



Laboratorio 1: Pulse Amplitude Modulation (PAM) y Pulse Code Modulation (PCM)

Lucas Gonzalez, lucasgonzalez1@mail.udp.cl
 Sebastian Gulfo, Sebastian.gulfo@mail.udp.cl
 Escuela de Informática y Telecomunicaciones
 Universidad Diego Portales

Resumen—El presente laboratorio aborda la codificación de formas de onda analógicas provenientes de fuentes analógicas en señales digitales de banda base. Se demuestra que la aproximación digital a la señal analógica puede alcanzarse con gran precisión, permitiendo la emisión de señales con un ancho de banda optimizado. Mediante la aplicación de técnicas de modulación por amplitud de pulso (PAM) y modulación por código de pulsos (PCM), se analizó la influencia de variables como la frecuencia de muestreo y el ciclo de trabajo en la conformación del espectro. Los resultados obtenidos confirman que, combinando de forma eficiente datos digitales y digitalizados, es posible implementar sistemas de comunicación robustos y de propósito general, beneficiándose de la flexibilidad y economía que ofrecen los circuitos digitales.

I. INTRODUCCION

La transmisión y recepción de señales es algo esencial en las redes, ya que hay varios procesos importantes que se relacionan con esto. Uno de esos procesos es la conversión de una señal análoga a una digital, y ahí es donde entran las modulaciones PAM y PCM. En este laboratorio vamos a usar MATLAB para simular cómo una señal análoga se transforma en una señal digital usando estas dos modulaciones. El objetivo principal es entender cómo funcionan estos métodos y analizar la modulación PAM tanto en el tiempo como en la frecuencia, para así comprender mejor las variables que intervienen y cómo afectan a la señal.

II. METODOLOGÍA

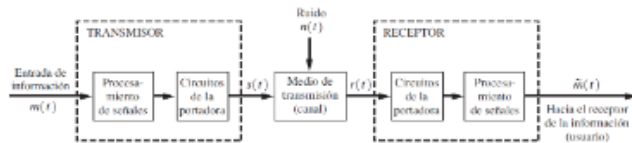


Figura 1. Función Sistema de comunicación

III. RESULTADOS Y ANÁLISIS

Esta parte del laboratorio se relaciona con la sección de "Procesamiento de señales" que aparece en la Figura 1. Lo que se hace aquí es aplicar la modulación PAM a una señal senoidal usando dos tipos de muestreo: natural e instantáneo. La señal que se usa es $m(t)$, tiene una amplitud de 1 y una frecuencia de 1000 Hz. Como MATLAB no trabaja directamente con señales analógicas, se toma la señal muestreada cada 10 microsegundos

La variable t_m (10 μs) define cada cuánto se toman muestras de la señal $m(t)$. Por otro lado, f_s es la frecuencia de muestreo que se usa en la modulación PAM, y t_s representa cuánto dura cada ciclo de muestreo. La variable d indica el tiempo durante el cual se mantiene el valor de la muestra (que en este caso es la mitad de t_s), y δ es el ciclo de trabajo, que muestra qué proporción del ciclo se mantiene el valor de la muestra. Estos mismos parámetros se usan tanto para el muestreo instantáneo como para el natural.

El vector t corresponde al vector del tiempo y m_t corresponde a la señal senoidal $m(t)$. R y S son variables auxiliares que se usan al realizar el muestreo natural e instantáneo para determinar los ciclos de muestreo. La variable R indica cada cuántas muestras de la señal senoidal original se van a tomar muestras para la modulación PAM (Cada 10 muestras de la señal senoidal se va a tomar 1 muestra para la modulación). La variable S indica cuántas muestras de la señal senoidal se van a mantener en el nivel máximo en la señal modulada (Cada 5 muestras de la señal senoidal se va a mantener el nivel máximo en la señal modulada).

En esa parte del código (Figura 4), se implementa el muestreo natural. Primero se crea un vector llamado $snat$ que tiene el mismo tamaño que el vector de tiempo, pero lleno de ceros. Después, se recorre la señal $m(t)$ y, cada cierta cantidad de muestras (definida por R), se insertan unos durante S posiciones seguidas. Esos unos marcan los momentos en los que se toma una muestra. Finalmente, se multiplica $snat$ por la señal original $m(t)$, y así se obtiene la señal modulada con muestreo natural, que se guarda como $mtnat$.

