DESIGN OF TEST SEQUENCES FOR G.729 ANNEX E

Stéphane Ragot, Redwan Salami, and Roch Lefebvre

Department of Electrical Engineering, University of Sherbrooke,
Sherbrooke, Québec, Canada J1K 2R1

ABSTRACT

The 11.8 kb/s extension of the G.729 codec, also known as Annex E of the G.729 Recommendation, has recently been ratified by the ITU-T. The purpose of this paper is to describe how the related test sequences have been designed, using the fixed-point C simulation of the codec.

The design method is based on the concept of coverage, already used in the design of test sequences for the G.729 codec. Coverage ensures that all possible parameter values are observed in the bitstream, and all portions of the algorithm are executed at least once. Experiments showed that this approach garantees a satifying reliability.

1. INTRODUCTION

Traditionnaly, since the ADPCM, the ITU-T standardization process of a codec has included the development of test sequences in order to verify that an implementation on a given platform (e.g. DSP, VLSI, software) matches exactly the recommended algorithm. The compliance is evaluated on a bit-to-bit basis, and the codec is considered as a black box. Since an exhaustive test is intractable, the test sequences are designed to give the best possible coverage of the codec and to help in the detection of implementation errors.

The 11.8 kb/s extension [1] of the G.729 Recommendation has been designed so as to minimize the number of additions or changes to the CS-ACELP model (Conjugate-Structure Algebraic-Code-Excited Linear-Prediction) operating at 8 kb/s with 10 ms frames [2]. The only significant changes concern the linear prediction (LP) analysis and the innovation codebook. A mixed forward/backward LP structure is used. The forward mode is basically devoted to speech signals and it is the same as in G.729 while the backward mode was introduced to improve the performance for music signals. Also slight changes have been introduced to the perceptual weigthing, the postfilter, and the error concealment procedure. The 17-bit innovation codebook in G.729 is replaced by either a 35-bit codebook in LP forward mode or a 44-bit codebook in LP backward mode. This provides a better representation of the excitation and improves the performance of G.729 in case of background noise or music signals. The bit allocation depends on the LP mode and is shown in Table 1.

The paper is organized as follows. The design of the test sequences is first placed in an appropriate framework by defining several reliability criteria. The strategies which have been used to generate each sequence are then outlined. The details are not all included here, but can be found in [4]. The results of some experiments are given in order to validate the reliability of the test.

	11.8 kb/s extension		8 kb/s reference
	forward	backward	
Forward/backward mode	1 + 1 (parity)		–
Linear prediction	18	–	18
Long-term prediction delay*	8 + 1 (parity) / 5		
Innovation codebook index*	35 / 35	44 / 44	17 / 17
Codebook and pitch gains*	7 / 7		
Total	118		80

* a / b : a for subframe 1 and b for subframe 2

Table 1: Bit allocation in the G.729E and G.729 models (with a 10 ms frame).

2. DESIGN OF THE TEST SEQUENCES

2.1. Design objectives and methods

In this work, we followed a procedure similar to that used for designing the test sequences for the G.729 Recommendation, where compliance is formulated in terms of coverage. This procedure is appropriate since the G.729E model is similar to the G.729 model. The coverage is defined in three parts [3]:

- parameter coverage: for each (transmitted) parameter, all of its possible values should be observed at least once in the bitstream;
- state coverage: the test should go trough all the codec internal states;
- algorithmic coverage: each line of the algorithm should be executed at least once.

These objectives provide an efficient and rigourous framework to generate the test sequences. Yet it may not be clear whether the sequences designed to satisfy an expected coverage can be reliable when validating the implementation of the codec algorithm. Therefore their reliability needs to be checked after they have been created. Specifically, intentional modifications are inserted in the code, and the test sequences are processed through the modified codec.

An implementation is verified by applying the codec algorithm to each test sequence, and checking the compliance of the outputs. It is important to minimize the number of frames being processed while maximizing the reliability. Several strategies have been investigated. The most efficient in terms of storage requirement have been selected. The simplest is to use natural speech or music samples until coverage is ensured. It is straigthforward, however it generally proves to be very limited, due to the low variability of the speech or music parameters. A more direct way is to generate artificial input samples or an artificial bitstream.

0-7803-5651-9/99/$10.00

2.2. Parameter coverage

Full coverage of the bitstream is an unrealistic goal. For example, the G.729E coder has 118 bits per frame, which represents 2^{118} possibilities. Instead it is more reasonable to cover separately each parameter of the model. The G729E model is characterized by several parameters, which are quantized and transmitted to the decoder: the LSP (in forward mode only), the pitch delay for each subframe, the innovation codebook index and the codebook gains. A test sequence has been designed for each of these parameters, when necessary. The main idea is to leave no hole in the histogram of their respective indices or subindices. If b bits are allocated to a given parameter, at least 2^b frames or subframes are then required for full coverage.

