Published in Proceedings of the 3rd COST 276 Workshop on
Information and Knowledge Management for Integrated Media Communication, 114-119, 2002
which should be used for any reference to this work

Design Techniques for Spectral Quantization in Wideband Speech Coding

G. Biundo, S. Grassi, M. Ansorge, F. Pellandini, P.-A. Farine

Institute of Microtechnology (ESPLAB), University of Neuchâtel,
Rue A.-L. Breguet 2, 2000 Neuchâtel, Switzerland.
Phone: +41 32 718 34 32; Fax: +41 32 718 34 02; e-mail: Giuseppina.Biundo@unine.ch.

Abstract – In this paper we describe the design of spectral quantization for wideband speech coding in order to achieve low bit-rate, low-complexity, transparent quantization schemes. The theoretical bases are given as well as the experimental design, listening tests and the implementation in a proprietary CELP (Code Excited Linear Prediction) coder.

1. Introduction

Wideband speech compression (coding) is currently a predominant research topic, driven by recent and ongoing standardization activities. Wideband speech signals have a 0.05-7.0 kHz bandwidth and a sampling frequency (f_s) of 16 kHz, whereas narrowband speech has a 0.3-3.4 kHz bandwidth and is sampled at 8 kHz. With respect to narrowband coding, wideband coding improves naturalness and intelligibility of the decoded speech and finds applications in video conferencing, telephony, circuit-switched and packet networks, and multimedia broadcast. It could also improve human-machine interfaces and recognition over coded speech.

Spectral quantization in narrowband speech coding has been extensively studied and there exist a lot of scientific publications on the subject, as well as implementation in a large amount of speech coding standards such as the ITU-T G.729 [1] and the ETSI NB-AMR [2]. On the other hand, the topic of spectral quantization for wideband speech coders is newer, with lesser amount of information available. Section 2 reviews some basis on spectral quantization. Experimental design of a quantization scheme for our proprietary CELP coder, achieving the best tradeoff between bit rate and complexity is given in Section 3. A short revision of previous work on spectral quantization for wideband speech coders is given in [3], whereas the most recent example of wideband spectral quantization in standards is described in Section 4. Conclusions and further work are given in Section 5.

2. Spectral Quantization

In Linear Prediction (LP) - based speech coders, the spectral envelope of a speech frame is modeled with the all-pole filter $1/A_m(z)$ given by:

$$A_m(z) = 1 + \sum_{i=1}^{m} a_i \cdot z^{-i}, \qquad (1)$$

where the a_i are the LP coefficients and m is the order of the model. The order m is typically 10 in narrowband coders, and 16 in wideband coders. Hereafter, an order of 16 is used.

In forward LP-based coders, the spectral information, contained in the LP coefficients, is quantized and transmitted. As the LP coefficients are difficult to quantize, different one-to-one representations such as Parcor Coefficients, Log-area-ratio, Line Spectrum Pairs (LSP) and Immittance Spectral Pairs (ISP) [4] are used. We focus on LSP and ISP, as LSPs are widely used in recent narrowband standards as well as in most speech coders in the scientific literature, whereas ISPs are used in the new ETSI WB-AMR speech coder [5].

LSPs, also called LSFs (Line Spectrum Frequency), are a spectral envelope representation with good quantization properties, such as bounded range, intra- and inter-frame correlation, localized spectral sensitivity and simple check of filter stability. The LSPs are calculated from $A_m(z)$ using the symmetric polynomials $P(z)$ and $Q(z)$ given by:

$$P(z) = \left[A_m(z) + z^{-(m+1)} A_m(z^{-1}) \right] \big/ (1 + z^{-1}); \qquad Q(z) = \left[A_m(z) - z^{-(m+1)} A_m(z^{-1}) \right] \big/ (1 - z^{-1}). \tag{2}$$

$P(z)$ and $Q(z)$ are completely specified by the angular position of their roots on the upper semicircle of the z-plane. The corresponding angles are the *m* LSP parameters, denoted by ω_i. Hereafter we use the LSPs in the frequency domain, with $f_i = f_s \cdot \omega_i / 2\pi$. If $1/A_m(z)$ is strictly stable, the LSPs are ordered as $0 < f_1 < f_2 < \ldots < f_m < f_s/2$. The converse is also true, namely that if an LSP set is ordered, its corresponding filter $1/A_m(z)$ is strictly stable.

