

Comunicaciones Digitales

## Práctica ADPCM

GALLEGOS RUIZ DIANA ABIGAIL

GRUPO: 2TV10

2022



# 1. Introducción

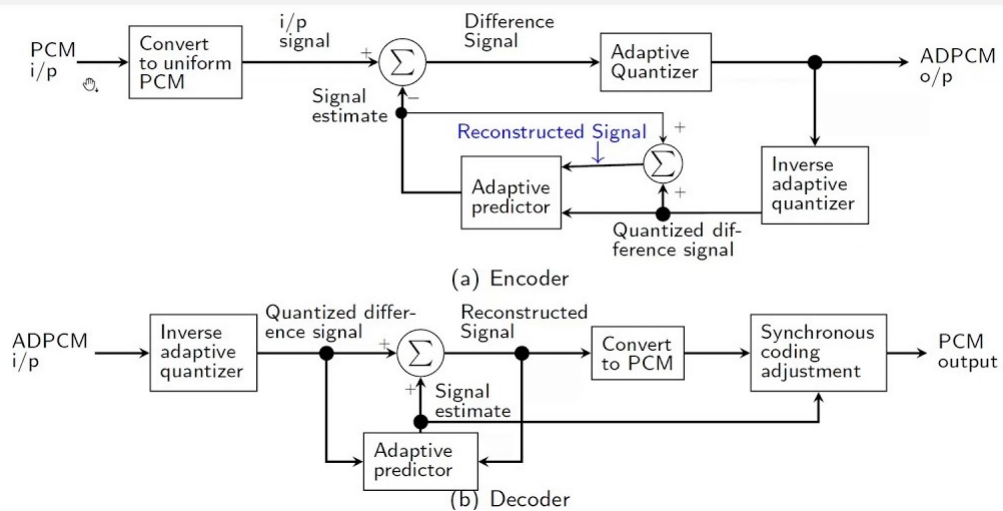
La modulación por código de impulsos diferencial adaptable (ADPCM) es un método utilizado para convertir señales analógicas en señales binarias. La técnica convierte las señales analógicas tomando muestras frecuentes del sonido y representando el valor de la modulación muestreada en forma binaria.

## Aplicaciones

La técnica ADPCM se emplea para enviar señales de sonido a través de líneas de fibra óptica de larga distancia. Es útil sobre todo para las organizaciones que establecen líneas digitales entre sitios remotos para transmitir tanto voz como datos. Las señales de voz se digitalizan antes de ser emitidas.

En el campo de las telecomunicaciones, la técnica ADPCM se utiliza principalmente en la compresión de voz porque el método permite reducir el flujo de bits sin comprometer la calidad. El método ADPCM puede aplicarse a todas las formas de onda, audio de alta calidad, imágenes y otros datos modernos.

## ADAPTIVE DIFFERENTIAL PULSE CODE MODULATION



## 2. Desarrollo

Código

```
1  clc
2  clear all
3
4  filename = 'OSR_us_000_0016_8k.wav';
5  [y,Fs] = audioread(filename);
6  ts=1/Fs;
7  fs = y(:,1);
8  t = 0:ts:(length(y)*ts)-ts;
9
10
11  fs1 =circshift(fs,1);
12  fs1(1)=0;
13  for ro=1:8
14
15  qu=0.0023;    %intervalo  reconstruido
16
17
18  %----- DIFERENCIA -----
19  i=length(fs)-1;
20  d=fs-fs1;
21  % ----- CUANTIZACION -----
22
23  for k=1:length(fs)
24  q(k)=d(k)/qu;
25  if q(k) <0
26  q(k)=floor(q(k));
27  else
28  q(k)=ceil(q(k));
29  end
30  end
31
32  % ----- DPCM -----
33  % Con retroalimentacion
34
35  for k=1:length(fs)
36  if abs(q(k)) <= 2^(ro-1)
37  DPCM(k)= q(k);
38  elseif abs(q(k)) > 2^(ro-1)
39  DPCM(k)= 2^(ro-1);
40  else
41  DPCM(k)= 0;
42  end
43
44
45  %----- DIFERENCIA RECONSTRUIDA -----
46
47  if DPCM(k)<0
48  dc(k)= (DPCM(k) * qu)+ (qu/2);
49  else
50  dc(k)= (DPCM(k)*qu)- (qu/2);
51  end
```

```

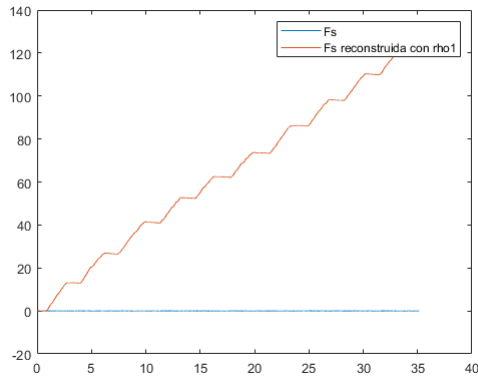
52
53 end
54 %----- MUESTRA RECONSTRUIDA -----
55
56 fsc= zeros(1,length(fs));
57 for k=1:i
58 fsc(k+1) =dc(k)+fsc(k);
59 end
60
61 figure (ro)
62 plot (t,fs)
63 hold on
64 plot(t,fsc)
65 legend('Fs',strcat('Fs reconstruida con rho', num2str(ro)))
66
67 end
68
69 %[fs1' fs' d' q' DPCM' dc' fsc']