El muestreo instantáneo funciona parecido al natural, pero con una diferencia importante. En vez de marcar con unos los momentos en que se toma la muestra, como se

hizo antes, acá se copia directamente el valor de la señal $m(t)$ en ese instante y se repite durante S muestras. Para eso, se parte con un vector de ceros llamado mt_{inst} , y se va llenando con esos valores cada R muestras. Luego, al graficar la señal original junto con las versiones con muestreo natural e instantáneo, se obtienen las curvas que se muestran en la Figura 6.

IV. TRANSFORMADAS DE FOURIER

Usando el mismo código de antes, ahora se calcula la transformada de Fourier. Esta se aplica tanto a la señal original como a sus versiones con muestreo natural e instantáneo, para poder analizar su comportamiento en el dominio de la frecuencia.

Para poder graficar las transformadas de las señales, primero se crea el eje de frecuencias. Ahí se calculan las FFT de la señal original (Mt), del muestreo natural (Mt_{nat}) y del instantáneo (Mt_{inst}). Además del uso de la función `fft()`, el código también está ajustado para que solo se muestre la parte positiva del espectro, que es la que realmente interesa en este análisis. Al hacer esto, se obtienen las gráficas que se muestran en la Figura 8.

V. TRANSFORMADAS CON VARIACIÓN EN EL CICLO DE TRABAJO

En esta parte se analiza qué pasa con las transformadas de Fourier cuando se cambia el ciclo de trabajo. Este valor se calcula como $\frac{\tau}{t_s}$, donde τ es el tiempo en que se mantiene la muestra y t_s es la duración del ciclo completo. Para hacer la prueba, se aumenta el ciclo de trabajo de 0,5 a 0,8 y se observan los efectos en las señales muestreadas (Figura9) y en sus transformadas (Figura10).

Después, al reducir el ciclo de trabajo a 0,2, se pueden ver cómo cambian las señales moduladas y sus respectivas transformadas. Estas se muestran en la Figura 11 y la Figura 12.

VI. TRANSFORMADAS CON VARIACIÓN EN LA FRECUENCIA DE MUESTREO

Después de volver a usar un ciclo de trabajo de 0,5, se analiza qué efecto tiene cambiar la frecuencia de muestreo f_s en las transformadas de Fourier. Primero, se baja f_s de 10.000 Hz a 5.000 Hz, y con eso se obtienen las señales y transformadas que aparecen en las Figuras 14 y 15

Después, al aumentar la frecuencia de muestreo a 25.000 Hz, se obtienen las señales muestreadas que aparecen en la Figura 16, y sus respectivas transformadas de Fourier en la Figura 17.

VII. PCM Y CUANTIFICACIÓN

Para cerrar el laboratorio, se aplicó modulación por pulsos codificados (PCM) a la señal $m(t)$ que ya había sido muestreada de forma instantánea. Para ello, se utilizó un código específico que permitió realizar el proceso paso a paso.

En esta parte se define N como la cantidad de bits usados para la codificación PCM. En este caso, se comenzó utilizando un valor de $N = 64$. A partir de este valor, se calcula la variable `pcm_levels`, que representa la cantidad de niveles de cuantización disponibles, y que depende directamente del valor de N mediante la relación:

$$\text{pcm_levels} = 2^N$$

Para poder cuantificar la señal instantánea $m_{inst}(t)$, primero se le suma 1 a todos sus valores para que queden en un rango positivo, lo cual facilita el proceso de cuantificación. Luego, la señal se multiplica por `pcm_levels - 1` para distribuir sus valores dentro del rango permitido por el número de niveles. Finalmente, se divide por 2, centrando así la señal alrededor del valor medio y simplificando el proceso de codificación.

Las tres señales — $m(t)$, $m_{inst}(t)$ y `pcm_signal_inst`— se normalizan para que todas queden en el mismo rango de amplitud, el cual va desde 0 hasta 1.

Calcular el error de cuantización:

El error de cuantización se obtiene restando la señal original modulada (`mt_inst`) con la señal que fue cuantizada y luego reconstruida desde PCM a PAM. Para hacer este cálculo, se usa la siguiente fórmula:

$$\text{quantization_error_inst} = m_{t_inst} - \left(\frac{2 \times \text{pcm_signal_inst}}{\text{pcm_levels} - 1} - 1 \right)$$

Esta operación sirve para ajustar los valores cuantizados y que queden dentro del rango de -1 a 1. Una vez que la señal PCM se reconstruye, el error se calcula simplemente restando esa señal reconstruida a la señal original modulada.