Immittance spectral pairs (ISP) [4] are said to have slightly better quantization properties and lesser computational complexity. Their derivation is similar to LSP. In fact, the first $m-1$ ISPs of a system of order m are the LSPs of the system of order $m-1$, while the mth ISP is derived from the last LP coefficient a_m using $f_m = f_s \cdot \cos^{-1}(a_m) / 4\pi$ (note that $-1.0 < a_m < 1.0$).

The Spectral Distortion (SD), expressed in dB, of a speech frame is given by [6]:

$$SD = \left(\left(100/(f_2 - f_1)\right) \cdot \int_{f_1}^{f_2} \left[\log_{10} S(f) - \log_{10} Sq(f)\right]^2 df \right)^{1/2}, \tag{3}$$

where f is the frequency in Hz, f_1 and f_2 specify the frequency range and $S(f)$ and $Sq(f)$ are the original and quantized spectrum of $1/A_m(z)$. In case of narrowband coders, the quantization process is considered "transparent" (i.e. does not introduce audible distortion) if the average SD is less than 1 dB, and if there are less than 2% of outliers with 2 dB < SD ≤ 4 dB, and no outliers with SD > 4 dB [6]. A different criterion for transparency in wideband speech coders is proposed in [7]. We have found that (at least for our speech databases) this criterion does not yield at all transparency and that the criterion used for narrowband coders, with a frequency range of 0-7 kHz, seems to yield transparency also for wideband coders. We thus decided to use the latter criterion, coupled with listening test verifications.

a. Quantization techniques

Vector Quantization (VQ) exploits the (intra-frame) correlation between neighboring LSPs for bit-rate reduction [6]. Sub-optimal VQ, such as Split Vector Quantization (SVQ) and Multi-Stage Vector Quantization (MSVQ) [8], is used to decrease the storage and complexity of the full-search VQ. In SVQ the LSP vector is partitioned into smaller sub-vectors, and each sub-vector is quantized using a full-search VQ. SVQ is particularly adapted to LSP quantization, as the localized spectral sensitivity of the LSPs limits the spectral distortion leakage from one region to the other. MSVQ consists of a sequence of vector quantization stages, each operating on the error signal of the previous stage.

Split Multistage VQ (S-MSVQ) combines SVQ and MSVQ [8] as depicted in the shadowed box of Figure 1 (two-stage case). At the first stage, SVQ is applied, and then the residual (difference between the unquantized and quantized LSP vector) is used as input of a second SVQ stage. The notation used hereafter with this quantization scheme is illustrated with an example: In the first stage, a 16-dimensional input vector is split into two sub-vectors of dimension 5 and 11 respectively, and each of these sub-vectors is coded using 7 bits. In the second stage, the residual vector is split into five sub-vectors of dimensions 2-4-3-3-4 and these sub-vectors are coded with 5-6-6-6-5 bits respectively. A total of 42 bits is used. We refer to this case as the 42-$[(5,11)_{7,7}; (2,4,3,3,4)_{5,6,6,6,5}]$ quantization scheme.

The VQ methods discussed above are *memoryless*, as no information from previous frames is used in the quantization of the current frame, whereas *predictive* VQ [9] exploits the temporal (inter-frame) correlation of consecutive LSP vectors. We use moving average (MA) prediction, instead of auto-regressive (AR) prediction, due to its robustness to channel transmission errors. The MA predictors were calculated open-loop, using high order AR approximation of the MA process [3]. The 2-stage S-MSVQ quantization scheme including 1st order MA prediction is shown in Figure 1. Note that the mean LSP vector is removed before quantization and added to the decoded LSP. The experimental choice of the number of splits and bit allocation, in order to achieve transparency while minimizing the total number of bits and complexity, is described in Section 3.

