# An Efficient Scheme for Lossy Realtime Audio Data Compression

3922 (O-3)

Robert C. Maher
University of Nebraska–Lincoln
Lincoln, NE 68588-0511, USA

**Presented at the 97th Convention 1994 November 10–13 San Francisco**

AES



**AN AUDIO ENGINEERING SOCIETY PREPRINT**

# An Efficient Scheme for Lossy Realtime Audio Data Compression

Robert C. Maher
Department of Electrical Engineering
and
Center for Communication and Information Science
University of Nebraska-Lincoln
209N WSEC, Lincoln, NE 68588-0511
Voice: (402) 472-2081 Fax: (402) 472-4732 Internet: rmaher@unl.edu

**Abstract:**

The increasing use of digital audio in personal computers indicates an ongoing need for computationally efficient audio data compression. An implementation of adaptive differential pulse-code modulation incorporating a recursively-indexed quantizer (RIQ-DPCM) is presented as an alternative method for efficient audio signal compression on general-purpose computers. The performance characteristics of the RIQ-DPCM procedure are compared with other standard coding techniques of similar computational complexity, and several realtime implementation issues are discussed.

## 0. INTRODUCTION

Audio signal compression is useful in many situations where it is necessary to store or transmit digitized speech and music. Unfortunately, most general-purpose *lossless* data compression schemes used for general computer files perform relatively poorly on digital audio signals. However, it is possible to develop *lossy* compression schemes which retain much of the perceptual quality of the original signal. Existing lossy audio data compression systems specifically intended for high quality mono or stereo applications (e.g., MUSICAM, ATRAC, Dolby AC-2) can achieve approximately 6:1 compression with results nearly indistinguishable from the original signal to human listeners [1]. This nearly transparent compression at a data rate below 128 kbits/s/channel requires considerable computational resources and/or purpose-built hardware. Thus, in situations where the available computational capability is insufficient to support the goal of transparent coding one must consider the inevitable tradeoffs among bit rate, distortion, and computation.

In this paper the implementation and testing of a low-complexity digital audio coder are described. The coder is a recent variant of the well-known Differential Pulse-Code·Modulation (DPCM) framework. The new coder incorporates a Recursively-

Indexed Quantizer (RIQ), which eliminates the problem of quantizer overload. First, a brief introduction to conventional DPCM-based coders is presented, followed by a description of the RIQ concept. Several implementation issues are discussed next, along with a subjective and objective performance comparison of the RIQ-DPCM technique.

## 1. LOW-COMPLEXITY CODING VIA DPCM

A common choice for lossy audio data compression in computationally constrained situations is differential pulse-code modulation (DPCM) [2]. DPCM obtains data compression by exploiting the observation that the sequence of adjacent sample *differences* in the audio data stream often has a smaller variance than the sample sequence itself, meaning that fewer bits are needed to maintain some specified average distortion behavior. This so-called *source memory* is a form of signal redundancy that can be removed.

### 1.1 DPCM Structure

The conventional DPCM encoder - decoder pair is shown in Figure 1. The encoder contains a quantizer and a linear predictor. The linear predictor (which can be as simple as a unit sample delay) creates an estimate for the "next" input sample based on knowledge of the previous quantized output values. This predicted value, $\hat{x}[m]$, is subtracted from the input sample, $x[m]$, resulting in a difference signal $e[m]$ which is quantized ($\tilde{e}[m]$) to a fewer number of bits and stored or transmitted to the decoder.

The DPCM decoder incorporates the same linear predictor as the encoder. Assuming no transmission or storage errors for $\tilde{e}[m]$, the decoder reconstructs $\hat{x}[m]$, the same approximation of the input signal used in the encoder.

### 1.2 Distortion in DPCM Systems

Typical DPCM implementations suffer from two types of quantization distortion: *granulation* and *overload*. Granulation refers to the small difference between the reconstructed signal sample and the input sample under normal operating conditions where the inter-sample difference falls within the inner levels of the quantizer (no quantizer clipping). Granulation distortion is bounded in amplitude by the quantizer step size, $\Delta$, and is typically perceived as broadband, low-level background noise. In general, reducing $\Delta$ reduces the amount of granulation distortion, so it is desirable to keep $\Delta$ as small as possible.

