STOCHASTIC ANALYSIS OF AN ITERATIVE SEMI-BLIND ADAPTIVE BEAMFORMING ALGORITHM

Myung-Hoon Yeon and John J. Shynk

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560 USA
phone: + (1) 805-893-3977, fax: + (1) 805-893-3262, email: shynk@ece.ucsb.edu

ABSTRACT

We present an iterative semi-blind adaptive beamformer for time-division multiple-access (TDMA) cellular systems and analyze its performance using a stochastic model. The array inputs are iteratively processed using re-encoded data with blind adaptation and several stages that include beamforming, demodulation, equalization, and decoding. In each stage, the initial beamformer weights computed by the TDMA training data are refined by a semi-blind technique based on the constant modulus algorithm (CMA). The receiver algorithms are verified using simulated and real TDMA data collected by an eight-element antenna array. The stochastic analysis of the iterative semi-blind beamforming algorithm is based on a fourth-order moment, and demonstrates the improved performance of the receiver.

1. INTRODUCTION

In a cellular radio system, cochannel interference (CCI) influences a receiver's ability to detect a signal of interest (SOI). We consider TDMA signals with an embedded training sequence and coded data, and examine the performance of the receiver for the Global System for Mobile communications (GSM). In recent years, there has been work on techniques to mitigate the effects of CCI in such systems (e.g., [1]), including receivers based on an adaptive antenna array (beamformer) that spatially isolates the SOI via directional nulling (e.g., [2] and [3]). However, in these previous works there is a common drawback: the adaptive beamformer weights are computed only using the known training sequence. Since the coded data are located outside the range of the training sequence, as shown in Figure 1, the CCI affecting the coded data can be quite different from that estimated by the known training sequence (e.g., in fast-fading channels). To solve this problem, we previously introduced an iterative beamforming algorithm using re-encoded data [4] where the received signals are recursively processed across multiple stages. These stages include a series of steps: burst detection, beamforming, demodulation, channel equalization using the soft-output Viterbi algorithm (SOVA) [5], and decoding.

In this paper, we extend the iterative beamformer to incorporate training-based as well as blind adaptation strategies using CMA [6] (i.e., semi-blind adaptation). At each stage, the initial beamformer weights are computed using training data (either the known training sequence or re-encoded data) and are refined using the semi-blind technique as summarized in Figure 2. The motivation for this approach is as follows. In the first stage, the array data have not yet been decoded, so re-encoded data cannot be used to generate the beamformer weights to suppress the CCI affecting the coded data. Instead, a blind adaptation technique is combined with the known training sequence to implement a semi-blind algorithm. In the second stage (and any successive stages) as shown in Figure 2, a CMA-based approach is employed to compensate for the weight distortion caused by errors in the re-encoded data.

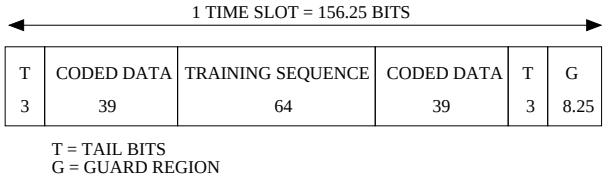

Figure 1: SCH slot structure.

The stochastic analysis of the semi-blind approach presented here uses Wiener filter theory [7] and thus assumes wide-sense stationary signals. A model using fourth-order moments [8] is employed to derive closed-form expressions for the Wiener weights. Although we focus on GSM signals and Gaussian minimum-shift keying (GMSK) because real data is available, the technique can be applied to other TDMA formats. Computer simulations demonstrate that the semi-blind approach enhances the performance of the iterative multistage beamformer.

2. SIGNAL MODEL AND APPROXIMATION

Figure 1 shows the slot (burst) structure for the synchronization channel (SCH) in GSM [9]. Although we examine only the SCH channel which has a relatively high and stable power, and thus is more easily collected, we intend to extend this work to the traffic channels. Also, the SCH coding structure is simpler than that of the traffic channels, so the corresponding additional complexity (e.g., due to interleaving) does not have to be considered. A known 64-bit training sequence is located near the middle of the burst and 39 bits of encoded data are located on each end. Using Laurent's approximation [10], the GMSK baseband signal $s(t)$ can be approximated by the main pulse $C_0(t)$ according to

$$s(t) \approx \sum_{k=0}^{\infty} s(k) C_0(t - kT_b) \tag{1}$$

where $s(k) = j^{\sum_{n=0}^{k} d(n)} = \prod_{n=0}^{k} j^{d(n)} = j^{k+1}(-a(k))$ and $a(-1) = -1$ from the GSM specification. The differentially encoded output is represented by $d(k) = a(k-1)a(k)$, $a(k)$ is a nonreturn-to-zero (NRZ) symbol with alphabet $\{-1, 1\}$, and $T_b = 3.69\ \mu$s is the bit duration. In order to avoid having

