Applicability of blind beamforming techniques to FMA-1 with spreading

Alle-Jan van der Veen
Delft University of Technology, Dept. Electrical Engineering/DIMES, 2628 CD Delft, The Netherlands

Abstract—The purpose of adaptive antenna techniques is to separate and equalize superpositions of source signals impinging on a phased antenna array. The required space-time equalizer coefficients are obtained from deterministic properties of the signals such as known training bits, known spreading codes, Toeplitz structures and/or constant modulus properties. Alternatively, we can assume a parametric channel model and estimate directions-of-arrival and propagation delays. Within the European ACTS program, the FRAMES project is responsible to define a candidate multiple access system for UMTS. We will look at the proposed FMA-1 modulation format with spreading, discuss the properties that are available for estimating the equalizer coefficients, and indicate generic algorithms that exploit them.

1. INTRODUCTION

In the context of array signal processing, beamforming is concerned with the reconstruction of source signals from the outputs of an sensor array. This can be done either by coherently adding the contributions of the desired source, or by nulling out the interfering sources. The latter is an instance of the more general problem of source separation.

Classically, beamforming requires knowledge of a look direction, which is the direction of the desired source. Currently, “blind beamforming” is a hot topic within signal processing: the idea is to separate sources without direction information, relying instead on various structural properties of the problem. Even more recent is “semi-blind beamforming”, in which the structure provided by known training sequences is also taken into account.

The first blind beamforming techniques proposed were based on direction finding. The direction of each incoming wavefront is estimated, at the same time producing a beamformer to recover the signal from that direction. This requires at least that the antenna array is calibrated. If a source comes in via several directions (coherent multipath), then direction finding is more complicated. Depending on the situation, we also need to consider delay spread. Thus, the applicability of these techniques is much dependent on the channel conditions and in general requires a small number of well defined propagation paths per source (a parametric channel model).

More recently, new types of blind beamformers have been proposed that are not based on specific channel models, but instead exploit properties of the signals. A striking example is the constant modulus algorithm (CMA), which separates sources on the fact that their baseband representation has a constant amplitude, such as is the case for FM or phase modulated signals. A prime advantage is that these beamformers are not dependent on channel properties or array calibration, which makes them very robust. Several other properties are often available, for example cyclostationarity caused by the bauded nature of digital communication signals or introduced by small differences in carrier frequencies.

In this paper, we will look at the applicability of such techniques to the FRAMES Multiple Access format, mode 1 (TDMA).

This is a rather diverse format. To limit the discussion, we will look only at the FMA-1 mode with spreading, which is most interesting from a signal processing point of view. In this mode, the slots have fixed size, up to 8 cochannel users are allowed in a single slot, and the users modulate their symbols with short orthogonal codes of length 16. The symbol alphabet is either 4-PSK or 16-QAM. A linearized GMSK pulse shape provides for some constant modulus structure (limited by the symbol alphabet), and training symbols are available as well. The delay spread necessitates equalization of up to 2-3 symbols, or 33 chips. Altogether, a rich structure is available. The objective is to find space-time beamforming coefficients to jointly separate the 8 cochannel users and to equalize them.

2. FMA1-SPREAD MODULATION FORMAT

The FRAMES Multiple Access format mode 1 with spreading is described in document [1]. For our purposes here, it can be summarized as follows.

FMA1-spread is a TDMA scheme with 8 time slots per frame. The length of a time slot is 577 µs. The symbol period is $T_s = 7.39µs$. Each symbol is modulated with a code of length $Q = 16$. Thus, the chip period is $T_c = 0.46µs$ (more precisely, the slot has length 1250 chips, approximately 78 symbols). There are 16 different orthogonal codes, but only 8 are used, to accommodate 8 synchronous cochannel users. The time slot consists of 2*28 symbols, and a training m-damble of 296 chips (an equivalent of 18.5 symbols). The remaining part of the slot is used as a guard period. The training sequences of the 8 users are selected in a special way, which allows channel estimation using properties of cyclic matrices [2].

The delay spread is considered to be less than 15 µs, corresponding to 32 or 33 chips, or 2 symbol periods. As in GSM, the doppler shift is sufficiently low in comparison to the length of the time slot so that the channel can be regarded stationary within the slot, but not from one frame to the next.

