Differential Pulse Code Modulation (DPCM)

According to the Nyquist sampling criterion, a signal must be sampled at a sampling rate that is at least twice the highest frequency in the signal to be able to reconstruct it without aliasing. The samples of a signal that is sampled at that rate or close to generally have little correlation between each other (knowing a sample does not give much information about the next sample). However, when a signal is highly oversampled (sampled at several times the Nyquist rate, the signal does not change a lot between from one sample to another. Consider, for example, a sine function that is sampled at the Nyquist rate. Consecutive samples of this signal may alternate over the whole range of amplitudes from –1 and 1. However, when this signal is sampled at a rate that is 100 times the Nyquist rate (sampling period is 1/100 of the sampling period in the previous case), consecutive samples will change a little from each other. This fact can be used to improve the performance of quantizers significantly by quantizing a signal that is the difference between consecutive samples instead of quantizing the original signal. This will result in either requiring a quantizer with much less number of bits (less information to transmit) or a quantizer with the same number of bits but much smaller quantization intervals (less quantization noise and much higher SNR).

Consider a signal $x(t)$ that is sampled to obtain the samples $x(kT_s)$, where T_s is the sampling period and k is an integer representing the sample number. For simplicity, the samples can be written in the form $x[k]$, where the sample period T_s is implied. Assume that the signal $x(t)$ is sampled at a very high sampling rate. We can define $d[k]$ to be the difference between the present sample of a signal and the previous sample, or

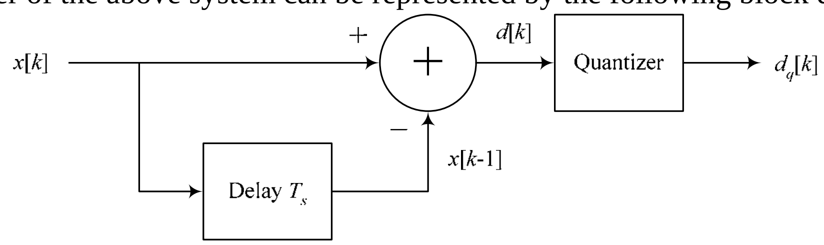
$$d[k] = x[k] - x[k-1].$$

Now this signal $d[k]$ can be quantized instead of $x[k]$ to give the quantized signal $d_q[k]$. As mentioned above, for signals $x(t)$ that are sampled at a rate much higher than the Nyquist rate, the range of values of $d[k]$ will be less than the range of values of $x[k]$.

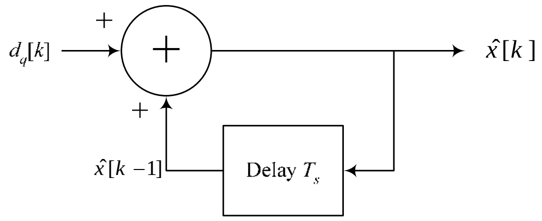
After the transmission of the quatized signal $d_q[k]$, theoretically we can reconstruct the original signal by doing an operation that is the inverse of the above operation. So, we can obtain an approximation of $x[k]$ using

$$\hat{x}[k] = d_q[k] + \hat{x}[k-1].$$

So, if $d_q[k]$ is close to $d[k]$, it appears from the above equation that obtained $\hat{x}[k]$ will be close to $d[k]$. However, this is generally not the case as will be shown later. The transmitter of the above system can be represented by the following block diagram.



The receiver that will attempt to reconstruct the original signal after transmitting it through the channel can be represented by the following block diagram.



Because we are quantizing a difference signal and transmitting that difference over the channel, the reconstructed signal may suffer from one or two possible problems.

1. Accumulation of quantization noise: the above system suffer from the possible accumulation of the quantization noise. Unlike the quantization of a signal where quantization error in each sample of that signal is completely independent from the quantization error in other samples, the quantization error in this system may accumulate to the point that it will result in a reconstructed signal that is very different from the original signal. This is illustrated using the following table. Consider the samples of the input signal $x[k]$ given in the table. The reconstructed signal is given by $\hat{x}[k]$ shown in table. Assume the quantizer used to quantize $d[k]$ is an 8–level quantizer with quantization intervals [–4,–3), [–3,–2), [–2,–1), … , [3,4) and the output quantization levels are the center points in each interval (–3.5, –2.5, –1.5, … , 3.5).

