

EM Based Channel Estimation in an Amplify-and-Forward Relaying Network

Nico Aerts, Iancu Avram, Dieter Duyck and Marc Moeneclaey

Department of Telecommunications

and Information Processing

Ghent University

9000 Gent, Belgium

Email: firstname.lastname@telin.ugent.be

Abstract—Cooperative communication offers a way to obtain spatial diversity in a wireless network without increasing hardware demands. The different cooperation protocols proposed in the literature [1] are often studied under the assumption that all channel state information is available at the destination. In a practical scenario, channel estimates need to be derived from the broadcasted signals. In this paper, we study the Amplify-and-Forward protocol and use the expectation-maximization (EM) algorithm to obtain the channel estimates in an iterative way. Our results show that the performance of the system that knows the channels can be approached at the cost of an increased computational complexity. In case a small constellation is used, a low complexity approximation is proposed with a similar performance.

Index Terms—cooperative communication, Amplify-and-Forward, EM algorithm, Viterbi decoding

I. INTRODUCTION

Due to multipath propagation, communication over a wireless channel suffers from signal level fluctuations. The channel between the source and the destination is said to be in deep fading if the signal level at the destination is too small to extract the information sent by the source. The detrimental effect of fading can be counteracted by creating multiple channels between the source and the destination. Because of this spatial diversity, the information from the source cannot be extracted at the destination only when all channels are simultaneously experiencing deep fading.

A. Cooperative communication

An obvious way to achieve spatial diversity is by using communication terminals with multiple antennas [2]. Besides increased hardware demands, it also implies that the terminal must be large enough to guarantee enough spacing between the antennas¹ (which is not the case for mobile phones, wireless sensors, ...). These limitations can be overcome by using a cooperative protocol, which utilizes other terminals in the network to relay the information broadcasted by the source. In practice, the timeframe allocated to a source terminal is divided in several timeslots. The source terminal broadcasts

information during the first timeslot while the remaining timeslots are used by the other terminals to relay this information in nonoverlapping slots. The destination will then combine the received signals to reconstruct the information transmitted by the source. In this article we study the amplify-and-forward (AF) protocol. As the name indicates, the relay is restricted to simply amplify and retransmit the signal it receives.

B. Estimation

In reality channel state information is not available and needs to be estimated. Therefore the source sends additional pilot symbols and the corresponding received signals are used to obtain the required estimates. This type of estimation is called data-aided estimation and is applied on the AF protocol in e.g. [3-5]. The drawback of this method is that large numbers of pilot symbols are needed to ensure good estimates. To that end, we will only transmit a few pilot symbols from which initial estimates are calculated. Next we use the expectation-maximization (EM) algorithm [6], [7] to improve the estimates in an iterative way. In order to ensure a low complexity relay, all estimates will be derived at the destination only.

C. Notations

All vectors are row vectors and are boldface; the Hermitian, statistical expectation, estimate and real part of the vector \mathbf{x} are denoted by \mathbf{x}^H , $\mathbb{E}[\mathbf{x}]$, $\hat{\mathbf{x}}$ and $\Re\{\mathbf{x}\}$ respectively; $x(k)$ is the k^{th} element of \mathbf{x} and $[\mathbf{x}, \mathbf{y}]$ denotes the concatenation of \mathbf{x} and \mathbf{y} ; the cardinality of the set \mathbb{X} is denoted as $|\mathbb{X}|$.

II. SYSTEM DESCRIPTION

We consider the system given in Fig. 1, consisting of a source, a destination and one relay; the extension to multiple relays transmitting in nonoverlapping slots is straightforward. The source S broadcasts K_d complex data symbols \mathbf{c} during the first timeslot, which are received by both the destination D and the relay R . As we assume independent frequency flat Rayleigh fading channels with additive white Gaussian noise, the signals received by the relay and the destination are given by

$$\begin{aligned} \mathbf{r}_S &= h_1 \mathbf{c} + \mathbf{n}_1, \\ \mathbf{d}_S &= h_2 \mathbf{c} + \mathbf{n}_2, \end{aligned} \tag{1}$$

