

Laboratorio

Modulación por Código de Pulso: Muestreo, Cuantización y Codificación Códigos de Línea

La mayoría de las fuentes de información son analógicas por naturaleza, e.g., sonido e imágenes. Por lo tanto, es importante considerar la conversión de fuentes analógicas a secuencias digitales que se puedan utilizar en los sistemas de comunicación digital.

Los procesos de muestreo, cuantización y codificación ofrecen la posibilidad de transmitir un mensaje en forma de pulsos modulados, y no con una señal continua. Los pulsos resultantes del muestreo tienen la característica de que su duración es muy pequeña con relación al intervalo entre ellos, lo que implica muchas ventajas en términos de modulación y transmisión de estas señales.

En un sistema de comunicación digital donde la fuente y destino de información sean analógicos son fundamentales dos operaciones: (i) Conversión Analógico-Digital (ADC), que incluye las operaciones de muestreo, cuantización y codificación, y (ii) Conversión Digital-Analógica (DAC), que es la operación de reconstrucción de la señal analógica a partir de la señal digital.

La obtención de señales muestreadas (pulsos modulados en amplitud) es la primera etapa de la modulación por código de pulso (PCM). Pero es necesario ir más allá, hasta la construcción de una palabra código (grupo de bits) para cada muestra, que se transmite, generalmente, en forma serial. Una gran ventaja de PCM sobre PAM es que las señales transmitidas tienen igual amplitud como se muestra en la Fig. 1.

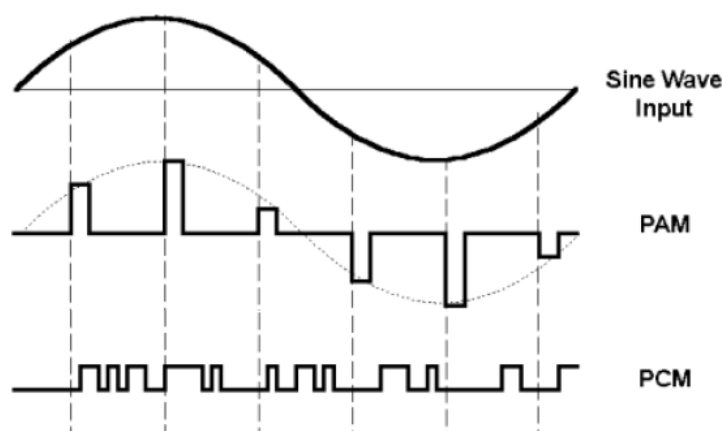


Figura 1: Señal de entrada, muestras (PAM) y señal PCM correspondiente.

Una vez se generan las palabras-código, éstas deben convertirse de niveles lógicos estándar (TTL, CMOS, etc.) a información más apropiada para la transmisión por líneas telefónicas. Esencialmente, existen cuatro factores principales que deben ser considerados cuando se selecciona un formato de codificación de líneas: (i) *recuperación del reloj*, (ii) *ancho de banda de transmisión*, (iii) *facilidad de detección y decodificación*, y (iv) *detección de error*.

En esta experiencia se analizará el teorema de muestreo, tanto en el dominio del tiempo como de la frecuencia, estudiando sus efectos y verificando la operación de reconstrucción de la señal original a partir de

las muestras. Además, se estudiará el proceso de cuantización y el efecto de variar la resolución (número de bits) de las palabras código.

Objetivos

Muestreo, cuantización y codificación

- Ilustrar el teorema de muestreo y utilizar el análisis de Fourier para describir la operación de muestreo.
- Analizar los efectos del muestreo en la representación y reconstrucción de una señal.
- Evaluar la operación de un cuantizador y el efecto de su resolución.
- Implementar un codificador para generar palabras-código.

Códigos de línea

- Observar la operación de los codificadores de línea PCM.
- Observar las características de transmisión de los esquemas de codificación de línea: UPNRZ, BPNRZ, UPRZ y BPRZ.

Materiales y Equipo:

1 computador con software de simulación

Experiencias

Para el desarrollo de las siguientes experiencias debe aplicar los conocimientos adquiridos durante las sesiones de clase del curso y su experiencia previa en MATLAB. No todos los puntos o preguntas en los problemas se pueden solucionar directamente con el software. En algunos casos deberá desarrollar parcial o totalmente algunos puntos en forma analítica antes de poder implementar alguna aplicación en el programa.

Funciones de interés

sinc stairs spline quant de2bi

`y = sinc(x)` returns an array, `y`, whose elements are the sinc of the elements of the input, `x`. `y` is the same size as `x`. ($\text{sinc}(x) \equiv \sin(\pi x)/(\pi x)$)

`stairs(x,y)` draws a staircase graph of the elements in `y` at the locations specified by `x`.

