2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

On-Line Gaussian Mixture Density Estimator for Adaptive Minimum Bit-Error-Rate Beamforming Receivers

Sheng Chen, Xia Hong and Chris J. Harris

Abstract— **We develop an on-line Gaussian mixture density estimator (OGMDE) in the complex-valued domain to facilitate adaptive minimum bit-error-rate (MBER) beamforming receiver for multiple antenna based space-division multiple-access systems. Specifically, the novel OGMDE is proposed to adaptively model the probability density function of the beamformer's output by tracking the incoming data sample by sample. With the aid of the proposed OGMDE, our adaptive beamformer is capable of updating the beamformer's weights sample by sample to directly minimize the achievable bit error rate (BER). We show that this OGMDE based MBER beamformer outperforms the existing on-line MBER beamformer, known as the least BER beamformer, in terms of both the convergence speed and the achievable BER.**

I. Introduction

The ever-increasing demand for mobile communication capacity has motivated the development of multi-antenna based space-division multiple access (SDMA) systems to further enhance the achievable system capacity [1]–[5]. Adaptive beamforming is capable of separating users in the spatial domain and provides a practical means of supporting multiusers in SDMA systems. Traditionally, the beamforming process is based on the minimum mean square error (MMSE) criterion, and adaptive MMSE beamforming can be implemented using the well-known least mean square (LMS) algorithm which updates the beamformer's weights sample by sample. However, for a communication system, it is the bit error rate (BER) that really matters, and adaptive minimum BER (MBER) beamforming design was proposed in [4], [5], which significantly outperforms the adaptive MMSE beamformer, in terms of achievable system capacity and BER. Adaptively estimating the probability density function (PDF) of the beamformer's output with the new coming data sample by sample is the key to implementing an on-line MBER beamformer. The least BER (LBER) algorithm [4], [5] adopts a stochastic one-sample or single Gaussian kernel estimate for the PDF of the beamformer's output to realize sample-by-sample adaptation of the beamformer's weights, in a manner similar to the LMS algorithm which uses a one sample estimate for the mean square error (MSE) [6].

Despite of its simplicity and its superior performance over the adaptive MMSE beamformer, the LBER beamformer [4], [5] is sensitive to the noise in the received signal sample, owing to its stochastic nature of one-sample PDF estimate. In the literature, there exist large amount of works [7]–[13] using the Gaussian mixture model to estimate PDF in the real-valued (RV) domain. These PDF estimators based on a mixture of Gaussians are batch learning algorithms and they are unsuitable for on-line applications. In this paper, we propose a new on-line Gaussian mixture density estimator (OGMDE) in the complex-valued (CV) domain to update the PDF estimate of the beamformer's output sample by sample. Specifically, a new Gaussian kernel is formed for every new data and it is then merged with the "nearest" existing Gaussian kernel in the OGMDE. With the aid of this OGMDE for on-line estimation of the beamformer output's PDF, the beamformer's weights can be adapted sample by sample to minimise the beamformer receiver's BER in a manner similar to the LBER beamformer [4], [5]. This new adaptive MBER beamformer is referred to as the OGMDE aided adaptive MBER (OGMDE-AMBER) beamformer. Because this OGMDE-AMBER beamformer relies on a more accurate on-line PDF estimate, unlike the one-sample PDF estimator of the LBER beamformer, it outperforms the latter in terms of both the convergence speed and the achievable BER. Simulation results obtained demonstrate that the OGMDE-AMBER beamformer significantly improve the receiver's performance, compared to the existing LBER beamformer.

The remaining of this paper is organised as follows. Section II first introduces the multi-antenna based SDMA system and then reviews the existing adaptive beamforming receivers, including the LMS based MMSE beamformer and the LBER based MBER beamformer. Section III derives the new OGMDE, followed by the proposed OGMDE-AMBER beamforming algorithm. Simulation results are presented in Section IV to compare the performance of the OGMDE-AMBER beamformer with the existing LBER beamformer, while our conclusions are offered in Section V.

II. Existing Adaptive Beamforming Receivers

A. System Model

The system supports M single-antenna users, and each user transmits a quadrature phase shift keying (QPSK) signal on the same carrier frequency of $\omega_c = 2\pi f_c$. The baseband CV signal of user m, sampled at the symbol rate, is given by $d_m(k) = A_m b_m(k)$, where $1 \leq m \leq M$, the transmitted QPSK symbol $b_m(k)$ takes the value from the symbol set $\{\pm 1 \pm \mathrm{j}\}$ and A_m is the CV channel coefficient for user m, while k denotes the sample index. The channel is assumed to be narrow-band and does not induce intersymbol interference. In order to separate users in the spatial or angular

S. Chen and C.J. Harris are with Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (E-mails: sqc@ecs.soton.ac.uk, cjh@ecs.soton.ac.uk). S. Chen is also with King Abdulaziz University, Jeddah 21589, Saudi Arabia.

X. Hong is with School of Systems Engineering, University of Reading, Reading RG6 6AY, U.K. (E-mail: x.hong@reading.ac.uk, hao.chen@pgr.reading.ac.uk).

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

domain, the base station receiver is equipped with a linear antenna array consisting of L uniformly spaced elements. The symbol-rate received signal samples at the output of the L-element antenna array can be expressed as

$$\begin{aligned} x_l(k) =& \sum_{m=1}^{M} A_m b_m(k) e^{\mathrm{j}\omega_c t_l(\theta_m)} + n_l(k) \\ =& \bar{x}_l(k) + n_l(k),\ 1 \le l \le L, \end{aligned} \tag{1}$$

where $t_l(\theta_m)$ is the relative time delay at array element l for user m with θ_m being the direction of arrival for user m and $n_l(k)$ is a CV additive white Gaussian noise (AWGN) with $E[|n_l(k)|^2] = 2\sigma_n^2$. Without loss of generality, user 1 is the desired user, and the rest of the users are the interfering users. Since the received signal power for user m is $2|A_m|^2$, the system's desired signal to noise ratio (SNR) is defined by $\mathrm{SNR} = |A_1|^2/\sigma_n^2$, and the desired signal to interference ratio (SIR) with respect to the interferer m is given by $\mathrm{SIR}_m = |A_1|^2/|A_m|^2$ for $2 \le m \le M$. The received signal vector $\boldsymbol{x}(k) = \big[x_1(k)\ x_2(k) \cdots x_L(k)\big]^{\mathrm{T}}$ is given by

$$\boldsymbol{x}(k) = \boldsymbol{P}\boldsymbol{b}(k) + \boldsymbol{n}(k) = \bar{\boldsymbol{x}}(k) + \boldsymbol{n}(k), \tag{2}$$

where the noise vector $\boldsymbol{n}(k) = \big[n_1(k)\ n_2(k) \cdots n_L(k)\big]^{\mathrm{T}}$, the system matrix $\boldsymbol{P} = \big[A_1\boldsymbol{s}_1\ A_2\boldsymbol{s}_2 \cdots A_M\boldsymbol{s}_M\big]$ with the steering vector for user m given by $\boldsymbol{s}_m = \big[e^{\mathrm{j}\omega_c t_1(\theta_m)}\ e^{\mathrm{j}\omega_c t_2(\theta_m)} \ldots e^{\mathrm{j}\omega_c t_L(\theta_m)}\big]^{\mathrm{T}}$, and the transmitted QPSK symbol vector $\boldsymbol{b}(k) = \big[b_1(k)\ b_2(k) \cdots b_M(k)\big]^{\mathrm{T}}$.

