# BLIND SYNCHRONIZATION IN ASYNCHRONOUS MULTIUSER PACKET NETWORKS USING KMA

*Relja Djapic and Alle-Jan van der Veen*

Delft Univ. of Technology, Dept. Electrical Eng./DIMES, Mekelweg 4, 2628 CD Delft, The Netherlands

In a system with multiple (interfering) users transmitting data packets asynchronously, we consider to separate the user of interest by blind beamforming. The transmitter gives the signal a unique "color code" by modulating the power of each symbol, in a pattern known to the receiver. The beamformer can then be computed using CMA-like algorithms. In this paper, we consider the associated synchronization problem: the receiver does not know when a packet will arrive and has to detect the onset of a packet. We propose an extension of the Algebraic Known Modulus Algorithm (AKMA) to take this problem into account.

## 1. INTRODUCTION

The increasing popularity of wireless ad-hoc networks such as Bluetooth and WLAN gives rise to problems of interference. Several such systems can be present in the same band. Currently, if two packets are overlapping, both are lost and have to be retransmitted. Obviously, such a scheme will break down if the traffic load becomes too large, unless new approaches to medium access control are considered, e.g., [1].

To combat interference in an uncontrolled environment without changing the MAC layer, we propose to equip the receiver with a small antenna array. The desired transmitter can then be separated from the competing signals by antenna combining (beamforming or interference nulling), see figure 1. To compute the beamformer, the receiver has to know some distinguishing feature of the transmitted packet. Training symbols are of course possible but lead to loss of bandwidth. Instead, we proposed in [2] to give a small modulation to the power of each transmitted symbol, using a pattern ("color code") known to the receiver. Assuming that the unmodulated data stream had a constant modulus, the receiver can then compute the separating beamformer using extensions of the CMA, leading to "known modulus algorithms" (KMAs). The advantage of this technique is that the modulation can be rather small, even below the noise level, since the beamformer is computed by combining all the symbols in the packet.

Several modulation approaches have been proposed. Stochastic techniques such as "transmitter induced cyclostationarity", initially derived for single user blind equalization [3–5] have recently been extended to multi-user convolutive channels [6] and OFDM [7]. An example of a deterministic source separation technique is [8], but it needs multiple transmit antennas per source (spatial redundancy). The amplitude modulations and resulting KMAs considered here are simpler than ACMA etc, do not reduce the

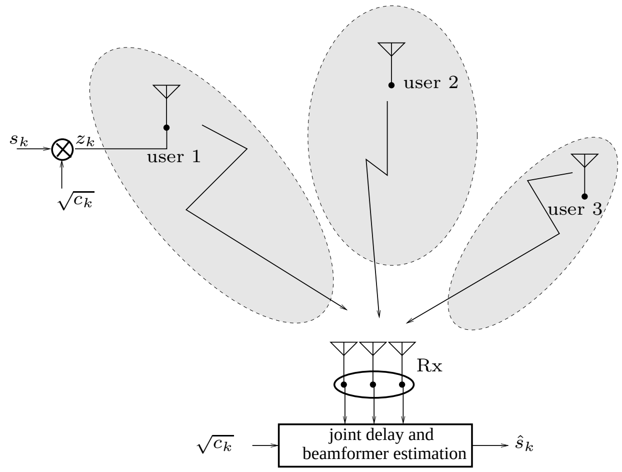


**Figure 1**. Wireless ad-hoc network

capacity, and find only the desired user. This modulation technique was proposed independently by [9, 10].

A problem not considered before is that, in an asynchronous packet network, the receiver does not know precisely when the next packet will arrive. Thus, in a given window of collected samples, it has to try to match the code to the data window at every possible offset, and detect the symbols only if it has found a good match. This is obviously rather inefficient. Instead, in this paper we consider algorithms based on our previous work in delay estimation: after a Fourier transform of the data window, the offset delay becomes a phase shift, which can be estimated using ESPRIT-like techniques. Such algorithms are algebraic and do not contain a combinatorial search. A new aspect compared to ESPRIT is that only a single column in the column span has the expected parametric structure, rather than all columns. This column is computed using a column-span intersection. Since there is only a single parameter, the usual eigenvalue decomposition can then be omitted.

