ITERATIVE SOURCE AND CHANNEL DECODING FOR GSM

Ralf Perkert, Markus Kaindl

Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)
Arcisstr. 21, 80290 Munich, Germany
Email: Markus.Kaindl@ei.tum.de

Thomas Hindelang

ICM MD MP RD MCH 81
Siemens AG
Grillparzer Str. 10-18, 81675 Munich, Germany
Email: Thomas.Hindelang@mch.siemens.de

ABSTRACT

When transmitting speech signals, residual redundancy is still left in the signal after source coding, due to limited complexity of the coding algorithms and delay constraints. This redundancy expresses in correlations inside one frame as well as in a time correlation of subsequent speech frames. The method of iterative channel and source decoding applied in this paper is based on exploiting this redundancy in terms of a priori knowledge of the source to improve decoding of the transmitted parameters. The used source a priori information is obtained and exploited directly on parameter level. In this paper we show the application of this theory to a real world mobile communication system. Here GSM speech transmission was chosen, but the presented method could also be applied to any other system which leaves a certain amount of redundancy in the speech coded signal.

1. INTRODUCTION

A lot of theoretical investigations about exploiting source residual redundancy have been done in the past [4][5][6][7][8]. In this work, the attempt was made to apply the idea of exploiting this redundancy to a current mobile communication system. The GSM Full Rate codec was chosen for various reason: It is a wide spread system and the speech coder uses scalar quantization that leaves a fair amount of redundancy in the signal which is necessary for this application.

In this paper, we show that high gains can be achieved applying scalar quantization and exploiting the residual correlations. Considering the results of this integrated approach of channel- and source coding the question arises if the individually optimized coders like ACELP (*Algebraic Code Excited Linear Prediction*) used in the GSM AMR codec in combination with more error protection are heading in the right direction. These are not only more complex, their speech quality also degrades abruptly in bad channel conditions, while the results of this work show a graceful degradation. In [9] a low complexity coding system was proposed for the IS-641 speech codec which performed better than the standard codec based on ACELP.

It is to mention that the approach used in this work can also be applied to any other codec as long as its speech coder is modified in terms of the employed quantization technique.

This work aimed at developing an iterative decoder for joint channel and source decoding, which is able to exploit source residual redundancy of the GSM Full Rate signal. In section 2 the correlation of the parameters is described. The used iterative decoder is then shown in section 3. In section 4 the simulation results are given for multiple configurations of the iterative decoder. Finally, in section 5 we conclude and give a perspective for possible further applications.

2. REDUNDANCY IN THE SPEECH CODED SIGNAL

The GSM speech coder [1] reduces the redundancy of the input speech signal by extracting 76 speech parameters (per frame) representing the signal. In a LPC analysis 8 filter coefficients are determined which are coded as LAR parameters. In the subsequent LTP analysis and RPE coding the LTP lags and block-amplitudes are extracted among others for each of the four subframes. For Full Rate channel coding [2] the bits of the parameters are ordered in classes of importance (50 most important bits: class 1a, 132 bits class 1b and 78 least important bits: class 2). The 185 class 1 bits (including 3 CRC bits for the class 1a) are error protected by convolutional coding with rate 1/2.

In this section the correlations of the Full Rate speech coded signal are expressed in form of mutual information. It describes the information a parameter contains about another one. The unit of the mutual information, is given in "bit". Table (1) shows the results for the most correlated parameters, the time correlation of the LAR filter coefficients (LAR) in the first column, and the intraframe correlation of neighbored LAR coefficients in the 2nd column. The LTP lag (LTP) and the RPE block-amplitude (Xmx) are coded on a subframe basis and we show the time correlation on this subframe basis. It is the same from the 4th subframe of the previous frame to the 1st subframe of the current frame, then from the 1st to the 2nd, and so on.

