \quad \textstyle\sum_{k=1}^{L} \sqrt{c_k} \cdot d_k \cdot p(t-k) = \sum_{k=1}^{L} z_k \cdot p(t-k) \,.$$

We consider $d = 2$ equal-power users transmitting data packets of equal size $L = 64$ symbols. The receiver has $M = 2$ antennas and employs $P = 2$ times oversampling. The analysis window is $N = 256$ samples, or 128 symbol periods. The delays of the two sources were set at [17.65, 35.92].

In figure $9(a)$, the BER versus input SNR performance of the KMA beamformer for the cases with and without resampling via interpolation are presented. In the case without interpolation, the estimated τ is rounded to the nearest sample instant. For reference, we also show the performance of the Linear Minimum Mean Squared Error (MMSE) receiver, where the packet offsets τ_q and the transmitted sequences $z^{(q)}(t)$ are known for all users $q = 1, 2$, and the performance of an MMSE receiver in the case where no amplitude modulation code was inserted by the transmitters.

From the figure, we observe that the performance of KMA improves due to the interpolation; as before, at higher SNRs the performance comes close to that of the MMSE. We also observe a gap between the MMSE receiver for amplitude-modulated signals and unmodulated CM signals. The explanation is that for the modulated signal, the bits that happen to have less power $(1-\epsilon)$ have less protection against noise than those that have higher power $(1+\epsilon)$. This motivates the use of a small ϵ. On the other hand, figure $5(c)$ showed that ϵ should be sufficiently large to guarantee a good estimate of $\mathbf{w}$. Also the packet length L enters into this trade-off because this minimum value is inversely proportional to $\sqrt{L}$.

Since, with oversampling, we require that the pulse shape is a smooth analog function, we can compute the Cramer-Rao bound (CRB) of the estimation of τ for this case. The CRB specifies a lower bound on the variance of any unbiased estimator for τ and is derived in Appendix C. To verify the performance of the delay estimation algorithm, we performed a simulation using the same parameters as before. The Cramer-Rao Bound for the estimates of the packet offset is depicted in figure $9(b)$ with a dashed line. The solid line represents the standard deviation of the estimates of τ_1 for the user of interest. The gap between the CRB and the standard deviation of the MUSIC-type search algorithm (about a factor 2) shows that the algorithm is not efficient. This can be explained by the fact that the data is squared in the proposed algorithm, which essentially doubles the noise.

In figure $9(c)$ the packet offset recovery failure rate versus input SNR is presented. A delay estimate $\hat{\tau}$ is considered as good if $\tau - \frac{T}{2} < \hat{\tau} < \tau + \frac{T}{2}$ is satisfied.

V. Conclusion

In this paper, we considered an ad hoc network where users are unsynchronized and send packets at random. An antenna array is used by the receiver to focus on the user of interest and suppress unwanted interfering packets. We introduced the KMA algorithms to estimate the timing offset and separating beamformer of the packet of a user of interest. The algorithm recognizes the user of interest from his 'color code', an amplitude modulation of the constant modulus data signal. This color code does not increase the bandwidth of the signal, and the amplitude modulation can be small (e.g., 25% or 1 dB) if the packet length is sufficiently large (e.g., 50 symbols).

An advantage of the scheme is that it can be used as an upgrade to existing WLAN services or ad hoc networks (provided they use constant modulus signals). A transmitter employing a color code will have an advantageous reception at a KMA receiver, but will also be compatible with legacy receivers since the modulation index is small (around the noise level for sufficiently large packets). The algorithms can tolerate frequency-hopping schemes, since non-stationary interfering signals *i.e.*, randomly appearing packets are admitted.

Appendices

A. Proof of equation (6)

First consider a scalar variable, $z = \sqrt{c} \cdot s$, where s is a constant modulus random variable, $c = 1 + e$, and $e = \pm\epsilon$ is a zero mean binary random variable. Then $\mathrm{E}(|z|^4) = \mathrm{E}(1+e)^2 = 1+\epsilon^2$. The kurtosis of z is $\kappa_z = \mathrm{E}(|z|^4) - 2\mathrm{E}(|z|^2)^2 = -1 + \epsilon^2$.

