IMPROVED OFDM RECEIVER WITH ITERATIVE CHANNEL ESTIMATION AND TURBO DECODING

Marta Perez Portugal[1] *& Yeheskel Bar-Ness*[2]

[1] Universitat Politecnica de Catalunya (UPC),Barcelona, Spain & NJIT
[2] New Jersey Institute of Technology (NJIT) Newark, NJ 07102 (USA).
[1] marta29@casal.upc.edu,[2]barness@yegal.njit.edu

ABSTRACT

Error correcting codes have become necessary in wireless digital communications systems. Turbo codes are one of the most powerful types of error control codes currently available [1]. There is a tradeoff in turbo codes between latency and BER which depends on the choice of interleaver size, number of decoding iterations and the algorithm used in the decoding process. In this paper we propose a reduced-complexity OFDM receiver with joint iterative channel estimation and iterative decoding employing turbo coding. Improved performance of this receiver is observed in comparison with others current channel estimation and decoding techniques.[1]

1. INTRODUCTION

Due to the interest in mobile-radio communications, it became necessary to increase the code diversity. New coding techniques other than Trellis Coded Modulation (TCM)[3] have been proposed to maximize the code diversity; such as the Bit-Interleaved Coded Modulation (BICM)[4], or Bit-Interleaved Coded Modulation with Iterative Decoding (BICM-ID) [5], which uses soft decision metrics and the soft-input soft-output decoder based on a posteriori probabilities (APP-SISO decoder) [6]. The later outperforms the first by providing larger coding gains. Another coding technique based on turbo codes [1], achieves near channel capacity error correcting, i.e it is able to transmit information across the channel with low (approaching zero) bit error rate.

Furthermore, multimedia services in wireless communications systems requires the use of multi-carrier transmission schemes, like Orthogonal Frequency Divison Multiplexing (OFDM), as it allows high transmission rates with high spectral efficiency.However channel estimation in OFDM systems is a critical problem when the channel undergoes frequency selective and fast time variation. Iterative channel estimation is an adequate solution for this problem as it improves system performance and reduces pilot overhead when using Pilot-Symbol Aided (PSA) techniques. Our proposed receiver uses comb-type channel estimation (technique of inserting pilot tones used when the channel varies rapidly ,even over one OFDM symbol).

In this paper similar to [2] we propose a scheme for joint channel estimation and decoding via the Expectation Maximization (EM) Algorithm using turbo decoding that perfoms less calculations than in the conventional receiver obtainig better performance [2] when only a small number of decoding iterations are carried out.Iterative channel estimation fit perfectly to be combined with the iterative decoding process, since the probabilities of the transmitted symbols, which is the information that the APP-SISO decoder provides, is the information that the EM Algorithm needs for the channel estimation. Nevertheless, the resulting computational load at the receiver, in terms of number of calculations, is quite high, leading to high latency. Thus the proposed scheme goal is to reduce the latency at the receiver.

2. THE TRANSMITTER AND CHANNEL MODEL

2.1 The encoder

The encoder is a standard turbo encoder composed of a rate $\frac{1}{2}$ Parallel Concatenated Convolutiona Code (PCCC), which is a concatenation of two rate $\frac{1}{2}$ recursive systematic convolutional (RSC) codes separated by a pseudo-random interleaver. The output of the encoder is divided in groups of $\log_2(Q)$ bits (Q is the constellation size) and modulated following a labelling map. In addition equidistant pilot symbols are inserted into the OFDM symbol.

2.2 OFDM channel model

After encoding, the bit sequence is mapped to a symbol sequence for transmission on a set of *N* OFDM subcarries.A guard interval *GI* of cyclic prefix is inserted in order to aviod *ISI* from previous symbols. The OFDM symbols are pulse-shaped and sent over the multipath time varying frequency selective channel. At the receiver, assuming proper baud synchronization, after filtering, the *GI* is removed, and the received data samples are tranformed into the frequency domain via *N*-point *FFT*. Thus we have at the receiver the following data model in the frequency domain:

$$\mathbf{y} = diag(\mathbf{x})\mathbf{h} + \mathbf{n} \tag{1}$$

where **y** and **h** are *N* dimensional vectors of the frequency domain received data and channel response, and $diag(\mathbf{x})$ denotes a diagonal matrix formed from the symbol vector **x**. Adding the cyclic prefix turning **h** to be the *FFT* of the time domain channel *h*, which is the complete channel response including transmitter filter, the multipath channel and the receiver filter.