Parameter	$I(\underline{x};\underline{y})$	Parameter	$I(\underline{x};\underline{y})$
$\mathrm{LAR}_1(t\text{-}1);\mathrm{LAR}_1(t)$	1.063	$\mathrm{LAR}_1(t);\mathrm{LAR}_2(t)$	0.203
$\mathrm{LAR}_2(t\text{-}1);\mathrm{LAR}_2(t)$	0.574	$\mathrm{LAR}_2(t);\mathrm{LAR}_3(t)$	0.214
$\mathrm{LAR}_3(t\text{-}1);\mathrm{LAR}_3(t)$	0.578	$\mathrm{LAR}_3(t);\mathrm{LAR}_4(t)$	0.085
$\mathrm{LAR}_4(t\text{-}1);\mathrm{LAR}_4(t)$	0.409	$\mathrm{LAR}_4(t);\mathrm{LAR}_5(t)$	0.078
$\mathrm{LAR}_5(t\text{-}1);\mathrm{LAR}_5(t)$	0.306	$\mathrm{LAR}_5(t);\mathrm{LAR}_6(t)$	0.036
$\mathrm{LAR}_6(t\text{-}1);\mathrm{LAR}_6(t)$	0.389	$\mathrm{LAR}_6(t);\mathrm{LAR}_7(t)$	0.040
$\mathrm{LAR}_7(t\text{-}1);\mathrm{LAR}_7(t)$	0.328	$\mathrm{LAR}_7(t);\mathrm{LAR}_8(t)$	0.037
$\mathrm{LAR}_8(t\text{-}1);\mathrm{LAR}_8(t)$	0.349		
$\mathrm{LTP}_1(t);\mathrm{LTP}_2(t)$	0.343	$\mathrm{Xmx}_1(t);\mathrm{Xmx}_2(t)$	1.379

Table 1. Mutual Information of Parameters in GSM Full Rate

The parameters themselves also contain redundancy $R(\underline{x})=H_0(\underline{x})-H(\underline{x})$ due to their unequal distribution ($H_0(\underline{x})$: number of bits of $\underline{x}$ and $H(\underline{x})$: Entropy of $\underline{x}$). The values are given in Table (2).

This redundancy together with the mutual information is exploited for improving the parameter estimation. The LAR parameters show interframe correlation (between frames) as well as intraframe correlation (between neighbored parameters). LTP lag and block-amplitude both contain intersubframe correlation (time correlation between subsequent subframes, e.g. subframe 1 and 2 in Table (1)) and the block-amplitude shows the highest mutual information of all

Parameter	R($\underline{x}$)	Parameter	R($\underline{x}$)
LAR$_1(t)$	0.63	LAR$_2(t)$	1.19
LAR$_3(t)$	0.37	LAR$_4(t)$	0.62
LAR$_5(t)$	0.42	LAR$_6(t)$	0.27
LAR$_7(t)$	0.25	LAR$_8(t)$	0.29
LTP$_1(t)$	0.67	Xmx$_1(t)$	1.64

Table 2. Redundancy of the parameters

parameters. The found correlations are stored in so-called a priori tables which contain the conditional probabilities for transitions between the considered pairs of parameters. Stationarity is assumed and simulations showed, that storing the a priori knowledge once delivers better results than estimation in the decoder.

3. THE ITERATIVE DECODER

Fig. (1) shows the block diagram of the iterative decoder which aims at optimizing the speech parameters. The subsequent speech decoder is not shown.