To prove (6), we use some properties of random CM vector signals that are listed in [22]. In particular, if $\mathbf{s}$ is a random vector whose entries are statistically independent and constant modulus, then

$$\begin{aligned} \mathbf{C}_s \;\; &:= \;\; \mathrm{E}\{(\bar{\mathbf{s}} \otimes \mathbf{s})(\bar{\mathbf{s}} \otimes \mathbf{s})^{\mathrm{H}}\} \\ &= \;\; \mathrm{E}\{\mathbf{s}\mathbf{s}^{\mathrm{H}}\}\mathrm{E}\{\bar{\mathbf{s}}\bar{\mathbf{s}}^{\mathrm{H}}\} + \mathrm{E}\{\bar{\mathbf{s}} \otimes \mathbf{s}\}\mathrm{E}\{\bar{\mathbf{s}} \otimes \mathbf{s}\}^{\mathrm{H}} + \mathbf{K}_s \\ &= \;\; \mathbf{I} + \mathrm{vec}(\mathbf{I})\mathrm{vec}(\mathbf{I})^{\mathrm{H}} - (\mathbf{I} \circ \mathbf{I})(\mathbf{I} \circ \mathbf{I})^{\mathrm{H}} \,, \end{aligned}$$

where the $d^2 \times d^2$ matrix $\mathbf{K}_s = -(\mathbf{I} \circ \mathbf{I})(\mathbf{I} \circ \mathbf{I})^{\mathrm{H}}$ is the kurtosis of $\mathbf{s}$. It is a diagonal matrix with only d nonzero entries (-1) at selected locations on its main diagonal, which reflects the independence of the entries of $\mathbf{s}$.

[1] The linear modulation is introduced for simplicity, nonlinear modulations are certainly also applicable.

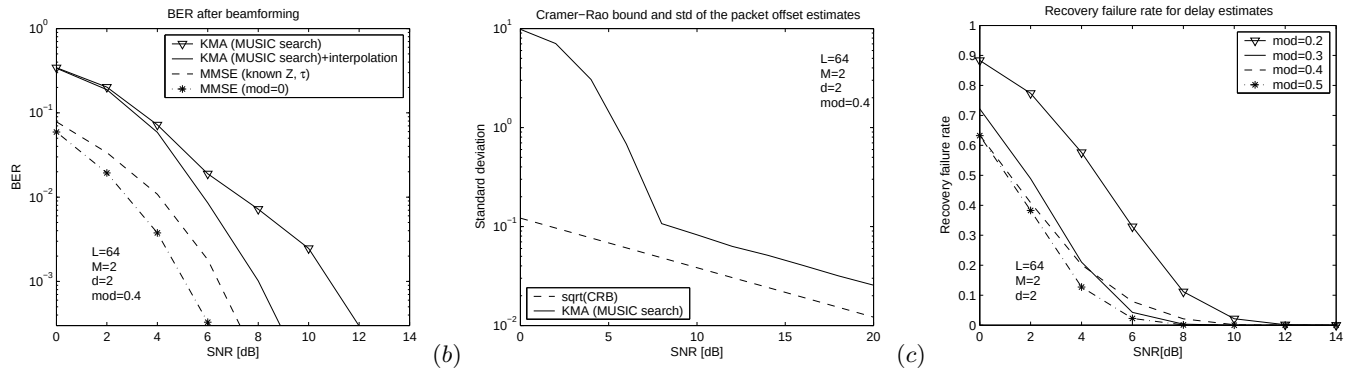

Fig. 9. Case 2 beamformer performance (asynchronous sources with equal-length packets) where MSK modulation scheme is implemented: (a) BER after beamforming; (b) the standard deviation of estimated delays $\hat{\tau}$ using the MUSIC-type search, compared to the CRB; (c) the delay offset failure rate for different values of the modulation level $\epsilon = [0.2, 0.3, 0.4, 0.5]$.

