

# PROCESAMIENTO DE SEÑALES EN TIEMPO DISCRETO

## Informes de Laboratorios

*Alumno: Ing. Martín N. Gonella*

*Profesor: Dr. Ing. Mario R. Hueda*

8 de mayo de 2017

# Resumen

En el presente trabajo, se exponen los resultados de simulación obtenidos a partir de la realización de los diferentes Laboratorios del curso de *Procesamiento de Señales en Tiempo Discreto*.

Los Laboratorios consisten mayormente en simulaciones, las cuales fueron realizadas en su totalidad en MATLAB.

En aquellos casos que se consideró necesario, se agregó una breve introducción teórica a la temática tratada. Sin embargo, el objetivo principal del presente trabajo es mostrar los resultados y las conclusiones que se destacan a partir de estos últimos.

Los códigos utilizados para la realización de las simulaciones fueron provistos por el docente de la materia, el *Dr. Ing. Mario R. Hueda* y se pueden encontrar en la sección de Anexos, más precisamente en el *Anexo B*. En algunos casos los códigos originales fueron levemente modificados, además se agregó código propio cuando se consideró apropiado.

# Índice general

<b>Resumen</b>	<b>I</b>
<b>Índice de Figuras</b>	<b>III</b>
<b>1. Filtro Coseno Realzado</b>	<b>1</b>
1.1. Descripción y Consignas . . . . .	1
1.2. Resultados . . . . .	2
<b>2. Transformador de Hilbert</b>	<b>28</b>
2.1. Descripción y Consignas . . . . .	28
2.2. Resultados . . . . .	30
<b>3. Lazo de Enganche de Fase (PLL)</b>	<b>50</b>
3.1. Descripción y Consignas . . . . .	50
3.2. Resultados . . . . .	52
<b>4. Procesamiento Tiempo Discreto-Continuo</b>	<b>81</b>
4.1. Descripción y Consignas . . . . .	81
4.2. Resultados . . . . .	82
<b>5. Modulador Sigma-Delta de Segundo Órden</b>	<b>100</b>
5.1. Descripción y Consignas . . . . .	100
5.2. Resultados . . . . .	103
<b>6. Conclusiones</b>	<b>125</b>
<b>Bibliografía</b>	<b>127</b>
<b>Apéndices</b>	<b>128</b>
<b>A. Algoritmo de Welch para la Estimación de la PSD</b>	<b>129</b>
A.1. Código de MATLAB . . . . .	130

<b>B.</b>	<b>132</b>
B.1. Código del Laboratorio N°1 . . . . .	132
B.2. Código del Laboratorio N°2 . . . . .	136
B.3. Código del Laboratorio N°2 - Parte B . . . . .	140
B.4. Código Laboratorio N°3 . . . . .	143
B.5. Código del Lab. N°3 - Jitter Transfer Teórico . . . . .	151
B.6. Código Laboratorio N°4 . . . . .	153
B.7. Código Laboratorio N°5 . . . . .	159

# Índice de figuras

1.1.	Respuesta al impulso del filtro coseno realzado, con $\beta = 0$ . . . . .	2
1.2.	Respuesta en frecuencia (Magnitud y Fase) del filtro coseno realzado, con $\beta = 0$ . . . . .	3
1.3.	Respuesta al impulso del filtro coseno realzado, con $\beta = 0,2$ . . . . .	4
1.4.	Respuesta en frecuencia (Magnitud y Fase) del filtro coseno realzado, con $\beta = 0,2$ . . . . .	5
1.5.	Respuesta al impulso del filtro coseno realzado, con $\beta = 0,8$ . . . . .	6
1.6.	Respuesta en frecuencia (Magnitud y Fase) del filtro coseno realzado, con $\beta = 0,8$ . . . . .	7
1.7.	Respuesta al impulso del filtro coseno realzado, con $\beta = 1$ . . . . .	8
1.8.	Respuesta en frecuencia (Magnitud y Fase) del filtro coseno realzado, con $\beta = 1$ . . . . .	9
1.9.	Símbolos generados con tasa de sobremuestreo, $M = 1$ (Ningún cero insertado). . . . .	10
1.10.	Símbolos generados con tasa de sobremuestreo, $M = 4$ (3 ceros insertados). . . . .	10
1.11.	Símbolos generados con tasa de sobremuestreo, $M = 8$ (7 ceros insertados). . . . .	11
1.12.	Símbolos generados con tasa de sobremuestreo, $M = 16$ (15 ceros insertados). . . . .	11
1.13.	Señal transmitida, $M = 8$ y $\beta = 0$ . . . . .	12
1.14.	Señal transmitida (Zoom), $M = 8$ y $\beta = 0$ . . . . .	13
1.15.	Señal transmitida, $M = 8$ y $\beta = 0,1$ . . . . .	14
1.16.	Señal transmitida (Zoom), $M = 8$ y $\beta = 0,1$ . . . . .	14
1.17.	Señal transmitida, $M = 8$ y $\beta = 0,5$ . . . . .	15
1.18.	Señal transmitida (Zoom), $M = 8$ y $\beta = 0,5$ . . . . .	15
1.19.	Señal transmitida, $M = 8$ y $\beta = 1$ . . . . .	16
1.20.	Señal transmitida (Zoom), $M = 8$ y $\beta = 1$ . . . . .	17
1.21.	Ejemplo de Diagrama de Ojo. . . . .	18
1.22.	Diagrama de Ojo, $M = 8$ y $\beta = 0$ . . . . .	19
1.23.	Diagrama de Ojo, $M = 8$ y $\beta = 0,1$ . . . . .	20
1.24.	Diagrama de Ojo, $M = 8$ y $\beta = 0,5$ . . . . .	21
1.25.	Diagrama de Ojo, $M = 8$ y $\beta = 1$ . . . . .	22
1.26.	Símbolos generados con tasa de sobremuestreo $M = 4$ y $\beta = 0,2$ . . . . .	23
1.27.	Símbolos generados, con tasa de sobremuestreo $M = 16$ y $\beta = 0,2$ . . . . .	24
1.28.	Señal transmitida, con tasa de sobremuestreo $M = 4$ y $\beta = 0,2$ . . . . .	25
1.29.	Señal transmitida, con tasa de sobremuestreo $M = 16$ y $\beta = 0,2$ . . . . .	25
1.30.	Diagrama de Ojo, $M = 4$ y $\beta = 0,2$ . . . . .	26

1.31. Diagrama de Ojo, $M = 16$ y $\beta = 0,2$ . . . . .	27
2.1. Sistema de Comunicaciones Digitales Elemental. . . . .	29
2.2. Respuesta al impulso del filtro coseno realizada (arriba) y del filtro transformador de Hilbert (abajo). . . . .	31
2.3. Respuesta en frecuencia (magnitud y fase) del filtro Transformador de Hilbert.	32
2.4. Símbolos complejos 4-QAM. . . . .	33
2.5. PSD de la señal en banda base $y[n]$ (No analítica). . . . .	34
2.6. PSD de la señal analítica $s[n]$ . . . . .	35
2.7. PSD de la señal en banda base $y[n]$ . . . . .	36
2.8. PSD de la señal analítica $s[n]$ . . . . .	37
2.9. PSD de la señal de la parte Real de $s[n]$ . . . . .	38
2.10. PSD de la señal a la salida del filtro de partición de fase. . . . .	39
2.11. PSD de la señal recuperada. . . . .	40
2.12. Comparación entre la parte Imaginaria de la señal transmitida y la recibida.	41
2.13. Tono de pulsación presente en el espectro de la señal $E\{ p[n] ^2\}$ , con $M = 8$ .	45
2.14. Tono de pulsación presente en el espectro de la señal $E\{ p[n] ^2\}$ , con $M = 4$ .	46
2.15. Tono de pulsación presente en el espectro de la señal $E\{ \tilde{p}[n] ^2\}$ . . . . .	48
2.16. Tono de pulsación cancelado, en el espectro de la señal $E\{ u[n] ^2\}$ . . . . .	49
 3.1. Diagrama en Bloques de un PLL Discreto. . . . .	50
3.2. Diagrama en Bloques básico de un PLL de tiempo continuo. . . . .	53
3.3. Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo $L_1(z)$ , $D = 0$ y varios $K_p$ . . . . .	58
3.4. Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo $L_1(z)$ , $D = 20$ y varios $K_p$ . . . . .	59
3.5. Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo $L_1(z)$ , $D = 1000$ y varios $K_p$ . . . . .	61
3.6. Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo $L_2(z)$ , $D = 0$ y varios $K_p$ y $K_i$ . . . . .	64
3.7. Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo $L_2(z)$ , $D = 20$ y varios $K_p$ y $K_i$ . . . . .	65
3.8. Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo $L_2(z)$ , $D = 100$ y varios $K_p$ y $K_i$ . . . . .	66
3.9. Función de Transferencia del PLL, $D = 0$ y distintos valores de $K_p$ y $K_i$ . . .	68
3.10. Función de Transferencia del PLL, $D = 20$ y distintos valores de $K_p$ y $K_i$ . . .	69
3.11. Función de Transferencia del PLL, $D = 100$ y distintos valores de $K_p$ y $K_i$ . . .	70
3.12. Salidas $\phi[n]$ , para diferentes entradas $\theta[n]$ . . . . .	71
3.13. Señales Intermedias: Error de Fase, Salida Parte Proporcional y Salida Parte Integral. . . . .	72
3.14. Presencia de tono de Jitter en la PSD de la Salida del PPL. . . . .	73
3.15. Curvas de Jitter Transfer con $K_p/K_i = 100$ . . . . .	75

3.16. Curvas de Jitter Transfer con $K_p/K_i = 1000$ .	77
3.17. Curvas de Jitter Transfer con $K_p/K_i = 10000$ .	79
4.1. Diagrama de Bloques de un Módem.	81
4.2. Pulso Coseno Realzado $g[n]$ con $\beta = 0,2$ , $M = 4$ .	84
4.3. Filtro Reconstructor, Coseno realzado con $\beta = 0,8$ , $M = 4$ .	85
4.4. Filtro Reconstructor, Coseno realzado con $\beta = 0,1$ , $M = 4$ .	86
4.5. Filtro Reconstructor, Pulso Rectangular, $M = 4$ .	86
4.6. Filtro Reconstructor, Seno Cardinal, $M = 4$ .	87
4.7. Filtro Reconstructor, Butterworth órden= 10, $M = 4$ .	88
4.8. Filtro Reconstructor, Butterworth órden= 10 con ventana Rectangular, $M = 4$ .	88
4.9. Todos los Filtros Reconstructores en una misma gráfica, $M = 4$ .	89
4.10. PSD de la Salida para los distintos filtros de reconstrucción, $M = 4$ .	90
4.11. Señal $y[n]$ vs. Salidas de los Filtros de Reconstrucción, $M = 4$ .	91
4.12. Pulso Coseno Realzado, $g[n]$ , $\beta = 0,2$ , $M = 8$ .	92
4.13. Filtro Reconstructor, Coseno realzado con $\beta = 0,8$ , $M = 8$ .	93
4.14. Filtro Reconstructor, Coseno realzado con $\beta = 0,1$ , $M = 8$ .	93
4.15. Filtro Reconstructor, Pulso Rectangular, $M8$ .	94
4.16. Filtro Reconstructor,Seno Cardinal, $M8$ .	94
4.17. Filtro Reconstructor, Butterworth órden= 10, $M = 8$ .	95
4.18. Filtro Reconstructor, Butterworth órden= 10 con ventana Rectangular, $M = 8$ .	95
4.19. Todos los Filtros Reconstructores en una misma gráfica, $M = 8$ .	96
4.20. PSD de la Salida para los distintos filtros de reconstrucción, $M = 8$ .	97
4.21. Señal $y[n]$ vs. Salidas de los Filtros de Reconstrucción, $M = 8$ .	97
4.22. Respuesta total del filtro interpolador y los diferentes filtros reconstructores, $M = 4$ .	98
4.23. Respuesta total del filtro interpolador y los diferentes filtros reconstructores, $M = 4$ .	99
5.1. Modulador $\Sigma\Delta$ de segundo orden.	100
5.2. Diezmador usando Filtro CIC de dos etapas.	100
5.3. Diagrama de Bloques de un Convertidor A/D $\Sigma\Delta$ .	103
5.4. Modulador $\Sigma\Delta$ de segundo orden.	104
5.5. Modulador $\Sigma\Delta$ de segundo orden.	106
5.6. Diezmador usando Filtro CIC de dos etapas.	106
5.7. Salidas Modulador $\Sigma\Delta$ y CIC, $M = 16$ , $R = 4$ .	107
5.8. Análisis de las diferentes señales del modulador $\Sigma\Delta$ .	108
5.9. Diagramas de Ojo, $M = 16$ , $R = 4$ .	109
5.10. PSD Salida CIC con y sin Filtrada LPF, $M = 16$ , $R = 4$ .	110
5.11. Salidas Modulador $\Sigma\Delta$ y CIC, $M = 16$ , $R = 8$ .	111
5.12. Análisis de las diferentes señales del modulador $\Sigma\Delta$ .	112
5.13. Diagramas de Ojo, $M = 16$ , $R = 8$ .	112

5.14. PSD Salida CIC con y sin Filtrada LPF, $M = 16, R = 8$ .	113
5.15. Salidas Modulador $\Sigma\Delta$ y CIC, $M = 8, R = 8$ .	114
5.16. Análisis de las diferentes señales del modulador $\Sigma\Delta$ .	115
5.17. Diagramas de Ojo, $M = 8, R = 8$ .	115
5.18. PSD Salida CIC con y sin Filtrada LPF, $M = 8, R = 8$ .	116
5.19. Salidas Modulador $\Sigma\Delta$ y CIC, $M = 8, R = 4$ .	117
5.20. Análisis de las diferentes señales del modulador $\Sigma\Delta$ .	118
5.21. Diagramas de Ojo, $M = 8, R = 4$ .	118
5.22. PSD Salida CIC con y sin Filtrada LPF, $M = 8, R = 4$ .	119
5.23. Salidas Modulador $\Sigma\Delta$ y CIC, $M = 16, R = 4$ .	120
5.24. Diagramas de Ojo, $M = 16, R = 4$ .	121
5.25. PSD Salida CIC con y sin Filtrada LPF, $M = 16, R = 4$ .	122
5.26. Ruido de Cuantización y Factor de Sobremuestreo.	123

# Capítulo 1

## Filtro Coseno Realzado

### 1.1. Descripción y Consignas

En primer lugar se define un filtro muy utilizado en comunicaciones digitales, el filtro de caída cosenoidal ó filtro coseno realzado, dado por:

$$g(t) = \text{sinc}(t/T) \frac{\cos(\pi\beta t/T)}{1 - 4\beta^2 t^2/T^2} \quad (1.1)$$

Donde  $\beta$ , es el factor de roll-off y cuyos valores pueden ser  $0 \leq \beta \leq 1$ . Con  $T$  como el tiempo entre símbolos.

Además, sea  $g[n] = g(nT_s)$ , con  $1/T_s$  la frecuencia de sobremuestreo, tal que  $T/T_s = M$ , con  $M$  un número entero, por ejemplo  $M = 8$ .

Se propone desarrollar las siguientes actividades:

1. Utilizar la autofunción  $e^{j2\pi f T_s n}$ , para obtener por simulación la respuesta  $|G(e^{j2\pi f T_s})|$ , en función de  $2\pi f T_s$  con  $|f| < 1/T$ , para  $\beta = 1$ .
2. Sea  $x[n] = \sum_{k=0}^{L-1} a[k] \delta[n - kM]$  con  $a[k] \in \{\pm 1\}$ , siendo un proceso aleatorio blanco con símbolos equiprobables. Determinar por simulación la señal de salida y generar el diagrama de ojo correspondiente. Además, se debe repetir la simulación para  $\beta = 0,1$  y  $0,5$ .

## 1.2. Resultados

Un filtro coseno realizado es un tipo de filtro electrónico, utilizado frecuentemente en sistemas de comunicaciones debido a que es capaz de reducir al mínimo la interferencia entre símbolos (ISI). Se llama así, porque la parte no nula del espectro frecuencial es un coseno que, en su forma más simple ( $\beta = 1$ ), se encuentra *alzado* para situarse por encima del eje horizontal (eje de frecuencias).

El filtro de coseno realizado es una implementación de un filtro pasa bajo de Nyquist. Con lo cual, el espectro tendrá simetría par en  $1/2T$ , donde  $T$  es el período del sistema de comunicaciones.

A continuación, se muestran las respuestas al impulso y las respuestas en frecuencia del filtro coseno realizado, para distintos valores del factor de roll-off ( $\beta$ ) y con una tasa de sobremuestreo fija de  $M = 8$ .

Respuestas al Impulso y en Frecuencia, para  $\beta = 0$  y  $M = 8$

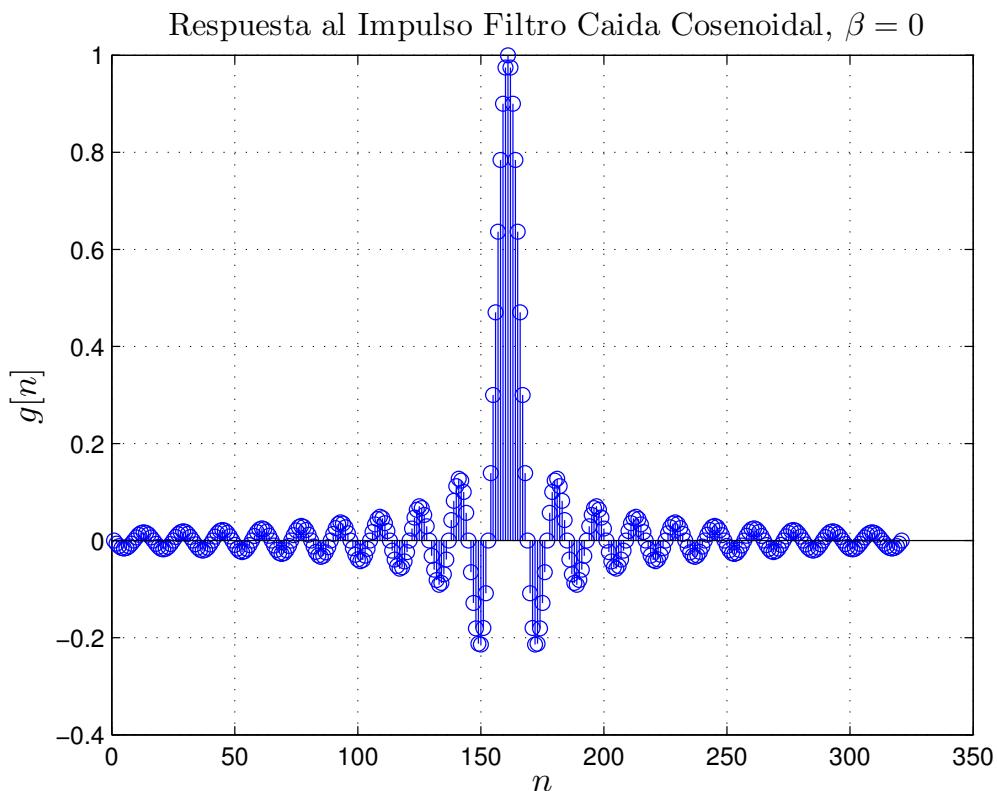


Figura 1.1: Respuesta al impulso del filtro coseno realizado, con  $\beta = 0$ .

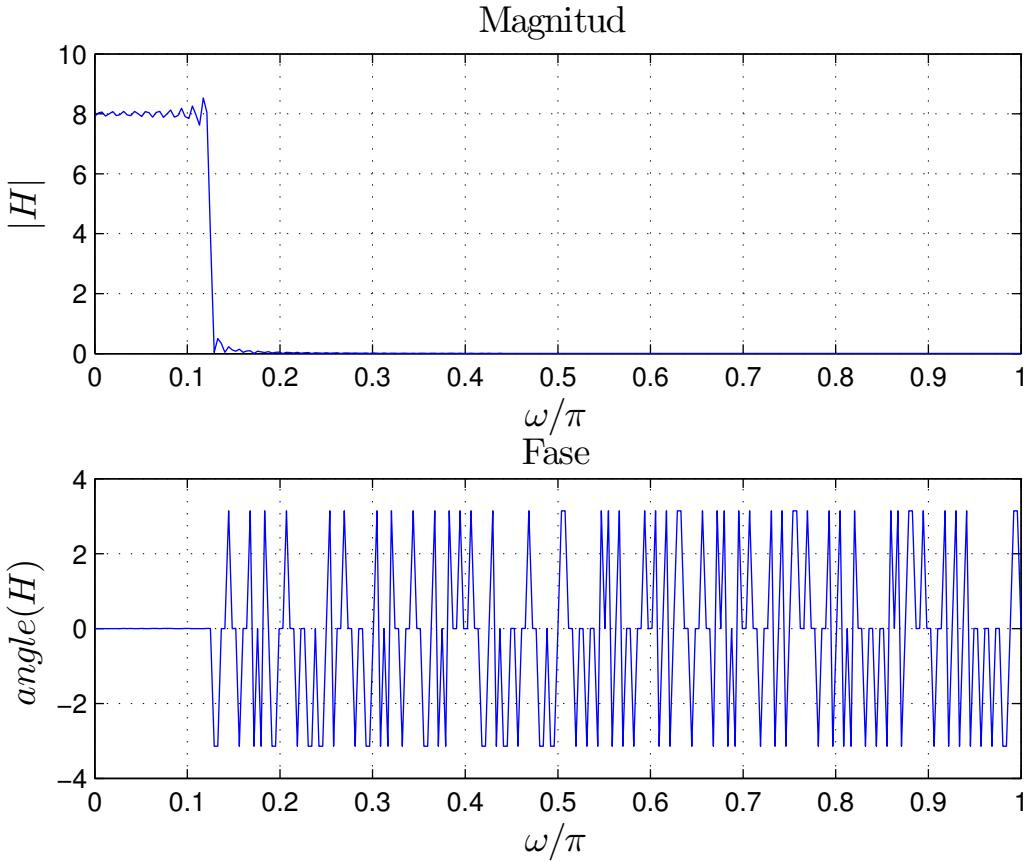


Figura 1.2: Respuesta en frecuencia (Magnitud y Fase) del filtro coseno idealizado, con  $\beta = 0$ .

Al elegir un valor de  $\beta = 0$ , la Ecuación (1.1) se transforma en una función seno cardinal, tal y como se puede observar en la Figura (1.1):

$$g(t) = \text{sinc}\left(\frac{t}{T}\right) \quad (1.2)$$

Con el valor de  $\beta = 0$ , se tiene un filtro pasa bajo ideal (Figura 1.2), con mínimo ancho de banda, de forma que la parte con caída cosenoidal desaparece.

En la Figura (1.2), se observan ciertas oscilaciones en forma de *picos* en la magnitud de la respuesta en frecuencia, las cuales se deben al fenómeno de Gibbs, debido a que cuando la función que se está desarrollando en serie de Fourier tiene discontinuidades (señales de variación rápida) no es posible que haya una buena convergencia en los entornos de las mismas. En tales entornos, las sumas parciales muestran tanto sobrevalores como subvalores alrededor del valor real de la función, que pueden llegar a un 18% del salto en la discontinuidad.