```

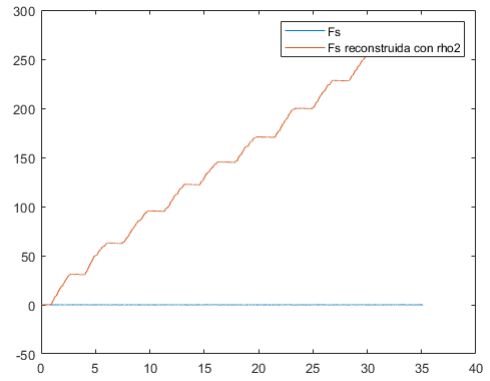
### 3. Resultados

Utilizando el intervalo de cuantificación obtenido en clase ( $q = 0,0023$ ), se obtuvieron los siguientes resultados para la longitud de palabra  $\rho = 1, 2, \dots, 8$ :

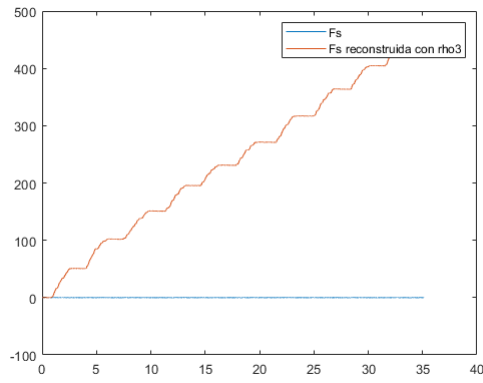
Aunque se había calculado una longitud adecuada de 8 bits, la mejor de las aproximaciones en este sistema se logra con 7 bits.



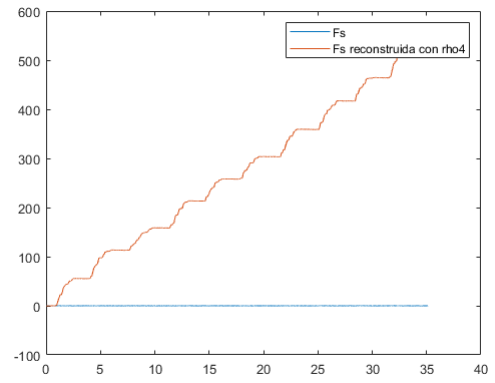
(a)  $\rho = 1$



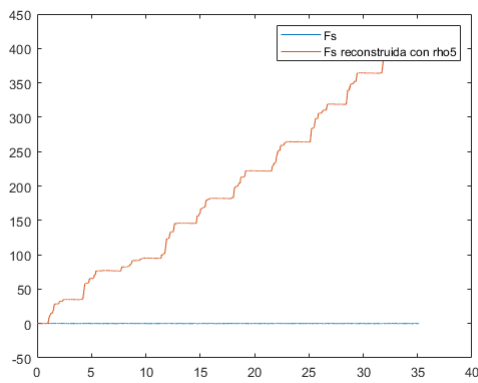
(b)  $\rho = 2$



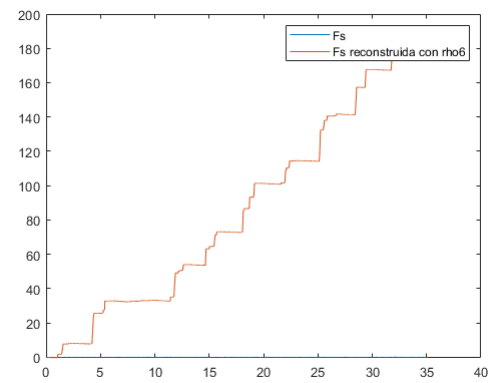
(c)  $\rho = 3$



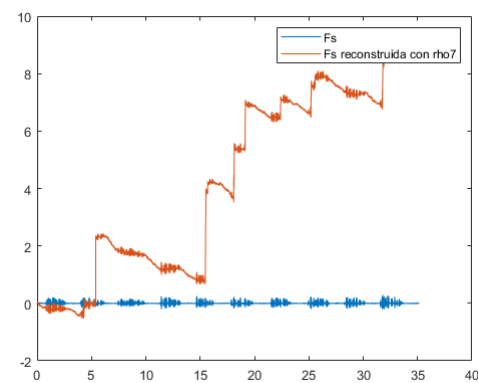
(d)  $\rho = 4$



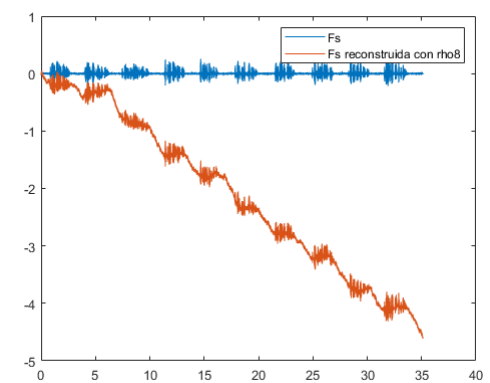
(e)  $\rho = 5$



(f)  $\rho = 6$



(g)  $\rho = 7$



(h)  $\rho = 8$

## 4. Conclusiones

La modulación PCM es una modulación sencilla que consiste en el proceso de cuantificación, sin embargo al quedar modular una señal con muchas muestras, en este caso una señal de voz, PCM comienza a ser ineficiente, por lo que se tuvo que cambiar el código en Matlab implementando DPCM. DPCM hace más sencillo la modulación al transmitir únicamente la diferencia entre las muestras, lo que permite un IDS menor y a su vez un error de cuantización menor. ADPCM implementa la misma técnica que DPCM pero utilizando un algoritmo de predicción más eficiente pero a su vez, un poco más complicado de implementar en Matlab.