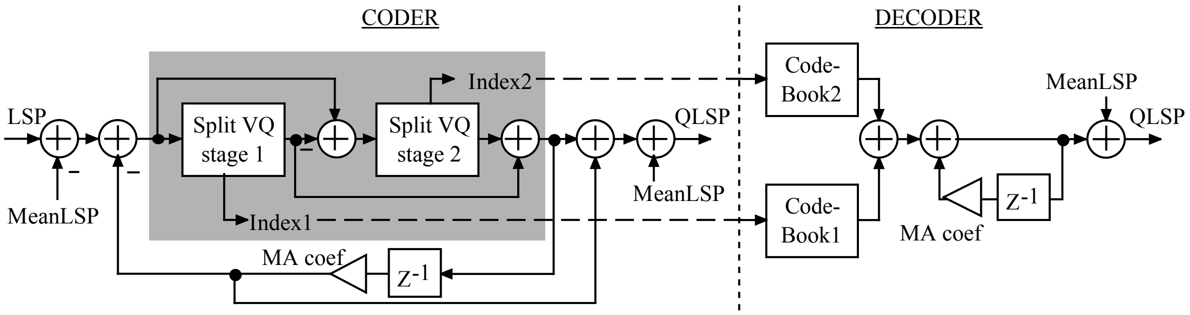

Figure 1: 2-stage S-MSVQ quantization scheme with 1^{st} order MA prediction.

3. Experimental design

The training and testing LSP databases are built up using three speech databases [3]: the TIMIT (US English), the BDSONS (French), and the ITU multi-Lingual Speech Database. The LSPs are calculated from the speech data, as it is done in our proprietary Wideband CELP coder [3]. This coder uses a full-band approach, 16 kHz sampling rate and a 20 ms frame size, with four 5 ms sub-frames. An asymmetric window is applied on a 30 ms frame: 20 ms of present speech and 10 ms of past speech. A 16^{th} order LP analysis is done and the resulting LP coefficients are transformed to LSP.

a. Determination of the optimal split and bit allocation

In previous work [3] we have used a one-stage SVQ with a fixed split of 5 codebooks with dimension 4-3-3-3-3. The optimal bit allocation to these codebooks was determined iteratively, at every iteration increasing the bit budget by one bit, allocating it tentatively to each codebook, and choosing the configuration that gives the best marginal improvement in SD. The codebooks were trained using Linde-Buzo-Gray algorithm, with Euclidean distance for training and testing [10]. Transparency is achieved with the 45-$[(4,3,3,3,3)_{12,10,9,8,6}]$ quantization scheme (#1 in Table 1). MA prediction was tested, observing considerable reduction in the SD with prediction order of one and little improvement in further increasing the prediction order. Perceptual weighting was also tested, obtaining an improvement of 0.03 dB in the average SD, at the cost of a considerable complexity increase. The procedure for optimal bit allocation was then repeated, this time with 1^{st} order MA prediction and weighting. Transparency was achieved with the 40-bit $[(4,3,3,3,3)_{11,9,8,7,5}]$ scheme (#2 in Table 1). So far we focused on minimizing the bit rate only. The work described hereafter consists in minimizing both the bit rate and the complexity, by choosing the optimal split and bit allocation to each split. Based on the results of previous work, we focus on a 40-45 bit budget, 1st order MA prediction and no weighting.

In SVQ, for a given bit-budget, the best solutions in terms of complexity are those who assign an equal number of bits to each split [11]. We found that, although not strictly true, this is true for the range of values considered in practice. Firstly we wanted to improve in complexity upon scheme #1. For a 45-bit budget, we tested all possible 5-split configurations (with at least 2 LSP per split) assigning 9 bits per split. For each configuration, training and testing was performed, choosing the best in terms of SD (#3 in Table 1). This scheme is much less complex than #1, for a slightly higher SD. We have also tested 4- and 6- split solutions using the same approach, producing schemes #4 (too complex) and #5 (less complex but not transparent at 45-bits). Here we observe the trade-off among complexity and number of splits typical of SVQ.

