Iterative Channel Estimation and Decoding with Product Codes in Multicarrier Systems

Frieder Sanzi, Stephan ten Brink
Institute of Telecommunications, University of Stuttgart, Germany
{sanzi,tenbrink}@inue.uni-stuttgart.de

Abstract

A two–dimensional channel estimation scheme for coherent detection of multicarrier modulated signals is presented which is based on the A Posteriori Probability (APP) calculation algorithm. An iterative estimation and decoding loop over inner estimator/product code and outer soft in/soft out APP decoder allows to further reduce the bit error rate. Heavily punctured recursive systematic convolutional codes are introduced as high–rate component codes of the inner product code. The proposed system outperforms the reference system which applies conventional two–dimensional Wiener filters for channel estimation.

1. Introduction

Channel estimation for multicarrier systems is typically performed by applying a two–dimensional FIR interpolation filter whose coefficients are based on the Wiener design criterion, as described e. g. in [1]. For this, pilot symbols are periodically inserted into the transmitted signal to allow for coherent detection at the receiver.

In [2] the authors propose a modified A Posteriori Probability calculator (APP or MAP algorithm, [3, 4]) for the purpose of joint channel estimation and differential detection of DPSK–signals in frequency non–selective (flat) Rayleigh fading channels. In this paper we extend the idea of APP estimation to the two–dimensional multicarrier scenario and combine it with an inner high–rate product code. The estimation of the two–dimensional channel transfer function is performed by a concatenation of two one–dimensional APP estimators in time and frequency direction respectively. For the product code, heavily punctured recursive systematic convolutional (RSC) codes serve as high–rate component codes with low decoding complexity. In addition to bit error rate (BER) charts, the performance of the iterative estimation and decoding loop is studied with the Extrinsic Information Transfer Chart technique (EXIT chart [5, 6]).

2. System Configuration

2.1. Transmitter

The signal from the binary source is convolutionally encoded, interleaved, and encoded a second time by an inner product code (Fig. 1). The implicit block interleaver represents the rearrangement of the systematic bits from time to frequency order prior to vertical encoding. After serial to parallel conversion, the binary symbols (BPSK) are modulated onto N orthogonal sub–carriers by an iFFT–block. Pilot symbols are periodically inserted to allow for coherent detection at the receiver.

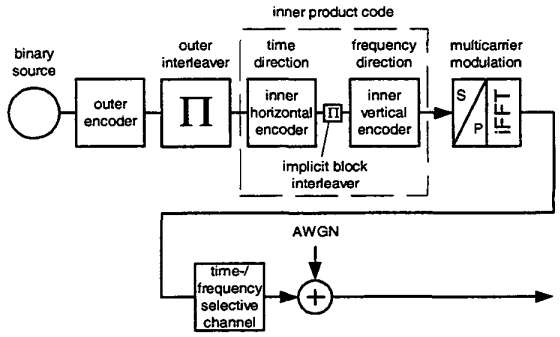

Figure 1. Transmitter and channel model.

2.2. Channel Model

For the mobile communication channel we apply the wide–sense stationary uncorrelated scattering (WSSUS) channel model as described in [7]. The Doppler shifts are distributed according to Jakes' power spectral density; the delays are exponentially distributed. A snapshot of the transfer function $H(t = kT_s, f = n\,\Delta f)$ is given in Fig. 2, with channel parameters as chosen in the simulations of Section 5.

0-7803-6507-0/00/$10.00 ©2000 IEEE

IEEE Vehicular Technology Conference (VTC), Boston, USA, Sept. 2000, S. 1338 - 1344

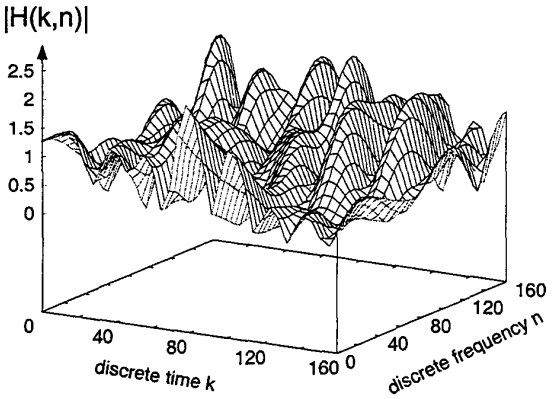

Figure 2. Snapshot of a channel transfer function; Doppler shift $f_{D,max} = 100\text{Hz}$, delay spread $\tau_{max} = 20\mu s$, symbol duration $T_s = 300\mu s$, sub–carrier spacing $\Delta f = 4\text{kHz}$.

