$(l+1)$ symbols, the average number of symbols per run will be

$$\mu = \sum_{l=0}^{M-1} (l+1)p^l(1-p) + Mp^M = \frac{(1-p^M)}{(1-p)} \tag{11.7}$$

Thus it takes m bits to establish a run-length code for a sequence of μ binary symbols, on the average. The compression achieved is, therefore,

$$C = \frac{\mu}{m} = \frac{(1-p^M)}{m(1-p)} \tag{11.8}$$

For $p = 0.9$ and $M = 15$ we obtain $m = 4$, $\mu = 7.94$, and $C = 1.985$. The achieved average rate is $B_a = m/\mu = 0.516$ bit per pixel and the code efficiency, defined as H/B_a, is $0.469/0.516 = 91\%$. For a given value of p, the optimum value of M can be determined to give the highest efficiency. RLC efficiency can be improved further by going to a variable length coding method such as Huffman coding for the blocks of length m. Another alternative is to use *arithmetic coding* [10] instead of the RLC.

Bit-Plane Encoding [11]

A 256 gray-level image can be considered as a set of eight *1-bit planes,* each of which can be run-length encoded. For 8-bit monochrome images, compression ratios of 1.5 to 2 can be achieved. This method becomes very sensitive to channel errors unless the significant bit planes are carefully protected.

11.3 PREDICTIVE TECHNIQUES

Basic Principle

The philosophy underlying predictive techniques is to remove mutual redundancy between successive pixels and encode only the new information. Consider a sampled sequence $u(m)$, which has been coded up to $m = n-1$ and let $u^{\cdot}(n-1)$, $u^{\cdot}(n-2), \ldots$ be the values of the reproduced (decoded) sequence. At $m = n$, when $u(n)$ arrives, a quantity $\bar{u}^{\cdot}(n)$, an estimate of $u(n)$, is predicted from the previously decoded samples $u^{\cdot}(n-1), u^{\cdot}(n-2)\ldots$, that is,

$$\bar{u}^{\cdot}(n) = \psi(u^{\cdot}(n-1), u^{\cdot}(n-2), \ldots) \tag{11.9}$$

where $\psi(\cdot,\cdot,\ldots)$ denotes the prediction rule. Now it is sufficient to code the *prediction error*

$$e(n) \triangleq u(n) - \bar{u}^{\cdot}(n) \tag{11.10}$$

If $e^{\cdot}(n)$ is the quantized value of $e(n)$, then the reproduced value of $u(n)$ is taken as

$$u^{\cdot}(n) = \bar{u}^{\cdot}(n) + e^{\cdot}(n) \tag{11.11}$$

The coding process continues *recursively* in this manner. This method is called *differential pulse code modulation* (DPCM) or *differential* PCM. Figure 11.5 shows

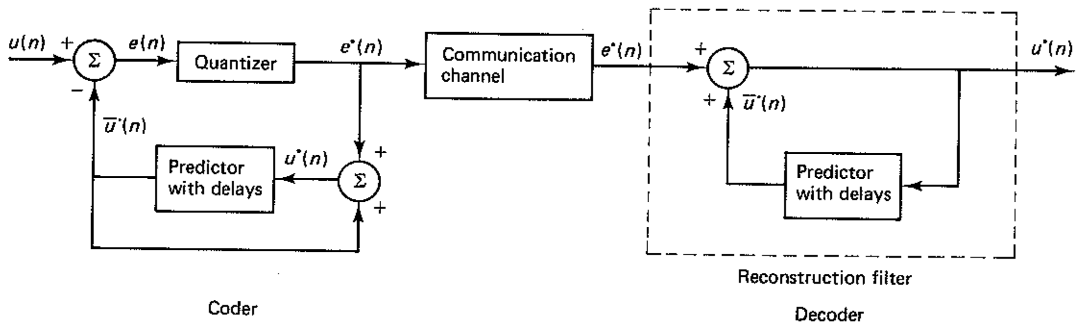

Figure 11.5 Differential pulse code modulation (DPCM) CODEC.

the DPCM *codec* (coder-decoder). Note that the coder has to calculate the reproduced sequence $u^\cdot(n)$. The decoder is simply the predictor loop of the coder. Rewriting (11.10) as

$$u(n) = \bar{u}^\cdot(n) + e(n) \tag{11.12}$$

and subtracting (11.11) from (11.12), we obtain

$$\delta u(n) \overset{\Delta}{=} u(n) - u^\cdot(n) = e(n) - e^\cdot(n) = q(n) \tag{11.13}$$

Thus, the pointwise coding error in the input sequence is exactly equal to $q(n)$, the quantization error in $e(n)$. With a reasonable predictor the mean square value of the differential signal $e(n)$ is much smaller than that of $u(n)$. This means, for the same mean square quantization error, $e(n)$ requires fewer quantization bits than $u(n)$.