Overload distortion, on the other hand, occurs when the sample-to-sample difference is large enough to cause the quantizer to clip. The effects of overload can be much larger than granulation distortion since overload is not bounded by $\Delta$. Overload distortion can also tend to linger for many samples because the system output is fed back to predict the next input sample: the clipped output signal causes the prediction signal to be unreliable, possibly leading to subsequent overloads.

### 1.3 Rate-Distortion Tradeoff

The rate-distortion tradeoff is a fundamental aspect of communications theory. In the context of DPCM, it is possible to choose two parameters: the quantizer step size, $\Delta$, and the number of quantizer steps, $2N+1$. The $2N+1$ quantizer steps can be encoded with $\log_2(2N+1)$ bits per quantizer output symbol. A small value of $\Delta$ is advantageous to reduce the amount of granulation distortion. However, for a given $N$, the reduction in $\Delta$ increases quantizer overload distortion, since the maximum unclipped range of the quantizer $[-N\Delta, N\Delta]$ shrinks as $\Delta$ is reduced. It is possible to increase $N$ to compensate for the smaller value of $\Delta$, but this requires an increase in the number of bits per symbol. Thus, a design decision must be made either to determine the required bit rate necessary to accommodate an allowable level of distortion, or to determine the worst-case distortion for an allowable channel capacity. In many practical cases it is the latter situation--fixed channel capacity--that must be addressed.

Since the precise statistical behavior of the musical input signal usually varies with time, it is convenient to be able to adjust $\Delta$ automatically to optimize the short-term distortion performance. Therefore, the basic DPCM concept can be improved by including an adaptive linear predictor and an adaptive quantizer (ADPCM). Although ADPCM does not usually incorporate a full-fledged perceptual model, the "quality-to-computation" ratio is favorable in computationally constrained applications [2].

## 2. RECURSIVELY-INDEXED QUANTIZER

For a fixed number of quantizer steps and a fixed number of symbols per sample, it is only possible to reduce quantizer overload distortion by increasing the step size $\Delta$ -- at the expense of a corresponding increase in granulation distortion. However, in situations that can accommodate a variable number of symbols per sample (i.e., a variable bit rate), it is also possible to reduce overload distortion by detecting quantizer clipping and instantaneously increasing $N$ to compensate. As long as the likelihood of quantizer overload is relatively low, the increase in average bit rate should be minimal. This is the basic idea of the Recursively-Indexed Quantizer (RIQ).

## 2.1 RIQ in a Fixed-Length Coder

RIQ provides a guaranteed distortion level on a *per sample* basis allowing an increase in bit rate when the quantizer encounters large magnitude inputs. A formal description of RIQ is [3, 4]:

---

For a given quantizer step size $\Delta$ and a positive integer $K$, the smallest and largest quantizer output levels, $x_L$ and $x_H$, are defined:

$$x_L = -\left\lfloor \frac{K-1}{2} \right\rfloor \Delta, \qquad x_H = x_L + (K-1)\Delta$$

where $\lfloor y \rfloor$ is the largest integer not exceeding $y$. The quantizer defined this way is a uniform quantizer with step size $\Delta$ and always has zero as one of the output levels (mid-tread).

The recursive aspect of the quantization rule $Q$ is defined:

For a given input value $x$:

(i) If $x$ falls in the interval $(x_L + (\Delta/2), x_H - (\Delta/2))$, then $Q(x)$ is simply the output level nearest to $x$.

(ii) If $x$ is greater than $x_H - (\Delta/2)$, determine whether the difference $x_1 \equiv (x - x_H) \in (x_L + (\Delta/2), x_H - (\Delta/2))$.
If so, the quantized representation $Q(x) = (x_H, Q(x_1))$.
If not, form the difference $x_2 = x - 2x_H$, then do the same as for $x_1$.

This recursive process continues until for some $m$, $x_m = x - m\,x_H$ falls in the interval $(x_L + (\Delta/2), x_H - (\Delta/2))$, in which case the quantized representation becomes the sequence $Q(x) = \left(\underbrace{x_H, x_H, \cdots, x_H}_{m}, Q(x_m)\right)$.

