Iterative Channel Estimation and Decoding for the Multi-path Channels with RAKE reception

Naresh Sharma, *Member, IEEE,* and Alexei Ashikhmin, *Member, IEEE*

***Abstract*— We study an improved receiver with iterative channel estimation and decoding for multi-path channels with RAKE reception. A soft modulator processes the soft output of the Turbo decoder after a decoding iteration. If some symbols are deemed reliable at the output of the soft modulator, then these symbols act as *extra* pilot symbols for the channel estimator to refine the channel estimates that aid the demodulator, whose output is given to the Turbo decoder for another decoding iteration. The conventional Turbo decoder needs to be modified to yield soft information for both the systematic as well as the parity bits and two methods are presented to accomplish this. To keep the complexity in terms of memory and processing low, the iterative channel estimation is done on the *equivalent channel* at the output of the RAKE combiner instead of per-finger iterative channel estimation before the RAKE combining. Numerical results are presented that show that good improvement is possible over the conventional receiver.**

I. Introduction

There has been a significant interest in the last few years to improve the capacity of 3G (third generation) systems. For CDMA 2000 systems, much of the work has focussed on the Forward Link (FL) over the last few years. Reverse Link (RL) that has lower average throughput, has received renewed focus in recent times.

A limiting factor for the RL throughput is the multi-access interference (MAI) that results in increased noise as seen by a given user due to the signals from other users since there is no orthogonality between users. MAI is caused by the traffic and control channels (including the pilot channel) of other users. Pilot channel of a given user also interferes with its own traffic channel due to multi-path distortions. Even if there are few active users transmitting data at a given time, the pilot channels (that are power-controlled) of other users may cause significant interference to adversely affect the throughput. Total power of the pilot channels when summed over all the users is a significant fraction of the total received power. If a receiver can work with lower pilot values or with smaller traffic power for the same pilot power, then it can reduce the MAI and have a positive effect on the system throughput.

In this paper, we study an improved receiver that iteratively refines the channel estimate based on the soft output of the decoder at the end of a decoding iteration. This refined estimate is given to the demodulator that gives its soft output as an input to the Turbo decoder with higher *quality*. Thus both the decoder and the channel estimator help each other. This iterative process is repeated a pre-specified number of times to yield a better reception of a user signal as compared to a receiver that estimates the channel only once followed by decoding. Our focus is essentially to have an efficient reception of the signal of a given user. We don't resort to any multi-user detection schemes in this work.

Joint channel estimation and decoding has been proposed before in [11]–[14]. One of the differences is that multi-path is not considered in the previous work. We assume that the RAKE receiver is employed and the weights to do maximal ratio combining (MRC) across the fingers are determined by the pilot.

Commonly used algorithms for Turbo decoding compute log-likelihood ratios (LLR) of only the systematic bits. However, one needs LLR of both the systematic as well as the parity bits for iterative decoding and channel estimation. We provide two methods to accomplish this. The first method computes the LLR of the parity bits by soft encoding the extrinsic information of the systematic bits as given by the Turbo decoder. The second method computes the extrinsic information of parity bits by comparing the probabilities of state transitions when the chosen parity bit is zero and one. The soft information of all the coded bits is then converted into the probabilities of each modulation symbol (QPSK for example) by a soft modulator. If the symbol is deemed reliable as can be inferred from the probabilities, then it is *assumed* that it is the transmitted symbol. If there are many such reliable symbols after a decoding iteration, then one could estimate the channel through these *extra pilot symbols*.

The channel estimation should ideally be done on a per-finger basis so that one refines the channel estimates for all the fingers before they are combined. However, this has larger memory requirements making the implementation difficult. To keep the memory requirements small, we estimate the *equivalent* channel at the output of the RAKE combiner.

II. System Model and RAKE reception

In the section, we briefly describe the system model and the RAKE receiver for CDMA 2000 systems.