2.3 WSSUS channel

The channel is assumed a Wide Sense Stationay Uncorrelated Scattering (WSSUS)[7] mobile radio channel whose instantaneus response is given by,

$$h(\tau,t) = \lim_{L\to\infty} \frac{1}{\sqrt{L}} \sum_{n=1}^{L} \exp^{j(\theta_n + 2\pi f_{D_n}t)} \delta(\tau - \tau_n) \tag{2}$$

where *L* is the number of paths that superpose the signal. Each echo has a random delay τ_n, a random phase θ_n and rotates with a random Doppler shift f_{D_n}. The Power Delay Profile is an exponential probability density function with $p(T_{max})/p(0) = 10^{-5}$ and the Doppler Power Spectrum corresponds to a Jake's Spectrum which models an isotropic scattering.

3. CONVENTIONAL JOINT ITERATIVE CHANNEL ESTIMATION AND DECODING VIA THE EM ALGORITHM

This receiver performs EM Algorithm following the decoding iteration . It uses the a posteriori probabilities of the bits provided by the turbo code (see Fig.1 B) in order to get new channel estimates (see Fig.1 C), at the same time these bits are used by the demodulator to obtain new metrics and send them to the turbo decoder again (see Fig.1 A).

Figure 1: Receiver

3.1 The decoder

The decoder is basically the concatenation of two soft-input soft-output (SISO) APP estimators. The APP module has two inputs and two outputs. The two inputs corresponds to the probabilities $\lambda(c;I)$ and $\lambda(u;I)$ which are the a priori probabilities of the coded sequence(systematic and parity data) and uncoded sequence (only systematic data). The outputs of the APP module are $\lambda(c;O)$ and $\lambda(u;O)$ which constitute the extrinsic information to be exchanged between both decoders. The a posteriori probabilities of the coded symbols are never used by the decoding algorithm, but we use this probabilities for the channel estimation and in the demodulator in order to update the metrics of the received sequence. In Fig. 1 we show the receiver scheme. The APP probabilities of the symbols $p(x_k \mid \mathbf{y}, \mathbf{h})$ computed from $\lambda(c;O)$ are expressed as [6],

$$
\begin{aligned}
\lambda_k(c;O) = \log(& \sum_{e:c(e)=c} \exp\{\alpha_{k-1}[S^S(e)] + \lambda_k[u(e);I] \\
& + \beta_k[S^E(e)]\}) + K_c
\end{aligned}
\tag{3}
$$

$$
\begin{aligned}
\lambda_k(u;O) = \log(& \sum_{e:u(e)=c} \exp\{\alpha_{k-1}[S^S(e)] + \lambda_k[c(e);I] \\
& + \beta_k[S^E(e)]\}) + K_u
\end{aligned}
\tag{4}
$$

where K_c and K_u are normalization constants, e is the edge in the trellis section which represent the transitions between the trellis states, $S^S(e)$ is the starting state at time k in a section of the trellis, $S^E(e)$ is the ending state, $u(e)$ is the uncoded symbol and $c(e)$ is the coded symbol. In order to simplify the computational load of the algorithm we use the log-domain algorithm. Assuming proper channel interleaving (the bits are independent) the APP symbol probabilities are the product of the APP corresponding to all bits. Fot the computation of the forward probabilities ($\alpha_k(s)$) and backward probabilitis ($\beta_k(s)$)see [6]. In the scheme the demodulator computes the log-probabilities of the constellation symbols bit ($\lambda(c;I)$) from the received sequence y and the channel estimates $\hat{h}$ obtained from interpolation of the pilot symbols in the first iteration. After the first iteration, the demodulator uses also the a priori information ($\lambda(c;O)$) provided by the turbo decoder in each decoding iteration, in order to compute the metrics. The turbo decoder uses these metrics to obtain the extrinsic information for the other decoder.