Now, define $\mathbf{z} = \mathbf{\Gamma}^{1/2}\mathbf{s}$, where $\mathbf{\Gamma} = \mathbf{I} + \mathbf{E}$ and $\mathbf{E}$ is diagonal with independent entries $\pm\epsilon$, then

$$\mathrm{E}\{\mathbf{z}\mathbf{z}^{\mathrm{H}}\} = \mathrm{E}\{\mathbf{\Gamma}\} = \mathbf{I}\,,$$

and similarly $\mathrm{E}\{\bar{\mathbf{z}} \otimes \mathbf{z}\} = \mathrm{vec}(\mathbf{I})$. We already showed that the kurtosis of a single entry of $\mathbf{z}$ is $-1+\epsilon^2$. Because of independence, the kurtosis of $\mathbf{z}$ is therefore $\mathbf{K}_z = (-1+\epsilon^2)(\mathbf{I}\circ\mathbf{I})(\mathbf{I}\circ\mathbf{I})^{\mathrm{H}}$. It follows that

$$\begin{aligned}\mathbf{C}_z &= \mathrm{E}\{\mathbf{z}\mathbf{z}^{\mathrm{H}}\}\mathrm{E}\{\bar{\mathbf{z}}\bar{\mathbf{z}}^{\mathrm{H}}\} + \mathrm{E}\{\bar{\mathbf{z}}\otimes\mathbf{z}\}\mathrm{E}\{\bar{\mathbf{z}}\otimes\mathbf{z}\}^{\mathrm{H}} + \mathbf{K}_z \\ &= \mathbf{I} + \mathrm{vec}(\mathbf{I})\mathrm{vec}(\mathbf{I})^{\mathrm{H}} + (-1+\epsilon^2)(\mathbf{I}\circ\mathbf{I})(\mathbf{I}\circ\mathbf{I})^{\mathrm{H}}.\end{aligned}$$

B. Proof of equation (7)

Inspection of the structure of $\mathbf{C}_z$ shows that, after column and row permutations, it can be written as

$$\mathbf{C}_z \sim \begin{bmatrix} \mathbf{1}_d\mathbf{1}_d^{\mathrm{H}} + \epsilon^2\mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{d^2-d} \end{bmatrix}$$

where $\mathbf{1}_d$ is a vector consisting of d entries '1', and $\mathbf{I}_d$ is the $d \times d$ identity matrix. Hence $d^2 - d$ eigenvalues are equal to 1. The eigenvalues of $\mathbf{1}_d\mathbf{1}_d^{\mathrm{H}}$ are $\{d, 0, \cdots, 0\}$, hence the eigenvalues of $\mathbf{1}_d\mathbf{1}_d^{\mathrm{H}} + \epsilon^2\mathbf{I}_d$ are $\{d+\epsilon^2, \epsilon^2, \cdots, \epsilon^2\}$.

C. Cramer-Rao Bound

In this section we indicate the derivation of the Cramer-Rao bound for the estimates of the packet offset delay. For the derivation, we slightly redefine the data model as $\mathbf{x}(t) = \mathbf{A}\mathbf{z}_{\boldsymbol{\tau}}(t) + \mathbf{n}(t)$ where $\mathbf{x}(t)$ is as before, $\mathbf{A} = [\mathbf{a}^{(1)}, \cdots, \mathbf{a}^{(d)}]$ where $\mathbf{a}^{(i)}$ is the signature vector of source i. $\mathbf{z}_{\boldsymbol{\tau}}(t) = [z^{(1)}(t-\tau^{(1)}), \cdots, z^{(d)}(t-\tau^{(d)})]^T$, where $z^{(i)}(t) = \sqrt{c^{(i)}(t)}e^{j\phi^{(i)}(t)}$ represents for the i-th source the product of a constant modulus (CM) data signal $e^{j\phi^{(i)}(t)}$ and the known modulation code $c^{(i)}(t)$. We further define a vector of user phase rotations as $\boldsymbol{\phi}(t) = [\phi^{(1)}(t), \cdots, \phi^{(d)}(t)]^T$. Finally, $\mathbf{n}(t)$ is a column vector of size M and is assumed to be circularly symmetric zero mean white Gaussian noise with covariance matrix $\mathbf{R}_{nn} = \sigma^2\mathbf{I}$, where σ^2 is the noise variance as received on a single antenna. As usual the noise variance can be estimated separately from the other parameters, and therefore we can assume it is known in the derivation of the CRB.