A beamforming receiver is employed to recover the desired user 1 data, and the beamformer's output is given by

$$\begin{aligned} y(k) =& \boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}(k) = \boldsymbol{w}^{\mathrm{H}}\bar{\boldsymbol{x}}(k) + \boldsymbol{w}^{\mathrm{H}}\boldsymbol{n}(k) = \bar{y}(k) + e(k) \\ =& c_1 b_1(k) + \sum_{m=2}^{M} c_m b_m(k) + e(k), \end{aligned} \tag{3}$$

where $\boldsymbol{w} = \big[w_1\ w_2 \cdots w_L\big]^{\mathrm{T}}$ is the CV beamformer weight vector, and $e(k)$ is Gaussian distributed having a zero mean and a variance of $E[|e(k)|^2] = 2\sigma_n^2\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}$, while

$$\boldsymbol{w}^{\mathrm{H}}\boldsymbol{P} = \big[c_1\ c_2 \cdots c_M\big]. \tag{4}$$

Provided that c_1 is real and positive, the optimal decision rule for detecting $b_1(k)$ is given by

$$\widehat{b}_1(k) = \mathrm{sgn}\big(y_R(k)\big) + \mathrm{j}\,\mathrm{sgn}\big(y_I(k)\big), \tag{5}$$

where $y_R(k) = \Re[y(k)]$ and $y_I(k) = \Im[y(k)]$ are the real and imaginary parts of $y(k)$. Noting $c_1 = \boldsymbol{w}^{\mathrm{H}} A_1\boldsymbol{s}_1 = \boldsymbol{w}^{\mathrm{H}}\boldsymbol{p}_1$, where $\boldsymbol{p}_1$ is the first column of $\boldsymbol{P}$, we can see that the steering vector $\boldsymbol{s}_1$ and the channel coefficient A_1 of the desired user are required at the receiver. To ensure a real and positive c_1, the following rotation operation can be applied to the weight vector

$$\boldsymbol{w}^{\mathrm{new}} = \frac{c_1^{\mathrm{old}}}{|c_1^{\mathrm{old}}|}\boldsymbol{w}^{\mathrm{old}}. \tag{6}$$

This does not alter the BER, as the BER is invariant to a positive scaling of $\boldsymbol{w}$ [14]. Given a block of training data $\{b_1(k), \boldsymbol{x}(k)\}_{k=1}^{K}$, an estimate of $\boldsymbol{p}_1$ is readily given by

$$\widehat{\boldsymbol{p}}_1 = \frac{1}{K}\sum_{k=1}^{K}\frac{\boldsymbol{x}(k)}{b_1(k)}. \tag{7}$$

Alternatively, the moving average can be used to track $\boldsymbol{p}_1$

$$\widetilde{\boldsymbol{p}}_1(k+1) = (1-\mu_p)\widetilde{\boldsymbol{p}}_1(k) + \mu_p\frac{\boldsymbol{x}(k)}{b_1(k)}, \tag{8}$$

where $0 < \mu_p < 1$ is the step size of the moving average.

B. Adaptive MMSE Beamforming

Classically, the beamformer's weight vector is determined by minimising the MSE criterion of $E[|b_1(k) - y(k)|^2]$, which results in the following MMSE solution

$$\boldsymbol{w}_{\mathrm{mmse}} = \big(\boldsymbol{P}\boldsymbol{P}^{\mathrm{H}} + \sigma_n^2\boldsymbol{I}_L\big)^{-1}\boldsymbol{p}_1, \tag{9}$$

where $\boldsymbol{I}_L$ denotes the $L \times L$ identity matrix. Adaptive MMSE beamforming can be realised on a sample-by-sample basis using the LMS algorithm, yielding the beamforming weight updating equations

$$\widetilde{\boldsymbol{w}}_{\mathrm{lms}}(k+1) = \widetilde{\boldsymbol{w}}_{\mathrm{lms}}(k) + \mu_{\mathrm{lms}}\big(b_1(k) - y(k)\big)^{*}\boldsymbol{x}(k), \tag{10}$$

$$c_1 = \widetilde{\boldsymbol{w}}_{\mathrm{lms}}^{\mathrm{H}}(k+1)\boldsymbol{p}_1, \tag{11}$$

$$\widetilde{\boldsymbol{w}}_{\mathrm{lms}}(k+1) = \frac{c_1}{|c_1|}\widetilde{\boldsymbol{w}}_{\mathrm{lms}}(k+1), \tag{12}$$

where μ_{lms} is the step-size of the LMS algorithm and $(\ )^{*}$ denotes the conjugate operator.

C. Existing Adaptive MBER Beamforming

Denote the $N_b = 4^M$ number of legitimate sequences of $\boldsymbol{b}(k)$ as $\boldsymbol{b}^{(i)}$, $1 \le i \le N_b$. Further denote the first element of $\boldsymbol{b}^{(i)}$, corresponding to the desired user, as $b_1^{(i)} = b_{1_R}^{(i)} + \mathrm{j}b_{1_I}^{(i)}$. The noise-free part of the received signal vector $\bar{\boldsymbol{x}}(k)$ takes the value from the CV vector signal set defined by $\mathbb{X} \triangleq \big\{\bar{\boldsymbol{x}}^{(i)} = \boldsymbol{P}\boldsymbol{b}^{(i)}, 1 \le i \le N_b\big\}$. The set $\mathbb{X}$ can be partitioned into the four subsets, depending on the value of $b_1(k)$ as $\mathbb{X}_{\pm,\pm} \triangleq \big\{\bar{\boldsymbol{x}}^{(i)} \in \mathbb{X} : b_1(k) = \pm 1 \pm \mathrm{j}\big\}$. Similarly the noise-free part of the beamformer's output $\bar{y}(k)$ takes the value from the CV scalar signal set

$$\mathbb{Y} \triangleq \big\{\bar{y}^{(i)} = \boldsymbol{w}^{\mathrm{H}}\bar{\boldsymbol{x}}^{(i)}, 1 \le i \le N_b\big\}, \tag{13}$$

which can be divided into the four conditional subsets

$$\mathbb{Y}_{\pm,\pm} \triangleq \big\{\bar{y}^{(i)} \in \mathbb{Y} : b_1(k) = \pm 1 \pm \mathrm{j}\big\}, \tag{14}$$

each having the size of $N_{sb} = N_b/4$. The four subsets defined in (14) are symmetrically distributed in the CV plane with respect to the real axis and/or imaginary axis, and we can readily obtain the following relationships [15]

$$\mathbb{Y}_{-,+} = \mathbb{Y}_{+,+} - 2c_1, \tag{15}$$

$$\mathbb{Y}_{+,-} = \mathbb{Y}_{+,+} - 2c_1\mathrm{j}, \tag{16}$$

$$\mathbb{Y}_{-,-} = \mathbb{Y}_{+,+} - 2c_1(1+\mathrm{j}). \tag{17}$$

The conditional PDF of $y(k)$ given $b_1(k)=+1+\mathrm{j}$ is

$$p(y|+1+\mathrm{j}) = \frac{1}{N_{sb}2\pi\sigma_n^2\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}} \sum_{\bar{y}^{(i)}\in\mathbb{Y}_{+,+}} e^{-\frac{\left|y-\bar{y}^{(i)}\right|^2}{2\sigma_n^2\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}}. \tag{18}$$

