Adaptive Channel Estimation in Decode and Forward Relay Networks

Qingxiong Deng, Raquel G. Machado, and Andrew G. Klein
Department of Electrical & Computer Engineering
Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609
Email: {qxdeng, raquel, klein}@wpi.edu

***Abstract*—Channel state information is vital for exploiting diversity in cooperative networks. The existing literature on cooperative channel estimation assumes block lengths are long and that channel estimation takes place within a fading block. To reduce the forwarding delay, short block lengths are preferred and adaptive estimation through multiple blocks is required. In this paper, we consider estimating the relay-to-destination channel in decode-and-forward relay systems for which the presence of training data is probabilistic since it is unknown whether the relay participates in the forwarding round. A detector is used before the update of the least mean square channel estimate so that an update is made only when the detector decides the presence of training data. We use the generalized likelihood ratio test and focus on the detector threshold for deciding whether the training sequence is present. To improve the convergence speed and reduce the estimation error, a heuristic objective function is proposed which leads to an "optimal" threshold. Extensive numerical results show the superior performance of using this threshold as opposed to fixed thresholds.**

***Index Terms*—channel estimation, decode-and-forward relay, LMS algorithm.**

I. Introduction

Cooperative relay networks have emerged as a powerful technique to combat multipath fading and increase energy efficiency [1]. To reap the benefits of cooperative diversity, accurate channel state information is required at the transmitter or receiver, or both. Much of the current literature assumes perfect channel information; however, channel estimation in relay systems is a practical challenge that must be addressed. Some pioneering works on relay channel estimation are [2], [3] for amplify-and-forward (AF) relay channels and [4] for decode-and-forward (DF) relay channels. These works assume quasi-static Rayleigh flat fading channels, which means that the channel remains constant within one block but varies independently from block to block. While the block fading assumption renders nice analytical results, it may not be applicable to practical situations. Under this assumption, the training in each block is assumed long enough for channel estimation. Since the training data usually comprises a relatively small proportion of the whole block, the total duration of one block could be very long. As each block is processed individually at the end of the transmission for relaying, large delay results. To reduce the delay and computation, short block lengths are preferred and the training data should be spread across each block. In this situation, training data is periodic for AF channel estimation [5]. However, the DF relay only forwards the signal it receives from the source when it can correctly decode the message (verified, for example, by using cyclic redundancy check). As such, the DF relay switches between forwarding and silence, so the presence of training data in each block for the relay-to-destination channel is therefore probabilistic.

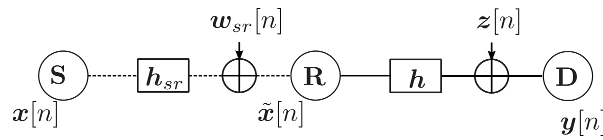

Fig. 1. System model.

In this work, we consider adaptive channel estimation in decode-and-forward (DF) relay channels when the amount of training inside one block is not sufficient for the required estimation accuracy even when the channel remains the same from block to block. In particular, we are interested in estimation of the relay-to-destination channel since the probabilistic nature of the training data poses a challenge in estimating the channel for this link. For its simplicity and robustness, we consider the least mean square (LMS) algorithm [6] for channel estimation. Due to the probabilistic presence of training data for the relay-to-destination channel, we explore the possibility of combining detection and adaptation of the LMS algorithm to achieve a better convergence rate if larger computational cost is allowed. As analysis of the classical LMS algorithm assumes the knowledge of the training data, the performance of the LMS algorithm is unknown when the presence of training data is probabilistic. We aim to determine if the adaptive algorithm combining detection and LMS algorithm converges and what the average convergence rate is. We also study the impact of the detector threshold on convergence speed and estimation error.

II. System Model

We investigate a system as shown in Fig. 1, which consists of a source node ($\mathbf{S}$), a relay node ($\mathbf{R}$), and a destination node ($\mathbf{D}$). We consider pilot-assisted estimation of the relay-to-destination channel $\boldsymbol{h}$, which is assumed to be static from block to block. The relay operates in half-duplex mode, and thus does not transmit and receive at the same time. In addition, the relay node and the source use the same transmission bandwidth but employ time division so that the relay transmits on a channel orthogonal to the source. We assume that the transmitted symbol block from the source is $\boldsymbol{x}[n] \in \mathbb{C}^{N\times 1}$ where n is the block index. To have independent receiving blocks at the destination, we assume transmission takes place with guard intervals. We do not put any assumption on the source-to-relay channel $\boldsymbol{h}_{sr}$ and the noise at the relay $\boldsymbol{w}_{sr}$ but assume the probability that the relay can correctly decode the message from the source is P. Without loss of generality, we assume that the length of $\boldsymbol{h}$ is larger than the length of $\boldsymbol{h}_{sr}$ and that the transmitter knows the length of the channel L_h and becomes silent for $L_h - 1$ symbol duration after it transmits $\boldsymbol{x}[n]$. Hence the input to the relay-to-destination channel $\tilde{\boldsymbol{x}}[n]$ is probabilistic as $\Pr(\tilde{\boldsymbol{x}}[n] = \boldsymbol{x}[n]) = P$ and $\Pr(\tilde{\boldsymbol{x}}[n] = \mathbf{0}) = 1 - P$. The corresponding output $\boldsymbol{y}[n]$ is also probabilistic as

