Stop-and-Go Algorithms for Blind Channel Equalization in QAM Data Communication Systems

Shafayat Abrar
Electrical Engineering Department,
COMSATS-IIT (CIIT), Islamabad, Pakistan.
sabrar@comsats.edu.pk, shafayat1972@yahoo.com

Abstract: *This paper presents a framework of stop-and-go algorithms for the blind equalization of QAM data communication systems. Based on the proposed framework, three new stop-and-go algorithms are presented. It is discussed that each of the existing and proposed stop-and-go algorithm forces the equalizer output to match some statistical contour on QAM constellation space. Using computer simulations, it is shown that the Picchi and Prati's "stop-and-go decision-directed algorithm", which forces the equalizer output to lie on point contours, is the best among other stop-and-go algorithms.*

Keywords: *Blind equalization, adaptive equalizers, stop-and-go, decision-directed, QAM.*

I. INTRODUCTION

In most digital communication systems, inter-symbol interference (ISI) occurs due to bandwidth limited channel or multipath propagation. Channel equalization is one of the techniques to mitigate the effect of ISI. Adaptive algorithms are used to initialize and adjust equalizer coefficients when a channel is unknown and possibly time-varying. Conventionally, initial setting of the equalizer tap weights is achieved by a training sequence before data transmission.

However, when sending a training sequence is impractical or impossible, it is desirable to equalize a channel without the aid of a training sequence. Equalizing a channel without training mode is known as *blind equalization*. A typical blind equalization setup is depicted in Fig. 1 using a simple system diagram. The complex baseband model for a typical QAM (quadrature amplitude modulated) data communication system consists of an unknown linear time-invariant (LTI) channel c_n which represents the physical inter-connection between the transmitter and the receiver at baseband. The transmitter generates a sequence of complex-valued random input data $\{a_n\}$, each element of which belongs to a complex alphabet $\mathcal{A}$ of QAM symbols. The data sequence $\{a_n\}$ is sent through a complex LTI channel whose output x_n is observed by the receiver. The function of the blind equalizer at the receiver is to estimate the original data $\{a_n\}$ from the received signal x_n. The input/output relationship of the QAM system can be written as:

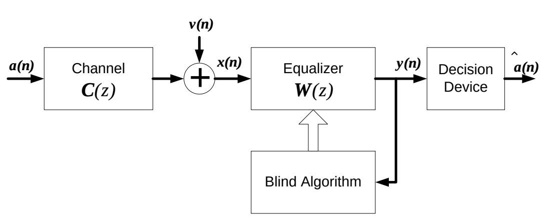

Fig. 1. Baseband model of communication system.

$$x_n = \sum_{i=0}^{K-1} a_{n-iT} c_i + \nu_n, \tag{1}$$

where T is the symbol (or baud) period and K is the length of channel impulse-response. The channel noise ν_n is assumed to be stationary, Gaussian, and independent of the channel input a_n. Denote the equalizer parameter vector with $N+1$ elements as $\mathbf{w}_n = [w_{0,n}, w_{1,n}, \cdots, w_{N,n}]^T$, where the superscript ^{T} represents transpose. In addition, define the received signal vector as $\mathbf{x}_n = [x_n, x_{n-1}, \cdots, x_{n-N+1}]^T$. The output signal of the equalizer is thus $y_n = \mathbf{w}_n^T \mathbf{x}_n = y_{R,n} + \jmath\, y_{I,n}$, where $_R$ and $_I$ represent the real and imaginary parts of y_n, respectively. If $\{h\} = \{c\} * \{w\}$ represents the overall channel-equalizer impulse response. The channel output x_n can be expressed as:

$$x_n \;=\; \sum_i a_{n-iT} h_i + \nu_n = \underbrace{h_0 a_n + \sum_{i \neq 0} a_i h_{n-iT} + \nu_n}_{\text{signal+ISI+noise}}.$$

In blind equalization, the channel input a_{n-D} is unavailable, and thus different minimization criteria are explored. The crudest blind equalization algorithm is the decision-directed scheme that updates the adaptive equalizer coefficients according to

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu(y_n - \mathcal{D}[y_n])\mathbf{x}_n^* \tag{2}$$

where $\mathcal{D}[y_n]$ is the closest symbol to y_n. Under high ISI, the convergence behavior of decision-directed equalizer is very poor. Better blind adaptive equalization algorithms are designed to minimizing special non-MSE cost functions that do not directly involve the input a_n while still reflect the current level of ISI in the equalizer output. Define the mean cost function as:

$$J(\mathbf{w}) = E[\Psi[y_n]] \tag{3}$$

where $\Psi[\cdot]$ is a scalar function of the equalizer output. $J(\mathbf{w})$ should be specified such that at its minimum, the corresponding $\mathbf{w}_n$ results in a minimum ISI or MSE equalizer. Using

(3), the stochastic gradient descent minimization algorithm is easily derived as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \cdot \frac{\partial \Psi[y_n]}{\partial \mathbf{w}_n} = \mathbf{w}_n - \mu \cdot \Psi'[y_n]\mathbf{x}_n^*.$$

Let ψ be the first derivative of Ψ. ψ is often called the *error function* since it replaces the prediction error in the LMS adaptation. The resulting blind equalization algorithm can be written as:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \cdot \psi[y_n]\mathbf{x}_n^*. \tag{4}$$

Thus the design of the blind equalizer thus translates into the selection of a suitable function Ψ (or ψ) such that the local minima of $J(\mathbf{w})$ correspond to a significant removal of ISI from the equalizer output y_n.