The two marginal conditional PDFs are then given by

$$p(y_R|+1+\mathrm{j}) =\frac{1}{N_{sb}\sqrt{2\pi\sigma_n^2\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}} \sum_{\bar{y}^{(i)}\in\mathbb{Y}_{+,+}} e^{-\frac{\left(y_R-\bar{y}_R^{(i)}\right)^2}{2\sigma_n^2\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}}, \tag{19}$$

$$p(y_I|+1+\mathrm{j}) =\frac{1}{N_{sb}\sqrt{2\pi\sigma_n^2\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}} \sum_{\bar{y}^{(i)}\in\mathbb{Y}_{+,+}} e^{-\frac{\left(y_I-\bar{y}_I^{(i)}\right)^2}{2\sigma_n^2\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}}. \tag{20}$$

Define $b_1(k)=b_{1_R}(k)+\mathrm{j}b_{1_I}(k)$, $\widehat{b}_1(k)=\widehat{b}_{1_R}(k)+\mathrm{j}\widehat{b}_{1_I}(k)$, and

$$P_{E_R}(\boldsymbol{w}) =\mathrm{Prob}\left\{\widehat{b}_{1_R}(k)\neq b_{1_R}(k)\right\}, \tag{21}$$

$$P_{E_I}(\boldsymbol{w}) =\mathrm{Prob}\left\{\widehat{b}_{1_I}(k)\neq b_{1_I}(k)\right\}. \tag{22}$$

It is straightforward to verify that

$$P_{E_R}(\boldsymbol{w}) =\frac{1}{N_{sb}}\sum_{\bar{y}^{(i)}\in\mathbb{Y}_{+,+}} Q\left(g_R^{(i)}(\boldsymbol{w})\right), \tag{23}$$

$$P_{E_I}(\boldsymbol{w}) =\frac{1}{N_{sb}}\sum_{\bar{y}^{(i)}\in\mathbb{Y}_{+,+}} Q\left(g_I^{(i)}(\boldsymbol{w})\right), \tag{24}$$

where

$$Q(u) =\frac{1}{\sqrt{2\pi}}\int_u^{\infty} e^{-\frac{v^2}{2}}\,dv, \tag{25}$$

$$g_R^{(i)}(\boldsymbol{w}) =\frac{\mathrm{sgn}\left(b_{1_R}^{(i)}\right)\bar{y}_R^{(i)}}{\sigma_n\sqrt{\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}} = \frac{\mathrm{sgn}\left(b_{1_R}^{(i)}\right)\Re\left[\boldsymbol{w}^{\mathrm{H}}\bar{\boldsymbol{x}}^{(i)}\right]}{\sigma_n\sqrt{\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}}, \tag{26}$$

$$g_I^{(i)}(\boldsymbol{w}) =\frac{\mathrm{sgn}\left(b_{1_I}^{(i)}\right)\bar{y}_I^{(i)}}{\sigma_n\sqrt{\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}} = \frac{\mathrm{sgn}\left(b_{1_I}^{(i)}\right)\Im\left[\boldsymbol{w}^{\mathrm{H}}\bar{\boldsymbol{x}}^{(i)}\right]}{\sigma_n\sqrt{\boldsymbol{w}^{\mathrm{H}}\boldsymbol{w}}}, \tag{27}$$

Then the BER of the beamformer with $\boldsymbol{w}$ is given by

$$P_E(\boldsymbol{w}) =\frac{1}{2}\left(P_{E_R}(\boldsymbol{w})+P_{E_I}(\boldsymbol{w})\right). \tag{28}$$

The BER can also be calculated based on $\mathbb{Y}_{+,-}$, $\mathbb{Y}_{-,+}$ or $\mathbb{Y}_{-,-}$. The MBER solution is determined by the optimisation

$$\boldsymbol{w}_{\mathrm{mber}} =\arg\min_{\boldsymbol{w}} P_E(\boldsymbol{w}). \tag{29}$$

which can be solved by a gradient-based algorithm, such as the simplified conjugate gradient algorithm [5], [14]–[16].

To adaptively implement the MMSE solution, the unknown second-order statistics or MSE can be estimated based on a block of training data. Furthermore, by considering a single-sample "estimate" of the MSE, the stochastic adaptive algorithm known as the LMS algorithm is derived in (10) to (12). A similar adaptive implementation strategy can be adopted for adaptive MBER beamforming. The PDF $p(y)$ of $y(k)$ can be estimated using the Parzen window estimate [17]–[19] based on a block of training data. This leads to an estimated BER for the beamformer. Minimising this estimated BER based on a gradient optimisation yields an approximated MBER solution [4], [5]. In order to derive a sample-by-sample adaptive algorithm, the works [4], [5] further consider a single-sample "estimate" of $p(y)$

$$\widetilde{p}(y,k) = \frac{1}{2\pi\rho^2} e^{-\frac{|y-y(k)|^2}{2\rho^2}}, \tag{30}$$

where the parameter ρ is known as the kernel width. The corresponding one-sample BER "estimate" is given by

$$\begin{aligned}\widetilde{P}_E(\boldsymbol{w},k) =&\frac{1}{2}\left(\widetilde{P}_{E_R}(\boldsymbol{w},k)+\widetilde{P}_{E_I}(\boldsymbol{w},k)\right)\\ =&\frac{1}{2}\left(Q\left(\widetilde{g}_R(\boldsymbol{w},k)\right)+Q\left(\widetilde{g}_I(\boldsymbol{w},k)\right)\right)\end{aligned} \tag{31}$$

with

$$\widetilde{g}_R(\boldsymbol{w},k) =\frac{\mathrm{sgn}\left(b_{1_R}(k)\right)y_R(k)}{\rho}, \tag{32}$$

$$\widetilde{g}_I(\boldsymbol{w},k) =\frac{\mathrm{sgn}\left(b_{1_I}(k)\right)y_I(k)}{\rho}. \tag{33}$$

The "instantaneous" gradient of $\widetilde{P}_E(\boldsymbol{w},k)$ is given by

$$\nabla\widetilde{P}_E(\boldsymbol{w},k) =\frac{1}{2}\left(\nabla\widetilde{P}_{E_R}(\boldsymbol{w},k)+\nabla\widetilde{P}_{E_I}(\boldsymbol{w},k)\right) \tag{34}$$

with

$$\nabla\widetilde{P}_{E_R}(\boldsymbol{w},k) =-\frac{1}{2\rho\sqrt{2\pi}} e^{-\frac{y_R^2(k)}{2\rho^2}}\,\mathrm{sgn}\left(b_{1_R}(k)\right)\boldsymbol{x}(k), \tag{35}$$

$$\nabla\widetilde{P}_{E_I}(\boldsymbol{w},k) =\frac{1}{2\rho\sqrt{2\pi}} e^{-\frac{y_I^2(k)}{2\rho^2}}\,\mathrm{sgn}\left(b_{1_I}(k)\right)\mathrm{j}\boldsymbol{x}(k). \tag{36}$$

This leads to the stochastic gradient adaptive algorithm referred to as the LBER algorithm [4], [5]

$$\widetilde{\boldsymbol{w}}_{\mathrm{lber}}(k+1) =\widetilde{\boldsymbol{w}}_{\mathrm{lber}}(k)-\mu_{\mathrm{lber}}\nabla\widetilde{P}_E\left(\widetilde{\boldsymbol{w}}_{\mathrm{lber}}(k),k\right), \tag{37}$$

$$c_1 =\widetilde{\boldsymbol{w}}_{\mathrm{lber}}^{\mathrm{H}}(k+1)\boldsymbol{p}_1, \tag{38}$$

$$\widetilde{\boldsymbol{w}}_{\mathrm{lber}}(k+1) =\frac{c_1}{|c_1|}\widetilde{\boldsymbol{w}}_{\mathrm{lber}}(k+1), \tag{39}$$

where μ_{lber} is the step size of the LBER algorithm.