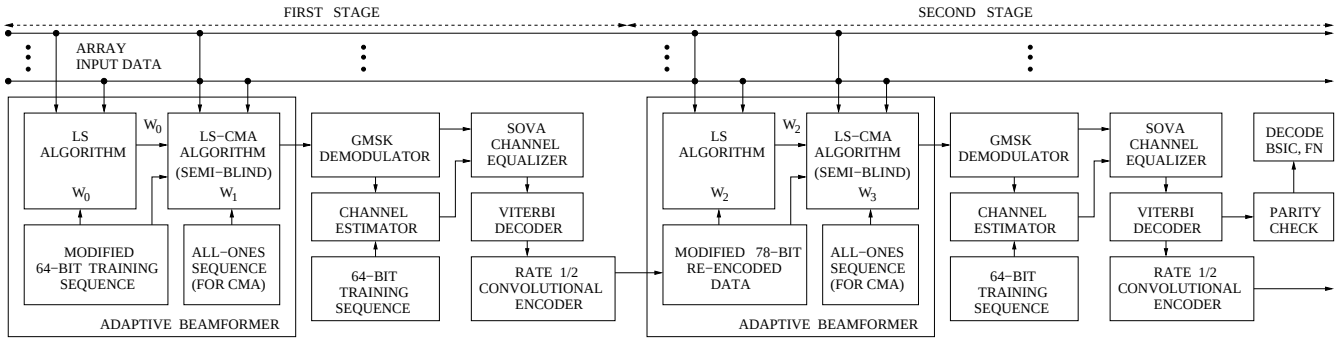

Figure 2: Iterative semi-blind receiver.

to keep track of the negative sign, $s(k)$ is represented instead by $s(k) = j^{k+1}\hat{a}(k)$ where we have defined $\hat{a}(k) \stackrel{\Delta}{=} -a(k)$. The linear nature of the approximated signal leads to a simplified receiver design.

3. ITERATIVE SEMI-BLIND BEAMFORMER

In the first stage shown in Figure 2, the starting positions of every SCH burst are found and the initial beamformer weights $\mathbf{w}_0$ are obtained by minimizing the following least-squares (LS) cost function [4]:

$$J_0(\mathbf{w}_0) = \sum_{j=0}^{63} |\mathbf{w}_0^H \mathbf{x}(k+j) - t_b(j)|^2 \qquad (2)$$

where $\mathbf{x}(k+j) \stackrel{\Delta}{=} [x_1(k+j), \ldots, x_N(k+j)]^T$ is the received signal vector, N is the number of antenna elements, and the superscript ^{H} denotes complex conjugate transpose. The original SCH training sequence $\mathbf{t} = [1, 0, \ldots, 1]^T$ is modified to $\mathbf{t}_b = [j^{43}, -j^{44}, \ldots, j^{106}]^T$, where $C_0(t)$ in (1) has been suppressed and the subscript b denotes that this training sequence is used only for the beamformer (and not the follow-on equalizer). For blind updating of the weights using the coded data, we employ the following modified CMA cost function:

$$\begin{aligned} J(\mathbf{w}_1) &= \sum_k (\mathbf{w}_1^H \mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}_1 - 1)^2 \\ &\approx \sum_k |\mathbf{w}_1^H \mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}_0 - 1|^2 \\ &= \sum_k |\mathbf{w}_1^H \mathbf{z}_0(k) - 1|^2, \end{aligned} \qquad (3)$$

which can be solved using LS techniques because it is quadratic in $\mathbf{w}_1$ (and is similar to (2)). Note that we have used the approximation $\mathbf{x}^H(k)\mathbf{w}_1 \approx \mathbf{x}^H(k)\mathbf{w}_0$, and defined $\mathbf{z}_0(k) \stackrel{\Delta}{=} \mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}_0$, which is an extension of the approach in [11]. Based on this result, the refined beamformer weights $\mathbf{w}_1$ for the GSM receiver are computed by minimizing

$$\begin{aligned} J_1(\mathbf{w}_1) &= \sum_{k=3}^{41} |\mathbf{w}_1^H \mathbf{z}_0(k) - 1|^2 \\ &\quad + \sum_{k=42}^{105} |\mathbf{w}_1^H \mathbf{x}(k) - t_b(k-42)|^2 \\ &\quad + \sum_{k=106}^{144} |\mathbf{w}_1^H \mathbf{z}_0(k) - 1|^2. \end{aligned} \qquad (4)$$