For a 40-bit budget and (1st order) MA prediction, we tested 5 splits with 8 bits per split, choosing the two best configurations (#6 and #7). As transparency is not achieved, we optimally allocate (to the split that gives the best marginal improvement) an extra bit, obtaining scheme #8, which achieves transparency with less complexity than #2. Tests were repeated for all combinations of 5 splits and 4*8 +1*9 bits, finding that #8 is definitely the best solution and one of the few that are transparent. We also studied the 6-split solution, obtaining scheme #9, less complex, but at the cost of two extra bits. We try then to improve (in complexity) upon #8 by using the 2-stage S-MSVQ depicted in Figure 1.

#	*Splits and bit allocation*	$\overline{SD}$ [dB]	$2<SD\leq 4$ dB [%]	$SD>4$ dB [%]	*ADD*	*MAC*	*MEM*
1	45-[$(4,3,3,3,3)_{12,10,9,8,6}$]	0.9889	0.35	0	21952	21952	21952
2	40-[$(4,3,3,3,3)_{11,9,8,7,5}$] MA, weight	0.9815	0.73	0	10976	21952	10976
3	45-[$(3,3,3,3,4)_{9,9,9,9,9}$]	1.0017	0.47	0	8192	8192	8192
4	44-[$(4,3,4,5)_{11,11,11,11}$]	0.9915	0.31	0	32768	32768	32768
5	45-[$(3,2,3,2,2,4)_{8,8,8,7,7,7}$]	1.0789	0.93	0	3072	3072	3072
6	40-[$(3,2,3,3,5)_{8,8,8,8,8}$] MA	1.0297	1.47	0	4096	4096	4096
7	40-[$(3,3,3,3,4)_{8,8,8,8,8}$] MA	1.0334	1.63	0	4096	4096	4096
8	41-[$(3,3,3,3,4)_{8,9,8,8,8}$] MA	0.9818	1.03	0	4864	4864	4864
9	43-[$(3,2,2,2,3,4)_{8,7,7,7,7,7}$] MA	0.9594	0.76	0	2432	2432	2432
10	21-[$(4,4,8)_{7,7,7}$] MA	1.9946	44.50	0.31	2048	2048	2048
11	42-[$(4,4,8)_{7,7,7}$;$(5,5,6)_{7,7,7}$] MA	0.9773	1.34	0	4096	4096	4096
12	41-[$(4,4,8)_{7,7,7}$;$(4,4,3,5)_{5,5,5,5}$] MA	1.0344	1.53	0	2560	2560	2560
13	41-[$(4,4,8)_{7,7,7}$;$(2,4,3,3,4)_{4,4,4,4,4}$] MA	1.0372	1.63	0.01	2304	2304	2304
14	42-[$(4,4,8)_{7,7,7}$;$(3,5,3,5)_{5,5,5,6}$] MA	0.9901	1.21	0	2720	2720	2720
15	42-[$(4,4,8)_{7,7,7}$;$(3,3,3,3,4)_{4,4,4,5,4}$] MA	1.0016	1.30	0	2352	2352	2352
16	43-[$(4,4,8)_{7,7,7}$;$(2,3,3,3,5)_{4,4,4,5,5}$] MA	0.9654	1.10	0	2432	2432	2432
17	28-[$(3,3,4,6)_{7,7,7,7}$] MA	1.5720	15.09	0.02	2048	2048	2048
18	42-[$(3,3,4,6)_{7,7,7,7}$; $(8,8)_{7,7}$] MA	0.9920	1.31	0	4096	4096	4096
19	43-[$(3,3,4,6)_{7,7,7,7}$; $(6,5,5)_{5,5,5}$] MA	0.9748	1.23	0	2560	2560	2560
20	30-[$(3,2,3,3,5)_{6,6,6,6,6}$] MA	1.5164	11.89	0.02	1024	1024	1024
21	42-[$(3,2,3,3,5)_{6,6,6,6,6}$; $(4,6,6)_{4,4,4}$] MA	1.0439	1.72	0	1280	1280	1280
22	44-[$(3,2,3,3,5)_{6,6,6,6,6}$; $(4,6,6)_{4,5,5}$] MA	0.9683	1.19	0	1472	1472	1472
23	43-[$(3,2,3,3,5)_{6,6,7,6,6}$; $(4,7,5)_{4,4,4}$] MA	1.0073	1.63	0	1472	1472	1472
24	42-[$(3,2,3,3,5)_{6,6,6,6,6}$; $(3,5,4,4)_{3,3,3,3}$] MA	1.0429	1.64	0.01	1152	1152	1152
25	44-[$(3,2,3,3,5)_{6,6,6,6,6}$; $(4,4,3,5)_{4,3,3,4}$] MA	0.9727	1.30	0	1224	1224	1224
26	42-[$(5,11)_{7,7}$; $(2,4,3,3,4)_{5,6,6,6,5}$] MA	0.9656	0.96	0	2880	2880	2880
27	42-[$(5,11)_{7,7}$; $(2,3,2,2,3,4)_{4,5,4,5,5,5}$] MA	0.9763	0.98	0	2496	2496	2496
28	42-[$(16)_{6}$; $(16)_{6}$; $(2,3,3,3,5)_{6,6,6,6,6}$] MA	1.0018	0.65	0	3072	3072	3072