2.2.1. LSP coverage

The prediction coefficients are transmitted only in forward mode. Ideally the codec should always stay in this mode to maximize the efficiency of the related sequence. The coverage has to be divided into several parts for pratical feasibility. The test should therefore cover totaly each LSP subindex l_0, l_1, l_2 and l_3 coded with 1, 7, 5 and 5 bits respectively. The design is actually focused on the coverage of the biggest index l_1. Indeed, if it is fully covered, the other subindices l_0, l_2 and l_3 may normally be also fully covered, since their histograms have much less values to fill. An artificial bistream has been created and decoded with the postfilter switched off to avoid spectral artifacts. The bits corresponding to the forward mode were forced for each frame. The innovation codebook index was chosen randomly to avoid stationnarity and backward mode. The values of the gains were calculated to minimize the influence of the adaptive codebook, and kept constant. The pitch indices were set to some arbitrary constant values. The LSP subindices were varied in the same way as in [3] and kept constant for 5 frames : l_0 and l_1 were taken as independant variables, the values of l_2 and l_3 were calculated by taking the modulo 32 of l_1 and the one's complement of l_1 respectively. to ensure the full coverage of each of the subindices. The other parameter indices were kept constant in the bistream.

2.2.2. Pitch coverage

The pitch delay is coded in absolute value T in subframe 1 using 8 bits and in relative value ΔT in subframe 2 using 5 bits. It makes sense to cover separately each of these delay indices. The design has been focused on the biggest one, the 8-bit absolute delay. The coverage of the other delay needed to be checked afterwards. An artificial input sequence has been generated for the pitch delays. Following [3], it consists of spacing rectangular pulses at the expected delays. However it appeared that the pitch values found by the algorithm cannot be imposed so directly. This basic principle had to be adapted. In total, 4 subsequences have been created and concatenated to achieve the expected coverage: one for the fractionnal delays $T = 0 \ldots 195$, one for the integer delays $T = 197 \ldots 255$, one for a missing relative delay $\Delta_T = 1$, and one the missing relative delay $\Delta_T = 1$. The first two subsequences have been created at 24 kHz and filtered through a linear-phase filter and decimated by 3. Delays were kept constant long enough before increasing them, to force their detection. This left only one hole in the histogram of the absolute delay. As pointed out in [3], the special value of $84\frac{2}{3}$ (at 8 kHz) or 196 when coded is seldom selected by the algorithm. A subsequence was thus developped directly at 8 kHz for this value. Variable-shape pulses have been spaced with a fixed delay of 84 samples. After covering totally the absolute delay, the coverage of the relative delay has been checked. Surprisingly its histogram had one hole at coded value $\Delta_T = 1$. Another subsequence has been created at 24 kHz positioning rectangular pulses at variable delays.

2.2.3. Innovation codebooks coverage

A white noise was generated to cover totally the range of possible pulse positions in the subindices of the forward innovation codebook. The signs of the pulses were alternated randomly.

The method that was chosen to cover the backward innovation codebook index was directly inspired by the design of the sequence devoted to the LSP index. An artificial bitstream was created and decoded without the postfilter. The backward mode was forced by repeating once each frame of bits. All the parameter indices were held constant in the bistream, except the subindices of the 44-bit innovation codebook that were chosen randomly. The value of the gains index was taken so as to minimize the contribution of the adaptive codebook. The signs of the pulses were also found to alternate randomly.

2.2.4. Gains coverage

No sequence has been generated for the gains index. It has been checked a posteriori that the whole set of test sequences ensured its full coverage.

2.3. State coverage

The G729E codec is a higly complex finite state machine, very dependent on its memory. Total coverage of its internal states is an unfeasible task, even more than the coverage of its quantized parameters. For the sake of simplicity and to reduce the storage requirement for the tests, it was assumed that the state coverage is satisfactory whenever the parameter space is well covered, as in [3]. Nonetheless, a specific sequence was added, consisting of a speech file and a regular rate switching control, because the codec can operate at two different bitrates (8 and 11.8 kb/s)

2.4. Algorithmic coverage

An additional sequence consisting of speech samples was incorporated in order to test the codec with representative signals. The algorithmic coverage was verified by a utility program, which determines how many times a line in a C source code was executed. All the lines of code that have not been executed have been identified this way. Some additional sequences have been created to reduce their number as much as possible.