After the refined beamformer weights are computed, the data are processed to generate the beamformer output $y(k) = \mathbf{w}_1^H \mathbf{x}(k)$ for $k = 0, \ldots, 147$, which then passes to the GMSK demodulator and the channel equalizer as shown in Figure 2. We employ a SOVA equalizer (based on a five-tap channel estimator) whose output enters the Viterbi decoder. SCH bursts passing the 10-bit parity check are identified using the 6-bit base station identification code (BSIC) and the 19-bit frame identification number (FN).

In the second stage, the Viterbi decoder of the first stage produces 39 bits of data. The 78 bits of re-encoded data $\mathbf{e}_r$ are generated by the convolutional encoder, where the subscript r denotes re-encoded. The data are modified for the beamformer using Laurent's approximation, yielding the training data $\mathbf{t}_r \stackrel{\Delta}{=} [e_r(0)j^4, \ldots, e_r(38)j^{42}, e_r(39)j^{107}, \ldots, e_r(77)j^{145}]^T$, where in the exponent of j^{k+1}, k represents the bit position relative to the coded data in an SCH burst. The initial beamformer weights $\mathbf{w}_2$ in the second stage are computed by minimizing

$$\begin{aligned} J_2(\mathbf{w}_2) &= \sum_{j=0}^{38} |\mathbf{w}_2{}^H \mathbf{x}(3+j) - t_r(j)|^2 \\ &\quad + \sum_{j=39}^{77} |\mathbf{w}_2{}^H \mathbf{x}(67+j) - t_r(j)|^2. \end{aligned} \qquad (5)$$

The refined beamformer weights are computed using the following update:

$$\mathbf{w}_3(k) = \mathbf{w}_2 + \mathbf{k}(k)e^*(k) \qquad (6)$$

where $e(k) = 1 - \mathbf{w}_2^H \mathbf{z}_s(k)$ is the estimation error, and the superscript * denotes complex conjugation. The input data are given by $\mathbf{z}_s(k) = \mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}_2$ where the subscript s denotes the second stage. The gain vector is

$$\mathbf{k}(k) = \mathbf{P}(k)\mathbf{z}_s(k)/[\alpha + \mathbf{z}_s^H(k)\mathbf{P}(k)\mathbf{z}_s(k)] \qquad (7)$$

where $\alpha = 1$, $\mathbf{P}(k) \stackrel{\Delta}{=} [\mathbf{X}_s(k)\mathbf{X}_s^H(k)]^{-1}$, $\mathbf{X}_s(k)$ is the $N \times N_{RT}$ data matrix used in the LS solution of (5), and $N_{RT} = 78$ is the size of the re-encoded data. The array data are processed to generate the beamformer output $y_s(k) = \mathbf{w}_3^H(k)\mathbf{x}(k)$ for $k = 0, \ldots, 147$. The remaining steps (demodulation, channel equalization, and decoding) are similar to those used in the first stage. The decoder output of the second stage is delivered to the third stage as implied in Figure 2.

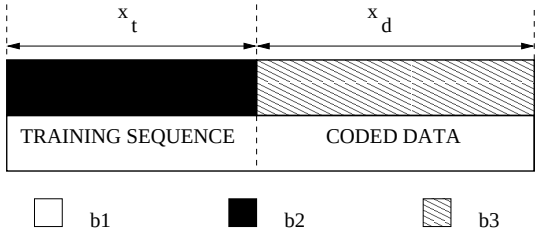

Figure 3: Scenario for the stochastic analysis.