$$\Pr(\boldsymbol{y}[n] = \boldsymbol{X}[n]\boldsymbol{h} + \boldsymbol{z}[n]) = P$$

and

$$\Pr(\boldsymbol{y}[n] = \boldsymbol{z}[n]) = 1 - P$$

where $\boldsymbol{z}[n] \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_{N+L_h-1})$ is i.i.d. noise, and $\boldsymbol{X}[n] \in \mathbb{C}^{(N+L_h-1)\times L_h}$ is a tall Tœplitz matrix with $X_{i,j}[n] = x_{i-j}[n]$. The transmission of a complete message is divided into two phases:

1) In phase one, the source broadcasts the message to the destination and the relay. The relay attempts to decode the message.
2) In phase two, the source is silent. If the relay can successfully decoded the message, it forwards the message to the destination. Otherwise, it remains silent.

The source and relay then alternate between these two phases. As shown in Fig. 2, the relay does not participate in the second phase of the second block, and the destination receives the composite signal corrupted by intersymbol interference and additive noise, but not by the interblock interference.

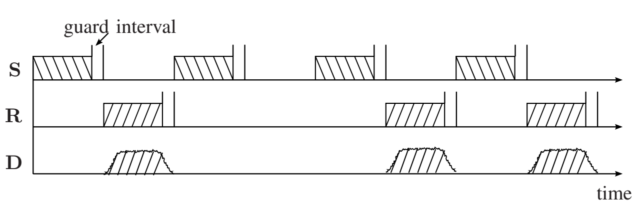

Fig. 2. Transmission process.

III. Combining detection and adaptation

In this section, we propose an adaptive algorithm to estimate the relay-to-destination channel. The algorithm is based on the LMS algorithm and combines detection and adaptation so that the adaptation happens only when the detector determines there is training data in the received signal. We develop the condition of convergence in the mean square. We emphasize that the updating proceeds in a block-by-block fashion, as the blocks are made mutually independent by transmission with guard intervals, which is useful for detection of the presence of training and is sometimes referred to as the "Periodic LMS algorithm" [7]. An alternative could be updating channel estimate in a symbol-by-symbol fashion. However, due to the guard-interval, the input process is not stationary but cyclostationary, and subsequently the convergence analysis for symbol-by-symbol updating LMS algorithm is complicated and more onerous [8].

Because of the probabilistic transmission, the receiver needs to decide whether it should use the observation to update the channel estimate. Mathematically, the detector is used to make a decision between the following two hypotheses

$$\begin{array}{lcl} \mathcal{H}_0 & : & \boldsymbol{y}[n] = \boldsymbol{z}[n] \\ \mathcal{H}_1 & : & \boldsymbol{y}[n] = \boldsymbol{X}[n]\boldsymbol{h} + \boldsymbol{z}[n]. \end{array} \tag{1}$$

As shown in Fig. 3, the observation used for updating is

$$\hat{\boldsymbol{y}}[n] = \begin{cases} \boldsymbol{y}[n] & \text{if } \mathcal{H}_1 \text{ is decided} \\ \mathbf{0} & \text{otherwise,} \end{cases}$$

and the input to the channel estimate $\hat{\boldsymbol{h}}[n]$ at time n is

$$\hat{\boldsymbol{x}}[n] = \begin{cases} \boldsymbol{x}[n] & \text{if } \mathcal{H}_1 \text{ is decided} \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

The LMS update equation is

$$\hat{\boldsymbol{h}}[n+1] = \hat{\boldsymbol{h}}[n] + \mu \hat{\boldsymbol{X}}^H[n](\hat{\boldsymbol{y}}[n] - \boldsymbol{d}[n]) \tag{2}$$

where μ is the stepsize, $\boldsymbol{d}[n] = \hat{\boldsymbol{X}}[n]\hat{\boldsymbol{h}}[n]$ is the filtered output. The estimation error is defined as $\boldsymbol{e}[n] = \hat{\boldsymbol{y}}[n] - \boldsymbol{d}[n]$. If $\mathcal{H}_0$ is decided, the channel estimate remains unchanged since adaptation does not take place.