¹In order to achieve the maximum benefit from diversity the spacing must be in the order of a wavelength, so that the channels are essentially uncorrelated

respectively. The channel coefficients h_x and the elements of the noise vectors \mathbf{n}_x are independent ZMCSCG (zero-mean circular symmetric complex Gaussian) distributed with variances H_x and N_x , respectively, with $x \in \{1, 2\}$. During the second timeslot, the destination receives the amplified signal sent by the relay

$$\mathbf{d}_R = h_3 \beta \mathbf{r}_S + \mathbf{n}_3 = \beta h_4 \mathbf{c} + \mathbf{n}_4, \quad (2)$$

where $h_4 = h_1 h_3$ and \mathbf{n}_4 consists of independent ZMCSCG variables with variance $N_4 = |h_3|^2 \beta^2 N_1 + N_3$. This way the information reaches the destination via two independent paths; the direct source-destination (S-D) link and the path over the relay (S-R-D). If the power constraint at the relay dictates the average energy $\frac{\mathbb{E}[|\beta \mathbf{r}_S|^2]}{K_d}$ per sent symbol to be E_r , a constant gain factor β can be taken:

$$\beta = \sqrt{\frac{E_r}{H_1 E_s + N_1}},$$

where E_s denotes the energy of the symbols sent by the source.

The broadcasted symbols \mathbf{c} result from coding and mapping

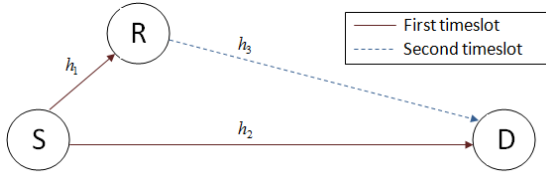


Fig. 1. Network containing a Source, a Relay and a Destination.

the useful information, represented by a vector of information bits \mathbf{b} . In this contribution we restrict ourselves to a source that uses a rate- $\frac{k}{n}$ convolutional code [8]. At the destination the 2 different versions of the same codeword ((1) and (2)), are combined, and applied to a rate- $\frac{k}{n}$ Viterbi decoder [9] to obtain the ML decision $\hat{\mathbf{b}}$ of \mathbf{b} . The ML decision of the codeword \mathbf{c} is defined as

$$\hat{\mathbf{c}} = \arg \max_{\tilde{\mathbf{c}} \in \mathbf{C}} p(\mathbf{d}_S, \mathbf{d}_R | \tilde{\mathbf{c}}), \quad (3)$$

where \mathbf{C} is the set of all possible codewords. When the trellis diagram of the code consists of M sections, the memoryless nature of the channel reduces (3) to

$$\hat{\mathbf{c}} = \arg \max_{\tilde{\mathbf{c}} \in \mathbf{C}} \prod_{m=1}^M p(\mathbf{d}_{S_m}, \mathbf{d}_{R_m} | \tilde{\mathbf{c}}_m), \quad (4)$$

where the notation \mathbf{x}_m is used to denote the components of \mathbf{x} corresponding to the m^{th} trellis section. Using (1) and (2), (4) can be transformed into

$$\hat{\mathbf{c}} = \arg \max_{\tilde{\mathbf{c}} \in \mathbf{C}} \sum_{m=1}^M \lambda(\mathbf{d}_{S_m}, \mathbf{d}_{R_m}, \tilde{\mathbf{c}}_m),$$

where

$$\lambda(\mathbf{d}_{S_m}, \mathbf{d}_{R_m}, \mathbf{c}_m) = \Re \left\{ \left(\frac{h_2^* \mathbf{d}_{S_m}}{N_2} + \frac{\beta h_4^* \mathbf{d}_{R_m}}{N_4} \right) \mathbf{c}_m^H \right\} \quad (5)$$

is the Viterbi decoder branch metric for the m^{th} trellis section. Notice that this branchmetric involves the maximum ratio combining of the S-D and S-R-D signals received at the destination.

III. ESTIMATION

As the noise variances N_x ($x = 1 \dots 3$) are long-term properties, we assume them to be known. The destination computes the branch metrics (5), but with h_2 , h_4 and N_4 replaced by their estimates. As mentioned before, the source broadcasts K_p pilot symbols \mathbf{c}_p prior to the information symbols \mathbf{c} . Based on the corresponding received signals, initial estimates can be derived at the destination and the expectation-maximization (EM) algorithm [6] is used to improve them in an iterative way.

If we now use the notation \mathbf{x}_t to indicate the concatenation of \mathbf{x}_p and \mathbf{x} , where \mathbf{x}_p relates to the pilot symbols, the signals received by the destination are given by

$$\begin{aligned} \mathbf{d}_{S_t} &= h_2 \mathbf{c}_t + \mathbf{n}_{2,t}, \\ \mathbf{d}_{R_t} &= \beta h_4 \mathbf{c}_t + \mathbf{n}_{4,t}. \end{aligned}$$

The receive structure at the destination is given by Fig. 2. The notation \hat{x}^i is used to indicate the estimate of x after the i^{th} iteration.