Feedback Versus Feedforward Prediction

An important aspect of DPCM is (11.9) which says the prediction is based on the output—the quantized samples—rather than the input—the unquantized samples. This results in the predictor being in the feedback loop around the quantizer, so that the quantizer error at a given step is fed back to the quantizer input at the next step. This has a stabilizing effect that prevents dc drift and accumulation of error in the reconstructed signal $u^\cdot(n)$.

On the other hand, if the prediction rule is based on the *past inputs* (Fig. 11.6a), the signal reconstruction error would depend on all the past and present

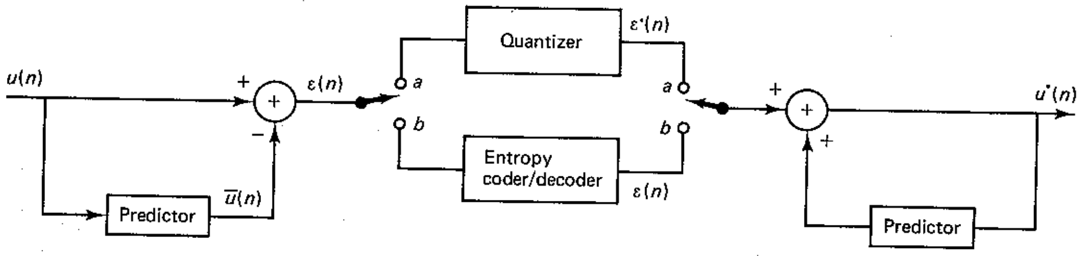

Figure 11.6 Feedforward coding (a) with distortion; (b) distortionless.

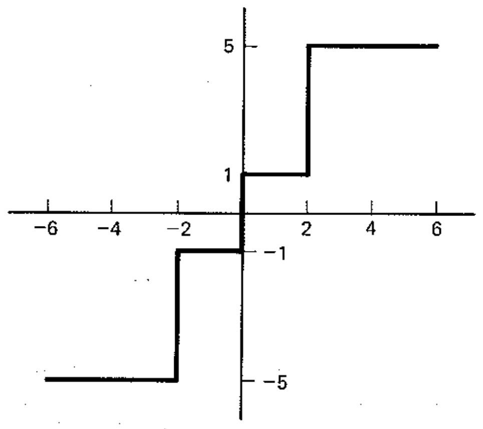

Figure 11.7 Quantizer used in Example 11.2.

quantization errors in the feedforward prediction-error sequence $\varepsilon(n)$. Generally, the mean square value of this reconstruction error will be greater than that in DPCM, as illustrated by the following example (also see Problem 11.3).

Example 11.2

The sequence 100, 102, 120, 120, 120, 118, 116 is to be predictively coded using the previous element prediction rule, $\bar{u}^\cdot(n) = u^\cdot(n-1)$ for DPCM and $\bar{u}(n) = u(n-1)$ for the feedforward predictive coder. Assume a 2-bit quantizer shown in Fig. 11.7 is used, except the first sample is quantized separately by a 7-bit uniform quantizer, giving $u^\cdot(0) = u(0) = 100$. The following table shows how reconstruction error builds up with a feedforward predictive coder, whereas it tends to stabilize with the feedback system of DPCM.

	Input		DPCM					Feedforward Predictive Coder				
	n	$u(n)$	$\bar{u}^\cdot(n)$	$e(n)$	$e^\cdot(n)$	$u^\cdot(n)$	$\delta u(n)$	$\bar{u}(n)$	$\varepsilon(n)$	$\varepsilon^\cdot(n)$	$u^\cdot(n)$	$\delta u(n)$
	0	100	—	—	—	100	0	—	—	—	100	0
	1	102	100	2	1	101	1	100	2	1	101	1
Edge →	2	120	101	19	5	106	14	102	18	5	106	14
	3	120	106	14	5	111	9	120	0	−1	105	15
	4	120	111	9	5	116	4	120	0	−1	104	16
	5	118	116	2	1	117	1	120	−2	−5	99	19

Distortionless Predictive Coding

In digital processing the input sequence $u(n)$ is generally digitized at the source itself by a sufficient number of bits (typically 8 for images). Then, $u(n)$ may be considered as an integer sequence. By requiring the predictor outputs to be integer values, the prediction error sequence will also take integer values and can be entropy coded without distortion. This gives a distortionless predictive codec (Fig. 11.6b), whose minimum achievable rate would be equal to the entropy of the prediction-error sequence $\varepsilon(n)$.