VIII. RESULTADOS Y ANÁLISIS

VIII-A. Modulación PAM de una señal senoidal

Al mirar los gráficos de la modulación, se nota claramente la diferencia entre el muestreo natural y el instantáneo. En el muestreo instantáneo, el valor que se toma se mantiene fijo durante todo el tiempo τ . En cambio, en el muestreo natural, dentro de ese mismo intervalo, la señal sigue el mismo comportamiento que tenía la función seno original.

VIII-B. Fourier

En las tres transformadas aparece un pico en $X = 1000$, lo que tiene sentido porque esa es justamente la frecuencia de la señal senoidal original, que se mantiene constante durante toda la señal. Ahora bien, si se comparan las transformadas de las señales moduladas (tanto la natural como la instantánea) con la de la señal original, se ven diferencias: aparecen otros picos en $X = 9000$ y $X = 11000$. Estos valores están cerca de la frecuencia de muestreo $f_s = 10000$ Hz, por lo que su posición podría estar relacionada con ese parámetro.

Cuando se modifica el ciclo de trabajo, lo que cambia es la proporción del tiempo en que se toma la muestra dentro de cada ciclo de muestreo. Si ese valor se hace más grande, la transformada de la señal modulada empieza a parecerse más a la transformada de la señal original $m(t)$. Además, se puede notar que al aumentar el ciclo de trabajo, la diferencia entre el pico en $X = 1000$ y el resto de los picos se hace más notoria. En cambio, si se reduce el ciclo, esa diferencia disminuye.

Ciclo de trabajo = 0.5, $f_c = 5000[\text{Hz}]$:

Ya se había mencionado antes que aparecen picos cerca del valor de f_c , y eso vuelve a verse claramente cuando se modifican los valores de esa frecuencia. En los tres casos analizados, el promedio de las posiciones de esos picos coincide con f_c . Además, se nota que mientras más alto es el valor de f_c , más chicos se ven esos picos, y si f_c es menor, los picos aumentan. Por otro lado, al comparar las gráficas también se observa que cuando se sube la frecuencia de muestreo f_s , las señales resultantes se parecen más a la señal original, aunque eso puede implicar un mayor uso de recursos.

VIII-C. PCM y Cuantización

Al graficar las tres señales juntas, se observa que la envolvente del tren de pulsos generado por la cuantización PCM sigue el mismo perfil que la señal PAM instantánea.

A continuación, se analiza el **error de cuantificación** para distintos valores del número de bits utilizados en PCM:

Número de bits para PCM: $N = 64$: En el gráfico del error de cuantificación con $N = 64$, se aprecia que la diferencia entre la señal original y la cuantizada varía entre 0 y 1, considerando el valor absoluto de dicha diferencia.

Número de bits para PCM: $N = 128$: Al aumentar N a 128, el comportamiento del error se mantiene muy similar: los valores siguen oscilando dentro del rango de 0 a 1. Esto indica que, al usar una cantidad mayor de bits, el sistema mantiene una cuantificación precisa y estable.

Número de bits para PCM: $N = 32$: En cambio, al reducir N a 32, el error de cuantificación aumenta notablemente. La diferencia entre la señal original y la cuantizada puede llegar hasta 2,5, y se observa que los errores son más frecuentes a lo largo del tiempo. Esto refleja una menor precisión debido a la reducción en la cantidad de niveles de cuantización disponibles.

Conclusión: Estos resultados demuestran que el error de cuantificación está directamente relacionado con el número de bits utilizados en el sistema PCM. Cuando se utilizan 64 bits o más (preferentemente potencias de 2), el error se mantiene prácticamente constante y bajo. Sin embargo, al utilizar menos de 64 bits, la cantidad de errores aumenta significativamente, ya que hay menos niveles para representar los valores de la señal original.

IX. CONCLUSIONES

En resumen, este laboratorio sirvió para entender mejor cómo una señal analógica puede transformarse en una

señal digital usando modulación PAM y PCM. A través de las simulaciones hechas en MATLAB, fue posible analizar cómo se comportan distintas variables durante los procesos de muestreo, modulación y cuantificación.