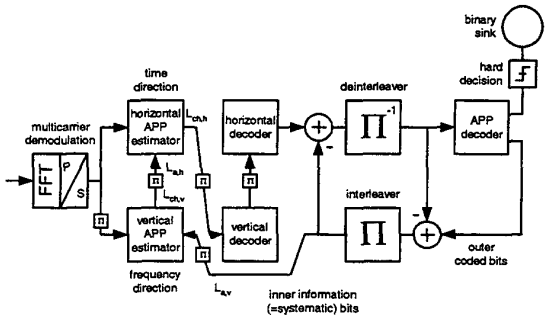

Figure 3. Receiver.

2.3. Receiver

After demodulation of the N sub-carriers and parallel to serial conversion, the signal is fed to the channel estimation and inner product decoding stage (Fig. 3). The estimation and inner decoding outputs log–likelihood ratios (L–values [8]) on the transmitted outer coded bits which are deinterleaved and decoded in an outer APP channel decoder. Iterative channel estimation and decoding is performed by feeding back extrinsic information on the outer coded bits; after re–interleaving it becomes the *a priori* knowledge to the inner APP estimation and product decoding stage. Note that the outer decoder provides new information only on the outer coded bits, which corresponds to *a priori* knowledge on the inner information (systematic) bits after re–interleaving. The *a priori* knowledge on the inner parity bits is set to zero. The extrinsic output of the outer decoder is exploited by both the APP estimators and the inner component decoders: For the L–values on the systematic bits, the additive structure of channel L–values and new inner horizontal/vertical extrinsic information is preserved such that the subtraction of outer extrinsic information (i. e. inner *a priori* knowledge $L_{a,v}$) needs to be performed only once, right after the vertical component decoder.

3. Channel Estimation

For APP estimation, the symbol–by–symbol MAP–algorithm according to [3] is applied to an appropriately chosen metric. For this, the transmitted coded bits can be thought of being put into a virtual shift register at the output of the transmitter. This "artifical grouping" allows to exploit the time and frequency continuity of the channel transfer function at the receiver by trellis processing. The received symbol from the channel at discrete time k, $0 \le k < K$ and discrete frequency (i. e. sub–carrier index) n, $0 \le n < N$ is $z_{k,n} = a_{k,n} \cdot x_{k,n} + n_{k,n}$, with $a_{k,n} = H(k,n)$ being the complex fading coefficient, x being the transmitted binary symbol $x \in \{\pm 1\}$ and n the complex additive white Gaussian noise with component–wise noise power $\sigma_n^2 = N_0/2$ (double–sided noise power spectral density).

At frequency index n (time index k omitted), the logarithmic metric increment for the vertical APP estimation (frequency direction, BPSK–modulated signals) can be simplified to

$$\gamma_n = \frac{\text{Re}\left\{z_n \cdot \hat{x}_n \cdot \hat{a}_{v,n}^*\right\}}{\sigma_{v,pred}^2} + \tfrac{1}{2} L_{a,v,n} \qquad (1)$$

with estimated channel coefficient (linear FIR–prediction of order m_v)

$$\hat{a}_{v,n} = \sum_{i=1}^{m_v} c_{v,i} \cdot z_{n-i} \cdot \hat{x}_{n-i} \qquad (2)$$

whereby the vertical FIR–filter coefficients $c_{v,i}$ are calculated with the Wiener–Hopf equation based on the frequency auto–correlation of the channel transfer function. The variable z_n denotes the received channel observation at frequency index n (and time index k), $\hat{x}_n \in \{\pm 1\}$ the hypothesized transmitted bits according to the trellis structure, $L_{a,v,n}$ the vertical *a priori* L–values and $\sigma_{v,pred}^2$ the variance of the vertical estimation error. The asterisk "*" is the complex–conjugate operator.