A. Turbo code and channel models

The Turbo code employed in CDMA 2000 systems has two constituent parallel concatenated recursive convolutional codes that have a constraint length of 3. The transfer function for each convolutional code is given by

$$G(D) = \left[1 \ \frac{n_0(D)}{d(D)} \ \frac{n_1(D)}{d(D)}\right],$$

where $d(D) = 1 + D^2 + D^3$, $n_0(D) = 1 + D + D^3$, and $n_1(D) = 1 + D + D^2 + D^3$. There are 3 tail bits added to the systematic bits for each constituent convolutional encoder

N. Sharma is with the Open Innovations Laboratory, Lucent Technologies, 67 Whippany Rd., Whippany, NJ 07981 USA (nareshs@lucent.com). A. Ashikhmin is with the Bell Labs, Lucent Technologies, 600 Mountain Ave., Murray Hill, NJ 07974 USA (aea@lucent.com).

to take the state of each constituent convolutional code back to the all zero state. For example, for information size of 378 bits with code rate of $1/2$, the number of coded bits are $2 \times (378+6) = 768$. Turbo inter-leavers for different packet sizes are specified. Since this code has rate of $1/5$ (ignoring the tail bits and 1 branch of systematic bits), a puncturer is needed to generate rates higher than $1/5$.

The multi-path channel models used for evaluating CDMA 2000 systems are given in terms of the number of multi-paths, and the delay and average power of each path, which is modeled as Rician or Rayleigh. The channel model is obtained after convolving the impulse response with the transmit and receive pulse shaping filters. Some paths are not resolvable and the energy of these paths is quantified as Fractionally Unrecovered Power (FURP) and this contributes to the noise. As an example, a channel model called 'Model B' has three resolvable Rayleigh paths with average powers as -1.64dB, -7.8dB, -11.7dB with delays of 0.0, 1.23, 2.83 respectively in multiples of chip time period ($= 813.8$ nano seconds) respectively with FURP of -10.92dB.

B. RAKE Reception

We assume that each mobile has a single antenna. Let there be L fingers for a given channel that can be resolved at the receiver and N receive base-station antennas. The received signal model for the ith finger, jth chip and nth receive antenna is given by

$$r_{i,n}^{j} = \sqrt{E_c} h_{i,n}^{j} c_j + \sqrt{E_c + E_{c,\text{pilot}}} \times \sum_{\substack{k=1 \\ k \neq i}}^{L} h_{k,n}^{j} c_{j+\delta_i-\delta_k} + \nu_{i,n}^{j}, \quad (1)$$

where c_j is the transmitted signal for the jth chip, δ_i is the delay for the ith finger that is assumed to be the same across all the receive antennas, E_c is the traffic chip energy, $E_{c,\text{pilot}}$ is the pilot chip energy, $h_{i,n}^{j}$, $\nu_{i,n}^{j}$ are the channel gain and noise for the ith finger, jth chip and nth receive antenna respectively. Note that the noise includes contribution due to the thermal noise as well as the FURP and let N_0 be the variance of zero mean complex circularly symmetric Gaussian thermal noise $\nu_{i,n}^{j}$. Let us assume that $\hat{h}_{i,n}^{j}$ is the estimate of the $h_{i,n}^{j}$ that is obtained after processing the pilot channel(s). The RAKE combiner does a MRC across all the fingers though with corrupt channel estimates, to get the statistic

$$\begin{aligned} R_j &= \sqrt{E_c} \sum_{n=1}^{N} \sum_{i=1}^{L} r_{i,n}^{j} \left(\hat{h}_{i,n}^{j}\right)^{*} \\ &= \sqrt{E_c} \left[\sum_{n=1}^{N} \sum_{i=1}^{L} h_{i,n}^{j} \left(\hat{h}_{i,n}^{j}\right)^{*} \right] c_j + \mu^{j}, \end{aligned} \quad (2)$$

where x^* denotes the conjugate of x, μ_j is the noise at the output of the RAKE combiner and is complex, zero mean with

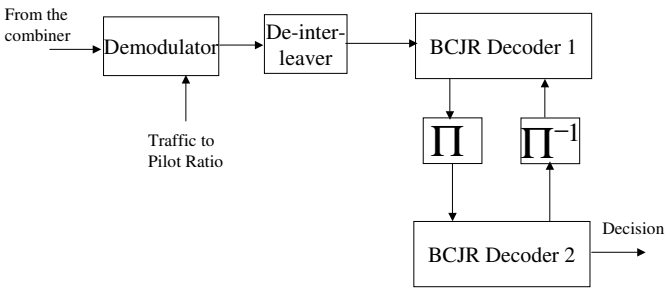


Fig. 1. Conventional receiver with single shot channel estimation and decoding.

the variance

$$\sigma_j^2 = \sum_{n=1}^{N} \sum_{i=1}^{L} \mid \hat{h}_{i,n}^{j} \mid^2 \times \left\{ (E_c + E_{c,\text{pilot}}) \left[\sum_{\substack{k=1 \\ k \neq i}}^{L} \mid h_{k,n}^{j} \mid^2 + \mid \lambda_n^{j} \mid^2 \right] + N_0 \right\},$$

where λ_n^{j} is the finger that models FURP for the nth receive antenna for the jth chip.