III. The OGMDE-AMBER Beamforming Receiver

We only need to estimate the conditional PDF $p(y|+1+\mathrm{j})$ of (18), which is associated with the conditional subset $\mathbb{Y}_{+,+}$ or the first quadrant of the CV plane. Noting the relationships of (15) to (17), the following shifting operation

$$y_s(k) =y(k)+a\boldsymbol{w}^{\mathrm{H}}\boldsymbol{p}_1 = \boldsymbol{w}^{\mathrm{H}}\left(\boldsymbol{x}(k)+a\boldsymbol{p}_1\right) = \boldsymbol{w}^{\mathrm{H}}\boldsymbol{z}(k), \tag{40}$$

"shifts" the beamformer output $y(k)$ to the first quadrant of the CV plane, where $c_1=\boldsymbol{w}^{\mathrm{H}}\boldsymbol{p}_1$ is real and positive, while

$$a =\left(1-\mathrm{sgn}\left(b_{1_R}(k)\right)\right)+\left(1-\mathrm{sgn}\left(b_{1_I}(k)\right)\right)\mathrm{j}. \tag{41}$$

A. PDF Estimation Using OGMDE

We consider the PDF estimator for $\widehat{p}(y|+1+\mathrm{j})$ based on the mixture of N Gaussians given by

$$\widehat{p}^{(N)}\big(y_s;\boldsymbol{\lambda}_N,\boldsymbol{\eta}_N,\boldsymbol{\rho}_N\big)=\sum_{i=1}^{N}\lambda_i G\big(y_s;\eta_i,\rho_i\big), \tag{42}$$

$$\text{s.t. } \lambda_i>0,\ 1\leq i\leq N, \text{ and } \boldsymbol{\lambda}_N^{\mathrm{T}}\mathbf{1}_N=1, \tag{43}$$

where λ_i, η_i and ρ_i are the RV mixing weight, the CV mean and the RV kernel width of the ith Gaussian kernel, respectively, $\boldsymbol{\lambda}_N=\big[\lambda_1\ \lambda_2\cdots\lambda_N\big]^{\mathrm{T}}$, $\boldsymbol{\eta}_N=\big[\eta_1\ \eta_2\cdots\eta_N\big]^{\mathrm{T}}$, $\boldsymbol{\rho}_N=\big[\rho_1\ \ \rho_2\cdots\rho_N\big]^{\mathrm{T}}$, $\mathbf{1}_N$ is the N-dimensional vector whose elements are all equal to one, and

$$G\big(y_s;\eta_i,\rho_i\big)=\frac{1}{2\pi\rho_i^2}e^{-\dfrac{|y_s-\eta_i|^2}{2\rho_i^2}}. \tag{44}$$

The two marginal PDFs of (44) are understood to be

$$G\big(y_{s_R};\eta_{i_R},\rho_i\big)=\frac{1}{\sqrt{2\pi}\rho_i}e^{-\dfrac{\big(y_{s_R}-\eta_{i_R}\big)^2}{2\rho_i^2}}, \tag{45}$$

$$G\big(y_{s_I};\eta_{i_I},\rho_i\big)=\frac{1}{\sqrt{2\pi}\rho_i}e^{-\dfrac{\big(y_{s_I}-\eta_{i_I}\big)^2}{2\rho_i^2}}, \tag{46}$$

where $\eta_i=\eta_{i_R}+\mathrm{j}\eta_{i_I}$. The PDF estimator (42) with a small number of Gaussian mixtures N is capable of accurately estimating an arbitrary and unknown PDF $p(y|+1+\mathrm{j})$, and $N=4$ to 8 is sufficient for our application. At sample time $k=0$, the initial estimate for $p(y|+1+\mathrm{j})$ is given by

$$\widehat{p}^{(N)}\big(y_s;\boldsymbol{\lambda}_N(0),\boldsymbol{\eta}_N(0),\boldsymbol{\rho}_N(0)\big)=\sum_{i=1}^{N}\frac{1}{N}G\big(y_s;\eta_i(0),\rho_0\big), \tag{47}$$

where for $1\leq i\leq N$, $\lambda_i(0)=\frac{1}{N}$ and $\rho_i(0)=\rho_0$ with ρ_0 a predetermined kernel width, while $\eta_i(0)$ are randomly drawn points from the first quadrant of the CV plane.

At sample time k, the new data point $y_s(k)$ is received, and we need to update $\big\{\boldsymbol{\lambda}_N,\boldsymbol{\eta}_N,\boldsymbol{\rho}_N\big\}$ in the PDF estimate

$$\widehat{p}^{(N)}\big(y_s;\boldsymbol{\lambda}_N(k-1),\boldsymbol{\eta}_N(k-1),\boldsymbol{\rho}_N(k-1)\big)=\sum_{i=1}^{N}\lambda_i(k-1)G\big(y_s;\eta_i(k-1),\rho_i(k-1)\big) \tag{48}$$

accordingly, while keeping the same number of mixtures N as well as meeting the constraint (43). A nature way is to place a Gaussian kernel on $y_s(k)$ and to merge this new kernel with its nearest existing mixture component $G\big(y_s;\eta_{i'}(k-1),\rho_{i'}(k-1)\big)$, where

$$i^{'}=\arg\min_{1\leq i\leq N}\big|y_s(k)-\eta_i(k-1)\big|. \tag{49}$$

This can be realised in the following two steps.

1) A temporary estimate with $(N+1)$ Gaussian mixtures is first created by adding the newly created $(N+1)$th Gaussian kernel based on $y_s(k)$ to the estimate (48) according to

$$\begin{aligned}\widehat{p}^{(N+1)}\big(y_s;\boldsymbol{\lambda}_{N+1}(k),\boldsymbol{\eta}_{N+1}(k),\boldsymbol{\rho}_{N+1}(k)\big)&=\\ \frac{N}{N+1}\sum_{i=1}^{N}\lambda_i(k-1)G\big(y_s;\eta_i(k-1),\rho_i(k-1)\big)&\\ +\frac{1}{N+1}G\big(y_s;y_s(k),\rho_0\big).&\end{aligned} \tag{50}$$

Clearly, we set $\lambda_{N+1}(k)=\dfrac{1}{N+1}$ and

$$\lambda_i(k)=\frac{N\lambda_i(k-1)}{N+1},\ 1\leq i\leq N, \tag{51}$$

to satisfy the constraints $\lambda_i(k)>0$ for $1\leq i\leq N+1$ and $\boldsymbol{\lambda}_{N+1}^{\mathrm{T}}(k)\mathbf{1}_{N+1}=1$, while using $\eta_{N+1}(k)=y_s(k)$, $\rho_{N+1}(k)=\rho_0$ as well as for $1\leq i\leq N$

$$\eta_i(k)=\eta_i(k-1), \tag{52}$$

$$\rho_i(k)=\rho_i(k-1). \tag{53}$$

2) Merge the $i^{'}$th, where $i^{'}$ is determined in (49), and $(N+1)$th mixtures in the temporary estimate (50) into the single new $i^{'}$th mixture, so that

$$\begin{aligned}\lambda_{i'}(k)G\big(y_s;\eta_{i'}(k),\rho_{i'}(k)\big)&\approx\\ \frac{N\lambda_{i'}(k-1)}{N+1}G\big(y_s;\eta_{i'}(k-1),\rho_{i'}(k-1)\big)&\\ +\frac{1}{N+1}G\big(y_s;y_s(k),\rho_0\big).&\end{aligned} \tag{54}$$