The codec has been subjected to extreme situations such as errors in the bistream, overflows or stability problems. All the conditions that deal with channel errors in the decoder have been covered by an appropriate artificial bistream. The cases of single and burst errors (bad frame indication, bad mode parity or bad pitch parity) in forward/backward mode or after a voiced segment were all considered. Another artificial bistream has been created to deal with the instability of the synthesis filter. It reused one part of the design of the test sequence devoted to LSP. Once again all the quantized parameter indices were kept constant, except the

LSP subindices that were set to some specific values causing instability. Several additional conditions have been covered using artificial signals in the same way as [3]. A silence, a chirp with an exponential shape, two sinusoids with close frequencies and a sinusoid with a modulated magnitude improved the total algorithmic coverage. A last sequence produced the overflow conditions in the pitch gain calculation. It was obtained by exchanging the MSB and LSB bytes in a specific 16-bit-integer file. Note that, despite all the test sequences, 9 conditions remained unexecuted in the coder and 12 in the decoder. The coverage of static or stored data (e.g. table indices) has not been checked.

3. IMPLEMENTATION VERIFICATION

3.1. Organization of the test sequences

The sequences are a set of files, consisting of coder inputs, bitstreams and decoder outputs. All these files are stored as 16-bit (little-endian) integer files and shown in Table 2.

Files *.118/xxx : coder input *.bit : bitstream *.pst : decoder output	Test	Number of frames	Size (bytes)
lspe.118	LSP quantization	1280	204800
lspe.bit			307200
lspe.pst			204800
pitche.118	Pitch delay search	1479	236640
pitche.bit			354960
pitche.pst			236640
fixed35.118	35-bit innovation codebook search	507	81120
fixed35.bit			121680
fixed35.pst			81120
fixed44.118	44-bit innovation codebook search	1430	228800
fixed44.bit			343200
fixed44.pst			228800
speeche.118	coding/decoding of a representative signal	3750	600064
speeche.bit			900000
speeche.pst			600000
stabilit.bit	synthesis filter stability	157	37680
stabilit.pst			25120
erasuree.bit	frame erasures	47	11280
erasuree.pst			7520
algthme.118	algorithmic coverage	30	4800
algthme.bit			7200
algthme.pst			4800
ovpitch.118	overflow in the pitch gain calculation	500	80000
ovpitch.bit			120000
ovpitch.pst			80000
switch.xxx	rate switching	886	141824
switch.bit			182240
switch.pst			141760
switch (rate control)			1772

Table 2: Set of test sequences of the G.729 E codec.

3.2. Results of validation experiments

Implementation errors have been simulated, by altering some statements in the C source code. The modified codec processed the input test sequences and the outputs have been compared on a bit-to-bit basis to the reference. All simulated errors were detected, as shown in Table 3.

4. CONCLUSION

The two modes of linear prediction in the G.729E model made it difficult to control accurately the behavior of the codec. Yet the design of the related test sequences revealed the validity of the coverage approach. It may be generalized for other models.

ACKNOWLEDGMENTS

This work has been done in collaboration with the CNET (Centre National d'Étude des Télécommunications). The authors would like to thank especially Claude Lamblin.

REFERENCES

[1] ITU-T SG 16 White Contribution, "Draft description of Annex E to Recommandation G.729 - 11.8-kbit/s CS-ACELP speech coding algorithm," ITU Q. 19/16, Study Period 1997-2000, Geneva, Sept. 1998.

[2] ITU Recommandation G.729, "Coding of Speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)," Geneva, 1996.

[3] R. Pinkett and P. Kroon, "Design of Test Sequences for Implementation Verification of the G.729 CS-ACELP Speech Coder," Lucent Technologies, October 4, 1995 (Internal Document).

[4] ITU-T SG16 Delayed Contribution, "Design of Test Sequences for the G.729E codec," ITU Q. 19/16, Study Period 1997-2000, Geneva, Sept. 1998 (Source: France Telecom/ University of Sherbrooke).

Error	lspe	pitche	fixed35	fixed44	speeche	algthme	ovpitch	switch
Pre-processed signal rounded to Q14	×	×	×	×	×	×	×	×
Prediction coefficient # 10 set to zero	×	×	×	×	×	×	×	×
Expansion factor GAMMA1_0 0.98 → 0.8	×				×	×		×
Expansion factor GAMMA2_0_H 0.7 → 0.8	×				×			×
Maximum open-loop pitch delay 143 → 144	×	×	×	×	×	×	×	×
Mode set to 1 when `glob_stat` < 13000	×	×	×	×	×	×	×	×
Pitch gain rounded to Q15	×	×		×	×	×	×	×

Table 3: Detection of implementation errors using the test sequences (× = detection).