HIGH RATE DATA HIDING IN ACELP SPEECH CODECS

Bernd Geiser and Peter Vary

Institute of Communication Systems and Data Processing (ind)
RWTH Aachen University, Germany
{geiser|vary}@ind.rwth-aachen.de

ABSTRACT

A new method for hiding digital data in the bitstream of an ACELP speech codec is proposed in this paper. The key element of our method is an alternative search strategy for the ACELP codebook which allows for *joint* data hiding and speech coding. The concept has been examplarily applied to the AMR speech codec (12.2 kbit/s mode) and it is shown that steganographic data can be reliably transmitted at a rate of up to 2 kbit/s both with a negligible effect on the subjective quality of the coded speech and with reasonable computational complexity. Apart from data hiding, it is further pointed out that our method can also be exploited to reduce the codec bit rate.

***Index Terms*—** ACELP speech coding, data hiding, watermarking, steganography

1. INTRODUCTION

In a digital communication system, techniques for "data hiding" or "digital watermarking" [1] allow to establish a virtual communication channel that is embedded within the transmitted "host signal". In practice, this host signal usually represents some multimedia data, i.e., audio, image, or video content. In general, a good data hiding scheme for such signals has to be designed such that

- the hidden data can be detected/extracted reliably at the receiving end (possibly even after deliberate "attacks"),
- the minimum required data rate is guaranteed,
- and the modified host signal is not (or hardly) subjectively distinguishable from the original signal.

A widespread application of data hiding is the indication of the host signal's origin (e.g., for authentication or digital rights management). In contrast, we aim at the hidden transmission of auxiliary data (steganography). In this case, the robustness to deliberate attacks might be less relevant, but other transmission characteristics are more important such as a higher hidden data rate, the need for a *constant* (minimum) rate, and robustness to transmission errors.

Specifically, in this paper, we focus on data hiding for *speech host signals*. In the literature, data hiding for speech signals is mostly performed directly on the digital speech signal or in a transformed domain, where the latter usually aims at reduced audibility of the embedded watermark. Alternatively, speech features like the pitch structure in voiced speech segments may be modified to convey the additional information. Common methods for the data hiding are, e.g., "spread spectrum watermarking" or quantization based techniques such as the "Scalar Costa Scheme". Several speech watermarking systems that use one or a combination of these methods have been proposed in the literature, e.g., [2–7].

In contrast to these "classical" approaches, steganographic data can alternatively be embedded into a *compressed* or *encoded* representation of the host signal. This method is called "bitstream watermarking" or "compressed domain watermarking". Naturally, it is only applicable if the considered transmission system implements signal compression, i.e., a *speech codec* in our case. The data embedding is then performed either directly on the content of the bitstream (e.g., by overwriting least significant bits) or by modifying some partially decoded parameters (requantization), cf. [8]. Bitstream watermarking has been realized for various multimedia source coding schemes such as JPEG image coding [9], H.264 video coding [10], or MPEG-2 Advanced Audio Coding [11]. In general, the respective embedding methods are specific to the codec for which they have been designed. Bitstream watermarking for *speech codecs* has so far been investigated in [8, 12–14]. In particular, the present paper extends our previous work on data hiding for ACELP coded speech signals [8, 14] and gives a practical implementation within the AMR speech codec [15] that allows for a comparatively high steganographic data rate of 2 kbit/s. The embedded bits can be *perfectly* reconstructed directly from the AMR bitstream.

The remainder of the paper is organized as follows: Sec. 2 reviews and discusses the principle of CELP watermarking. Sec. 3 describes an actual implementation in the AMR speech codec. Sec. 4 provides subjective and objective test results for our realization and Sec. 5 concludes the paper.

2. JOINT CELP CODING AND DATA HIDING

The principle of joint *code excited linear prediction* (CELP) coding and data hiding has been proposed by [12] where a rather low steganographic capacity of 37 bit/s has been achieved. A solution for state-of-the-art ACELP codecs has been introduced in [14] and it could be shown that bit rates of several 100 bit/s can be reliably transmitted without compromising the quality of the coded speech signal. The respective techniques are briefly reviewed in this section.