Thus, the new $i^{'}$th weight $\lambda_{i'}(k)$ is given by

$$\lambda_{i'}(k)=\frac{N\lambda_{i'}(k-1)+1}{N+1}, \tag{55}$$

while the new $i^{'}$th mean and kernel width, $\eta_{i'}(k)$ and $\rho_{i'}(k)$, are updated by matching the mean and kernel width of the two mixtures with the new single Gaussian, leading to

$$\eta_{i'}(k)=\frac{N\lambda_{i'}(k-1)\eta_{i'}(k-1)+y_s(k)}{N\lambda_{i'}(k-1)+1}, \tag{56}$$

$$\begin{aligned}2\rho_{i'}^2(k)=&\frac{N\lambda_{i'}(k-1)\big(2\rho_{i'}^2(k-1)+\big|\eta_{i'}(k-1)\big|^2\big)}{N\lambda_{i'}(k-1)+1}\\ &+\frac{2\rho_0^2+|y_s(k)|^2}{N\lambda_{i'}(k-1)+1}-\big|\eta_{i'}(k)\big|^2.\end{aligned} \tag{57}$$

The derivations of (56) and (57) are given in Appendix A.

The PDF of the decision variable $y_s(k)$ at sample time k can thus be approximated by

$$\begin{aligned}\widehat{p}\big(y_s,k\big)=&\widehat{p}^{(N)}\big(y_s;\boldsymbol{\lambda}_N(k),\boldsymbol{\eta}_N(k),\boldsymbol{\rho}_N(k)\big)\\ =&\sum_{i=1}^{N}\lambda_i(k)G\left(y_s;\eta_i(k),\rho_i(k)\right),\end{aligned} \tag{58}$$

in which the $i^{'}$th weight, mean and kernel width are given in (55), (56) and (57), respectively, while the ith weights, means and kernel widths, where $1\leq i\leq N$ and $i\neq i^{'}$, are given in (51), (52) and (53), respectively. Note that only $\eta_{i'}(k)$ and $\rho_{i'}^2(k)$ contain the new information provided by $y_s(k)$ and, therefore only $\eta_{i'}(k)$ and $\rho_{i'}^2(k)$ depend on the current beamformer wight vector $\boldsymbol{w}$.

B. OGMDE-AMBER Beamforming

Given the PDF estimate (58) provided by the OGMDE, we have the corresponding approximate BER expression

$$\widehat{P}_E(\boldsymbol{w},k) = \frac{1}{2}\big(\widehat{P}_{E_R}(\boldsymbol{w},k) + \widehat{P}_{E_I}(\boldsymbol{w},k)\big). \tag{59}$$

We now explicitly derive $\widehat{P}_{E_R}(\boldsymbol{w},k)$ as follows

$$\begin{aligned}\widehat{P}_{E_R}(\boldsymbol{w},k) &= \int_{-\infty}^{0} \sum_{i=1}^{N} \lambda_i(k) G\left(y_{s_R}; \eta_{i_R}(k), \rho_i(k)\right) dy_{s_R} \\ &= \sum_{i=1, i\neq i'}^{N} \lambda_i(k) \int_{-\infty}^{0} G\left(y_{s_R}; \eta_{i_R}(k), \rho_i(k)\right) dy_{s_R} + \widehat{B}_{E_R}(\boldsymbol{w},k),\end{aligned} \tag{60}$$

where

$$\begin{aligned}\widehat{B}_{E_R}(\boldsymbol{w},k) =& \frac{\lambda_{i'}(k)}{\sqrt{2\pi}\rho_{i'}(k)} \int_{-\infty}^{0} e^{-\frac{\left(y_{s_R} - \eta_{i'_R}(k)\right)^2}{2\rho_{i'}^2(k)}} dy_{s_R} \\ =& \lambda_{i'}(k) Q\big(g_{i'_R}(\boldsymbol{w},k)\big)\end{aligned} \tag{61}$$

with

$$g_{i'_R}(\boldsymbol{w},k) = \frac{\eta_{i'_R}(k)}{\rho_{i'}(k)}. \tag{62}$$

Similarly,

$$\begin{aligned}\widehat{P}_{E_I}(\boldsymbol{w},k) =& \sum_{i=1, i\neq i'}^{N} \lambda_i(k) \int_{-\infty}^{0} G\left(y_{s_I}; \eta_{i_I}(k), \rho_i(k)\right) dy_{s_I} \\ &+ \widehat{B}_{E_I}(\boldsymbol{w},k),\end{aligned} \tag{63}$$

where

$$\widehat{B}_{E_I}(\boldsymbol{w},k) = \lambda_{i'}(k) Q\big(g_{i'_I}(\boldsymbol{w},k)\big) \tag{64}$$

with

$$g_{i'_I}(\boldsymbol{w},k) = \frac{\eta_{i'_I}(k)}{\rho_{i'}(k)}. \tag{65}$$

The "instantaneous" gradient of $\widehat{P}_E(\boldsymbol{w},k)$ is given by

$$\nabla \widehat{P}_E(\boldsymbol{w},k) = \frac{1}{2}\Big(\nabla \widehat{B}_{E_R}(\boldsymbol{w},k) + \nabla \widehat{B}_{E_I}(\boldsymbol{w},k)\Big) \tag{66}$$

in which

$$\nabla \widehat{B}_{E_R}(\boldsymbol{w},k) = -\frac{\lambda_{i'}(k)}{\sqrt{2\pi}} e^{-\frac{\eta_{i'_R}^2(k)}{2\rho_{i'}^2(k)}} \frac{\partial g_{i'_R}(\boldsymbol{w},k)}{\partial \boldsymbol{w}}, \tag{67}$$

$$\nabla \widehat{B}_{E_I}(\boldsymbol{w},k) = -\frac{\lambda_{i'}(k)}{\sqrt{2\pi}} e^{-\frac{\eta_{i'_I}^2(k)}{2\rho_{i'}^2(k)}} \frac{\partial g_{i'_I}(\boldsymbol{w},k)}{\partial \boldsymbol{w}}. \tag{68}$$

After some manipulations as shown in Appendix B, we have

$$\frac{\partial g_{i'_R}(\boldsymbol{w},k)}{\partial \boldsymbol{w}} = \frac{2\rho_{i'}^2(k) - \eta_{i'_R}(k)\big(y_s^*(k) - \eta_{i'}^*(k)\big)}{4\rho_{i'}^3(k)\big(N\lambda_{i'}(k-1)+1\big)} \boldsymbol{z}(k), \tag{69}$$

$$\frac{\partial g_{i'_I}(\boldsymbol{w},k)}{\partial \boldsymbol{w}} = \frac{-2\rho_{i'}^2(k)\mathrm{j} - \eta_{i'_I}(k)\big(y_s^*(k) - \eta_{i'}^*(k)\big)}{4\rho_{i'}^3(k)\big(N\lambda_{i'}(k-1)+1\big)} \boldsymbol{z}(k). \tag{70}$$

This leads to the OGMDE-AMBER algorithm

$$\widehat{\boldsymbol{w}}_{\text{amber}}(k+1) = \widehat{\boldsymbol{w}}_{\text{amber}}(k) - \mu_{\text{amber}} \nabla \widehat{P}_E\big(\widehat{\boldsymbol{w}}_{\text{amber}}(k), k\big), \tag{71}$$

$$c_1 = \widehat{\boldsymbol{w}}_{\text{amber}}^{\mathrm{H}}(k+1)\boldsymbol{p}_1, \tag{72}$$

$$\widehat{\boldsymbol{w}}_{\text{amber}}(k+1) = \frac{c_1}{|c_1|} \widehat{\boldsymbol{w}}_{\text{amber}}(k+1), \tag{73}$$

where μ_{amber} is the step size of the OGMDE-AMBER algorithm.