II. STOP-AND-GO BLIND EQUALIZATION ALGORITHMS

Given the standard form of the blind equalization algorithm in (4), it is apparent that the convergence characteristics of blind algorithms are largely determined by the sign of the error signal $\psi[y_n]$. In order, for the coefficients of a blind equalizer, to converge to the vicinity of the optimum minimum-MSE (MMSE) solution as achieved by LMS adaptation (under supervision), the sign of its error signal should agree with the sign of the LMS prediction error $y_n - a_{n-D}$ most of the time. Slow or ill convergence can occur if the sign of the two errors differ sufficiently often.

The idea behind the stop-and-go algorithms is to allow the adaptation "to go" only when the error function is more likely to have the correct sign for the gradient descent direction. Given several criteria for blind equalization, one can expect a more accurate descent direction when more than one of the existing algorithms agree on the sign (direction) of the error functions. When the error signs differ for a particular output sample, parameter adaptation is "stopped". Consider two algorithms with error functions ψ_1 and ψ_2. The following stop-and-go algorithm can be devised [1]:

$$\mathbf{w}_{n+1} = \begin{cases} \mathbf{w}_n - \mu\psi_1[y_n]\mathbf{x}_n^*, & \text{if } \mathrm{sgn}[\psi_1[y_n]] = \mathrm{sgn}[\psi_2[y_n]] \\ \mathbf{w}_n, & \text{if } \mathrm{sgn}[\psi_1[y_n]] \neq \mathrm{sgn}[\psi_2[y_n]] \end{cases} \tag{5}$$

Error functions ψ_1 and ψ_2 should be selected such that they maximize reliable regions and make most of the local and the global knowledge of the constellation. Given the equalizer output y_n, the closest symbol $\hat{a}_n = \mathcal{D}[y_n]$ can be considered as a local information; while the size (number of alphabets), shape (square or cross) and energy (mean distance between the symbols) of the constellation can be considered as global information. $\hat{a}_n$ is termed local as it may change from one output to another; while the size, shape and energy are fixed and don't depend on any specific value of y_n. Most of the stochastic gradient descent algorithms employ error functions which exhibit global information. They compute an estimate of a_n, by doing some nonlinear operation on the current equalizer output y_n such that the certain statistics of y_n are forced to match with global statistics of the transmitted data symbols. Let us consider the following error function (say global), which is used in adaptation process to minimize the difference between the statistics of y_n and some pre-calculated statistics (R_R and R_I) of the transmitted QAM signal:

$$\begin{aligned}\psi_G[y_n] = &(g_1(y_{R,n}, y_{I,n}) - R_R) \cdot g_2(y_{R,n}, y_{I,n}) \\ &+ \jmath\ (g_3(y_{R,n}, y_{I,n}) - R_I) \cdot g_4(y_{R,n}, y_{I,n}).\end{aligned} \tag{6}$$

where $g_1(\cdot)$, $g_2(\cdot)$, $g_3(\cdot)$ and $g_4(\cdot)$ are zero-memory (and preferably continuous) functions. The resulting weight adaptation process is $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \cdot \psi_G[y_n]\mathbf{x}_n^*$. R_R and R_I are positive (dispersion constants) and are computed so that:

$$E[\psi_G[a_n]a_n^*] = 0. \tag{7}$$

The error functions of almost all existing stochastic gradient based blind equalization algorithms can be mapped onto (6).

Now we want to design an error function $\psi_L[y_n]$ that incorporates $\hat{a}_n$ (the local information). There are many ways to do it. One way is to allow dual mode; that is based on the error level between y_n and $\hat{a}_n$, the equalizer will switch between certain blind and decision-directed adaptations, as reported in [2]. So the algorithm switches between a suitable $\psi_G[y_n]$ and $\psi_L[y_n] = y_n - \hat{a}_n$. Another method exploits $\hat{a}_n$ directly in the weight adaptation process without going into decision-directed mode [3]. It embeds $\hat{a}_n$ and dispersion constants (R) together, such that $\psi_G[y_n]$ becomes function of both dispersion constant(s) and $\hat{a}_n$.