Respuestas al Impulso y en Frecuencia, para  $\beta = 0,2$  y  $M = 8$

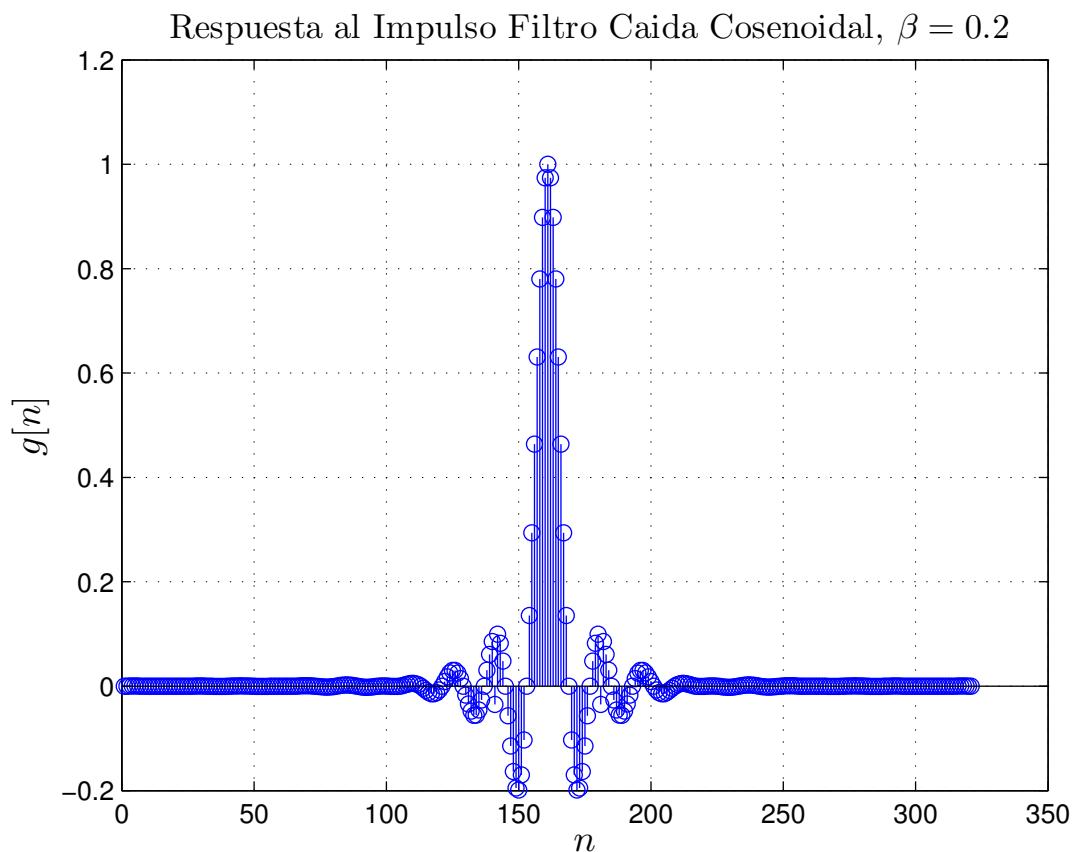


Figura 1.3: Respuesta al impulso del filtro coseno realizado, con  $\beta = 0,2$ .

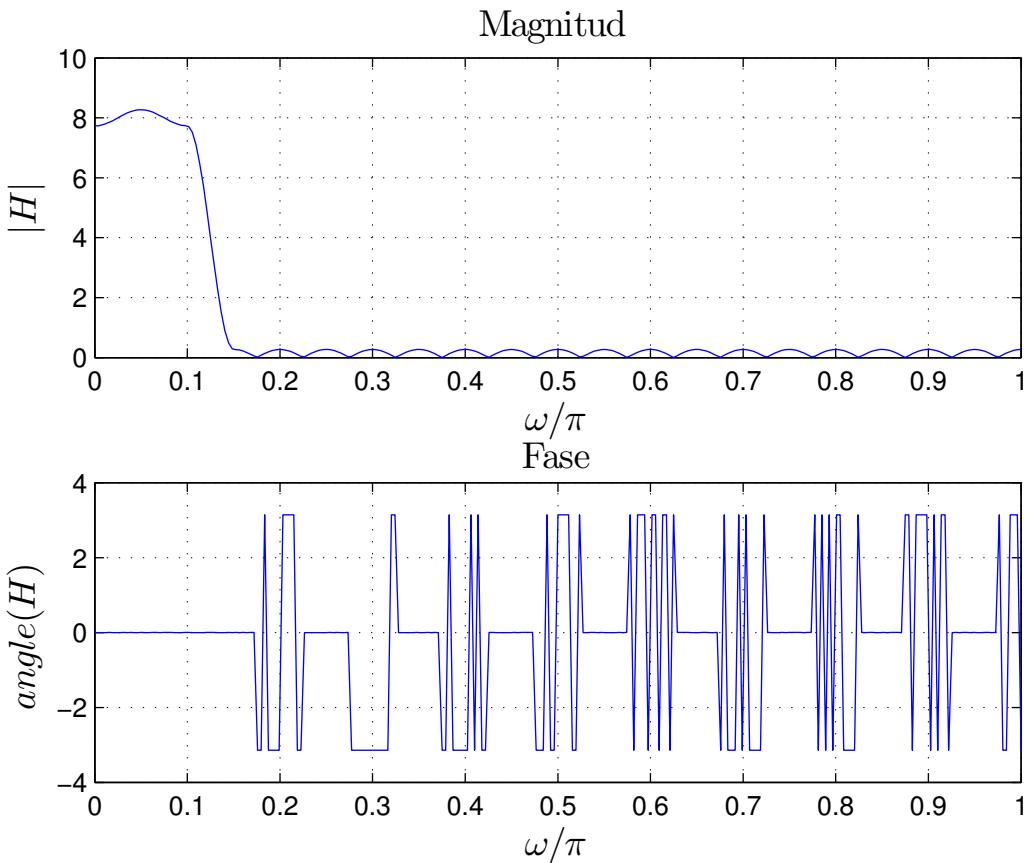


Figura 1.4: Respuesta en frecuencia (Magnitud y Fase) del filtro coseno realizado, con  $\beta = 0,2$ .

A medida que aumentamos el valor de  $\beta$ , la longitud de las colas de  $g[n]$ , comienza a disminuir y empieza a aparecer un exceso de ancho de banda en el dominio de la frecuencia, en comparación con el caso de un pulso pasa bajo ideal (Ecuación 1.2). Si bien este exceso de ancho de banda puede resultar a simple vista indeseado, en realidad, dota a este filtro de la capacidad de ser un filtro realizable, ya que el filtro pasa bajo ideal ó pulso seno cardinal, no es físicamente realizable.

En la Figura (1.3), se observa el comportamiento de las colas mencionado en el párrafo anterior. Y por otra parte, en la Figura (1.4), se puede ver el exceso de ancho de banda con respecto al pulso pasa bajo ideal (Figura 1.2).

Respuestas al Impulso y en Frecuencia, para  $\beta = 0,8$  y  $M = 8$

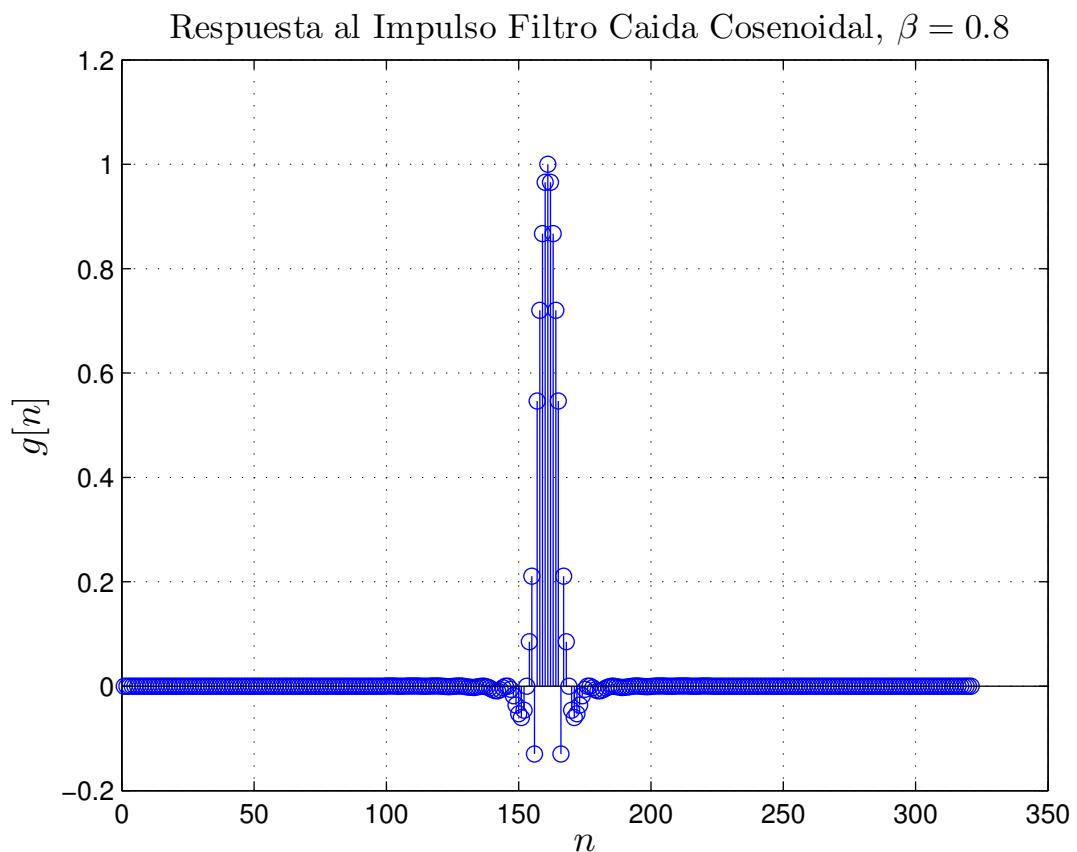


Figura 1.5: Respuesta al impulso del filtro coseno realizado, con  $\beta = 0,8$ .

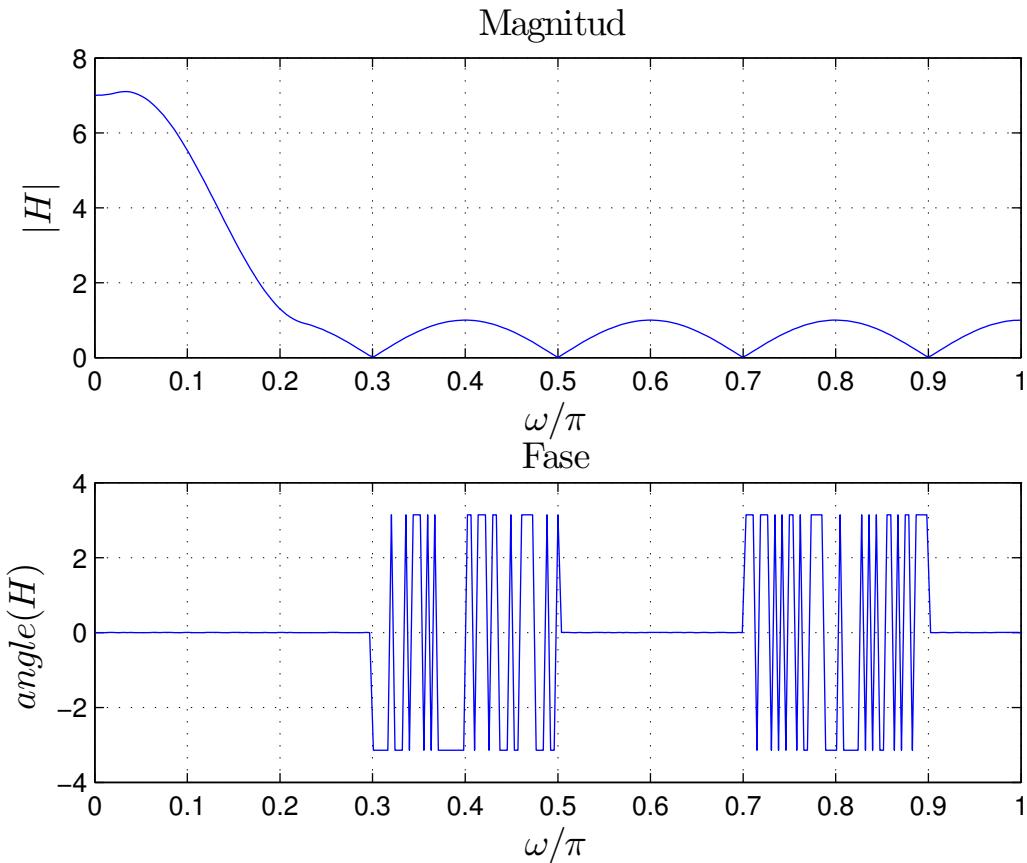


Figura 1.6: Respuesta en frecuencia (Magnitud y Fase) del filtro coseno realizado, con  $\beta = 0,8$ .

Nuevamente en las Figuras (1.5) y (1.6), se observa una menor longitud de las colas de  $g[n]$  y un mayor exceso de ancho de banda en la magnitud de su respuesta en frecuencia, debido a que en este caso el valor de  $\beta$  se ha aumentado nuevamente, llegando a ser  $\beta = 0,8$ . El tener un exceso de ancho de banda, significa que el pulso consumirá u ocupará una mayor porción del espectro, el cual recordemos, es un recurso valioso.

Respuestas al Impulso y en Frecuencia, para  $\beta = 1$  y  $M = 8$

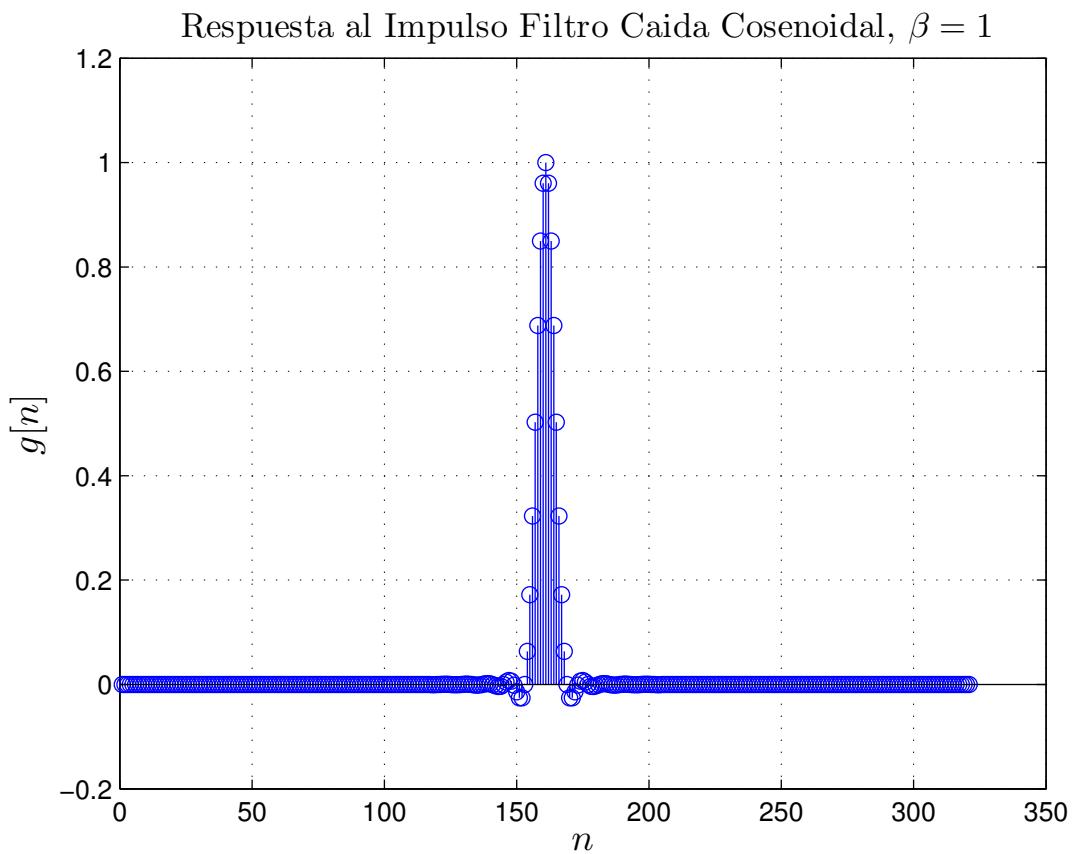


Figura 1.7: Respuesta al impulso del filtro coseno realizado, con  $\beta = 1$ .

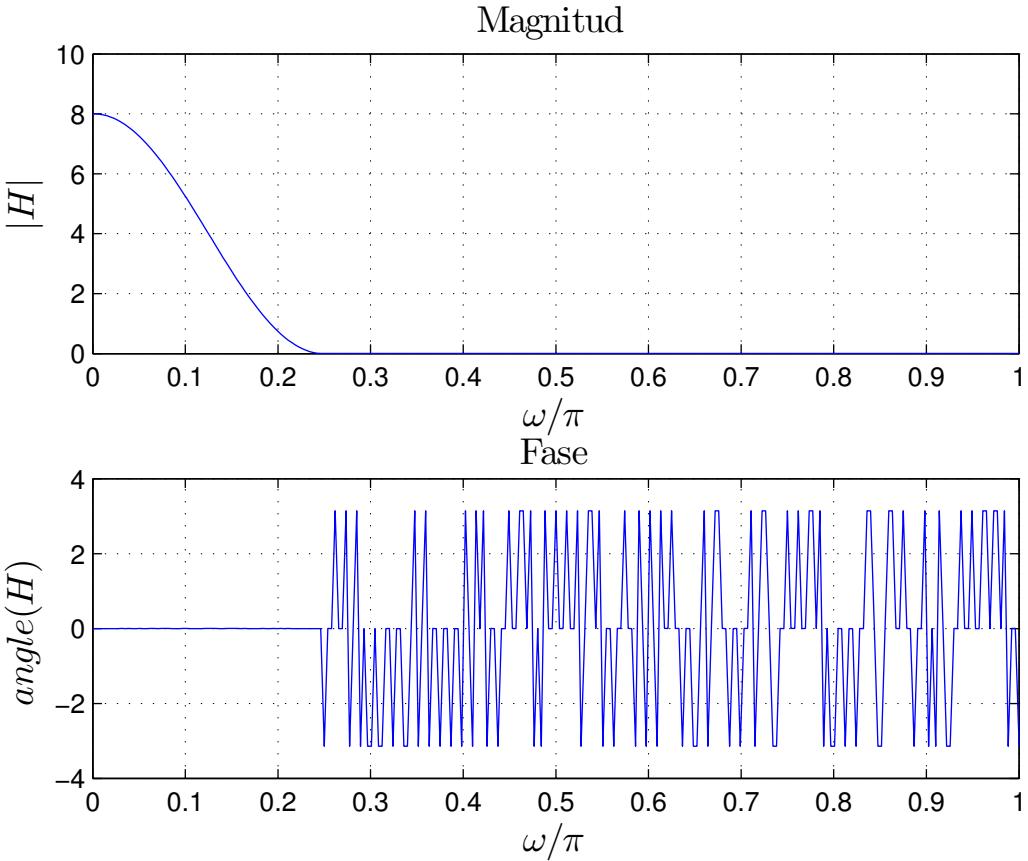


Figura 1.8: Respuesta en frecuencia (Magnitud y Fase) del filtro coseno realizado, con  $\beta = 1$ .

Finalmente, se observa el comportamiento obtenido para un valor de  $\beta = 1$  (Figuras 1.7 y 1.8), el cual justamente es el que le da origen a su nombre, ya que el espectro se asemeja a un coseno *alzado* o *levantado* por sobre el eje horizontal. Las colas antes presentes, casi desaparecen por completo y la ocupación del espectro es el doble que para el pulso con  $\beta = 0$ .

La importancia de la disminución de las colas es la de permitir que dicho pulso sea realizable, cuando analizamos el caso de  $\beta = 0$ , vimos que la duración del pulso era infinita hacia ambos lados, lo cual imposibilitaba su reproducción en el mundo físico real. En cambio, al disminuir las colas, los pulsos se vuelven físicamente realizable, aunque se debe pagar un precio por ello, dicho precio es el exceso de ancho de banda necesario para su transmisión.

Es muy normal encontrar en la teoría de comunicaciones, relaciones de *trade-off* entre un parámetro y otro. Por ende, todos los esfuerzos se centran en encontrar un balance entre dichos parámetros, en este caso, entre realizabilidad y ancho de banda necesario.

Símbolos Generados para Distintos Valores de Sobremuestreo

La generación de símbolos es independiente del valor de  $\beta$ , sin embargo la cantidad de ceros que se deben insertar entre cada símbolo, varía dependiendo del factor de sobremuestreo, por ejemplo: si se tiene un factor de sobremuestreo de  $M = 8$ , se deben insertar  $M - 1$  ceros entre cada nuevo símbolo. Luego el filtro coseno realizado será el encargado de interpolar (filtrado pasa bajos) los ceros insertados entre símbolos a sus valores correspondientes. A continuación, se ilustra la inserción de ceros:

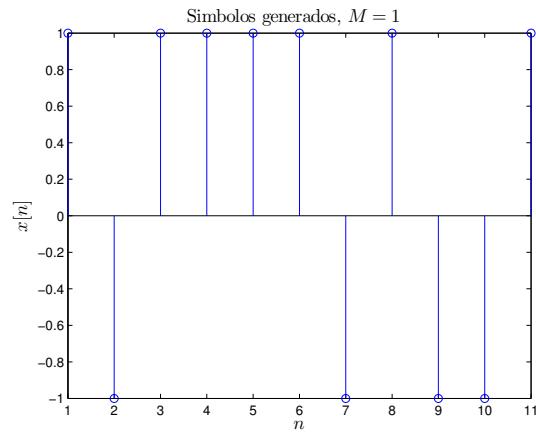


Figura 1.9: Símbolos generados con tasa de sobremuestreo,  $M = 1$  (Ningún cero insertado).

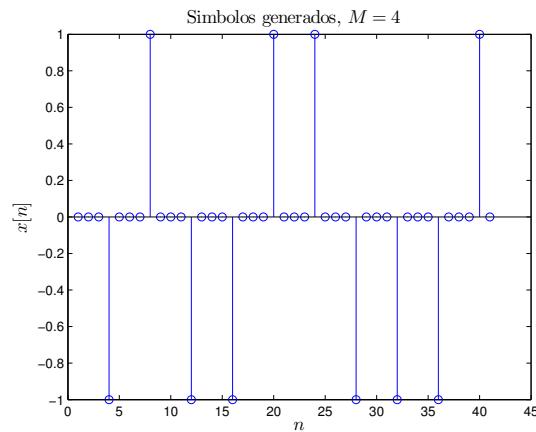


Figura 1.10: Símbolos generados con tasa de sobremuestreo,  $M = 4$  (3 ceros insertados).