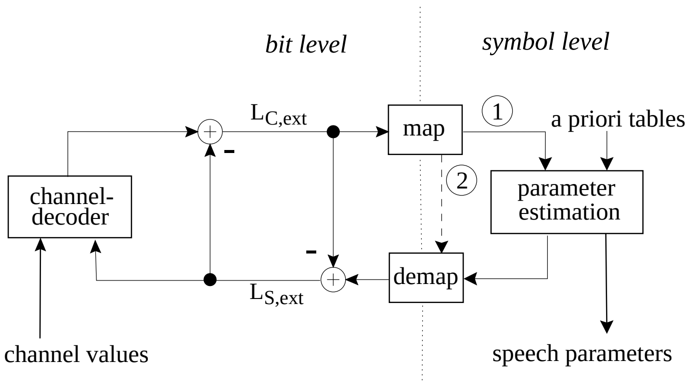

Fig. 1. Block diagram of the iterative decoder

The single steps of the loop are as follows:
At first the MAP channel decoder receives 378 soft values from the channel and delivers 185 class 1 soft values, the 78 uncoded class 2 channel values are attached. The outputs of the channel decoder are 263 (including the CRC bits) so-called log-likelihood ratios ("L-values") [10], which contain the soft information of the decoded bits. Now, the extrinsic information L$_{C,ext}$ is calculated by subtracting (in the logarithmic domain) the input to channel decoder (L$_{S,ext}$) from the L-values after the channel decoder.
In the block "map" the L-values of the bits are first converted to bit probabilities and then we calculate parameter probabilities (see Sec. 3.1). For the 8 LAR coefficients, the four LTP lags, and the 4 block amplitudes (Xmx) we apply a parameter estimation (1 in Fig. 1). The remaining parameters are uncorrelated and thus, bypass the parameter estimation (2 in Fig. 1).
The parameter estimation now separately decodes the LAR, LTP-lag and block-amplitude parameters on the parameter level by exploiting the a priori tables as described in section (3.2). The outputs are optimized probability estimates for each parameter.
In the branch back to the channel decoder the steps are made in the reverse direction. In the demapper the parameter probabilities are mapped back to bit probabilities and converted into L-values again. After that the extrinsic information L$_{S,ext}$ for the channel decoder is determined by subtracting the input to the parameter estimator (L$_{C,ext}$) from its output.
This extrinsic information of all bits from the correlated parameters is then used as a priori information for the next step of channel decoding. Now, the whole loop is executed several times until the values do not improve any more. Finally, the 76 speech parameters including their reliability are fed to a mean square estimation and afterwards to the speech decoder.
For simplification, the quantized parameter $\underline{x}_i$ denotes a bit vector

$$\underline{x}_i = \{x_{i,0}, x_{i,1}, \ldots, x_{i,M-1}\}, \text{ with } x_{i,m} \in \{0, 1\},$$

where $\underline{x}_i$ stands for any of the correlated speech parameters, namely LAR, LTP, or Xmx. The vector $\underline{x}_i$ consists of a number of M bits of the quantized parameter (e.g., LAR$_1$: $M = 6$). The index i of $\underline{x}_i$ denotes the number of the considered parameter within a frame (1..8 for LAR$_i$, 1..4 for LTP$_i$, and 1..4 for Xmx$_i$). If necessary, we denote the time, then $\underline{x}_i(t)$ means the parameter $\underline{x}_i$ in the current frame, whereas $\underline{x}_i(t-1)$ denotes the same parameter in the previous frame.

3.1. The Mapping

The mapping consists of two different steps: Converting L-values and probabilities into each other and transforming bit probabilities to parameter probability distributions.
The L-values from the channel decoder are desirable for calculating the extrinsic information. However, for the parameter estimation these L-values need to be converted to probabilities.
The second task of the mapper is to calculate probabilities on parameter level for the LAR, LTP-lag and block amplitude parameters out of the appropriate bit probabilities. For each parameter the output of the mapper are the probabilities for all possible values of this parameter. E.g., for the LAR 1 parameter consisting of 6 bits, the probability for all of the 64 possible values is calculated from the 6 bit probabilities by assuming independence of the bits. These probability distributions are needed in the parameter estimation to be able to use the a priori information. The assumption of independence of the bits is justified, as the memory of the convolutional code is small and the parameters are wide spread over the block.
The demapper in the backward branch first calculates the bit probabilities out of the parameter probability distribution. The probability for a certain bit to be zero is determined by summing up all parameter values for which this bit is zero. Finally, the probabilities are converted back to L-values.

3.2. The Parameter Estimation

In this block, time correlations (interframe) as well as correlations of the parameters between frames (intraframe) are exploited yielding an estimate of the a posteriori probability of the parameter i if all parameters of this kind of the current frame $\hat{x}_1 ... \hat{x}_4$ and all previous with the same number $\hat{X}_i$ are received. E.g., for the LTP lag i:

$$P_{apost}(\underline{x}_i) = P(\underline{x}_i | \underline{\hat{x}}_1, \underline{\hat{x}}_2, \underline{\hat{x}}_3, \underline{\hat{x}}_4, \underline{\hat{X}}_i(t-1)), \qquad (1)$$

where $\hat{x}_1$, $\hat{x}_2$, $\hat{x}_3$, $\hat{x}_4$ denote the LTP lags of the 4 subframes.
The a posteriori probabilities can be calculated very efficiently by applying forward- and backward recursions similar to those executed in a MAP decoder using the BCJR [3] algorithm.
Taking a closer look at the details, there are various differences between the decoding algorithm of the LARs and the LTP lags/block

amplitudes. So the principle shall first be shown for the LTP lag parameters which is similar to the one for the block amplitudes. Afterwards, the extensions that are needed for the LARs are described.