The third approach, which is the subject of this paper, is "stop-and-go" scheme. In this approach, the adaptation process apparently uses *only* $\hat{a}_n$ and y_n from startup to final convergence. It is achieved by replacing the dispersion constants R_R and R_I with some suitable nonlinear functions of $\hat{a}_n$. For example, replacing R_R and R_I with $g_1(|\hat{a}_{R,n}|, |\hat{a}_{I,n}|)$ and $g_3(|\hat{a}_{R,n}|, |\hat{a}_{I,n}|)$ in (6) will incorporate local information in weight adaptation process. Note that, it will result in an increase in the number of contours and as a result, the QAM symbols in alphabet $\mathcal{A}$ lie on (at least) one or more contours. The resulting error function (say local) $\psi_L[y_n]$ can be given as

$$\begin{aligned}\psi_L[y_n] = &(g_1(y_{R,n}, y_{I,n}) - g_1(|\hat{a}_{R,n}|, |\hat{a}_{I,n}|)) \cdot g_2(y_{R,n}, y_{I,n}) \\ &+ \jmath\ (g_3(y_{R,n}, y_{I,n}) - g_3(|\hat{a}_{R,n}|, |\hat{a}_{I,n}|)) \cdot g_4(y_{R,n}, y_{I,n})\end{aligned} \tag{8}$$

Note that $\psi_L[y_n]$ forces y_n to lie on the contour which also contains the closest symbol $\hat{a}_n = \mathcal{D}(y_n)$ on it. It can easily be understood that due to multiple contours exhibited by $\psi_L[y_n]$, the steady-state misadjustment offered by $\psi_L[y_n]$ is much lower compared to that of $\psi_G[y_n]$. If $\psi_G[y_n]$ is capable of removing ISI, then the use of $\psi_L[y_n]$ will be beneficial, only when the sign of the two error functions, $\psi_G[y_n]$ and $\psi_L[y_n]$, match. If $\psi_L[y_n]$ is incorporated in the weight adaptation process then the real and the imaginary parts of $\psi_L[y_n]$ should be weighted with binary flags f_R and f_I, respectively, to indicate the sign match. Flags, f_R

and f_I, are obtained as follows:

$$f_R = \frac{1 + \mathrm{sgn}[\psi_L[y_n]_R] \cdot \mathrm{sgn}[\psi_G[y_n]_R]}{2} \quad (9)$$

$$f_I = \frac{1 + \mathrm{sgn}[\psi_L[y_n]_I] \cdot \mathrm{sgn}[\psi_G[y_n]_I]}{2} \quad (10)$$

The resulting weight adaptation rule is,

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \cdot (\psi_L[y_n]_R \cdot f_R + \jmath\ \psi_L[y_n]_I \cdot f_I)\mathbf{x}_n^*. \quad (11)$$

where subscripts $_R$ and $_I$ denote the real and imaginary components, respectively. Since $\psi_L[y_n]$ is being used in weight adaptation process, the values of R_R and R_I need not to satisfy (7); instead, they can be selected such that the sign of the two error functions agree most of the time. It is observed that if R_R and R_I are selected as the outermost contour, then the reliable regions can be maximized.

$$R_R = \max[g_1(\{|a_{R,n}|\}, \{|a_{I,n}|\})] \quad (12)$$

$$R_I = \max[g_3(\{|a_{R,n}|\}, \{|a_{I,n}|\})] \quad (13)$$

The above mentioned scheme (Equations (6), (8), (9), (10), (11), (12) and (13)) can be applied to any stochastic gradient descent based blind equalization scheme to develop its stop-and-go version. In subsequent sections, we will review two existing stop-and-go blind equalization algorithms based on this framework and three new stop-and-go algorithms will be presented.

III. POINT-CONTOUR STOP-AND-GO ALGORITHM

The first blind equalizer for multilevel PAM signals was introduced by Sato [4]. In essence, it is identical to the decision-directed algorithm when the PAM input is binary (± 1). For M-level PAM signals, it is defined by the error function

$$\psi_G[y_n] = y_n - R_1 \mathrm{sgn}[y_n], \text{ where } R_1 = \frac{E[a_n^2]}{E[|a_n|]}. \quad (14)$$

The Sato algorithm was extended to complex signals (QAM) by Benveniste et al. [5] by separating signals into their real and imaginary parts as

$$\psi_G[y_n] = y_{n,R} - R_R\ \mathrm{sgn}[y_{n,R}] + \jmath(y_{n,I} - R_I\ \mathrm{sgn}[y_{n,I}]) \quad (15)$$

where R_R and R_I are computed as $E[a_{n,R}^2]/E[|a_{n,R}|]$ and $E[a_{n,I}^2]/E[|a_{n,I}|]$, respectively. The resulting weight adaptation algorithm is called *reduced constellation algorithm* (RCA), as it attempts to resolve the output of the channel to belong to one of the four statistical symbols of a reduced constellation. Those four points are (R_R, R_I), $(R_R, -R_I)$, $(-R_R, -R_I)$ and $(-R_R, R_I)$.