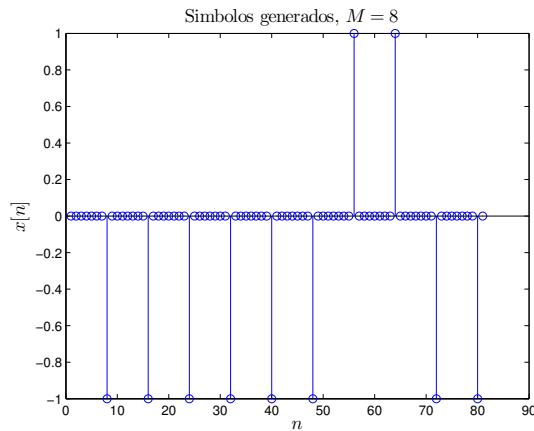


Figura 1.11: Símbolos generados con tasa de sobremuestreo,  $M = 8$  (7 ceros insertados).

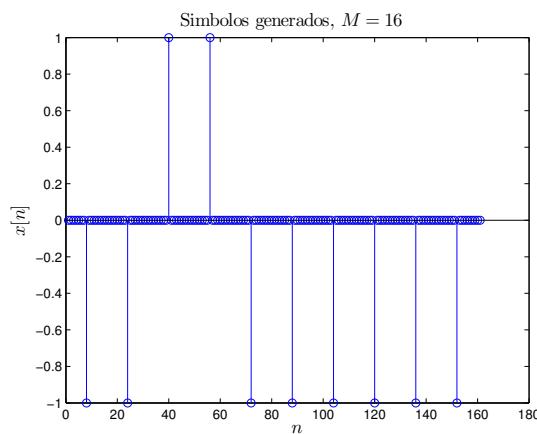


Figura 1.12: Símbolos generados con tasa de sobremuestreo,  $M = 16$  (15 ceros insertados).

Como se puede observar en las Figuras (1.9), (1.10), (1.11) y (1.12), a medida que se aumenta el factor de sobremuestreo  $M$ , la cantidad de ceros entre símbolos es mayor, más precisamente se tendrán  $M - 1$  ceros, entre cada símbolo aleatorio generado.

Señal Transmitida

La señal transmitida se obtiene convolucionando los símbolos aleatorios generados con la respuesta al impulso del filtro conformador de pulsos, que en nuestro caso se trata de un filtro coseno realzado. Ésta acción permite que los símbolos generados puedan ser transmitidos en un medio de comunicaciones en forma de pulsos de naturaleza analógica. Además, al usar el filtro coseno realzado como filtro conformador de pulsos, se logra que los sucesivos pulsos no interfieran unos con otros, ya que el mismo cumple con el criterio de Nyquist.

En caso de no cumplir con el criterio de Nyquist, cada pulso se vería interferido por los pulsos adyacentes, perdiendo inteligibilidad en el receptor y por lo tanto aumentando la tasa de errores del mismo.

A continuación, se muestran los símbolos transmitidos en dos vistas diferentes, para que el lector pueda apreciar mejor la forma de onda. *Se recuerda que la forma de onda corresponde a la señal de datos, convolucionada con la respuesta al impulso del filtro coseno realzado.*

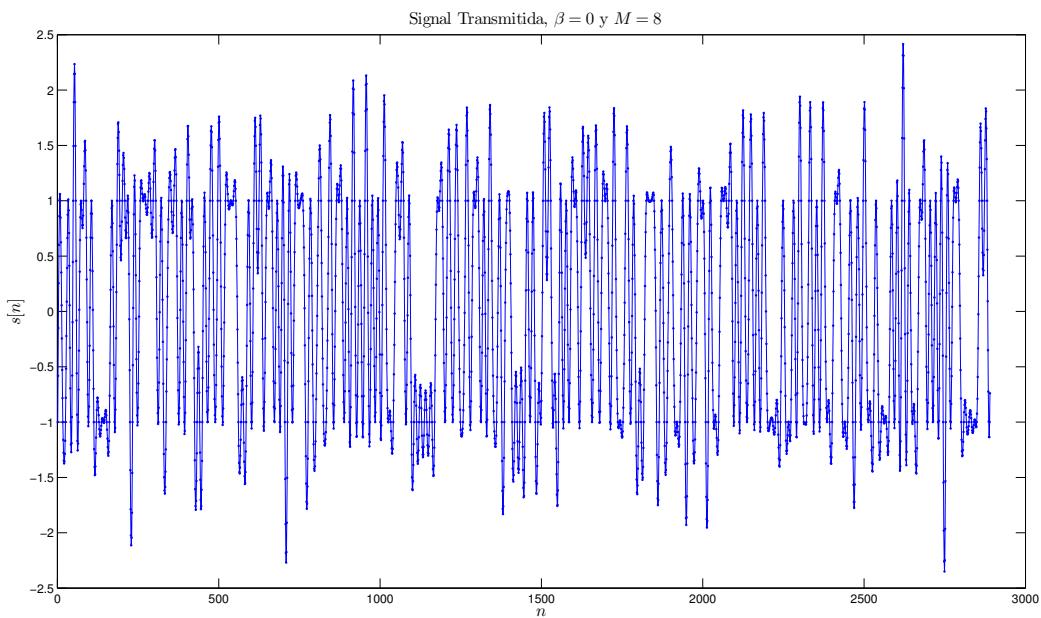


Figura 1.13: Señal transmitida,  $M = 8$  y  $\beta = 0$ .

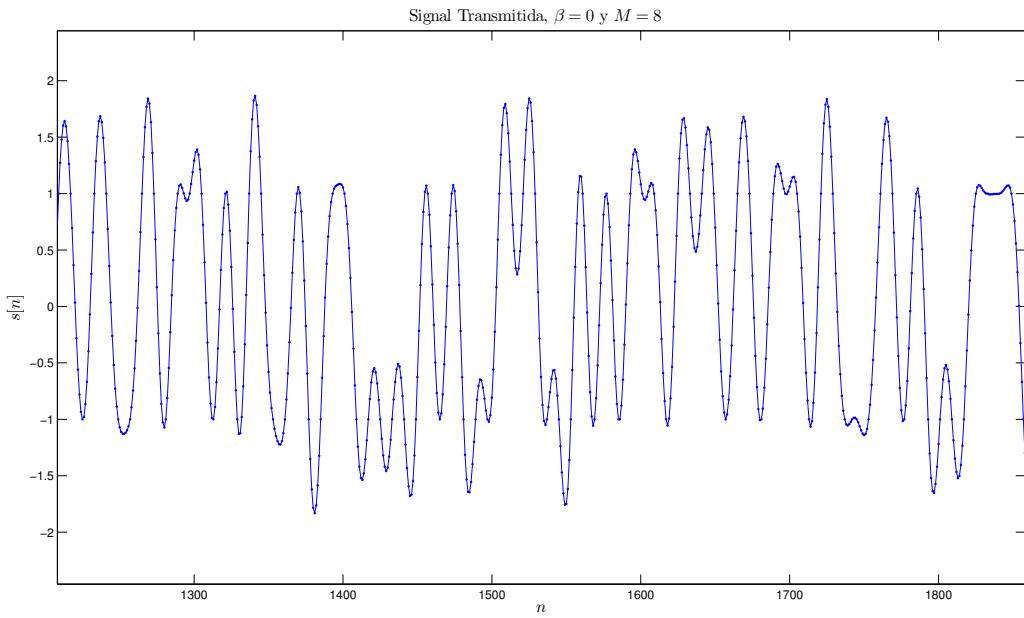


Figura 1.14: Señal transmitida (Zoom),  $M = 8$  y  $\beta = 0$ .

En este primer caso, en el cual se escogió un valor de  $\beta = 0$  (pulso seno cardinal), se puede observar una señal con un cierto grado de *sobrepicos* en torno a  $+1$  y *subpicos* en torno a  $-1$ . Aunque no exista ISI por tratarse de un pulso de Nyquist, se puede concluir que el hecho de usar un mínimo ancho de banda genera cierto grado de *sobrepasamiento* en la forma de onda, lo cual afectará luego en el diagrama de ojo (Figuras 1.13 y 1.14).

A medida que aumentamos el valor de  $\beta$ , se aumenta el exceso de ancho de banda (mayor consumo de recursos) y disminuyen las oscilaciones por sobre  $+1$  y por debajo de  $-1$ . Además, el filtro en este caso se trata de un filtro realizable (Figuras 1.15, 1.16, 1.17 y 1.18), en donde se ha producido una disminución de las colas del pulso conformador.

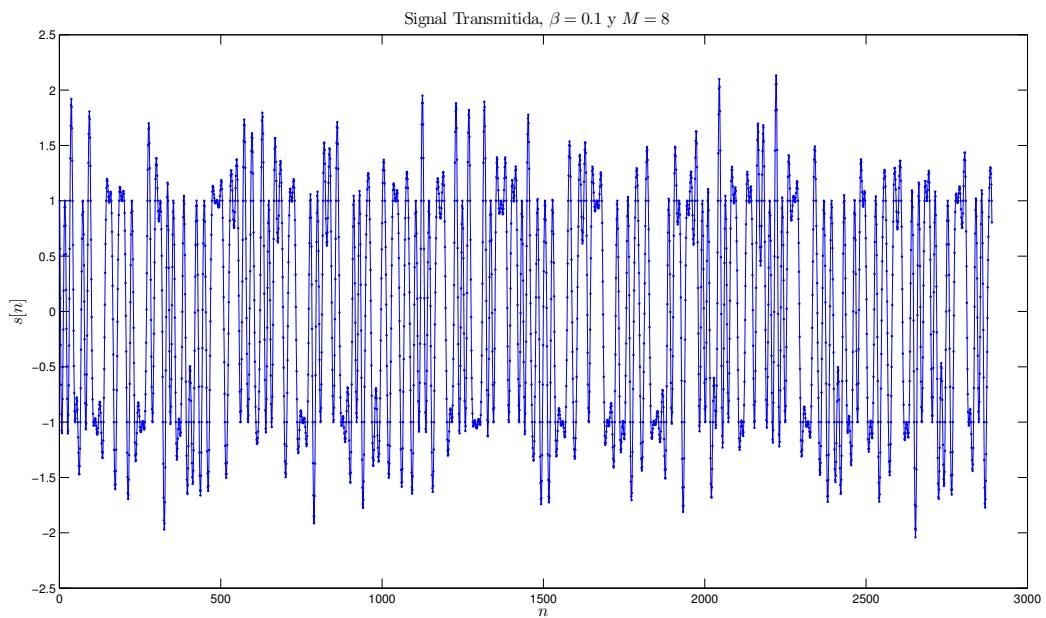


Figura 1.15: Señal transmitida,  $M = 8$  y  $\beta = 0,1$ .

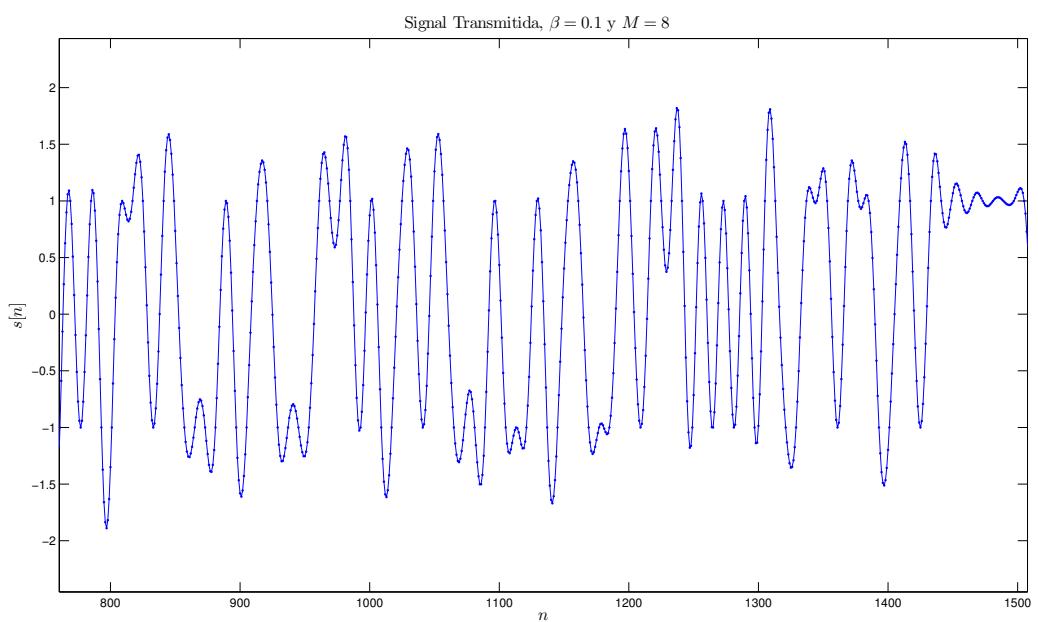


Figura 1.16: Señal transmitida (Zoom),  $M = 8$  y  $\beta = 0,1$ .

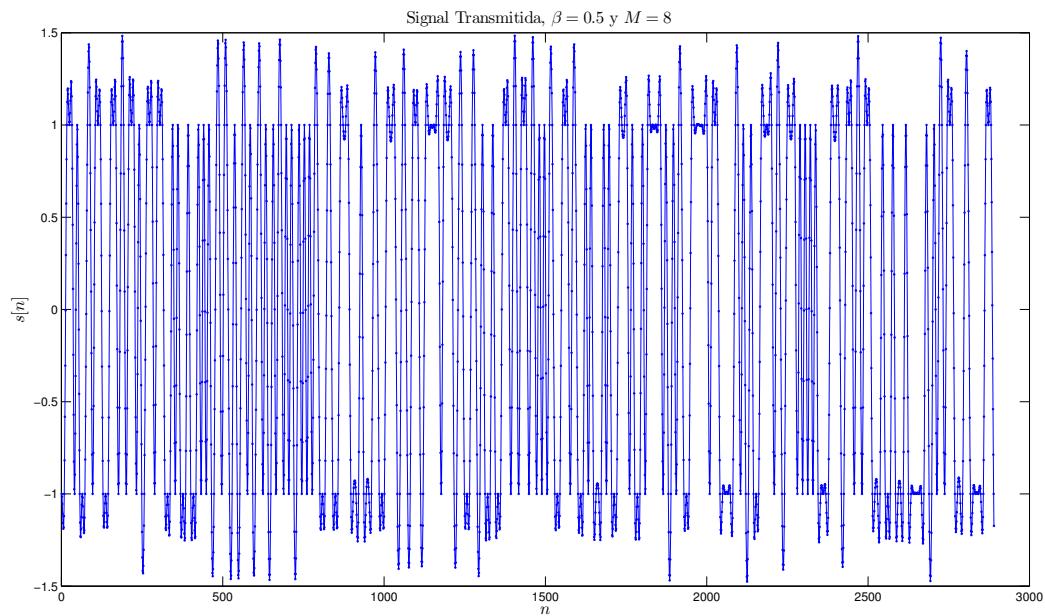


Figura 1.17: Señal transmitida,  $M = 8$  y  $\beta = 0,5$ .

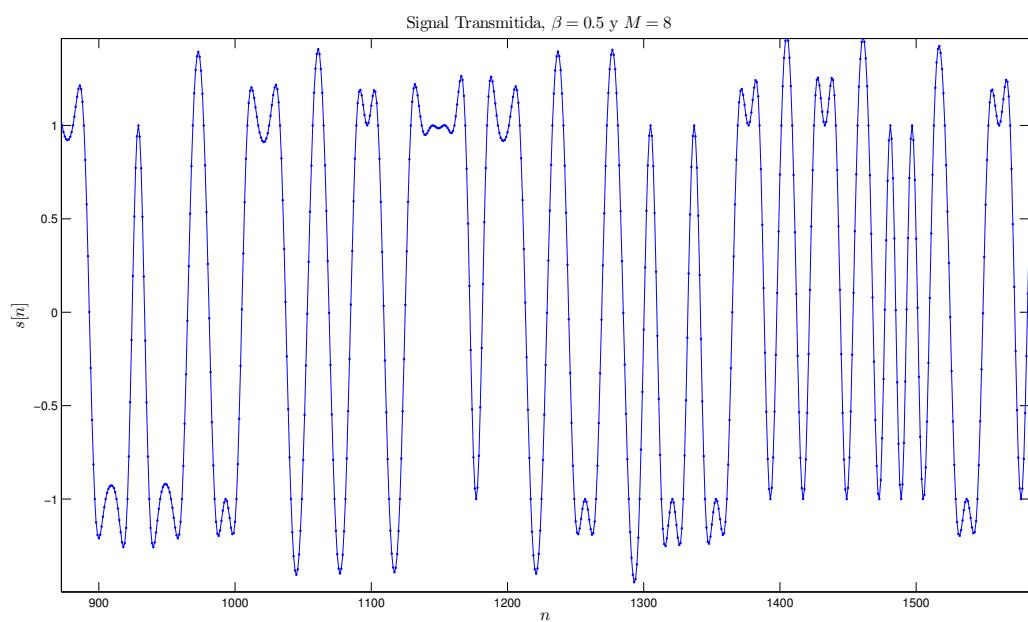


Figura 1.18: Señal transmitida (Zoom),  $M = 8$  y  $\beta = 0,5$ .

Finalmente, en las Figuras (1.19) y (1.20), se puede observar que las oscilaciones por sobre  $+1$  y por debajo de  $-1$ , se han reducido considerablemente, debido a que el ancho de banda del filtro se ha duplicado con respecto al filtro pasa bajo ideal.

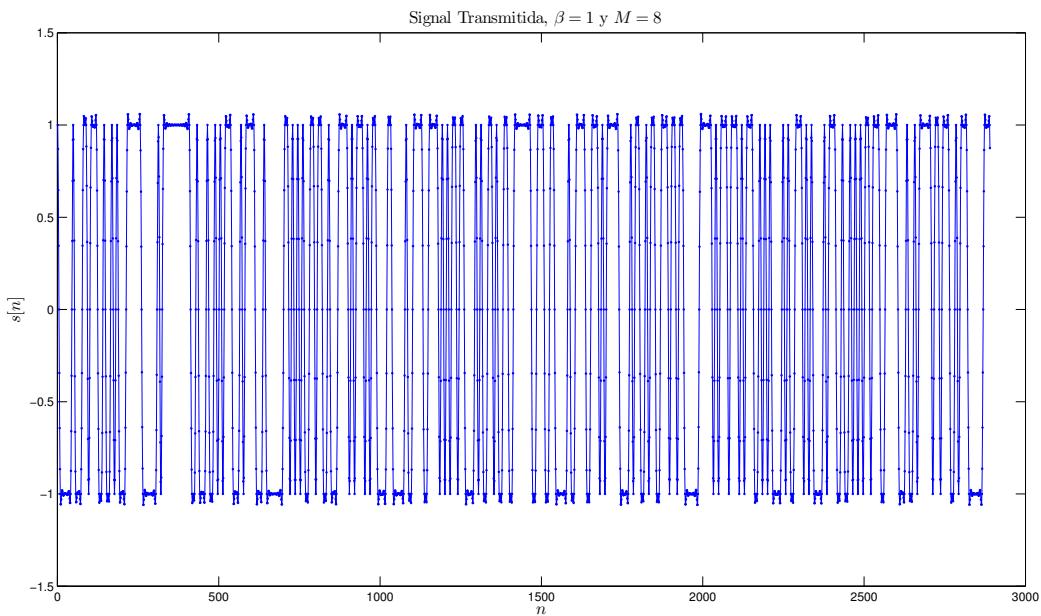


Figura 1.19: Señal transmitida,  $M = 8$  y  $\beta = 1$ .

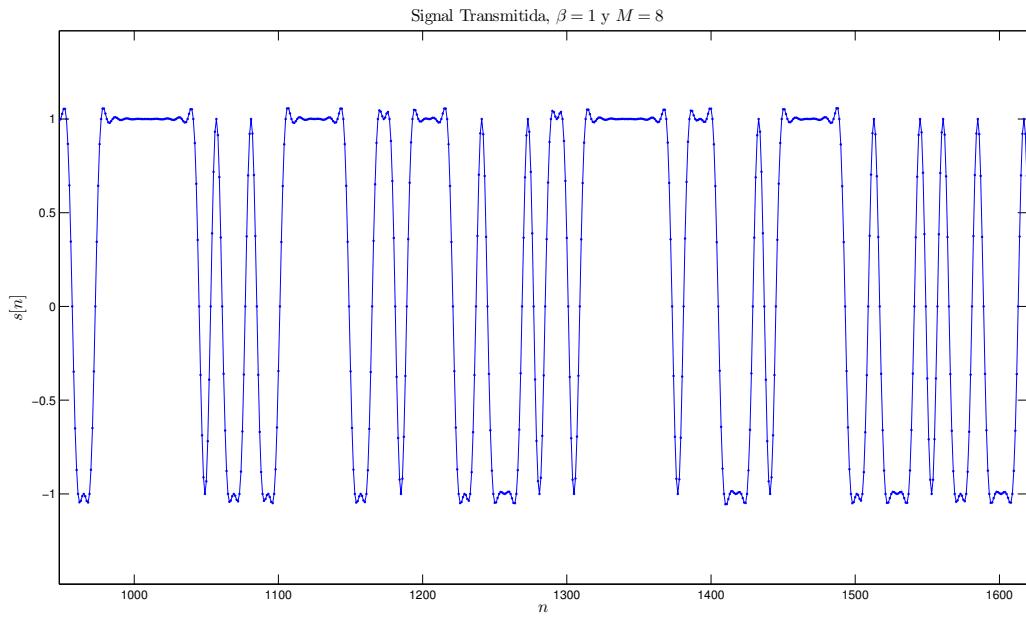


Figura 1.20: Señal transmitida (Zoom),  $M = 8$  y  $\beta = 1$ .

En las aplicaciones prácticas o de la industria, se elige un valor de  $\beta$  cercano a 0, ya que un valor elevado significa mayor ancho de banda necesario (mayor costo). Nuevamente sale a la luz la relación de compromiso latente, entre la realizabilidad de un pulso conformador que no produzca ISI y el consumo de recursos, como en este caso del ancho de banda.

### Diagramas de Ojo

El diagrama de ojo o patrón de ojo es un método utilizado para el análisis del comportamiento de los enlaces de transmisión y su nombre proviene de la similitud de dichos diagramas, con la forma de un ojo. El diagrama de ojo permite analizar las formas de onda de los pulsos que se propagan en un enlace de comunicaciones, para lograr observar sus formas, desfases, niveles de ruido, potencias de las señales, etc. y con ello apreciar la distorsión del canal (ISI), la severidad del ruido o interferencia y los errores de sincronismo en el receptor.

El diagrama de ojo corresponde a un gráfico que muestra la superposición de las distintas combinaciones posibles de unos y ceros en un rango de tiempo o cantidad de bits determinados. Dichas combinaciones transmitidas por el enlace, permiten obtener las características de los pulsos que se propagan por el medio de comunicación, sean estos por medio de fibra óptica, coaxial, par trenzado, enlaces satelitales, etc. El gráfico se forma superponiendo los trazos de la salida del filtro receptor en un osciloscopio (en nuestro caso dichos diagramas se

obtienen por simulación utilizando los modelos matemáticos pertinentes).