(iii) If $x$ is smaller than $x_L + (\Delta/2)$, a similar procedure is used: $x_m = x - m\,x_L$ is formed so that it falls in the interval $(x_L + (\Delta/2), x_H - (\Delta/2))$, and is quantized into the sequence $(x_L, x_L, \cdots, x_L, Q(x_m))$.

---

The resulting RIQ output behavior is depicted in **Figure 2.** The number of quantizer output symbols required to represent the input sample increases as the absolute value of the input sample increases. Thus, the RIQ performs best (lowest output bit rate) for centrally distributed input density functions, i.e., quantizer input signals which are most likely to be near zero. This is precisely the characteristic of the quantizer input in a DPCM system when the input signal has memory.

2.2 RIQ-DPCM Formulation

RIQ-DPCM is simply a conventional DPCM structure with a recursively-indexed quantizer. The output of the RIQ exhibits no overload distortion (only granular quantization error), but with a variable bit rate. Thus, the subsequent processing systems must be able to handle the variable number of symbols per sample without losing synchronization.

Sayood and Na [3] have studied the behavior of the RIQ-DPCM system with first-order Gauss-Markov and Laplace-Markov input sources. The RIQ-DPCM system was found to perform at or close to the optimum entropy constrained DPCM system, yet without the relatively complex iterative procedure typically required for quantizer design [5].

The available coder parameters for basic RIQ-DPCM are the quantizer step size, $\Delta$, and the number of quantizer levels, $2N+1$. The relationship between $N$ and the output bit rate, $R$, for a fixed binary encoder can be represented by [4]

$$R = \zeta \cdot \lceil \log_2(2N+1) \rceil,$$

where $\zeta$ is the average number of output symbols per input sample, and $\lceil y \rceil$ indicates the smallest integer greater than $y$. The value of $\zeta$ is determined by the likelihood that the quantizer operates in the recursive mode, while the `log` factor is determined by the number of quantizer levels. Note that as $N$ is increased the `log` factor increases, but the base range of the quantizer expands (for a fixed value of $\Delta$), reducing the likelihood of quantizer recursion, which in turn reduces $\zeta$. Conversely, as $N$ is decreased the `log` factor decreases, but the base range of the quantizer shrinks, causing an increase in $\zeta$. Thus, it is clear that one of the RIQ-DPCM design criteria is to choose the optimum value for $N$ so that the output bit rate will be minimized for a given value of $\Delta$. An alternative design criterion is to choose $N$ so that the quantizer step size $\Delta$ is minimized (lowest distortion) for a given bit rate.

The analytical relationship for RIQ-DPCM among rate, $N$, and $\Delta$ is shown in **Figure 3** for quantizer input described by a laplacian distribution. The "rate" surface defined in the figure shows that (a) for a fixed value of N, the rate decreases monotonically as $\Delta$ increases, (b) for a fixed value of $\Delta$, there is a particular value of N for which the rate can be minimized, and (c) for a fixed rate there is a particular value of N for which $\Delta$ can be minimized. There is no global minimum, of course, so it is not possible to simultaneously minimize rate and distortion.

An empirical example based on the RIQ-DPCM performance results for an actual music signal is shown in **Figure 4**. Note in particular that a minimum is evident (near $2N + 1 = 29$). Also note, however, that the rate curve is very shallow above the minimum, meaning that the RIQ-DPCM rate penalty for overestimating the number of quantizer levels is less than the penalty for underestimating it [6].

The performance of the RIQ-DPCM structure can be enhanced by the inclusion of an adaptive predictor and an adaptive quantizer in order to compensate for changes in the characteristics of the input signal. This allows the tradeoff between average rate and average distortion to be adjusted on a segmental basis, thereby improving the perceptual quality of the reconstructed signal.