Picchi and Prati [6] developed the first ever *stop-and-go* algorithm for blind equalization. They observed that a simple decision-directed adaptation can open a closed eye, provided *the adaptation is stopped for the small proportion of the instances when the decision-directed and the RCA errors have different sign*. The weight adaptation rule is given as $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu\,[\psi_L[y_n]_R \cdot f_R + \jmath\ \psi_L[y_n]_I \cdot f_I]\,\mathbf{x}_n^*$, where $\psi_L[y_n]$ is computed as decision-directed error:

$$\psi_L[y_n] = y_{R,n} - \hat{a}_{R,n} + \jmath\ (y_{I,n} - \hat{a}_{I,n}) \quad (16)$$

Note that (16) can be expressed in an equivalent form as follows:

$$\psi_L[y_n] = y_{R,n} - |\hat{a}_{R,n}|\ \mathrm{sgn}[y_{R,n}] + \jmath(y_{I,n} - |\hat{a}_{I,n}|\mathrm{sgn}[y_{I,n}]) \quad (17)$$

Comparing (15) and (17); it can be observed that Picchi and

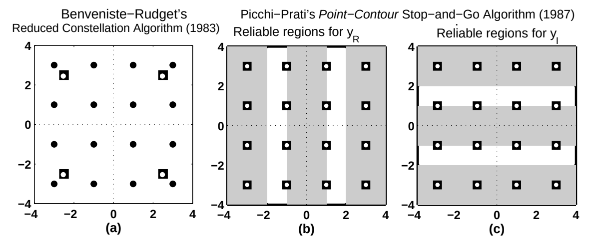

Fig. 2. Contours in (a) RCA (b)-(c) PC-SAGA

Prati algorithm is a stop-and-go version of RCA obtained by replacing R_R and R_I with $|\hat{a}_{R,n}|$ and $|\hat{a}_{I,n}|$, respectively. Similar to RCA, which forces y_n to belong to one of the four possible *point contours*, the Picchi and Prati algorithm forces y_n to belong to the nearest constellation symbol. It can be said that it modifies the RCA's *4-point contours* into an *M-point contours*. Picchi and Prati algorithm is thus named *Point-Contour Stop-and-Go Algorithm (PC-SAGA)*. Based on the proposed framework in (12) or (13), we compute $R_R = R_I = \beta$ for QAM as

$$\begin{aligned} \beta &= \max[\{|a_R|\}] = \max[\{|a_I|\}] \\ &= \begin{cases} \sqrt{M} - 1, & \text{for square QAM} \\ \frac{3}{2}\sqrt{\frac{M}{2}} - 1, & \text{for cross QAM.} \end{cases} \end{aligned} \quad (18)$$

IV. DIAMOND-CONTOUR STOP-AND-GO ALGORITHM

The first complexity-efficient (signed-error) blind equalizer was proposed by Weerackody et al. [7]. The Weerackody-Kassam hard-limited algorithm (WK-HLA) is specified by $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu\psi_G[y_n]\mathbf{x}_n^*$, where $\psi_G[y_n]$ is computed as:

$$\psi_G[y_n] = \mathrm{sgn}[|y_{R,n}| + |y_{I,n}| - R] \cdot (\mathrm{sgn}[y_{R,n}] + \jmath\ \mathrm{sgn}[y_{I,n}]) \quad (19)$$

This algorithm attempts to drive the equalizer output to reside on a 45° rotated-square (diamond) contour as depicted in Fig. 3(a). The final equalizer output is obtained by the removal of this rotation. It has been reported in [7] that it performs better than CMA (if properly initialized). Kim et al. [8] observed that Weerackody-Kassam algorithm can be transformed into a stop-and-go algorithm. The $\psi_L[y_n]$ proposed in Kim's *diamond-contour* stop-and-go algorithm is as follows:

$$\begin{aligned} \psi_L[y_n] = &\ \mathrm{sgn}[|y_{R,n}| + |y_{I,n}| - |\hat{a}_{R,n}| - |\hat{a}_{I,n}|] \\ &\cdot\ (\mathrm{sgn}[y_{R,n}] + \jmath\ \mathrm{sgn}[y_{I,n}]) \end{aligned} \quad (20)$$

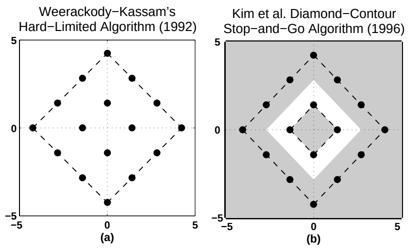

Fig. 3. Contours in (a) WK-HLA (b) Kim's DC-SAGA.

