Laboratorio de Procesamiento Digital de Voz

Practica 4

CUANTIZACION ESCALAR, LOGARITMICA, (A)DM y (A)DPCM

Objetivos:

Manejar los conceptos de cuantización escalar, logarítmica y manejo de cuantizadores A(DM) y (A)DPCM.

1. Introducción.

La conversión de señales analógicas de voz en señales digitales, es llamada normalmente *codificación de voz* o simplemente *codificación*. Un mayor objetivo en la codificación es la *compresión* de la señal, esto es, emplear un número de bits lo mas pequeño posible en la representación digital de la señal de voz. La eficiente representación digital de la señal de voz hace posible conseguir la eficiencia del ancho de banda en la transmisión de la señal sobre una gran variedad de canales de comunicación, o el almacenamiento eficiente sobre una variedad de medios.

En el transcurso las últimas cuatro o cinco décadas, una gran variedad de técnicas de codificación de voz han sido propuestas, analizadas y desarrolladas. Las técnicas más básicas se pueden subdividir en dos diferentes categorías: *codificadores de la forma de onda* (*waveform coders*) y *codificadores de voz* (*voice coders* <*vocoders*>). En la codificación de forma de onda se intenta codificar directamente la señal de voz en una forma eficiente para explotar las características temporales y/o espectrales. En contraste, la codificación de voz (*vocoders*) involucra la representación de la señal de voz por un conjunto de parámetros, la estimación de los parámetros de tramas de voz y la eficiente codificación de esos parámetros en forma digital para un posible almacenamiento o transmisión.

El proceso de comunicación (o almacenamiento) por conveniencia se divide en tres etapas: muestreo, cuantización y codificación. La figura 1, muestra los procesos independientes en un sistema de codificación y decodificación.

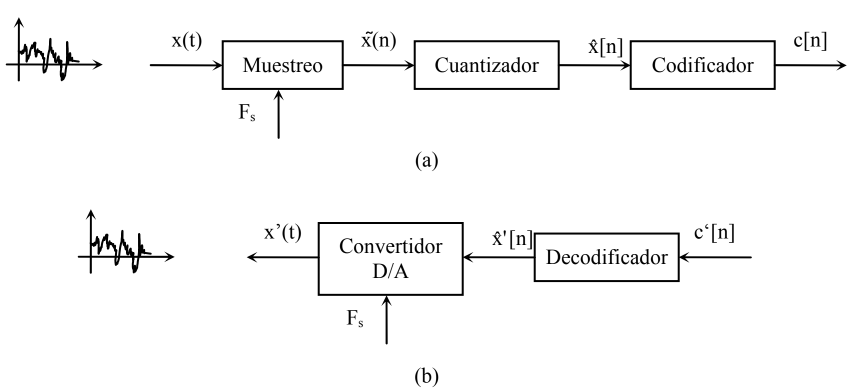


Figura 1. Sistema de comunicación. (a) Etapa de codificación, (b) Etapa de decodificación

En el proceso de codificación, la señal analógica es muestreada a intervalos de tiempo *1/F*$_s$ segundos, dando una secuencia, $\{\tilde{x}(n)\}$, que es conocida como precisión infinita. Para transmitir esta secuencia de muestras sobre un canal de comunicación digital (o almacenamiento digital), los valores muestreados deben ser cuantizados a un conjunto finito de amplitudes, de tal forma que puedan ser representadas por un conjunto finito de símbolos $\{\hat{x}[n]\}$ = {Q $[\tilde{x}(n)]$}. La última etapa que representa cada muestra cuantizada a una palabra código, c[n]. De igual forma el decodificador toma una secuencia de palabras código {c'[n]} y transforma esta a una secuencia de muestras cuantizadas $\{\hat{x}'[n]\}$. Si las palabras código c'[n] son iguales a las palabras código c[n] no se introdujeron errores, entonces $\hat{x}'[n] = \hat{x}[n]$.

2. Cuantización Instantánea.

En la cuantización, la amplitud de las muestras se cuantizan, dividiendo todo el rango de la amplitud, en un conjunto finito de rangos, y asignando el mismo valor de amplitud a todas las muestras que caen dentro de ese rango. Dos de los cuantizadores escalares más comunes son: cuantizador uniforme y el cuantizador logarítmico.

2.1. Cuantización Uniforme

Para la cuantización uniforme, los rangos de cuantización y los niveles son generalmente distribuidos uniformemente. La figura 2 muestra un cuantizador uniforme de tres bits.