## 3. AN EXAMPLE IMPLEMENTATION SCENARIO

A simple implementation was used in order to investigate the performance and behavior of the RIQ-DPCM system in a low-complexity context. In this example a simple adaptive method was used for the quantizer, while the "predictor" consisted of a unit sample delay. The quantizer adaptation rule was to multiply the step size $\Delta$ by the factor 1.5 if the previous quantizer input was large enough to require recursion, and to multiply $\Delta$ by the factor 0.9 if the previous quantizer input fell within the base range of the quantizer. More sophisticated techniques are possible, of course, but the goal here was to limit the required computation.

The simple adaptive RIQ-DPCM algorithm was implemented using an Ariel Corp. DSP-56 board equipped with a Motorola 56001 DSP processor (27 MHz), supporting a simultaneous stereo encode/decode in realtime (50 kHz sample rate per channel).

The DSP software was coded directly in 56000 assembly language, while attempting to follow the straight-forward block diagram as much as possible. In other words, no heroic efforts were needed to optimize the code for memory or execution speed efficiency.

In the formal RIQ definition each recursion requires a subtraction and a comparison operation. Since the number of recursions is not known in advance, the repeated subtraction-comparison operations require an unknown execution time. Instead, the basic RIQ formulation was implemented using a multiple-instruction integer divide operation since the execution time required for the division steps was constant no matter how many implicit recursions were needed.

The RIQ-DPCM output stream differs from a conventional DPCM system because of the variable number of output symbols per input sample. This means that sufficient memory must be provided for buffering the output data stream. In practice, the required amount of buffer memory is not a significant problem because the likelihood of multiple recursions is generally quite low. A typical example histogram showing the number of recursions per input sample is shown in **Figure 5**.

The variable number of output symbols per sample has other practical implications for the system. For example, if the output data stream is stored in a file and it is necessary to locate a particular time index, all of the preceding data must be examined to account for the variable number of samples represented by each byte of data. Similarly, there is no way to predict exactly what size the output data file will be for an arbitrary input signal. These problems can be overcome by carefully choosing the format of the output data stream, but they do indicate some additional processing overhead for editing operations.

The signal reconstruction process is trivial: the sequence of coded symbols is simply summed (accumulated) to generate the audio output sequence.

## 4. SUBJECTIVE COMPARISON WITH CODERS OF SIMILAR COMPLEXITY

A formal test was carried out to compare the subjective quality of RIQ-DPCM to three other low-complexity DPCM-based coding techniques. The other techniques were: basic DPCM, DPCM with noise feedback coding (NFC), and adaptive quantizer DPCM (DPCM-AQB) [2].

The test involved a set of A-B pairs comprising four audio signals (solo castanets, orchestra, solo soprano vocal, and rock and roll), four coder bit rates, and four Modulation Noise Reference Unit (MNRU) signals for each coder [7, 8]. The results of the test are summarized in **Figure 6**, averaged over the four audio input signals.

In all cases tested the RIQ-DPCM system was subjectively rated 2 to 5 dB higher than any of the other coders of similar computational complexity. This result is significant and encouraging because it indicates nearly one bit of quality improvement for a given bit rate when using RIQ-DPCM. Therefore, the no-overload property of RIQ-DPCM appears to be a valuable subjective characteristic for low-complexity audio data compression [9].

## 5. CONCLUSION

In this paper a low-complexity audio data compression scheme has been presented, including several practical implementation issues and subjective test results. The compression method is based on a conventional Differential Pulse-Code Modulation (DPCM) framework incorporating a Recursively-Indexed Quantizer (RIQ). The procedure is of similar computational complexity to other DPCM-based schemes, but the inclusion of the RIQ technique prevents quantizer overload distortion in the output data stream, albeit at the expense of a variable output bit rate.

In practice it has been found that RIQ-DPCM provides approximately one bit of subjective quality improvement (2 to 5 dB) at a given bit rate, compared to DPCM-based coders of similar sophistication.

## 6. ACKNOWLEDGMENTS

The work described in this paper was supported in part by the University of Nebraska Research Council, the Center for Communication and Information Science, and the Department of Electrical Engineering. The author would like to acknowledge the significant contributions of his students, Stella Joseph and Ram Peddibhotla, and his colleague and mentor, Khalid Sayood.