Due to 45° rotation, the sliced symbol $\hat{a}_n = \hat{a}_{R,n} + \jmath\, \hat{a}_{I,n}$ is obtained differently as $\hat{a}_n = \mathcal{D}[y_n \cdot e^{-\jmath\pi/4}] \cdot e^{\jmath\pi/4}$. The single flag f is obtained by comparing the sign of $[|y_{R,n}| + |y_{I,n}| - |\hat{a}_{R,n}| - |\hat{a}_{I,n}|]$ and $[|y_{R,n}| + |y_{I,n}| - R]$. It should be noted that no closed form expression for the computation of R has been reported in [8]. However, based on the proposed framework, the value of R (that maximizes the reliable regions) can be obtained as:

$$R = \sqrt{2} \cdot \max[\{|a_R|\}, \{|a_I|\}] = \sqrt{2} \cdot \beta \tag{21}$$

where β is obtained from (18). Fig. 3(b) depicts multiple diamond contours generated by Kim's stop-and-go algorithm. Kim's algorithm is thus named *Diamond-Contour Stop-and-Go Algorithm (DC-SAGA)*.

V. CIRCULAR-CONTOUR STOP-AND-GO ALGORITHM

The Godard algorithm [9] is one of the best known blind equalization algorithms and is a stochastic gradient algorithm for the cost function, $J_p = \frac{1}{2p} E(|y(n)|^p - R)^2$, where $p \in \{1, 2, \cdots\}$ and E denotes statistical expectation. The corresponding algorithm is $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu y_n |y_n|^{p-2}(|y_n|^2 - R)\mathbf{x}_n^*$. The constant modulus algorithm (CMA) proposed in [10] is a special case of Godard algorithm for $p = 2$. The corresponding algorithm is $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \psi_G[y_n] \mathbf{x}_n^*$, where $\psi_G[y_n]$ is given as

$$\psi_G[y_n] = y_n(|y_n|^2 - R) \tag{22}$$

and $R = E[|a(n)|^4]/E[|a(n)|^2]$. CMA forces the equalizer output y_n to reside on a circular contour as depicted in Fig. 4(a).

A decision-directed type CMA was proposed in [11], called *radius-directed equalization* (RDE), for QAM signals based on the known modulus (circular contour) of the constellation symbol radii. For example, 16 QAM has three radii ($\sqrt{2}$, $\sqrt{10}$ and $\sqrt{18}$); and 32 QAM has five radii ($\sqrt{2}$, $\sqrt{10}$, $\sqrt{18}$, $\sqrt{26}$ and $\sqrt{34}$). The algorithm uses the error between the equalizer output modulus and the nearest symbol radius to update the equalizer weights, as depicted in Fig. 4(b). It provides faster convergence than the CMA. However, the convergence of RDE is not guaranteed, as it operates totally in decision-directed mode. RDE uses error function $\psi_L[y_n]$ which is given as,

$$\psi_L[y_n] = y_n(|y_n|^2 - R_k) \tag{23}$$

where R_k is the square of the radii of the nearest constellation symbol for each equalizer output. To improve the RDE's convergence, different techniques have been suggested. In [12], the RDE is modified by incorporating stop-and-go flags such that adaptation takes place only when the sign of $\psi_G[y_n] = (|y_n|^2 - R)$ and $\psi_L[y_n] = (|y_n|^2 - \{\mathcal{D}[y_n]\}^2)$ match. It was named Stop-and-Go Decision-Directed Multiple-modulus Algorithm (SAG-DDMMA). The resulting reliable regions are depicted in Fig. 4(c).

In this paper, a straightforward "stop-and-go" version of CMA, similar to one given in [12], is proposed. The proposed *Circular-Contour Stop-and-Go Algorithm (CC-SAGA)* is based on the framework described in Section II, and is given as $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \cdot \psi_L[y_n] \cdot f \cdot \mathbf{x}_n^*$, where f is 1 when the sign of $\psi_G[y_n]$ (22) and $\psi_L[y_n]$ (23) match, and 0 otherwise. The value of R in $\psi_G[y_n]$, is computed as

$$\begin{aligned} R &= \max[\{R_k\}] = \max[\{|a_R^2| + |a_I^2|\}] \\ &= \begin{cases} 2 \cdot \beta^2, & \text{square QAM} \\ \left(\frac{2}{3}\beta - \frac{1}{3}\right)^2 + \beta^2, & \text{cross QAM} \end{cases} \end{aligned} \tag{24}$$

where β is computed form (18). The proposed algorithm differ from [12] in the formulation of reliable regions; otherwise, the weight adaptation process is similar. The reliable regions formed by the proposed algorithm are depicted in Fig. 4(d).