## 2. DATA MODEL

We assume the situation in figure 1 where several users occupy a common wireless channel. The channel is assumed to be narrowband. 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 resulting data model then has the form

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

*This research was supported in part by the Dutch Min. Econ. Affairs/Min. Education Freeband-impulse project *Air-Link*.

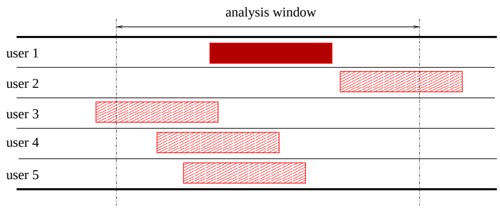


**Figure 2.** Packet transmission structure

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.

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

Data is transmitted in packets, see figure 2. We consider unslotted transmissions, where packets can have arbitrary starting times and fixed or variable packet lengths, The receiver collects $N$ consecutive samples from an analysis window in a data matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] : M \times N$. The number of active users in an analysis window is varying. The packet length of user 1 is denoted by $L$, and we assume that it is known at the receiver. The starting point of the packet of interest within the analysis window is unknown and has to be estimated. We assume that the analysis window contains only one packet of interest: the desired user does not transmit continuously but in intervals longer than $N$.

We will only try to detect packets which are contained completely within the analysis window. E.g., if $N = 2L$, then to ensure this the receiver can shift its window in steps of $L$ samples. Obviously this assumption limits the maximal data rate to 1/2; this can be increased at the expense of more overlap of analysis windows, i.e., more computations at the receiver.

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

$$\mathbf{X} = \mathbf{AS} + \mathbf{N}. \tag{2}$$

$\mathbf{A}$, $\mathbf{S}$ and $\mathbf{N}$ are unknown. The objective is to reconstruct the nonzero part of $\mathbf{s}^{(1)}$ using linear beamforming, i.e., to find a beamformer $\mathbf{w}$ such that $\hat{s}_k = \mathbf{w}^{\mathrm{H}}\mathbf{x}_k$ approximates $s_k^{(1)}$, $k = 1, \cdots, N$.

To distinguish the desired source from the other users, we give it a "color code", in the form of a known pseudo-random modulus variation. 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 (see figure 3). For notational convenience, we assume that $c_k = 0$ outside the support of the packet. The other users may or may not have this form of modulation. The data model (2) is replaced by

$$\mathbf{X} = \mathbf{AZ} + \mathbf{N}.$$

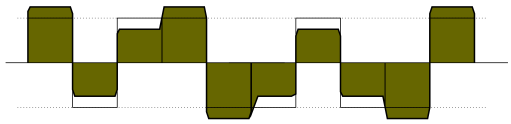

**Figure 3.** Constant modulus signal with coded amplitude variations

The objective of the beamformer is to find $\mathbf{w}$ such that $\hat{z}_k = \mathbf{w}^{\mathrm{H}}\mathbf{x}_k$ approximates $z_k^{(1)}$, $k = 1, \cdots, N$. This problem was considered in [2]. A complication not studied there is the fact that the receiver actually does not know the offset $\tau$ of the code, i.e., we have to compute both $\mathbf{w}$ and $\tau$ such that

$$|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 = |\hat{z}_k|^2 = c_{k-\tau}, \qquad k = 1, \cdots, N. \tag{3}$$

where we have used the assumption that $|s_k|^2 = 1$. With noise, we try to minimize the difference. This is the problem considered in this paper. We assume for simplicity that packets are delayed by an integer number of symbol intervals (the algorithm is easily extended).