The pulse modulation function $c_0(t)$ is a linearized GMSK pulse of length $5T_c$ [1], see figure 1. The chips in a chip sequence are chosen alternatingly real $(\pm 1)$ and imaginary $(\pm j)$: for the $i$-th user,

$$c_i = [c_{i0} \ldots c_{i15}] = [\pm 1 \pm j \ldots \pm 1 \pm j]$$

$$c_i(t) = \sum_{k=0}^{15} c_{ik} \delta(t-kT_c)$$

The symbol modulating function $g_i(t)$ is thus

$$g_i(t) = c_i(t) \ast c_0(t)$$

and it has approximately a constant modulus over a single symbol period.
The symbol alphabet is either 4PSK, i.e., $\Omega = \{\pm 1, \pm j\}$, or 16-QAM, i.e., $\Omega = \{\pm 1, \pm 2\} \oplus \{\pm j, \pm 2j\}$. The transmitted signal for the $i$-th user then has the form

$$z_i(t) = d_i(t) \ast c_i(t) \ast c_0(t),$$

where

$$d_i(t) = \sum_{k=1}^{N} d_{ik} \delta(t-kT_s), \quad d_{ik} \in \Omega_i.$$

Here, $d_{ik}$ denotes the $k$-th symbol of the $i$-th user and $N$ is the total number of transmitted symbols in a data block. (In fact, this is not an entirely correct representation because the training sequence does not have the same form: it is an “arbitrary” sequence of length 296 chips, alternatingly real $(\pm 1)$ and imaginary $(\pm j)$.)

### 3. CHANNEL MODEL

The propagation of signals through a radio channel is fairly complicated to model. A correct treatment would require a complete description of the physical environment, not very suitable for the design of signal processing algorithms. To arrive at a more useful parametric model, we have to make simplifying assumptions regarding the wave propagation. Provided this model is reasonably valid, we can, in a second stage, try to derive statistical models for the parameters to obtain agreement with measurements.

In the present section, we will look at some general data models as used in the signal processing community. In the next section, we will then give an overview of the types of structure that may be available and (blind) beamforming algorithms that use this structure. We will then come back to the FRAMES modulation format and discuss how the general techniques apply to it.

#### 3.1. Instantaneous mixtures

Assume that $d$ source signals $s_1(t), \ldots, s_d(t)$ are transmitted from $d$ independent sources at different locations. If the delay spread is small, then we will receive a simple linear combination of these signals:

$$x(t) = a_1 s_1(t) + \cdots + a_d s_d(t)$$

where $x(t)$ is a stack of the output of the $M$ antennas. Suppose we collect a batch of $N$ samples, then

$$X = AS, \quad A = [a_1 \cdots a_d],$$

where $X = [x(0) \cdots x(N-1)]$ and $S = [s(0) \cdots s(N-1)]$. The resulting $X = AS$ model is called an instantaneous multi-input multi-output model, or I-MIMO for short. It is a generic of the dominant rays is much smaller than the inverse bandwidth of the signals.

The objective of beamforming for source separation is to construct a left-inverse $W$ of $A$, such that $WA = I$ and hence $WX = S$: see figure 2(a). This will recover the source signals from the observed mixture. It immediately follows that in this scenario it is necessary to have $d \leq M$ to ensure interference-free reception, i.e., not more sources than sensors. If we know already (part of) $S$, e.g., because of training, then $W = SX^\dagger$, where $X^\dagger$ denotes the Moore-Penrose pseudo-inverse of $X$. Blind beamforming is to find $W$ with knowledge only of $X$.

If each source is received from only a single direction (no multipath), then the columns of $A$ can be described by the array response vector $a(\theta)$. E.g., for a uniform linear array and a single source,

$$x(t) = \begin{bmatrix} s(\theta) \end{bmatrix} = [1, \theta, \cdots, \theta^{M-1}] s(t), \quad \theta = e^{j2\pi \Delta \sin(\alpha)}$$

where $\alpha$ is the direction of the source and $\Delta$ is the spacing between the elements of the array (in wavelengths). Without multipath, the columns of $A$ lie on the array manifold $\{a(\theta) : |\theta| = 1\}$.

