Turbo Equalization for Channels with High Memory Using a List-Sequential (LISS) Equalizer

Joachim Hagenauer and Christian Kuhn

Institute for Communications Engineering (LNT), Munich University of Technology (TUM)
D-80290 München, Germany
Phone: (+49) 89 289 23466, Fax: (+49) 89 289 23490
E-mail: {hagenauer,kuhn}@ei.tum.de

Abstract: *For multipath channels with a high number of taps and signal constellation points the APP (BCJR) soft-in/soft-out detector in a turbo-scheme has a prohibitively high number of states. We replace it by a novel list-sequential stack (LISS) algorithm using augmented paths and an auxiliary stack for a length dependent bias term. For BERs lower than 10^{-4} the turbo system with a LISS equalizer has almost the same performance as the turbo system with an APP equalizer, yet with much smaller complexity. Furthermore, our method still works with 10 or 20 taps and has a controllable complexity determined by the used stack sizes.*

Keywords: equalizer, sequential algorithms, intersymbol interference, turbo equalization

1. Introduction

For decoding according to the turbo principle [1], [2] a soft-in/soft-out algorithm to calculate the a posteriori probability (APP) for bits and symbols is needed. The most popular ones work on a trellis, such as the optimal symbol-by-symbol a posteriori BCJR algorithm [3]. In many applications the number of states is very high, e.g. multipath channels with high memory encountered in high speed HF transmission [4] and magnetic recording. Sequential decoding of high memory convolutional codes with the Fano and Stack algorithm [7], [9] was already popular in the sixties of the last century because its complexity is independent of the number of states. It explores the tree structure of the underlying finite state machine. The problem with sequential decoding is that it suffers from erased frames where decoding cannot be finished. With a turbo scheme this poses no problem, because the outer or parallel decoder will decode erasures easily if they are embedded, after deinterleaving, in correct and reliable bits. At the first iteration the outer decoder returns a priori information and the sequential decoder will improve significantly and in most cases will decode erasure free. We propose here a LISt-Sequential equalizer (LISS) using a modified stack algorithm operating on a tree, accepting soft channel and a priori values and supplying a soft-output for each individual bit to be used by the outer decoder.

2. Turbo Equalization with LISS

We consider a serially concatenated system as shown in Fig. 1. The outer FEC encoder typically utilizes a terminated or tailbited convolutional code and the outer decoder is a soft-in/soft-out APP decoder. The turbo scheme uses as input to the inner block a column vector of size $M \cdot N \times 1$ with binary elements

$$\mathbf{x} = (\mathbf{x_1}, ...\mathbf{x_n}, ..., \mathbf{x_N})^{\mathrm{T}} \tag{1}$$

with the M-bit row vectors of size $1 \times M$

$$\mathbf{x_n} = (x_{1,n}, ...x_{m,n}, ..., x_{M,n}), \quad x_{m,n} \in \{+1, -1\}. \tag{2}$$

M bits of the n-th subblock are mapped to a complex QAM signal $s_n = s_n(x_{1,n} \dots x_{m,n} \dots x_{M,n})$ appended by terminating symbols and transmitted over an intersymbol interference (ISI) channel with L-taps

$$y_n = \sum_{k=0}^{L-1} h_k s_{n-k} + w_n. \tag{3}$$

The receiving process is corrupted by complex-valued

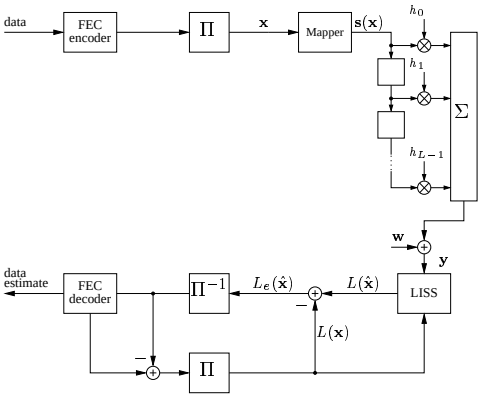


Figure 1: Transmission system with receiver performing turbo equalization.