## 3. JOINT OFFSET AND BEAMFORMER ESTIMATION

Let user 1 with code $\{c_k,\ k = 1, \cdots, L\}$ be the user of interest. We will extend the AKMA in [2] to estimate the starting point of the data packet of interest as well as the beamforming vector $\hat{\mathbf{w}}$. Rewriting (3), our aim is to compute

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

Here,

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

and $\mathbf{P} = (\bar{\mathbf{X}} \circ \mathbf{X})^{\mathrm{H}}$, where $\circ$ denotes a column-wise Kronecker product: $\bar{\mathbf{X}} \circ \mathbf{X} = [\bar{\mathbf{x}}_1 \otimes \mathbf{x}_1, \cdots, \bar{\mathbf{x}}_N \otimes \mathbf{x}_N]$. We used that $|\mathbf{w}^{\mathrm{H}}\mathbf{x}_k|^2 = \mathbf{w}^{\mathrm{H}}\mathbf{x}\mathbf{x}^{\mathrm{H}}\mathbf{w} = (\bar{\mathbf{x}} \otimes \mathbf{x})^H(\bar{\mathbf{w}} \otimes \mathbf{w})$.

### 3.1. AKMA

Consider first that $\tau$ is known (later we propose the method for its estimation). In that case, we proceed as in [2] and compute $\mathbf{w}$ by the 2-step optimization (hence slightly sub-optimal)

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

If $\mathbf{P}$ would have full column rank, the first problem has a unique solution, $\mathbf{y} = \mathbf{P}^\dagger\mathbf{c}_\tau$. $\mathbf{P}$ can be made full rank by a prewhitening step with dimension reduction [2]. After obtaining $\hat{\mathbf{y}}$, the second optimization problem is solved by rearranging the vector $\hat{\mathbf{y}}$ into a matrix $\hat{\mathbf{Y}}$, and solving

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

Hence, $\hat{\mathbf{w}}$ is computed up to scaling as the dominant eigenvector of $\hat{\mathbf{Y}}$.

### 3.2. Offset estimation

In the previous subsection, we assumed that the receiver is synchronized to the user of interest. We now extend the algorithm to also estimate the offset of the packet in the data block. For this, we exploit the fact that a delay in time domain corresponds to a phase shift in frequency domain. This can be expressed as

$$\mathbf{F}\mathbf{c}_\tau = \mathbf{F}\mathbf{c}_0 \odot \boldsymbol{\phi}_\tau \tag{4}$$

where $\mathbf{F}$ is the $N \times N$ Fourier transformation matrix,

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

and $\varphi = e^{-j\frac{2\pi\tau}{N}}$. The vector $\mathbf{c}_0$ is the unshifted code vector followed by $N-L$ zeros, and $\odot$ represents a pointwise multiplication (Schur-Hadamard product). Our objective will be to estimate $\varphi$ based on the shift-invariance structure exhibited by the vector $\boldsymbol{\phi}$. This then immediately determines the offset $\tau$. A similar approach was considered in the SI-JADE algorithm [11] for joint angle-delay estimation.

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

$$\begin{aligned} \mathbf{F}\mathbf{P}\mathbf{y} &= \mathbf{F}\mathbf{c}_0 \odot [1 \; \varphi \; \varphi^2 \; \ldots \; \varphi^{N-1}]^{\mathrm{T}} \Leftrightarrow \\ \mathbf{P}_f\mathbf{y} &= \mathbf{g} \odot \boldsymbol{\phi}_\tau \end{aligned} \tag{5}$$

where $\mathbf{P}_f = \mathbf{F}\mathbf{P}$ and $\mathbf{g} = \mathbf{F}\mathbf{c}_0$ are known while $\mathbf{y}$ and $\boldsymbol{\phi}_\tau$ have to be estimated. Dividing the rows of $\mathbf{P}_f$ with the corresponding entries of the vector $\mathbf{g}$ we arrive at

$$\tilde{\mathbf{P}}\mathbf{y} = \boldsymbol{\phi}_\tau \tag{6}$$

where $\tilde{\mathbf{P}} = \mathrm{inv}(\mathrm{diag}(\mathbf{g}))\mathbf{P}_f$ is known and the vector $\boldsymbol{\phi}_\tau$ is a known function of the unknown delay $\tau$.