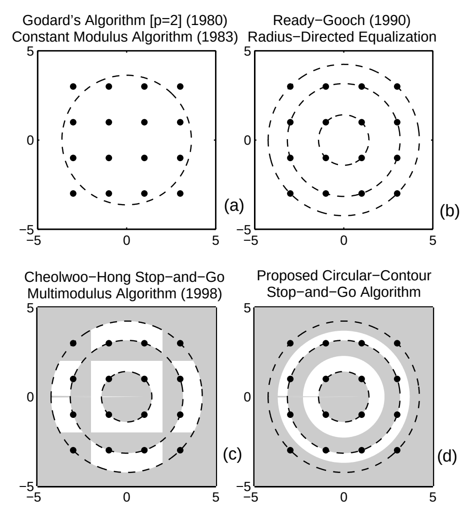

Fig. 4. (a) CMA, (b) RDE, (c) SAG-DDMMA, (d) Proposed CC-SAGA.

VI. LINE-CONTOUR STOP-AND-GO ALGORITHM

The constant modulus algorithm (CMA) is blind to carrier phase offset error. The CMA has been modified to incorporate phase information, it resulted in multimodulus algorithm (MMA). The multimodulus algorithm was proposed independently by many authors [13], [14], [15]. This algorithm minimizes the dispersion of real and imaginary parts, $y_{R,n}$ and $y_{I,n}$, and forces them to lie on straight-line contours. MMA error function $\psi_G[y_n]$ is given as $\psi_G[y_n] =$

$y_{R,n}(y_{R,n}^2 - R_R^2) + \jmath\, y_{I,n}(y_{I,n}^2 - R_I^2)$. Recently, a complexity efficient MMA is proposed, named soft-constraint satisfaction multimodulus algorithm (SCS-MMA) [16], [17]. The SCS-MMA error function $\psi_G[y_n]$ is expressed as

$$\psi_G[y_n] = y_{R,n}(|y_{R,n}| - R_R) + \jmath\, y_{I,n}(|y_{I,n}| - R_I) \quad (25)$$

where $R_R = E[|a_R|^3]/E[a_R^2]$ and $R_I = E[|a_I|^3]/E[a_I^2]$. The (straight) line-contours exhibited by SCS-MMA are shown in Fig. 5(a). In order to obtain a *stop-and-go* version of SCS-MMA, the dispersion constants R_R and R_I are replaced with $|\hat{a}_{R,n}|$ and $|\hat{a}_{I,n}|$, respectively, to get a new error function $\psi_L[y_n]$ as follows

$$\psi_L[y_n] = y_{R,n}(|y_{R,n}| - |\hat{a}_{R,n}|) + \jmath\, y_{I,n}(|y_{I,n}| - |\hat{a}_{I,n}|) \quad (26)$$

The proposed *Line-Contour Stop-and-Go Algorithm (LC-SAGA)* can be expressed as $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu\,[\psi_L[y_n]_R \cdot f_R + \jmath\, \psi_L[y_n]_I \cdot f_I]\, \mathbf{x}_n^*$, where f_R and f_I are binary-flags which indicate the match of signs of the real and the imaginary parts of $\psi_G[y_n]$ and $\psi_L[y_n]$, respectively. $\psi_G[y_n]$ and $\psi_L[y_n]$ are computed from (25) and (26), respectively. For the proposed algorithm for QAM signals, R_R and R_I are computed as the outermost line contour as follows:

$$R_R = R_I = \max[\{|a_R|\}] = \max[\{|a_I|\}] = \beta \quad (27)$$

where β is computed form (18). The reliable regions obtained by the proposed algorithm are depicted in Figure 5(b) for 16-QAM.

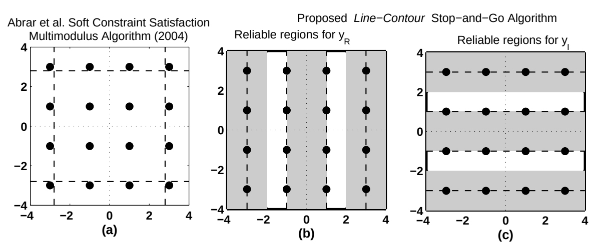

Fig. 5. Contours in (a) SCS-MMA and (b)-(c) Proposed LC-SAGA.

VII. SQUARE-CONTOUR STOP-AND-GO ALGORITHM

Recently, Thaiupathump and Kassam presented an interesting family of generalized square-contour algorithms (TW-GSCA) [18]. Unlike WK-HLA (as described in Section IV), these algorithms are free from unnecessary 45° rotation. The weight adaptation process is $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu\psi_G[y_n]\mathbf{x}_n^*$, where $\psi_G[y_n]$ is computed as,

$$\begin{aligned}\psi_G[y_n] = {} & ((|y_{R,n} + y_{I,n}| + |y_{R,n} - y_{I,n}|)^p - R^p) \\ & \cdot\, (|y_{R,n} + y_{I,n}| + |y_{R,n} - y_{I,n}|)^{p-1} \\ & \cdot\, \{\mathrm{sgn}[y_{R,n} + y_{I,n}] + \mathrm{sgn}[y_{R,n} - y_{I,n}] \\ & + \jmath\, (\mathrm{sgn}[y_{R,n} + y_{I,n}] - \mathrm{sgn}[y_{R,n} - y_{I,n}])\} \quad (28)\end{aligned}$$

(Refer to [18] for the computation of R^p.) The beauty of this algorithm is that it forces the equalizer outputs to reside on a square-contour; which is a desirable contour for square QAM constellations. Figure 6(a) depicts the square-contour generated by TW-GSCA for 16-QAM constellation.