AWGN with i.i.d. noise samples satisfying the pdf

$$p(w) = \frac{1}{2\pi\sigma_w^2} \mathrm{e}^{-\frac{|w|^2}{2\sigma_w^2}} \tag{4}$$

and σ_w^2 being the total noise power. For each individual bit the LISS equalizer should produce the a posteriori log-likelihood ratio (LLR) $L(\hat{x}_{m,n}) = L(x_{m,n}|\mathbf{y})$

$$L(\hat{x}_{m,n}) = \ln \frac{P(x_{m,n}=+1|\mathbf{y})}{P(x_{m,n}=-1|\mathbf{y})} \quad (5)$$

$$= \ln \frac{\sum_{\forall \mathbf{x}: x_{m,n}=+1} P(\mathbf{x}|\mathbf{y})}{\sum_{\forall \mathbf{x}: x_{m,n}=-1} P(\mathbf{x}|\mathbf{y})} \quad (6)$$

which can be split up into a priori and extrinsic parts

$$L(\hat{x}_{m,n}) = \ln \frac{p(\mathbf{y}|x_{m,n}=+1)}{p(\mathbf{y}|x_{m,n}=-1)} + \ln \frac{P(x_{m,n}=+1)}{P(x_{m,n}=-1)}$$

$$= L_e(\hat{x}_{m,n}) + L(x_{m,n}) \quad (7)$$

and is passed on to the outer decoder.

2.1. The path metric

For the LISS equalizer we evaluate expression (6) only for the best candidates of $\mathbf{x}$ found in a sequential search where we might have paths of different length. Defining the path metric

$$\ln P(\mathbf{x}|\mathbf{y}) = \ln p(\mathbf{y}|\mathbf{x}) + \ln P(\mathbf{x}) - \ln p(\mathbf{y}) \quad (8)$$

with the first two components

$$\ln p(\mathbf{y}|\mathbf{x}) = -N \cdot \ln(2\pi\sigma_w^2) - \sum_{n=1}^{N} \frac{|y_n - \sum_{k=0}^{L-1} h_k s_{n-k}|^2}{2\sigma_w^2} \quad (9)$$

$$\ln P(\mathbf{x}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \ln P(x_{m,n}), \quad (10)$$

$$\ln P(x_{m,n}) = x_{m,n} L(x_{m,n})/2 - \ln(\mathrm{e}^{+L(x_{m,n})/2} + \mathrm{e}^{-L(x_{m,n})/2}),$$

depending on the a priori knowledge. Contrary to the trellis approach the tree approach uses paths of different length N, meaning we cannot ignore the bias term $\ln p(\mathbf{y})$. On a trellis with $2^{M(L-1)}$ states it would be calculated from the forward recursion

$$\ln p(\mathbf{y}) = \sum_{n=1}^{N} \ln \sum_{S} \alpha_n(S), \quad (11)$$

$$\alpha_n(S) = \sum_{S'} \gamma_n(S', S) \frac{\alpha_{n-1}(S')}{\sum_{S'} \alpha_{n-1}(S')}. \quad (12)$$

For the branch probabilities

$$\gamma_n(S', S) = p(y_n|S', S) \cdot P(S|S') \quad (13)$$

we obtain

$$\ln \gamma_n = K_n + \ln \gamma_n', \quad (14)$$

$$K_n = -\ln(2\pi\sigma_w^2) - \sum_{m=1}^{M} \ln(\mathrm{e}^{+L(x_{m,n})/2} + \mathrm{e}^{-L(x_{m,n})/2}),$$

$$\ln \gamma_n' = -\frac{|y_n - \sum_{k=0}^{L-1} h_k s_{n-k}|^2}{2\sigma_w^2} + \sum_{m=1}^{M} x_{m,n} \frac{L(x_{m,n})}{2},$$

with K_n being constant within a trellis section. In practice, the normalized forward recursion (12) is performed with γ_n', because the influence of K_n cancels out for each n with the corresponding terms contained in (9) and (10). In section 2.4. we will approximate the bias term through an auxiliary stack to avoid the computationally complex trellis evaluation.

2.2. The stack algorithm

We use the stack algorithm [7] to perform a sequential search for the best paths. The ordered stack can have a maximal size (number of entries) of $|\mathcal{S}|$ with elements enumerated $l = 1, \ldots, L_S \leq |\mathcal{S}|$. An example with $M = 1$ is shown in Table 1. It contains the already explored L_S paths of different length and the metrics $\Lambda(l) = \ln P(\mathbf{x}|\mathbf{y})$ associated with the path $\mathbf{x}$. They are shown with their decision signs ± 1. The yet unexplored paths are indicated with a 0. The stack is ordered with descending metric values, the element with the highest metric $\Lambda(1)$ being at the top. We assume the finite stack size $|\mathcal{S}|$ to be usually much smaller than the total number of paths, i.e. $|\mathcal{S}| \ll 2^{NM}$.