3.2 The EM algorithm for channel estimation and decoding

The main reason to use the EM Algorithm is because a maximum likelihood solution has no closed form solution [9] .The EM Algorithm consists of two steps: an expectation step and a maximization step. Considering the terminology of the EM Algorithm we define the set $\mathbf{z} = \{\mathbf{y}, \mathbf{x}\}$ as the complete data set, $\mathbf{y}$ is the observed data set and $\mathbf{h}$ the parameter set to be calculated. The problem is to find is $\mathbf{h}$ that maximizes the log-likelihood function $\log p(\mathbf{z} \mid \mathbf{h})$. But since we do not have $\mathbf{z}$, the EM Algorithm maximizes instead the

expectation of $\log p(\mathbf{z}|\mathbf{h})$ given $\mathbf{y}$ in each iteration to find the current estimation of $\mathbf{h}$; $\mathbf{h}^{[i]}$. The expectation is carried out over all the possible values of $x \in X$.

3.2.1 Expectation step

The expectation step is given by,

$$
\begin{aligned}
E_{x \in X}(\log p(\mathbf{z} \mid \mathbf{h}) \mid \mathbf{y}, \mathbf{h}^{[i-1]}) = \\
= \sum_{x \in X} [p(\mathbf{x} \mid \mathbf{y}, \mathbf{h}^{[i-1]}) \log p(\mathbf{y}, \mathbf{x} \mid \mathbf{h}^{[i]})]
\end{aligned}
\tag{5}
$$

where the set X consist of all possible x (constellation set used by the transmitter), $\mathbf{h}^{[i]}$ is the current set of parameters to estimate, $\mathbf{h}^{[i-1]}$ is the set obtained in the previous iteration, and $p(\mathbf{x} \mid \mathbf{y}, \mathbf{h}^{[i-1]})$ are the a posteriori probability of the symbol sequence $\mathbf{x}$ computed from the previous decoding iteration, in which $\mathbf{y}$ and $\mathbf{h}^{[i-1]}$ were used.

3.2.2 Maximization step

This step is carried out by maximizing (8) since the noise vector $\mathbf{n}$ is Gaussian and assumed i.i.d,

$$
\mathbf{h}^{[i]} = \min_h \sum_{x \in X} p(\mathbf{x} \mid \mathbf{y}, \mathbf{h}^{[i-1]}) \parallel \mathbf{y} - diag(\mathbf{x})\mathbf{h}^{[i]} \parallel^2
\tag{6}
$$

This equation is a weigthed Least Squares problem (weighted-sum objective) ,similar to (??) where received ($\mathbf{y}$) and sent data $\mathbf{x}$ were known, except in this case instead of $\mathbf{x}$ we must express it as a function of its expectation. So the solution is given by,

$$
\begin{aligned}
h^{[i]} & = E(\mathbf{x}^*) \cdot E(\mid \mathbf{x} \mid^2)^{-1} \cdot \mathbf{y} \\
& = \left(\sum_{x \in X} P(\mathbf{x}) \cdot diag(\mathbf{x}^*) \right) \left(\sum_{x \in X} P(\mathbf{x}) \cdot diag(\mid \mathbf{x} \mid^2) \right)^{-1} \mathbf{y}
\end{aligned}
\tag{7}
$$

where $P(\mathbf{x}) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{h}^{[i-1]})$.

4. REDUCED COMPLEXITY JOINT ITERATIVE CHANNEL ESTIMATION AND DECODING VIA THE EM ALGORITHM

We call reduced complexity a system which performs few iterations between the block that perfoms the EM Algorithm and the demodulator (see Fig.1 C). Extra iteration is performed after obtaining the APP of the bits from the turbo decoder (see Fig.1 B) , then instead of calculating the channel estimates and forward them to the demodulator for obtaining new metrics and send them to the turbo code (see Fig.1 A), several iterations between the demodulator and the EM estimation block are carried out (see Fig.1 B). The main goal of these extra iteration, which is also called EM iteration, is to get better channel estimates and as a consequence, better metrics to send to the turbo decoder and avoid performing few decoding iterations which have high computational load. This proposed approach reduces the complexity because of performing less decoding iterations which also reduces the latency at the receiver, while obtaining good performance.

4.1 EM iteration

Basically we can summarize the EM iteration in two steps, one carried out by the demodulator and the other computed via the EM Algorithm, which they result, respectively, in refining the bit metrics and the channel estimates.

4.1.1 Refinement of the metrics of the bits of every symbol received

In this step we compute the metrics of the bits of every symbol received using the channel estimates $\mathbf{h}^{[j-1]}$ obtained via the EM Algorithm in the previous iteration. We calculate $\hat{p}'(\mathbf{x} \mid \mathbf{y}, \mathbf{h}_i^{[j-1]})$.