Accordingly, at time index k (frequency index n omitted), the horizontal APP estimation (time direction) is characterized by the metric increment

$$\gamma_k = \frac{\text{Re}\left\{z_k \cdot \hat{x}_k \cdot \hat{a}_{h,k}^*\right\}}{\sigma_{h,pred}^2} + \tfrac{1}{2} L_{a,h,k} \qquad (3)$$

with estimated channel coefficient

$$\hat{a}_{h,k} = \sum_{i=1}^{m_h} c_{h,i} \cdot z_{k-i} \cdot \hat{x}_{k-i} \qquad (4)$$

whereby the horizontal FIR–filter coefficients $c_{h,i}$ are based on the auto–correlation function in time direction. The $L_{a,h,k}$ denote the horizontal *a priori* L–values.

Separate one–dimensional APP estimation in frequency (i.e. vertical) and time (i.e. horizontal) direction is possible owing to the two–dimensional continuity of the channel transfer function. The channel coefficient $a_{k,n}$ does not change abruptly but continuously (smoothly) in time and frequency direction.

The inputs to the vertical APP estimator are the noise and fading affected channel observations $z_{k,n}$ (i.e. unprocessed discrete–time signal from channel), and *a priori* L–values $L_{a,v}$ (time/frequency indices k, n omitted for L–values) on the systematic bits of the product code, as fed back from the outer channel decoder. The pilot symbols are included into the APP estimation as "perfect *a priori* knowledge" (i.e. big positive or negative L–value) at the respective positions. For the remaining positions, the vertical *a priori* L–values $L_{a,v}$ are set to zero for the very first pass through the estimator and decoder. The vertical APP estimator outputs the estimated channel L–values $L_{ch,v}$ which are forwarded as *a priori* input $L_{a,h} = L_{ch,v}$ to the horizontal APP estimator. The horizontal APP estimator uses the *a priori* input $L_{a,h}$ and channel observations $z_{k,n}$ to calculate improved horizontal channel L–values $L_{ch,h}$.

Conceptually, $L_{ch,v}$ is *not* statistically independent of the channel observations $z_{k,n}$ (i. e. corrupted by same noise on channel) and thus it does not appear to be suited as *a priori* knowledge $L_{a,h}$ (which is additive, compare to (3)) for the horizontal estimation. However, the resulting noise terms on the vertical and horizontal channel estimates $\hat{a}_{v,n}$, $\hat{a}_{h,k}$ of (2), (4) are independent. It is important to notice that the uncertainty in the channel estimation dominates at the beginning of the iteration (or equivalently, for poor *a priori* and/or channel input) which means that $L_{ch,v}$ stays fairly uncorrelated from $z_{k,n}$ and thus contributes to significantly improve the horizontal estimation.

4. Inner Product Code

The big systematic part makes high–rate product codes attractive candidates as inner codes in an iterative estimation and decoding scheme. Only a few parity bits are added in horizontal and vertical direction. Low–complexity receivers can simply ignore this added redundancy and bypass the inner decoder to perform a "one–shot" decoding, only based on the outer channel code. More expensive, higher complexity receivers can make use of the parity bits to decode the inner product code, and, much more interestingly, can perform iterative decoding over the inner and outer code. This significantly improves the performance.

4.1. High–rate BCH component codes

High rate product codes are most commonly built of Hamming codes [8] or BCH–codes [9], if used in iterative decoding schemes. However, soft in/soft out decoding of those component codes can be rather complex. Our motivation for using an inner product code is to enable iterative decoding. We found that component codes based on simple memory one recursive systematic convolutional (RSC) codes are well–suited for this purpose provided that an appropriate puncturing scheme is applied.