#### 3.2. Convolutive mixtures

An often-used parametric channel model that is valid for wideband sources is

$$x(t) = \sum_{i=1}^r a(\theta_i) g(t-\tau_i) \ast s(t) = h(t) \ast s(t).$$

Here, it is assumed that the source is digital (more precisely, a dirac-pulse sequence), linearly modulated by a pulse shape function $g(t)$. The channel is supposed to be a simple multi-path propagation channel, consisting of $r$ distinct paths, each parametrized by a direction $\theta_i$, a relative path delay $\tau_i$, and a complex amplitude (fading) $\beta_i$. The channel has finite length $L$ symbols.

Suppose that the pulse-shape function $g(t)$ has support (length) $L_s$, augmented with zeros to length $L$, and that we sample at a rate $P$ times the symbol rate. We can then define the temporal signature vector

$$g(\tau) := \begin{bmatrix} g(0-\tau) \\ g(\frac{1}{P}-\tau) \\ \vdots \\ g(L_s-\frac{1}{P}-\tau) \end{bmatrix}.$$  

It is thus seen that $h(t) = \sum_i a(\theta_i) \beta_i g(t-\tau_i)$ has structure: let $g = g(\tau_i), a_i = a(\theta_i)$, then

$$h := \begin{bmatrix} h(0) \\ \vdots \\ h\left(\frac{1}{P}\right) \\ \vdots \\ h(L_s-\frac{1}{P}) \end{bmatrix} = [g_1 \otimes a_1, \cdots, g_r \otimes a_r] \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_r \end{bmatrix}.$$  

The combined vector $g(\tau) \otimes a(\theta)$ is the space-time response vector. ($\otimes$ denotes a Kronecker product.)
After collecting data samples during $N$ symbol periods, the convolutive model $x(t) = h(t) \ast s(t)$ can be written in matrix form as $X = HS_L$, where

$$X = \begin{bmatrix} x(0) & x(1) & \cdots & x(N-1) \\ x(\frac{L}{S}) & x(1+\frac{L}{S}) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x\left(\frac{L}{S}m\right) & \cdots & x(L-1) \end{bmatrix}$$

$$H = \begin{bmatrix} h(0) & \cdots & h(L-1) \\ h\left(\frac{L}{S}\right) & \cdots & \vdots \\ \vdots & \ddots & \vdots \\ h\left(\frac{L}{S}m\right) & \cdots & h\left(\frac{L}{S}(L-1)\right) \end{bmatrix}$$

$$S_L = \begin{bmatrix} s_0 & s_1 & \cdots & s_{N-2} & s_{N-1} \\ s_{-1} & s_0 & \cdots & s_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ s_{-L+1} & \cdots & s_{N-L} \end{bmatrix}$$

This factorization is typical for a linear FIR channel (single-source). Note in particular that $S_L$ has a Toeplitz structure (constant along diagonals), which is a consequence of the time-invariance of the channel. With multiple sources, a similar model holds, but $S_L$ becomes a block-Toeplitz structure with vector-entries $s_k$.

An equalizer in this context can be written as a vector $w$ which combines the rows of $X$ to generate an output $y = w^\top X$. In the model so far, we can only equalize among the antenna outputs (simple beamforming) and among the $P$ samples within one sample period (polyphase combining). More in general, we would want to filter over multiple sample periods, leading to a space-time equalizer. For an equalizer length of $m$ symbol periods, we have to augment $X$ with $m-1$ horizontally shifted copies of itself:

$$X_m = [X \ \ \ \ \ \ XH]$$

Each column of $X_m$ is a regression vector: the memory of the filter. Using $X_m$, a general space-time equalizer can be written as $y = w^\top X_m$, which combines $mP$ snapshots of $M$ antennas: see figure 2(b). The augmented data matrix $X_m$ has a factor-

$$X_m = H S_{L+m-1}$$

$$= \begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix}$$

where $H = H_m$ has size $mMP \times (L + m - 1)$ and the $m$ shifts of $H$ to the left are each over 1 position. $H$ has a block-Hankel structure: it is constant along antidiagonals. $S_{L+m-1}$ has the same structure as $S_L$. A necessary condition for space-time equalization ($y = w^\top X$) is that $H$ is tall, which gives minimal conditions on $m$ in terms of $M, P, L$ [3, 4]. Unlike spatial beamforming, it will not be necessary to find $H$; it suffices to reconstruct a single block row of $S$, which can be done with $d$ space-time equalizers $w$. Other equalizer structures than FIR filters are possible, e.g., by using feedback, but are not discussed here.