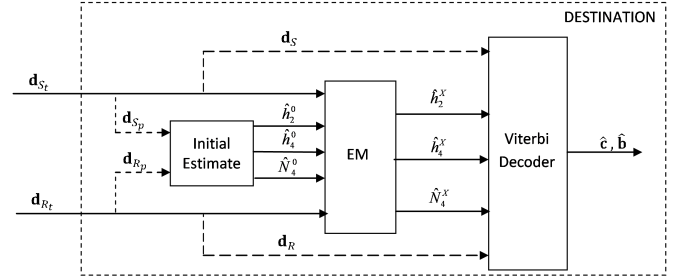


Fig. 2. Receive structure of the destination. X EM iterations are used to improve the initial ML estimates.

A. Initial estimate

Denoting $\mathbf{h} = (h_2, h_4, N_4)$, the pilot based ML estimates at the destination are found by solving

$$\hat{\mathbf{h}}^{(0)} = \arg \max_{\mathbf{h}} \ln p(\mathbf{d}_{S_p}, \mathbf{d}_{R_p} | \mathbf{h}).$$

Using the given channel models, it can be shown that

$$\hat{h}_2^{(0)} = \frac{\mathbf{d}_{S_p} \mathbf{c}_p^H}{K_p E_s}, \quad (6)$$

$$\hat{h}_4^{(0)} = \frac{\mathbf{d}_{R_p} \mathbf{c}_p^H}{\beta K_p E_s}, \quad (7)$$

$$\hat{N}_4^{(0)} = \frac{|\mathbf{d}_{R_p} - \hat{h}_4^{(0)} \beta \mathbf{c}_p|^2}{K_p}. \quad (8)$$

Notice that while the pilot based estimates of h_2 and h_4 are unbiased, the estimate of N_4 is not;

$$\mathbb{E} [\hat{N}_4^{(0)} | N_4] = N_4 \left(1 - \frac{1}{K_p} \right).$$

On average, the estimate of N_4 will be smaller than its actual value, making the contribution of the S-R-D path to the branch metric larger than it should. This effect can be avoided by using the unbiased estimate

$$\hat{N}_4^{(0)} = \frac{|\mathbf{d}_{R_p} - \hat{h}_4^{(0)} \beta \mathbf{c}_p|^2}{K_p - 1}.$$

B. EM algorithm

Previous estimates were solely based on the pilot symbol part of the received signals, i.e. \mathbf{d}_{S_p} and \mathbf{d}_{R_p} . The ML estimates that also utilize the information contained in \mathbf{d}_S and \mathbf{d}_R are found by solving

$$\hat{\mathbf{h}} = \arg \max_{\mathbf{h}} \ln p(\mathbf{d}_{S_t}, \mathbf{d}_{R_t} | \hat{\mathbf{h}}), \quad (9)$$

where $p(\mathbf{d}_{S_t}, \mathbf{d}_{R_t} | \mathbf{h})$ is given by

$$\frac{1}{|\tilde{\mathcal{C}}|} \sum_{\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}} p(\mathbf{d}_{S_t}, \mathbf{d}_{R_t} | \mathbf{h}, [\mathbf{c}_p, \tilde{\mathbf{c}}]),$$

as we assume to codewords to be equally likely. The summation runs over all possible codewords and makes a practical implementation impossible. To circumvent this brute force marginalization, the EM algorithm is used to approximate the estimates given by (9). The EM algorithm is an iterative algorithm with each iteration consisting of an Expectation step and a Maximization step [7]. The Expectation step during the i^{th} iteration boils down to calculating the function

$$Q(\mathbf{h}, \hat{\mathbf{h}}^{(i-1)}) = \mathbb{E} \left[\ln p(\mathbf{d}_{S_t}, \mathbf{d}_{R_t} | \mathbf{h}, \mathbf{c}_t) \middle| \mathbf{d}_{S_t}, \mathbf{d}_{R_t}, \hat{\mathbf{h}}^{(i-1)} \right].$$