4.2. Heavily punctured RSC component codes

Rather than using high–rate BCH–codes with big memory, we introduce a special puncturing scheme to make use of RSC component codes with small memory and low decoding complexity. Fig. 4 illustrates the concept of "heavy puncturing": Most of the parity bits (sometimes also referred to as coded bits) of a rate $1/2$ RSC mother code are discarded, such that from the total of n_s parity bits only n_p remain. The parity bits are periodically inserted into the systematic bit stream with an insertion period $\Delta P = n_s/n_p + 1$. Our first experiments were based on BCH component codes with 120 systematic and 7 parity bits; to work with a similar rate for the heavily punctured component codes, we chose $n_{s,h} = 136$, $n_{p,h} = 8$ in horizontal (time) direction, and $n_{s,v} = 1088$, $n_{p,v} = 64$ in vertical (frequency) direction ($\Delta P_h = \Delta P_v = 18$).

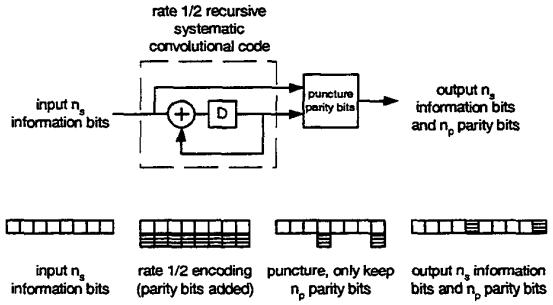

Figure 4. Puncturing scheme of inner rate $1/2$ mother code; $n_s = 8, n_p = 2$.

Owing to the heavy puncturing $n_p << n_s$ (code structure almost destroyed) the error correcting capabilities of the inner decoder are very poor. However, we noticed that

particularly those properties which are crucial for good iterative decoding performance, are still preserved, as will be further detailed in Section 5.2.

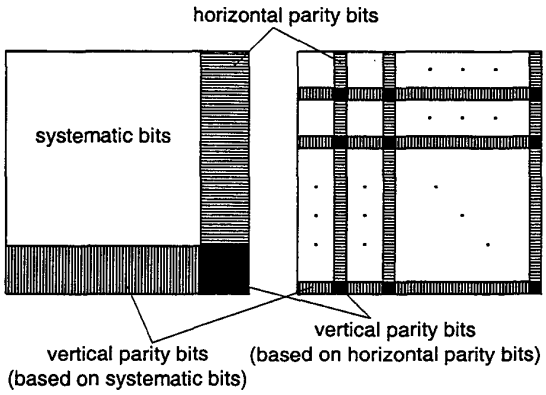

Figure 5. Product code in traditional (left) and modified arrangement (right) tailored to multicarrier modulation.

Fig. 5 shows a product code with parity bits in traditional and modified arrangement. In the modified arrangement the parity bits are spread over the whole multicarrier block. We prefer a periodical insertion rather than block–oriented insertion of parity bits because of the following two reasons: 1. The added redundancy is separated in time and frequency and thus is more likely to undergo independent fades; this is beneficial for good performance of the inner soft in/soft out trellis decoder. 2. As it turns out, the performance of the inner memory one code suffers less from the heavy puncturing if the parity bits are periodically outputted from the encoder, rather than using the same number of parity bits in a row at the very end of the encoding sequence.

5. Simulation Results

The multicarrier system parameters are given by the duration of one OFDM–symbol $T_s = 300\mu s$ and the sub–carrier spacing $\Delta f = 4$kHz. The channel is characterized by a channel delay spread of $\tau_{max} = 20\mu$s and a maximal Doppler shift of $f_D = f_{D,max} = 100$Hz; for some curves we increase the Doppler frequency $f_D = f_{D,max}$ to 200Hz and 300Hz respectively, to demonstrate the robustness of the proposed estimation and decoding scheme.

For the simulations without inner code, blocks of $N = 8 \cdot 136 = 1088$ adjacent sub–carriers and $K = 136$ consecutive OFDM symbols in time are used. The outer interleaving depth is $N \cdot K = 147968$ outer coded bits.