k	–1	0	1	2	3	4	5	6	7	8	9
$x[k]$	0	0.3	1.5	0.7	1.0	2.3	3.7	2.8	3.5	2.8	0
$x[k–1]$	0	0	0.3	1.5	0.7	1.0	2.3	3.7	2.8	3.5	3.1
$d[k]$	0	0.3	1.2	–0.8	0.3	1.3	1.4	–0.9	0.7	–0.7	–2.8
Quantization Up/Down	U (or D)	U	U	U	U	U	U	U	D	U	U
$d_q[k]$	0.5	0.5	1.5	–0.5	0.5	1.5	1.5	–0.5	0.5	–0.5	–2.5
$\hat{x}[k-1]$	0	0.5	1.0	2.5	2.0	2.5	4.0	5.5	5.0	5.5	5.0
$\hat{x}[k]$	0.5	1.0	2.5	2.0	2.5	4.0	5.5	5.0	5.5	5.0	2.5
$\hat{x}[k]-x[k]$	0.5	0.7	1.0	1.3	1.5	1.7	1.8	2.2	2.0	2.2	2.5
Err. Direction Up/Down	U	U	U	U	U	U	U	U	D	U	U

So, it is clear from this table that if the quantization error for a series of samples is going in one direction, the error adds up to produce a output signal that deviates from the original signal. Note that the error between the original and reconstructed samples always increased except when the quantization error switched direction at k = 7 (the shaded box).

2. Effect of transmission errors: in a regular PCM system, the effect of an error that happens in the transmitted signal is only limited to the sample in which the error occurs. In DPCM, an error that occurs in the transmitted signal will cause all the reconstructed samples at the receiver after that error occurs to have errors. Therefore, even if quantization error did not accumulate, an error caused by the channel will cause all successive samples to be wrong. Try this as an exercise by constructing g a table similar to the one above. Intentionally introduce an error in the reconstructed signal at a point and see what happens to the remainder of the reconstructed signal.

Differential Pulse Code Modulation (DPCM) (Continued)

The advantage of DPCM is the reduced amount of information that must be transmitted if we maintain the same SNR or an improved SNR if we maintain the same amount of information. To get an idea on the improvement in performance that we can get from using DPCM as compared to the performance of regular PCM, DPCM can increase the SNR for some signals by as much as 20 dB. This corresponds to an improvement in the signal power compared to the noise power by 100 times, or a reduction in the amount of information by more than 3 bits/sample. However, the system considered in the previous lecture for DPCM is not practical because of the two problems mentioned at the end of the last lecture.

These problems can be solved as follows:

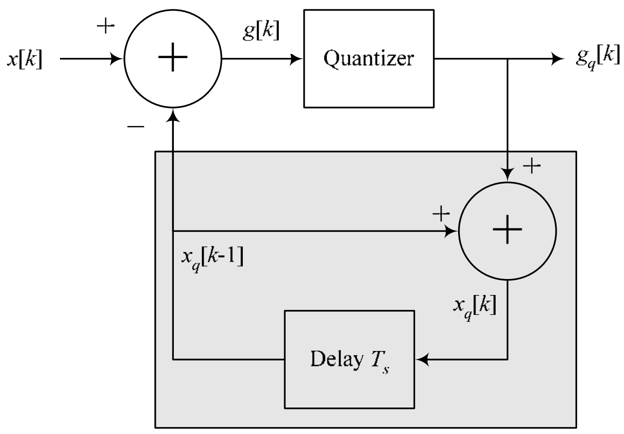
1. **Eliminating the problem of accumulation of quantization noise:** This problem can be solved by avoiding the quantization of the difference signal $d[k]$ between $x[k]$ and its previous sample $x[k–1]$, or

$$d[k] = x[k] - x[k-1],$$

and quantizing instead a difference signal (we will call it $g[k]$) that is the difference between $x[k]$ and the previous sample of its quantized form $x_q[k–1]$. Therefore, $g[k]$ is given by

$$g[k] = x[k] - x_q[k-1].$$

Apparently, this will require applying the quantizer on the signal $x[k]$ to obtain $x_q[k–1]$, which we are trying to avoid since the amplitude of $x[k]$ is generally larger than the amplitude of a difference signal like $d[k]$ or even $g[k]$. In fact, if both $x[k]$ and $g[k]$ are available, we can reconstruct the quantized form of $x[k]$ using the following system.



In the above system, we can easily prove that the resulting signal $x_q[k]$ is the quantized form of $x[k]$.