## 7. REFERENCES

[1] J. D. Johnston and K. Brandenburg, "Wideband Coding--Perceptual Considerations for Speech and Music," in Advances in Speech Signal Processing, S. Furui and M. M. Sondhi, eds., New York: Marcel Dekker, 1992.

[2] N. S. Jayant and P. Noll, Digital Coding of Waveforms, Englewood Cliffs, NJ: Prentice-Hall, 1984, pp. 109-140.

[3] K. Sayood and S. Na, "Recursively indexed quantization of memoryless sources," *IEEE Trans. Inf. Theory*, vol. 38, no. 5 (1992), pp. 1602-1609.

[4] K. Sayood and S. Na, "Recursively indexed differential pulse code modulation," *Joint DIMACS/IEEE Workshop on Coding and Quantization*, Rutgers Univ., Piscataway, NJ, Oct. 1992.

[5] N. Farvardin and J. W. Modestino, "Rate-distortion performance of DPCM schemes for autoregressive sources," *IEEE Trans. Inf. Theory*, vol. IT-31 (1985), pp. 402-418.

[6] R. Peddibhotla, "A low complexity approach to audio compression," M.S. Thesis, Electrical Engineering, University of Nebraska-Lincoln, 1994.

[7] CCITT (International Telegraph and Telephone Consultative Committee), Blue Book 5, 1989, pp. 198-203, 341-358.

[8] S. Joseph, "Subjective and objective evaluation of different digital coders," M.S. Thesis, Electrical Engineering, University of Nebraska-Lincoln, 1994.

[9] R. C. Maher, "Computationally efficient compression of audio material by means of RIQ-DPCM," *Proc. 1993 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, October, 1993, Mohonk, NY.

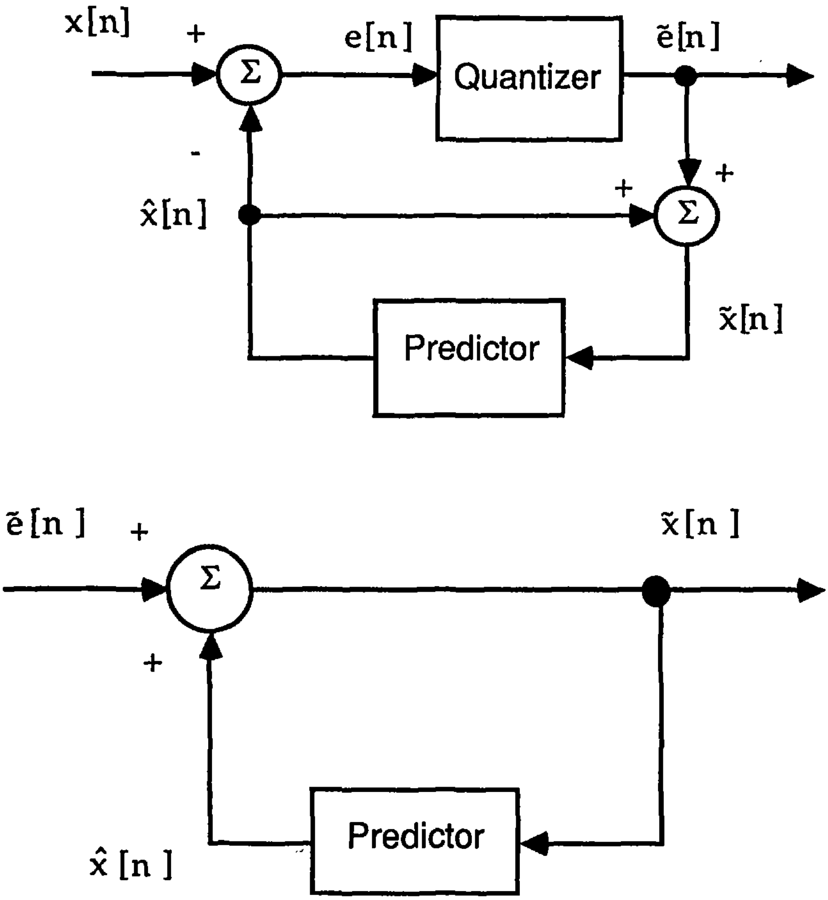


Figure 1: Conventional Differential Pulse-Code Modulation (DPCM) encoder (top) and decoder (bottom). DPCM uses a linear predictor to estimate the next input sample based on knowledge of the previous quantized output values (backward prediction).