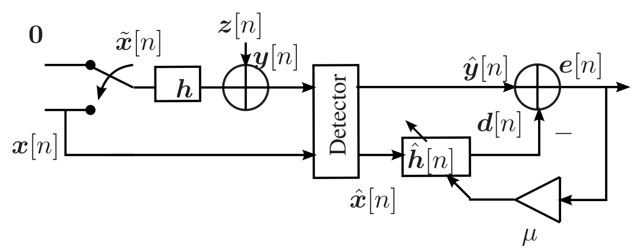

Fig. 3. Block diagram of LMS-based adaptive algorithm .

The hypothesis testing problem defined in (1) is a composite hypothesis testing problem. Before we proceed with specific the detector, we use an abstract detector to analyze the algorithm. We defined an abstract detector with $\Pr(\mathcal{H}_i|\mathcal{H}_j)$ as the probability of deciding $\mathcal{H}_i$ when $\mathcal{H}_j$ is true. Hence the probability of detection is $P_D = \Pr(\mathcal{H}_1|\mathcal{H}_1)$ and the probability of false alarm is $P_{FA} = \Pr(\mathcal{H}_1|\mathcal{H}_0)$. Thus the probability that the detector decides $\mathcal{H}_1$ is

$$P_u = PP_D + (1-P)P_{FA}. \tag{3}$$

Then we have $\Pr(\hat{\boldsymbol{x}}[n] = \boldsymbol{x}[n]) = P_u$, and $\Pr(\hat{\boldsymbol{x}}[n] = \mathbf{0}) = 1 - P_u$.

We focus on the update behavior of the algorithm and define the mean squared error (MSE) as

$$\begin{array}{rcl} J[n] & = & E[\|\boldsymbol{e}[n]\|^2] = E[\|\hat{\boldsymbol{y}}[n] - \boldsymbol{d}[n]\|^2] \\ & = & E[\|\boldsymbol{X}[n]\boldsymbol{h} + \boldsymbol{z}[n] - \boldsymbol{X}[n]\hat{\boldsymbol{h}}[n]\|^2]P_s \\ & + & E[\|\boldsymbol{z}[n] - \boldsymbol{X}[n]\hat{\boldsymbol{h}}[n]\|^2](1-P_s) \end{array} \tag{4}$$

where

$$P_s = \frac{PP_D}{P_u} \tag{5}$$

is the probability that the receiver correctly uses the received signal when updating the channel estimate, and the first item in (4) is the mean square error when a detection occurs and the second item in (4) is the mean square error when a false alarm occurs. Taking the derivative of $J[n]$ with respect to $\hat{\boldsymbol{h}}[n]$ and setting it to zero, the Weiner solution $\boldsymbol{h}_o$ which minimizes the mean squared error $J[n]$ is given by

$$\boldsymbol{h}_o = P_s\boldsymbol{h}.$$

It should be noted that the Weiner solution is a scaled version of the true channel. The minimum mean squared error is

$$J_{\min} = (N + L_h - 1)\sigma^2 + P_s(1 - P_s)\boldsymbol{h}^H\boldsymbol{K}_x\boldsymbol{h}$$

where $\boldsymbol{K}_x = E[\boldsymbol{X}^H[n]\boldsymbol{X}[n]]$ is the autocorrelation matrix of the Tœplitz matrix of the training data. We note that the minimum mean squared error $J_{\min}$ achieves its minimum at both $P_s = 0$ and $P_s = 1$. This situation arises from the nature of this channel estimation problem. Because of the probabilistic transmission of the training data, the observation at the receiver may be just noise, from which the channel estimate is $\mathbf{0}$. Naturally we want P_s as large as possible, so a proper alternative formulation could be to the minimize J with the constraint $P_s > 1/2$. The reason for requiring $P_s > 1/2$ is that we want the LMS channel estimate to be updated more by the observations with training data than by the noise. As P_s is only related to the detector, we leave this problem for the next section.

It is interesting to note that the cost function $J[n]$ in (4) in this case can be viewed as a weighted composition of two cost functions: one when the training data is always present, i.e., $\boldsymbol{y}[n] = \boldsymbol{X}[n]\boldsymbol{h} + \boldsymbol{z}[n]$, and other when only noise is present, i.e., $\boldsymbol{y}[n] = \boldsymbol{z}[n]$. Fig. 4 illustrates the situation. When the training data is always present, the Weiner solution $\boldsymbol{h}_o$ is equal to the true channel $\boldsymbol{h}$. On the other hand, when only noise is present, the Weiner solution is at the origin. It is not surprising then, that the minimum of the combined cost function is in between both solutions, being a scaled version of $\boldsymbol{h}$.