Assuming that the channel and the channel estimation are constant over all the chips making up the symbol, the SNR for the symbol s containing the chip j is given by

$$\left(\frac{E_s}{N_t}\right)_s = K_{\text{pg}} \frac{E_c}{\sigma_j^2} \left| \sum_{n=1}^{N} \sum_{i=1}^{L} h_{i,n}^{j} \left(\hat{h}_{i,n}^{j}\right)^{*} \right|^2, \quad (3)$$

where K_{pg} is the Processing Gain, E_s is the symbol energy after de-spreading and N_t is the noise energy as seen by a symbol.

Note that the *equivalent* channel at the output of the RAKE combiner denoted by γ_j is given by

$$\gamma_j = \sum_{n=1}^{N} \sum_{i=1}^{L} h_{i,n}^{j} \left(\hat{h}_{i,n}^{j}\right)^{*}. \quad (4)$$

The demodulator after the RAKE combiner *assumes* that the channel estimates are correct and the channel is given by

$$\hat{\gamma}_j = \sum_{n=1}^{N} \sum_{i=1}^{L} \left| \hat{h}_{i,n}^{j} \right|^2. \quad (5)$$

The statistic in (2) is given to the demodulator which demodulates it assuming that the channel was given by (5). The LLR of the bits is then de-interleaved with the channel de-interleaver and then given to the Turbo decoder for processing. The receiver is shown in Fig. 1.

Note that the channel estimator and the decoder are completely decoupled from each other. Channel estimation is done only once before the RAKE combining and the LLR at the output of the demodulator are corrupt because of noisy channel estimates. There is no effort to *refine* the channel estimates and this results in a degraded performance of the receiver.

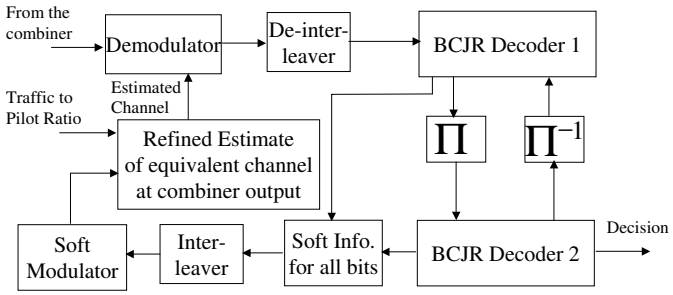


Fig. 2. Modified Receiver with iterative channel estimation (at the output of the RAKE combiner) and Turbo decoding.

III. Improved Receiver

Note that while the assumption is that the equivalent channel is a real number as seen in (5), the actual equivalent channel in (4) is complex in general. This leads to phase distortions that would result in demodulator making an erroneous assumption of a rotated constellation in relation to the true one.

Instead of a one-shot channel estimation, the LLR of coded bits from the decoder after each decoding iteration is processed and given to a soft modulator to compute the soft information of the modulated symbols. If a given modulation symbol is deemed as reliable as can be inferred from the probabilities, then one can assume that the modulation symbol is indeed the correct one. These reliable symbols are then used by the channel estimator as extra pilot symbols to aid better channel estimation, and then passed on to the demodulator, which then passes less corrupt LLR values to the decoder.

There can be two alternatives for iterative channel estimation. One option could be to refine the channel estimates on a per-finger basis and then redo RAKE combining in each decoding iteration. This however, requires storing the channel estimates of each of the fingers and then refining them with the help of Turbo decoder. This option is difficult to implement because of larger memory requirements since the per-user memory increase is equal to the number of fingers multiplied by the per-finger memory increase. We therefore, do not explore this option for the purposes of this paper.