4. STOCHASTIC ANALYSIS

Next, we analyze the performance of the semi-blind beamforming algorithm using a stochastic model. Consider a simple scenario where one desired BPSK signal b_1 and two interference BPSK signals b_2 and b_3 impinge on a two-element array as summarized in Figure 3. The SOI b_1 consists of a known training sequence and coded data, while b_2 is an interferer within the period of the training sequence of b_1, and b_3 is an interferer within the period of the coded data of b_1. At the two antenna elements, the received signals can be represented by

$$\begin{aligned} x_{t1} &= b_1 + b_2 + n_1 \\ x_{t2} &= b_1 e^{j\theta_1} + b_2 e^{j\theta_2} + n_2 \\ x_{d1} &= b_1 + b_3 + n_3 \\ x_{d2} &= b_1 e^{j\theta_1} + b_3 e^{j\theta_3} + n_4 \end{aligned} \tag{8}$$

where the subscripts t and d denote during the training sequence and coded data, respectively. The $\{b_i\}$ are independent and identically distributed with alphabet $\{-1,1\}$, and have zero mean and unit variance, i.e., $E[b_i b_j] = \delta_{ij}$. The additive noise n_i is white and Gaussian with zero mean and variance σ_n^2, i.e., $E[n_i n_j^*] = \sigma_n^2 \delta_{ij}$. The deterministic phase θ_i represents the carrier phase shift $-2\pi\Delta\cos\phi_i/\lambda$ across the two antenna elements, where ϕ_i is the angle of incidence, Δ is the interlement spacing, and λ is the wavelength. The received signals in the region of the training sequence of b_1 are represented by $\mathbf{x}_t = [x_{t1}, x_{t2}]^T$, and those in the region of the coded data of b_1 are given by $\mathbf{x}_d = [x_{d1}, x_{d2}]^T$. Thus, the received signals x_{ti} and x_{di} have the following statistics:

$$\begin{aligned} E[x_{ti}] &= E\{x_{di}\} = 0 \\ E[x_{ti}^2] &\triangleq \sigma_{t11} = \sigma_{t22} = 2 + \sigma_n^2 \\ E[x_{di}^2] &\triangleq \sigma_{d11} = \sigma_{d22} = 2 + \sigma_n^2 \\ E[x_{t1} x_{t2}^*] &\triangleq \sigma_{t12} = e^{-j\theta_1} + e^{-j\theta_2} \\ E[x_{t2} x_{t1}^*] &\triangleq \sigma_{t21} = e^{j\theta_1} + e^{j\theta_2} \\ E[x_{d1} x_{d2}^*] &\triangleq \sigma_{d12} = e^{-j\theta_1} + e^{-j\theta_3} \\ E[x_{d2} x_{d1}^*] &\triangleq \sigma_{d21} = e^{j\theta_1} + e^{j\theta_3}. \end{aligned} \tag{9}$$

In the analysis, we employ a Gaussian assumption and the following well-known fourth-order moment [8]:

$$E[x_i x_j^* x_k x_n^*] = \sigma_{ij}\sigma_{kn} + \sigma_{in}\sigma_{kj} \tag{10}$$

where $\sigma_{ij} \triangleq E[x_i x_j^*]$. Although the Gaussian model is not exact, it yields accurate results – especially for moderate signal-to-noise ratios (SNRs), and becomes more accurate with an increasing number of interferers. (Note that the high-order statistics in [12] could also be used to analyze this stochastic model.)

In the first stage of the receiver, the initial beamformer weights $\mathbf{w}_0$ are computed from the Wiener-Hopf equation [7], instead of the LS method shown in Figure 2. Define the correlation matrix

$$\begin{aligned} \mathbf{R}_t &\triangleq E[\mathbf{x}_t \mathbf{x}_t^H] = E\begin{bmatrix} x_{t1}x_{t1}^* & x_{t1}x_{t2}^* \\ x_{t2}x_{t1}^* & x_{t2}x_{t2}^* \end{bmatrix} \\ &= \begin{bmatrix} 2+\sigma_n^2 & e^{-j\theta_1} + e^{-j\theta_2} \\ e^{j\theta_1} + e^{j\theta_2} & 2+\sigma_n^2 \end{bmatrix} \end{aligned} \tag{11}$$

where $\mathbf{x}_t$ is shown in Figure 3. Also define the following cross-correlation vector:

$$\mathbf{p}_t \triangleq E[\mathbf{x}_t d_t^*] = E\begin{bmatrix} x_{t1}b_1^* \\ x_{t2}b_1^* \end{bmatrix} = \begin{bmatrix} 1 \\ e^{j\theta_1} \end{bmatrix}. \tag{12}$$