The stack (Zigangirov-Jelinek) algorithm [9] proceeds as follows with the full path metric (8) and with the last novel step **5)**:

$l=1$	+	+	-	-	-	+	+	+	+	$\Lambda(1)$
$l=2$	+	+	-	-	-	+	+	+	-	$\Lambda(2)$
$l=3$	+	+	+	-	+	0	0	0	0	$\Lambda(3)$
$l=4$	+	-	+	+	0	0	0	0	0	$\Lambda(4)$
$l=5$	+	+	+	-	+	0	0	0	0	$\Lambda(5)$
...										...
$l=L_S$	-	-	+	-	+	+	0	0	0	$\Lambda(L_S)$
$n=$	1	2	3	4	5	6	7	8	N	

Table 1: Example stack of a binary tree with L_S entries each of maximal length $N = 9$. Two entries are extended to full length 9.

0) Initialization: The TOP stack entry is the root node with metric zero and length zero.

1) Extend the path l in the stack with the highest metric and which has not yet reached depth N (denoted as TOP stack entry) by the 2^M possible extensions by adding the respective incremental metrics to the path metric $\Lambda(l)$.

2) Remove the TOP stack entry and insert the extended paths in the stack.

3) Order the complete stack, which has now $L_S := L_S + 2^M - 1$ entries, according to the metric values with the highest metric on top.

4) If the number of stack entries $L_S \leq |\mathcal{S}| - (2^M - 1)$ go to 1) else stop extending the paths and go to **5)**.

5) Option Augmented Stack: Extend all paths with length less than N to full length N by a random tail with a modified metric to obtain an augmented stack described in section 2.3.

2.3. Full length stack augmentation

We have to extend all paths to full length without increasing the stack size. This is done by augmenting the paths in the stack, where hard decisions $x = \pm 1$ are replaced for the augmentation by expectations (soft bits) using possibly available a priori information

$$\bar{x} = \mathrm{E}\{x\} = \tanh(L(x)/2)$$

with

$$P(x = \pm 1) = \frac{1 \pm \bar{x}}{2}. \tag{15}$$

An example for QPSK with soft mapping is given in Fig. 2. It consists of 4 fully extended paths and of the augmented paths A,B,C,D. Note, that due to the soft bits the signal points need not to be at their original locations. We have now available a set $\mathcal{S}$ of

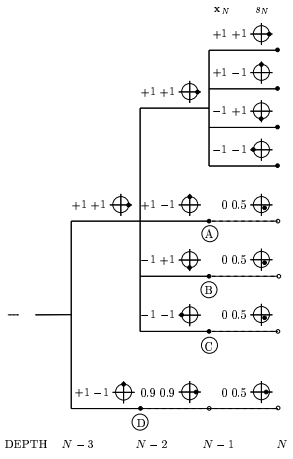


Figure 2: Example tree for QPSK transmission with soft augmentation.

augmented paths in the stack with stack size $|\mathcal{S}|$ and obtain for each bit the LLR by weighting in (6) the exponentiated metrics with the probabilities (15)

$$\begin{aligned} L(\hat{x}_{m,n}) &= \ln \frac{\sum_{\forall \mathbf{x} \in \mathcal{S}} P(x_{m,n} = +1) \cdot P(\mathbf{x}|\mathbf{y})}{\sum_{\forall \mathbf{x} \in \mathcal{S}} P(x_{m,n} = -1) \cdot P(\mathbf{x}|\mathbf{y})} \\ &= \ln \frac{\sum_{\forall \mathbf{x} \in \mathcal{S}} \frac{1+\bar{x}_{m,n}}{2} \cdot \mathrm{e}^{\Lambda(\mathbf{x})}}{\sum_{\forall \mathbf{x} \in \mathcal{S}} \frac{1-\bar{x}_{m,n}}{2} \cdot \mathrm{e}^{\Lambda(\mathbf{x})}} \end{aligned} \tag{16}$$