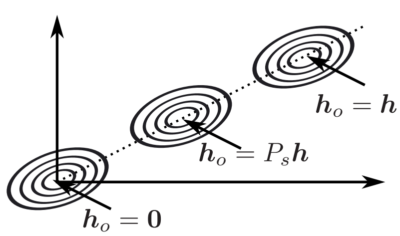

Fig. 4. Cost functions.

A. Convergence condition

For convergence analysis, we focus on the statistical behavior of the input random process $\hat{\boldsymbol{x}}[n]$ when an update happens, i.e., when $\mathcal{H}_1$ is decided. In this case, the input random process to this LMS-based block-updating algorithm with an abstract detector is still stationary and the input correlation matrix is $\boldsymbol{K}_x$. Hence, according to the small-step-size statistical theory, the channel estimate $\hat{\boldsymbol{h}}[n]$ converges to $P_s\boldsymbol{h}$ in the mean squared sense and the ensemble-average learning curve of the block LMS filter can be proved to converge to some constant value, with the following condition on the step-size parameter:

$$0 < \mu < \frac{2}{\max \lambda_i}$$

where λ_i is the ith eigenvalue of $\boldsymbol{K}_x$.

B. Average time constant

The average time constant τ_{av}, which reflects the average convergence speed, was originally defined to fit to the geometric series where the unit of time lasts the duration of one iteration cycle [6]. As the overall update probability is P_u, to make one update happen, the average number of iteration is $1/P_u$, hence the unit of time for an update lasts $1/P_u$ times the duration of one iteration cycle. And the τ_{av} is chosen such that

$$(1 - \mu\lambda_{av})^{2/P_u} = \exp\left(-\frac{1}{\tau_{av}}\right)$$

where $\lambda_{av} = \frac{1}{L_h}\sum_{i=1}^{L_h}\lambda_i$. The average time constant can be further expressed as

$$\tau_{av} = \frac{-1}{2P_u \ln(1 - \mu\lambda_{av})}.$$

When the step-size μ is very small, i.e., $\mu \ll 1$, then τ_{av} can be approximated as

$$\tau_{av} \approx \frac{1}{2P_u\mu\lambda_{av}}. \tag{6}$$

Hence the higher P_u, which is the probability of deciding $\mathcal{H}_1$, the faster the convergence speed. If the orthogonal transmission between source and relay is taken into consideration, τ_{av} will be doubled.

IV. Adaptation with Generalize Likelihood Ratio Test

In the previous section, we developed the general framework of applying LMS theory with an abstract detector to the situation where the presence of input training is probabilistic as described. It is generally known that there is no uniformly most powerful test for the composite hypothesis testing problem defined in (1). Hence we resort to the generalized likelihood ratio test (GLRT) [9] for the hypothesis testing problem and propose a heuristic approach for finding a "good" threshold which is used for hypothesis testing when the channel is unknown. Specifically, the hope is that the choice of threshold leads to improved convergence speed while minimizing estimation error. Without loss of generality, we assume that the training data at each block is the same, i.e., $\boldsymbol{x}[n] = \boldsymbol{x}$, and $\boldsymbol{X}$ is the tall

Tœplitz matrix with $X_{i,j} = x_{i-j}$. The pdf of the observation $\boldsymbol{y}[n]$ under $\mathcal{H}_1$ is

$$p(\boldsymbol{y}[n]; \boldsymbol{h}, \mathcal{H}_1) = \frac{1}{(\pi\sigma^2)^{(N+L_h-1)}} \exp\left(-\frac{1}{\sigma^2}(\boldsymbol{y}[n] - \boldsymbol{X}\boldsymbol{h})^H(\boldsymbol{y}[n] - \boldsymbol{X}\boldsymbol{h})\right) \quad (7)$$

and under $\mathcal{H}_0$ is

$$p(\boldsymbol{y}[n]; \mathcal{H}_0) = \frac{1}{(\pi\sigma^2)^{(N+L_h-1)}} \exp\left(-\frac{1}{\sigma^2}\boldsymbol{y}^H[n]\boldsymbol{y}[n]\right).$$