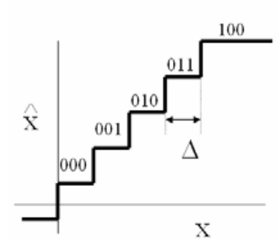


Figure 2. Características de entrada y salida de un cuantizador de 3 bits.

Funciones en Matlab

Para obtener los niveles de cuantización se puede usar la función *quantiz*, de la siguiente forma:

```
indice = quantiz(señal,partición);
[indice,quant] = quantiz(señal,partición,palabras_código);
```

señal: Es la señal de entrada a cuantizar.
Partición: Es un vector que representa las particiones o rangos del cuantizador
Palabras_código: A cada partición es asignada una palabra código, con este vector se determinan esas palabras código.

Índice: Si un valor (muestra) de la señal cae dentro de una región o partición, el índice determina esta región.
Quant: Palabra código correspondiente a la región o partición obtenida.

Ejemplo: la siguiente instrucción muestra la forma de obtener las palabras código, utilizando un cuantizador de tres bits (vea figura 2).

```
[indice,quant] = quantiz([ 1 0 5 3],1:7,[0 1.5 2.5 3.5 4.5 5.5 6.5 8])
```

```
indice =

     0
     0
     4
     2
quant =   0    0   4.5   2.5
```

2.2. Cuantización logarítmica.

La cuantización logarítmica establece los niveles de cuantización de forma logarítmica y asigna una palabra código a cada región. Los cuantizadores logarítmicos más comunes son llamados *ley* μ y *ley A*. La figura 3 muestra el diagrama de bloques de un sistema de *compresor/expansor* para cuantización.

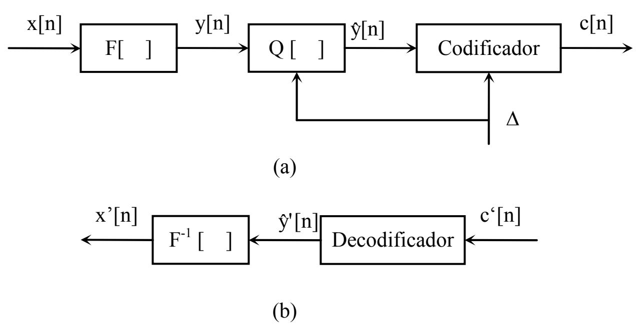


Figura 3. Diagrama de bloques de un sistema *compresor/expansor*

La función *F[ ]* se define con respecto a la ley de compresión. Para la *ley* μ se define *F[ ]* como:

$$y[n] = F[x[n]] = X_{\max} \frac{\log\left[1+\mu\frac{|x[n]|}{X_{\max}}\right]}{\log[1+\mu]} sign(x[n])$$

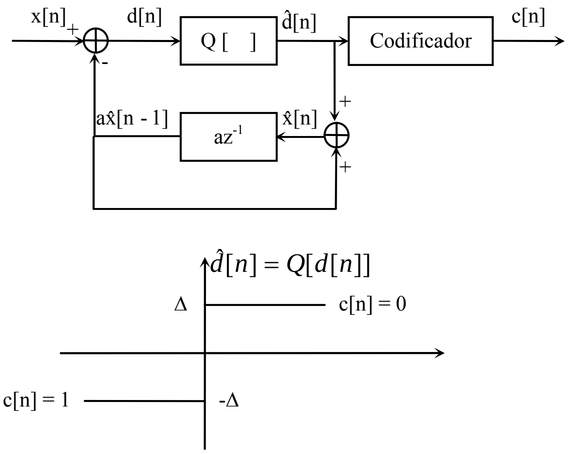
3. Cuantización Diferencial.

Se ha comprobado que la señal de voz no cambia rápidamente de una muestra a otra así que la diferencia entre muestras adyacentes debe tener menor varianza que la señal misma Este hecho provee la motivación la creación de un cuantizador diferencial. Existen varios tipos de cuantizadores diferenciales en los que se encuentran: Modulación Delta (DM), Modulación Delta Adaptiva (ADM), Modulación por Código de Pulso Diferencial (DPCM), DPCM con Cuantización Adaptiva, etc.

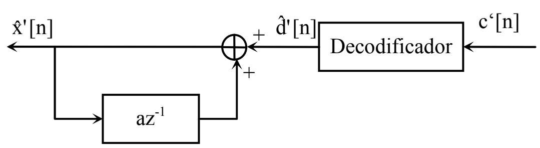
3.1. Modulación delta (DM).