For the application of data hiding to CELP coders, it turns out to be advantageous to integrate the watermarking procedure *into* the analysis-by-synthesis loop for the fixed codebook (FCB) search. This can be achieved by applying a "Binning Scheme" to the FCB. Hence, the embedding of N steganographic bits per (sub-)frame is achieved by partitioning the fixed excitation codebook $\mathcal{C}$ into $M = 2^N$ disjoint sub-codebooks $\mathcal{C}_m$ with $m \in \{0, \ldots, M-1\}$ such that $\bigcup_{m=0}^{M-1} \mathcal{C}_m \subseteq \mathcal{C}$. The FCB search *with information-embedding* can then be formulated as

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathcal{C}_m} \chi(\mathbf{c}), \tag{1}$$

where m is the *message* to be embedded, $\mathbf{c} \in \mathcal{C}_m$ are the examined candidate codevectors, and $\chi(\mathbf{c})$ is the CELP cost function

$$\chi(\mathbf{c}) = ||\mathbf{v}||^2 - \frac{\left(\mathbf{v}^{\mathrm{T}}\mathbf{Hc}\right)^2}{||\mathbf{Hc}||^2} \quad (2)$$

with the target vector $\mathbf{v}$ (pitch removed prediction residual) and the perceptually weighted filter matrix $\mathbf{H}$. The hidden message is *decoded* by identifying the sub-codebook that contains the received vector $\hat{\mathbf{c}}$, i.e., m is given by

$$m = \left\{m' \,:\, \hat{\mathbf{c}} \cap \mathcal{C}_{m'} = \hat{\mathbf{c}}\right\}. \quad (3)$$

Considering the described embedding scheme, one might argue that $|\mathcal{C}_m| \approx |\mathcal{C}|/M$, i.e., the number of examined FCB entries is decreased by a factor of M for each frame (cf. [12]). The inevitable consequence would be a decreased quality of the coded speech. Yet, when taking a closer look at practical ACELP codecs, it can be observed that the respective FCB search is — for reasons of complexity reduction — by far *non-exhaustive*, i.e., typically only a small heuristically selected subset $\mathcal{C}' \subset \mathcal{C}$ is examined during FCB search. Profiting from this fact, it is readily seen that much more advantageous FCB partitionings are feasible which include *additional* FCB entries, that have not been taken into account in the original search procedure. Ideally, such partitionings achieve $|\mathcal{C}_m| \geq |\mathcal{C}'|$. Moreover, if it is also possible to establish a FCB partitioning that provides M "equally good" sub-codebooks $\mathcal{C}_m$ (each yielding a coding performance that is comparable to that of the originally used sub-codebook $\mathcal{C}'$), then the data hiding procedure does not (or only insignificantly) degrade the resulting speech quality.

3. A STEGANOGRAPHIC AMR CODEC

This section reviews the standard FCB search procedure for the 12.2 kbit/s mode[1] of the AMR codec [15] and then introduces an alternative "steganographic" search strategy which allows a hidden data rate of 2 kbit/s. Compared to [14], the proposed method does not only provide higher data rates but it also requires less computational power (cf. Sec. 3.2.2).

3.1. Standard FCB Search

According to the ACELP principle, the AMR codec uses a ternary codebook where ten signed unit pulses are placed within a zero vector of length of 40 (or, equivalently, within a 5 ms subframe). Thereby, the admissible positions $i_0, \ldots, i_9$ for the ten pulses are defined by Tab. 1. In particular, the pulse pairs at positions (i_k, i_{k+5}) with $k \in \{0, \ldots, 4\}$ share one of the five possible "tracks" (interleaved sub-grids) within the considered subframe. The *signs* of all pulses are pre-selected "out of the loop" and therefore constant w.r.t. the standard closed-loop FCB search.

As shown in Tab. 1, each pulse can be placed on eight different positions, i.e., it can be coded with 3 bit. In total, this results in $2^{10\cdot 3} \approx 10^9$ possibilities for pulse position selection per subframe. Apparently, 10^9 evaluations of the CELP criterion (2) per subframe would yield a prohibitive complexity. Therefore, several measures have been taken in the AMR codec to reduce the computational load:

1. The first pulse is fixed on the global maximum of a function $b(n)$, $n \in \{0, \ldots, 39\}$, which is the sum of the normalized long term prediction (LTP) residual and the normalized and backward filtered target vector, i.e., $\mathbf{v}^{\mathrm{T}}\mathbf{H}/||\mathbf{v}^{\mathrm{T}}\mathbf{H}||$. See [15] for details.