IV. Simulation Results

The simulated system consisted of $M = 4$ users and the receiver linear uniform antenna array had $L = 3$ elements. The array element spacing was $\gamma/2$ with γ being the wavelength. Fig. 1 illustrates the locations of the desired user and three interfering users graphically. The simulated channel conditions were $A_m = b_m + \mathrm{j}\,0.0$ for $1 \le m \le 4$, with $b_m > 0$ so chosen to provided the required received signal powers. Specifically, the desired user 1 as well as the interfering users 2 and 3 had the equal power, but the interfering user 4 had 6 dB more power than the user 1. Therefore, $\text{SIR}_2 = \text{SIR}_3 = 0$ dB, while $\text{SIR}_4 = -6$ dB. Fig. 2 compares the BER of the MMSE beamformer with that of the MBER beamformer, where the superior performance of the MBER beamforming technique is evident. The MMSE solution was calculated according to the closed-form solution of (9), while the MBER solution was obtained numerically by solving the optimisation (29) using the simplified conjugate gradient algorithm. The performance difference between the MMSE solution and the MBER solution can be clearly explained by examining their corresponding marginal conditional PDFs $p(y_R|+1+\mathrm{j})$ depicted in Fig. 3, given SNR $= 17$ dB. Note that owing to the symmetric distribution of the conditional PDF $p(y|+1+\mathrm{j})$, its two marginal conditional PDFs, $p(y_R|+1+\mathrm{j})$ and $p(y_I|+1+\mathrm{j})$, are identical. Therefore, we only need to inspect one of them. From Fig. 3, it can be seen that the marginal conditional PDF $p(y_R|+1+\mathrm{j})$ of the MMSE solution clearly extends into the area of $y_R > 0$, and

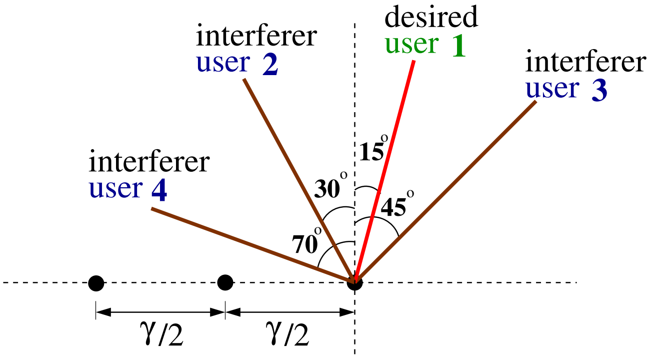

Fig. 1. Angular locations of the four users with respect to the three-element linear array having $\gamma/2$ element spacing, where γ is the wavelength.

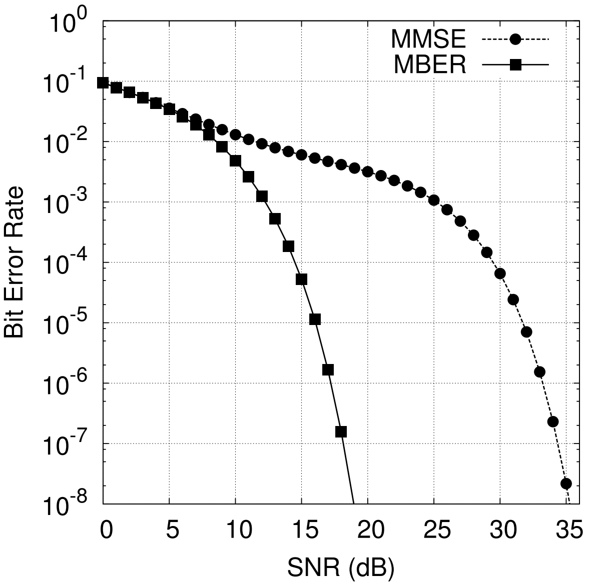

Fig. 2. Bit error rate comparison of the MMSE and MBER beamformers for the 4-user 3-element linear antenna array system shown in Fig. 1, where $\mathrm{SIR}_2 = \mathrm{SIR}_3 = 0\,\mathrm{dB}$ and $\mathrm{SIR}_4 = -6\,\mathrm{dB}$.

this explains its inferior BER performance shown in Fig. 2, compared with the MBER solution.

The performance of the stochastic LBER algorithm is well known to depend on the initial weight vector $\widetilde{\boldsymbol{w}}_{\mathrm{lber}}(0)$ [4], [5], [14], [15]. Clearly, the performance of the OGMDE-AMBER algorithm also depends on the initial weight vector $\widetilde{\boldsymbol{w}}_{\mathrm{amber}}(0)$, as it also relies on a stochastic gradient based sample-by-sample updating. We first set the initial weight vector to the MMSE solution and examined the convergence performance of the three stochastic gradient algorithms, the LMS, the LBER and the OGMDE-AMBER, in Fig. 4, where $\mathrm{SNR} = 17\,\mathrm{dB}$ and the learning curve of each adaptive algorithm was averaged over 100 runs. An appropriate step size for the LMS algorithm was found to be $\mu_{\mathrm{lms}} = 0.003$, while the step size $\mu_{\mathrm{lber}} = 0.03$ and the kernel variance $\rho^2 = 3\sigma_n^2 \approx 0.06$ were found to be appropriate for the LBER algorithm. For the OGMDE-AMBER algorithm, we chose $N = 4$ and $\rho_0^2 = 2\sigma_n^2 \approx 0.04$, while starting with a large step size of $\mu_{\mathrm{amber}} = 0.4$ and reducing it to $\mu_{\mathrm{amber}} = 0.2$ at sample number $k \approx 300$. As expected, the BER of the LMS algorithm could not be lower than the MMSE solution. Fig. 4 confirms that the OGMDE-AMBER algorithm achieved a faster convergence rate and attained a lower steady-state BER, compared with the LBER algorithm.

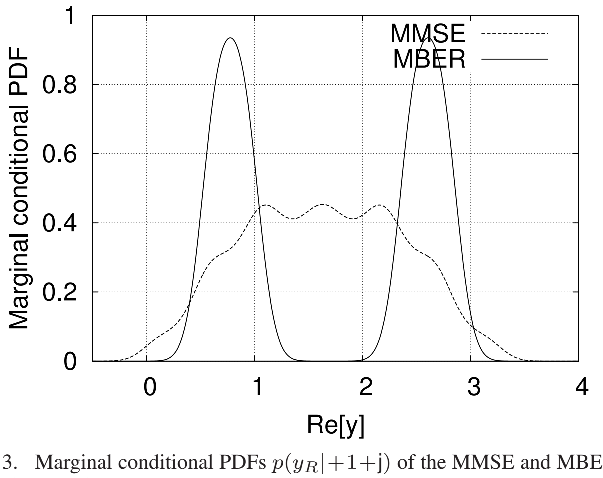

Fig. 3. Marginal conditional PDFs $p(y_R|+1+\mathrm{j})$ of the MMSE and MBER beamformers for the 4-user 3-element linear antenna array system shown in Fig. 1, where $\mathrm{SNR} = 17\,\mathrm{dB}$, $\mathrm{SIR}_2 = \mathrm{SIR}_3 = 0\,\mathrm{dB}$ and $\mathrm{SIR}_4 = -6\,\mathrm{dB}$.