For notational convenience, let $\mathbf{w}_t \triangleq [w_{t1}, w_{t2}]^T$ be the initial beamformer weights in the first stage instead of $\mathbf{w}_o$; the Wiener weights are given by

$$\mathbf{w}_t = \mathbf{R}_t^{-1}\mathbf{p}_t. \tag{13}$$

The refined beamformer weights $\mathbf{w}_1$ are computed using $\mathbf{x}_d$ and $\mathbf{w}_t$. Define the vector

$$\mathbf{z} \triangleq \mathbf{x}_d \mathbf{x}_d^H \mathbf{w}_t = \begin{bmatrix} w_{t1}x_{d1}x_{d1}^* + w_{t2}x_{d1}x_{d2}^* \\ w_{t1}x_{d2}x_{d1}^* + w_{t2}x_{d2}x_{d2}^* \end{bmatrix}, \tag{14}$$

and partition the correlation matrix as follows:

$$\mathbf{R}_{cd} \triangleq E[\mathbf{z}\mathbf{z}^H] = E\begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{15}$$

where the subscript cd denotes coded data and

$$\begin{aligned} E[A] &= w_{t1}E[x_{d1}x_{d1}^*x_{d1}x_{d1}^*]w_{t1}^* + w_{t2}E[x_{d1}x_{d2}^*x_{d1}x_{d1}^*]w_{t1}^* \\ &+ w_{t1}E[x_{d1}x_{d1}^*x_{d2}x_{d1}^*]w_{t2}^* + w_{t2}E[x_{d1}x_{d2}^*x_{d2}x_{d1}^*]w_{t2}^* \\ E[B] &= w_{t1}E[x_{d1}x_{d1}^*x_{d1}x_{d2}^*]w_{t1}^* + w_{t2}E[x_{d1}x_{d2}^*x_{d1}x_{d2}^*]w_{t1}^* \\ &+ w_{t1}E[x_{d1}x_{d1}^*x_{d2}x_{d2}^*]w_{t2}^* + w_{t2}E[x_{d1}x_{d2}^*x_{d2}x_{d2}^*]w_{t2}^* \\ E[C] &= w_{t1}E[x_{d2}x_{d1}^*x_{d1}x_{d1}^*]w_{t1}^* + w_{t2}E[x_{d2}x_{d2}^*x_{d1}x_{d1}^*]w_{t1}^* \\ &+ w_{t1}E[x_{d2}x_{d1}^*x_{d2}x_{d1}^*]w_{t2}^* + w_{t2}E[x_{d2}x_{d2}^*x_{d2}x_{d1}^*]w_{t2}^* \\ E[D] &= w_{t1}E[x_{d2}x_{d1}^*x_{d1}x_{d2}^*]w_{t1}^* + w_{t2}E[x_{d2}x_{d2}^*x_{d1}x_{d2}^*]w_{t1}^* \\ &+ w_{t1}E[x_{d2}x_{d1}^*x_{d2}x_{d2}^*]w_{t2}^* + w_{t2}E[x_{d2}x_{d2}^*x_{d2}x_{d2}^*]w_{t2}^*. \end{aligned} \tag{16}$$

Using (10), we can rewrite these terms as

$$\begin{aligned} E[A] &= w_{t1}[2\sigma_{d11}^2]w_{t1}^* + w_{t2}[2\sigma_{d11}\sigma_{d12}]w_{t1}^* \\ &+ w_{t1}[2\sigma_{d11}\sigma_{d21}]w_{t2}^* + w_{t2}[\sigma_{d11}\sigma_{d22} + \sigma_{d12}\sigma_{d21}]w_{t2}^* \\ E[B] &= w_{t1}[2\sigma_{d11}\sigma_{d12}]w_{t1}^* + w_{t2}[2\sigma_{d12}^2]w_{t1}^* \\ &+ w_{t1}[\sigma_{d11}\sigma_{d22} + \sigma_{d12}\sigma_{d21}]w_{t2}^* + w_{t2}[2\sigma_{d12}\sigma_{d22}]w_{t2}^* \\ E[C] &= w_{t1}[2\sigma_{d11}\sigma_{d21}]w_{t1}^* + w_{t2}[\sigma_{d22}\sigma_{d11} + \sigma_{d12}\sigma_{d21}]w_{t1}^* \\ &+ w_{t1}[2\sigma_{d21}^2]w_{t2}^* + w_{t2}[2\sigma_{d22}\sigma_{d21}]w_{t2}^* \\ E[D] &= w_{t1}[\sigma_{d12}\sigma_{d21} + \sigma_{d11}\sigma_{d22}]w_{t1}^* + w_{t2}[2\sigma_{d12}\sigma_{d22}]w_{t1}^* \\ &+ w_{t1}[2\sigma_{d21}\sigma_{d22}]w_{t2}^* + w_{t2}[2\sigma_{d22}^2]w_{t2}^*. \end{aligned} \tag{17}$$