Second option could be to refine the estimates of $\hat{\gamma}_j$ given by (5) i.e. estimating the noisy channel at the output of the RAKE receiver. The new receiver working on this second option is shown in Fig. 2.

Note that the commonly used version of the BCJR algorithm that is used for Turbo decoding outputs the LLR only for the systematic bits. Since the channel estimator works on the coded and modulated symbols, LLR for both systematic and parity bits are required. We get the LLR for parity bits by using two methods that are described below. Soft modulator that maps the coded bit probabilities into coded modulated symbol probabilities is also described below.

A. Convolutional Decoders

In this subsection, we consider three types of soft output decoders for a recursive systematic convolutional code. The first decoder computes soft output only for systematic bits of a codeword while the last two compute the soft output for all the coded bits.

Consider a time invariant convolutional code that encodes h bits of information into n coded bits at a given time instant. Let r denote the constraint length of the code. For the chosen Turbo code, we have $h = 1$, $n = 3$, and $r = 3$. Since the code is terminated at zero state, one has to append rh bits to $(q-r)h$ information bits to take the trellis state back to zero. The total number of output bits is qn.

Let $\underline{u}_{t,s} = (u_1, u_2, \ldots, u_n)$ be the binary vector corresponding to the trellis branch that connects states t and s. We assume without loss of generality that the first h bits $u_1, \ldots, u_h$ are systematic bits.

Let m denote the number of states and $\{0, \ldots, m-1\}$ be the set of states of the trellis associated with the convolutional code. Let $P(s_k)$ be the set of states at instance $k-1$ that are connected to s_k by trellis branches. $F(s_k)$ is the set of states at instance $k+1$ to which the state s_k is connected by trellis branches. In the case of time invariant convolutional codes $P(s_k)$ and $F(s_k)$ are the same for any k. For this reason, we will drop the subscript k and use $P(s)$ and $F(s)$.

Let $L_i^{k,x}$ and $L_o^{k,y}$ denote the input and output LLR of bits at instant k and positions x and y respectively. LLR is defined as the logarithm of the ratio of the probability that the bit is 0 to the probability that the bit is 1. Define a function $\gamma_{t,s}$ as follows

$$\gamma_{t,s} = f(u_1)f(u_2)\cdots f(u_n),$$

where

$$f(b) = \begin{cases} 1/\left[1+\exp(-L(b))\right], & \text{if } b = 0 \\ 1/\left[1+\exp(L(b))\right], & \text{if } b = 1, \end{cases} \tag{6}$$

where $L(b)$ is the LLR of bit b. Let $(S_1, T)_j$ denote the set of pairs of states s and t such that in the vector $\underline{u}_{s,t}$ the j-th bit equals 1 and let $\overline{(S_1,T)_j}$ be defined similarly for the case when the j-th bit equals 0. The first decoder describes the commonly used BCJR decoder [2].

$\underline{\text{Dec}_1\left[L_i^{1,1}, \ldots, L_i^{q,n}; L_o^{1,1}, \ldots L_o^{q,h}\right]}$

1) Initialization: $\alpha_1(0) = 1, \alpha_1(s) = 0 \ \forall \ s \neq 0$; $\beta_q(0) = 1, \beta_q(s) = 0 \ \forall \ s \neq 0$;
2) For k from 2 to q do
   - For s from 0 to $m-1$ do
   $$\alpha_k(s) = \sum_{t \in P(s)} \alpha_{k-1}(t)\gamma_{t,s}.$$
   - For s from 0 to $m-1$ do
   $$\beta_{q-k+1}(s) = \sum_{t \in F(s)} \beta_{q-k+2}(t)\gamma_{s,t}.$$
3) For k from 1 to q do
   For j from 1 to h do
   $$L_o^{k,j} = \log\left[\frac{\sum_{(s,t)\in(S_1,T)_j} \alpha_k(s)\gamma_{s,t}\beta_t}{\sum_{(s,t)\in\overline{(S_1,T)_j}} \alpha_k(s)\gamma_{s,t}\beta_t}\right].$$

$\underline{\text{Dec}_2\left[L_i^{1,1},\ldots,L_i^{q,n};L_o^{1,1},\ldots L_o^{q,n}\right]}$

This algorithm computes LLR for all code bits. It differs from Dec_1 only at step 3. For Dec_2 the step 3 has the following form:

3) For k from 1 to q do
For j from 1 to n do

$$L_o^{k,j}=\log\left[\frac{\sum_{(s,t)\in(S_1,T)_j}\alpha_k(s)\gamma_{s,t}\beta_t}{\sum_{(s,t)\in\overline{(S_1,T)_j}}\alpha_k(s)\gamma_{s,t}\beta_t}\right].$$

$\underline{\text{Dec}_3\left[L_i^{1,1},\ldots,L_i^{q,n};L_o^{1,1},\ldots L_o^{q,n}\right]}$

This algorithm computes LLR for all coded bits. Dec_1 is a subset of this decoder and is followed by a soft encoder that maps the LLR of the systematic bits to LLR of the parity bits. Soft encoder is separate from the Dec_1 block that is unchanged.

Let $(S,\underline{U})_j$ denote the set of pairs of the state s and h-length binary input vectors $\underline{u}_{s,j}=(u_1,u_2,\ldots,u_h)$ such that the lth ($l=h+1,\cdots,n$) output parity bit is 0. $\overline{(S,\underline{U})_j}$ is similarly defined when the chosen output parity bit is 1. Further, we will assume that the LLR of the jth element of h-length input vector is given by $L_o^{k,j}$, i.e. the extrinsic information obtained by Dec_1 is the input to the encoder. Let $p_k(s)$ denote the probability of a state s at time instant k.

The output LLR are computed with the following additional step:

4) For k from 1 to q do
For j from $(h+1)$ to n do

$$L_o^{k,j}=\log\left[\frac{\sum_{(s,\underline{u}_{s,j})\in(S,\underline{U})_j}p_k(s)G(\underline{u})}{\sum_{(s,\underline{u}_{s,j})\in\overline{(S,\underline{U})_j}}p_k(s)G(\underline{u})}\right],$$

where $G(\underline{u})=\prod_{l=1}^{h}f(u_l)$ where $f(u_l)$ is a function that is given by (6). This simplifies greatly when $h=1$. For $h=1$, as in present case, the probability of states at time $k+1$ is given by

$$p_{k+1}(s)=p_k(g_0(s))f(b)+p_k(g_1(s))[1-f(b)],$$

where $g_j(s)$ denotes the current state that results in a new state s when input bit is j, $j\in\{0,1\}$. A normalization step is carried out to make

$$\sum_{s=0}^{m-1}p_{k+1}(s)=1.$$

B. Soft Modulator

Soft modulator determines the probability of symbols from the probability of bits that make up the symbols. If c denotes a symbol drawn from a constellation $\mathcal{C}$ of size 2^q, and whose bit representation is given by $b_1,\cdots,b_q$, where b_i denotes the bit in the ith position. Then the probability that the transmitted symbol, denoted by x, is equal to c, is given by

$$\text{Prob}\{x=c\}=G\prod_{i=1}^{q}p_i, \tag{7}$$

where p_i denotes the probability of the ith bit and G is a normalization constant chosen such that

$$\sum_{c\in\mathcal{C}}\text{Prob}\{x=c\}=1.$$

C. Iterative Decoders of Turbo Codes

In this subsection, two iterative decoders are provided. The first decoder is the conventional decoder that computes the soft output of only the systematic bits and this information is exchanged between the decoders of two constituent convolutional codes. The second decoder computes the soft output of both systematic and parity bits to aid the channel estimator and both the channel estimator and the Turbo decoder synergistically help each other.

Let the received LLR for the two convolutional decoders after appropriate de-interleaving, de-puncturing, combining across the repetitions if any, and demodulating are given by

$$\underline{y}_1=(L_{i,1}^{1,1},\ldots,L_{i,2}^{q,n})$$

and

$$\underline{y}_2=(L_{i,2}^{1,1},\ldots,L_{i,2}^{q,n}).$$

$\underline{\text{Turbo Decoder}_1\left[\underline{y}_1,\underline{y}_2\ ;\ L_o^{1,1},\ldots,L_o^{q-r,h}\right]}$

1) Initialization:

$$\left[L_o^{1,1},\ldots,L_o^{q-r,h}\right]=\left[L_{i,1}^{1,1},\ldots,L_{i,1}^{q-r,h}\right].$$