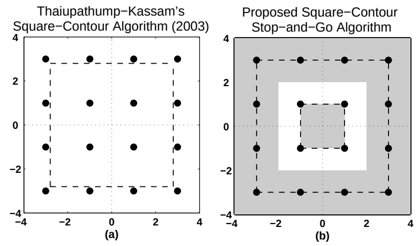

Fig. 6. Contours in (a) TW-GSCA (b) Proposed SC-SAGA.

In order to obtain a *stop-and-go* version of TW-GSCA, the dispersion constant R^p in TW-GSCA are replaced with $(|\hat{a}_{R,n} + \hat{a}_{I,n}| + |\hat{a}_{R,n} - \hat{a}_{I,n}|)^p$ to get $\psi_L[y_n]$ as follows

$$\begin{aligned}\psi_L[y_n] = {} & \{(|y_{R,n} + y_{I,n}| + |y_{R,n} - y_{I,n}|)^p \\ & -(|\hat{a}_{R,n} + \hat{a}_{I,n}| + |\hat{a}_{R,n} - \hat{a}_{I,n}|)^p\} \\ & \cdot\, (|y_{R,n} + y_{I,n}| + |y_{R,n} - y_{I,n}|)^{p-1} \\ & \cdot\, \{\mathrm{sgn}[y_{R,n} + y_{I,n}] + \mathrm{sgn}[y_{R,n} - y_{I,n}] \\ & + \jmath\, (\mathrm{sgn}[y_{R,n} + y_{I,n}] - \mathrm{sgn}[y_{R,n} - y_{I,n}])\} \quad (29)\end{aligned}$$

The proposed *Square-Contour Stop-and-Go Algorithm (SC-SAGA)* can be expressed as $\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \cdot f \cdot \psi_L[y_n]\mathbf{x}_n^*$, where f is the binary-flag which indicates the match of signs of $\psi_G[y_n]$ and $\psi_L[y_n]$. For the proposed algorithm, the dispersion constant R is computed as the outermost square contour as follows:

$$R = 2 \cdot \max[\{|a_R|\}, \{|a_I|\}] = 2 \cdot \beta \quad (30)$$

where β is computed form (18). The reliable regions obtained by the proposed algorithm are depicted in Figure 6(b) for 16-QAM.

VIII. SIMULATION RESULTS

In this section, the performance of existing and proposed stop-and-go algorithms are compared. In simulations, a complex-valued seven taps transversal equalizer was used and it was initialized so that the center tap was set to one and other taps were set to zero. The channel used in the simulation was taken from [6]. The signal to noise ratio (SNR) was taken as 30dB at the input of the equalizer. The residual ISI [19] and MSE are measured for 16-QAM signaling and compared as performance parameters. Each trace is the ensemble average of 200 independent runs with random initialization of noise and data source.

Fig. 7 depicts traces of residual ISI convergence. Examining this result, we observe that PC-SAGA and DC-SAGA are relatively the fastest and the slowest converging algorithms, respectively. The SC-SAGA is next to the PC-SAGA in performance. Fig. 8 depicts traces of MSE convergence, where we observe that PC-SAGA is consistently performing the best and SC-SAGA is next to it. While the CC-SAGA is performing worst by giving nonconverging MSE floor because of its incapability to remove (almost 45°) phase-offset error (introduced by the channel).

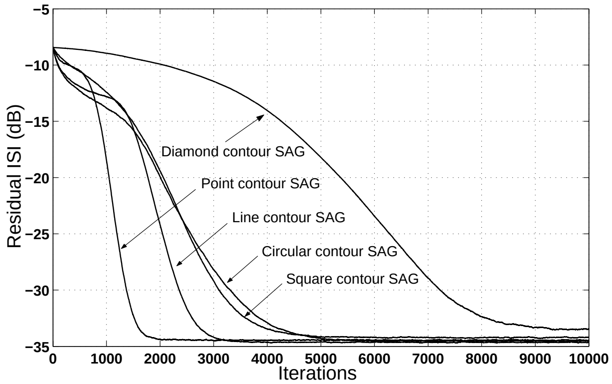

Fig. 7. Convergence traces for SER.