The unknown parameters are considered to be deterministic constants rather than stochastic variables. After sampling with period T the model becomes

$$\mathbf{x}_k = \mathbf{A}\mathbf{z}_k + \mathbf{n}_k$$

where $\mathbf{x}_k = \mathbf{x}(kT)$, and $\mathbf{z}_k = \mathbf{z}_{\boldsymbol{\tau}}(kT)$. Note that $\mathbf{z}_k$ is a function of $\boldsymbol{\tau}$ but this is not indicated in the notation for reasons of simplicity.

With little loss of generality we assume $T = 1$, i.e., no oversampling. In the case of oversampling the obtained bound will be slightly pessimistic because the model assumes independent phases whereas in the observed data they are dependent.

Based on the model and assuming N received sample vectors collected in a matrix $\mathbf{X}$, we can derive the likelihood function as

$$L(\mathbf{X}|\boldsymbol{\phi}, \mathbf{a}, \boldsymbol{\tau}) = \frac{1}{(\pi\sigma^2)^{MN}} \exp\left\{ -\frac{1}{\sigma^2}\sum_{k=1}^{N}(\mathbf{x}_k - \mathbf{A}\mathbf{z}_k)^{\mathrm{H}}(\mathbf{x}_k - \mathbf{A}\mathbf{z}_k) \right\}, \tag{21}$$

where

$$\boldsymbol{\phi} = \left[\boldsymbol{\phi}_1^T, \ldots, \boldsymbol{\phi}_N^T\right]^T. \tag{22}$$

and

$$\mathbf{a} = \left[\bar{\mathbf{a}}^{(1)T}, \tilde{\mathbf{a}}^{(1)T}, \cdots, \bar{\mathbf{a}}^{(d)T}, \tilde{\mathbf{a}}^{(d)T}\right]^T. \tag{23}$$

Here $\bar{\mathbf{a}}^{(i)} = \mathrm{Re}(\mathbf{a}^{(i)})$ while $\tilde{\mathbf{a}}^{(i)} = \mathrm{Im}(\mathbf{a}^{(i)})$.

Let $\mathcal{L}(\mathbf{X}|\boldsymbol{\phi}, \mathbf{a}, \boldsymbol{\tau}) = \ln L(\mathbf{X}|\boldsymbol{\phi}, \mathbf{a}, \boldsymbol{\tau})$. After omitting constants we obtain the log-likelihood function

$$\mathcal{L}(\mathbf{X}|\boldsymbol{\phi}, \mathbf{a}, \boldsymbol{\tau}) = -\frac{1}{\sigma^2}\sum_{k=1}^{N}\|\mathbf{x}_k - \mathbf{A}\mathbf{z}_k\|^2. \tag{24}$$

The derivation of the Cramer Rao Bound from the log-likelihood function follows along standard lines [25]. Indeed, the CRB is given by the main diagonal of the inverse of the Fisher Information Matrix (FIM), given by

$$\mathbf{F} = E\left\{ \frac{\partial\mathcal{L}}{\partial\boldsymbol{\rho}} \cdot (\frac{\partial\mathcal{L}}{\partial\boldsymbol{\rho}})^T \right\}$$

where $\boldsymbol{\rho}$ is a vector which collects all parameters,

$$\boldsymbol{\rho} = [\boldsymbol{\phi}^T,\ \mathbf{a}^T,\ \boldsymbol{\tau}^T]^T.$$

To specify the entries in closed form, we partition the FIM as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix} \tag{25}$$

where the partitioning follows the partitioning of $\boldsymbol{\rho}$ into $\boldsymbol{\phi}$ followed by $[\mathbf{a}; \boldsymbol{\tau}]$. Then

$$\mathbf{F}_{11} = \begin{bmatrix} \mathbf{H}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{H}_N \end{bmatrix},$$

$$\mathbf{F}_{21} = \begin{bmatrix} \bar{\mathbf{\Delta}}_1^{(1)} & \cdots & \bar{\mathbf{\Delta}}_N^{(1)} \\ \tilde{\mathbf{\Delta}}_1^{(1)} & \cdots & \tilde{\mathbf{\Delta}}_N^{(1)} \\ \vdots & & \vdots \\ \bar{\mathbf{\Delta}}_1^{(d)} & \cdots & \bar{\mathbf{\Delta}}_N^{(d)} \\ \tilde{\mathbf{\Delta}}_1^{(d)} & \cdots & \tilde{\mathbf{\Delta}}_N^{(d)} \\ \mathbf{E}_1 & \cdots & \mathbf{E}_N \end{bmatrix},$$