2) Repeat steps 3 to 6 M times:
3) $\text{Dec}_1\big[L_o^{1,1},\ldots,L_o^{q-r,h},L_{i,1}^{q-r,h+1},\ldots L_{i,1}^{q,n}\ ;$
$L_o^{1,1},\ldots,L_o^{q,h}\big]$;
4) Interleaver $\left[L_o^{1,1},\ldots,L_o^{q-r,h}\ ;\ L_o^{1,1},\ldots,L_o^{q-r,h}\right]$;
5) $\text{Dec}_1\big[L_o^{1,1},\ldots,L_o^{q-r,h},L_{i,2}^{q-r,h+1},\ldots,L_{i,2}^{q,n}\ ;$
$L_o^{1,1},\ldots,L_o^{q,h}\big]$;
6) Deinterleaver $\left[L_o^{1,1},\ldots,L_o^{q-r,h}\ ;\ L_o^{1,1},\ldots,L_o^{q-r,h}\right]$;

The second decoder uses the algorithms Dec_2 or Dec_3 (denoted by $\text{Dec}_{2,3}$) for decoding of the convolutional codes and alternates decoding of convolutional codes with obtaining new estimates of channel.

$\underline{\text{Turbo Decoder}_2\left[\underline{y}_1,\underline{y}_2\ ;\ L_o^{1},\ldots,L_o^{(q-r)h}\right]}$

1) Repeat steps 1 to 9 M times
2) Initialization:

$$\left[L_{o,1}^{1,1},\ldots,L_{o,1}^{q-r,h}\right]=\left[L_{i,1}^{1,1},\ldots,L_{i,1}^{q-r,h}\right].$$

3) Repeat steps 4 to 7 N times:
4) $\text{Dec}_{2,3}\big[L_{o,1}^{1,1},\ldots,L_{o,1}^{q-r,h},L_{i,1}^{q-r,h+1},\ldots L_{i,1}^{q,n}\ ;$
$L_{o,1}^{1,1},\ldots,L_{o,1}^{q,n}\big]$;
5) Interleaver $\left[L_{o,1}^{1,1},\ldots,L_{o,1}^{q-r,h}\ ;\ L_{o,2}^{1,1},\ldots,L_{o,2}^{q-r,h}\right]$;
6) $\text{Dec}_{2,3}\big[L_{o,2}^{1,1},\ldots,L_{o,2}^{q-r,h},L_{i,2}^{q-r,h+1},\ldots,L_{i,2}^{q,n}\ ;$
$L_{o,2}^{1,1},\ldots,L_{o,2}^{q,n}\big]$;
7) Interleaver $\left[L_{o,2}^{1,1},\ldots,L_{o,2}^{q-r,h}\ ;\ L_{o,1}^{1,1},\ldots,L_{o,1}^{q-r,h}\right]$;

8) Make channel packet

$$\left[L_{o,1}^{1,1},\ldots,L_{o,1}^{q,n}\ ;\ L_{o,2}^{1,1},\ldots,L_{o,2}^{q,n}\ ;\ \text{Channel Packet}\right];$$

9) Channel-Estimation $\left[\text{Channel Packet}\ ;\ \underline{y}_1\ ;\ \underline{y}_2\right]$;

IV. Numerical Results

In this section, we show the numerical results for both the conventional and the new receiver in terms of Frame Error Rate (FER) versus the traffic E_c/N_t for chosen values of $E_{c,\text{pilot}}/N_t$. We assume that $N = 2$ i.e. there are two receiver antennas at the base-station and the simulation parameters are chosen for the CDMA 2000 DV (data and voice) system.

A block of 1536 information bits are encoded by a Turbo code of rate $2/3$. QPSK (quadrature phase shift keying) modulation is used and the spreading factor is $4/3$. The fractional spreading factor could be obtained by using 3 out of 4 Walsh codes of length 4. The total transmission time is 1 slot or 1.25ms that gives a data rate of 1.2288 Mbps. In Fig. 3, we plot the FER versus traffic E_c/N_t for Model B that is described in Sec. II-A with $E_{c,\text{pilot}}/N_t = -20$ dB. The figure shows that there is a performance improvement of 1 dB at 4% FER. We also plot two lower bounds by assuming *genie knowledge*. The first lower bound is for the case when the demodulator knows the equivalent channel at the output of the combiner as given in (4) i.e. $\hat{\gamma}_j = \gamma$ for this case. This is a tight lower bound on the performance of the new receiver since it tries to estimate the *equivalent* channel and the lower bound is quite close to the actual performance. This indicates that any other strategy to do iterative channel estimation and decoding will only offer very marginal improvements.