Table 1: SD measure and complexity of the different designed quantization schemes (the complexity of the MA prediction, namely 16 ADD and 16 MULT, being not included).

If we assign an equal amount of bits to each split, the complexity depends only on the number of bits, and not on the amount or dimension of the splits. So, we should put no more than 7 bits per split in the first and second stages, to obtain schemes that are less complex than #8. We consider bit-budgets around 40-42. For the first stage we studied 2-, 3- , 4- and 5-split solutions, as more splits would make it difficult to achieve transparency with this bit-budget. We started with 3 splits of 7 bits each, in the first stage, followed by 4 splits with 5 bits or 5 splits with 4 bits, for a total of 41 bits, or 3 splits with 6-7 bits for a total of 39-42 bits. Considering all possible splits, the best solution found for the first stage is #10. Using this solution in the first stage, we found the best solution for the second stage, in the case of 3, 4, and 5 splits (#11, #12 and #13). For the 3-split case, transparency could not be achieved with 6 bits per split, and not even adding two extra bits. It was achieved with 7 bits per split (#11) but with little complexity decrease with respect to #8. In the 4-split case, to achieve transparency we tested all combinations of 4 splits and 3*5 +1*6 bits, obtaining scheme #14. In the 5-split case, we tested all combinations of 5 splits and 1*5 + 4*4 bits and 2*5 + 3*4 bits obtaining #15 and #16.

We considered 4 splits with 7 bits each in the first stage, obtaining best solution as #17. Using it, we tested different options for the second stage. The best 2-split with 7 bits each is #18. In the case of 3 splits on the 2^{nd} stage we could not achieve transparency with less than 5 bits per split (#19).

Then we tested a first stage of 5 splits of 6 bits, best result is #20. Using this solution, we tried a second stage of 3 splits with 4 bits, obtaining #21. We added up to two extra bits to this configuration, achieving transparency (at 44 bits) with #22. We also tried to increase one bit in the first stage, obtaining #23 which is basically transparent, for the same complexity of #22 and one bit less. In the case of a second stage of 4 splits with 3 bits each, we obtain configuration #24, and then #25 by adding two extra bits to achieve transparency. With a similar procedure as explained above we obtained schemes #26 and #27 (two splits of 7 bits in the first stage, followed by a 5- and 6-split second stage). Finally, we also studied a three-stage solution (#28).

The designed schemes, which are transparent and use MA prediction are given in Table 2, ordered first according to bit rate and within each bit rate according to complexity. Schemes #29 and #30 are the ones used in the WB-AMR coder (see Section 4) reported for comparison reasons. It is seen that there is a trade-off between bit-rate and complexity, as none of the (less complex) schemes has the same bit-rate of the (more-complex) scheme #8. We performed blind listening tests finding that scheme #26 yields the best quality. This scheme is for us a good compromise in quality and complexity and it was chosen for implementation in our proprietary CELP coder. The schemes with more than 1 % of outliers within 2-4 dB showed slight but noticeable distortion, suggesting that the criterion for transparency should be tightened, allowing only 1 % of outliers within 2-4 dB. Accordingly, to further reduce the complexity, at the cost of increasing the total amount of bits, schemes #9 and #22 should be used.