The demodulator updates the term $\lambda(\mathbf{y},\mathbf{x} \mid \hat{\mathbf{h}}_i^{[j]})$ in the EM iterative process, which is equivalent to $\log p(\mathbf{y},\mathbf{x} \mid \hat{\mathbf{h}}_i^{[j]})$. In presence of i.i.d Gaussian noise it is defined by,

$$\lambda(\mathbf{y},\mathbf{x} \mid \hat{\mathbf{h}}_i^{[j]}) = \log\left[\frac{\exp\left(\frac{-\|\mathbf{y}-diag(\mathbf{x})\hat{\mathbf{h}}_i^{[j-1]}\|^2}{2\sigma_n^2}\right)}{\left(\sqrt{2\pi\sigma_n^2}\right)^N}\right] \quad (8)$$

where N is the length of the received sequence, i is the index for the decoding iteration and j is the index of the EM iteration.

The operation computed by the demodulator in order to calculate the log probabilities for every channel symbol bit c^i is expressed as,

$$\lambda'(c^i = d) = \max_{x \in X_d^i}\left(\lambda(x,y \mid \hat{h}_i^{[j]}) + \sum_{j\neq i}\lambda(c^j = c^j(x))\right) \quad (9)$$

where $d = 0,1$.

4.1.2 Refinement of the channel estimates

In the second step we compute a new set of channel estimates, $\hat{\mathbf{h}}_i^{[j]}$. The first EM channel estimation is performed in the same way as in the conventional receiver, from the sequence $\mathbf{y}$ and the a priori probabilities on the transmitted symbols. In the following EM iterations, the operation performed at the EM channel estimator to maximize the expectation of the Likelihood function is given by,

$$\hat{\mathbf{h}}_i^{[j]} = \max_h E_{x \in X}(\log \hat{p}'(\mathbf{z} \mid \mathbf{h}) \mid \mathbf{y},\hat{\mathbf{h}}_i^{[j-1]}) \quad (10)$$

The vector $\hat{\mathbf{h}}_i^{[j]}$ corresponds to the vector of channel estimates over an OFDM symbol in the decoding iteration i and EM iteration j. The expectation function shown in (14) is resolved like in (8) but using $\hat{p}'(\mathbf{x} \mid \mathbf{y},\mathbf{h}^{[j]})$, which are the result of the new metrics computed at the demodulator in the current EM iteration, after transforming bit probabilities into symbol probabilities and convert to linear values.

4.2 Complexity analisys comparison

The study of complexity depends on the processor it is using, because the weight of every operation is different if one use a DSP, FPGA or ASIC implementation. A detailed complexity evaluation for each block at the receiver is given in terms of FLOP in [8]. The number of operations calculated during a decoder iteration based on a single EM channel estimation is expressed as,

$$N_{dec+Em} = N_d \cdot k(S_{tc} \cdot 40 + 62 + 2 \cdot Q) + N_p(Q-1) \\ + N_d \cdot 7 \cdot (Q+1) + 10 \cdot N \cdot log_2 N \quad (11)$$

where N is the number of subcarriers, N_d is the number of subcarriers carrying data symbols, Q is the size of the constellation, k is the number of bits per data symbol and S_{tc} is the number of states in the RSC encoder. The complexity of the Turbo Decoder is mainly limited by the number of iterations performed. The number of operations carried out in a EM iteration is,

$$N_{EM_{iter}} = 7 \cdot N_d \cdot (1+Q) + N_p(Q-1) + 10 \cdot N \log_2 N \\ + 2 \cdot N_d \cdot k(18+Q) \quad (12)$$

comparing with (16) one can conclude the number of operation needed for a decoding operation. In order to understand better the complexity of both systems we calculate in terms of MFLOP (millions of floating points) the operations performed for both of them. The parameters that we use are the same parameters used in the simulations: $N = 256$, $N_p = 16$, $N_d = 240$, $S_{tc} = 4$, $Q = 4$,

Table 1: Complexity Analysis

N_{op}^{Rx1}	N_{op}^{Rx2}	N_{op}^{Rx3}	N_{op}^{Rx4}
0.139 MFLOP	0.228	0.317	0.18

$k = 2$. We compute the number of MFLOP for the follow different receivers:

Receiver 1: performs 1EM and 1 decoding iterations.
Receiver 2: performs 1EM and 2 decoding iterations.
Receiver 3: performs 1EM and 3 decoding iterations.
Receiver 4: performs 2EM and 1 decoding iterations.