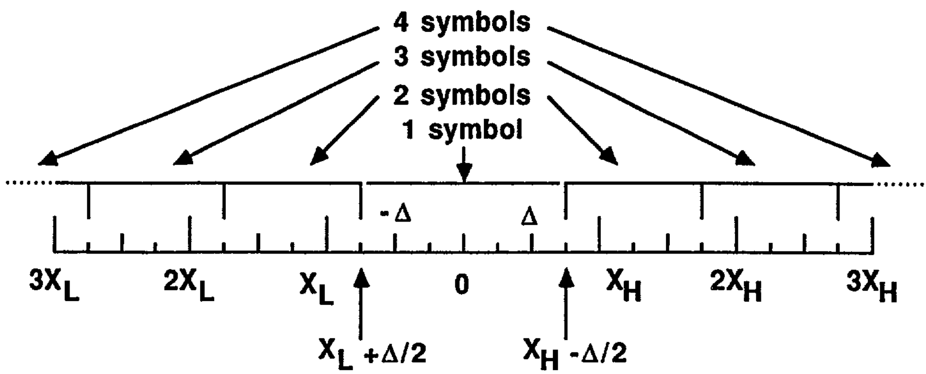


Figure 2: Recursively-Indexed Quantizer (RIQ) produces a set of output symbols per input sample. For centrally-distributed quantizer inputs, the effect on output rate is minimal, yet quantizer overload is eliminated.

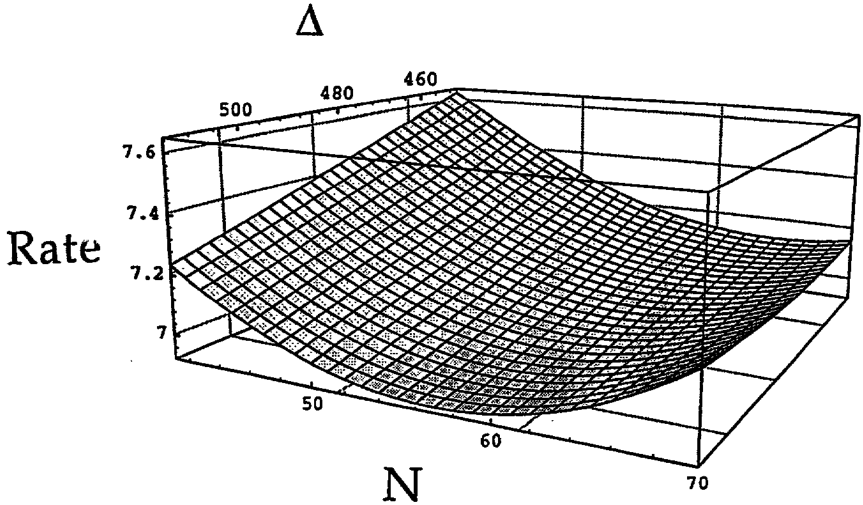


Figure 3: Analytical relationship among output bit rate, number of quantizer levels, and quantizer step size for quantizer input with laplacian distribution (see text).