2.4. Auxiliary stack for the bias term $\ln p(\mathbf{y})$

In order to reduce the complexity of the bias computation we obtain a suitable estimate for $\ln p(\mathbf{y})$ from a tree rather than from a trellis. This method uses an additional stack, the **auxiliary stack** $\mathcal{S}_x$, with size $|\mathcal{S}_x|$ and derives estimates based on the approximate calculation of

$$\ln p(\mathbf{y}) = \ln \sum_{\mathbf{x}} \mathrm{e}^{\ln p(\mathbf{y}|\mathbf{x}) + \ln P(\mathbf{x})}. \tag{17}$$

The evaluation of (17) is done recursively by exploring a tree structure with metric $\Lambda_x(\mathbf{x}) = \ln p(\mathbf{y}|\mathbf{x}) + \ln P(\mathbf{x})$, which is equal to the metric for the main stack without the bias term. Furthermore, we apply the following algorithm:

0) Initialization: The TOP stack entry is the root node with metric zero and length zero.

1) Extend **all** paths in the auxiliary stack by adding the respective incremental metrics to each path metric $\Lambda_x(\mathbf{x})$.

2) Remove all previous entries and insert the extended paths in the auxiliary stack.

3) Calculate the bias estimate

$$\ln p(\mathbf{y}) \approx \ln \sum_{\mathbf{x} \in \mathcal{S}_x} \mathrm{e}^{\Lambda_x(\mathbf{x})} \approx \max_{\mathbf{x}} \Lambda_x(\mathbf{x}) \tag{18}$$

for the actual tree depth n.

4) Order stack entries according to metric values with highest metric on top.

5) If $n \geq N_x$ assuming a maximal auxiliary stack size of $|\mathcal{S}_x| = 2^{M \cdot N_x}$ the stack is filled up and all entries below $2^{(N_x - 1)M}$ have to be removed.

6) If n=N stop, else go to 1).

This method delivers exact bias values up to depth $n = N_x$, because in that case the auxiliary stack contains all paths of the particular length. Step 5) guarantees a limited size $|\mathcal{S}_x|$ of the auxiliary stack, but nevertheless it leads to good approximate values for bias terms with $n > N_x$.

3. Simulation Results

We present simulation results with BPSK modulation ($M = 1$) for a 3-tap reference channel with impulse response $\mathbf{h} = [h_0, \ldots, h_{L-1}]^T = [1/\sqrt{2}, 1/2, 1/2]^T$, which offers the possibility for a comparison between LISS and the APP equalizer. Furthermore, we consider a worst case 11-tap channel from Proakis [10], with coefficients $\mathbf{h} = [0.04, -0.05, 0.07, -0.21, -0.5, 0.72, 0.36, 0, 0.21, 0.03, 0.07]^T$, where the BCJR algorithm with 1024 states is no longer feasible.

3.1. Evaluation of channel capacity

Since we have calculated $\ln p(\mathbf{y})$ from (11) or at least approximatively from (18) we can also evaluate the channel capacity, respectively the mutual information following [5] as:

$$\begin{aligned} I(\mathbf{x};\mathbf{y}) &= H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x}) \\ &= -\frac{\log_2 \mathrm{e}}{N} \cdot \ln p(\mathbf{y}) - \log_2(2\pi \mathrm{e}\sigma_w^2) \end{aligned} \quad (19)$$

and obtain the result of Fig. 3.

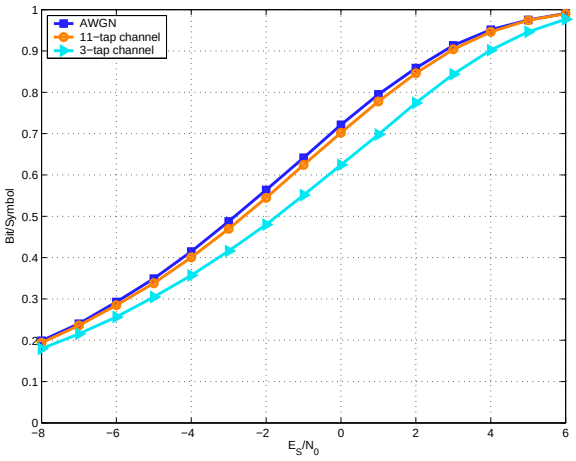


Figure 3: Mutual information for AWGN, 11-tap and 3-tap reference channel.