[1]This AMR mode is virtually identical with the GSM EFR codec.

Table 1. FCB structure of the AMR codec (12.2 kbit/s)

pulse	valid positions
i_0, i_5	$0, 5, 10, 15, 20, 25, 30, 35$
i_1, i_6	$1, 6, 11, 16, 21, 26, 31, 36$
i_2, i_7	$2, 7, 12, 17, 22, 27, 32, 37$
i_3, i_8	$3, 8, 13, 18, 23, 28, 33, 38$
i_4, i_9	$4, 9, 14, 19, 24, 29, 34, 39$

2. Within four iterations, the second pulse is tentatively set on the local maximum of $b(n)$ within each of the four tracks that have not yet been occupied by the first pulse.
3. For each iteration, the remaining eight pulses are not jointly optimized, but the codevector is constructed successively by adding *pulse pairs*. Thereby, the CELP criterion is updated per pulse pair.

In total, $4 \cdot (4 \cdot 8^2) = 1024$ contributions to the criterion are computed, but only four *complete* candidates (with all 10 pulses set) are examined, i.e., just *one* for each iteration of the search algorithm.

3.2. Steganographic FCB Search

For our modified FCB search strategy that allows to embed 2 kbit/s of hidden data into the AMR bitstream, we need to define the "message" m that shall be embedded into a 5 ms subframe. Here, m is given as a 5 ms $\cdot$ 2 kbit/s $= 10$ bit binary sequence. This message is split into five sub-messages with two bits each. The sub-messages are denoted by, e.g., $(m)_{0,1}$ for the first two bits of m. To enable the embedding of $N = 10$ steganographic bits according to the description in Sec. 2, the FCB from Tab. 1 needs to be partitioned into $M = 2^{10}$ sub-codebooks. Accordingly, the proposed FCB search algorithm has been derived by

1. restricting the set of admissible FCB codevectors in order to establish 2^{10} disjoint sub-codebooks, and
2. re-expanding the search space (as compared to the standard FCB search from Sec. 3.1) such that a good trade-off between speech quality and computational complexity is found.

3.2.1. Codebook Partitioning

In our realization, we restrict the admissible pulse positions for the *second* pulse in each track. Specifically, $i_5, \ldots, i_9$ may only take two out of eight possible values. Obviously, as there are four possibilities to achieve a unique constellation, this allows the embedding of $\log_2(8/2) = 2$ bit per track. So for this specific configuration, the total steganographic data rate is $5 \cdot 2$ bit $= 10$ bit per subframe. Now, due to the specific *sign* encoding in the AMR codec [15], the decoder cannot distinguish anymore if i_k or i_{k+5} is received. Normally, this does not matter, since both pulses are located within the same track. However, in our application, this information is crucial. As a solution, we couple the admissible positions of the *second* pulse i_{k+5} in each track to the position i_k of the respective *first* pulse by employing the rules from Tab. 2. Naturally, this strategy requires i_k to be known before the admissible positions i_{k+5} for the second pulse can be computed. A suitable organization of the codebook search can ensure this property, see Sec. 3.2.2. An alternative FCB restriction rule could also take the pre-selected pulse *signs* into account.

Table 2. Restricted FCB structure of the steganographic AMR codec at 12.2 kbit/s (pulse positions $i_5, \dots, i_9$); the search space for the pulse positions $i_0, \dots, i_4$ is identical to the standard method from Tab. 1 — $(m)_{i,j} \in \{0, \dots, 3\}$ specifies the bits at positions i and j of the steganographic message m in binary representation; $\mathcal{G}$ / $\mathcal{G}^{-1}$: Gray encoding and decoding by table lookup; $X \oplus Y$ is the bitwise exclusive disjunction (XOR) of two binary strings; $\lfloor x \rfloor \doteq \max\{n \in \mathbb{Z} \mid n \le x\}$, particularly $\lfloor i_n/5 \rfloor$ gives a 3 bit pulse position index within track n.