A. GLRT

The GLRT decides $\mathcal{H}_1$ if

$$\frac{p(\boldsymbol{y}[n]; \tilde{\boldsymbol{h}}, \mathcal{H}_1)}{p(\boldsymbol{y}[n]; \mathcal{H}_0)} > \gamma$$

where γ is such that P_{FA} does not exceed some maximum value, $\tilde{\boldsymbol{h}} = (\boldsymbol{X}^H\boldsymbol{X})^{-1}\boldsymbol{X}^H\boldsymbol{y}$ is the maximum likelihood estimate of $\boldsymbol{h}$ under $\mathcal{H}_1$ and $p(\boldsymbol{y}[n]; \tilde{\boldsymbol{h}}, \mathcal{H}_1)$ is similarly defined as in (7) by replacing $\boldsymbol{h}$ with $\tilde{\boldsymbol{h}}$. It is equivalent to say that the GLRT decides $\mathcal{H}_1$ if

$$T[n] > \log\gamma$$

where

$$\begin{aligned} T[n] &= \frac{-\tilde{\boldsymbol{h}}^H\boldsymbol{X}^H\boldsymbol{X}\tilde{\boldsymbol{h}} + 2\mathrm{Re}(\tilde{\boldsymbol{h}}^H\boldsymbol{X}^H\boldsymbol{y}[n])}{\sigma^2} \\ &= \frac{\boldsymbol{u}^H[n]\boldsymbol{u}[n]}{\sigma^2} \end{aligned}$$

with $\boldsymbol{u}[n] = (\boldsymbol{X}^H\boldsymbol{X})^{-1/2}\boldsymbol{X}^H\boldsymbol{y}[n]$. When $\mathcal{H}_0$ is true, $\boldsymbol{u}[n] \sim \mathcal{CN}(0, \sigma^2\boldsymbol{I})$ and $T[n]$ is Gamma distributed, i.e., $T[n] \sim \Gamma(L_h, 1)$. Define $P_{FA,\max}$ as the maximum probability of false alarm, i.e.,

$$\frac{1}{(L_h-1)!}\int_{\log\gamma}^{+\infty} t^{L_h-1}e^{-t}dt < P_{FA,\max}.$$

The threshold of $\log\gamma$ can be found by reversing the integration. When $\mathcal{H}_1$ is true, $\boldsymbol{u}[n] \sim \mathcal{CN}(\boldsymbol{X}\boldsymbol{h}, \sigma^2\boldsymbol{I})$, and $2T[n]$ is non-central chi-squared distributed with $2L_h$ degrees of freedom and mean $2T_{half}$ where

$$T_{half} = \frac{(\boldsymbol{X}\boldsymbol{h})^H(\boldsymbol{X}\boldsymbol{h})}{\sigma^2}. \quad (8)$$

For the same training symbols at each block, we can calculate the theoretical probability of detection (PD) by using the Marcum Q-function. As can be seen in Fig. 5, the simulated PD and theoretical PD agree with each other very well, as expected.

B. Finding the "optimal" threshold $\log\gamma$

In this subsection, we assume there is no constraint on maximum probability of false alarm $P_{FA,\max}$ and consider finding the "optimal" threshold $\log\gamma$ to "maximize" the convergence rate. As previously mentioned, the Wiener solution for (4) is a scaled version of the actual channel $P_s\boldsymbol{h}$. To have an estimate of the true channel, either an estimate of P_s is required or an

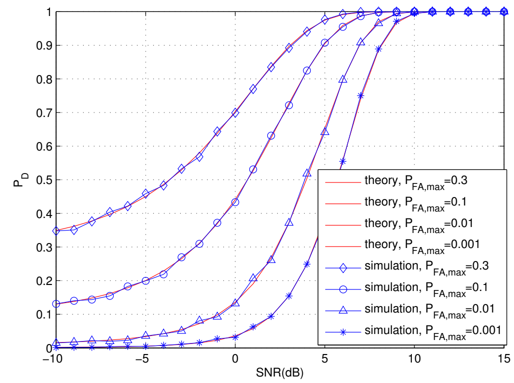

Fig. 5. Probability of detection when $P = 0.5, N = 2$, normalized $\boldsymbol{h} = [0.5 + 1i \quad 4 + 3i \quad 0.75 + 2.7i]$, SNR $= 5dB$, and $\boldsymbol{x} = \frac{1}{\sqrt{2}}[-1 - 1i \quad 1 - 1i]$

estimate of the channel gain, which is computed as the sum-square of the channel. We leave the accurate estimation of P_s or the estimation of the channel gain as separate problems and use the following performance metric [10]:

$$\xi(\boldsymbol{h}, \hat{\boldsymbol{h}}[n]) = E\left[\left\|\frac{\hat{\boldsymbol{h}}[n]}{\|\hat{\boldsymbol{h}}[n]\|} - \frac{\boldsymbol{h}}{\|\boldsymbol{h}\|}\right\|^2\right]. \quad (9)$$

This performance metric measures how close $\boldsymbol{h}[n]$ is to $\boldsymbol{h}$ in terms of the shape of the impulse response. For this performance metric, it makes intuitive sense that the larger P_s is, the smaller the minimum mean performance metric is.