The above pointwise division puts a constraint on the code: the code should be designed such that (after zero padding to length $N$) it does not contain zeros in the frequency domain. Low values of the DFT of the code vector can increase the noise and result in poor estimates. During code design, we can put a threshold value for the lowest allowed value of vector $\mathbf{g}$. We have chosen Gold sequences in order to minimize the cross correlation between the codes of different users.

Equation (6) can be treated in several different ways. Essentially we have to search for the (unique) vector in the column span of $\tilde{\mathbf{P}}$ that has a shift invariance structure. Obviously a MUSIC-type search is applicable: if $\mathbf{U}_s$ is a basis for the signal space of $\tilde{\mathbf{P}}$, then

$$\hat{\tau} = \underset{\tau}{\operatorname{argmax}} \; \|\boldsymbol{\phi}_\tau^{\mathrm{H}}\mathbf{U}_s\|^2 \tag{7}$$

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

$$\begin{aligned} \tilde{\mathbf{P}}_x\mathbf{y} &= [1 \; \varphi \cdots \varphi^{N-2}]^{\mathrm{T}} \\ \tilde{\mathbf{P}}_y\mathbf{y} &= [\varphi \; \varphi^2 \cdots \varphi^{N-1}]^{\mathrm{T}} \end{aligned} \tag{8}$$

Letting $*$ represent a complex conjugation, this can also be written as $\tilde{\mathbf{P}}_x\mathbf{y} = \varphi^*\tilde{\mathbf{P}}_y\mathbf{y}$, which (because $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$ are tall) is a matrix pencil problem. To solve it, we must first find the common column span of $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$.

*Algorithm 1* The simplest technique for this is to compute the SVD of $[\tilde{\mathbf{P}}_x \quad \tilde{\mathbf{P}}_y]$. Indeed, from equation (8), we see that

$$[\tilde{\mathbf{P}}_x \quad \tilde{\mathbf{P}}_y] \begin{bmatrix} \mathbf{y} \\ -\varphi^*\mathbf{y} \end{bmatrix} = \mathbf{0}\,. \tag{9}$$

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 the solutions in the intersection of the column span of $\tilde{\mathbf{P}}_x$ with that of $\tilde{\mathbf{P}}_y$. In general, if $\mathbf{P}$ is full column rank, we expect only a single solution $\mathbf{v}_n$, which then will have the form

$$\mathbf{v}_n =: \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ -\varphi^*\mathbf{y} \end{bmatrix} \tag{10}$$

After finding $\mathbf{v}_n$, we can estimate the phaseshift $\varphi$ as $\hat{\varphi} = -(\mathbf{v}_x^\dagger\mathbf{v}_y)^*$, which directly specifies the coarse delay estimate $\hat{\tau}$. In the vicinity of the coarse estimate a MUSIC-type search (7) is performed in order to obtain more accurate result.

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

$$\begin{aligned} \tilde{\mathbf{P}}_x &= \hat{\mathbf{U}}_x\hat{\boldsymbol{\Sigma}}_x\hat{\mathbf{V}}_x^{\mathrm{H}} \\ \tilde{\mathbf{P}}_y &= \hat{\mathbf{U}}_y\hat{\boldsymbol{\Sigma}}_y\hat{\mathbf{V}}_y^{\mathrm{H}} \end{aligned} \tag{11}$$

Similar to (9) we find

$$[\hat{\mathbf{U}}_x \quad \hat{\mathbf{U}}_y] \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_x\hat{\mathbf{V}}_x^{\mathrm{H}}\mathbf{y} \\ -\varphi^*\hat{\boldsymbol{\Sigma}}_y\hat{\mathbf{V}}_y^{\mathrm{H}}\mathbf{y} \end{bmatrix} = 0 \tag{12}$$