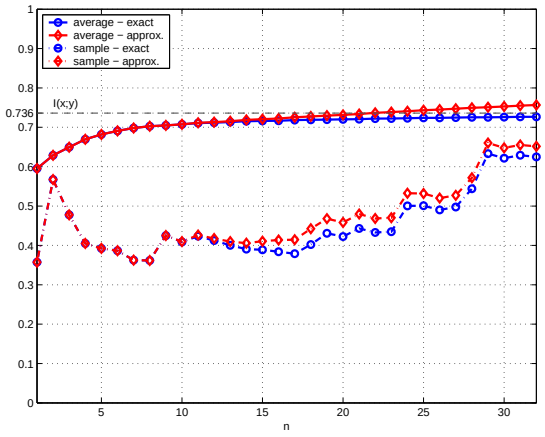
3.2. Results for Turbo equalization

We use blocks of $N = 32$, arranged in an interleaver of size 8192, with a rate 1/2 outer convolutional code with memory 4 and octal generators [46, 66].

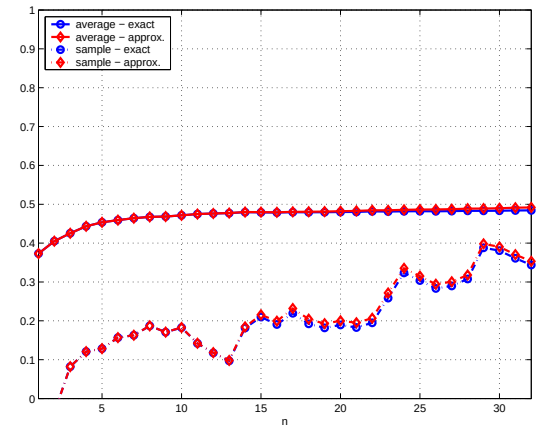
3.2.1. Reference 3-tap channel

First we simulate the reference 3-tap channel with a main stack of size $|\mathcal{S}| = 256$ and an auxiliary stack of size $|\mathcal{S}_x| = 128$, both reduced from 2^{32}. Figures 4(a) and 4(b) compare bias terms, scaled according to (19), calculated exactly using the trellis approach to the corresponding estimates computed with the auxiliary stack. Each figure shows sample bias terms and average values over 10^4 bias term realisations. In Fig. 4(a) no a priori information was used for the bias calculation. Fig. 4(b) depicts the result for a scaled bias term calculated using a priori information. The L-values $L(x_{m,n})$ were Gaussian distributed with mean $\sigma_L^2/2$ and variance σ_L^2. It can be seen that with some a priori information the bias term from the auxiliary stack is as good as the one from the full trellis.

Fig. 5 shows the BER after 5 iterations using the BCJR equalizer operating on a trellis (most left curve). The LISS achieves the same performance at a BER of 10^{-4}. Increasing $|\mathcal{S}|$ leads to a more graceful degradation. It was observed that although the bias term is needed, it can be approximated with the auxiliary stack.



(a) No a priori information: $\sigma_L^2 = 0$



(b) With random a priori information: $\sigma_L^2 = 2.5$

Figure 4: Averaged and sample bias terms scaled to mutual information (19) for the 3-tap channel, E_S/N_0 of 1.5 dB and $|\mathcal{S}_x| = 128$.

3.2.2. 11-tap multipath channel

The Proakis A (L=11) channel with BPSK on a frame with 32 symbols was used. With such a channel the APP BCJR trellis equalizer becomes too complex and we are forced to use the LISS selecting a stack size of $|\mathcal{S}| = 512$ and an auxiliary stack size of $|\mathcal{S}_x| = 256$, both down from 2^{32}. Fig. 6 shows iterations 0 to 6 and thus the feasibility of our new LISS