The result with the number of operations of each receiver is shown in Table 1 from which we note in that scenario **Receiver 4** (our proposed scheme) requires 22% more operations than **Receiver 1**, 27% less operations than **Receiver 2** and 77% less than **Receiver 3**.

We also make a complexity comparison between our proposed receiver (System B: 2EM iterations and 1 decoding iteration using turbo decoding) and the system used in [2] (System A). At the transmitter, in system A, a 16-state rate $\frac{1}{2}$ convolutional code with generators [37 21] is used. We use the same parameters for the complexity analisys as before, except the number of states of the convolutional code is $S_{cc} = 16$.

5. SIMULATIONS

The simulated system uses Turbo Code-OFDM with $N = 256$ subcarriers, out of which only 16 of them are pilot subcarriers.The sample rate is 16 MHz which corresponds to a sample time of $T_s = 0.0625\mu s$. In the simulated system we use a Cyclic Prefix length of 65 samples, and OFDM symbol duration is $T_{OFDM} = 20\mu s$. This Cyclic Prefix accounts for up to $4\mu s$ maximum delay spread. The channel model is the WSSUS channel described in Section II.The recursive systematic convolutional encoder in the Turbo Code is rate $\frac{1}{2}$ with constraint length $K = 3$, and generator polinomial $G_R(D) = \{7,5\}$.

Fig. 3 depicts the channel refination quality of the proposed reduced complexity scheme for a different number of EM iterations. As it can be seen, with every extra EM iteration we get more accurate channel estimates.From this figure we can notice that as the number of decoding iterations increase the improvement obtained from one extra EM iteration became smaller.

A comparison between the System A [2] and System B (our proposed scheme with 2EM iterations and 1 decoding iteration) is shown in Fig.4. System B outperfoms System A in all the cases, obtaining a great improvement. This comparison is made performing the same number of EM Iterations in both systems and varying the number of decoding iterations in System A. We compare our proposed scheme with a scheme that has 18% less computational load (System A with 3 decoding iterations) , with another with similar number of operations (System A with 4 decoding iterations) and finally with the System A performing 20% percentil more of calculations (i.e System A performing 5 decoding iterations), achieving in all the scenarios a considerable better performance.

Fig. 2(a) and Fig. 2(b) show the performance of the reduced-complexity (2EM iterations) with n ($n = 1,2$) decoding iterations, compare with the conventional receiver (1 EM iteration) with n, $n+1$ and $n+2$ decoding iterations in terms of BER (in Fig. 2(a) $n = 1$ and Fig. 2(b) $n = 2$). Observing the BER performance in both figures we can notice that the perfomance of the proposed scheme is better than the performance of the conventional scheme for the different number of decoding iterations. We obtain a solid improvement in the performance when we compare with the conventional scheme with the same number of decoding iterations. In particu-

(a) Case I	(b) Case II
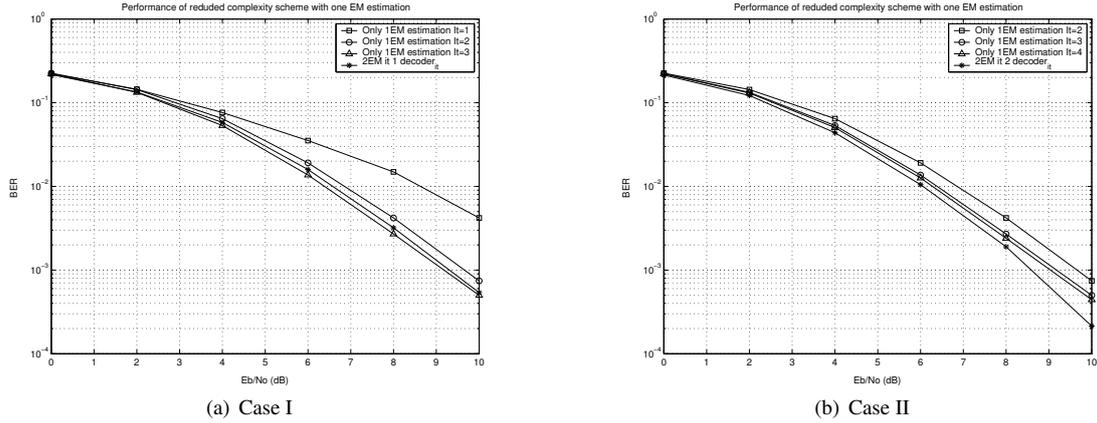
Figure 2: Ber performance proposed scheme for different number of decoding iterations

