

PRACTICA 3: Muestreo y Reconstrucción de señales

El objetivo de esta práctica es trabajar con algunas de las herramientas que proporciona MATLAB sobre el tema del muestreo y la reconstrucción de señales continuas.

1. Genera y representa dos señales senoidales, una de 8 Hz y otra de 28 Hz durante 1 s y con una resolución de una milésima de segundo.
2. Obtén por ejemplo las primeras 20 muestras de las dos señales anteriores muestreadas a una frecuencia de 20 Hz y represéntalas gráficamente.
3. ¿ Qué observas en las gráficas obtenidas anteriormente y a qué crees que se puede deber.?
4. Muestrea las dos señales del apartado 1 con una frecuencia de 60 Hz en vez de 20 Hz y represéntalas gráficamente.
5. Qué observas en las gráficas obtenidas y a qué crees que se debe.
6. Suma las dos señales del apartado 1 y represéntalas gráficamente.
7. Obtén 60 muestras de la señal suma obtenida en el apartado anterior muestreada a 60 Hz.
8. Representa gráficamente la señal suma (continua) y la misma señal muestreada.
9. Si la señal a reconstruir es causal, la reconstrucción ideal se puede realizar utilizando la expresión $x(t) = x_r(t) = \sum_{n=0}^L x[n] \sin c(F_s t - n)$, donde $x[n]$ representa la señal muestreada con periodo de muestreo F_s , $x(t)$ es la señal original, $x_r(t)$ representa a la señal reconstruida y L es el número de muestras tomadas de la señal. Intenta reconstruir la señal suma a partir del vector de muestras y de la expresión dada anteriormente. Representa gráficamente ambas señales y observa los resultados.
10. Un retenedor de orden cero intenta reconstruir la señal a partir de las muestras usando la expresión $x_r(t) = x(nT) \quad nT \leq t \leq (n + 1)T$. Intenta reconstruir la señal usando un retenedor de orden cero (puedes ayudarte de la función floor() de matlab).
11. Un retenedor de orden 1 intenta reconstruir la señal a partir de las muestras usando la expresión $x_r(t) = x(nT) + \{[x(nT)-x(nT-T)]/T\}(t-nT) \quad nT \leq t \leq (n + 1)T$. Intenta reconstruir la señal usando un retenedor de orden uno (puedes ayudarte de la función floor() de matlab).