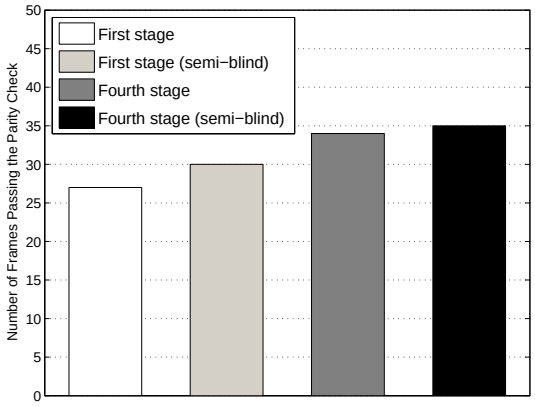

Figure 4: Number of frames passing the parity check (real GSM data).

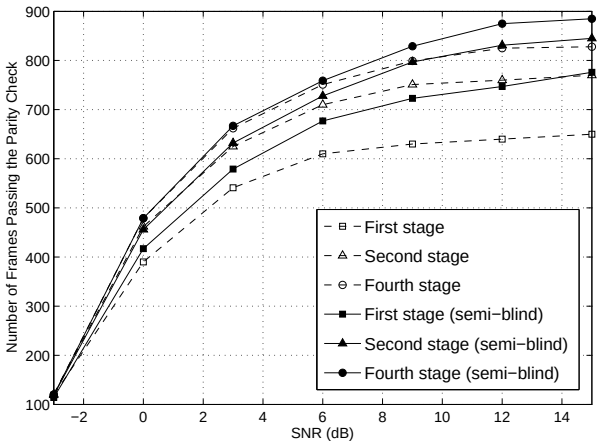

Figure 5: Number of frames passing the parity check (simulated GSM data).

Similarly, the cross-correlation vector is rewritten as

$$\mathbf{p}_{cd} \stackrel{\Delta}{=} E[\mathbf{z}d_d^*] = E[\mathbf{z}] = E[\mathbf{x}_d\mathbf{x}_d^H\mathbf{w}_t] = \begin{bmatrix} w_{t1}\sigma_{d11} + w_{t2}\sigma_{d12} \\ w_{t1}\sigma_{d21} + w_{t2}\sigma_{d22} \end{bmatrix} \quad (18)$$

where $d_d = 1$ is the CMA modulus (because of the BPSK data). For the semi-blind algorithm, the input correlation matrix can be expressed as

$$\mathbf{R}_{sb} = E\left[\begin{bmatrix}\mathbf{x}_t & \mathbf{z}\end{bmatrix}\begin{bmatrix}\mathbf{x}_t^H \\ \mathbf{z}^H\end{bmatrix}\right] = \mathbf{R}_t + \mathbf{R}_{cd} \quad (19)$$

where the subscript *sb* denotes semi-blind. The corresponding cross-correlation vector is

$$\mathbf{p}_{sb} = E\left[\begin{bmatrix}\mathbf{x}_t & \mathbf{z}\end{bmatrix}\begin{bmatrix}d_t^* \\ d_d^*\end{bmatrix}\right] = \mathbf{p}_t + \mathbf{p}_{cd}. \quad (20)$$

Finally, in the first stage, the refined beamformer weights are

$$\mathbf{w}_1 = \mathbf{R}_{sb}^{-1}\mathbf{p}_{sb}. \quad (21)$$

Thus, using the statistics for this example, we have derived closed-form expressions for the Wiener weights $\mathbf{w}_0$ (relabeled as $\mathbf{w}_t$) and $\mathbf{w}_1$. In the second stage, $\mathbf{w}_2$ and $\mathbf{w}_3$ can be derived using a similar procedure. (Note that the initial beamformer weights $\mathbf{w}_2$ would be calculated using $\mathbf{x}_d$ instead of $\mathbf{x}_t$.)