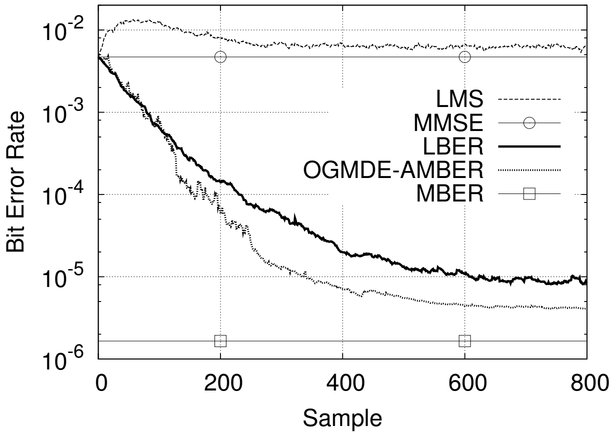

Fig. 4. Learning curves of the three stochastic gradient adaptive algorithms, the LMS, the LBER and the OGMDE-AMBER, averaged over 100 runs for the 4-user 3-element linear antenna array system shown in Fig. 1, where $\mathrm{SNR} = 17\,\mathrm{dB}$, $\mathrm{SIR}_2 = \mathrm{SIR}_3 = 0\,\mathrm{dB}$ and $\mathrm{SIR}_4 = -6\,\mathrm{dB}$. The initial weight vector was set to the MMSE solution.

We next set the initial beamformer's weight vector to $[0.0+0.1\mathrm{j}\ 0.1+0.0\mathrm{j}\ 0.1+0.0\mathrm{j}]^{\mathrm{T}}$ and examined the learning curves of the three stochastic gradient adaptive algorithms in Fig. 5, where $\mathrm{SNR} = 17\,\mathrm{dB}$ and the results were averaged over 100 runs. Again, we set the step size of the LMS algorithm to $\mu_{\mathrm{lms}} = 0.003$, while we used the step size $\mu_{\mathrm{lber}} = 0.03$ and the kernel variance $\rho^2 = 3\sigma_n^2 \approx 0.06$ for the LBER algorithm. For the OGMDE-AMBER algorithm, we used $N = 5$ Gaussian components and set $\rho_0^2 = 2\sigma_n^2 \approx 0.04$. We started with the step size of $\mu_{\mathrm{amber}} = 0.4$ and reduced it to $\mu_{\mathrm{amber}} = 0.2$ at sample number $k \approx 200$.

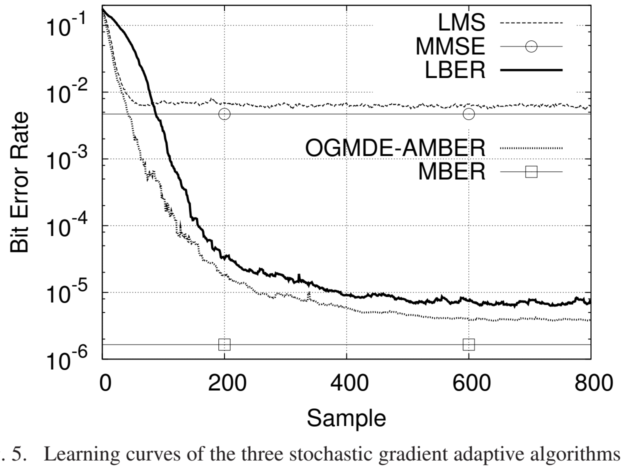

Fig. 5. Learning curves of the three stochastic gradient adaptive algorithms, the LMS, the LBER and the OGMDE-AMBER, averaged over 100 runs for the 4-user 3-element linear antenna array system shown in Fig. 1, where $\mathrm{SNR} = 17\,\mathrm{dB}$, $\mathrm{SIR}_2 = \mathrm{SIR}_3 = 0\,\mathrm{dB}$ and $\mathrm{SIR}_4 = -6\,\mathrm{dB}$. The initial weight vector was set to $[0.0+0.1\mathrm{j}\ 0.1+0.0\mathrm{j}\ 0.1+0.0\mathrm{j}]^{\mathrm{T}}$.

The results shown in Fig. 5 again demonstrate the superior performance of the OGMDE-AMBER algorithm over the LBER algorithm, in terms of both convergence rate and steady-state BER.

V. Conclusions

In this contribution, we have developed an on-line Gaussian mixture density estimator in the complex-valued domain, which adaptively model the probability density function of the beamformer output by tracking the incoming data sample by sample. With the aid of this novel OGMDE, our proposed stochastic-gradient based adaptive minimum bit error rate beamforming receiver is capable of directly minimising the system's achievable bit error rate by adapting the bermforming weight vector sample by sample. The simulation results obtained have demonstrated that this new OGMDE-AMBER algorithm outperforms the existing stochastic-gradient based adaptive MBER algorithm, known as the least bit error rate algorithm, in terms of both convergence rate and steady-state bit error rate.

References

[1] J. Litva and T. K. Y. Lo, *Digital Beamforming in Wireless Communications*. London: Artech House, 1996.

[2] P. Vandenameele, L. van Der Perre, and M. Engels, *Space Division Multiple Access for Wireless Local Area Networks*. Boston: Kluwer Academic Publishers, 2001.

[3] J. S. Blogh and L. Hanzo, *Third Generation Systems and Intelligent Wireless Networking – Smart Antenna and Adaptive Modulation*. Chichester, U.K.: John Wiley, 2002.

[4] S. Chen, N. N. Ahmad, and L. Hanzo, "Adaptive minimum bit error rate beamforming," *IEEE Trans. Wireless Communications*, vol. 4, no. 2, pp. 341–348, March 2005.

[5] S. Chen, L. Hanzo, N. N. Ahmad, and A. Wolfgang, "Adaptive minimum bit error rate beamforming assisted receiver for QPSK wireless communication," *Digital Signal Processing*, vol. 15, no. 6, pp. 545–567, Nov. 2005.

[6] S. Haykin, *Adaptive Filter Theory* (2nd Edition). Englewood, NJ: Prentice Hall, 1991.

[7] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute*, vol. 4, no. 510, 281 pages, 1998.

[8] G. McLachlan and D. Peel, *Finite Mixture Models*. John Wiley & Sons, Inc., 2004.

[9] S. Chen, X. Hong, and C. J. Harris, "Sparse kernel density construction using orthogonal forward regression with leave-one-out test score and local regularization," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 34, no. 4, pp. 1708–1717, Aug. 2004.

[10] X. Hong, S. Chen, and C. J. Harris, "A forward-constrained regression algorithm for sparse kernel density estimation," *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 193–198, Jan. 2008.