Con el uso de la transformada de Fourier, se pudo ver cómo influyen el ciclo de trabajo y la frecuencia de muestreo en la forma final de la señal. Por ejemplo, al aumentar f_s , las señales muestreadas se parecen más a la señal original. También se compararon los dos tipos de muestreo en PAM —natural e instantáneo— y se vieron sus diferencias en cuanto a fidelidad.

Finalmente, al trabajar con modulación PCM, se estudió cómo el número de bits afecta la calidad de la señal. Se vio que usar más bits mejora la precisión, ya que hay más niveles para representar los valores. Esto ayudó a entender mejor el equilibrio entre la cantidad de información y la calidad de la señal digital que se obtiene.

REFERENCIAS

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.
- [2] L. Gonzalez y S. Gulfo, *Laboratorio 1 - Comunicaciones Digitales (PAM y PCM)*, Repositorio GitHub, 2025. Disponible en: <https://github.com/lucowskyx/lab-1-comunicaciones>

APÉNDICE A ANEXO DE FIGURAS

```
% parámetros de configuración
fm = 10000; % Hz
tm = 1/fm; % segundos
ls = 500; % largo de la señal
fs = 10000; % Hz
ts = 1/fs; % segundos
tau = 0.5*ts; % segundos
d = tau/ts; % ciclo de trabajo
```

Figura 2. Parámetros de configuración

```
% vectores
t = (0:ls-1)*tm;
mt = sin(2*pi*f_c*t);
% auxiliaries
r = floor(ts/tm);
s = floor(tau/tm);
```

Figura 3. Declaración de vectores y variables auxiliares

```
% muestreo natural]
s_nat = zeros(1,length(t));
for i=1:length(m_t)
    if mod(i,r)==0
        s_nat(i:i+s) = 1;
    end
end
s_nat = s_nat(1:length(t));
m_t_nat = m_t.*s_nat;
```

Figura 4. Muestreo natural en MATLAB

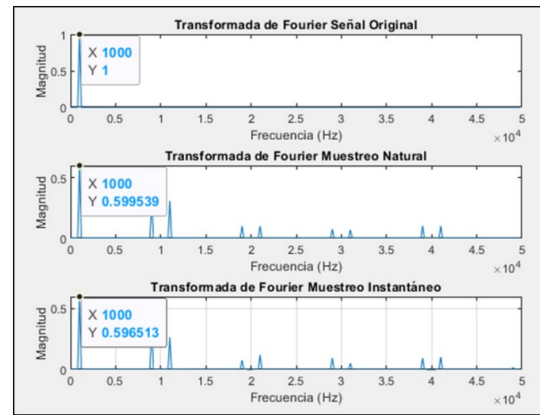


Figura 8. Gráficas de las transformadas de Fourier

```
% muestreo instantaneo
m_t_inst = zeros(1,length(t));
for i=1:length(m_t)
    if mod(i,r)==0
        m_t_inst(i:i+s) = m_t(i);
    end
end
m_t_inst = m_t_inst(1:length(t));
```