5. COMPUTER SIMULATION RESULTS

5.1 Real GSM Data

We verify the performance of the iterative beamformer with and without the semi-blind algorithm by processing real GSM data collected for about 1.2 s using an eight-element antenna array, which were sampled at about 2.5 Msps, filtered, and then re-sampled at the GSM bit rate. Figure 4 compares the performance of the receiver using the iterative beamformer with that using the semi-blind approach; the performance measure is the number of frames passing the parity check. Observe that an increased number of stages improves the performance of the receivers, and that the semi-blind algorithm provides further improvement.

5.2 Simulated GSM Data

We also evaluate the performance of the receiver using simulated GSM data. One thousand independent SCH frames were generated for a range of SNRs and a signal-to-interference ratio (SIR) of 0 dB. The desired SCH signal (i.e., SOI) and two cochannel SCH interferers impinged on the linear array with random angles chosen independently in the range $[0^\circ, 180^\circ]$ for each trial. Note that multipath channels were not considered here. Four successive stages were employed using the receiver architecture in Figure 2. The results in Figure 5 demonstrate that the adaptive beamformer using iteratively re-encoded data improves the performance of the receiver – especially in the first two stages. Furthermore, the performance of the beamformer using iterative training and the semi-blind technique is clearly better.

5.3 Stochastic Model

Finally, we investigate the performance of the semi-blind algorithm using the beamformer weights derived from the stochastic model. Specifically, the re-encoded data error (i.e., training data error) is assumed to occur during the period of $\mathbf{x}_d$, such that the weights $\mathbf{w}_2$ are distorted according to the SNR level. The simulation steps were as follows. (i) Calculate the beamformer weights $\mathbf{w}_0$ and $\mathbf{w}_1$ using (13) and (21), respectively, and using similar expressions for $\mathbf{w}_2$ and $\mathbf{w}_3$ in the second stage. We assume that $\mathbf{w}_2 = \mathbf{w}_{\text{opt}} + \beta$, where $\beta \propto 1/\text{SNR}$, to model the re-encoded data errors, and $\mathbf{w}_{\text{opt}}$ is the Wiener weight vector obtained without any re-encoding errors. The angles of arrival for b_1, b_2, and b_3 were 60°, 90°, and 150°, respectively. (ii) 10^6 independent samples of $\mathbf{x}_d$ were generated and processed by the beamformers using the Wiener weights. (iii) The bit error rate (BER) was measured at the output of the various beamformers. As shown in Figure 6, the semi-blind algorithm improves the performance of the beamformer, even though the CCI affecting the coded data of b_1 is different from that of the known training sequence (i.e., the first stage) and the initial weights are distorted by re-encoding errors (i.e., the second stage). Figure 7 shows the corresponding beampatterns of the weights, where we see that the beampattern of $\mathbf{w}_1$ has moved to that of $\mathbf{w}_{\text{opt}}$ and away from that of $\mathbf{w}_0$. In addition, the nulling angle of

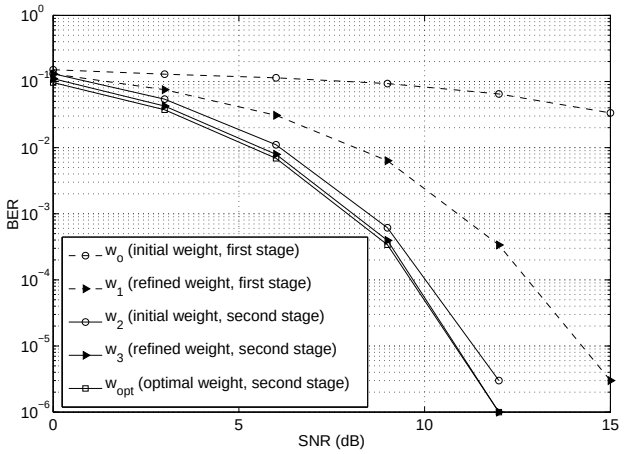

Figure 6: BER results using the stochastic model.