During the maximization step, $\hat{\mathbf{h}}^i$ is found as the value of \mathbf{h} that maximizes $Q(\mathbf{h}, \hat{\mathbf{h}}^{(i-1)})$. This yields

$$\hat{h}_2^{(i)} = \frac{\mathbf{d}_{S_t} \mathbf{u}_t^{(i),H}}{K_t E_s}, \quad (10)$$

$$\hat{h}_4^{(i)} = \frac{\mathbf{d}_{R_t} \mathbf{u}_t^{(i),H}}{\beta K_t E_s}, \quad (11)$$

$$\hat{N}_4^{(i)} = \frac{|\mathbf{d}_{R_t} - \beta \hat{h}_4^{(i)} \mathbf{u}_t^{(i)}|^2 + \beta^2 |\hat{h}_4^{(i)}|^2 (K_t E_s - |\mathbf{u}_t^{(i)}|^2)}{K_t}, \quad (12)$$

where $K_t = K_p + K_d$ and

$$\mathbf{u}_t^{(i)} = \mathbb{E} \left[\mathbf{c}_t \middle| \mathbf{d}_{S_t}, \mathbf{d}_{R_t}, \hat{\mathbf{h}}^{(i-1)} \right]$$

denotes the a posteriori expectation of \mathbf{c}_t .

C. Computation of the a posteriori expectation

First of all we note that, as the pilot symbols are known, \mathbf{u}_p^i equals \mathbf{c}_p . The problem is thus reduced to the computation of \mathbf{u}^i which can be expressed as

$$\mathbf{u}^{(i)} = \frac{\sum_{\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}} \tilde{\mathbf{c}} p(\mathbf{d}_S, \mathbf{d}_R | \hat{\mathbf{h}}^{(i-1)}, \tilde{\mathbf{c}})}{\sum_{\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}} p(\mathbf{d}_S, \mathbf{d}_R | \hat{\mathbf{h}}^{(i-1)}, \tilde{\mathbf{c}})}. \quad (13)$$

Brute-force evaluation of (13) is not feasible, as the summations run over $|\tilde{\mathcal{C}}|$ terms. Instead, we will derive the a posteriori symbol expectations from the trellis diagram using a forward recursion only. A rate- $\frac{k}{n}$ convolutional code is used with memory order $K-1$, so each section of the trellis diagram consists of $2^{k(K-1)}$ states and 2^{kK} branches. At each state T_j 2^k branches enter, with corresponding symbols denoted as \mathbf{c}_{lj} if the emerging state is T_l , with $j \in \{1, \dots, 2^{k(K-1)}\}$. As illustration, Fig. 3 shows the first 4 sections of the trellis representation of the rate $\frac{1}{2}$ (15,17) convolutional code with initial state T_1 . Again assuming the trellis diagram to have

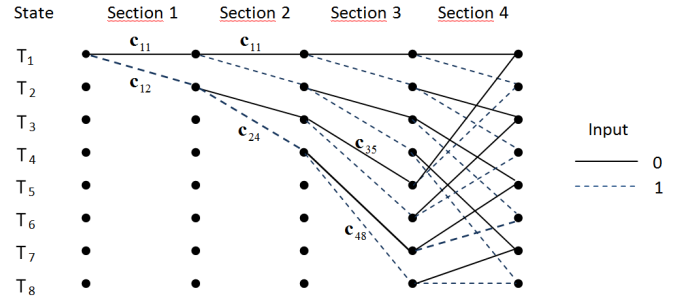


Fig. 3. First 4 sections of the trellis representation of the terminated (15,17) convolutional code.

M sections, the likelihood of a codeword $\tilde{\mathbf{c}}$, conditioned on $\mathbf{h} = \hat{\mathbf{h}}^{(i-1)}$ can be expressed as

$$p(\mathbf{d}_S, \mathbf{d}_R | \hat{\mathbf{h}}^{i-1}, \tilde{\mathbf{c}}) = \prod_{m=1}^M p(\mathbf{d}_{S_m}, \mathbf{d}_{R_m} | \hat{\mathbf{h}}^{i-1}, \tilde{\mathbf{c}}_m)$$

and (13) is found by following the steps listed below.

We assume a terminated convolutional code with T_1 as common starting and ending state. For each state T_j we now introduce the parameters $P_j^{(m)}$ and $C_j^{(m)}$, where m indicates the section of the trellis diagram.

Initialization step The vectors $C_j^{(0)}$ are empty and all $P_j^{(0)}$ are zero except $P_1^{(0)}$ which is equal to one. The next step will start evaluating section $m = 1$.