pulse	first valid position	second valid position
i_5	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_0}{5} \rfloor) \oplus (m)_{0,1}\right) \cdot 5$	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_0}{5} \rfloor) \oplus ((m)_{0,1} + 4)\right) \cdot 5$
i_6	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_1}{5} \rfloor) \oplus (m)_{2,3}\right) \cdot 5 + 1$	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_1}{5} \rfloor) \oplus ((m)_{2,3} + 4)\right) \cdot 5 + 1$
i_7	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_2}{5} \rfloor) \oplus (m)_{4,5}\right) \cdot 5 + 2$	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_2}{5} \rfloor) \oplus ((m)_{4,5} + 4)\right) \cdot 5 + 2$
i_8	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_3}{5} \rfloor) \oplus (m)_{6,7}\right) \cdot 5 + 3$	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_3}{5} \rfloor) \oplus ((m)_{6,7} + 4)\right) \cdot 5 + 3$
i_9	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_4}{5} \rfloor) \oplus (m)_{8,9}\right) \cdot 5 + 4$	$\mathcal{G}^{-1}\left(\mathcal{G}(\lfloor \frac{i_4}{5} \rfloor) \oplus ((m)_{8,9} + 4)\right) \cdot 5 + 4$

With the codebook partitioning defined in Tab. 2, the *decoding* of the hidden information is performed by simply computing

$$(m)_{2k,2k+1} = \left[\mathcal{G}\left(\left\lfloor \frac{i_k}{5} \right\rfloor\right) \oplus \mathcal{G}\left(\left\lfloor \frac{i_{k+5}}{5} \right\rfloor\right)\right] \bmod 4 \quad (4)$$

for $k \in \{0, \dots, 4\}$, where $X \oplus Y$ is the bitwise exclusive disjunction (XOR) of two binary strings. It is worth noting that the restriction rules from Tab. 2 and also the decoding equation (4) take the standardized index assignment of the pulse position codewords (via Gray coding, cf. [15]) into account. Naturally, $\mathcal{G}$ and $\mathcal{G}^{-1}$ could also be omitted, but the reason for their inclusion is an increased robustness of the hidden information to transmission errors. The Gray coding has the undesired property that a single bit error in a Gray coded codeword may result in *two* bit errors within the decoded codeword. Hence, the error rate for the steganographic data would be needlessly increased by a significant amount. Further note that, when computing (4), the Gray coded versions of the codewords are directly available from the AMR bitstream, i.e., no further decoding operation is necessary to retrieve the hidden bits.

3.2.2. Expansion of the Search Space

Using the restricted FCB from Tab. 2 in conjunction with the standard search procedure that has been described in Sec. 3.1 would result in a significantly degraded speech quality, because instead of 1024, now merely $4 \cdot (4 \cdot (8 \cdot 2)) = 256$ contributions to the CELP criterion would be examined. To that effect, remedy can be found by applying a modified search strategy that examines more codevectors. In principle, all of the three complexity reduction methods mentioned in Sec. 3.1 could be relaxed in order to ensure that an adequate part of the FCB is covered for each of the 2^{10} sub-codebooks.

Our implementation retains the first two methods, i.e., the fixed position of the first pulse on the maximum of $b(n)$ and also the four iterations, where the second pulse is tentatively placed on the maxima within the unoccupied tracks. In fact, the search space expansion is achieved by jointly optimizing the positions of *more* than two pulses. Specifically, in our implementation, the search is organized such that *four* pulse positions are jointly optimized. Moreover, it is ensured that the first pulse in each track is known before the admissible positions of the respective second pulse are computed according to the rules from Tab. 2 (cf. Sec. 3.2.1). The modified FCB search method is detailed in the box below. Within this algorithm, Step 7 provides $8^2 \cdot 2^2$ contributions to the CELP criterion while Step 9 provides merely $8 \cdot 2^3$. In total, this amounts to the computation of $4 \cdot (8^2 \cdot 2^2 + 8 \cdot 2^3) = 1280$ contributions (in contrast to 1024 in the standard implementation). To obtain an estimate of the required computational power, we have instrumented the C source code of the *floating point* version of the AMR codec and we measured a complexity increase of approximately 3.3 MIPS compared to the standard AMR encoder (less than 20% relative). Though, note that our implementation does not yet exploit the full optimization potential (such as pre-computations, etc.).