The LTP lags and the block-amplitudes

Each of the 4 LTP lags per frame is quantized with 7 bits and therefore 128 parameter values could be possible. These values of the LTP lag 1 to 4 are arranged in a kind of trellis (step $i = 1$ to $i = 4$) where each of them is initialized with the (extrinsic) probability P_d of the parameter taking this particular value. Additionally there is the correlation between $\text{LTP}_4(t\text{-}1)$ and $\text{LTP}_1(t)$ and thus we can initialize the trellis at step $i = 0$ with the a posteriori probabilities $P_{apost}(\underline{x}_4(t\text{-}1))$.

Now, the forward probability P_{fwd} for each parameter value is determined considering all forward probabilities from the previous state multiplied with their (extrinsic) probability from the channel decoder $P_d(\underline{x}_i)$ and weighted by the transition probability $P(\underline{x}_{i-1}|\underline{x}_i)$ into this state found in the a priori table:

$$P_{fwd}(\underline{x}_i) = \sum_{\underline{x}_{i-1}} P_{fwd}(\underline{x}_{i-1}) \cdot P_d(\underline{x}_{i-1}) \cdot P(\underline{x}_{i-1}|\underline{x}_i), \quad (2)$$

with $i \in \{1, 2, 3, 4\}$. The derivation for the formula can be found in [9]. In full analogy, the backward probabilities are calculated as:

$$P_{bwd}(\underline{x}_i) = \sum_{\underline{x}_{i+1}} P_{bwd}(\underline{x}_{i+1}) \cdot P_d(\underline{x}_{i+1}) \cdot P(\underline{x}_{i+1}|\underline{x}_i) \quad (3)$$

For time correlated signals the joint probabilities $P(\underline{x}_{i-1}|\underline{x}_i)$ and $P(\underline{x}_i|\underline{x}_{i-1})$ are identical and thus the same a priori table can be used for the forward and backward recursion.

Finally, the product of forward and backward probabilities together with the probabilities from the channel decoder yield the a posteriori probabilities:

$$P_{apost}(\underline{x}_i) = C \cdot P_{fwd}(\underline{x}_i) \cdot P_{bwd}(\underline{x}_i) \cdot P_d(\underline{x}_i) \cdot P(\underline{x}_i) \quad (4)$$

with C being a constant normalizing the sum of the probabilities of all parameters to 1 and $P(\underline{x}_i)$ denoting the 0th order statistics of the source given by the unequal distribution of the parameter values.

Now these are the output probabilities of the parameter estimation for the LTP lag parameters which are then sent to the channel decoder again. The decoding of the block amplitudes is identical, except that only 64 parameter values are possible.

The LAR Parameters

All 8 LAR parameters are determined once per frame and each of them denotes a different speech filter parameter. Thus in this case it is necessary to differ between intra and interframe (time) correlation and both kinds can be exploited.

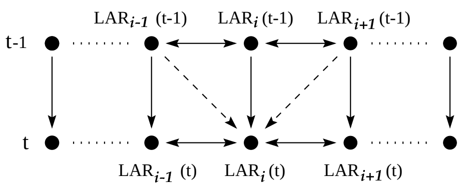

Fig. 2. Correlations of 1st order (—) and cross correlation (- - -)

We assume a first order Markov Model. Thus only correlations between directly neighbored (time and inside frame) LARs are considered, as illustrated by the solid lines in Fig. (2). As the cross correlations, e.g., $\text{LAR}_{i-1}(t\text{-}1)$ with $\text{LAR}_i(t)$ (dashed line) are to be neglected, in this case the forward and backward probabilities in the recursions need to be calculated first for the intraframe correlation without considering the time correlations. These are then exploited in an additional step.

Intraframe correlation: Basically the algorithm works very similar to the one described above for the LTP lags and the block-amplitudes. The LAR parameter values are also initialized in a kind of trellis. However in this case the number of states is not constant as the LAR 1 to 8 are quantized with different numbers of bits. Thus, the trellis has 64 states at the first step and 8 states at the last step.