$$\mathbf{F}_{22} = \begin{bmatrix} \bar{\mathbf{\Gamma}}^{(1)} & -\tilde{\mathbf{\Gamma}}^{(1)} & 0 & 0 & 0 & 0 & \bar{\mathbf{\Lambda}}^{(1)T} \\ \tilde{\mathbf{\Gamma}}^{(1)} & \bar{\mathbf{\Gamma}}^{(1)} & 0 & 0 & 0 & 0 & \tilde{\mathbf{\Lambda}}^{(1)T} \\ 0 & 0 & \ddots & & 0 & 0 & \vdots \\ 0 & 0 & & \ddots & 0 & 0 & \vdots \\ 0 & 0 & 0 & 0 & \bar{\mathbf{\Gamma}}^{(d)} & -\tilde{\mathbf{\Gamma}}^{(d)} & \bar{\mathbf{\Lambda}}^{(d)T} \\ 0 & 0 & 0 & 0 & \tilde{\mathbf{\Gamma}}^{(d)} & \bar{\mathbf{\Gamma}}^{(d)} & \tilde{\mathbf{\Lambda}}^{(d)T} \\ \bar{\mathbf{\Lambda}}^{(1)} & \tilde{\mathbf{\Lambda}}^{(1)} & \dots & \dots & \bar{\mathbf{\Lambda}}^{(d)} & \tilde{\mathbf{\Lambda}}^{(d)} & \mathbf{\Upsilon} \end{bmatrix},$$

where $\mathbf{F}_{12} = \mathbf{F}_{21}^{\mathrm{T}}$ and the submatrices can be derived as

$$\begin{array}{lcll}
\mathbf{H}_k & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k})^T\right\} & = \frac{2}{\sigma^2}\mathrm{Re}(\mathbf{Q}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{Q}_k) \\
\bar{\mathbf{\Delta}}_k^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{a}}^{(i)}}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k})^T\right\} & = -\frac{2}{\sigma^2}\mathrm{Im}(z_k^{(i)*}\mathbf{A}\mathbf{Q}_k) \\
\tilde{\mathbf{\Delta}}_k^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}^{(i)}}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k})^T\right\} & = \frac{2}{\sigma^2}\mathrm{Re}(z_k^{(i)*}\mathbf{A}\mathbf{Q}_k) \\
\mathbf{E}_k & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_k})^T\right\} & = -\frac{2}{\sigma^2}\mathrm{Im}(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathbf{Q}_k) \\
\bar{\mathbf{\Gamma}}^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{a}}^{(i)}}(\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{a}}^{(i)}})^T\right\} & = \frac{2}{\sigma^2}\sum_{k=1}^N \mathrm{Re}(|z_k^{(i)}|^2\mathbf{I}) \\
\tilde{\mathbf{\Gamma}}^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}^{(i)}}(\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}^{(i)}})^T\right\} & = \frac{2}{\sigma^2}\sum_{k=1}^N \mathrm{Im}(|z_k^{(i)}|^2\mathbf{I}) = \mathbf{0} \\
\bar{\mathbf{\Lambda}}^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}(\frac{\partial \mathcal{L}}{\partial \bar{\mathbf{a}}^{(i)}})^T\right\} & = \frac{2}{\sigma^2}\sum_{k=1}^N \mathrm{Re}(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}z_k^{(i)}) \\
\tilde{\mathbf{\Lambda}}^{(i)} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}(\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{a}}^{(i)}})^T\right\} & = -\frac{2}{\sigma^2}\sum_{k=1}^N \mathrm{Im}(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}z_k^{(i)}) \\
\mathbf{\Upsilon} & := & E\left\{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}}(\frac{\partial \mathcal{L}}{\partial \boldsymbol{\tau}})^T\right\} & = \frac{2}{\sigma^2}\sum_{k=1}^N \mathrm{Re}(\mathcal{Z}_k^{\mathrm{H}}\mathbf{A}^{\mathrm{H}}\mathbf{A}\mathcal{Z}_k)
\end{array}$$

(superscript * indicates complex conjugate), and matrix $\mathcal{Z}_k$ is constructed as

$$\mathcal{Z}_k = \mathrm{diag}([\frac{\partial z^{(1)}(t-\tau_1)}{\partial \tau_1}[k] \cdots \frac{\partial z^{(d)}(t-\tau_d)}{\partial \tau_d}[k]])\,.$$

The derivation is straightforward but tedious, and therefore omitted.