Figura 5. Muestreo instantáneo en MATLAB

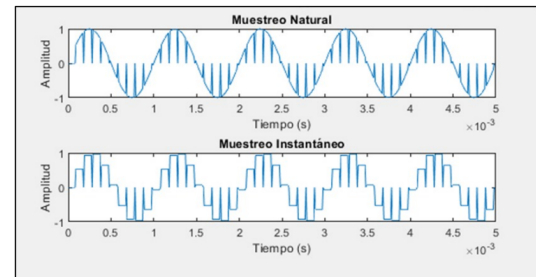


Figura 9. Muestreo con ciclo de trabajo = 0.8

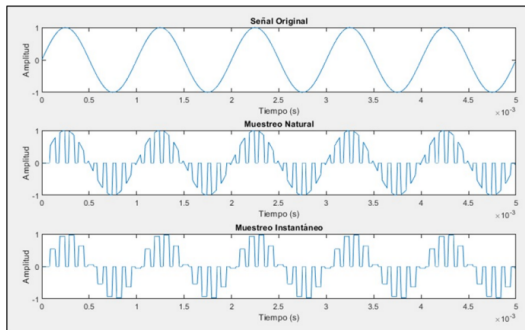


Figura 6. Señal original, muestreo natural e instantáneo

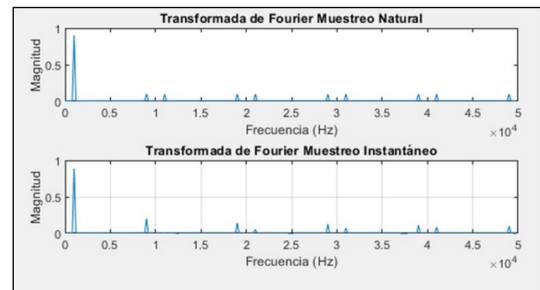


Figura 10. Transformada con ciclo de trabajo = 0.8

```
% Transformadas de Fourier
N = length(t);
f_axis = (0:N/2-1) * fm / N; % Solo la mitad positiva del eje de frecuencia
H_t = 2.*abs(fft(m_t)/N);
H_t_nat = 2.*abs(fft(m_t_nat)/N);
H_t_inst = 2.*abs(fft(m_t_inst)/N);
```

Figura 7. Transformadas de Fourier en MATLAB

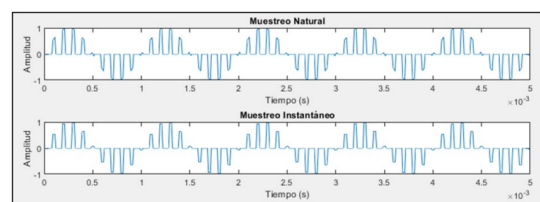


Figura 11. Muestreo con ciclo de trabajo = 0.2

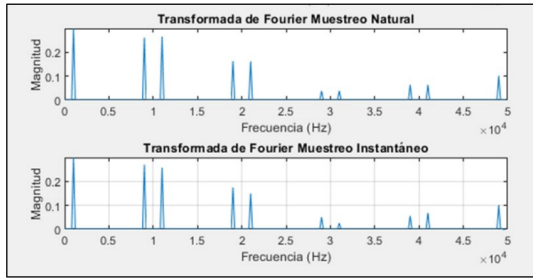


Figura 12. Transformada con ciclo de trabajo = 0.2

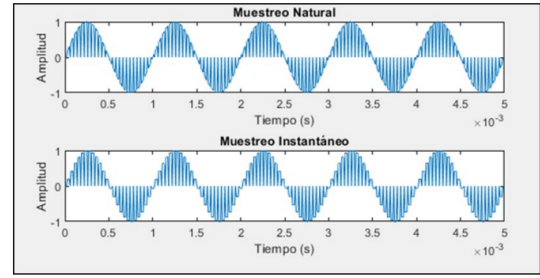


Figura 16. Muestreo con frecuencia de muestreo $f_s = 25000$ Hz

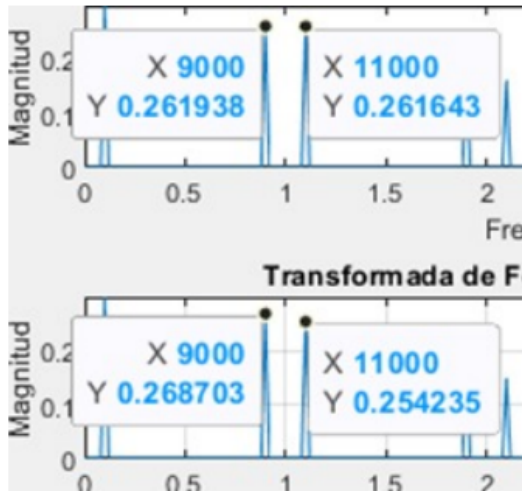


Figura 13. Transformada con ciclo de trabajo = 0.2

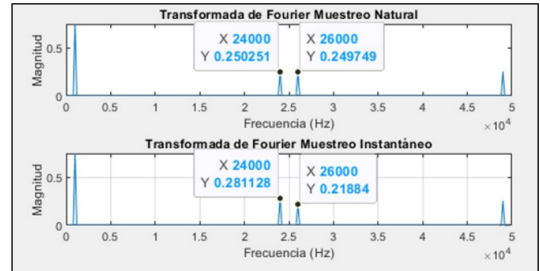


Figura 17. Transformada con frecuencia de muestreo $f_s = 25000$ Hz

```
% Parámetros PCM
N = 64; % Número de bits para PCM
pcm_levels = 2^N; % Total de niveles PCM

% Cuantizar la señal instantánea usando PCM
pcm_signal_inst = round((m_t_inst + 1) * (pcm_levels - 1) / 2); % Cuantización
```