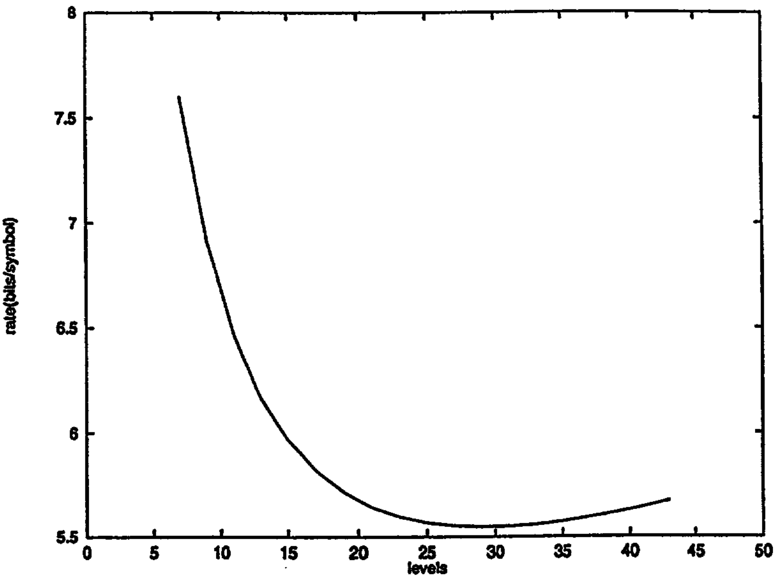


Figure 4: Empirical results based on the RIQ-DPCM performance for an actual music signal. Note minimum value of rate achieved for *2N+1* near 29.

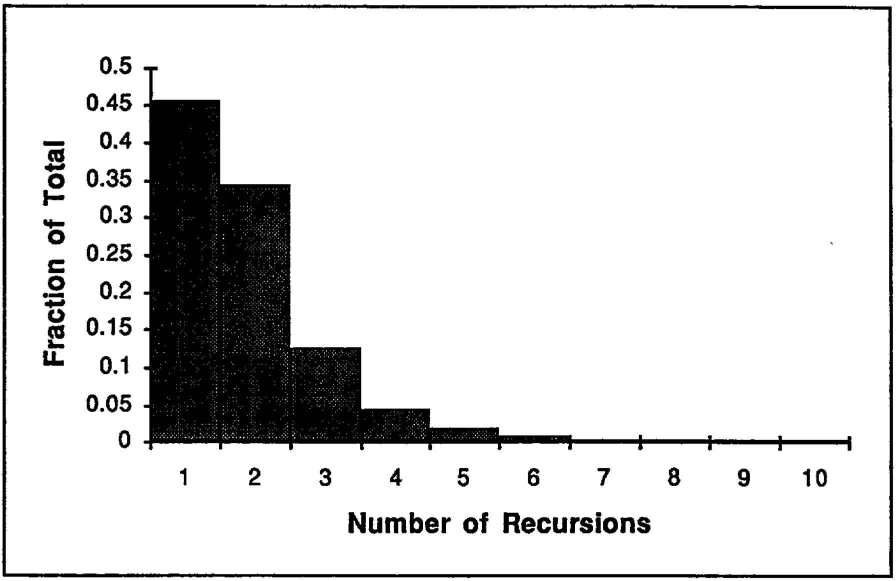


Figure 5: Histogram showing typical number of recursions for RIQ-DPCM system with music input.

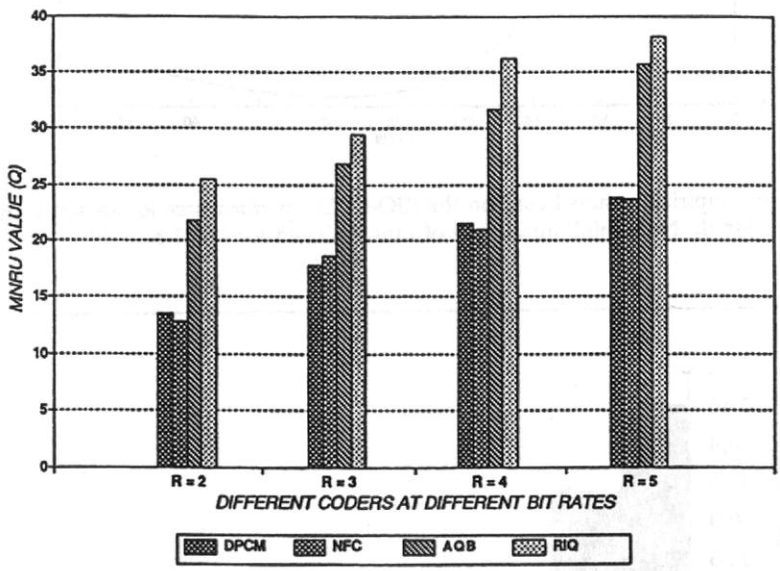


**Figure 6: Subjective test results for preference among DPCM, NFC, ADPCM, and RIQ-DPCM for a variety of musical source signals.**