3.3. Alternative Use Case: Bit Rate Reduction

So far, the proposed ACELP search method has been exploited to retain compatibility with the standard AMR codec while allowing to embed steganographic data. Yet, a second approach is feasible, too: Let the steganographic message m be constant. Then, encode the pulse position $i_0, \dots, i_4$ with 3 bit per pulse and the pulse positions $i_5, \dots, i_9$ with merely *one* bit per pulse. The encoding with a single bit is possible because the two admissible pulse positions are now also known at the decoder. Unfortunately, the sign coding scheme of

Steganographic FCB Search for the AMR Codec at 12.2 kbit/s

1. Let $(k_0, \dots, k_4)$ denote a pre-selected track permutation which ensures that the maximum of $b(n)$ lies in track k_0.
2. Fix i_{k_0} on the global maximum of $b(n)$.
3. *Compute the admissible values for* i_{k_0+5}.
4. Initialize the iteration counter.
5. Set i_{k_1} to the position of the maximum of $b(n)$ within track k_1.
6. *Compute the admissible values for* i_{k_1+5}.
7. Jointly optimize $i_{k_2}, i_{k_3}, i_{k_0+5}, i_{k_1+5}$ subject to a partial evaluation of (2).
8. *Compute the admissible values for* i_{k_2+5} *and* i_{k_3+5}.
9. Jointly optimize $i_{k_4}, i_{k_2+5}, i_{k_3+5}, i_{k_4+5}$ subject to a partial evaluation and update of (2). Thereby, *update the admissible values for* i_{k_4+5} *suitably.*
10. Final evaluation of (2). Remember the selected codevector, if an improvement is obtained.
11. Cyclically shift the current permutation of tracks $k_1, \dots, k_4$: $k_l \leftarrow k_{(l \bmod 4)+1}$ for $l \in \{1, \dots, 4\}$.
12. Increase the iteration counter and stop if four iterations have been carried out; otherwise go to step 5.

the AMR coder does not work anymore in this case. Hence, the full 10 bit (instead of 5 bit) are required for sign transmission. In total, this yields $3\,\text{bit}\cdot 5\,\text{pulses} + 1\,\text{bit}\cdot 5\,\text{pulses} + 10\,\text{bit} = 30\,\text{bit}$. In contrast to the standard encoding with 35 bit per subframe, this scheme saves $5\,\text{bit}/5\,\text{ms} = 1\,\text{kbit/s}$ which is exactly the data rate of the hidden information minus the additional rate for five pulse signs. In this respect, the hidden transmission of steganographic data can also be viewed as a *redundant encoding* of the admissible pulse positions.

4. TEST RESULTS

Here, the results of subjective and objective comparisons of our "steganographic" AMR codec with the standardized version are presented based on the AMR *floating point* implementation.

In order to assess if any *difference* can be perceived between decoded speech samples from both versions of the AMR codec, we conducted an ABX test where 11 experienced listeners — using closed back headphones in a quiet environment — had to decide whether the presented test sample X was equal to reference A or B. The options A and B have been randomly assigned to "standard AMR speech" and "speech with 2 kbit/s of hidden data". For the test, six short utterances from the NTT corpus (3 female and 3 male speakers) have been processed by both versions of the coder and presented to the subjects. Each utterance had to be judged four times. Before making a judgment, the samples A, B, and X could be played ad libitum. In total, $11 \cdot 4 \cdot 6 = 264$ votes have been received, and only in 162 cases (61%), the correct decision was made. A *statistically significant* number of correct votes was only observed for the female speech samples (66%, 66%, 70%). All listeners agreed that the (possibly) perceived differences were very hard to detect and that the difference between samples A and B in terms of speech quality is very small.

As an *objective* measurement, we computed an averaged "Host-to-Watermark" ratio (HWR). The HWR is the logarithmic ratio between the power of the unmodified decoded speech and the power of the "watermark signal" which is defined as the difference signal between the unmodified decoded speech and the decoded watermarked speech. On average, we measured a HWR of 19.3 dB. To provide a meaningful comparison, we also measured the average SNR that is incurred when migrating from the standard *floating point* implementation to the standard *fixed point* implementation of the AMR codec. The respective measurement yields 20.3 dB which is merely 1 dB above the HWR of our data hiding scheme.