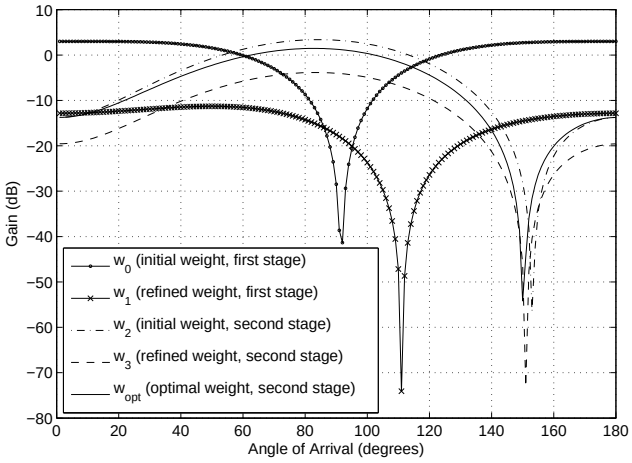

Figure 7: Beampatterns using the stochastic model.

$\mathbf{w}_3$ is closer to that of $\mathbf{w}_{\text{opt}}$ than is $\mathbf{w}_2$.

6. CONCLUSION

We have analyzed an iterative semi-blind beamforming algorithm for TDMA signals using a stochastic (Wiener) model. The array input data are processed in several stages that include beamforming, demodulation, equalization, and decoding. In each stage, the beamformer weights are calculated using a two-step procedure: the initial beamformer weights are computed using training data, and these are refined by a semi-blind technique using CMA. In the first stage, a modified CMA cost function allows us to derive the semi-blind algorithm using a least-squares formulation. In the second stage (and any successive stages), the semi-blind technique is used to compensate for weight distortions caused by re-encoding errors. The performance of the beamforming algorithm was evaluated by processing simulated and real GSM data, and using the results of the stochastic model. The simulations show that the iterative beamformer using re-encoded data improves the performance of the multistage receiver with only a few stages, and the semi-blind beamforming further enhances the results. Although we focused on GSM signals (for which real data is readily available), these algorithms can be applied to other cellular radio systems such as GPRS/EDGE and UMTS.

REFERENCES

[1] A. Hafeez, K. J. Molnar, H. Arslan, G. E. Bottomley, and R. Ramesh, "Adaptive joint detection of cochannel signals for TDMA handsets," *IEEE Trans. Communications*, vol. 52, pp. 1722–1732, Oct. 2004.

[2] J. Leary and R. P. Gooch, "Adaptive beamforming for TDMA signals," in *Proc. Twenty-Ninth Asilomar Conf. Signals, Systems, Computers*, (Pacific Grove, CA), pp. 1378–1382, Nov. 1995.

[3] E. Lindskog, "Making SMI-beamforming insensitive to the sampling timing for GSM signals," in *Proc. IEEE Int. Symp. Personal, Indoor, Mobile Radio Communications*, (Toronto, Ontario, Canada), pp. 664–668, Sept. 1995.

[4] M. H. Yeon, J. J. Shynk, and R. P. Gooch, "A comparative performance investigation of adaptive beamformers with iterative weight refinement for GSM signal detection," in *Proc. IEEE Workshop Sensor Array and Multichannel Processing*, (Waltham, MA), pp. 571–575, July 2006.

[5] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Global Telecommunications Conf.*, (Dallas, TX), pp. 1680–1686, Nov. 1989.

[6] J. R. Treichler and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-31, pp. 459–472, Apr. 1983.

[7] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice-Hall, third ed., 1996.

[8] I. S. Reed, "On a moment theorem for complex Gaussian processes," *IRE Trans. Information Theory*, vol. IT-8, pp. 194–195, Apr. 1962.

[9] S. M. Redl, M. K. Weber, and M. W. Oliphant, *An Introduction to GSM*. Norwood, MA: Artech House, 1995.

[10] P. A. Laurent, "Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP)," *IEEE Trans. Communications*, vol. COM-34, pp. 150–160, Feb. 1986.

[11] Y. Chen, T. Le-Ngoc, B. Champagne, and C. Xu, "Recursive least squares constant modulus algorithm for blind adaptive array," *IEEE Trans. Signal Processing*, vol. 52, pp. 1452–1456, May 2004.

[12] A. Teschioni, C. Sacchi, and C. Regazzoni, "Non-Gaussian characterization of DS/CDMA noise in few-user systems with complex signature sequences," *IEEE Trans. Signal Processing*, vol. 47, pp. 234–237, Jan. 1999.