Updating step For each branch of this section, the following metric is calculated

$$p_{lj}^{(m)} = e^{\frac{2}{N_2} \Re \{ \hat{h}_2^* \mathbf{d}_{S_m} \mathbf{c}_{lj}^H \} + \frac{2}{N_4} \Re \{ \hat{h}_4^* \beta \mathbf{d}_{R_m} \mathbf{c}_{lj}^H \}}$$

which is proportional to the likelihood of the transition from state T_l to state T_j during the m^{th} trellis section. We update $P_j^{(m)}$ as

$$P_j^{(m)} = \sum_{l \in \mathbb{T}_j} P_l^{(m-1)} p_{lj}^{(m)}.$$

Next, $C_j^{(m)}$ is found as

$$\sum_{l \in \mathbb{T}_j} [C_l^{(m-1)}, c_{lj} P_l^{(m-1)}] p_{lj}^{(m)}. \quad (14)$$

This step is repeated with m incremented by 1 until section M is evaluated.

Termination step As the convolutional code is terminated, T_1 is also the ending state and $u^{(i)}$ is found as

$$\frac{C_1^{(M)}}{P_1^{(M)}}.$$

IV. COMPLEXITY CONSIDERATIONS

We express the implementation complexity of the receive structure in terms of the number of real valued summations and multiplications that are needed to obtain $\hat{\mathbf{b}}$ from the received signals \mathbf{d}_{S_t} and \mathbf{d}_{R_t} . We hereby neglect the small number of table look-ups needed to evaluate functions (e.g. exp). If channel state information is available, this computational complexity is determined by the complexity of the Viterbi decoder. Introducing \mathbb{S} , the set of constellation points used by the mapper, the dominant terms of the number of summations and multiplications in the Viterbi decoder are given by

$$2 \frac{K_d \log_2 |\mathbb{S}|}{n} 2^{kK} + 4K_d 2^{\min(kK, n)} \quad (15)$$

and

$$4K_d 2^{\min(kK, n)}, \quad (16)$$

respectively. The first term in (15) results from the calculation of the path metrics and the comparison of these path metrics in every state while the second term in (15) and the term in (16) result from the calculation of all branch metrics. If channel information is not known, u^i needs to be calculated which introduces at least an extra

$$\frac{K_d^2 \log_2 |\mathbb{S}|}{2n} 2^{kK}$$

summations and multiplications per EM iteration, which corresponds to the calculation of (14). Given the fact that K_d is usually much larger than $4n$ in a practical implementation, estimating the channel parameters by means of the EM algorithm will dramatically increase the complexity at the destination, as compared to the complexity of the Viterbi decoder. This problem can be circumvented by computing $u^{(i)}$ under the assumption that the transmitted symbols are uncoded; in this case, the l^{th} element of the conditional expectation is given by

$$u^{(i)}(l) = \frac{\sum_{s \in \mathbb{S}} s p(d_S(l), d_R(l)) | c(l) = s, \hat{\mathbf{h}}^{(i-1)}}{\sum_{\tilde{s} \in \mathbb{S}} p(d_S(l), d_R(l)) | c(l) = \tilde{s}, \hat{\mathbf{h}}^{(i-1)}} \quad (17)$$

and the complexity of this calculation, in case no further simplification of (17) is possible, is reduced to $10|\mathbb{S}|K_d$ summations and $21|\mathbb{S}|K_d$ multiplications per EM iteration.

V. SIMULATIONS

In this section, we study the performance of the AF communication system that includes the channel estimates. By means of computer simulation results we determine the Frame Error Rate (FER) and the mean-square error (MSE) of the estimates. Unless otherwise specified, we use the same system parameters as described in [10], i.e., $K_d = 260$, $H_1 = H_2 = H_3 = 1$ and a rate- $\frac{1}{2}$ (15,17) convolutional code is used. Furthermore we will assume $N_1 = N_3 = 2N_2$ and the power constraints at the source and the relay to be identical, i.e. $E_r = E_s$. Hence, the S-D link has a 3 dB higher SNR than the S-R and R-D links. For each constellation Gray mapping is applied.

A. Number of pilot symbols

The pilot based estimates provide the initialization for the EM algorithm. However, if these initial estimates are inaccurate, the EM algorithm will converge to a stationary point of $p(\mathbf{d}_{S_t}, \mathbf{d}_{R_t} | h_2, h_4, N_4)$ that might be different from the desired global maximum. Adding more pilot symbols can improve this accuracy but at the cost of a reduced power and bandwidth efficiency. Indeed, the energy per information bit (E_b), the energy per symbol (E_s), the information bitrate (R_b) and the symbol rate (R_s) are related by