On one hand, P_s determines where the update equation in (2) converges to and the estimation performance. On the other hand, as shown in (6), the convergence speed is dependent on P_u. Ideally for a fixed P, we want both P_s and P_u as large as possible. However, there are situations in which we cannot make both P_s and P_u as large as possible at the same time. As shown in Fig. 6, P_s is increasing with $\log\gamma$ and P_u is decreasing with $\log\gamma$. Hence an appropriate threshold $\log\gamma$ should be chosen to balance the convergence speed and the accuracy of channel estimate.

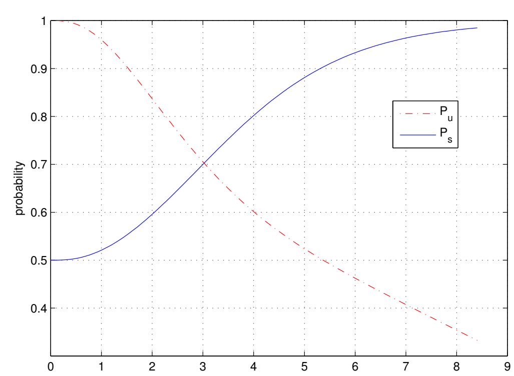

Fig. 6. P_S and P_D with parameters as in Fig.5

As it is mathematically intractable to find the optimal threshold $\log\gamma$ in general, we need to find a proper function of P_u and P_s which reflects how the threshold affects the convergence speed and the estimation performance. Based on the intuition about P_u and P_s, we propose the following heuristic approach to finding a good threshold:

$$\arg \max_{\log \gamma \in [0, +\infty]} P_u(P_s - 0.5). \quad (10)$$

As P_u affects convergence speed and P_s affects the accuracy of channel estimate, the threshold which gives the maximum of the product of P_u and $P_s - 0.5$ attempts to achieve a balance. In the function to be maximized, i.e., $P_u(P_s - 0.5)$, we use a term containing $P_s - 0.5$ rather than just P_s for the following two reasons. First, by using $P_s - 0.5$, the threshold which results in $P_s < 0.5$ is eliminated as a choice since P_u is never less than 0. Hence the condition $P_s > 0.5$ is automatically satisfied. Secondly, if we were to just use $P_u P_s$ as the heuristic, this would be equivalent to maximizing PP_D which can be made equal to P if P_D is 1. To make $P_D = 1$, we can always set the threshold 0, which is not necessarily a good choice at all situations.

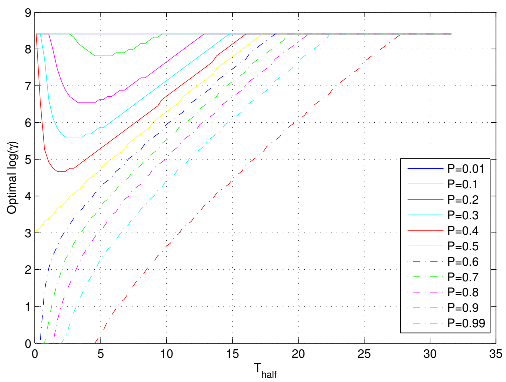

Fig. 7. Simulated optimal $\log \gamma$ versus T_{half} with $L_h = 3$ for maximizing $P_u(P_s - 0.5)$

As can be seen from distribution of T under different hypotheses, $P_D(P_s - 0.5)$ is only related to P, $\log \gamma$, L_h and T_{half}, which is originally defined in (8) and is half of the mean of $2T$ when $\mathcal{H}_1$ is true. Hence we plot in Fig. 7 the optimal $\log \gamma$ versus T_{half} with different transmission probability P, where the word "optimal" is in the sense of maximizing $P_u(P_s - 0.5)$. In calculating P_s, we use $P_{FA,\max}$ as for P_{FA} as an approximation. The plot makes intuitive sense: when P is small, e.g., $P = 0.01$, rather than setting a low threshold to make $P_D = 1$, a proper threshold should be large to also make the $P_{FA,\max}$ small such that the updating is not primarily by noise; when P is large, for a small T_{half}, e.g., $P = 0.99$ and $T_{half} = 3$, we use the threshold $\log \gamma = 0$ which makes both $P_D = P_{FA\max} = 1$, as the proportion of updating by noise, $(1 - P)P_{FA,\max}$, is very small. Another observation is that when T_{half} is large, no matter what the transmission probability is, the "optimal" threshold is always high. Of course there are undoubtedly numerous other approaches of finding a "good" threshold; yet the proposed one shows a good performance as will be seen in the simulation of an extensive number of scenarios.

