5i'

7.2 Coding of analogue signals by methods other than PCM

Recommendation G.721

32 kbit/s ADAPTIVE DIFFERENTIAL PULSE CODE MODULATION (ADPCM)

(Melbourne, 1988)

1 General

The characteristics below are recommended for the conversion of a 64 kbit/s A-law or μ-law PCM channel to and from a 32 kbit/s channel. The conversion is applied to the PCM bit stream using an ADPCM transcoding technique. The relationship between the voice frequency signals and the PCM encoding/decoding laws is fully specified in Recommendation G.711.

Paragraphs 1.1 and 1.2 of this Recommendation provide an outline description of the ADPCM transcoding algorithm, §§ 2 and 3 provide the principles and functional descriptions of the ADPCM encoding and decoding algorithms respectively, whilst § 4 is the precise specification for the algorithm computations. Networking aspects and digital test sequences are addressed in Apendices I and II respectively to this Recommendation.

Simplified block diagrams of both the ADPCM encoder and decoder are shown in Figure 1/G.721.

In § 4, 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 § 4 will incur performance penalties which may be severe.

Note 1 — For the time being, the 32 kbit/s ADPCM algorithm defined in this Recommendation is intended for transmission purposes since switching applications at this bit rate are a subject for further study by the CCITT.

Note 2 — Prior to the definition of this Recommendation, other 32 kbit/s ADPCM algorithms of similar performance have been incorporated in equipment designs and used in national telecommunications networks.

Note 3 — In the short term, due to the limited availability of 32 kbit/s ADPCM equipment, the use of 32 kbit/s ADPCM in the international network, when requested by one of the Administrations concerned, will require bilateral and/or multilateral agreement.

Note 4 — Signalling and multiplexing considerations are beyond the scope of this Recommendation (see for example Recommendation G.761).

1.1 *ADPCM encoder*

Subsequent to the conversion of the A-law or μ-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 adaptative 15-level quantizer is

---

This Recommendation G.721 completely replaces the text of Recommendation G.721 published in Fascicle III.3 of the *Red Book* . It should be noted that systems designed in accordance with the present Recommendation will not be compatible with systems designed in accordance with the *Red Book* version.

used to assign four binary digits to the value of the difference signal for transmission to the decoder. An inverse quantizer produces a quantized difference signal from these same four 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

1.2 *ADPCM decoder*

The decoder includes a structure identical to the feedback portion of the encoder, together with a uniform PCM to A-law or μ-law conversion and a synchronous coding adjustment

The synchronous coding adjustment prevents cumulative distortion occurring on synchronous tandem codings (ADPCM-PCM-ADPCM, etc. digital connections) under certain conditions (see § 3.7). The synchronous coding adjustment is achieved by adjusting the PCM output codes in a manner which attempts to eliminate quantizing distortion in the next ADPCM encoding stage

Figure 1/G.721, p.

2 ADPCM encoder principles

Figure 2/G.721 is a block schematic of the encoder. For each variable to be described, k is the sampling index and samples are taken at 125 μs intervals. A fundamental description of each block is given below in §§ 2.1 to 2.8.

Figure 2/G.721, p.

2.1 *Input PCM format conversion*

This block converts the input signal $s(k)$ from A-law or μ-law PCM to a uniform PCM signal $s_1(k)$.

2.2 *Difference signal computation*

This block calculates the difference signal $d(k)$ from the uniform PCM signal $s_l(k)$ and the signal estimate $s_e(k)$:

$$d(k) = s_l(k) - s_e(k) \tag{2-1}$$

2.3 *Adaptive quantizer*

A 15-level non-uniform 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 Table 1/G.721. Four binary digits are used to specify the quantized level representing $d(k)$ (three for the magnitude and one for the sign). The 4-bit quantizer output $I(k)$ forms the 32 kbit/s output signal; it is also fed to the inverse adaptive quantizer, the adaptation speed control and the quantizer scale factor adaptation blocks.

2.4 *Inverse adaptive quantizer*

A quantized version $d_q(k)$ of the difference signal is produced by scaling, using $y(k)$, specific values selected from the normalized quantizing characteristic given in Table 1/G.721 and then transforming the result from the logarithmic domain.

