§

INTERNATIONAL TELECOMMUNICATION UNION

CCITT **G.727**

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

GENERAL ASPECTS OF DIGITAL TRANSMISSION SYSTEMS; TERMINAL EQUIPMENTS

5-, 4-, 3- AND 2-BITS SAMPLE EMBEDDED ADAPTIVE DIFFERENTIAL PULSE CODE MODULATION (ADPCM)

Recommendation G.727

§

Geneva, 1990

Printed in Switzerland

FOREWORD

permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation G.727 was prepared by Study Group XV and was approved under the Resolution No. 2 procedure on the 14th of December 1990.

CCITT NOTE

indicate both a telecommunication Administration and a recognized private operating agency.

PAGE BLANCHE

Recommendation G.727

Recommendation G.727

5-, 4-, 3- AND 2-bits SAMPLE EMBEDDED ADAPTIVE
DIFFERENTIAL PULSE CODE MODULATION (ADPCM)

1 Introduction

This Recommendation contains the specification of an embedded Adaptive Differential Pulse Code Modulation (ADPCM) algorithms with 5-, 4-, 3- and 2-bits per sample (i.e., at rates of 40, 32, 24 and 16 kbit/s). The characteristics below are recommended for the conversion of 64 kbit/s. A-law or m-law PCM channels to/ from variable rate-embedded ADPCM channels.

The Recommendation defines the transcoding law when the source signal is a pulse-code modulation signal at a pulse rate of 64 kbit/s developed from voice frequency analog signals as fully specified by Blue Book Volume, Recommendation G.711.

Applications where the encoder is aware and the decoder is not aware of the way in which the ADPCM codeword bits have been altered, or when both the encoder and decoder are aware of the ways the codewords are altered, or where neither the encoder nor the decoder are aware of the ways in which the bits have been altered can benefit from other embedded ADPCM algorithms.

2 General

The embedded ADPCM algorithms specified here are extensions of the ADPCM algorithms defined in Recommendation G.726 and are recommended for use in packetized speech systems operating according to the Packetized Voice Protocol (PVP) specified in draft Recommendation G.764.

PVP is able to relieve congestion by modifying the size of a speech packet when the need arises. Utilizing the embedded property of the algorithm described here, the least significant bit(s) of each codeword can be disregarded at packetization points and/or intermediate nodes to relieve congestion. This provides for significantly better performance than by dropping packets during congestion.

Section 3 outlines a description of the ADPCM transcoding algorithm. Figure 1/G.727 shows a simplified block diagram of the encoder and the decoder. Sections 4 and 5 provide the principles and functional descriptions of the ADPCM encoding and decoding algorithms, respectively. Section 6 contains the computational details of the algorithm. In this section, each sub-block in the encoder and decoder is precisely defined using one particular logical sequence. If other methods of computation are used, extreme care should be taken to ensure that they yield *exactly* the same value for the output processing variables. Any further departures from the processes detailed in Section 6 will incur performance penalties which may be severe.

Figure 1/G.727 = 14,5 cm

3 Embedded ADPCM algorithms

Embedded ADPCM algorithms are variable bit rate coding algorthms with the capability of bit dropping outside the encoder and decoder blocks. They consist of a series of algorithms such that the decision levels of the lower rates quantizers are subsets of the quantizer at the highest rate. This allows bit reductions at any point in the network without the need of coordination between the transmitter and the receiver. In contrast, the decision levels of the conventional ADPCM algorithms. such as those in Recommendation G.726, are not subsets of one another and therefore, the transmitter must inform the receiver of the coding rate the encoding algorithm.

Embedded algorithms can accommodate the unpredictable and bursty characteristics of traffic patterns that require congestion relief. Because congestion relief may occur after the encoding is performed, embedded coding is different from variable-rate coding where the encoder and decoder must use the same number of bits in each sample. In both cases, the decoder must be told the number of bits to use in each sample.

Embedded algorithms produce code words which contain enhancement bits and core bits. The Feed-Forward (FF) path utilizes enhancement and core bits, while the Feedback (FB) path uses core bits only. The inverse quantizer and the predictor of both the encoder and the decoder use the core bits. With this structure, enhancement bits can be discarded or dropped during network congestion. However, the number of core bits in the FB paths of both the encoder and decoder must remain the same to avoid mistracking.

The four embedded ADPCM rates are 40, 32, 24 and 16 kbit/s, where the decision levels for the 32, 24 and 16 kbit/s quantizers are sub-sets of those for the 40 kbit/s quantizer. Embedded ADPCM algorithms are referred to by (x, y) pairs where x refers to the FF (enhancement and core) ADPCM bits and y refers to the FB (core) ADPCM bits. For example, if y is set to 2 bits, (5,2) will represent the 40 kbits/s embedded algorithm, (4,2) will represent the 32 kbit/s embedded algorithm, (3,2) will represent the 24 kbit/s embedded algorithm and (2,2) the 16 kbit/s algorithm. The bit rate is never less than 16 kbit/s because the minimum number of core bits is 2. Simplified block diagrams of both the embedded ADPCM encoder and decoder are shown in Figure 1/G.727.