To calculate P_D for computing P_u and P_s, it requires the knowledge of T_{half}, which is generally unknown before the channel estimation. In the following, we use a function of the mean of T to estimate T_{half}. The mean of $T[n]$ can be calculated as $T_{mean} = E[T[n]] = PT_{half} + L_h$. Hence $T_{half} = (E[T[n]] - L_h)/P$. In the implementation of the LMS algorithm, we use a moving average to represent $E[T]$. Hence a heuristic adaptive algorithm for the channel estimation problem is as follows:

Step 1: *Compute optimal* $\log \gamma$ *for the known* P *according to (10) offline. Initialize the channel estimate* $\hat{\boldsymbol{h}}[0]$.

Step 2: *Compute the estimate of* $\hat{T}_{half}[n] = (\hat{T}_{mean}[n] - L_h)/P$ *where* $\hat{T}_{mean}[n] = (\hat{T}_{mean}[n-1](n-1) + T[n])/n$ *is the estimate of the mean of* $T[n]$.

Step 3: *Find the optimal* $\log \gamma$ *for* $\hat{T}_{half}[n]$*; If* $T[n]$ *is greater than the found optimal* $\log \gamma$*, the channel estimate* $\hat{\boldsymbol{h}}[n]$ *is updated by (2); otherwise, the channel estimate remains the same* $\hat{\boldsymbol{h}}[n] = \hat{\boldsymbol{h}}[n-1]$.

Step 4: *Increase* n *by* 1. *Check if* n *is greater the total number of training blocks. If yes, the algorithm stops; otherwise, go to Step 2.*

V. Numerical results

In this section, we plot the error measure in (9) and we are primarily interested in the performance improvement of using the "optimal" threshold for *GLRT*. For purpose of comparison, we define the *genie-aided* LMS algorithm, in which we assume that the receiver knows if the observation contains training. In this ideal case, the destination can perfectly detect whether training data exists or not, thus $P_D = 1$, $P_{FA} = 0$, and subsequently $P_u = P$ and $P_s = 1$. The Weiner solution for the *genie-aided* LMS algorithm is $\boldsymbol{h}$, which is exactly the true channel.

In the simulation, we specify the parameter T_{half} to aid comparison. We use the normalized $\boldsymbol{h} = [0.5 + 1i \; 4 + 3i \; 0.75 + 2.7i]$, $N = 2$, and the input training is $\boldsymbol{x} = \frac{1}{\sqrt{2}} [-1 - 1i \quad 1 - 1i]$, to calculate the noise covariance $\sigma^2 = \frac{(\boldsymbol{Xh})^H \boldsymbol{Xh}}{T_{half}}$. We compare the performance of the heuristic algorithm using the "optimal" threshold with the performance of the adaptive algorithms with GLRT using the fixed threshold $\log \gamma = 0$ ($P_{FA,\max} = 1$) and $\log \gamma = 8.4059$ ($P_{FA,\max} = 0.01$). We consider transmission probabilities $P \in \{0.1, 0.5, 0.8, 0.99\}$. The performance is shown in Fig. 8 for $T_{half} = 2$, Fig. 9 for $T_{half} = 8$ and Fig. 10 for $T_{half} = 20$. With the increased transmission probability and increased T_{half}, the performance of the adaptive algorithm with the "optimal" threshold becomes closer to the performance of the *genie-aided* algorithm. In Fig. 8 at a low $T_{half} = 2$, for low transmission probability, $P = 0.1$, the fixed threshold $\log \gamma = 8.4059$, gives better performance than the fixed threshold $\log \gamma = 0$ while for $P = 0.8$ and $P = 0.99$, the fixed threshold $\log \gamma = 0$ gives better performance than the fixed threshold $\log \gamma = 8.4059$. For all these different transmission probabilities, the "optimal" threshold always arrives at the threshold which gives better performance. For the 12 scenarios shown from Fig. 8 to Fig. 10, we see that to arrive at the same final estimation error, the heuristic algorithm with the "optimal" threshold gives faster convergence speed than the algorithms with the fixed thresholds. This means that the "optimal" threshold which maximizes $P_u(P_{FA} - 0.5)$ can improve the convergence speed.