Por ejemplo, en una secuencia de 3 bits hay 8 combinaciones posibles, las que pueden ser observadas en la Figura (1.21).

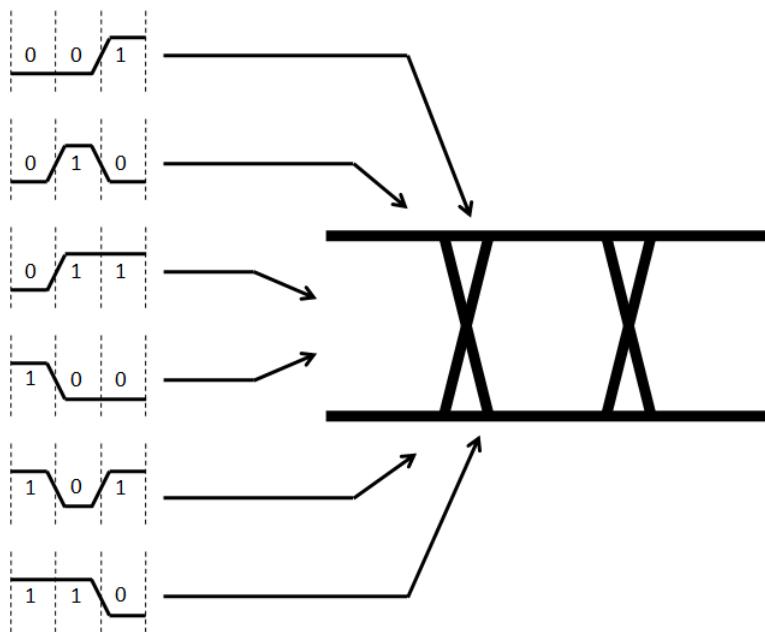


Figura 1.21: Ejemplo de Diagrama de Ojo.

A continuación, se muestran los Diagramas de Ojo obtenidos por simulación, para distintos valores de *roll-off*.

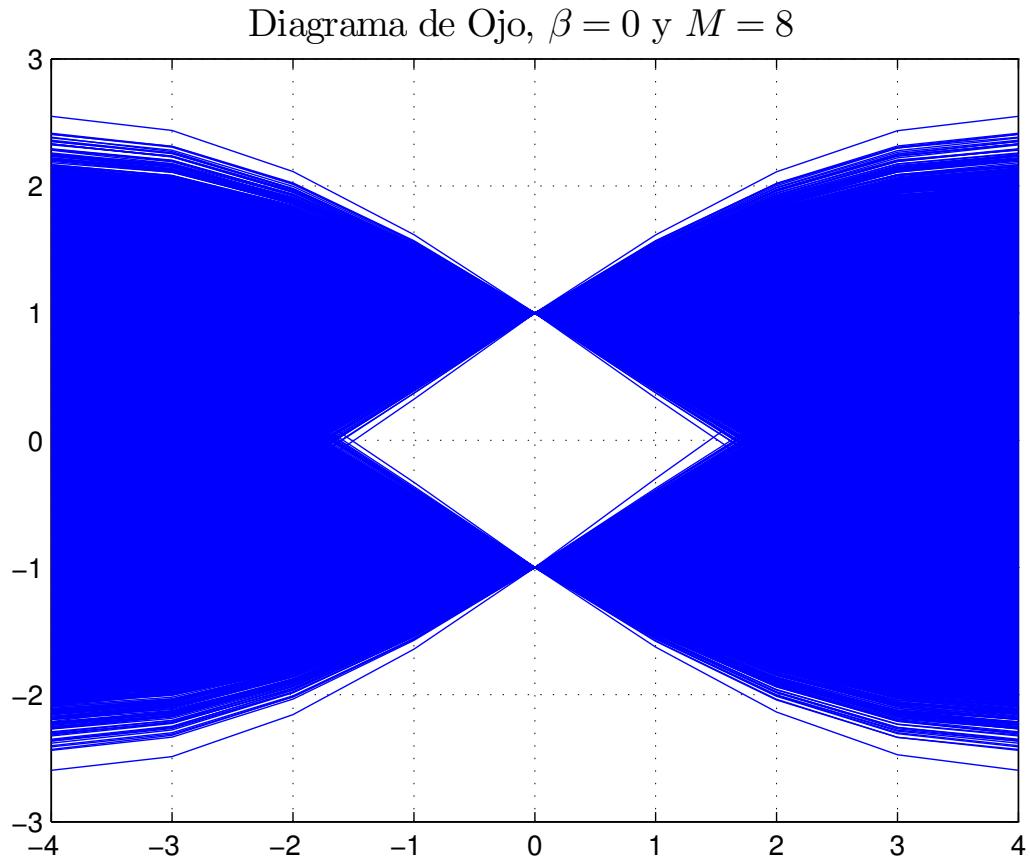


Figura 1.22: Diagrama de Ojo,  $M = 8$  y  $\beta = 0$ .

En la Figura (1.22), se puede que observar que no existe ISI, ya que en el instante de muestreo 0, sólo hay influencia del pulso actual y no de sus vecinos. Sin embargo, cualquier desviación en el instante de muestreo (instante 0) puede generar problemas ya que estaríamos muestreando el pulso deseado y algunos de sus vecinos. Este comportamiento se puede observar en la apertura horizontal de ojo, la cual en este caso, no es muy grande.

El hecho de tener una apertura horizontal pobre, genera problemas en las aplicaciones reales (por presencia de jitter) de comunicaciones donde el muestreo no es ideal y el mismo presenta pequeñas variaciones/desviaciones del instante ideal, lo que produce que el muestreo se realice en zonas donde el pulso actual es interferido por sus vecinos.

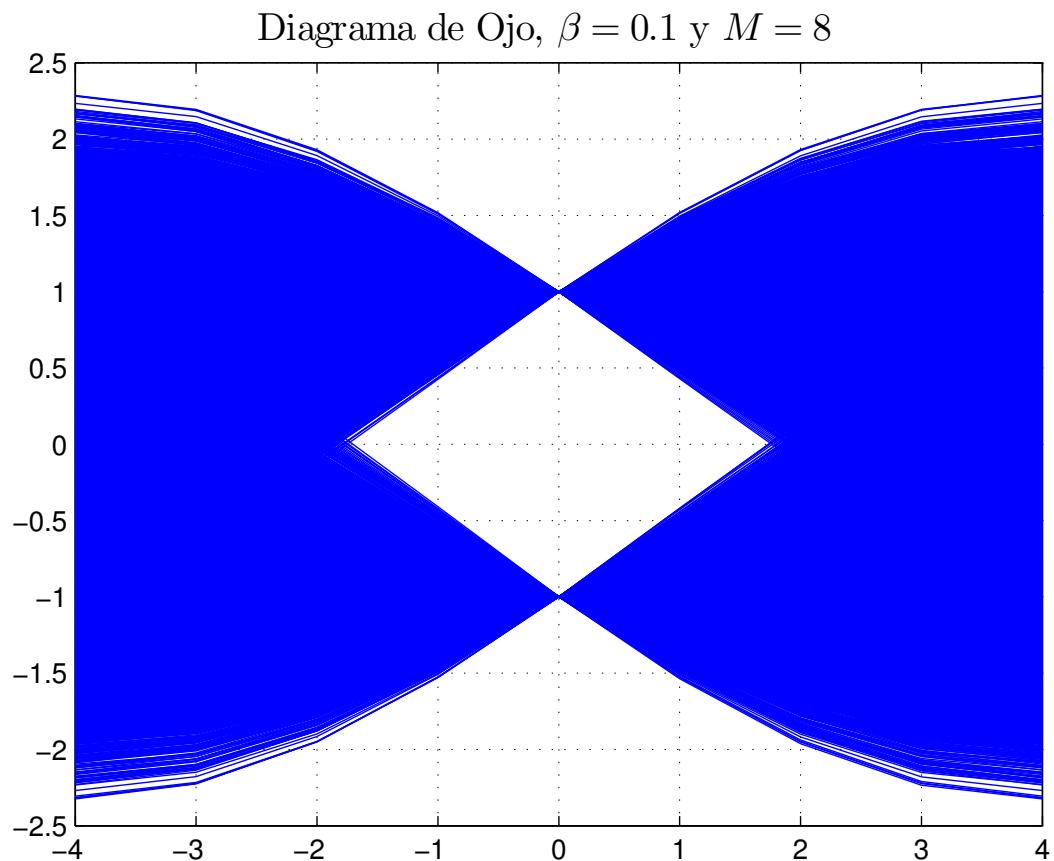


Figura 1.23: Diagrama de Ojo,  $M = 8$  y  $\beta = 0,1$ .

A medida que aumentamos el valor de  $\beta$ , como en la Figura (1.23), la apertura horizontal mejora debido a una menor fluctuación en el cruce por cero, a costa de un mayor ancho de banda. Al igual que en el caso anterior al tratarse de un pulso de Nyquist, no hay ISI.

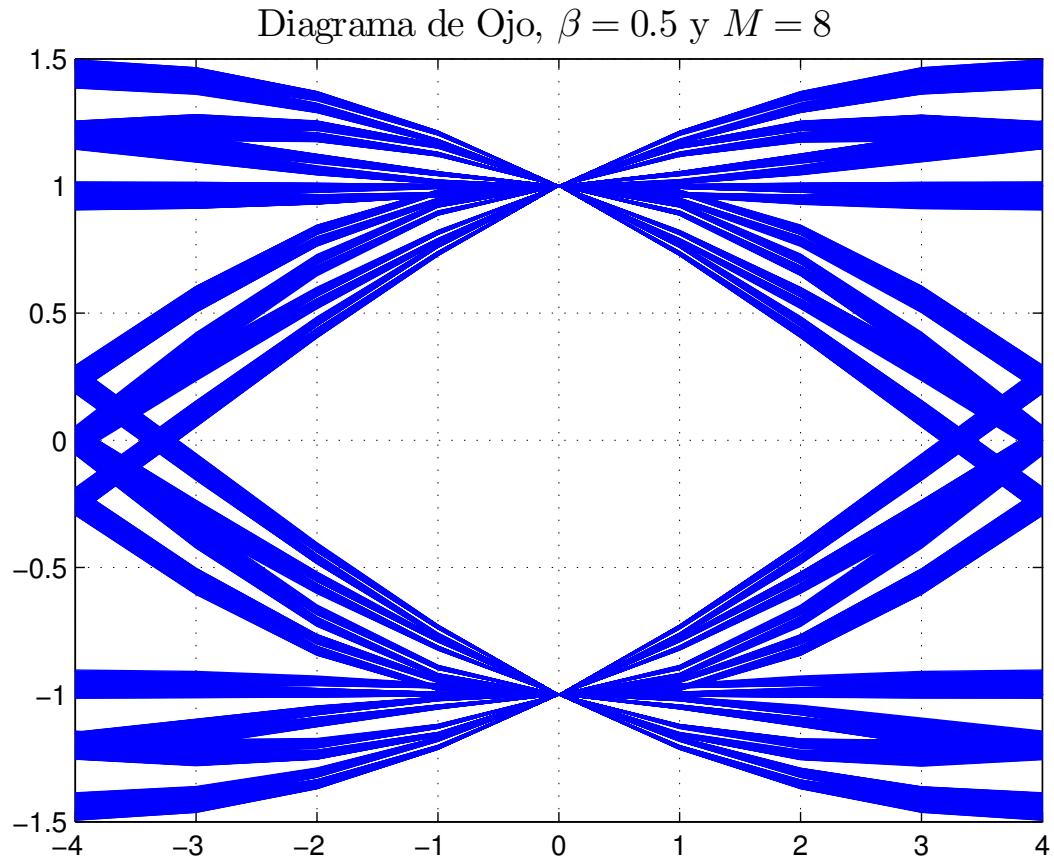


Figura 1.24: Diagrama de Ojo,  $M = 8$  y  $\beta = 0,5$ .

Nuevamente, en la Figura (1.24) se ve un aumento de la apertura horizontal del ojo, ya que el valor de  $\beta = 0,5$ . Esta mayor apertura se traduce en un aumento en el exceso de ancho de banda del filtro (mayor costo), sin embargo permite transiciones más abruptas en los pulsos.

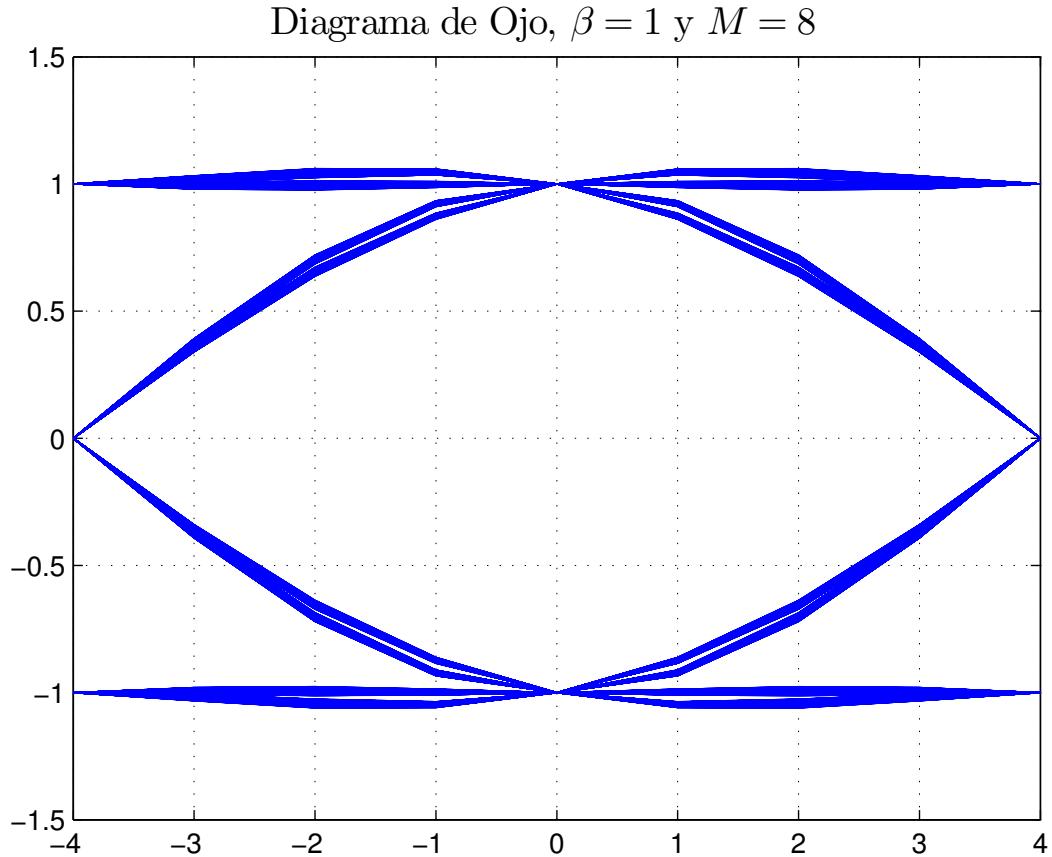


Figura 1.25: Diagrama de Ojo,  $M = 8$  y  $\beta = 1$ .

Finalmente, en la Figura (1.25), el exceso de ancho de banda se duplicó con respecto al caso ideal (Ecuación 1.2) y se logró una máxima apertura horizontal del ojo, eliminando el problema del jitter. Además, como en todos los casos anteriores no hay ISI, por tratarse de un pulso de Nyquist.

Se debe tener en cuenta que para todos los valores de  $\beta$  analizados, no existe la interferencia entre símbolos, ya que todos los casos corresponden a pulsos de Nyquist. Sin embargo, como vimos, el exceso de ancho de banda no es el mismo y aumenta a medida que  $\beta$  se acerca a 1. Además, el factor de roll-off tiene un efecto directo sobre la apertura horizontal del ojo, ya que la misma aumenta a medida que aumenta  $\beta$ , lo que a su vez mejora los escenarios donde el instante de muestreo no es ideal y existen pequeñas variaciones.

Variando el Factor de Sobremuestreo,  $M$

En ésta subsección, se escogió un valor de  $\beta = 0,2$ , el cual se mantuvo constante para todos los casos y se fue variando el factor de sobremuestreo  $M$ .

A continuación, se muestran los mismos resultados que se habían mostrado en la subsección anterior, buscando observar el efecto de variar el factor de sobremuestreo para un mismo valor de roll-off.

El hecho de modificar el factor de sobremuestreo trae un cambio importante en la inserción de ceros en la generación de símbolos (tal y como se vió anteriormente). A continuación, se muestran los símbolos generados para  $M = 4$  y  $M = 16$  (Figuras 1.26 y 1.27).

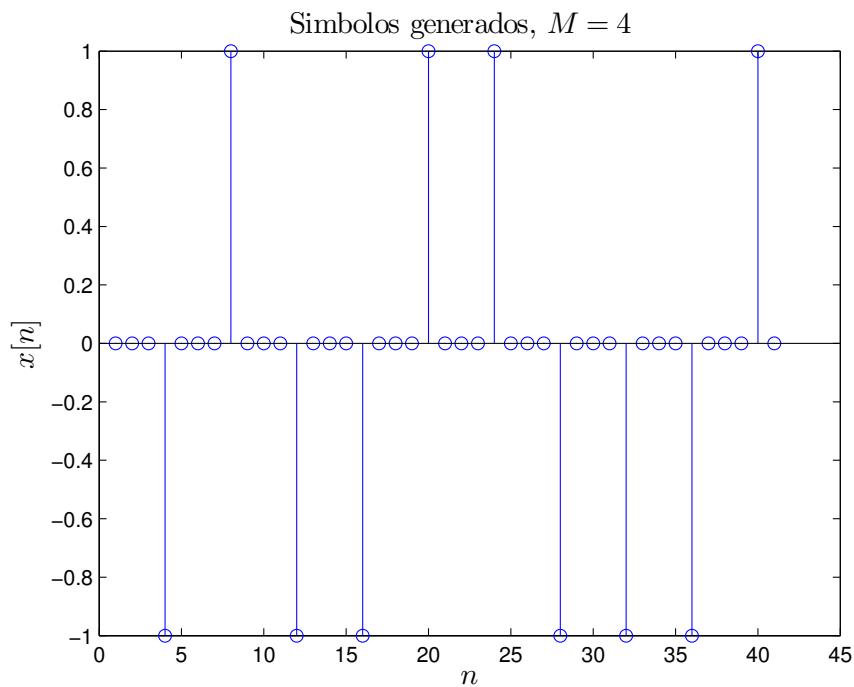


Figura 1.26: Símbolos generados con tasa de sobremuestreo  $M = 4$  y  $\beta = 0,2$ .

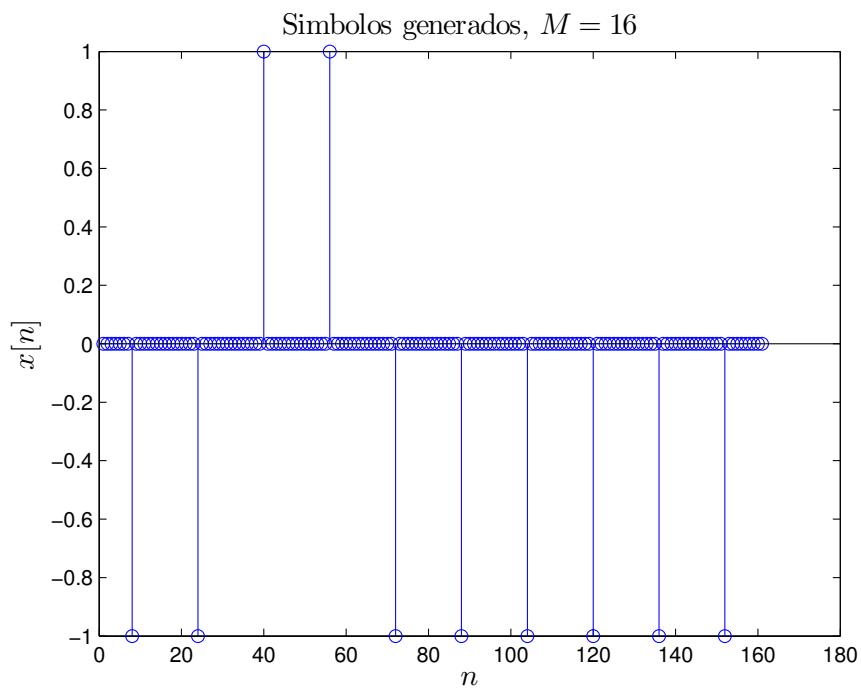


Figura 1.27: Símbolos generados, con tasa de sobremuestreo  $M = 16$  y  $\beta = 0,2$ .

Al aumentar el factor de sobremuestreo, se logra tener mayor cantidad de muestras intermedias entre dos símbolos, tal y como se puede observar en la Figura (1.29), con respecto a la Figura (1.28).

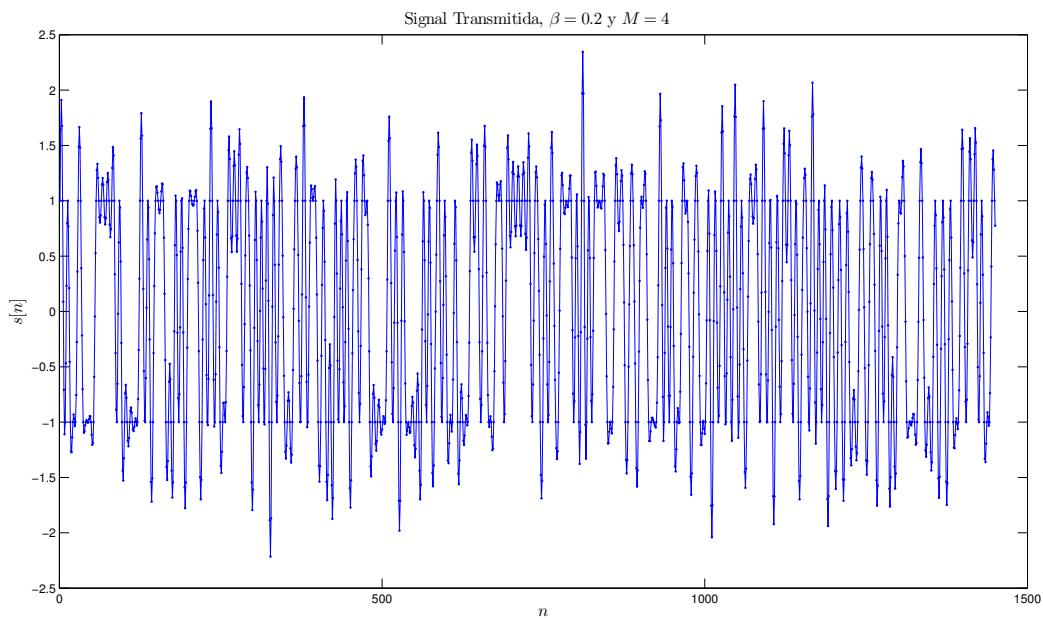


Figura 1.28: Señal transmitida, con tasa de sobremuestreo  $M = 4$  y  $\beta = 0,2$ .