With inner code, we use blocks of $N' = 8 \cdot (136+8)$ sub–carriers and $K' = 136 + 8 = 144$ OFDM symbols in time. Hence, the inner code rate is $R_1 = (K \cdot N)/(K' \cdot N') \approx 0.892$.

As it turns out from simulations, the predictor coefficients $c_{h,i}, c_{v,i}$ of (1), (3) can be chosen in a rather relaxed way, just ensuring to have a low–pass behavior; we applied the *pragmatic choice* of $c_{h,i} = 1/m_h, c_{v,i} = 1/m_v$ and $\sigma^2_{pred} = \sigma^2_n$. The estimator memories are set to $m_h = 2, m_v = 2$.

BER charts are used for performance analysis of different system configurations. In addition to that, we apply the Extrinsic Information Transfer Chart to better compare the different estimation and decoding algorithms, and to gain more insight into the convergence behavior of iterative estimation and decoding.

The outer convolutional code is recursive systematic with feedback polynomial $G_r = 023$, feedforward polynomial $G = 037$ and outer code rate $R_2 = 0.5$. For every horizontal line of the inner systematic product code, we use one systematic bit as a pilot symbol. The pilot symbol rate is $R_p \approx 0.9926$, that is, only 0.74% of the transmitted systematic bits are used as pilots. For sub–carrier n, $n = 0, ..., N-1$, the discrete–time position of the pilot symbol is calculated to $k = (29 \cdot n) \,\mathrm{MOD}\, K$. As opposed to the Wiener estimator, the two–dimensional APP estimator is not very sensitive to the particular choice of the pilot symbol arrangement.

Note that in the following all E_b/N_0–values are given with respect to an overall information rate of $R = R_2 \cdot R_p \approx 0.496$ without inner code, and $R = R_1 \cdot R_2 \cdot R_p \approx 0.443$ if an inner code is used.

5.1. APP Estimators

We first leave out the inner product encoding/decoding and focus on the iterative estimation and outer decoding loop. Fig. 6 depicts mutual information transfer characteristics of the APP estimation stage. The *a priori* input to the APP estimator is on the abscissa, with $0 \le I_{A_1} \le 1$ expressed in terms of mutual information (bit per BPSK symbol). The *a posteriori* (channel) output is on the ordinate, $0 \le I_{E_1} \le 1$. As in [5, 6], the mutual information transfer characteristics are calculated based on a Gaussian assumption for the *a priori* input. The transfer characteristics describe the quality of the channel estimation provided that a certain *a priori* knowledge is available at the APP estimators.

For no *a priori* knowledge $I_{A_1} = 0$, the fading rate has a strong impact on the output I_{E_1}, which, however, is leveled out with increasing I_{A_1}. The APP estimator can almost achieve the capacity of the coherently detected fully interleaved, binary input/continuous output Rayleigh channel

[10] which is shown as a horizontal line for $E_b/N_0 = 9$dB.

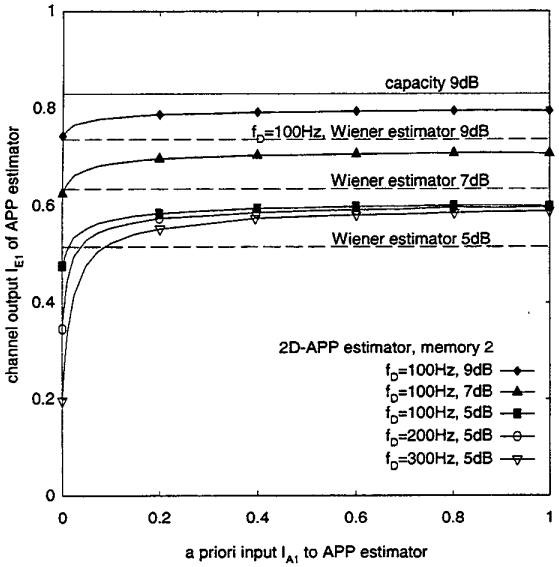

Figure 6. Mutual information transfer characteristics of APP estimator.