[11] S. Chen, X. Hong, and C. J. Harris, "An orthogonal forward regression technique for sparse kernel density estimation," *Neurocomputing*, vol. 71, nos. 4-6, pp. 931–943, Jan. 2008.

[12] S. Chen, X. Hong, and C. J. Harris, "Regression based D-optimality experimental design for sparse kernel density estimation," *Neurocomputing*, vol. 73, nos. 4-6, pp. 727–739, Jan. 2010.

[13] S. Chen, X. Hong, and C. J. Harris, "Particle swarm optimization aided orthogonal forward regression for unified data modelling," *IEEE Trans. Evolutionary Computation*, vol. 14, no. 4, pp. 477–499, Aug. 2010.

[14] S. Chen, A. K. Samingan, B. Mulgrew, and L. Hanzo, "Adaptive minimum-BER linear multiuser detection for DS-CDMA signals in multipath channels," *IEEE Trans. Signal Processing*, vol. 49, no. 6, pp. 1240–1247, June 2001.

[15] S. Chen, A. Livingstone, H.-Q. Du, and L. Hanzo, "Adaptive minimum symbol error rate beamforming assisted detection for quadrature amplitude modulation," *IEEE Trans. Wireless Communications*, vol. 7, no. 4, pp. 1140–1145, April 2008.

[16] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. New York: John Wiley, 1993.

[17] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, pp. 1066–1076, 1962.

[18] B. W. Silverman, *Density Estimation*. London: Chapman Hall, 1996.

[19] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis*. Oxford, U.K.: Oxford University Press, 1997.

Appendix

A. Merging Two Gaussians as One

Consider merging a mixture of two Gaussians

$$\widehat{p}^{(2)}\big(y_s;\boldsymbol{\lambda}_2,\boldsymbol{\eta}_2,\boldsymbol{\rho}_2\big)=\sum_{i=1}^{2}\lambda_i G\big(y_s;\eta_i,\rho_i\big), \tag{74}$$

into one mixture by matching the resultant mean and kernel width. The CV mean η of the two mixtures is given by

$$\begin{aligned}\eta =&\sum_{i=1}^{2}\lambda_i\bigg(\int y_{s_R}G\big(y_{s_R};\eta_{i_R},\rho_i\big)\,dy_{s_R}\\ &+\mathrm{j}\int y_{s_I}G\big(y_{s_I};\eta_{i_I},\rho_i\big)\,dy_{s_I}\bigg)=\sum_{i=1}^{2}\lambda_i\big(\eta_{i_R}+\mathrm{j}\eta_{i_I}\big)\\ =&\sum_{i=1}^{2}\lambda_i\eta_i,\end{aligned} \tag{75}$$

while the RV kernel width ρ of the two mixtures satisfies

$$\begin{aligned}2\rho^2 =&\sum_{i=1}^{2}\lambda_i\int y_{s_R}^2 G\big(y_{s_R},\eta_{i_R},\rho_i\big)\,dy_{s_R}-\eta_R^2\\ &+\sum_{i=1}^{2}\lambda_i\int y_{s_I}^2 G\big(y_{s_I},\eta_{i_I},\rho_i\big)\,dy_{s_I}-\eta_I^2\\ =&\sum_{i=1}^{2}\lambda_i\big(\rho_i^2+\eta_{i_R}^2\big)-\eta_R^2+\sum_{i=1}^{2}\lambda_i\big(\rho_i^2+\eta_{i_I}^2\big)-\eta_I^2\\ =&\sum_{i=1}^{2}\lambda_i\big(2\rho_i^2+\big|\eta_i\big|^2\big)-\big|\eta\big|^2.\end{aligned} \tag{76}$$

B. The Derivation of $\frac{\partial g_{i'\,R}(\boldsymbol{w},k)}{\partial \boldsymbol{w}}$ and $\frac{\partial g_{i'\,I}(\boldsymbol{w},k)}{\partial \boldsymbol{w}}$

From (62) and (65), we have

$$\frac{\partial g_{i'_R}(\boldsymbol{w},k)}{\partial \boldsymbol{w}}=\frac{1}{\rho_{i'}^2(k)}\bigg(\rho_{i'}(k)\frac{\partial \eta_{i'_R}(k)}{\partial \boldsymbol{w}}-\eta_{i'_R}(k)\frac{\partial \rho_{i'}(k)}{\partial \boldsymbol{w}}\bigg), \tag{77}$$

$$\frac{\partial g_{i'_I}(\boldsymbol{w},k)}{\partial \boldsymbol{w}}=\frac{1}{\rho_{i'}^2(k)}\bigg(\rho_{i'}(k)\frac{\partial \eta_{i'_I}(k)}{\partial \boldsymbol{w}}-\eta_{i'_I}(k)\frac{\partial \rho_{i'}(k)}{\partial \boldsymbol{w}}\bigg). \tag{78}$$

Noting $y_s(k)=\boldsymbol{w}^{\mathrm{H}}\boldsymbol{z}(k)$ of (40) and $\eta_{i'}(k)$ of (56), the partial derivatives of $\eta_{i'_R}(k)$ and $\eta_{i'_I}(k)$ with respect to $\boldsymbol{w}$ are given respectively by

$$\frac{\partial \eta_{i'_R}(k)}{\partial \boldsymbol{w}}=\frac{1}{2\big(N\lambda_{i'}(k-1)+1\big)}\boldsymbol{z}(k), \tag{79}$$

$$\frac{\partial \eta_{i'_I}(k)}{\partial \boldsymbol{w}} = -\frac{\mathrm{j}}{2\big(N\lambda_{i'}(k-1)+1\big)}\boldsymbol{z}(k). \tag{80}$$

Also we have

$$\frac{\partial \rho_{i'}(k)}{\partial \boldsymbol{w}} = \frac{1}{4\rho_{i'}(k)}\frac{\partial 2\rho^2_{i'}(k)}{\partial \boldsymbol{w}}. \tag{81}$$

From (57) as well as (79) and (80), we have

$$\frac{\partial 2\rho^2_{i'}(k)}{\partial \boldsymbol{w}} = \frac{y^*_s(k) - \eta^*_{i'}(k)}{N\lambda_{i'}(k-1)+1}\boldsymbol{z}(k). \tag{82}$$

Thus

$$\frac{\partial \rho_{i'}(k)}{\partial \boldsymbol{w}} = \frac{y^*_s(k) - \eta^*_{i'}(k)}{4\rho_{i'}(k)\big(N\lambda_{i'}(k-1)+1\big)}\boldsymbol{z}(k). \tag{83}$$

Substituting (79) and (83) into (77) yields

$$\frac{\partial g_{i'_R}(\boldsymbol{w},k)}{\partial \boldsymbol{w}} = \frac{2\rho^2_{i'}(k) - \eta_{i'_R}(k)\big(y^*_s(k) - \eta^*_{i'}(k)\big)}{4\rho^3_{i'}(k)\big(N\lambda_{i'}(k-1)+1\big)}\boldsymbol{z}(k), \tag{84}$$

while using (80) and (83) in (78) leads to

$$\frac{\partial g_{i'_I}(\boldsymbol{w},k)}{\partial \boldsymbol{w}} = \frac{-2\rho^2_{i'}(k)\mathrm{j} - \eta_{i'_I}(k)\big(y^*_s(k) - \eta^*_{i'}(k)\big)}{4\rho^3_{i'}(k)\big(N\lambda_{i'}(k-1)+1\big)}\boldsymbol{z}(k). \tag{85}$$