lar the BER is better than the receiver that requires 27% more calculations (**Receiver 2**) and comparable with the one that performs 77% more calculations (**Receiver 3**) (see complexity analysis Section III). Moreover, we can notice that this improvement is larger when a small number of decoding iterations are carried out. When the number of decoding iterations increase the improvement that we obtain with the proposed scheme becames smaller.
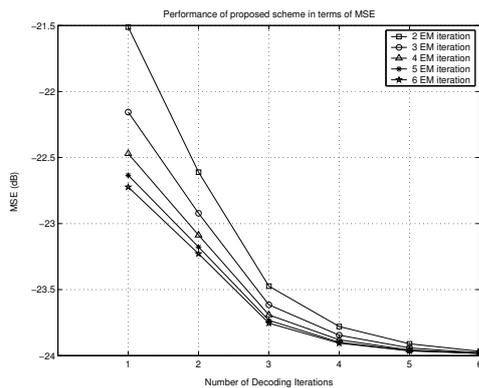
Figure 3: MSE for different number of EM iterations
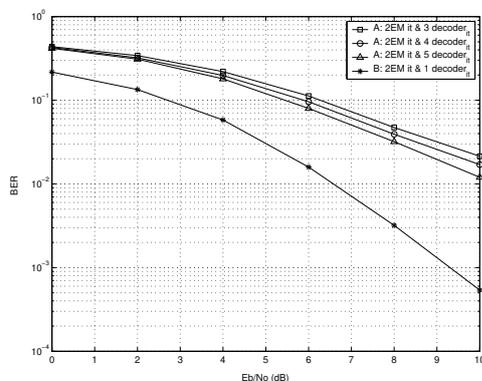
Figure 4: Comparison between System A and System B

6. CONCLUSION

We have introduced and evaluated a reduced-complexity receiver based in TC-OFDM system (i.e the one which uses more EM iteration estimation), and its performance was compared with the conventional receiver (the one that uses only one iteration in channel estimation). A new iteration is performed to improve the channel estimates and get better metrics , before sending them to the Turbo decoder. Iterating in that way, we avoid the use of decoding iterations, that have higher computational load than the EM iteration, obtaining the same performance or in some cases even better with such reduced complexity.

The proposed receiver outperfoms the conventional receiver in terms of BER in all the situations, being this improvement smaller as the number of decoding iterarions increase. In particular such approach is suitable for applications that require low latency.

Furthermore, as expected , the propose scheme outperforms a similar set up [2], which does use turbo decoding, though with comparible complexity.

REFERENCES

[1] C. Berrou and A. Glavieux , "Near optimum error correcting coding and decoding : Turbo Codes", *IEEE Transactions on Communications*, October 1996.

[2] R. De Francisco and Y. Bar-Ness,"A novel reduced-complexity EM based receiver with joint iterative channel estimation and decoding", *41st Annual Allerton Conference*, October 2003.

[3] G. Ungerboeck, "Channel coding with multilevel/phase signals", *IEEE Trans. Inform. Theory*, January 1982

[4] G. Caire, G. Taricco and E. Biglieri, "Bit-interleaved coded modulation", *IEEE Trans. Inform. Theory*, May 1998

[5] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding", *Proc. of ICC'99*, June 1999.

[6] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "A soft-input soft-output maximum a posteriori (MAP) module to decode parallel and serial concatenated codes", *TDA Progress* , November 1996.

[7] P. Hoeher, "A statistical discrete-time model for WSSUS multipath channel", *IEEE Trans.Veh. Tech*, November 1992.

[8] M. Perez, "Improved OFDM receiver for joint iterative channel estimation and decoding employing turbo code", *MS thesis, Universitat Politecnica de Catalunya*,July 2004.

[9] T.K. Moon, "The expectation-maximization algorithm" *Signal Processing Magazine*, November 1996.