Now the forward and backward recursions are calculated as described for the LTP lags but this time exploiting the intraframe correlation between neighbored LAR. Note, in contrast to the LTP lags and the block-amplitudes, for the LARs the correlation of different parameters is considered (even with different numbers of bits). Thus the correlation matrices are not symmetrical any more and therefore two different a priori tables are necessary for $P(\underline{x}_{i-1}|\underline{x}_i)$ and $P(\underline{x}_i|\underline{x}_{i-1})$ respectively.

Interframe correlation: In a second step the time correlation of the LAR parameters is exploited yielding time prediction probabilities for the current frame.

The prediction probability of a parameter having a certain value in the current frame can be estimated by summing up all probabilities from the last frame multiplied with the transition probabilities from the old values to the current one. The transition probability is found in the time a priori tables for the LAR parameters.

$$P_{time}(\underline{x}_i(t)) = \sum_{\underline{x}_{i,t-1}} P(\underline{x}_i(t)|\underline{x}_i(t-1)) \cdot P_{apost}(\underline{x}_i(t-1)) \quad (5)$$

with the so-called a posteriori probability of the last frame $P_{apost}(\underline{x}(t-1))$ denoting the final result after parameter estimation of the previous frame.

Finally, to get the a posteriori probabilities, the above calculated prediction probabilities need to be merged together. This is done in a similar way to the LTP lags and the block amplitudes but now the time prediction probabilities need also to be taken into account. Thus for each parameter value the a posteriori probability is the product of the forward probability, the backward probability, the probability from the channel decoder and the time prediction probability:

$$P_{apost}(\underline{x}_i) = C \cdot P_{fwd}(\underline{x}_i) \cdot P_{bwd}(\underline{x}_i) \cdot P_d(\underline{x}_i) \cdot P_{time}(\underline{x}_i) \quad (6)$$

with C again being a normalization constant. Compared to (4) $P(\underline{x}_i)$ has been replaced by $P_{time}(\underline{x}_i)$. The time prediction probabilities include the 0th order statistics [9]. The a posteriori probabilities have to be stored for the calculation of the time prediction probability of the next frame.

Mean Square Estimation

After the last iteration, a mean square estimation as in [9] is carried out for each parameter. In speech coding a squared error measure like the SNR describes the transmission reasonably well. Thus, a mean square estimation is more appropriate compared to decoding the most probable value, especially for the LARs and the block amplitudes. Due to effects like pitch doubling the maximum a posteriori estimation could be taken for the LTP lags. In simulations it turned out that the results are almost the same.

4. SIMULATION RESULTS

For the simulations the GSM full rate transmission system was used. This standard is not changed and therefore this approach could be directly applied to the existing system. We used a burst Rayleigh fading

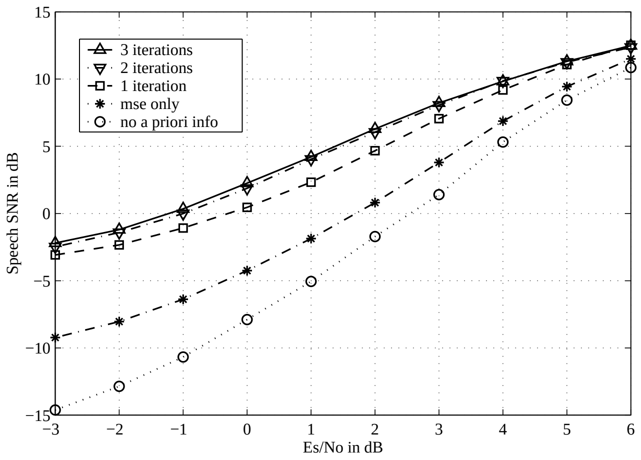

Fig. 3. Speech SNR compared to the standard decoder

channel which is a quite good assumption for a typical GSM channel, like typical urban (TU), low speed and ideal frequency hopping (IFH). At the receiver side the deinterleaver, the iterative decoder and the speech decoder follow.