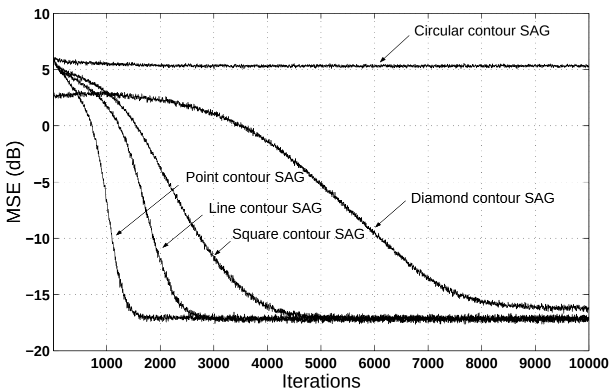

Fig. 8. Convergence traces for MSE.

IX. CONCLUSION

In this work, a framework for stop-and-go based blind equalization algorithms is presented. Based on the proposed framework, three new stop-and-go algorithms are proposed. Using computer simulations, it is shown that the existing (point-contour) "Stop-and-Go Decision-Directed Algorithm" [6] is the best among other stop-and-go algorithms. A detailed convergence analysis of these algorithms is under study.

REFERENCES

[1] Zhi Ding and Ye (Geofferey) Li. *"Blind Equalization and Identification"*. Marcel Dekker, Inc., New York, 2001.

[2] V. Weerackody and S.A. Kassam. "Dual-Mode Type Algorithms for Blind Equalization". *IEEE Trans Commun.*, 42(1):22–28, Jan. 1994.

[3] S. Abrar. "Compact Constellation Algorithm for Blind Equalization of QAM Signals". *IEEE Intl Networking and Communications Conf.*, Lahore, Pakistan, pages 170–174, June, 11-13 2004.

[4] Y. Sato. "A Method of Self-Recovering Equalization for Multilevel Amplitude Modulation Systems". *IEEE Trans. Commun.*, COM-23:679–682, June 1975.

[5] A. Benveniste and M. Goursat. "Blind Equalizers". *IEEE Trans. Commun.*, COM-32:871–883, 1984.

[6] G. Picchi and G. Prati. "Blind Equalization and Carrier Recovery using a 'Stop-and-Go' Decision-Directed Algorithm". *IEEE Trans. Commun.*, COM-35:877–887, 1987.

[7] V. Weerackody and S.A. Kassam. A Simple Hard-Limited Adaptive Algorithm for Blind Equalization. *IEEE Trans. Circuit and Systems-II*, 39(7):482–487, July 1992.

[8] Youngkyun Kim, Sungjo Kim, and Mintaig Kim. "The Derivation of a New Blind Equalization Algorithm". *ETRI Journal*, 18(2):53–60, July 1996.

[9] D.N. Godard. "Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communications Systems". *IEEE Trans. Commun.*, COM-28:1867–1875, Nov. 1980.

[10] J.R. Treichler and B.G. Agee. "A New Approach to Multipath Correction of Constant Modulus Signals". *IEEE Trans. acoust. Speech Signal Process.*, ASSP-31:459–471, 1983.

[11] M. J. Ready and R. P. Gooch. "Blind Equalization Based on Radius Directed Adaptation". *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 1699–1702, April 1990.

[12] C.W. You and D.S. Hong. "Nonlinear Blind Equalization Schemes Using Complex-Valued Multilayer Feedforward Neural Networks". *IEEE Trans. Neural Network*, 9(6):1442–1455, Nov. 1998.

[13] K. Wesolowsky. "Self-Recovering Adaptive Equalization Algorithms for Digital Radio and Voiceband Data Modems". *Proc. European Conf. Circuit Theory and Design*, pages 19–24, 1987.

[14] K.N. Oh and Y.O. Chin. "Modified Constant Modulus Algorithm: Blind Equalization and Carrier Phase Recovery Algorithm". *Proc. 1995 IEEE Int. Conf. Commun.*, 1:498–502, June 1995.

[15] J. Yang, J.-J. Werner, and G.A. Dumont. "The Multimodulus Blind Equalization Algorithm". *IEEE Intl. Conf. on DSP*, 1:127–130, 1997.

[16] S. Abrar, A. Zerguine, and M. Deriche. "Soft Constraint Satisfaction Multimodulus Blind Equalization Algorithms". *IEEE Intl. Conf. Acoustics, Speech, & Signal Processing (ICASSP 2004) Montreal, Quebec, Canada.*, II:853–856, May 17-21 2004.

[17] S. Abrar and A. Zerguine. "A New Multimodulus Blind Equalization Algorithm". *IEEE Intl. Networking & Commun. Conf. (INCC 2004) LUMS, Lahore, Pakistan.*, pages 165–169, June 11-13 2004.

[18] T. Thaiupathump and S.A. Kassam. "Square Contour Algorithm: A New Algorithm for Blind Equalization and Carrier Phase Recovery". *IEEE GLOBECOM*, pages 647–651, Dec. 2003.

[19] O. Shalvi and E. Weinstein. "New Criteria for Blind Equalization of Non-minimum Phase Systems". *IEEE Trans. Inf. Theory*, vol. 36:312–321, 1990.