Figura 18. Parámetros PCM y cuantización

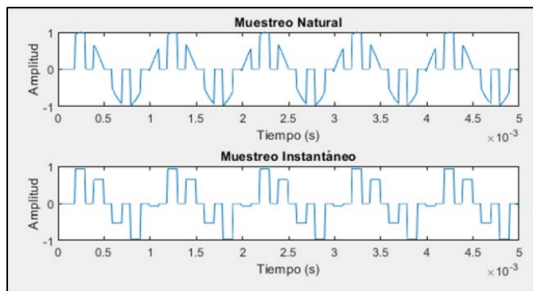


Figura 14. Muestreo con frecuencia de muestreo $f_s = 5000$ Hz

```
% Normaliza las señales para que estén en el mismo rango de amplitud (0 a 1)
m_t_norm = (m_t - min(m_t)) / (max(m_t) - min(m_t));
m_t_inst_norm = (m_t_inst - min(m_t_inst)) / (max(m_t_inst) - min(m_t_inst));
pcm_signal_inst_norm = (pcm_signal_inst - min(pcm_signal_inst)) / (max(pcm_signal_inst) - min(pcm_signal_inst));
```

Figura 19. Parámetros PCM y cuantización (parte 2)

```
% Calcular el error de cuantización para la señal PAM cuantificada (PCM)
quantization_error_inst = m_t_inst - ((2 * pcm_signal_inst / (pcm_levels - 1)) - 1);
```

Figura 20. Cálculo del error de cuantización

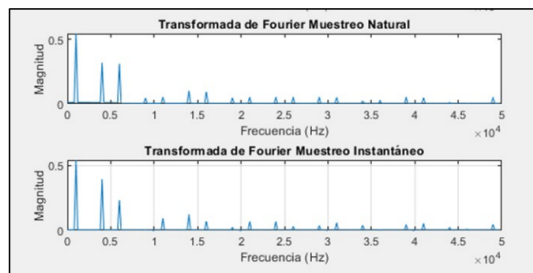


Figura 15. Transformada con frecuencia de muestreo $f_s = 5000$ Hz

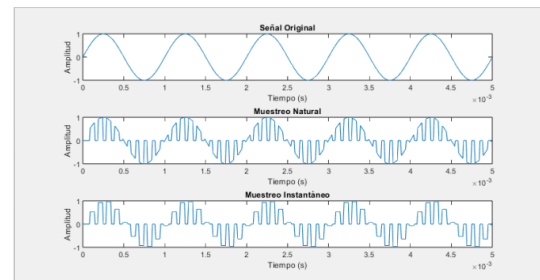


Figura 21. Señal original, PAM instantánea y señal cuantificada (PCM)