As a reference, the results for the conventional FIR–interpolation approach ("Wiener estimator", [1]) are given as dashed lines for $f_D = 100$Hz. In each dimension, the interpolation filter uses the 10 nearest pilot symbols. The pilot symbols are arranged in a rectangular grid (every 10th symbol in frequency and every 15th symbol in time direction), with a block size of $N = 1191$ sub–carriers by $K = 136$ OFDM–symbols. The filter coefficients are optimized to the channel auto–correlation functions which are assumed to be perfectly known. Note that the Wiener estimator runs into problems for $f_D > 111$Hz, owing to the fact that the pilot symbol spacing in time direction becomes larger than postulated by the sampling theorem. If no *a priori* knowledge is available, the Wiener estimator outperforms the APP estimator at lower E_b/N_0–values. However, it should be kept in mind that for the APP estimation the channel statistics is assumed to be unknown, and the parameters like memory and variance of prediction error are just pragmatically chosen, whereas the Wiener filter coefficients are optimized to the channel auto–correlation functions.

The APP estimators can overcome the weak beginning of their transfer characteristics by performing iterative estimation and decoding, as illustrated in Fig. 7. The trajectory of iterative decoding visualizes the exchange of channel and extrinsic information between estimator and outer decoder in the Extrinsic Information Transfer Chart. The trajectory starts at the origin and proceeds up to the first intersection of inner and outer transfer characteristics. The trajectory is a simulation result of the iterative decoding scheme, whereas the transfer characteristics are computed *individually* for the inner APP estimator and the outer decoder, applying independent Gaussian distributed random variables as *a priori* inputs.

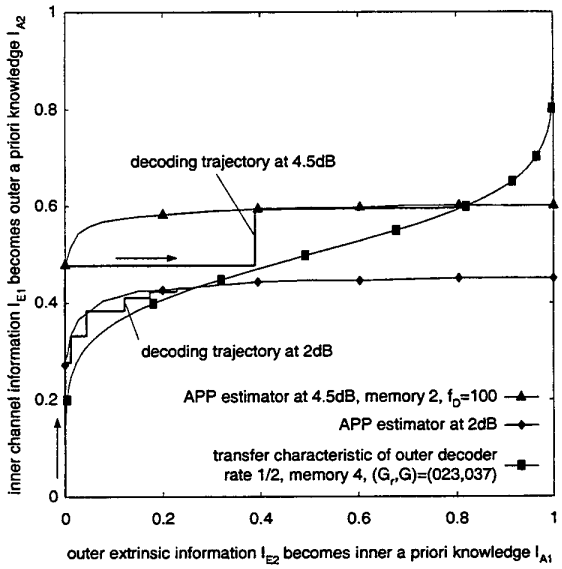

Figure 7. EXIT chart with transfer characteristics of inner APP estimator and outer memory 4 decoder.

Using a big interleaver of about $1.5 \cdot 10^5$ outer coded bits keeps correlations of extrinsic information small over several iterations, and the trajectory matches with the characteristics fairly well. It, however, slightly misses the outer decoder transfer characteristic. A more careful modeling of the outer *a priori* input distributions would remove this mismatch: Rather than using a simple Gaussian model, the distribution of a coherently detected Rayleigh fading signal should be applied. Still, the prediction accuracy is sufficient to derive some design guidelines for the iterative decoding scheme.

Transfer characteristics of the inner estimator are given at 2dB and 4.5dB. Note that the outer decoder transfer characteristic is independent of the E_b/N_0–value. It is just taken from [6]. The decoding trajectory at 2dB gets stuck after about 6 iterations, owing to the intersection of both characteristics. For 4.5dB the trajectory manages to reach higher extrinsic output I_{E_2} at the outer decoder, which corresponds to achieving a lower BER.

Fig. 8 compares the performance of the APP estimation with the FIR–interpolation method of the reference system.

The advantage of the system with APP estimation is about 1.5dB at BER 10^{-4}. The APP estimator proves to be very robust against higher fading rates, whereas the Wiener estimator fails for $f_D = 200$Hz and 300Hz (not shown).