$$E_s = \frac{K_d}{K_p + K_d} \frac{k}{n} \log_2(|\mathbb{S}|) E_b,$$

$$R_s = \frac{K_p + K_d}{K_d} \frac{n}{k} \frac{R_b}{\log_2(|\mathbb{S}|)}.$$

In table I the E_b/N_2 ratio needed to achieve a FER of 0.01 is given for several number of EM iterations in case two, five and ten pilot symbols are used and when BPSK mapping is applied. In terms of FER, it is possible to find an optimal

| | $K_p = 10$ | $K_p = 5$ | $K_p = 2$ |
|------------------|------------|-----------|-----------|
| Pilot based only | 15.68 | 16.07 | 18.90 |
| 1 EM iteration | 14.99 | 15.16 | 16.04 |
| 2 EM iterations | 14.83 | 14.89 | 15.47 |
| 3 EM iterations | 14.79 | 14.80 | 15.29 |

TABLE I
 E_b/N_2 (dB) RATIO NEEDED TO ACHIEVE A FER OF 0.01 (BPSK).

value of K_p (tradeoff between a good initial estimate and a high enough value of E_s). However, in order to limit the reduction of the information bitrate caused by adding pilot symbols, we will settle with a number of pilot symbols which is less than this optimum. Table I shows that the benefit of using ten pilot symbols over five is negligible when the EM algorithm is applied. As the difference between two and five pilot symbols is significant, we will use $K_p = 5$ throughout the rest of this paper, unless noted otherwise.

B. FER and MSE performance

Fig. 4 shows the FER for several cases, assuming QPSK transmission: (i) channel parameters are known to the destination ($K_p = 0$); (ii) the estimator perfectly knows all symbols transmitted (pilot and data), yielding a lower bound on FER; (iii) the estimator makes use of pilot symbols only; (iv) the estimator makes use of 2 EM iterations; (iv) a non-cooperative system with channel information known to the destination ($K_p = 0$), using a rate- $\frac{1}{4}$ (15,17,13,15) convolutional code. In cases (ii), (ii) and (iv), the estimate $N_4^{(0)}$ is according to (8).

As compared to case (i), the FER from case (ii) is degraded because of the presence of pilot symbols (this degradation amounts to $10 \log_{10}(\frac{265}{260}) = 0.08\text{dB}$ and the imperfect channel estimation (this degradation turns out to be negligibly small). The FER resulting from 2 EM iterations is close to the FER from cases (i) and (ii), whereas a degradation of about 1 dB is observed when using pilot symbols only. The non-cooperative system achieves a diversity order of only 1, whereas the cooperative system has diversity order 2.

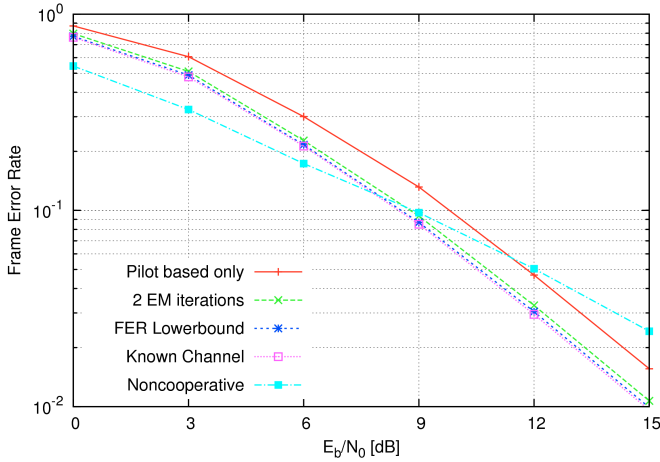


Fig. 4. Frame Error Rate in case QPSK mapping is used.

The mean square error (MSE) related to the estimation of h_4 using pilot symbols only or assuming all symbols transmitted to be known to the estimator (yielding a lower bound on MSE) is given by

$$\mathbb{E} [|\hat{h}_4 - h_4|^2] = \frac{N_3 + \beta^2 N_1}{K \beta^2 E_s}$$

with K equal to K_p and K_t respectively. Similar results are obtained for h_2 :

$$\mathbb{E} [|\hat{h}_2 - h_2|^2] = \frac{N_2}{K E_s}$$

To characterize the behavior of the MSE of the estimates of N_4 , a MSE lower bound can be found by assuming the estimate of h_4 to be perfect, and all transmitted symbols to be