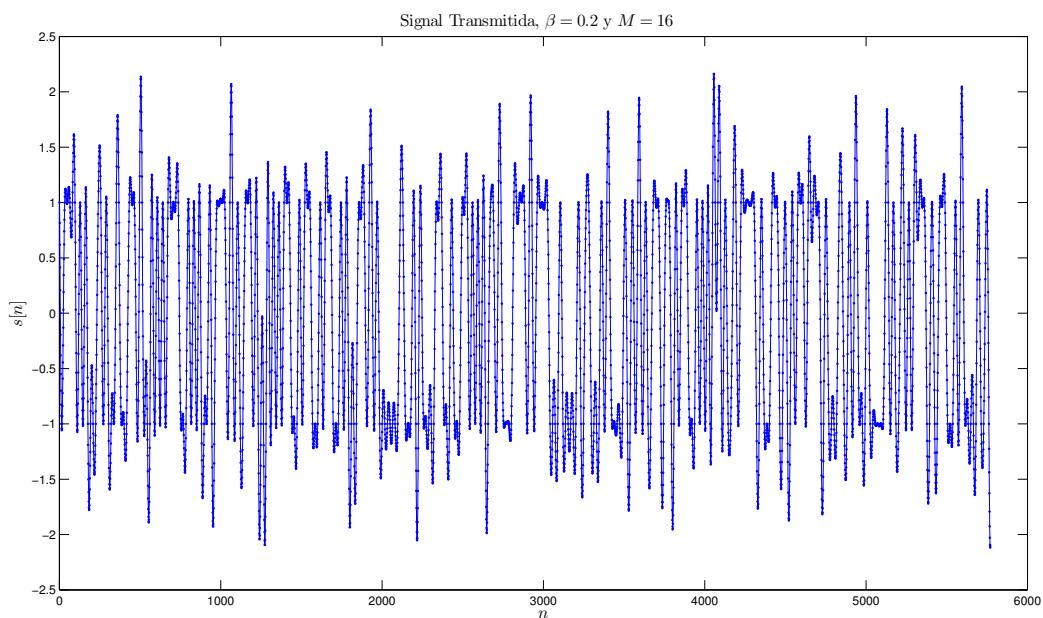


Figura 1.29: Señal transmitida, con tasa de sobremuestreo  $M = 16$  y  $\beta = 0,2$ .

Finalmente, en las Figuras (1.30) y (1.31), se observan los diagramas de ojo obtenidos para los dos valores de sobremuestreo ( $M = 4$  y  $M = 16$ ). El hecho de aumentar la tasa de sobremuestreo permite tener una mayor cantidad de muestras entre símbolos, lo cual es bueno para la recuperación de sincronismo. Sin embargo, no se puede aumentar sin límite la tasa de sobremuestreo, ya que esto se traduce en la necesidad de un reloj de trabajo de mayor tasa, lo que trae aparejados complicaciones físicas y mayores costos de implementación.

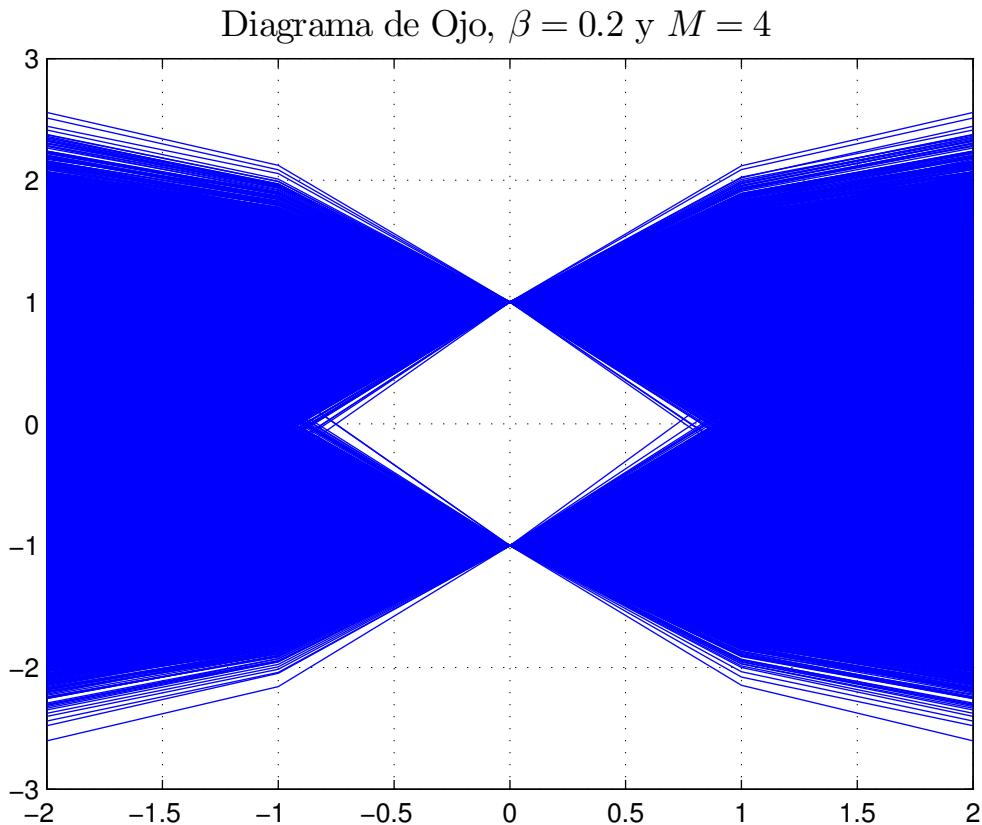


Figura 1.30: Diagrama de Ojo,  $M = 4$  y  $\beta = 0,2$ .

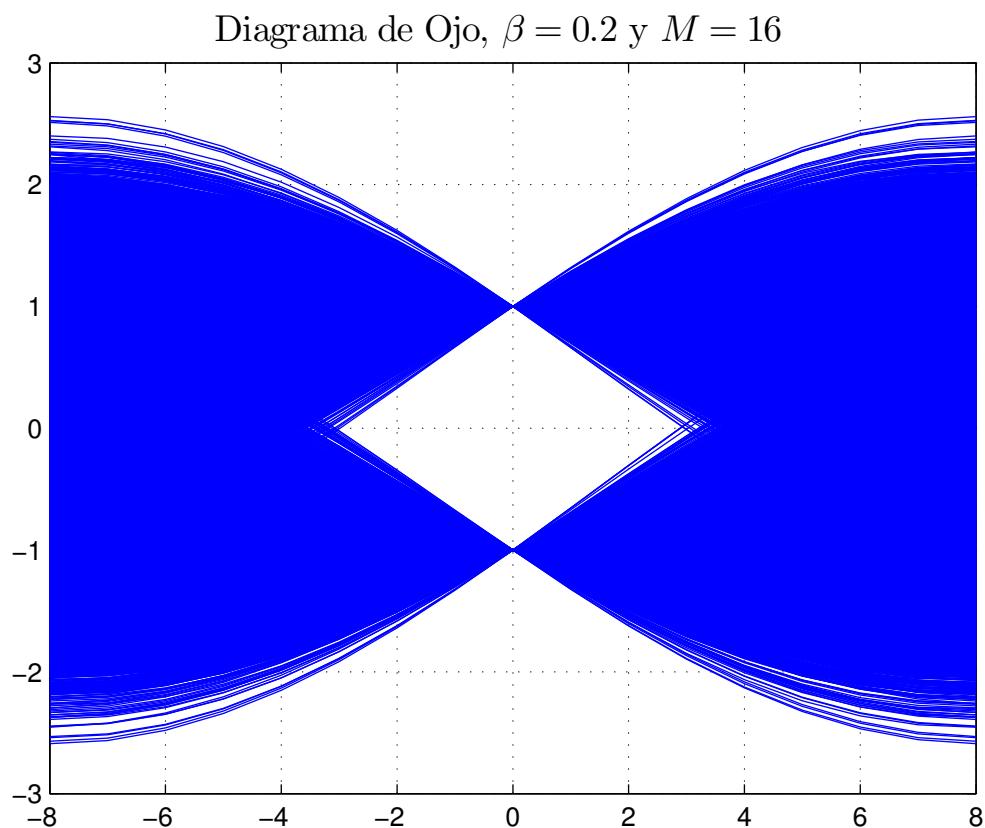


Figura 1.31: Diagrama de Ojo,  $M = 16$  y  $\beta = 0,2$ .

# Capítulo 2

## Transformador de Hilbert

### 2.1. Descripción y Consignas

Otro filtro muy utilizado en comunicaciones digitales (así como el filtro coseno realizado tratado en el capítulo anterior), es el filtro Transformador de Hilbert, cuya ecuación es la siguiente:

$$f[n] = \frac{2\sin^2(\pi n/2)}{\pi n}, \quad n \neq 0; \quad f[0] = 0 \quad (2.1)$$

Se propone desarrollar las siguientes actividades:

1. Utilizar la autofunción  $e^{j\Omega n}$ , para obtener por simulación la respuesta  $|F(e^{j\Omega})|$  y  $\angle F(e^{j\Omega})$ .
2. Sea  $x[n] = \sum_{k=0}^{L-1} a[k]\delta[n - kM]$  con  $a[k] \in \{\pm 1 \pm j\}$  y  $g[n]$  la respuesta al impulso del filtro de caída cosenoidal visto en el capítulo anterior. Seleccionar un valor de  $\Omega_0$ , para que la señal  $s[n] = e^{j\Omega_0 n} (x[n] * g[n])$ , sea analítica ( $S(e^{j\Omega}) = 0$ , para  $-\pi < \Omega < 0$ ).
3. Verificar por simulación (utilizando herramientas como el cálculo de la Densidad Espectral de Potencia ó PSD) que:

$$\Im\{s[n]\} = \Re\{s[n]\} * f[n]$$

4. Implementar en MATLAB un simulador del sistema de comunicaciones digitales elemental de la Figura (2.1).

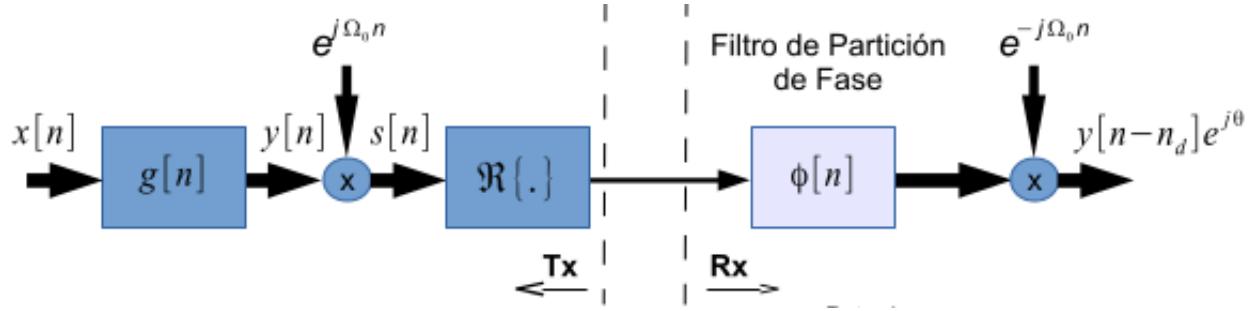


Figura 2.1: Sistema de Comunicaciones Digitales Elemental.

5. Verificar que:

$$E\{| p[n] |^2\} = \sum_k | g[n - kM] |^2 \quad (2.2)$$

6. Verificar que  $E\{| p[n] |^2\}$  es una secuencia real periódica, con periodo M y serie de Fourier dada por:

$$E\{| p[n] |^2\} = \sum_{q=<M>} b_q e^{jq\frac{2\pi}{M}n} \quad (2.3)$$

Con  $b_q$  dado por:

$$b_q = \frac{1}{M2\pi} \int_{2\pi} G(e^{j\Omega}) G^*(e^{j(\Omega - \frac{q2\pi}{M})}) d\Omega \quad (2.4)$$

7. Verificar la existencia de un tono senoidal periódico con período M, en la señal  $E\{| p[n] |^2\}$ , si  $\beta < 1$ .
8. Verificar la existencia de un tono senoidal periódico con período M, en una segunda señal transmitida por un canal independiente, donde dicha señal posee un retardo  $n_d = M/2$ .
9. Verifique que si sumamos la señal  $p[n]$  y la segunda señal  $\hat{p}[n]$  (con retardo), el tono de sincronismo desaparece por cancelación.

## 2.2. Resultados

En ésta sección, se presentan los resultados obtenidos por simulación y además, en los momentos que se crea oportuno, se mencionan aspectos teóricos que ayudan a la compresión de dichos resultados. De la misma manera, se agregan comentarios extraídos de la observación de cada una de las simulaciones.

Un transformador de Hilbert es un filtro que introduce un desplazamiento de fase de  $\pi/2$  en la señal de entrada. Su respuesta en frecuencia ideal está dada por la siguiente ecuación,

$$F(e^{j\Omega}) = \begin{cases} +j & \text{si } -\pi < \Omega < 0 \\ -j & \text{si } 0 < \Omega < \pi \end{cases} \quad (2.5)$$

El Transformador de Hilbert es útil para calcular el contenido en frecuencia de una señal de energía o de potencia. Así se pueden analizar y diseñar filtros selectivos en frecuencia para poder separar señales según su contenido espectral.

Otro criterio para separar señales es el que está basado en la selectividad en fase ó discriminación de fase, que defasa las señales pertinentes de modo que se puedan separar fácilmente. En particular el Transformador de Hilbert, realiza un defasaje de  $\pm 90$  grados.

### Respuesta al Impulso y en Frecuencia del Transformador de Hilbert

En la Figura (2.2), se pueden observar las respuestas al impulso de los dos filtros analizados hasta el momento. En la subfigura superior, se observa el filtro de caída cosenoidal ( $\beta = 0,5$ ) y en la subfigura inferior, el filtro que será analizado en el presente capítulo, el Transformador de Hilbert. A simple vista podemos notar las diferencias entre una respuesta y la otra.

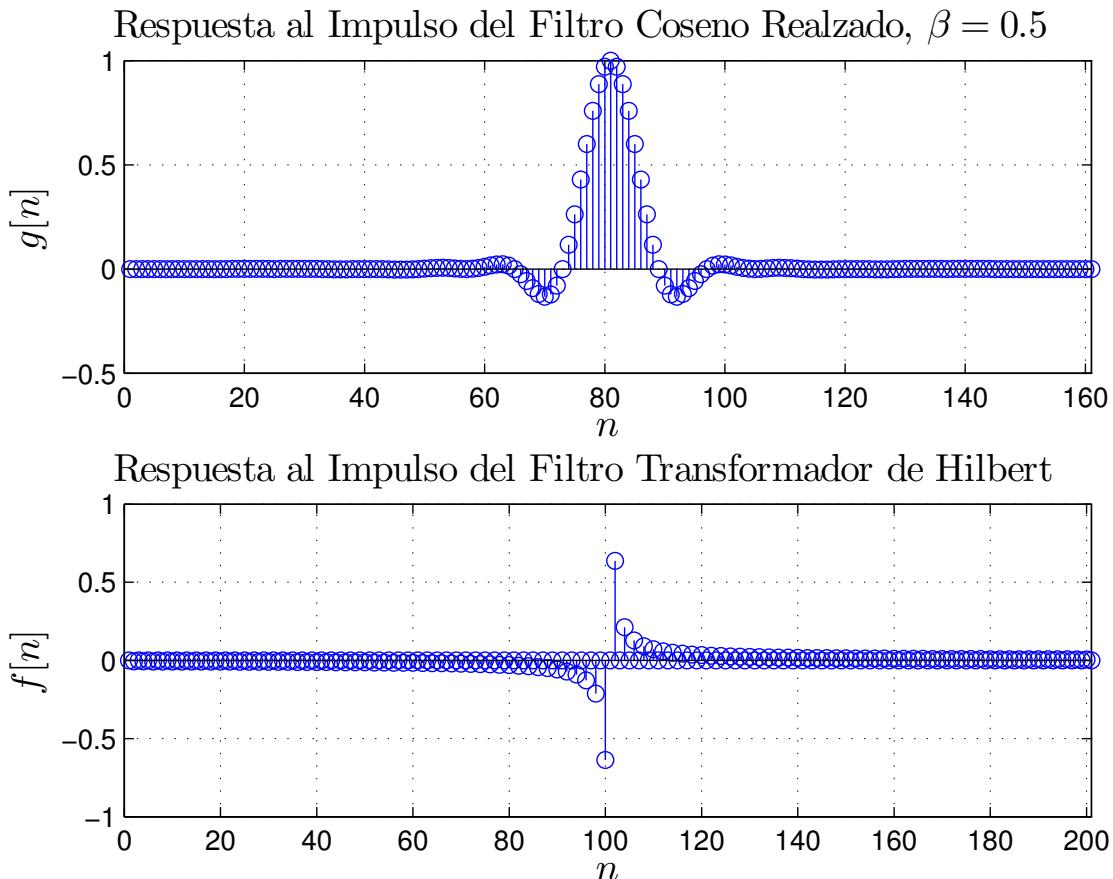


Figura 2.2: Respuesta al impulso del filtro coseno realizado (arriba) y del filtro transformador de Hilbert (abajo).

Una vez presentada la respuesta al impulso del filtro Transformador de Hilbert, se procede a calcular su respuesta en frecuencia. Para ello, se hace uso de la autofunción  $e^{j\Omega n}$ , tal y como se hizo previamente con el filtro de caída cosenoidal. El hecho de que el sistema es un sistema LIT (Lineal e Invariante con el Tiempo), nos permite hallar la respuesta en frecuencia, la cual es un autovalor de la autofunción usada como señal de entrada.

En la Figura (2.3), se muestra la magnitud y la fase de la respuesta en frecuencia del Transformador de Hilbert, obtenido mediante simulación, gracias a la autofunción  $e^{j\Omega n}$  usada como señal de entrada del sistema.

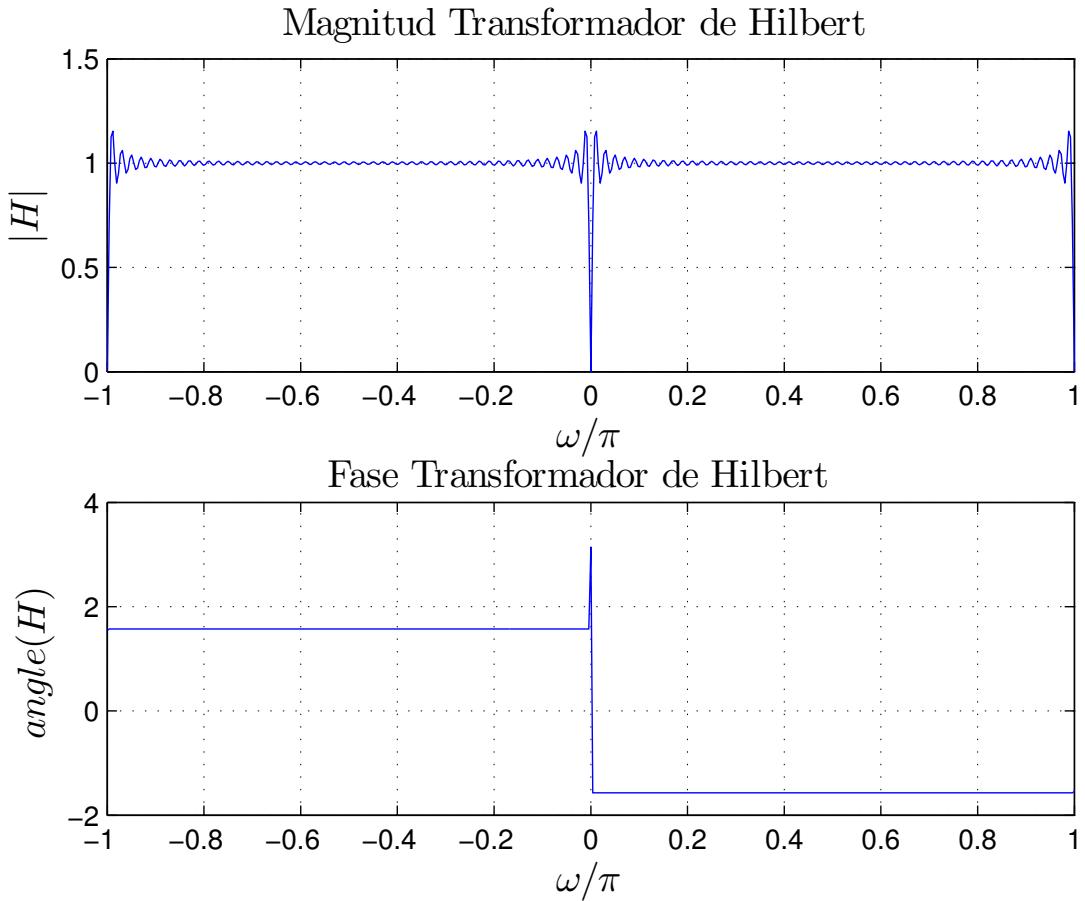


Figura 2.3: Respuesta en frecuencia (magnitud y fase) del filtro Transformador de Hilbert.

Como se puede observar en la Figura (2.3), la magnitud es plana (salvo por las oscilaciones debido al fenómeno de Gibbs) y la fase alterna entre  $+90$  y  $-90$  grados para  $-\pi < \Omega < 0$  y  $0 < \Omega < \pi$  respectivamente, por lo tanto el transformador de Hilbert cae dentro de la clasificación de sistema pasa todo, ya que posee respuesta en magnitud plana para todo el espectro, sin embargo, la fase de la respuesta en frecuencia es *no lineal*.

### Generación de Símbolos 4-QAM

Seguidamente, se generan los símbolos a transmitir. En el capítulo anterior, los símbolos eran reales, en cambio, en este capítulo, los símbolos que se generaron fueron complejos (correspondiente a un esquema de modulación 4-QAM). En la Figura (2.4), se pueden observar los símbolos complejos generados. Donde sus valores corresponden a  $\pm 1 \pm j$ .