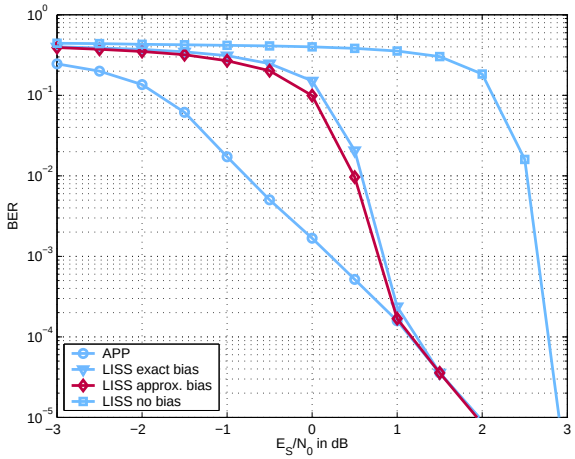


Figure 5: BER performance after the decoder for the reference 3-tap channel and a memory 4, rate 1/2 outer code. 5 iterations between the inner equalizer and the outer BCJR decoder.

equalizer in a turbo scheme. The performance limit for the equalizer in the selected simulation setup is given by the left most curve. This corresponds to an extrinsic genie LISS equalizer which receives perfect a priori information, except for the current bit.

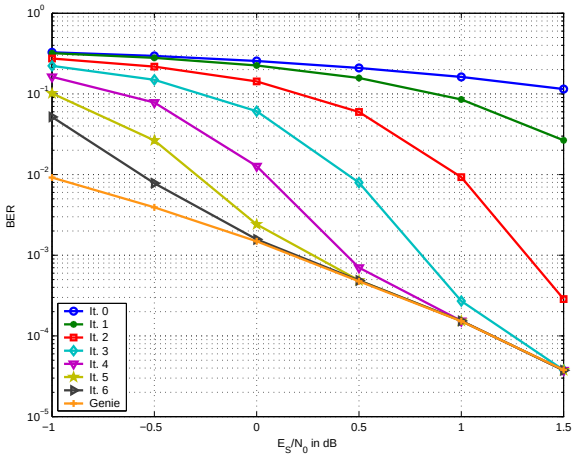


Figure 6: BER performance after the decoder for the Proakis A 11-tap channel and a memory 4, rate 1/2 outer code. Zero to 6 iterations 0 to 6 between the inner LISS equalizer and the outer BCJR decoder, compared to an extrinsic genie LISS equalizer.

4. Summary and Outlook

We have tackled the problem of turbo equalization for channels with high memory and high modulation signal constellation where the APP BCJR trellis equalization becomes infeasible due to excessive complexity. Instead we use a list-sequential equalizer (LISS) with a modified stack algorithm as inner soft-in/soft-out device. Only the size of the stack determines the complexity (storage and sorting). Therefore low- and high end equalizers can be used. The erasure problem due to unfinished paths is solved by augmenting paths with expected values from the outer decoder's a priori values. Because of paths with different length we have to estimate $\ln p(\mathbf{y})$ for different lengths. A useful byproduct is that we can use this to estimate the mutual information and therefore the constrained channel capacity of high memory channels. In ongoing work we make a more detailed complexity comparison and investigate the required stack size for different channel types. The same LISS has been used for high complexity MIMO [6] and multiuser turbo detection schemes.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo codes (1)", in *Proc. IEEE Int. Conf. on Commun.* ICC93, Switzerland, May 1993, pp. 1064-1070.

[2] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art", in *International Symposium on Turbo Codes.* ENST de Bretagne, September 1997, pp. 1-11.

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

[4] R. Otnes, M. Tüchler, "Block SISO linear equalizers for Turbo equalization in serial-tone HF modems ", in *Proc. Norwegian Signal Processing Symp. (NORSIG)*, Norway, October 2001, pp. 93-98.

[5] D. Arnold and H. Loeliger, "On the information rate of binary- input channels with memory", in *Proc. IEEE Int. Conf. on Commun.* ICC01, Finland, June 2001, pp. 2692-2695.

[6] S. Bäro, J. Hagenauer, and M. Witzke, "Iterative detection of MIMO transmission using a List-Sequential (LISS) detector", in *Proc. IEEE Int. Conf. on Commun.* ICC03, May 2003.

[7] J. Massey, "Variable length codes and the Fano metric", in *IEEE Trans. on Inform. Theory*, vol. IT-18, no. 1, January 1972, pp. 196-198.

[8] C. Weiß, S. Riedel, and J. Hagenauer, "Sequential decoding using a priori information", *Elec. Lett.*, vol. 32, no. 13, pp. 1190-1191, June 1996.

[9] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, 1983.

[10] J. Proakis, *Digital Communications, 4th Ed.* McGraw-Hill, 2001.