The second lower bound is for the case when the RAKE receiver knows the channel perfectly and does the perfect combining across all fingers and receive antennas. This results in bigger improvement in performance. This curve is a lower bound for the iterative receiver that could perform per-finger channel estimation. While the memory requirements are larger for such a receiver, it is instructive to see how much performance improvement is possible. The large performance improvement is essentially because the combining at the RAKE receiver results in a loss of signal quality due to sub-optimal (and not necessarily constructive) combining of fingers in MRC.

V. Conclusions

We studied a modified receiver that estimates the equivalent channel at the output of the RAKE combiner by iterative channel estimation and soft decoding. We show that this receiver offers significant improvement over the current receiver. The new receiver has low memory and processing requirements. We also show that if per-finger channel estimation could be implemented by sorting out the memory and processing constraints, then larger improvements are available.

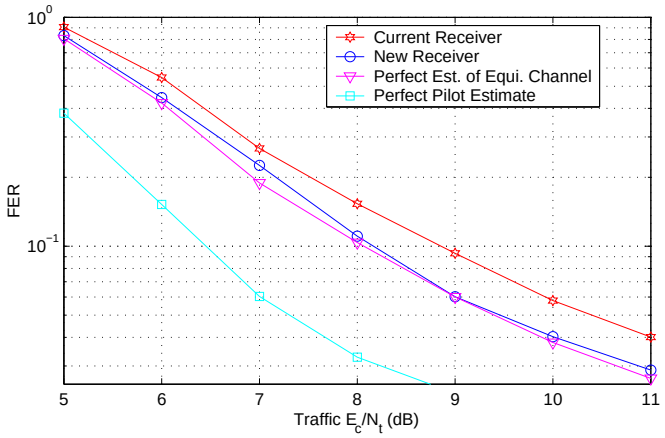


Fig. 3. FER versus Traffic E_c/N_t for 1.2288 Mbps with QPSK modulation, and $E_{c,\text{pilot}}/N_t = -20$ dB for Model B.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC'93*, pp. 1064-1070.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284-287, Mar 1974.

[3] E. K. Hall and S. G. Wilson, "Design and analysis of turbo codes on Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 160-174, Feb. 1998.

[4] M. C. Valenti and B. D. Woerner, "Refined channel estimation for coherent detection of turbo codes over flat-fading channels," Electron. Lett., vol. 34, pp. 1648-1649, Aug. 20, 1998.

[5] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429-445, Mar. 1996.

[6] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, pp. 927-945, May 1998.

[7] A. J. Goldsmith and P. P. Varaiya, "Capacity, mutual information, and coding for finite-state Markov channels," *IEEE Trans. Inform. Theory*, vol. 42, pp. 868-886, May 1996.

[8] J. K. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 686-693, Nov. 1991.

[9] J. G. Proakis, *Digital Communications*, 3rd ed. New York, NY: McGraw- Hill, 1995.

[10] W.C. Jakes, Jr., *Microwave Mobile Communications*. New York: Wiley, 1974.

[11] M. C. Valenti and B. D. Woerner, "Iterative channel estimation and decoding of pilot symbol assisted turbo codes over flat-fading channels," vol. 19, pp. 1697-1705, *IEEE J. Select. Areas Commun.*, Sept. 2001.

[12] C. Komninakis and R. D. Wesel, "Joint iterative channel estimation and decoding in flat correlated Rayleigh fading," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1706-1717, Sept. 2001.

[13] H.-J. Su and E. Geraniotis, "Low-complexity joint channel estimation and decoding for pilot symbol-assisted modulation and multiple differential detection systems with correlated Rayleigh fading," *IEEE Trans. Commun.*, vol. 50, pp. 249-261, Feb. 2002.

[14] Y. Xu, H.-J. Su, and E. Geraniotis, "Pilot symbol assisted QAM with iterative filtering and turbo decoding over Rayleigh flat-fading channels," *Proc. IEEE Military Commun. Conf. (MILCOM)*, vol. 1, pp. 86-91, Nov. 1999.