The Recommendation provides coding rates of 40, 32, 24 and 16 kbit/s and core rates of 32, 24 and 16 kbit/s. This corresponds to the following pairs: (5,2), (4,2), (3,2), (2,2); (5,3), (4,3), (3,3); (5,4), (4,4).

3.1 *ADPCM encoder*

Subsequent to the conversion of the A-law or m-law PCM input signal to uniform PCM, a difference signal is obtained by subtracting an estimate of the input signal from the input signal itself. An adaptive 4-, 8-, 16- or 32-level quantizer is used to assign 2, 3, 4 or 5 binary digits to the value of the difference signal for transmission to the decoder. (Not all the bits necessarily arrive at the decoder since some of these bits can be dropped to relieve congestion in the packet network. For a given received sample, however, the core bits are guaranteed arrival if there are no transmission errors and the packets arrive at destination.) FB bits are fed to the inverse quantizer. The number of core bits depends on the embedded algorithm selected. For example, the (5,2) algorithm will always contain 2 core bits. The inverse quantizer produces a quantized difference signal from these binary digits. The signal estimate is added to this quantized difference signal to produce the reconstructed version of the input signal. Both the reconstructed signal and the quantized difference signal are operated upon by an adaptive predictor which produces the estimate of the input signal, thereby completing the feedback loop.

3.2 *ADPCM decoder*

The decoder includes a structure identical to the FB portion of the encoder. In addition, there is also an FF path that contains a uniform PCM to A-law or m-law conversion. The core as well as the enhancement bits are used by the synchronous coding adjustment block to prevent cumulative distortion on synchronous tandem codings (ADPCM-PCM-ADPCM, etc., digital connections) under certain conditions (see § 5.10). The synchronous coding adjustment is achieved by adjusting the PCM output codes to eliminate quantizing distorsion in the next ADPCM encoding stage.

3.3 *One's density requirements*

These algorithms produce the all-zero code words. If requirements on one's density exist in national networks, other methods should be used to ensure that this requirement is satisfied.

3.4 *Applications*

In the anticipated application with G.764, the Coding Type (CT) field and the block Dropping Indicator (BDI) field in the packet header defined in G.764 will inform the coder of what algorithm to use. For all other applications, the information that PVP supplies must be made known to the decoder.

4 ADPCM encoder principles

Figure 2/G.727 is a block schematic of the encoder. For each variable to be described, *k is* the sampling index and samples are taken at 125 ms intervals. A description of each block is given in §§ 4.1 to 4.9.
Fig. 2/G.727 = 9,5 cm

4.1 *Input PCM format conversion*

This block converts the input signal *s*(*k*) from A-law or m-law PCM to a uniform PCM signal *sl*(*k*).

4.2 *Difference signal computation*

This block calculates the difference signal *d*(*k*) from the uniform PCM signal *sl*(*k*) and the signal estimate *se*(*k*).

µ (4-1)

§

4.3 *Adaptive quantizer*

A 4-, 8-, 16- or 32-level non-uniform mid-rise adaptive quantizer is used to quantize the difference signal $d(k)$. Prior to quantization, $d(k)$ is converted to a base 2 logarithmic representation and scaled by $y(k)$ which is computed by the scale factor adaptation block. The normalized input/output characteristic (infinite precision values) of the quantizer is given in Tables 1/G.727 through 4/G.727 for the 16, 24, 32 and 40 kbit/s algorithms, respectively. Two, three, four or five binary digits are used to specify the quantized level representing $d(k)$ (the most significant bit represents the sign bit and the remaining bits represent the magnitude). The 2-, 3-, 4- or 5-bit quantizer output $I(k)$ forms the 16, 24, 32 or 40 kbit/s output signal and is also fed to the bit-masking block. $I(k)$ includes both the enhancement and core bits.

4.4 *Bit masking*

This block produces the core bits $Ic(k)$ by logically right-shifting the input signal $I(k)$ so as to mask the maximum droppable (least significant) bits. The number of bits to mask and the number of places to right shift depend on the embedded algorithm selected. For example, this block will mask the two least significant bits (LSB's) and shift the remaining bits two places to the right when the (4,2) algorithm is selected. The output of the bit-masking block $Ic(k)$ is fed to the inverse adaptive quantizer, the quantizer scale factor adaptation and the adaptation speed control blocks.