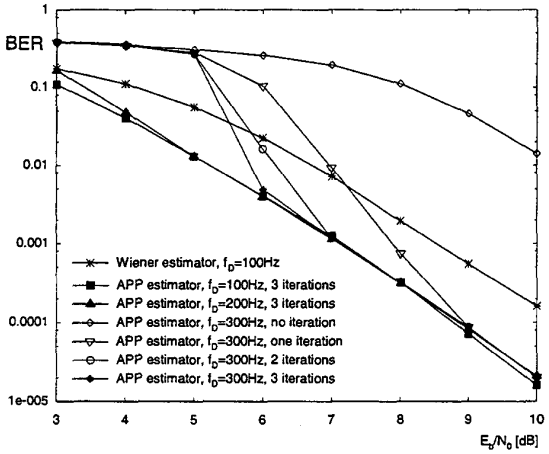

Figure 8. BER curves of APP estimation for different Doppler frequencies and different number of iterations.

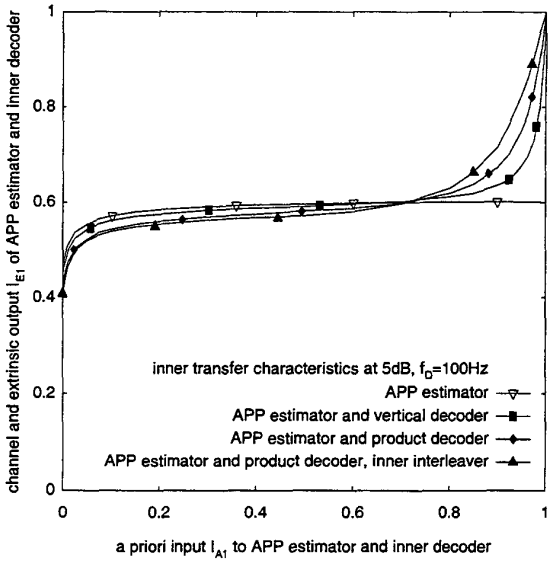

Figure 9. Transfer characteristics of inner estimation and decoding stage.

5.2. With Inner Product Code

Now we include the inner product code into our considerations. Fig. 9 allows to compare the different inner estimation and decoding stages.

The transfer characteristic of the plain APP estimator is taken from Fig. 6. The flattening out for higher I_{A_1}–input results in an intersection with the outer decoder characteristic and thus an early stop of the iterative estimation and decoding algorithm. However, if we add a few parity bits, e. g. those of the heavily punctured vertical memory one code, we can drive the mutual information transfer characteristics up to $(I_{A_1}, I_{E_1}) \approx (1, 1)$ — which is the key property to achieve a very low BER in an iterative estimation and decoding scheme. This effect can be amplified by introducing the full product code with vertical and horizontal encoding/decoding. If we additionally use an inner deinterleaver in between estimator and inner decoder (inner interleaver on transmitting side respectively), we can further improve the "extrinsic" gain of the inner code: It is a well–known property of convolutional codes that the corresponding trellis decoder performs better if neighboring input samples are uncorrelated. To keep the E_b/N_0–value constant, the channel noise is increased if more redundant bits are added. Hence, the stepwise lowering of the code rate explains the small shift towards lower I_{E_1}–output at the beginning of the transfer characteristics of Fig. 9.

The advantage of the system with inner decoder is visualized in the EXIT chart of Fig. 10. Now, the decoding trajectory at 4.5dB can converge towards $(I_{A_1}, I_{E_1}) \approx (1, 1)$, which directly relates to reaching a very low BER. Similarly to the situation of Fig. 7, the trajectory at 2dB gets stuck after about 9 iterations.

The BER chart of Fig. 11 gives results for the system with inner product code after 5 iterations. No inner interleaver is used. The BER curves for the system with inner interleaver (not shown) are very much the same, but require — on the average — a little less iterations to achieve the same BER.

6. Conclusions

A robust channel estimation algorithm based on the concatenation of two one–dimensional APP estimators has been presented. The new scheme outperforms the conventional Wiener filtering approach. An inner product code with heavily punctured RSC component codes has been introduced to further improve the performance of the iterative decoder. The Extrinsic Information Transfer Chart turns out to be a useful tool for studying the influence of different system parameters on the convergence properties of the iterative estimation and decoding loop.