El sistema de modulación delta más simple es ilustrado en la figura 3. En este caso el cuantizador tiene solo dos niveles y el tamaño del paso es fijo. El nivel de cuantización positivo es representado por *c[n]* = 0 y el negativo por *c[n]* = 1, así, $\hat{d}[n]$ es:

$$\hat{d}[n] = \begin{cases} \Delta & si \quad c[n] = 0 \\ -\Delta & si \quad c[n] = 1 \end{cases}$$



(a)

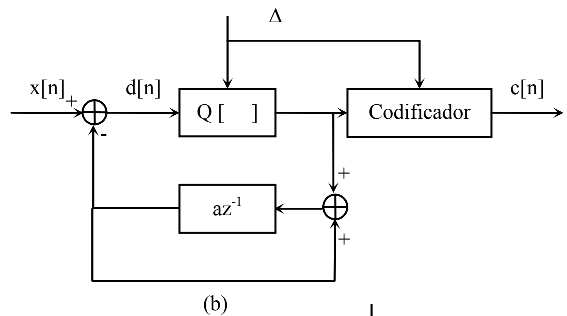


(b)

Figura 3. Diagrama de bloques de un sistema de modulación delta, con predictor fijo de primer orden.

3.2. Modulación por Código de Pulso Diferencial (DPCM).

Los sistemas de cuantización diferencial en los cuales el cuantizador tiene más de dos niveles se conocen como DPCM. La Modulación Delta (DM) puede ser llamado como un Sistema DPCM de 1-bit. La figura 4 muestra un diagrama de bloques de un sistema DMPC.



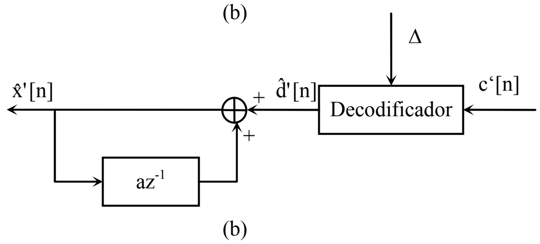


Figura 4. Diagrama de bloques de un sistema de modulación DPCM, con predictor fijo de primer orden

4. Desarrollo.

1. Escribir un programa que realice la gráfica de la función

$$y[n] = F[x[n]] = X_{\max} \frac{\log\left[1+\mu \frac{|x[n]|}{X_{\max}}\right]}{\log[1+\mu]} sign(x[n])$$

   Donde *x[n]* varía de -10 hasta 10 con incrementos de 0.1. Para valores de μ = 1, 5, 15, 40, 100, 255. Colocar las funciones en una misma gráfica y observar las características (use el comando de Matlab *hold on*).

2. Elegir un archivo de voz y escribir un programa que realice la función del punto 1 para los valores de μ = 1, 40, 255. Escuchar cada una de las señales de salida. Anotar sus comentarios.
3. Elegir una señal de voz y escribir un programa que realice la cuantización uniforme de 32 y 64 niveles. Escuchar las señales de salida y anotar sus observaciones. Nota: para la cuantización considere el rango dinámico de la señal y utilice la función de Matlab.

4. Proyecto.

1. Escribir un programa que realice la modulación DM. Utilice los parámetros Δ = 0.025, 0.075, 0.15 y 0.25.; a = 0.9 y palabras código c[n] = {0, 1}.
2. Escribir un programa que realice la modulación DPCM. Utilice los parámetros Δ = {±0.025, ±0.075, ±0.15 y ±0.3}; a = 0.9 y con niveles de cuantización: c[n] = {00, 01,10 y 11} (2 bit) y un bit de signo. Es decir,

$$c[n]=\begin{cases} 00 & Para \quad |\hat{d}[n]| \leq 0.05 \\ 01 & Para \quad 0.05 < |\hat{d}[n]| \leq 0.1 \\ 10 & Para \quad 0.1 < |\hat{d}[n]| \leq 0.2 \\ 11 & Para \quad |\hat{d}[n]| > 0.2 \end{cases}$$

OBSERVACIONES:

- Para tener una mejor codificación, es necesario que las señales de voz sean normalizadas, esto es:

  VozNorm = Voz / X_{max} ;

  Donde X_{max} = max(abs(Voz)).

  De allí el porque los incrementos Δ

- Los archivos que se introducirán son los archivos binarios RAW (sin encabezados, solo las muestras de la señal y en formato big-endian).
- El archivo de salida será un archivo binario con todas las palabras código obtenidas en la codificación.

Bibliografía

Practica 1 Practica 2 Practica 3

"Discrete–Time Processing of Speech Signals", John R. Deller Jr, John G. Proakis, John H. L. Hansen, Ed. Prentice Hall 1987

"Apuntes de clases de Procesamiento Digital de Voz", Abel Herrera Camacho