	#	*Splits and bit allocation*	$\overline{SD}$ [dB]	*2<SD≤4* dB [%]	*SD>4* dB [%]	*ADD*	*MAC*	*MEM*
a)	8	41-$[(3,3,3,3,4)_{8,9,8,8,8}]$ MA	0.9818	1.03	0	4864	4864	4864
	15	42-$[(4,4,8)_{7,7,7};(3,3,3,3,4)_{4,4,4,5,4}]$ MA	1.0016	1.30	0	2352	2352	2352
	27	42-$[(5,11)_{7,7};(2,3,2,2,3,4)_{4,5,4,5,5,5}]$ MA	0.9763	0.98	0	2496	2496	2496
	14	42-$[(4,4,8)_{7,7,7};(3,5,3,5)_{5,5,5,6}]$ MA	0.9901	1.21	0	2720	2720	2720
	26	42-$[(5,11)_{7,7};(2,4,3,3,4)_{5,6,6,6,5}]$ MA	0.9656	0.96	0	2880	2880	2880
	28	42-$[(16)_{6};(16)_{6};(2,3,3,3,5)_{6,6,6,6,6}]$ MA	1.0018	0.65	0	3072	3072	3072
	11	42-$[(4,4,8)_{7,7,7};(5,5,6)_{7,7,7}]$ MA	0.9773	1.34	0	4096	4096	4096
	18	42-$[(3,3,4,6)_{7,7,7,7};(8,8)_{7,7}]$ MA	0.9920	1.31	0	4096	4096	4096
	23	43-$[(3,2,3,3,5)_{6,6,7,6,6};(4,7,5)_{4,4,4}]$ MA	1.0073	1.63	0	1472	1472	1472
	9	43-$[(3,2,2,2,3,4)_{8,7,7,7,7,7}]$ MA	0.9594	0.76	0	2432	2432	2432
	16	43-$[(4,4,8)_{7,7,7};(2,3,3,3,5)_{4,4,4,5,5}]$ MA	0.9654	1.10	0	2432	2432	2432
	19	43-$[(3,3,4,6)_{7,7,7,7};(6,5,5)_{5,5,5}]$ MA	0.9748	1.23	0	2560	2560	2560
	25	44-$[(3,2,3,3,5)_{6,6,6,6,6};(4,4,3,5)_{4,3,3,4}]$ MA	0.9727	1.30	0	1224	1224	1224
	22	44-$[(3,2,3,3,5)_{6,6,6,6,6};(4,6,6)_{4,5,5}]$ MA	0.9683	1.19	0	1472	1472	1472
b)	29	46- $[(9,7)_{8,8};(3,3,3,3,4)_{6,7,7,5,5}]$ MA	0.8761	0.46	0	5280	5280	5280
	30	36- $[(9,7)_{8,8};(5,4,7)_{7,7,6}]$ MA	1.1979	2.64	0	5696	5696	5696

Table 2: a) SD measure and complexity of the different designed quantization schemes of Table 1, which are transparent and have MA prediction, ordered first according to bit rate and within each bit rate according to complexity. b) Schemes used in the WB-AMR [5].

4. SPECTRAL QUANTIZATION IN THE ETSI WB-AMR [5]

In the new ETSI WB-AMR the 16 kHz input speech signal is first decimated to 12.8 kHz, and then used to perform 16-order LPC analysis, every 20 ms, with a 30 ms window size (5 ms look-ahead and 5 ms past speech). The computed LP coefficients are converted to ISPs, and quantized using two-stage S-MSVQ, with 1st order MA prediction and no-weighting. The coder has 9 modes, corresponding to different bit rates from 6.6 to 23.85 kbps. ISP quantization uses a 46-$[(9,7)_{8,8};(3,3,3,3,4)_{6,7,7,5,5}]$ scheme in all modes except the lowest bit-rate mode, which uses a 36-$[(9,7)_{8,8};(5,4,7)_{7,7,6}]$ scheme. These schemes are reported in Table 2, together with complexity and measured SD.