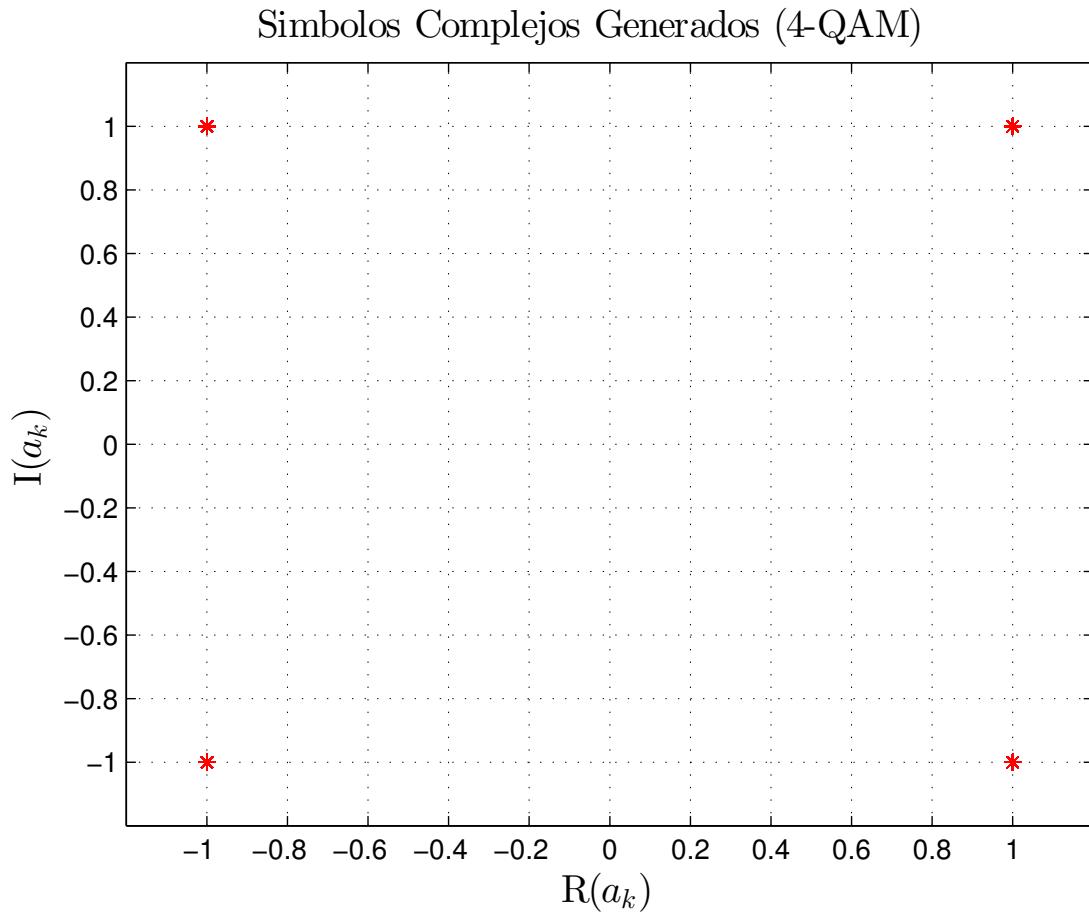


Figura 2.4: Símbolos complejos 4-QAM.

El hecho de utilizar esquemas de modulación cada vez mayores (con una mayor cantidad de bits por símbolo) habilita a un mejor aprovechamiento del ancho de banda disponible, ya que cada símbolo representa un segmento de bits de información y no un sólo bit. Para poner un simple ejemplo; supongamos que deseamos enviar una cadena de bits, donde cada bit está representado por un símbolo, en el lado del receptor, cada vez que se reciba un símbolo, se traducirá en un sólo bit. Por el contrario, supongamos que ahora se representa la misma cadena de bits, pero agrupada en segmentos de subcadenas de 4 bits; entonces en el lado del receptor, cada símbolo recibido corresponderá a 4 bits de información enviada. Esto permite un mejor aprovechamiento del ancho de banda disponible.

## Señales Analíticas

Antes de continuar presentando los resultados obtenidos, se realiza una breve mención de las señales analíticas. Esto nos va a permitir entender los resultados que se presentan a continuación.

Una señal compleja  $s[n] = \Re\{s[n]\} + j\Im\{s[n]\}$ , se dice analítica si su transformada de Fourier satisface:

$$S(e^{j\Omega}) = 0, \quad -\pi < \Omega < 0 \quad (2.6)$$

Aquellas señales no analíticas de ancho de banda limitado, se pueden transformar en una señal analítica, mediante una simple traslación en frecuencia. Por el contrario las señales de ancho de banda ilimitado ó pasa todo, donde existe contenido espectral de importancia en todas las frecuencias, no pueden volverse analíticas debido a su propia definición, ya que por más que sean desplazadas en frecuencia, siempre habrá contenido en las frecuencias negativas.

A continuación, la señal  $y[n] = x[n] * g[n]$ , (donde  $g[n]$  es el filtro de caída cosenoidal) se la convierte en una señal analítica  $s[n]$ , para ello se realiza un desplazamiento de su espectro, dado por  $s[n] = e^{j\Omega_0 n} y[n]$  ó  $s[n] = e^{j\Omega_0 n}(x[n] * g[n])$ . En las Figuras (2.5) y (2.6), se muestra la Densidad Espectral de Potencia ó PSD de la señal  $y[n]$  en banda base (no analítica) y de la nueva señal analítica  $s[n]$ , respectivamente.

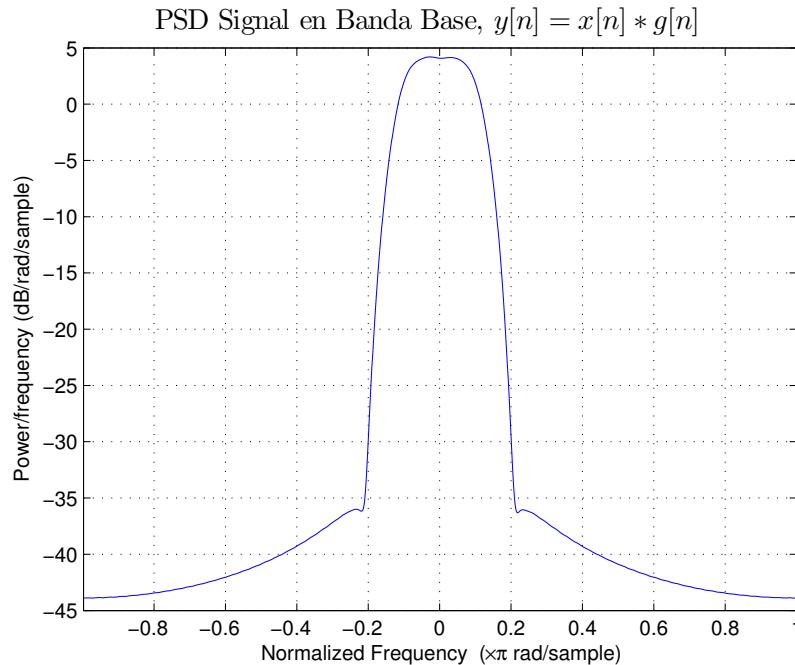
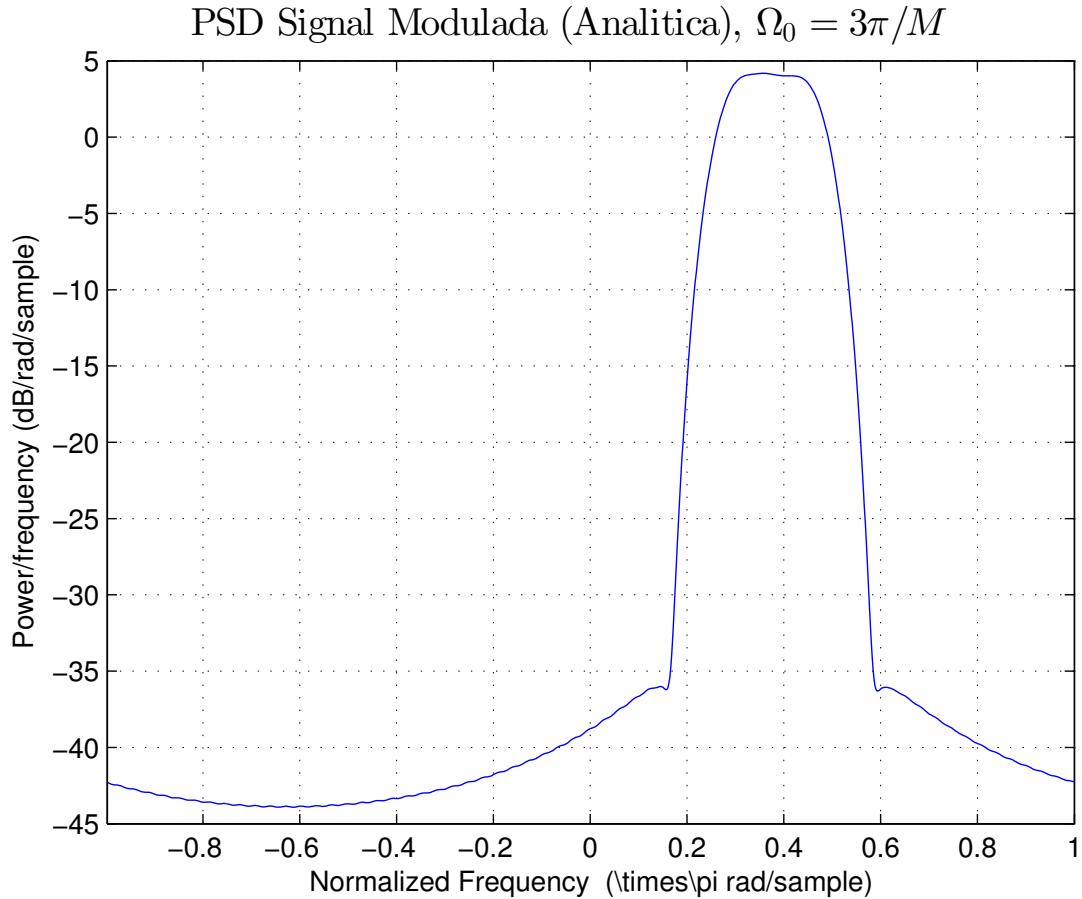


Figura 2.5: PSD de la señal en banda base  $y[n]$  (No analítica).

Figura 2.6: PSD de la señal analítica  $s[n]$ .

Para poder volver analítica la señal  $y[n]$ , se debió escoger un valor de  $\Omega_0$  tal que el desplazamiento en frecuencia produzca que todo el espectro entre  $-\pi < \Omega < 0$  sea 0. El valor escogido fue  $\Omega_0 = 3\pi/M$ , donde  $M = 8$  es el factor de sobremuestreo utilizado con  $g[n]$ .

### Filtro de Partición de Fase

Como se vió durante las clases, el filtro de partición de fase permite recuperar la parte Imaginaria de una señal original compleja, a partir de su parte Real. Esto trae muchos beneficios para las comunicaciones digitales, ya que permite recuperar únicamente toda la información original, a partir de la recepción de una porción de ella, reduciendo de esta manera los costos del sistema.

Expresando estas ideas matemáticamente, se tiene que el filtro de partición de fase permite obtener:

$$\Im\{s[n]\} = \Re\{s[n]\} * f[n]$$

Para poder demostrar por simulación la expresión anterior, se hizo uso del cálculo de la Densidad Espectral de Potencia (PSD) en las diferentes etapas del simulador (utilizando las herramientas de MATLAB). En primer lugar (Figura 2.7) se muestra la PSD de la señal original en banda base  $y[n] = x[n] * g[n]$ .

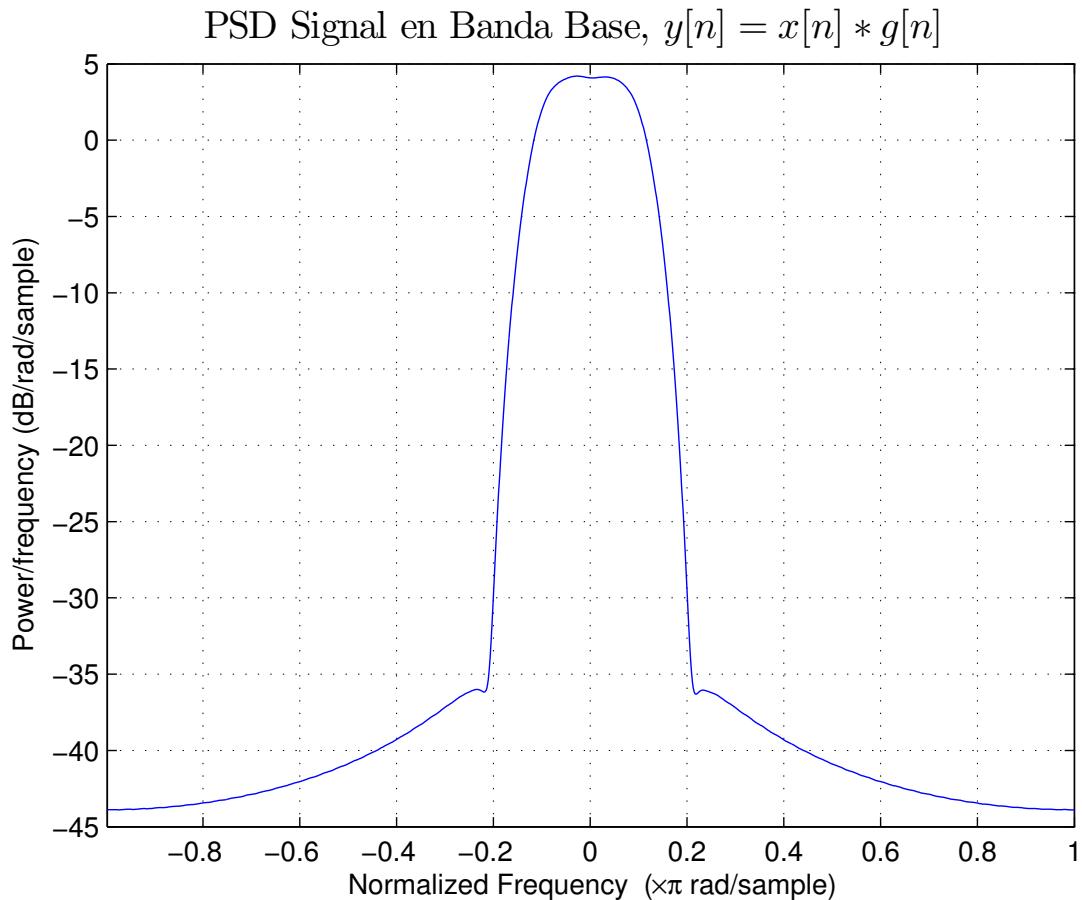


Figura 2.7: PSD de la señal en banda base  $y[n]$ .

Luego, en la Figura (2.8), se observa la PSD de la señal analítica  $s[n] = e^{j\Omega_0 n}y[n]$ , donde se confirma que el desplazamiento en frecuencia ha sido exitoso ya que se cumplen las premisas de la definición de una señal analítica.

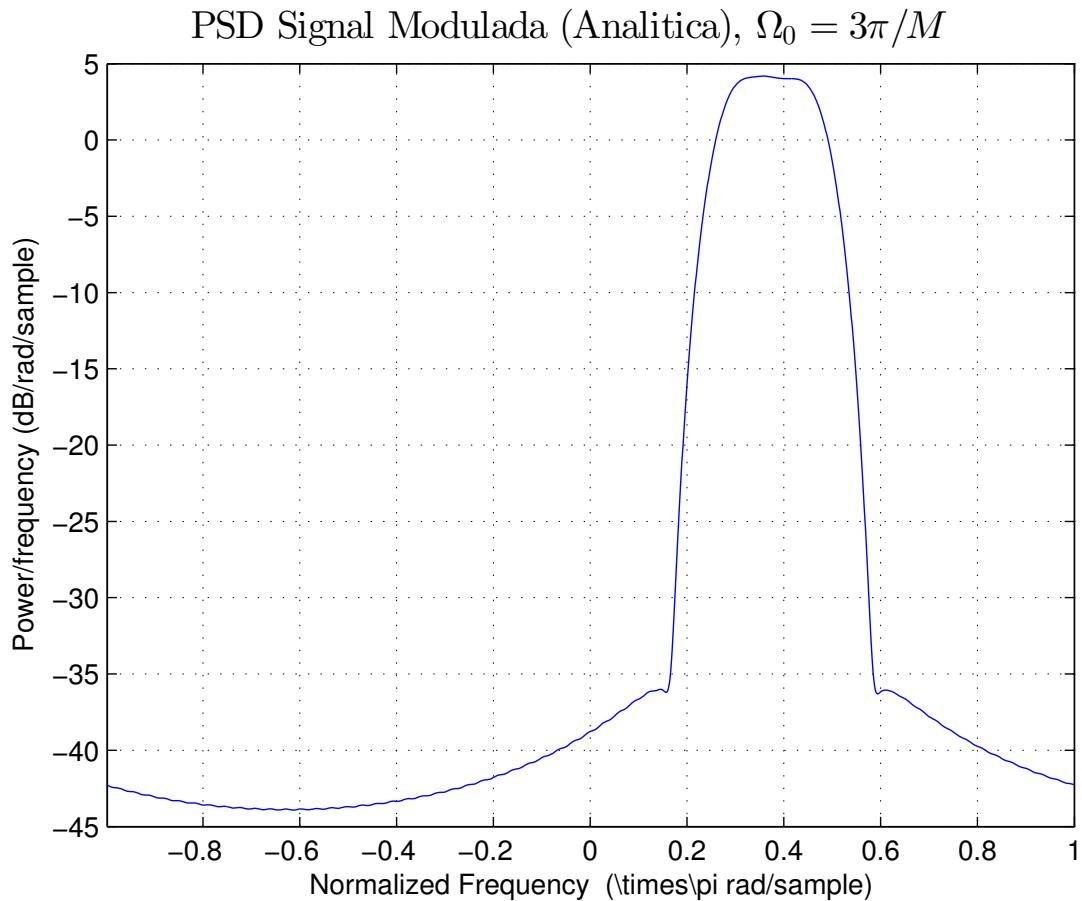


Figura 2.8: PSD de la señal analítica  $s[n]$ .

En la Figura (2.9), se puede observar la PSD de la señal  $\Re\{s[n]\}$ , donde ésta señal corresponde a la señal transmitida.

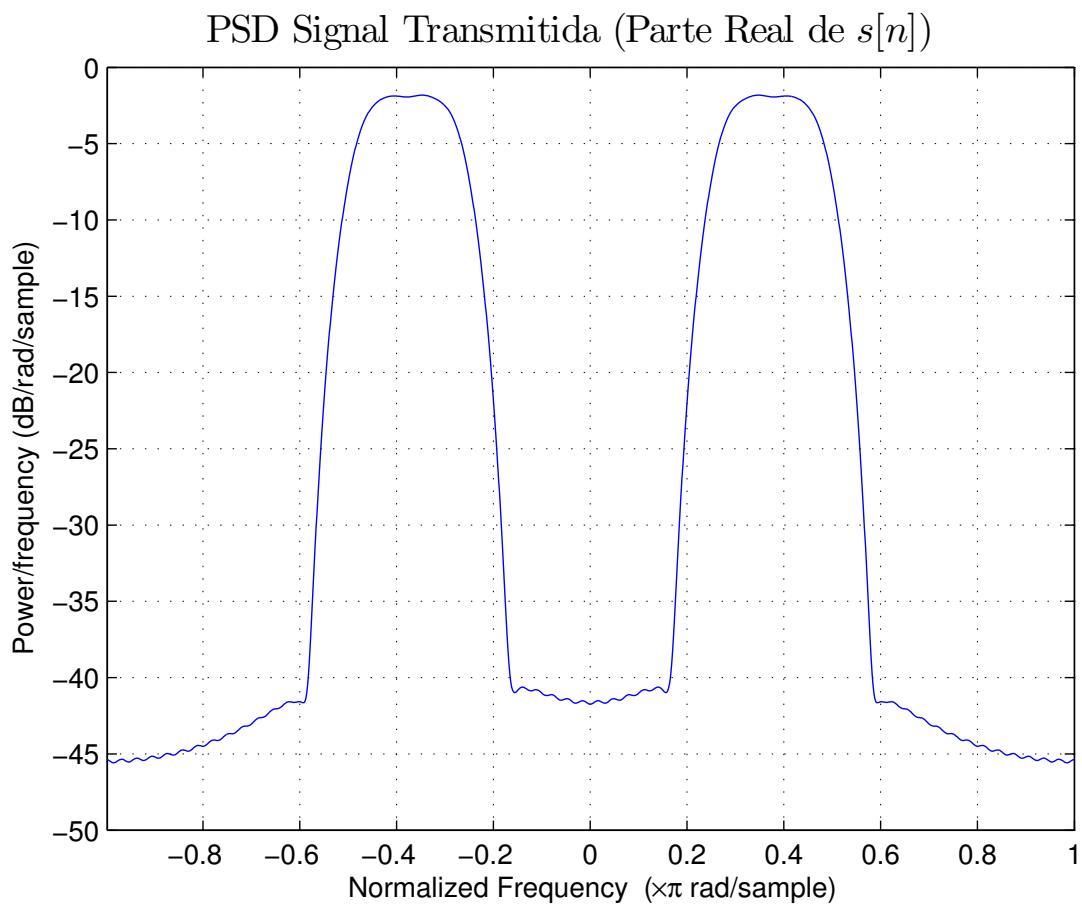


Figura 2.9: PSD de la señal de la parte Real de  $s[n]$ .

A la señal transmitida (parte real de la señal  $s[n]$ ), se la pasa por el filtro de partición de fase, dando como resultado una señal cuya PSD se muestra en la Figura (2.10).

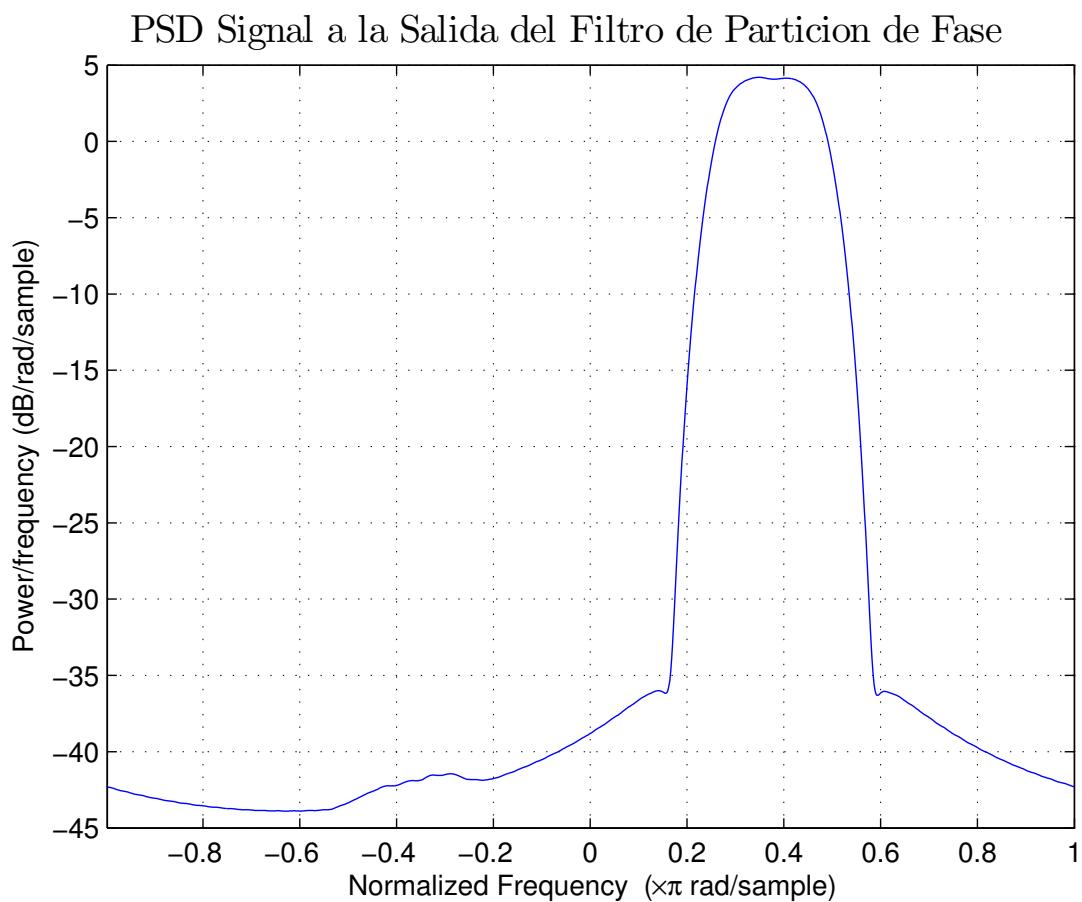


Figura 2.10: PSD de la señal a la salida del filtro de partición de fase.