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:

$$\mathbf{v}_n =: \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_x\hat{\mathbf{V}}_x^{\mathrm{H}}\mathbf{y} \\ -\varphi^*\hat{\boldsymbol{\Sigma}}_y\hat{\mathbf{V}}_y^{\mathrm{H}}\mathbf{y} \end{bmatrix} \tag{13}$$

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

*Algorithm 2* Another algorithm for subspace intersection is mentioned in [12]: 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 $\varphi$, 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 $\varphi$ 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 $\hat{\varphi} = \mathbf{u}_y^\dagger\mathbf{u}_x$. It is possible to obtain a better estimate of $\varphi$ by the additional limited search using the complete known structure of $\boldsymbol{\phi}$.

### 3.3. Postprocessing: alternating projections

The estimated beamformer can be improved by a few iterations of an alternating projection algorithm:

$$\begin{bmatrix} \mathbf{y} := \mathbf{w}^{\mathrm{H}}\mathbf{X} \\ \hat{z}_k := \frac{y_k}{|y_k|}\sqrt{c_k}\,, \qquad k = 1, \cdots, N \\ \mathbf{w} := (\hat{\mathbf{z}}\mathbf{X}^\dagger)^{\mathrm{H}} \end{bmatrix}$$

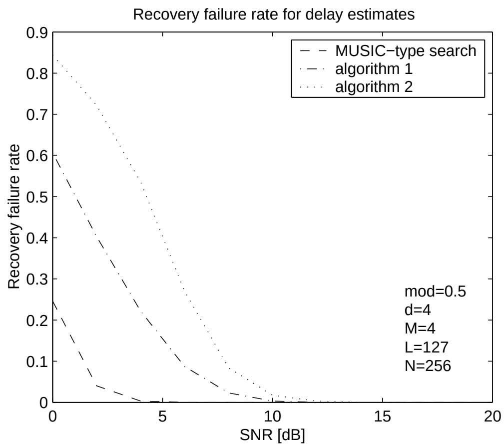


**Figure 4**. Failure rate for the offset estimation

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 good initial point, this algorithm is stable and converges usually nicely.

## 4. SIMULATIONS

In the simulations we consider $d = 4$ users, $M = 4$ antennas in a uniform linear array with half-wavelength spacing. All users have the same transmitting power, and they are arriving at the ULA at $-10^\circ$ , $20^\circ$ , $40^\circ$ , $-40^\circ$ with respect to the array broadside. The packet length is $L = 127$ while the analysis window size is $N = 256$. All sources are transmitting unit amplitude constant modulus signals modulated by a power modulation with index $\epsilon = .5$. After coarse estimate of the packet offset $\hat{\tau}$ is obtained an additional MUSIC-type search is performed in the $[\hat{\tau}-10, \hat{\tau}+10]$ region. Figure 4 shows the percentage of inaccurate delay estimates for the three algorithms proposed.

Figure 4 shows the percentage of cases where the delay offset was not estimated correctly, as a function of the input SNR. An estimate is labeled as failure if its rounded value is not equal to the true (integer) delay offset. 1000 Monte-Carlo runs for each value of the input SNR were performed. From the figure we see that the MUSIC search over all possible delays (7) performs better than the limited MUSIC-search in the vicinity of the coarse estimate $\hat{\tau}$ obtained using algorithms 1 and 2.

For the cases without failure, figure 5 presents the signal to interference and noise ratio (SINR) at the output, i.e., after beamforming. The solid line is a theoretical reference line showing the performance of the Wiener beamformer assuming the transmitted signal, code and offset of the user of interest are known. The three other lines represent the performance after 30 iterations of the alternating projection algorithm, initialized by MUSIC, algorithm 1 and algorithm 2, respectively.

Similarly, figure 6 presents the root mean square error (RMSE) of the estimated data sequence for each of the proposed algorithms, where the error is $\|(\hat{\mathbf{z}}_i - \mathbf{z}_i)\|/\sqrt{L}$, and the average is computed only over those cases where the delay offset was computed accurately.