4. Principles of Blind Beamforming

A summary of the data model developed so far is

I-MIMO: $X = AS, \quad A = [a(\theta_1), \ldots, a(\theta_d)]$

FIR: $X = HS, \quad h = \text{vec}(H) = [g_1 \otimes a_1, \ldots, g_d \otimes a_d] \beta$.

The first part of these model equations is generally valid for LTI channels, whereas the second part is a consequence of the adopted multiray model.

Based on this model, the received data matrix $X$ has several structural properties. In several combinations, these are often strong enough to allow to find the factors $A$ (or $H$) and $S$ (or $S_L$), from knowledge of $X$ alone. A number of properties are discussed below.

4.1. Training

If training symbols are present, then a number of columns of $S$ or $S_L$ are known. This number should be such that this known submatrix $S_i$ is a wide matrix, in which case it generally has a right inverse $S_i^T$. This directly allows estimation of $A$ or $H$ as $X S_i^T$, where $X_i$ is the corresponding window of the data matrix. With $A$ or $H$ known, there are a large number of suitable space-time equalizers (e.g., zero-forcing, MMSE, decision-feedback), differing in performance, complexity and symbols/noise assumptions. The literature is abundant; see e.g., [5, 6] and references therein.

4.2. Toeplitz structure

The fixed baud rate of communication signals, along with time invariance, result in the fact that $X$ has a factorization in
The temporal structure relates to \( s(t) \) as well, but now with regard to its temporal properties. These can include knowledge of its pulse shape function and, in the case of CDMA signals, knowledge of the source codes, but also certain statistical properties for sources that are temporally non-white.

**CDMA codes.** In direct-sequence CDMA, the emitted `chip symbols’ \( s_k \) are in fact modulations of low-rate source symbols \( d_n \) by known code vectors \( \mathbf{c} \) of length \( Q \):

\[
[s_k \cdots s_{k+Q-1}] = d_n \mathbf{c} = d_n [c_1 \cdots c_Q], \quad n = \lfloor k/Q \rfloor.
\]

The codes vectors are different for each source.) Because the only unknowns are the \( d_n \), this reduces the number of unknowns in \( \mathbf{S} \) by a factor \( Q \). The source symbols can e.g., be recovered by row span template matching techniques [22–24], essentially straightforward least squares algorithms.

**Temporally non-white and independence.** If the sources are independent and temporally non-white, separation is possible by using the fact that the cross-covariance and cross-cumulants of the signals at the output of the beamformer should be zero for all time lags. This allows to separate sources, but in this form cannot be used to equalize them. Often, already the second-order conditions are sufficient to find the beamformer; an example of an algebraic technique is [25].

**Cyclostationarity.** Many signals exhibit cyclo-stationary properties, i.e., their cyclic autocorrelation function \( R^c_{xy}(\tau) = E(x(t)x(t-\tau)^* e^{-j2\pi\alpha\tau}) \) is wide-sense stationary and has spectral lines at selective lags \( \tau \) and frequencies \( \alpha \). This is typically caused by periodicities such as the symbol rate in baued communication signals, or residual carrier frequencies. If two sources have spectral peaks for different \((\alpha, \tau)\), then they can be separated based on this [26]. It is usually required that these parameters are known, although they can be estimated in specific cases.

For digital communication signals, a straightforward way in which the cyclostationarity property can be expressed is by oversampling the antenna outputs. The samples obtained during one symbol period presumably give independent linear combinations of the same transmitted bits, just as antennas give independent linear combinations from sampling in space. This fact was noted first in [7] and has stirred a lot of interest since; see e.g. [8, 9, 3]. Although initially called a second-order technique, the Toeplitz structure is a deterministic rather than a stochastic property.

It is also possible to design modulation formats specifically to contain cyclo-stationary properties. There are several possibilities, e.g., by simply repeating the data twice, or inserting zero symbols at periodic locations [27–30].

### 4.5. Parametric properties

Parametric properties relate to the multipath model that we have derived, and extensions of this. It makes sense to use such models if the number of parameters is much smaller than, e.g., the number of coefficients in an unstructured FIR model.