4.1. Exploiting all a priori information

In Fig. (3) the PCM speech SNR is illustrated when exploiting a priori information about all three considered parameters and using the mean square estimator. For comparison the dotted line illustrates decoding result for the standard GSM Full Rate (without any additional error concealment). The dash-dotted line shows the result if only the MSE is applied. The 3 upper lines show the results for the iterative decoder after 1, 2 or 3 iterations. More iterations do not achieve further gains. For typical GSM transmission conditions a gain of about 2.5 dB in channel SNR can be achieved

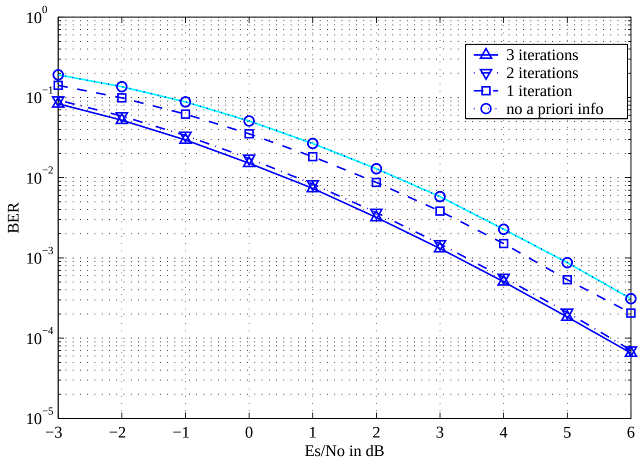

Fig. 4. BER class 1a bits (exploiting all a priori info & MSE)

Although the target of the decoder is to optimize parameters, comparing the Bit Error Rate (BER) in Fig. (4) shows a relatively constant gain in channel SNR of about 1.8 dB.
The Frame Error Rate (FER) is lowered by ratios around 1/2, which corresponds to a gain of about 1 dB in channel SNR.

4.2. Complexity Aspects

For complexity reasons not the complete decoder may be implemented in a future application. The complexity is about 5 times per iteration higher compared to the standard decoder.
Thus it is of interest how decoding still improves if only some a priori knowledge is exploited. In particular, parameter estimation only for the block-amplitudes still shows remarkable gains which are close to the curves above, whereas exploiting only the LARs or LTP lags lowers the gain.
Another approach is to consider only the most significant bits of the parameters. In particular, for the LTP parameters (7 bit) this reduces the number of states and transitions in the trellis for the recursions a lot. Neglecting the 2 least significant bits for all parameters lowers the complexity a lot. It is less than 2 times per iteration higher compared to the standard decoder, i.e. the parameter estimation is less complex than channel decoding.

5. CONCLUSIONS

We applied the concept of iterative source and channel decoding that exploits source residual redundancy to a standard communication system. The parameter estimation was done directly on parameter level for the LAR coefficients, the LTP lag and the block-amplitude. We found remarkable gains considering the BER and the speech SNR. Extensive listening tests with 60 speech samples and 20 listeners showed a gain of about 2 dB. The results show the capabilities of low complexity scalar quantization techniques for the encoder in combination with exploiting a priori knowledge in the decoder. In particular for bad channels, the graceful degradation of the speech quality shows the advantages of the presented concept.

6. REFERENCES

[1] GSM Technical Specification 06.10, ”Full Rate Speech Transcoding”, ETSI 1997

[2] GSM Technical Specification 05.03, ”Channel Coding”, ETSI 1996

[3] 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. IT-20, pp. 284-287, March 1974

[4] J. Hagenauer, ”Source-controlled channel decoding,” *IEEE Trans. Commun.*, vol.43, pp.2449-2457, September 1995

[5] N. Farvardin and V. Vaishampayan, ”Optimal Quantizer Design for Noisy Channels: An Approach to Combined Source-Channel Coding,” IEEE Trans. on Inf. Theory, vol. 33, pp. 827-838, Nov. 1987

[6] T. Hindelang, J. Hagenauer and S. Heinen, ”Source-controlled channel decoding: estimation of correlated parameters,” *ITG-Fachbericht, Conference Source and Channel Coding*, vol. 159, pp.251-258, January 2000

[7] T. Hindelang, T. Fingscheidt,N. Seshadri and R.V. Cox, ”Combined source/channel (de-)coding: Can a priori information be used twice?” in *Proceed. IEEE ICC 2000*, New Orleans, USA, June 2000, pp.1208-1212

[8] F. Alajaji, N. Phamdo and T. Fuja, ”Channel Codes that Exploit the Residual Redundancy in CELP-Encoded Speech,” IEEE Trans. on Speech and Audio Proc., vol. 4, no. 5, Sept. 1996

[9] T. Fingscheidt, T. Hindelang, R.V. Cox, N. Seshadri, ”Joint source-channel (de-)coding for mobile communications,” accepted for *IEEE Trans. Commun.*, July 2000

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