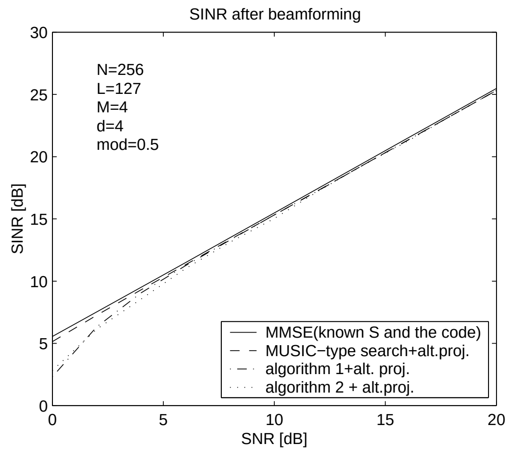


**Figure 5**. SINR after beamforming for user 1

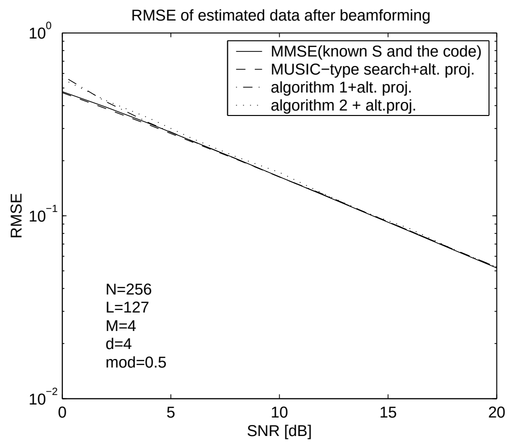


**Figure 6**. MSE of estimated data for user 1

## 5. DISCUSSION

We have introduced several algorithms for delay offset estimation and beamformer calculation. These are tentative algorithms and more research is needed to resolve the following issues:

1. At several places in the algorithms, we assumed full rank conditions. However, singularities can occur. E.g., if two sources are not completely overlapping, $\mathbf{P}$ is not full rank [2]. Another case where $\mathbf{P}$ is not full rank is the case where two users have the same code (in particular, if two users are constant modulus and are not code-modulated) [2]. If $\mathbf{P}$ is not of full column rank, then there will exist additional solutions $\mathbf{y}_0$ to $\mathbf{P}\mathbf{y}_0 = \mathbf{0}$ which will add to the desired solution $\mathbf{y} = \bar{\mathbf{w}}_1 \otimes \mathbf{w}_1$, producing a result that cannot be factored. In the present algorithms, we also have assumptions on the intersection of $\tilde{\mathbf{P}}_x$ and $\tilde{\mathbf{P}}_y$: we assumed there is only 1 vector in the intersection. This seems to hold in simulations, but has to be analyzed further.
2. The algorithms have different properties and accuracies. These have to be studied further. More efficient algorithms can be derived especially for delay estimation at low SNR.
3. Code design: as mentioned, the zero-padded code

should have no zeros on the unit circle. Also, perhaps the pointwise division by the transformed code can be avoided, at least in the MUSIC-type estimator (7) this is very simple. Alternatively, in subsequent steps weighted estimates are needed.

## References


[1] M. K. Tsatsanis, R. Zhang, and S. Banerjee, "Network Assisted Diversity for Random Access Wireless Networks," *IEEE Trans. Signal Processing*, vol. SP-48, pp. 702–711, March 2000.

[2] 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.

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

[4] 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.

[5] 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.

[6] A. Chevreuil and P. Loubaton, "MIMO blind second-order identification method and conjugate cyclostationarity," *IEEE Trans. Signal Processing*, vol. 47, pp. 572–578, feb 1999.

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

[8] 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.

[9] A. Orozco-Lugo and D. McLernon, "Blind signal separation for SDMA based on periodically time varying modulation," in *Proc. National conference on antennas and propagation*, (U.K.), Apr. 1999.

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

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

[12] 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.