µTABLE 1/G.727

Quantizer normalized input/output
Characteristic for 16 kbit/s embedded operation

Normalized quantizer input range log2½*d*(*k*)½— *y*(*k*)	½ *I*(*k*)½ ½ *Ic*(*k*)½	Normalized quantizer output log2½*dq*(*k*)½— *y*(*k*)
(-¥, 2.04)	0	0.91
[2.04, ¥)	1	2.85

Note — In Tables 1/G.727 through 4/G.727, “[” indicates that the endpoint value is included in the range, and “(” or “)” indicates that the endpoint value is excluded from the range.

§

µTABLE 2/G.727

Quantizer normalized input/output
Characteristic for 24 kbit/s embedded operation

Normalized quantizer input range log2½*d*(*k*)½— *y*(*k*)	½ *I*(*k*)½ ½ *Ic*(*k*)½	Normalized quantizer output log2½*dq*(*k*)½— *y*(*k*)

§

µTABLE 3/G.727

Quantizer normalized input/output
Characteristic for 32 kbit/s embedded operation

Normalized quantizer input range log2½*d*(*k*)½— *y*(*k*)	½ *I*(*k*)½ ½ *Ic*(*k*)½	Normalized quantizer output log2½*dq*(*k*)½— *y*(*k*)
(-¥, -0.05)	0	-1.06
[-0.05, 0.96)	1	-0.53
[0.96, 1.58)	2	-1.29
[1.58, 2.04)	3	-1.81
[2.04, 2.42)	4	-2.23
[2.42, 2.78)	5	-2.59
[2.78, 3.16)	6	-2.95
[3.16, ¥)	7	-3.34

§§4.5

The inverse quantizer uses the core bits to compute a quantized version *dq*(*k*) of the difference signal using the scale factor *y*(*k*) and Table 1/G.727, 2/G.727, 3/G.727 or 4/G.727 and then taking the antilog to the base 2 of the result. The estimated difference *se*(*k*) is added to *dq*(*k*) to produce the reconstructed version *sr*(*k*) of the input signal. Table 1/G.727, 2/G.727, 3/G.727 or 4/G.727 will be applicable only when there are 2, 3, 4 or 5 bits, respectively, in the FF path.

4.6 *Quantizer scale factor adaptation*

This bloc computes *y*(*k*), the scaling factor for the quantizer and the inverse quantizer. (The scaling factor *y*(*k*) is also fed to the adaptation speed control block.) The inputs are the bit-masked output *Ic*(*k*) and the adaptation speed control parameter a*l*(*k*).

The basic principle used in scaling the quantizer is bimodal adaptation:

- fast for signals (e.g., speech) that produce difference signals with large fluctuations,
- slow for signals (e.g., voiceband data, tones) that produce difference signals with small fluctuations.

The speed of adaptation is controlled by a combination of fast and slow scale factors.

The fast (unlocked) scale factor *yu*(*k*) is recursively computed in the base 2 logarithmic domain from the resultant logarithmic scale factor *y*(*k*):

µ (4-2)

§

where

yu(*k*) is limited by 1.06 £ *yu*(*k*) £ 10.00.

For 2-core-bit operation (1 sign bit), the discrete function *W*[*Ic*(*k*)] is defined as follows (infinite precision values):

§§

µTABLE 16/G.727

Quantizer output levels and 2-bit outputs

ENCODER

DSX

IS

A-law

µTABLE 15/G.727

DLNX

40 kbit/s quantizer output levels

IMAG

m-law

PCM code word SP

DQSFF

Comparison of

IDQ+IM

½ *Ic*(*k*) ½

Number of bits

Description

m-law

SP

W[*Ic*(*k*)]

Input

Input

Input

— ½ Positive portion of decision interval

Output

— —½ Positive portion of interval

S

LAW

— ½ Negative portion of decision interval

R (optional)

½ *Ic*(*k*) ½

— —½ Negative portion of interval

I

ID + IM

— ½ Negative portion of decision interval

— —½ Negative portion of interval

— ½ Negative portion of decision interval

PCM law select, 0 = m-law, 1 = A-law

— —½ Positive portion of interval

Reset

C = core bits, E = Enhancement bits

— ½ Negative portion of decision interval

— ½ Positive portion of decision interval

DECODER

— ½ Positive portion of decision interval

§

µTABLE 4/G.727

Quantizer normalized input/output
Characteristic for 40 kbit/s embedded operation

Normalized quantizer input range log2½*d*(*k*)½— *y*(*k*)	½ *I*(*k*)½	Normalized quantizer output log2½*dq*(*k*)½— *y*(*k*)
(-¥, -1.05)	~~10~~	-2.06
[-1.05, -0.05)	~~11~~	~~-0.48~~
[-0.05, 0.54)	~~1~~2	~~-0.27~~
[0.54, 0.96)	~~13~~	-0.76
[0.96, 1.30)	~~14~~	-1.13
[1.30, 1.58)	~~15~~	-1.44
[1.58, 1.82)	~~16~~	-1.70
[1.82, 2.04)	~~1~~7	-1.92
[2.04, 2.23)	18	