5. CONCLUSIONS

We have introduced a method to hide steganographic data with a comparatively high rate in the bitstream of an ACELP speech codec. The data hiding is performed jointly with the analysis-by-synthesis codebook search. The hidden bits can be directly reconstructed from the bitstream, whereby bit errors may only occur due to a noisy transmission channel. Moreover, we have presented an implementation in the AMR speech codec (12.2 kbit/s mode) which hides 2 kbit/s of data (16% of the codec rate) in the bitstream or, alternatively, reduces the codec bit rate by 1 kbit/s. Thereby, it was observed that the speech quality is only insignificantly affected.

As a potential application, the hidden bits may be used for speech bandwidth extension (BWE) purposes (see, e.g., [8, 14]). For example, the BWE information from the ITU-T G.729.1 codec (1.65 kbit/s, see [16]) would easily fit as steganographic payload along with some additional error protection. Other applications might include the transmission of information that is beneficial for frame erasure concealment (FEC). Finally, it is even imaginable to include, e.g., the *spectral envelope* information as steganographic payload. This would reduce the codec's bit rate by up to 2 kbit/s.

6. REFERENCES

[1] P. Moulin and R. Koetter, "Data-hiding codes," *Proceedings of the IEEE*, vol. 93, no. 12, pp. 2083–2126, Dec. 2005.

[2] Q. Cheng and J. Sorensen, "Spread spectrum signaling for speech watermarking," in *Proc. of ICASSP*, Salt Lake City, UT, USA, May 2001.

[3] A. Sagi and D. Malah, "Data embedding in speech signals using perceptual masking," in *Proc. of EUSIPCO*, Vienna, Austria, Sept. 2004.

[4] M. Celik, G. Sharma, and A. M. Tekalp, "Pitch and duration modification for speech watermarking," in *Proc. of ICASSP*, Philadelphia, PA, USA, Mar. 2005.

[5] S. Chen and H. Leung, "Concurrent data transmission through analog speech channel using data hiding," *IEEE Signal Processing Letters*, vol. 12, no. 8, pp. 581–584, Aug. 2005.

[6] B. Geiser, P. Jax, and P. Vary, "Artificial bandwidth extension of speech supported by watermark transmitted side information," in *Proc. of INTERSPEECH*, Lisbon, Portugal, Sept. 2005.

[7] K. Hofbauer and G. Kubin, "High-rate data embedding in unvoiced speech," in *Proc. of INTERSPEECH*, Pittsburgh, PA, USA, Sept. 2006.

[8] P. Vary and B. Geiser, "Steganographic wideband telephony using narrowband speech codecs," in *Conference Record of Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Nov. 2007, invited talk.

[9] P. Wong and O. Au, "A blind watermarking technique in JPEG compressed domain," in *Proc. of ICIP*, Rochester, NY, USA, Sept. 2002.

[10] M. Noorkami and R. Mersereau, "Compressed-domain video watermarking for H.264," in *Proc. of ICIP*, Genova, Italy, Sept. 2005.

[11] F. Siebenhaar, C. Neubauer, and J. Herre, "Combined compression/watermarking for audio signals," in *110th Conv. of the AES*, Amsterdam, The Netherlands, May 2001.

[12] Z.-M. Lu, B. Yan, and S.-H. Sun, "Watermarking combined with CELP speech coding for authentication," *IEICE Trans. on Inf. and Systems*, vol. E88-D, no. 2, pp. 330–334, 2005.

[13] N. Chétry and M. Davies, "Embedding side information into a speech codec residual," in *Proc. of EUSIPCO*, Florence, Italy, Sept. 2006.

[14] B. Geiser and P. Vary, "Backwards compatible wideband telephony in mobile networks: CELP watermarking and bandwidth extension," in *Proc. of ICASSP*, Honolulu, Hawai'i, USA, Apr. 2007.

[15] ETSI Recommendation GSM 06.90, "Digital cellular telecommunications system (phase 2+); adaptive multi-rate (AMR) speech transcoding," version 7.2.1, release 1998, Apr. 2000.

[16] B. Geiser, P. Jax, P. Vary, H. Taddei, S. Schandl, M. Gartner, C. Guillaumé, and S. Ragot, "Bandwidth extension for hierarchical speech and audio coding in ITU-T Rec. G.729.1," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, Nov. 2007.