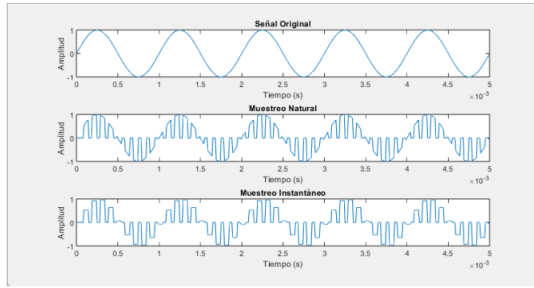
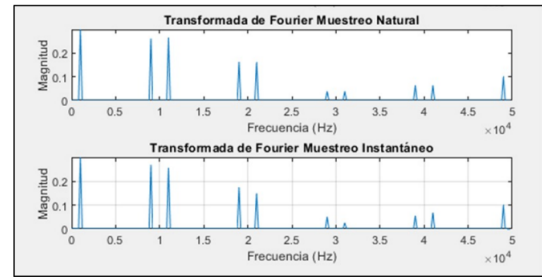
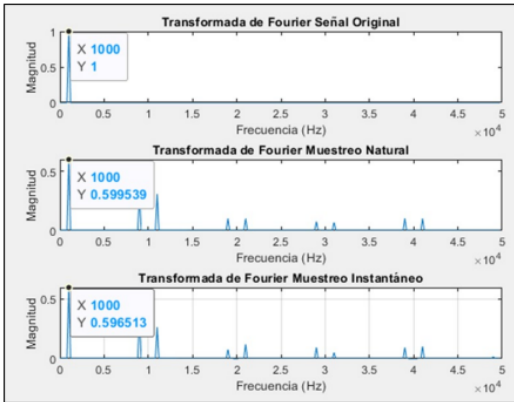
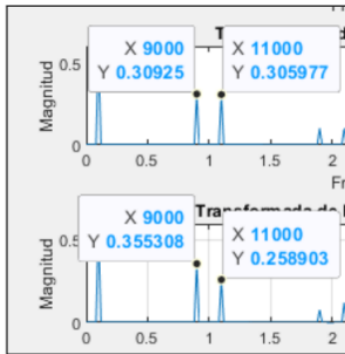
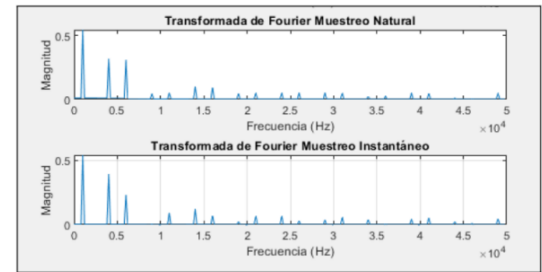
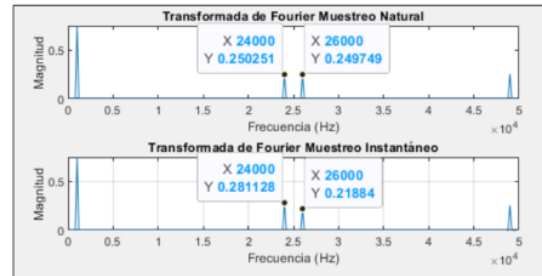
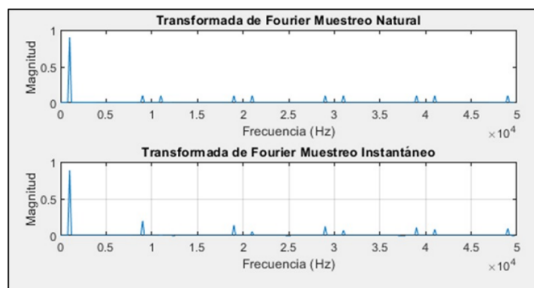
Figura 22. Transformada: ciclo 0.5, $f_c = 10000$ HzFigura 25. Transformada: ciclo 0.5, $f_c = 5000$ HzFigura 23. Transformada: ciclo 0.8, $f_c = 10000$ HzFigura 24. Transformada: ciclo 0.2, $f_c = 10000$ HzFigura 26. Transformada: ciclo 0.5, $f_c = 25000$ HzFigura 27. Transformada con variación de f_s 

Figura 28. PCM vs PAM instantánea

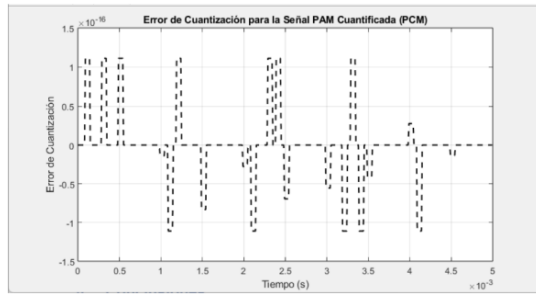


Figura 29. Error de cuantización con $N = 64$

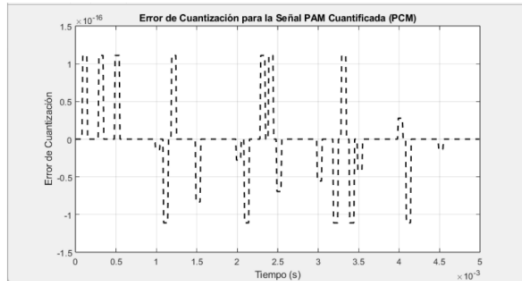


Figura 30. Error de cuantización con $N = 128$

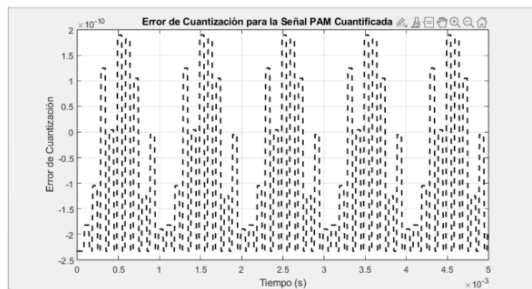


Figura 31. Error de cuantificación con $N = 32$