Finalmente en la Figura (2.11), se observa la señal demodulada recuperada, la cual corresponde a desplazar la señal en el sentido opuesto, al desplazamiento que se llevó a cabo para la conversión en una señal analítica.

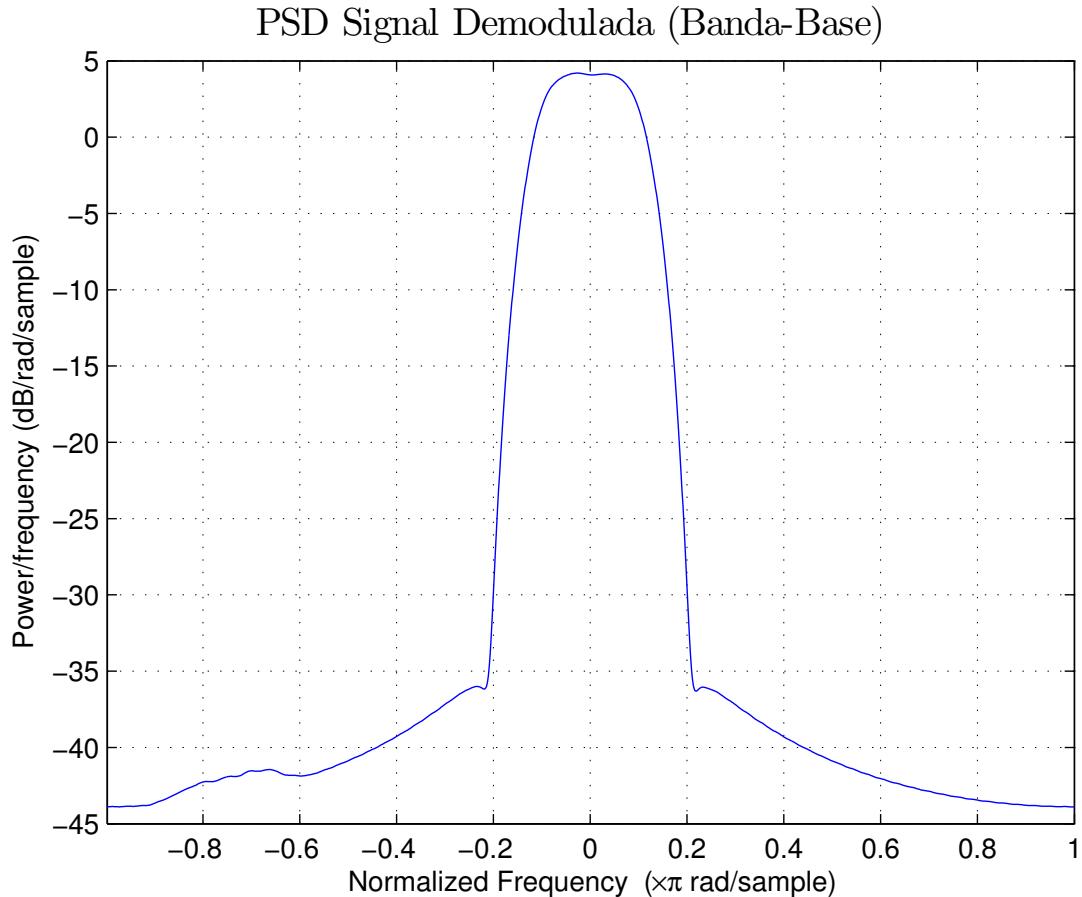


Figura 2.11: PSD de la señal recuperada.

Como se puede ver, la PSD de las señales de las Figuras (2.7) (señal original con parte real e imaginaria) y (2.11) (señal recuperada a partir de la parte real de la señal transmitida), coinciden. Por lo tanto se concluye que gracias al uso de señales analíticas y al filtro de partición de fase, se logró recuperar la información original (parte Real e Imaginaria) a partir de la transmisión y posterior recepción de la parte Real de la información original!

Como muestra final, en la Figura (2.12), se compara una porción de la parte Imaginaria de la señal original transmitida y la parte Imaginaria de la señal recuperada y se concluye que la operación ha sido exitosa y la información recuperada es totalmente válida.

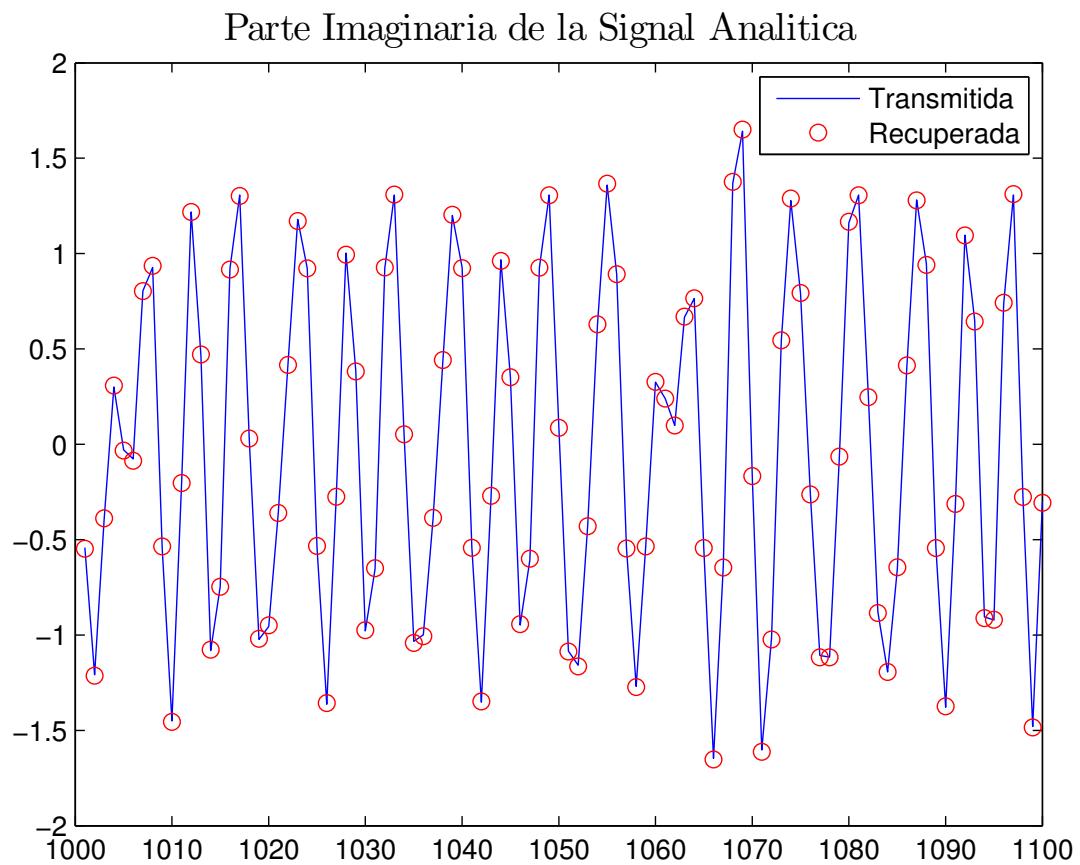


Figura 2.12: Comparación entre la parte Imaginaria de la señal transmitida y la recibida.

Un poco de matemática...

En ésta subsección del Capítulo 2 se presentan algunas verificaciones, demostraciones y deducciones matemáticas, propuestas en las consignas del laboratorio.

Sea  $p[n]$ , la salida del filtro transmisor dado por:

$$p[n] = \sum_k a[k]g[n - kM] \quad (2.7)$$

Y considerando que  $E\{|a_k|^2\} = 1$ . Se puede verificar que:

$$E\{|p[n]|^2\} = \sum_k |g[n - kM]|^2 \quad (2.8)$$

Para ello, a partir de  $p[n]$  hallamos:

$$p[n] = \sum_k a[k]g[n - kM] \quad (2.9)$$

$$p^*[n] = \sum_{k'} a^*[k']g^*[n - k'M] \quad (2.10)$$

A partir de las Ecuaciones (2.8), (2.9) y (2.10), podemos operar hasta llegar a la expresión de la Ecuación (2.8):

$$E\{|p[n]|^2\} = E\{p[n] \cdot p^*[n]\} \quad (2.11)$$

$$= E \left\{ \sum_k \sum_{k'} a[k]a^*[k']g[n - kM]g^*[n - k'M] \right\} \quad (2.12)$$

$$= \sum_k \sum_{k'} E\{a[k]a^*[k']\} g[n - kM]g^*[n - k'M] \quad (2.13)$$

$$= \sum_k \sum_{k'} \delta[k - k']g[n - kM]g^*[n - k'M] \quad (2.14)$$

$$= \sum_k |g[n - kM]|^2 \quad (2.15)$$

Seguidamente, se verifica que  $E\{|p[n]|^2\}$  es periódica y con período  $M$ :

$$E\{|p[n+M]|^2\} = \sum_k |g[n - (1-k)M]|^2 \quad (2.16)$$

$$= \sum_l |g[n - lM]|^2, \quad l = 1 - k \quad (2.17)$$

$$= E\{|p[n]|^2\} \quad (2.18)$$

sumar para todos los valores de  $k$ , es equivalente a sumar para todos los valores de  $l$ . Por lo tanto, queda verificada la periodicidad y el período de  $E\{|p[n]|^2\}$ , además por tratarse del valor absoluto al cuadrado de  $p[n]$ , la señal  $E\{|p[n]|^2\}$  es también real, aún cuando  $p[n]$  sea una señal compleja.

Luego, se verifica que la serie de Fourier de  $E\{|p[n]|^2\}$ , está dada por:

$$E\{|p[n]|^2\} = \sum_{q=<M>} b_q e^{jq\frac{2\pi}{M}n} \quad (2.19)$$

$$b_q = \frac{1}{M2\pi} \int_{2\pi} G(e^{j\Omega})G^*(e^{j(\Omega - \frac{q2\pi}{M})})d\Omega \quad (2.20)$$

Por definición, los coeficientes de la serie de Fourier de una señal periódica con período  $M$ , pueden calcularse mediante la siguiente expresión:

$$b_q = \frac{1}{M} X(e^{j\Omega})|_{\Omega=\frac{q2\pi}{M}} \quad (2.21)$$

Por lo tanto, procedemos a encontrar la transformada de Fourier en un período de la señal  $E\{|p[n]|^2\}$ . Sea  $x[n] = g[n]g^*[n]$ , se puede obtener  $X(e^{j\Omega})$  mediante la convolución circular de  $G(e^{j\Omega})$  y  $G^*(e^{-j\Omega})$  (las cuales corresponden a las transformadas de Fourier de  $g[n]$  y  $g^*[n]$  respectivamente) valuadas en  $\Omega = \frac{q2\pi}{M}$ .

$$X(e^{j\Omega}) = \frac{1}{2\pi} \int_{2\pi} G(e^{j\Theta})G^*(e^{j(\Theta - \Omega)})d\Theta \quad (2.22)$$

Finalmente, para poder obtener  $b_q$ :

$$b_q = \frac{1}{M2\pi} \int_{2\pi} G(e^{j\Theta}) G^*(e^{j(\Theta-\Omega)}) d\Theta \Big|_{\Omega=\frac{q2\pi}{M}} \quad (2.23)$$

$$b_q = \frac{1}{M2\pi} \int_{2\pi} G(e^{j\Omega}) G^*(e^{j(\Omega-\frac{q2\pi}{M})}) d\Omega \quad (2.24)$$

Seguidamente, partiendo de las siguientes ecuaciones:

$$E\{|p[n]|^2\} = \sum_{q=<M>} b_q e^{jq\frac{2\pi n}{M}} \quad (2.25)$$

$$b_q = \frac{1}{2\pi M} \int_{2\pi} G(e^{j\Omega}) G\left(e^{j(\Omega-q\frac{2\pi}{M})}\right) d\Omega \quad (2.26)$$

Ya que  $\beta < 1$ , sabemos que  $G(e^{j\Omega}) = 0$ ,  $\Omega > \frac{2\pi}{M}$ , por lo tanto  $b_k$ ,  $k \geq 2$  no tendrá valores ya que el solapamiento de  $G(e^{j\Omega})$  con sus desplazamientos mayores o iguales a  $\frac{4\pi}{M}$  es nulo, en cuyo caso de la ecuación 2.26, se desprende que sólo los coeficientes  $b_0$  y  $b_{\pm 1}$  tienen valores distintos de cero, los cuales corresponden a DC y la frecuencia fundamental:

$$b_0 = \frac{1}{2\pi M} \int_{2\pi} G(e^{j\Omega}) G^*(e^{j\Omega}) d\Omega \quad (2.27)$$

$$= \frac{1}{2\pi M} \int_{2\pi} |G(e^{j\Omega})|^2 d\Omega \quad (2.28)$$

$$b_1 = \frac{1}{2\pi M} \int_{2\pi} G(e^{j\Omega}) G\left(e^{j(\Omega-\frac{2\pi}{M})}\right) d\Omega \quad (2.29)$$

$$b_{-1} = \frac{1}{2\pi M} \int_{2\pi} G(e^{j\Omega}) G\left(e^{j(\Omega+\frac{2\pi}{M})}\right) d\Omega \quad (2.30)$$

Además, el solapamiento de las réplicas de  $G(e^{j\Omega})$ , aumenta a medida que  $\beta$  se approxima a 1, por ende el tono de sincronismo será de mayor potencia a medida que  $\beta \rightarrow 1$ .

Reemplazando (2.28), (2.29) y (2.30) en (2.25), se llega a la siguiente expresión:

$$E\{|p[n]|^2\} = \sum_{q=<M>} b_q e^{jq\frac{2\pi n}{M}} \quad (2.31)$$

$$= \frac{1}{2\pi M} \int_{2\pi} |G(e^{j\Omega})|^2 + \frac{1}{2\pi M} \int_{2\pi} G(e^{j\Omega}) G\left(e^{j(\Omega-\frac{2\pi}{M})}\right) e^{j\frac{2\pi n}{M}} d\Omega \quad (2.32)$$

$$+ \frac{1}{2\pi M} \int_{2\pi} G(e^{j\Omega}) G\left(e^{j(\Omega+\frac{2\pi}{M})}\right) e^{-j\frac{2\pi n}{M}} d\Omega \quad (2.33)$$

$$= b_0 + 2 |b_1| \cos\left(\frac{2\pi}{M}n + \angle b_1\right) \quad (2.34)$$

O visto de otra manera, como la señal  $E\{|p[n]|^2\}$  es periódica y real, entonces  $b_1 = b_{-1}^*$ , se llega a la expresión:

$$E\{|p[n]|^2\} = b_0 + b_1 e^{(j \frac{2\pi}{M} n)} + b_1^* e^{(-j \frac{2\pi}{M} n)} \quad (2.35)$$

$$= b_0 + 2\Re(b_1 e^{(j \frac{2\pi}{M} n)}) \quad (2.36)$$

Se concluye, por lo tanto que existe un tono senoidal periódico, con período  $M$ . Como fue mencionando en clases, a este método se lo conoce como método de línea espectral y es ampliamente utilizado para la recuperación de sincronismo. En las Figuras (2.13) y (2.14), se puede observar y comprobar por simulación, la presencia de dichos tonos para distintos valores de sobremuestreo.

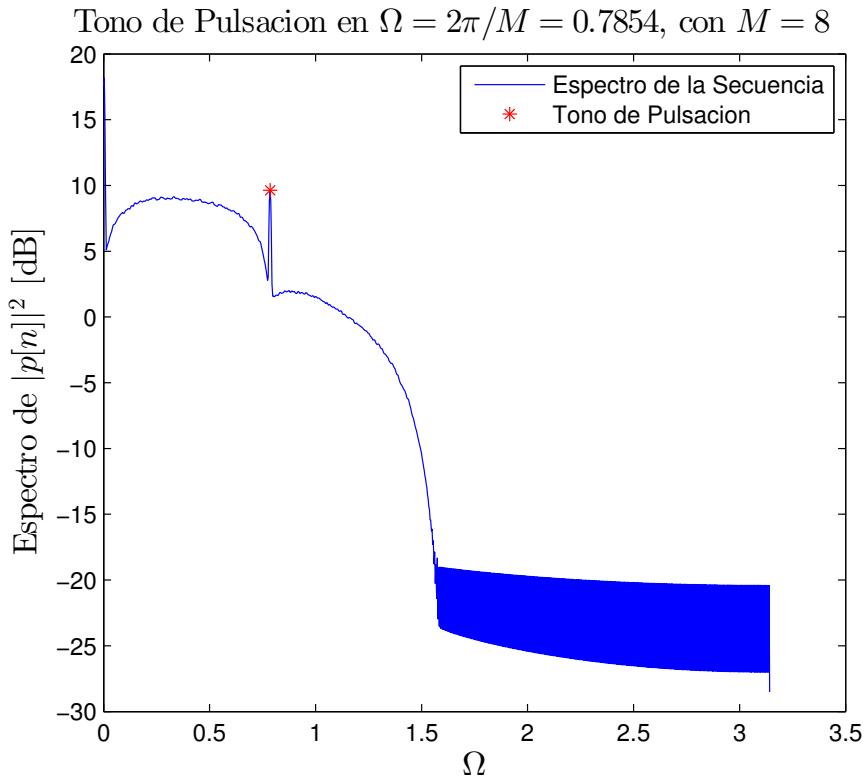


Figura 2.13: Tono de pulsación presente en el espectro de la señal  $E\{|p[n]|^2\}$ , con  $M = 8$ .

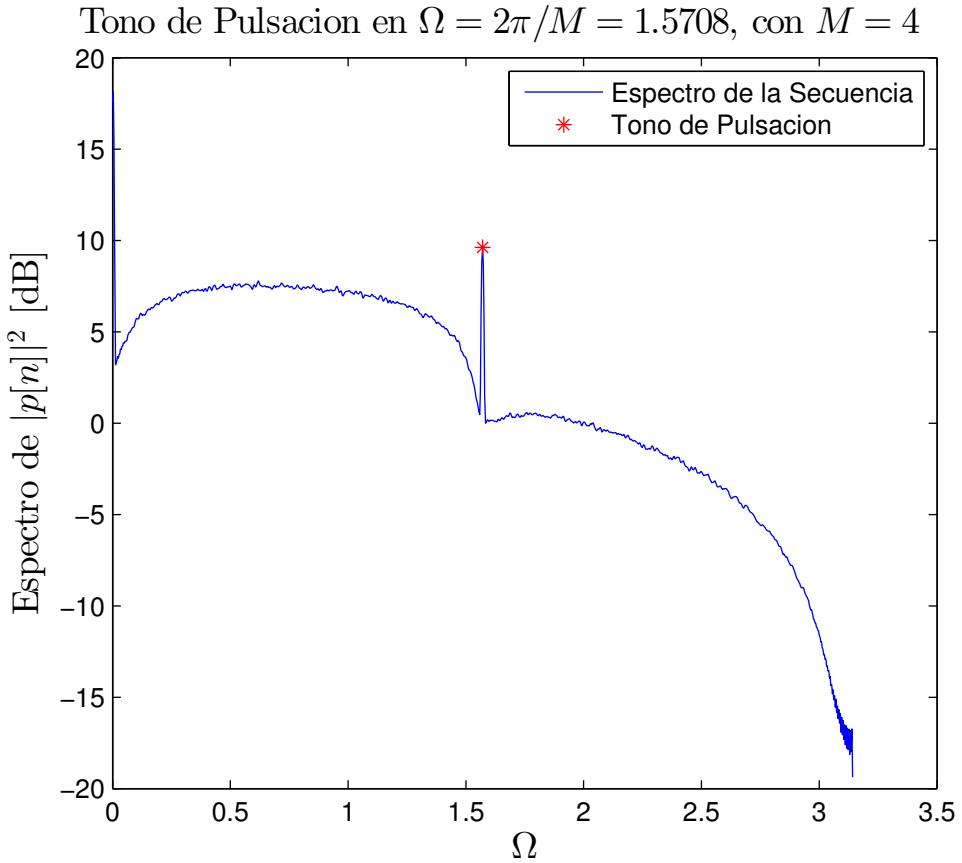


Figura 2.14: Tono de pulsación presente en el espectro de la señal  $E\{|p[n]|^2\}$ , con  $M = 4$ .

Siguiendo con el mismo razonamiento, se define una nueva señal para un segundo canal independiente:

$$\hat{p}[n] = \sum_k \hat{a}[k]g[n - kM - n_d] \quad (2.37)$$

Donde  $n_d$  corresponde a una cierta demora y  $\hat{a}[k]$  es independiente de  $a[k]$ . Anteriormente, se calcularon los coeficientes de la serie de Fourier usando:

$$b_q = \frac{1}{M} X(e^{j\Omega}) \Big|_{\Omega=\frac{q2\pi}{M}} \quad (2.38)$$

$$= \frac{1}{2\pi M} \int_{2\pi} G(e^{j\Omega})G\left(e^{j(\Omega-q\frac{2\pi}{M})}\right) d\Omega \quad (2.39)$$

Sin embargo, debido a la presencia del delay  $n_d = M/2$  y la propiedad de la transformada de Fourier;  $\mathbb{F}\{g[n - n_d]\} \rightarrow e^{-j\Omega n_d} G(e^{j\Omega})$ , se definen un nuevos  $b'_k$ :

$$b'_k = \frac{e^{-j\Omega n_d}}{M} X(e^{j\Omega}) \Big|_{\Omega=\frac{k2\pi}{M}} \quad (2.40)$$

$$b'_k = \frac{e^{-j\Omega \frac{M}{2}}}{M} X(e^{j\Omega}) \Big|_{\Omega=\frac{k2\pi}{M}} \quad (2.41)$$

Reemplazando  $\Omega = \frac{k2\pi}{M}$ :

$$b'_k = \frac{e^{-j\frac{k2\pi}{M} \frac{M}{2}}}{M} X(e^{j\Omega}) \Big|_{\Omega=\frac{k2\pi}{M}} \quad (2.42)$$

$$b'_k = \frac{e^{-jk\pi}}{M} X(e^{j\Omega}) \Big|_{\Omega=\frac{k2\pi}{M}} \quad (2.43)$$

$$b'_k = e^{-jk\pi} b_k \quad (2.44)$$

De la misma manera que el caso analizado anteriormente, cuando  $\beta < 1$ , sólo  $b_0$ ,  $b_{\pm 1}$  tiene valores distintos de cero.

$$b'_1 = e^{-j\pi} b_1 = -b_1 \quad (2.45)$$

$$b'_{-1} = e^{j\pi} b_{-1} = -b_{-1} \quad (2.46)$$

$$E\{|\tilde{p}[n]|^2\} = b_0 - b_1 e^{(j\frac{2\pi}{M}n)} - b_1^* e^{(-j\frac{2\pi}{M}n)} \quad (2.47)$$

$$= b_0 - (b_1 e^{(j\frac{2\pi}{M}n)} + b_1^* e^{(-j\frac{2\pi}{M}n)}) \quad (2.48)$$

$$= b_0 - 2\Re(b_1 e^{(j\frac{2\pi}{M}n)}) \quad (2.49)$$

En la Figura (2.15), se puede observar la presencia del tono en la señal  $\hat{p}[n]$ , cuando  $\beta < 1$  y  $n_d = M/2$ .