An attempt of comparing our designed schemes with those of the WB-AMR is done, with the caveats that (slightly) different quantization parameters are used (ISP instead of LSP), and especially that the WB-AMR uses a 6.4 kHz bandwidth instead of 7.0 kHz, rendering the quantization task somewhat easier. To measure the WB-AMR SD, the ISP databases for training and testing were generated using the same speech material as in Section 3 and in [3]. The ISPs were calculated as in the ETSI WB-AMR, but using double precision instead of fixed-point arithmetic. We observe that all our S-MSVQ schemes have lower complexity than the WB-AMR schemes. Regarding SD performance, the comparison is difficult as our bit rates are different from the WB-AMR. A rough estimation of the WB-AMR SD obtained with linear interpolation at 42 bits results in an SD that is higher than for the "best" found (#26) scheme.

Note also that in the ETSI WB-AMR, the 4 closest vectors in the first stage (survivors) are kept, and then the search in the second stage is done for each of the survivors. Complexity would increase to 8832 (ADD, MAC and MEM) for the 46-bit scheme and 10496 (ADD, MAC and MEM) for the 36-bit scheme, for a performance improvement of 0.07 dB in SD.

5. Conclusions and future work

We have investigated the design of spectral quantization for a wideband CELP coder. Different 2-stage S-MSVQ schemes with 1st order MA prediction were designed in order to achieve transparency while minimizing total number of bits and complexity. Extensive listening tests were applied to the obtained schemes. Compared to our initial one stage SVQ solution (#2 in Table 1), the finally chosen scheme (#26) uses 2-bits more, but complexity is reduced by a factor of 5. This scheme also compares favorably to the existing schemes of the WB-AMR coder, in spite of the fact that it processes 12.8 kHz-sampled speech. The design procedure should be repeated using ISPs, to see if there is an improvement in performance.

6. Acknowledgements

This work was partly supported by the Swiss Federal Office for Education and Science under Grant OFES C00.0105 (COST 276 project).

References

[1] L. Hanzo, F. C. A. Somerville and J. P. Woodard, *Voice Compression and Communications* (Chapters 7, 4), IEEE Series on Digital & Mobile Communication, John Wiley & Sons, Inc., Publication, NY, USA, 2001.

[2] 3GPP TS 26.090 V3.1.0 (1999–12) document, in ftp://ftp.3gpp.org/Specs/2000-09/R1999/26_series/ (September 20, 2002).

[3] G. Biundo, S. Grassi, M. Ansorge, F. Pellandini, "Spectral Quantization for Wideband Speech Coding", *Proc. of 1st COST 276 Workshop on Information and Knowledge Management for Integrated Media Communication,* (CD-ROM), Leganés (Madrid), Spain, Nov. 21-23, 2001.

[4] C.-F. Chan, "Efficient quantization of LPC parameters using a mixed LSP/PARCOR representation", *Proc. EUSIPCO-94*, pp. 939-942, Edinburgh, U.K., Sept. 13-17, 1994.

[5] 3GPP TS 26.190 V5.0.0 (2001–03) document, in ftp://ftp.3gpp.org/Specs/2001-09/Rel-5/26_series/ (November 14, 2001).

[6] W. B. Kleijn and K. K. Paliwal (Editors), *Speech Coding and Synthesis* (Chapters 9, 12), Elsevier, Amsterdam, The Netherlands, 1995.

[7] M. Ferhaoui and S. Van Gerven, "LSP quantization in wideband speech coders", *Proc. of 1999 IEEE Workshop on Speech Coding*, pp. 25-27, Porvoo, Finland, June 1999.

[8] G. Guibe, H. T. How and L. Hanzo, "Comparative study of wideband speech spectral quantization schemes", *Proc. 3rd ITG Conference Source and Channel Coding*, pp. 181-186, Munich, Germany, Jan. 2000. Also in [1] (Section 4.4).

[9] J. Skoglund and J. Linden, "Predictive VQ for noisy channel spectrum coding: AR or MA?", *Proc. ICASSP-97*, Munich, Germany, Vol. 2, pp. 1351 -1354, April 1997.

[10] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design", *IEEE Transactions on Communications*, Vol. 28, pp. 84-95, Jan. 1980.

[11] K. Paliwal and S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", *IEEE Transactions on speech and audio processing*, Vol. 1, no. 1, pp. 3-14, Jan. 1993.