**The spatial manifold.** In the I-MIMO model, each column of \( A \) is a linear combination of array response vectors \( \{\mathbf{a}(\theta)\} \), each of which is on the array manifold. If the array manifold is known, e.g., by calibration or from structural considerations, then we can try to fit the column span of \( X \) (hence \( A \)) to the
transmitted signal:  
\[ z_i(t) = d_i(t) \ast c_i(t) \ast c_0(t) = d_i(t) \ast g_i(t) \]

symbol pulse shape:  
\[ g_i(t) = c_i(t) \ast c_0(t). \]

The data is transmitted and subsequently received by \( M \) antennas. The outputs of these antennas are stacked in \( M \)-dimensional vectors \( \mathbf{x}(t) \). For a linear channel, \( \mathbf{x}(t) \) is the superposition of each of the \( d = 8 \) user responses \( \mathbf{s}_i(t) \). If the corresponding physical channel impulse response is denoted by \( \mathbf{h}_i(t) \), then

received signal:  
\[ \mathbf{x}_i(t) = d_i(t) \ast c_i(t) \ast c_0(t) \ast \mathbf{h}_i^T(t) = s_i(t) \ast \mathbf{h}_i(t) = d_i(t) \ast \mathbf{b}_i(t) \]

where

chip channel:  
\[ \mathbf{b}_i(t) = c_0(t) \ast \mathbf{h}_i^T(t) \]

is the channel as seen by the chipped data sequence, and

symbol channel:  
\[ \mathbf{b}_i(t) = g_i(t) \ast \mathbf{h}_i^T(t) = c_i(t) \ast \mathbf{h}_i(t) \]

is the channel for the symbol sequence.

If the channel can be represented by a discrete multipath model consisting of \( r_i \) paths with angles \( \theta_{ik} \), attenuations \( \beta_{ik} \) and delays \( \tau_{ik} \), then we further have

\[ \mathbf{b}_i(t) = \sum_{k=1}^{r_i} \alpha(\theta_{ik}) \beta_{ik} g_i(t - \tau_{ik}) \]

\[ \mathbf{h}_i(t) = \sum_{k=1}^{r_i} \alpha(\theta_{ik}) \beta_{ik} c_0(t - \tau_{ik}). \]

5.2. Data matrix factorization

The above convolutive models for \( \mathbf{x}(t) \) provide several possible factorizations of the data matrices.

The physical channel length is 2 symbols, but every symbol is spread by the code over the full symbol period, so the effective channel length is \( L = 3 \): every received sample at the antennas is some linear combination of (at most) 3 symbols per user, hence \( 3d = 24 \) symbols in total. Suppose that we sample at the chip rate.\(^1\)

Then we can collect a data matrix

\[
X = \begin{bmatrix}
\mathbf{x}(0) & \mathbf{x}(T_c) & \cdots \\
\mathbf{x}(T_c) & \mathbf{x}(2T_c) & \cdots \\
\vdots & \vdots & \ddots \\
\mathbf{x}(15T_c) & \cdots & \cdots 
\end{bmatrix}
\]

If we sample during \( N \) symbol periods, then \( X \) has size \( 16M \times N \). It has a factorization

\[ X = BD \]

where

\[
B = \begin{bmatrix}
B(0) & B(T_c) & B(2T_c) \\
B(T_c) & \cdots & \cdots \\
\vdots & \vdots & \ddots \\
B(15T_c) & \cdots & B(2+15T_c) 
\end{bmatrix}_{16M \times 3d}
\]

\[
D = \begin{bmatrix}
\mathbf{d}_0 & \mathbf{d}_1 & \mathbf{d}_2 & \cdots \\
\ast & \mathbf{d}_0 & \mathbf{d}_1 & \mathbf{d}_2 & \cdots \\
\ast & \ast & \mathbf{d}_0 & \mathbf{d}_1 & \mathbf{d}_2 & \cdots 
\end{bmatrix}_{3d \times N}
\]

\(^1\)More in general, we would sample at \( P \) times the chip rate to increase dimensionality and alleviate synchronization requirements. Typical would be \( P = 2 \).
This representation is at the chip level rather than at the symbol level.

5.4. Structure of $B$

In equation (1): this leads to a similar data model augmented data matrix by taking a shift of the data ($B$ obtained from a left inverse of $M$) which has a factorization $\Lambda$, where $\Lambda$ is a diagonal matrix. Using FFTs, application of $\Lambda$ is a known (codes) parametric matrix functions.