First we see that

$$g[k] = x[k] - x_q[k-1].$$

Now, the output of the quantizer is the quantized form of $g[k]$ which can be represented by adding a quantization noise $q[k]$ to the input of the quantizer. Therefore,

$$g_q[k] = g[k] + q[k].$$

Substituting for $g[k]$ in $g_q[k]$ gives

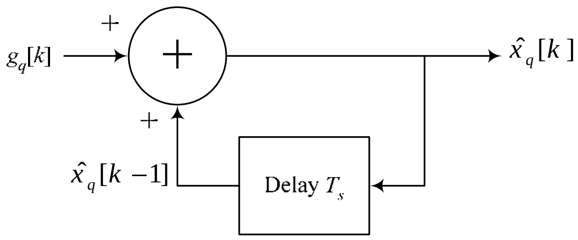
$$g_q[k] = x[k] - x_q[k-1] + q[k]$$

From the block diagram,

$$\begin{aligned} x_q[k] &= g_q[k] + x_q[k-1] \\ &= x[k] - x_q[k-1] + q[k] + x_q[k-1] \\ &= x[k] + q[k] \end{aligned}$$

So, in fact, the function $x_q[k]$ is the quantized form of $x[k]$ as seen by the last line of the above equation. A word of caution here, the above derivation does not mean that if we quantized $x[k]$ directly by the quantizer we will get $x_q[k]$. It just says that $x_q[k]$ is a quantized form of $x[k]$. If we passed $x[k]$ through the same quantizer in the block diagram above, we will get another function $x_{q2}[k]$ with samples that are generally different from $x_q[k]$.

At the receiver side of the DPCM system, we can use the gray block in the transmitter labeled "Predictor" since its input is the DPCM output $g_q[k]$ and its output is the desired signal $x_q[k]$. Therefore the block diagram would be as follows.



Now, assume the quantizer used to quantize $g[k]$ is again an 8–level quantizer with quantization intervals [–4,–3), [–3,–2), [–2,–1), … , [3,4) and the output quantization levels are the center points in each interval (–3.5, –2.5, –1.5, … , 3.5).

NOTE: THE FOLLOWING TABLE IS FOR ILLUSTRATION. DO NOT SPEND TOO MUCH TIME TRYING TO FIGURE OUT HOW IT IS COMPUTED (MATLAB HELPED ME).

k	–1	0	1	2	3	4	5	6	7	8	9
$x[k]$	0	0.3	1.5	0.7	1.0	2.3	3.7	2.8	3.4	2.8	0
$g[k]$ $(= x[k] - x_q[k-1])$	0	–0.2	1.5	–0.8	0	0.8	1.7	–0.7	0.4	–0.7	–3.0
$g_q[k]$	0.5	–0.5	1.5	–0.5	0.5	0.5	1.5	–0.5	0.5	–0.5	–2.5
Quantization Up/Down	U (or D)	D	–	U	U (or D)	D	D	U	U	U	U
$x_q[k]$ $(= g_q[k] + x_q[k-1])$	0.5	0	1.5	1.0	1.5	2.0	3.5	3.0	3.5	3.0	0.5
$x_q[k-1]$	0	0.5	0	1.5	1.0	1.5	2.0	3.5	3.0	3.5	3.0
$g_q[k]$	0.5	–0.5	1.5	–0.5	0.5	0.5	1.5	–0.5	0.5	–0.5	–2.5
$\hat{x}[k-1]$	0	0.5	0	1.5	1.0	1.5	2.0	3.5	3.0	3.5	3.0
$\hat{x}[k]$	0.5	0	1.5	1.0	1.5	2.0	3.5	3.0	3.5	3.0	0.5
$\hat{x}[k]-x[k]$	0.5	–0.3	0	0.3	0.5	–0.3	–0.2	0.2	0.1	0.2	0.5
Err. Direction Up/Down	U	D	–	U	U	D	D	U	–	U	U

This table illustrates that the above DPCM does not cause accumulation of error. Looking at the reconstructed signal and the original input signal, we see that the magnitude of the difference is always less than or equal to half the quantization interval (i.e. ≤ 0.5). Even when the quantization error for a sequence of samples had the same direction as it is the case for the last four columns of the table, the difference between the input and output of the system was always within half the quantization interval.

2. Reducing the effect of transmission errors: as mentioned before, transmission errors result in errors in all the reconstructed samples of the input signal that come after the transmission error. The best method to combat this problem is to divide the data into sets of samples and resent the transmitter and receiver after the transmission of each set of samples. This way, a transmission error that occurs will affect only the samples of that part of the data. Once the system is reset, the effect of that error will stop.