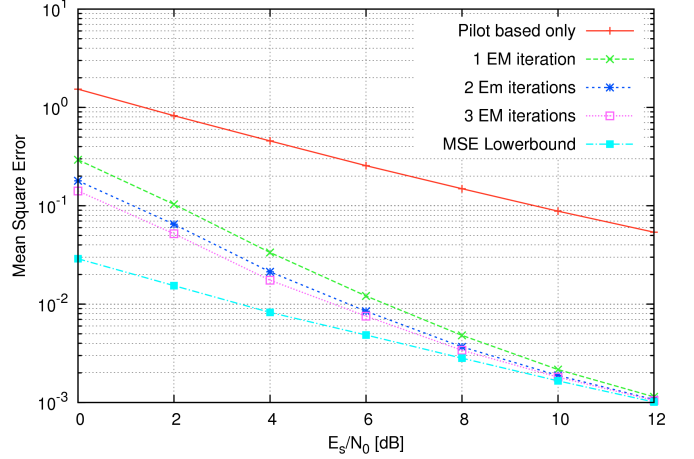


Fig. 5. Mean Square Error of h_4 in case QPSK mapping is used.

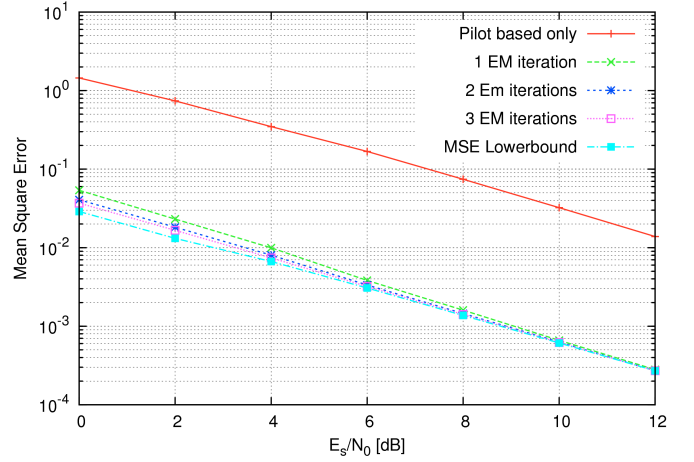


Fig. 6. Mean Square Error of N_4 in case QPSK mapping is used.

known;

$$\frac{\mathbb{E} [|\hat{N}_4 - N_4|^2]}{E_s^2} \geq \frac{\mathbb{E} [N_4^2]}{K E_s^2} = \frac{N_3^2 + 2\beta^2 N_1 N_3 + 2\beta^4 N_1^2}{K E_s^2}.$$

Figs. 5 and 6 show the MSE related to the estimation of h_4 and N_4 , respectively. Observe the rather poor performance when only pilot symbols are used. When using EM iterations, the MSE when estimating h_4 is larger than the corresponding MSE lower bound because the symbol a posteriori expectations are different from the symbols transmitted, especially at low SNR. The MSE when estimating N_4 by means of EM iterations is not very sensitive to not knowing neither h_4 nor the symbols transmitted.

In section III-A an unbiased pilot based estimation of N_4 was proposed as opposite to the biased ML estimation. We have verified that in the case of estimation from pilots only, the unbiased estimated yields a performance improvement on the order of 0.15 dB (BPSK) or 0.1 dB (QPSK) at FER = 0.01, whereas this effect is negligible (on the order of 0.01 dB) when the EM algorithm is applied. Nonetheless we will

use the unbiased pilot based estimate as there is no increase in complexity.

In section IV we argued that an exact implementation of the EM algorithm would have a large impact on the computational complexity at the destination, and an approximation was introduced to circumvent this disadvantage. To illustrate the difference, table II contains the exact number of summations (\sum) and multiplications (\prod) that are needed to obtain $\hat{\mathbf{b}}$ from the received signals \mathbf{d}_{S_t} and \mathbf{d}_{R_t} in case BPSK mapping is used. Given this constellation, (17) is expressed as

$$u^{(i+1)}(l) = \sqrt{E_s} \tanh \left(2\sqrt{E_s} \Re \left\{ \frac{d_S^H(l) \hat{h}_2^{(i)}}{N_2} + \frac{d_R^H(l) \beta \hat{h}_4^{(i)}}{\hat{N}_4^i} \right\} \right)$$

to reduce the complexity even further. We observe that the

| | Pilot based | EM algorithm (exact) | EM algorithm (approximation) |
|---------|-------------|----------------------|------------------------------|
| \sum | 6393 | 282831 | 11889 |
| \prod | 6898 | 556278 | 14018 |