Training symbols. The midamble of the data block consists of 296 = 33 · 8 + 32 known chips. This is geared towards equation (3), where it means that there is a known block of size 264 × 264, $T$ say, in the center of the data matrix $S$. This can be inverted and applied to the left hand side (a submatrix of $\tilde{X}$) to recover the chip channel coefficients $H$. In fact, the training block is fixed and can be inverted off-line. Moreover, in the proposal [1], the training sequences are constructed such that the matrix to be inverted has a cyclic Toeplitz structure. This means that $T$ has a factorization into $T = F^{-1} \Lambda F$, where $F$ is the matrix representation of a Fourier transformation, and $\Lambda$ is a diagonal matrix. Using FFTs, application of $T$ or its inverse to a vector can be done more efficiently, in order $n \log(n)$ rather than $n^2$, where $n = 264$. It has to be applied to $M$ vectors. \footnote{If the channel length would be defined as 32 chips, then we would obtain $n = 256$ and the FFT implementation becomes nicer.}
can compute $B_i = C_i h_i$ and construct the block Hankel matrix $B$ in (2). The required space-time equalizers which separate the signals and equalize them are left inverses of $B$: e.g., the zero-forcing equalizer is the Moore-Penrose pseudo-inverse.

Training symbols, alternative. Another way to estimate the channels from training is to use the equation $X = B D$. This is possible if the training would consist of known symbols modulating the known codes (this would require small changes in the definition of the training block). Since $B$ has $3 \cdot 8 = 24$ rows, we need 24 consecutive columns of $D$ to be known, which implies $24 + 3 - 1 = 26$ known symbols (this is more than the $296/16 = 18.5$ equivalent symbols required in the previous scheme). Another difference is that, now, to estimate the channels, we need to invert and apply only a $24 \times 24$ matrix. It is still possible to design the training symbols such that this matrix has a cyclic Toeplitz structure, although the dimensions are almost too small to effectively use this. This time, the inverted matrix has to be applied to $16M$ vectors to estimate the channels. This makes the overall complexity similar to the previous scheme. However, the computational latency can be significantly less.

An improved accuracy of $B$ can be obtained if we further constrain the solution to satisfy the coding structure $\text{vec}(B_i) = C_i h_i$, where $C_i$ is known. This constraint is not hard to apply (it involves a projection onto the column span of $C_i$, separately for each user).

It should also be possible to directly estimate a suitable left inverse $B_i^T$ of $B$, i.e., to directly estimate the space-time equalizers without first estimating the channels.

Blind equalization. The block Toeplitz structure of $D$ enables blind equalization. Several schemes are available; they are usually limited in either the assumption that the channel really has length $L = 3$ and not 2 or 1, or that the rows of $D$ are sufficiently orthogonal to each other. Blind equalization does not provide separation of the 8 users yet: we end up with an instantaneous MIMO model $X = AD$. For separation, the finite alphabet structure of the symbols can be used, but since $d = 8$ users is rather large, this will be computationally complex and not reliable without good initialization. It is more attractive to use the code structure provided by $\text{vec}(B_i) = C_i h_i$ for separation. Such schemes are possible but have not been investigated yet.

Semiblind. A semiblind approach would use both the training block and force the Toeplitz structure of $D$, and perhaps the finite alphabet structure as well. Such schemes are promising but at this point the techniques are immature.

Constant modulus algorithms. Some constant modulus structure is present: $g_i(t) * c_i(t)$ has a constant modulus over its support. If a BPSK alphabet would be used, then the modulated signal has constant modulus. In the present case, a 4PSK alphabet is used, and the modulated signal can be viewed as a sum of two constant-modulus signals, which by itself does not have constant modulus, and standard techniques do not apply.

It seems more natural to regard $g_i(t) * c_i(t)$ as a known pulse shape function. Hence, constant-modulus algorithms are not likely to play a role.

CDMA code structure. Techniques such as [22–24] use the model $\tilde{X} = HS$ in (3), and force the template structure that strings of 16 bits in the rows of $S$ have the form $d_{ik} [c_{i0} \cdots c_{i15}]$.
References


[21] L. De Lathauwer, B. De Moor, and J. Vandewalle, “Independent component analysis based on higher-order statistics only,” in Proc. IEEE SP