To obtain closed-form expressions for the inverse of the FIM, we exploit the block-diagonality of $\mathbf{F}_{11}$ and part of $\mathbf{F}_{22}$, and use the partitioned matrix inversion lemma twice, which leads to the following result. Define

$$\begin{bmatrix} \mathbf{\Xi}_{11} & \mathbf{\Xi}_{12} \\ \mathbf{\Xi}_{21} & \mathbf{\Xi}_{22} \end{bmatrix} := \begin{bmatrix} \sum_{k=1}^N \ddot{\mathbf{\Delta}}_k\mathbf{H}_k^{-1}\ddot{\mathbf{\Delta}}_k^T & \sum_{k=1}^N \ddot{\mathbf{\Delta}}_k\mathbf{H}_k^{-1}\mathbf{E}_k^T \\ \sum_{k=1}^N \mathbf{E}_k\mathbf{H}_k^{-1}\ddot{\mathbf{\Delta}}_k^T & \sum_{k=1}^N \mathbf{E}_k\mathbf{H}_k^{-1}\mathbf{E}_k^T \end{bmatrix}$$

where $\ddot{\mathbf{\Delta}}_k := [\bar{\mathbf{\Delta}}_k^{(1)T}\tilde{\mathbf{\Delta}}_k^{(1)T}, \cdots, \bar{\mathbf{\Delta}}_k^{(d)T}\tilde{\mathbf{\Delta}}_k^{(d)T}]^T$ is of size $[2Md \times d]$ and

$$\mathbf{\Psi} := \begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Lambda}^T \\ \mathbf{\Lambda} & \mathbf{\Upsilon} \end{bmatrix} - \begin{bmatrix} \mathbf{\Xi}_{11} & \mathbf{\Xi}_{12} \\ \mathbf{\Xi}_{21} & \mathbf{\Xi}_{22} \end{bmatrix}. \tag{26}$$

where $\mathbf{\Gamma}, \mathbf{\Lambda}$ and $\mathbf{\Upsilon}$ correspond to the block partition of $\mathbf{F}_{22}$. The CRB for $\boldsymbol{\tau}$ follows as

$$\begin{array}{rcl} CRB(\boldsymbol{\tau}) & = & (\mathbf{\Psi}^{-1})_{22} \\ & = & \mathrm{diag}\left[(\mathbf{\Upsilon} - \mathbf{\Xi}_{22}) - (\mathbf{\Lambda} - \mathbf{\Xi}_{21})(\mathbf{\Gamma} - \mathbf{\Xi}_{11})^{-1}(\mathbf{\Lambda}^T - \mathbf{\Xi}_{12})\right]^{-1}. \end{array} \tag{27}$$

Acknowledgement

We gratefully acknowledge the help of Geert Leus in deriving the Cramer-Rao bound.

References

[1] A. van der Veen and L. Tong, "Packet separation in wireless ad-hoc networks by known modulus algorithms," in *Proc. IEEE ICASSP*, (Orlando, FL, USA), May 2002.

[2] R. Djapic and A. van der Veen, "Blind synchronization in asynchronous multiuser packet networks using KMA," in *IEEE workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, (Rome (Italy)), June 2003.

[3] P. Karn, "MACA—a new channel access method for packet radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–140, ARRL, 1990.

[4] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part II—the hidden terminal problem in carrier sense multiple access and the busy-tone solution," *IEEE Tr. Communications*, vol. 23, pp. 1417–1433, Dec. 1975.

[5] A. van der Veen, "Algebraic methods for deterministic blind beamforming," *Proc. IEEE*, vol. 86, pp. 1987–2008, Oct. 1998.