TABLE II
COMPUTATIONAL COMPLEXITY AT THE DESTINATION

complexity reduction resulting from the approximation is significant. The effect of this approximation on FER performance is illustrated in table III, which shows the E_b/N_2 ratio needed to achieve a FER of 0.01 for a BPSK, QPSK or 8-PSK constellation. In case of the 8-PSK constellation a rate- $\frac{1}{3}$ (15,17,13) convolutional code is used.

| BPSK | E_b/N_2 | Difference |
|------------------|----------------|------------|
| FER Lowerbound | 14.69 | + 0.00 |
| EM algorithm | 14.83 | + 0.14 |
| Approximation | 14.91 | + 0.22 |
| Pilot based only | 16.07 | + 1.38 |
| QPSK | E_b/N_2 (dB) | Difference |
| FER Lowerbound | 15.06 | + 0.00 |
| EM algorithm | 15.14 | + 0.08 |
| Approximation | 15.35 | + 0.29 |
| Pilot based only | 16.18 | + 1.12 |
| 8-PSK | E_b/N_2 (dB) | Difference |
| FER Lowerbound | 13.70 | + 0.00 |
| EM algorithm | 13.77 | + 0.07 |
| Approximation | 14.36 | + 0.66 |
| Pilot based only | 14.90 | + 1.20 |

TABLE III
 E_b/N_2 (dB) RATIO NEEDED TO ACHIEVE A FER OF 0.01.

The results illustrate that a gain on the order of 1 dB is achievable as compared to estimation based on pilot symbols only, when the exact EM algorithm is used to improve the initial estimates. It also shows that the approximated EM yields a small degradation as compared to exact EM for small constellations only. For larger constellations the a posteriori

expectation will deteriorate considerably from the actual symbols and as a result the FER and the MSE of the estimates increases.

VI. CONCLUSIONS

In this contribution we discussed an iterative estimation algorithm for the cooperative Amplify-and-Forward protocol. The complexity at the relay has been kept minimal by estimating all unknown parameters at the destination. Here the EM algorithm is used to improve the initial pilot based ML estimates in an iterative way. Only a few pilot symbols are needed to achieve a satisfactory FER performance after 2 EM iterations.

In case a convolutional code is used, a method to compute the a posteriori symbol expectations, needed during the Expectation step, is proposed. However, the computational complexity increases dramatically as compared to a receiver that knows the channel. We showed that this problem can be tackled by assuming the received symbols are uncoded; for small constellations, the resulting FER performance is only slightly degraded as compared to using the exact EM algorithm.

ACKNOWLEDGMENT

The authors wish to acknowledge the activity of the Network of Excellence in Wireless COMMUNICATIONS NEW-COM++ of the European Commission (contract n. 216715) that motivated this work.

REFERENCES

- [1] J. N. Laneman *Cooperative Diversity in Wireless networks: Algorithms and Architectures* Massachusetts Institute of Technology, 2002
- [2] G. Foschini *Layered space time architecture for wireless communication in a fading environment when using multi-element antennas* Bell Labs. Tech., vol. 1, pages 41-59, 1996
- [3] C. Patel and G. Stuber *Channel Estimation for Amplify and Forward Relay Based Cooperation Diversity Systems* IEEE Trans. on Wireless Commun., vol. 6, No. 6, June 2007, p2348-2356
- [4] F. Gao, T. Cui and A. Nallanathan *On Channel Estimation and Optimal Training Design for Amplify and Forward Relay Networks* IEEE Trans. on Commun., 2008, volume 7, number 5, pages 1907-1916
- [5] B. Gedik and M. Uysal *Two channel estimation methods for amplify-and-forward relay networks* Canadian Conference on Electrical and Computer Engineering, May 4-7, 2008, pages 615-618
- [6] A. Dempster, N. Laird and D. B. Rubin *Maximum Like-lihood From Incomplete Data Via the EM Algorithm* J. Royal Statistical Soc., vol. 39, no. 1, pp. 1-38, 1977
- [7] H. Wymeersch, F. Simoons and M. Moeneclaey *Code-aided joint channel estimation and frame synchronization for MIMO systems* Fifth IEEE Workshop on Signal Processing Advances in Wireless Communications, Lisboa, Portugal, July 11-14, 2004
- [8] S. Lin and D. Costello *Error Control Coding* Pearson Education Inc., second edition, 2004
- [9] A. J. Viterbi *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm* IEEE Transactions Information Theory, 1967, volume 13, April, pages 260-269
- [10] A. Stefanov and E. Erkip *Cooperative Coding for Wireless Networks* IEEE Transactions on communications, 2004, volume 52, number 9, pages 1470-1476