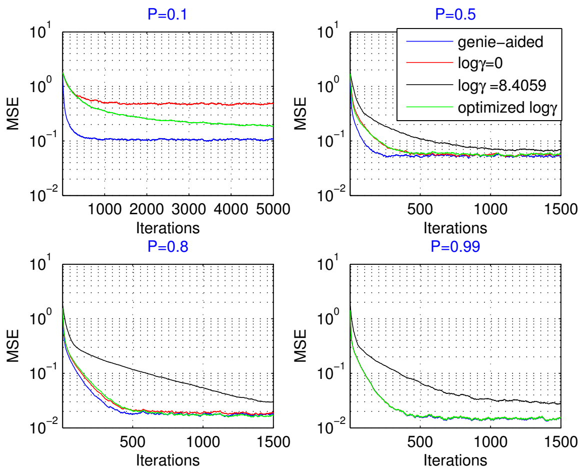

Fig. 8. Mean error measure with $T_{half} = 2$ at different P

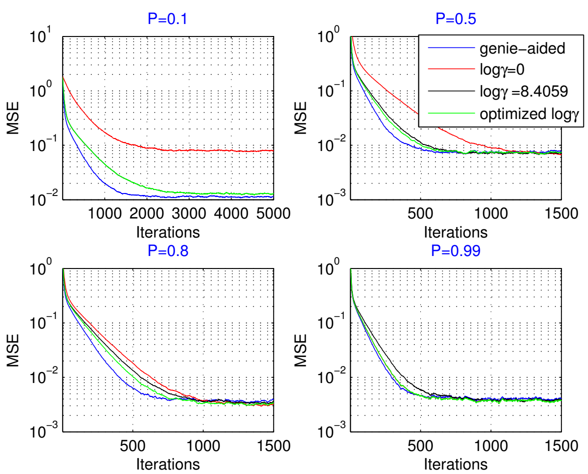

Fig. 9. Mean error measure with $T_{half} = 8$ at different P

VI. Conclusion

A new adaptive algorithm is proposed for the relay-to-destination channel estimation in DF relay networks when the presence of an input training sequence is probabilistic as a Bernoulli random variable. The adaptive algorithm combines a detector and the classical LMS adaptation. For each observation, detection of the presence of input training is performed before updating, as the update only happens with a positive decision. The threshold which the detector uses to make a decision is critical to balance the convergence speed and estimation error. On one hand, a low threshold

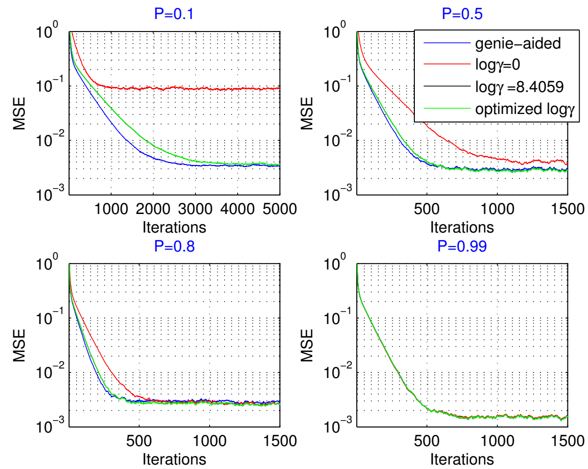

Fig. 10. Mean error measure with $T_{half} = 20$ at different P

increases the probability of deciding the presence of input training and results in increased convergence speed. On the other hand, the estimation performance is only related to the portion of correctly deciding the presence of input training and might require a high threshold. Hence, a heuristic objective function is proposed to capture the balance of convergence speed and estimation error. The objective function consists of the probability that the relay is forwarding, the probability of false alarm and the probability of detection, which is estimated in turn by using a moving average of the test statistics. Numerical results show that by using the "optimal" threshold which maximizes the proposed objective function, the adaptive algorithm has faster convergence speed than by using fixed thresholds.

References

[1] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—Part I: System description," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.

[2] C. Patel and G. Stuber, "Channel estimation for amplify and forward relay based cooperation diversity systems," *IEEE Trans. Wireless Commun.*, vol. 6, no. 6, pp. 2348–2356, Jun. 2007.

[3] F. Gao, T. Cui, and A. Nallanathan, "On channel estimation and optimal training design for amplify and forward relay networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1907–1916, May 2008.

[4] ——, "Optimal training design for channel estimation in decode-and-forward relay networks with individual and total power constraints," *IEEE Trans. Signal Process.*, vol. 56, no. 12, pp. 5937–5949, Dec. 2008.

[5] X. Zhou, T. Lamahewa, and P. Sadeghi, "Kalman filter-based channel estimation for amplify and forward relay communications," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, Nov. 2009, pp. 1498–1502.

[6] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2003.

[7] S. Werner, M. de Campos, and P. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 938–949, Apr. 2004.

[8] D. McLernon, "Analysis of LMS algorithm with inputs from cyclostationary random processes," *Electronics Letters*, vol. 27, no. 2, pp. 136–138, Jan. 1991.

[9] S. Kay, *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*. Prentice Hall, 1993.

[10] D. Morgan, J. Benesty, and M. Sondhi, "On the evaluation of estimated impulse responses," *IEEE Signal Process. Lett.*, vol. 5, no. 7, pp. 174–176, Jul. 1998.