[6] L. Tong, Q. Zhao, and G. Mergen, "Multipacket reception in random access wireless networks: From signal processing to optimal medium access control," *IEEE Communication Mag.*, Jan. 2002.

[7] M. Tsatsanis, R. Zhang, and S. Banerjee, "Network assisted diversity for random access wireless networks," *IEEE Tr. Signal Proc.*, vol. 48, Sept. 2000.

[8] Q. Zhao and L. Tong, "Semi-blind collision resolution in random access ad hoc wireless networks," *IEEE Tr. Signal Proc.*, vol. 48, Sept. 2000.

[9] J. Treichler and M. Larimore, "New processing techniques based on constant modulus adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, pp. 420–431, Apr. 1985.

[10] A. Orozco-Lugo and D. McLernon, "Blind signal separation for SDMA based on periodically time varying modulation," in *IEE National conference on antennas and propagation*, (York, U.K.), pp. 182–186, Apr. 1999.

[11] A. Orozco-Lugo and D. McLernon, "Blind ISI and MAI cancellation based on periodically time varying transmitted power," *Electronics letters*, vol. 37, July 2001.

[12] S. Talwar, M. Viberg, and A. Paulraj, "Blind estimation of synchronous cochannel digital signals using an antenna array. Part I: Algorithms," *IEEE Trans. Signal Proc.*, vol. 44, pp. 1184–1197, May 1996.

[13] A. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 44, pp. 1136–1155, May 1996.

[14] M. Tsatsanis and G. Giannakis, "Transmitter induced cyclostationarity for blind channel equalization," *IEEE Trans. Signal Proc.*, vol. 45, pp. 1785–1794, July 1997.

[15] E. Serpedin and G. Giannakis, "Blind channel identification and equalization with modulation-induced cyclostationarity," *IEEE Tr. Signal Proc.*, vol. 46, pp. 1930–1944, July 1998.

[16] A. Chevreuil, P. Loubaton, and L. Vandendorpe, "Performance of general transmitter induced cyclostationarity precoders: analysis based on a mmse-df receiver," *IEEE Tr. Signal Proc.*, vol. 48, pp. 3072–3086, Nov. 2000.

[17] A. Chevreuil and P. Loubaton, "MIMO blind second-order identification method and conjugate cyclostationarity," *IEEE Tr. Signal Proc.*, vol. 47, pp. 572–578, Feb. 1999.

[18] H. Bölcskei, R. Heath, and A. Paulraj, "Blind channel estimation in spatial multiplexing systems using nonredundant antenna precoding," in *33rd Asilomar conference on SSC*, vol. 2, pp. 1127–1132, IEEE, 1999.

[19] A. Orozco-Lugo, M. Lara, D. McLernon, and H. Muro-Lemus, "Multiple packet reception in wireless ad hoc networks using polynomial phase modulating sequences," *IEEE Trans. Signal Processing*, vol. 51, pp. 2093–2110, Aug. 2003.

[20] G. Leus, P. Vandaele, and M. Moonen, "Deterministic blind modulation-induced source separation for digital wireless communications," *IEEE Tr. Signal Proc.*, vol. 49, pp. 219–227, Jan. 2001.

[21] B. Agee, "The least-squares CMA: A new technique for rapid correction of constant modulus signals," in *Proc. IEEE ICASSP*, (Tokyo), pp. 953–956, 1986.

[22] A. van der Veen, "Asymptotic properties of the algebraic constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 49, pp. 1796–1807, Aug. 2001.

[23] A. van der Veen, M. Vanderveen, and A. Paulraj, "SI-JADE: An algorithm for joint angle and delay estimation using shift-invariance properties," *IEEE Signal Processing Letters*, vol. 4, pp. 142–145, May 1997.

[24] A. van der Veen, S. Talwar, and A. Paulraj, "A subspace approach to blind space-time signal processing for wireless communication systems," *IEEE Trans.Signal Processing*, vol. 45, pp. 173–190, Jan. 1997.

[25] P. Stoica and A. Nehorai, "MUSIC, Maximum Likelihood and Cramér-Rao bound," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 720–741, May 1989.