H.T. [T1.721]

TABLE 1/G.721

Quantizer normalized input/output characteristic

<table>
<tr><th>{ Normalized quantizer input range log | fId (k | (emy (k) } Normalized quantizer output log | fId q (k | (emy (k) }</th><th>| fII (k |</th><th>{</th></tr>
<tr><td>[3.12, +∞)</td><td>7</td><td>3.32</td></tr>
<tr><td>[2.72, 3.12)</td><td>6</td><td>2.91</td></tr>
<tr><td>[2.34, 2.72)</td><td>5</td><td>2.52</td></tr>
<tr><td>[1.91, 2.34)</td><td>4</td><td>2.13</td></tr>
<tr><td>[1.38, 1.91)</td><td>3</td><td>1.66</td></tr>
<tr><td>[0.62, 1.38)</td><td>2</td><td>1.05</td></tr>
<tr><td>[—0.98, 0.62)</td><td>1</td><td>0.031</td></tr>
<tr><td>(—∞, —0.98)</td><td>0</td><td>—∞</td></tr>
</table>

Note — The convention used here is that ‘‘[’’ indicates that the endpoint value is included in the range, ‘‘)’’ indicates that the endpoint value is excluded from the range.

Table 1/G.721 [T1.721], p.

2.5 *Quantizer scale factor adaptation*

This block computes $y(k)$, the scaling factor for the quantizer and the inverse quantizer. The inputs are the 4-bit quantizer output $I(k)$ 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 $y_u(k)$ is recursively computed in the base 2 logarithmic domain from the resultant logarithmic scale factor $y(k)$:

$$y_u(k) = (1 - 2^{-5})y(k) + 2^{-5} W[I(k)], \qquad (2\text{-}2)$$

where $y_u(k)$ is limited by 1.06 $y_u(k)$ 10.00.

The discrete function $W(I)$ is defined as follows (infinite precision values):

H.T. [T2.721]

| (k |	7	6	5	4	3	2	1	0
W(I)	70.13	22.19	12.38	7.00	4.00	2.56	1.13	—0.75

Table [T2.721], p.

The factor (1 — 2^D lF261^5) introduces finite memory into the adaptive process so that the states of the encoder and decoder converge following transmission errors.

The slow (locked) scale factor $y_l(k)$ is derived from $y_u(k)$ with a low pass filter operation:

$$y_l(k) = (1 - 2^{-6})\, y_l(k-1) + 2^{-6}\, y_u(k) \qquad (2\text{-}3)$$

The fast and slow scale factors are then combined to form the resultant scale factor:

$$y(k) = a_l(k)\, y_u(k-1) + [1 - a_l(k)]\, y_l(k-1), \qquad (2\text{-}4)$$

where $0 \le a_l(k) \le 1$ (see § 2.6).

2.6 *Adaptation speed control*

The controlling parameter $a_l(k)$ can assume values in the range [0, 1]. It tends towards unity for speech signals and towards zero for voiceband data signals and tones. It is derived from a measure of the rate-of-change of the difference signal values.

Two measures of the average magnitude of $I(k)$ are computed:

$$d_{ms}(k) = (1 - 2^{-5})\, d_{ms}(k-1) + 2^{-5}\, F[I(k)], \qquad (2\text{-}5)$$

and

$$d_{ml}(k) = (1 - 2^{-7})\, d_{ml}(k-1) + 2^{-7}\, F[I(k)], \qquad (2\text{-}6)$$

where $F[I(k)]$ is defined by

H.T. [T3.721]

$\lvert I(k) \rvert$	7	6	5	4	3	2	1	0
F[I(k)]	7	3	1	1	1	0	0	0

Table [T3.721], p.

$d_{ms}(k)$ is thus a relatively short term average of $F[I(k)]$ and $d_{ml}(k)$ is a relatively long term average of $F[I(k)]$.

Using these two averages, the variable $a_p(k)$ is defined:

$$
a_p(k) = \begin{cases} (1 - 2^{-4})a_p(k-1) + 2^{-3}, & \text{if } \left| d_{ms}(k) - d_{ml}(k) \right| \geq 2^{-3} d_{ml}(k) \\ (1 - 2^{-4})a_p(k-1) + 2^{-3}, & \text{if } y(k) < 3 \\ (1 - 2^{-4})a_p(k-1) + 2^{-3}, & \text{if } t_d(k) = 1 \\ 1, & \text{if } t_r(k) = 1 \\ (1 - 2^{-4})a_p(k-1), + 2^{-3} & \text{otherwise} \end{cases} \qquad (2\text{-}7)
$$

Thus, $a_p(k)$ tends towards the value 2 if the difference between $d_{ms}(k)$ and $d_{ml}(k)$ is large (average magnitude of $I(k)$ changing) and $a_p(k)$ tends towards the value 0 if the difference is small (average magnitude of $I(k)$ relatively constant). $a_p(k)$ also tends towards 2 for an idle channel (indicated by $y(k) < 3$) or partial band signals (indicated by $t_d(k) = 1$ as described in § 2.8). Note that $a_p(k)$ is set to 1 upon detection of a partial band signal transition (indicated by $t_r(k) = 1$, see § 2.8).

$a_p(k-1)$ is then limited to yield $a_l(k)$ used in Equation (2-4) above:

a

[Formula Deleted]