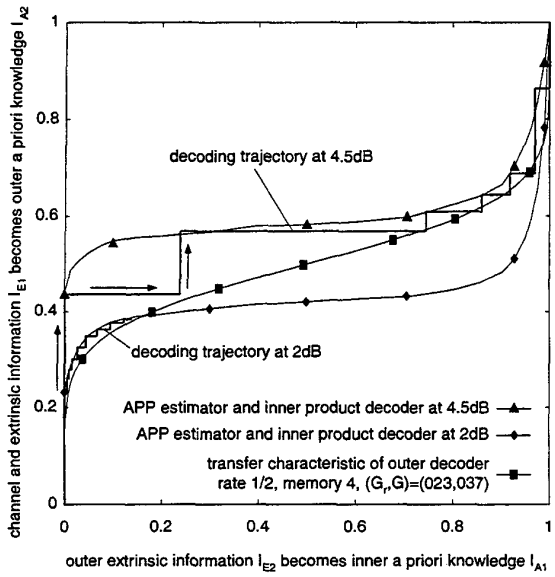

Figure 10. EXIT chart with transfer characteristics of inner APP estimator/product decoder and outer memory 4 decoder.

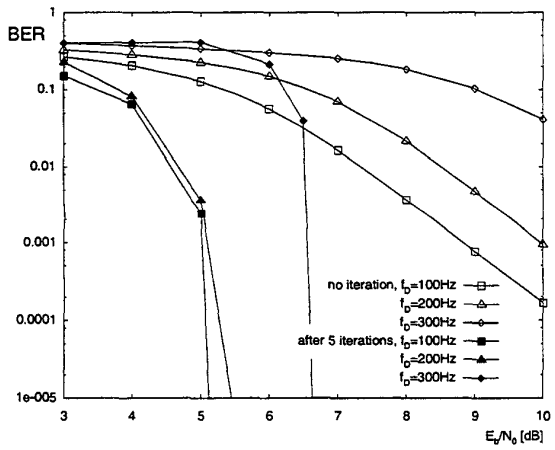

Figure 11. BER curves of system with inner product decoder; no inner interleaver.

References

[1] P. Hoeher, S. Kaiser, P. Robertson, "Two–dimensional pilot–symbol–aided channel estimation by Wiener filtering", *Proc. ICASSP*, pp. 1845–1848, April 1997

[2] P. Hoeher, J. Lodge, "Iterative Differential PSK Demodulation and Channel Decoding", *IEEE Trans. Comm.*, vol. 47, pp. 837-843, June 1999

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

[4] P. Robertson, E. Villebrun, P. Hoeher, "A comparison of optimal and sub–optimal MAP decoding algorithms operating in the log domain", *Proc. ICC*, pp. 1009–1013, June 1995

[5] S. ten Brink, "Iterative Decoding Trajectories of Parallel Concatenated Codes", *Proc. 3rd IEEE/ITG Conference on Source and Channel Coding*, pp. 75–80, Munich, Jan. 2000

[6] S. ten Brink, "Design of Serially Concatenated Codes based on Iterative Decoding Convergence", *2nd International Symposium on Turbo Codes*, Brest, France, Sep. 2000

[7] P. Hoeher, "A Statistical Discrete-time model for the WSSUS multipath channel", *IEEE Trans. Comm.*, vol 41, pp. 461-468, Nov. 1992

[8] J. Hagenauer, E. Offer, L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes", *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996

[9] R. M. Pyndiah, "Near–Optimum Decoding of Product Codes: Block Turbo Codes", *IEEE Trans. Comm.*, vol. 46, no. 8, Aug. 1998

[10] E. K. Hall, S. G. Wilson, "Design and Analysis of Turbo Codes on Rayleigh Fading Channels", *IEEE Journal on Sel. Areas in Comm.*, vol. 16, no. 2, pp. 160–174, Feb. 1998