`yy = spline(x,Y,xx)` uses a cubic spline interpolation to find `yy`, the values of the underlying function `Y` at the values of the interpolant `xx`. For the interpolation, the independent variable is assumed to be the final dimension of `Y` with the breakpoints defined by `x`. The values in `x` must be distinct.

```
function [y]=quant(x,n)
a=max(max(x))-min(min(x));
delta= a/(n-1);
z=(x-min(min(x)))/delta;
y=round(z);
end
```

`b = de2bi(d,n)` converts a nonnegative decimal integer `d` to a binary row vector with `n` columns. If `d` is a vector, the output `b` is a matrix in which each row is the binary form of the corresponding element in `d`. It assumes 'right-msb'.

Sección 1. Muestreo y reconstrucción

1.1 Muestreo

- a. Considere una señal de mensaje

$$m(t) = -3 + 2\cos(300\pi t) + 4\cos(500\pi t) + 6\cos^2(400\pi t)$$

en el intervalo $[0, 200 \text{ ms}]$.

- b. Determine la frecuencia fundamental (f_0) y la máxima componente de frecuencia (f_m) de la señal $m(t)$.
- c. Genere un vector m que simule la señal “analógica” $m(t)$ usando una frecuencia de muestreo 20 veces f_m .
- d. Grafique (usando plot) la señal en el intervalo $[0, 60 \text{ ms}]$ y su espectro de amplitud $M(f)$ de doble lado.
- e. Genere un vector m_1 a partir del vector m que corresponda a una señal muestreada a una frecuencia igual a 4 veces f_m . Determine el periodo de muestreo T_{s1} .
- f. Grafique, en el intervalo $[0, 60 \text{ ms}]$, la señal muestreada resultante $m(nT_{s1})$ (usando stem) y sobreponga la señal “analógica” $m(t)$ (usando plot). Además, grafique el espectro de amplitud $M_1(f)$ de doble lado de la señal muestreada.
- g. Genere un vector m_2 a partir del vector m que corresponda a una señal muestreada a una frecuencia igual a 2.5 veces f_m . Determine el periodo de muestreo T_{s2} .
- h. Grafique, en el intervalo $[0, 60 \text{ ms}]$, la señal muestreada resultante $m(nT_{s2})$ (usando stem) y sobreponga la señal “analógica” $m(t)$ (usando plot). Además, grafique el espectro de amplitud $M_2(f)$ de doble lado de la señal muestreada.
- i. Genere un vector m_3 a partir del vector m que corresponda a una señal muestreada a una frecuencia igual a 2 veces f_m . Determine el periodo de muestreo T_{s3} .
- j. Grafique, en el intervalo $[0, 60 \text{ ms}]$, la señal muestreada resultante $m(nT_{s3})$ (usando stem) y sobreponga la señal “analógica” $m(t)$ (usando plot). Además, grafique el espectro de amplitud $M_3(f)$ de doble lado de la señal muestreada.
- k. Genere un vector m_4 a partir del vector m que corresponda a una señal muestreada a una frecuencia igual a 1.25 veces f_m . Determine el periodo de muestreo T_{s4} .
- l. Grafique, en el intervalo $[0, 60 \text{ ms}]$, la señal muestreada resultante $m(nT_{s4})$ (usando stem) y sobreponga la señal “analógica” $m(t)$ (usando plot). Además, grafique el espectro de amplitud $M_4(f)$ de doble lado de la señal muestreada.
- m. Compare las señales generadas en el tiempo y sus espectros. ¿Qué observa?, ¿qué puede concluir?

1.2 Reconstrucción

Considere la señal de mensaje $m(t)$ anterior, y la señal muestreada obtenidas en la parte (g) de la sección 1.1.

- a. Utilizando la señal muestreada $m_2(t)$ reconstruya la señal original utilizando tres métodos: (i) función sinc, (ii) función stairs, (iii) función spline.
- b. Grafique los resultados y compare las señales recuperadas con la señal original $m(t)$. Comente sus resultados.

Sección 2. Cuantización y modulación por código de pulso

- a. Describa la función quant incluida en la sección inicial “Funciones de interés”
- b. Para 4 niveles de cuantización, determine la salida de la función quant cuando las muestras de entrada son $[0.7, 0.2, -0.4, -0.7]$.

- c. Genere una señal sinusoidal $x(t) = 2 \sin(10\pi t)$ con duración de un periodo y frecuencia de muestreo 20 veces la frecuencia de la señal. Calcule cuántas muestras se generan.
- d. Aplique una cuantización uniforme a las muestras de la señal con 4, 8, 16, 32 y 64 niveles, y grafique cada una de las salidas (usando stem). En cada gráfica, sobreponga la curva de la onda $x(t)$ usando plot.
- e. Desarrolle un script para codificar (en binario) la señal cuantizada.
- f. Genere una tabla con los valores de las muestras, de las muestras cuantizadas y el código binario correspondiente, para los distintos niveles de cuantización.
- g. Calcule cuántos bits se requieren transmitir en la secuencia cuando se usan 8 y 32 niveles de cuantización.
- h. Grafique la secuencia de bits para las primeras 10 muestras en los casos correspondientes a 8, 16 y 32 niveles de cuantización. Compare los resultados y comente sobre lo que observa.