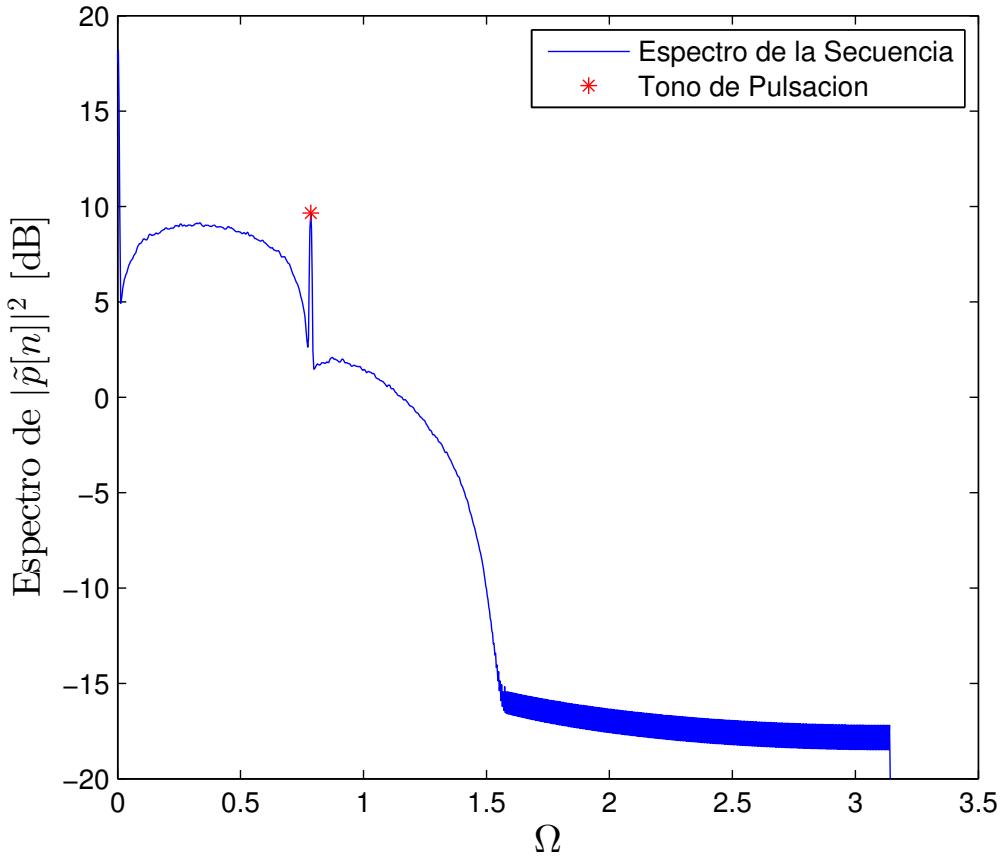
Tono de Pulsacion en Secuencia Independiente con  $n_d = M/2$  y  $M = 8$ 


Figura 2.15: Tono de pulsación presente en el espectro de la señal  $E\{|\tilde{p}[n]|^2\}$ .

Luego, si sumamos las señales tal que  $u[n] = p[n] + \hat{p}[n]$ , el tono desaparece por cancelación. Por lo tanto, en la señal  $u[n]$ , resulta imposible recuperar el tono de sincronismo mediante el método espectral de línea, debido que el tono ya no existe!

$$E\{|p[n]|^2\} = b_0 + 2\Re(b_1 e^{(j\frac{2\pi}{M}n)}) \quad (2.50)$$

$$E\{|\tilde{p}[n]|^2\} = b_0 - 2\Re(b_1 e^{(j\frac{2\pi}{M}n)}) \quad (2.51)$$

Sumando la Ecuación 2.41 y 2.42, se llega a:

$$E\{|p[n]|^2\} + E\{|\tilde{p}[n]|^2\} = 2b_0 \quad (2.52)$$

donde se queda en evidencia la cancelación del tono de sincronismo. En la Figura (2.16), se observa la cancelación del tono de sincronismo en la señal  $u[n]$ .

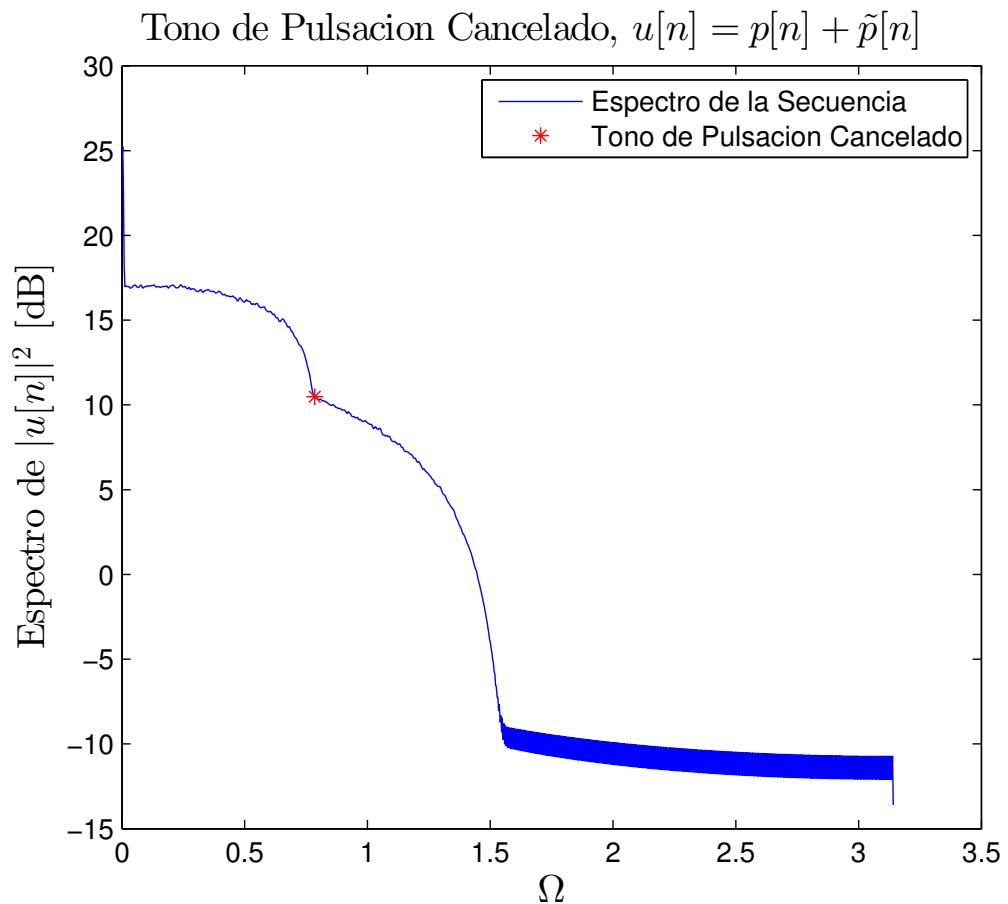


Figura 2.16: Tono de pulsación cancelado, en el espectro de la señal  $E\{|u[n]|^2\}$ .

El hecho que se haya cancelado el tono de sincronismo nos impide (al menos con éste método) implementar un sistema que siga los cambios en el reloj del transmisor, algo muy necesario en escenarios reales.

# Capítulo 3

## Lazo de Enganche de Fase (PLL)

### 3.1. Descripción y Consignas

Se presenta en la Figura (3.1), el diagrama en bloques de un dispositivo muy utilizado en ingeniería, el mismo recibe el nombre de Lazo de Enganche de Fase ó PLL, por sus siglas en inglés. En nuestro caso se trata de su versión en tiempo discreto, sin embargo existen versiones analógicas.

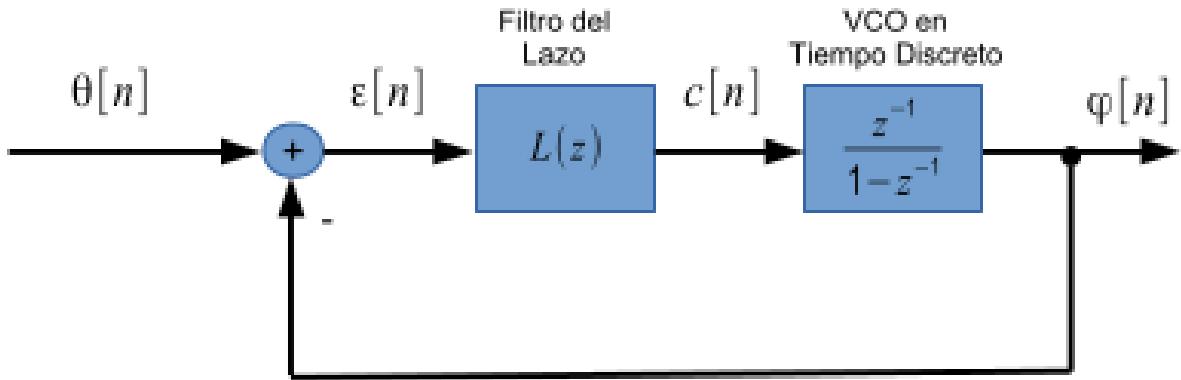


Figura 3.1: Diagrama en Bloques de un PLL Discreto.

Se propone desarrollar las siguientes actividades:

1. Utilizar el teorema del valor final,  $x[\infty] = \lim_{z \rightarrow 1} (z - 1)X(z)$ , para encontrar una expresión del error en régimen permanente, en función de la entrada y del filtro de lazo del sistema.
2. Demostrar que para  $\theta[n] = \Omega_0 n u[n]$ , el error en régimen permanente es cero si  $L(z)$  tiene un polo en  $z = 1$ .

3. Verificar que la función en lazo cerrado es:

$$H(e^{j\Omega}) = \frac{K_p}{e^{j\Omega} + K_p - 1} \quad (3.1)$$

con frecuencia de corte en  $\Omega_c \approx K_p$ .

4. Analizar la respuesta del lazo para los siguientes filtros:

$$\begin{aligned} L_1(z) &= z^{-D} K_p \\ L_2(z) &= z^{-D} \left( K_p + K_i \frac{z^{-1}}{1 - z^{-1}} \right) \end{aligned}$$

5. Analizar la estabilidad en función de la latencia de lazo  $D$ , para ello hacer uso del diagrama de polos y ceros.
6. Verificar magnitud y respuesta en frecuencia mediante simulación, para entradas del tipo  $\theta[n] = A \cos(\Omega n)$ .

## 3.2. Resultados

Antes de presentar los resultados de simulación, vamos a mencionar algunas características interesantes de los PLLs, a modo de introducción al tema.

El PLL ó Lazo de Enganche de Fase, es un sistema realimentado muy utilizado en ingeniería, el cual genera una señal de salida con amplitud fija y cuya frecuencia y fase coincide con la frecuencia y fase de la señal de referencia que ingresa al mismo (dentro de un margen dado). En rasgos generales un PLL está constituido por tres elementos básicos, los cuales se definen y describen a continuación.

1. **Detector de Fase:** También se lo suele llamar Discriminador de Fase ó Comparador de Fase, suministra una salida que depende del valor absoluto del defasaje entre la señal entregada por el VCO y la señal de entrada de referencia.
2. **Filtro Pasa Bajo:** Debido a que la salida del detector de fase es ruidosa, con el propósito de suavizar la estimación se utiliza un filtro pasa bajos, cuya salida entrega una componente de baja frecuencia destinada al control del VCO. En este laboratorio se utilizaron filtros proporcionales y proporcionales más integrales, a los cuales se los conoce como Filtros de Lazo (nombre más genérico utilizado).
3. **Oscilador Controlado por Tensión (VCO):** La señal de control, proporcionada por el filtro y cuyo valor es proporcional al error de fase, cambiará la frecuencia de oscilación del VCO y disminuirá la diferencia de fase con la referencia. De esta manera, una vez estabilizado el sistema, tanto la señal de entrada como la de salida del PLL (considerando la salida del PLL, como la salida del VCO) estarán enganchadas en fase y frecuencia.

En la Figura (3.2), se muestra un diagrama de bloques de un PLL analógico, donde se destacan los tres elementos descriptos en el párrafo anterior, además se puede observar el lazo de realimentación entre la entrada y la salida. Es justamente este lazo, el que permite que el sistema disminuya el error de fase (problema básico de control autómático). Si bien se muestra un diagrama de bloques de un PLL analógico, existen versiones completamente digitales, las cuales responden al mismo esquema.

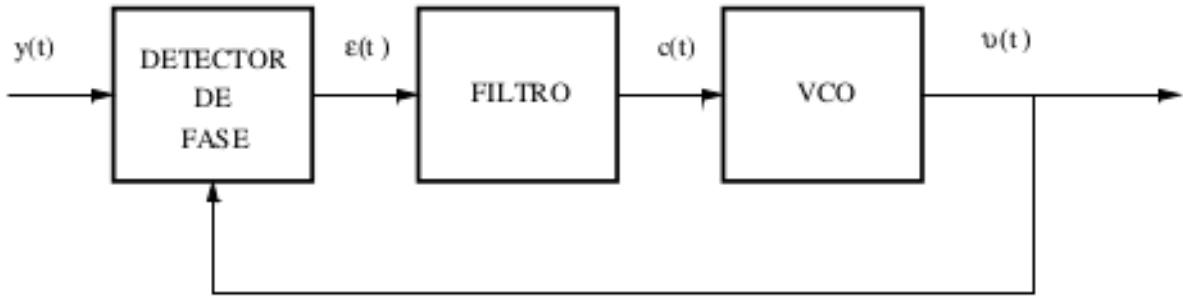


Figura 3.2: Diagrama en Bloques básico de un PLL de tiempo continuo.

Análisis PLL Discreto

A continuación vamos a hallar la función de lazo cerrado ( $H(z) = \Phi(z)/\Theta(z)$ ) de un PLL digital, donde  $\Phi(z)$  es la señal del salida del PLL y  $\Theta(z)$  la señal de entrada. Para ello vamos a definir las Ecuaciones (3.2) y a partir de ellas vamos a llegar a la expresión final de la función de lazo cerrado  $H(z)$ .

$$\begin{aligned} E(z) &= \Theta(z) - \Phi(z) \\ \Phi(z) &= E(z)L(z)\frac{z^{-1}}{1-z^{-1}} \end{aligned} \quad (3.2)$$

Donde  $E(z)$  es la señal de error de fase,  $L(z)$  es el filtro de lazo,  $\frac{z^{-1}}{1-z^{-1}}$  corresponde al modelo discreto del VCO,  $\Phi(z)$  es la señal del salida del PLL y  $\Theta(z)$  la señal de entrada.

Reemplazando  $E(z)$  en  $\Phi(z)$  y operando, se obtiene la expresión para la función de lazo cerrado:

$$\begin{aligned} \Phi(z) &= (\Theta(z) - \Phi(z))L(z)\frac{z^{-1}}{1-z^{-1}} \\ \Phi(z) &= \Theta(z)L(z)\frac{z^{-1}}{1-z^{-1}} - \Phi(z)L(z)\frac{z^{-1}}{1-z^{-1}} \\ \Phi(z)\left(1 + L(z)\frac{z^{-1}}{1-z^{-1}}\right) &= \Theta(z)L(z)\frac{z^{-1}}{1-z^{-1}} \\ \Phi(z)\left(\frac{1-z^{-1}+L(z)z^{-1}}{1-z^{-1}}\right) &= \Theta(z)L(z)\frac{z^{-1}}{1-z^{-1}} \\ H(z) = \frac{\Phi(z)}{\Theta(z)} &= \frac{L(z)z^{-1}}{1-z^{-1}+L(z)z^{-1}} \end{aligned}$$

Por otra parte, el error en regimen permanente  $\epsilon[n]$ , se define como:

$$\epsilon_{ss} = \lim_{n \rightarrow \infty} \epsilon[n] \quad (3.3)$$

Alternativamente, utilizando el teorema del valor final, se puede expresar el error en regimen permanente como:

$$\epsilon[\infty] = \lim_{z \rightarrow 1^-} (z - 1)E(z) \quad (3.4)$$

Partiendo de las Ecuaciones (3.2), hallamos un expresión para  $E(z)$  en función de  $\Theta(z)$  y  $L(z)$ , además por comodidad de cálculo, se expresa  $E(z)$  en potencias positivas de  $z$ :

$$\begin{aligned} E(z) \left( 1 + \frac{L(z)z^{-1}}{1 - z^{-1}} \right) &= \Theta(z) \\ E(z) \left( \frac{1 - z^{-1} + L(z)z^{-1}}{1 - z^{-1}} \right) &= \Theta(z) \\ E(z) &= \frac{1 - z^{-1}\theta(z)}{1 - z^{-1} + L(z)z^{-1}} \\ &= \frac{z(1 - z^{-1})\Theta(z)}{z(1 - z^{-1} + L(z)z^{-1})} \\ &= \frac{(z - 1)\Theta(z)}{(z - 1) + L(z)} \end{aligned}$$

Y aplicando el teorema del valor final, llegamos a la siguiente expresión:

$$\begin{aligned} \epsilon[\infty] &= \lim_{z \rightarrow 1^-} (z - 1)E(z) \\ &= \lim_{z \rightarrow 1^-} (z - 1) \frac{(z - 1)\Theta(z)}{(z - 1) + L(z)} \\ &= \lim_{z \rightarrow 1^-} \frac{(z - 1)^2\Theta(z)}{(z - 1) + L(z)} \end{aligned}$$

Como se puede ver en las ecuaciones anteriores, queda en evidencia que el error en regimen permanente, depende tanto de la respuesta del filtro de lazo  $L(z)$ , como de la señal de entrada  $\Theta(z)$ .

A continuación, se supone una entrada  $\theta[n] = \Omega_0 n u[n]$  y se demuestra que si el filtro de lazo posee un polo en  $z = 1$ , el error en régimen permanente es  $\epsilon_{ss} = 0$ . Para ello, se reemplaza  $\Theta(z)$ , en la expresión del error en régimen permanente y se opera, donde  $\Theta(z)$  es la transformada Z de la señal de entrada  $\theta[n]$  y  $L(z)$  posee un polo en  $z = 1$ .

$$\Theta(z) = \mathbb{Z}\{\theta[n]\} = \frac{\Omega_0 z}{(z - 1)^2} \quad (3.5)$$

$$L(z) = \frac{1}{z - 1} \quad (3.6)$$

Por lo tanto se tiene:

$$\begin{aligned} \epsilon[\infty] &= \lim_{z \rightarrow 1} \frac{(z - 1)^2 \Omega_0 z}{(z - 1 + L(z))(z - 1)^2} \\ &= \lim_{z \rightarrow 1} \frac{\Omega_0 z}{(z - 1 + L(z))} \\ &= \frac{\Omega_0}{L(z)}|_{z=1} \\ &= \frac{\Omega_0}{\left(\frac{1}{z-1}\right)}|_{z=1} \\ &= 0 \end{aligned}$$

Seguidamente, dado el filtro de lazo  $L(z) = K_p$ , se verifica que la función de lazo cerrado es:

$$H(e^{j\Omega}) = \frac{K_p}{e^{j\Omega} + K_p - 1} \quad (3.7)$$

Con frecuencia de corte ( $-3$  dB) en  $\Omega_c \approx K_p$ .

Anteriormente, se obtuvo la expresión de la transformada Z de la función de lazo cerrado del PLL, dado por:

$$H(z) = \frac{\Phi(z)}{\Theta(z)} = \frac{L(z)z^{-1}}{1 - z^{-1} + L(z)z^{-1}} \quad (3.8)$$

Teniendo en cuenta que  $H(e^{j\Omega}) = H(z)|_{z=e^{j\Omega}}$  y  $L(z) = K_p$ , se verifica que:

$$H(e^{j\Omega}) = H(z)|_{z=e^{j\Omega}} = \frac{K_p}{e^{j\Omega} + K_p - 1} \quad (3.9)$$

Por otra parte, la frecuencia de corte se define como la frecuencia  $\Omega_c$ , tal que  $|H(e^{j\Omega_c})| = 1/\sqrt{2}$ . Partiendo de:

$$\begin{aligned} \left| \frac{K_p}{e^{j\Omega} + K_p - 1} \right|^2 &= \left| \frac{K_p}{\cos(\Omega_c) + j\sin(\Omega_c) + K_p - 1} \right|^2 \\ &\approx \left| \frac{K_p}{K_p + j\Omega_c} \right|^2, \quad \Omega_c \ll 1 \end{aligned}$$

Además  $\Omega_c \approx K_p$ , entonces:

$$\begin{aligned} \left| \frac{K_p}{K_p + j\Omega_c} \right|^2 &\approx \left| \frac{K_p}{K_p + jK_p} \right|^2 \\ &\approx \left| \frac{K_p}{K_p(1+j)} \right|^2 \\ &\approx \left| \frac{1}{1+j} \right|^2 \\ &\approx \frac{1}{\sqrt{2}} \end{aligned}$$

Como la frecuencia de corte corresponde a la frecuencia donde la magnitud de  $|H(e^{j\Omega})|^2 = \frac{1}{\sqrt{2}}$ , se concluye que dicha frecuencia de corte corresponde a  $\Omega_c \approx K_p$ .

### Filtros de Lazo y Estabilidad

A continuación, se analiza la estabilidad del PLL para las siguientes funciones de lazo (con un modelado de latencia incluido en ellas):

$$\begin{aligned} L_1(z) &= z^{-D} K_p \\ L_2(z) &= z^{-D} \left( K_p + K_i \frac{z^{-1}}{1 - z^{-1}} \right) \end{aligned}$$

En primer lugar, se obtiene una expresión de los polinomios del numerador y del denominador de las funciones de lazo  $L_1(z)$  y  $L_2(z)$ , para luego utilizarlas en las simulaciones con MATLAB. La función de transferencia de un PLL con filtro de lazo  $L(z)$  es:

$$H(z) = \frac{\Phi(z)}{\Theta(z)} = \frac{L(z)z^{-1}}{1 - z^{-1} + L(z)z^{-1}}$$

Reemplazando  $L(z) = L_1(z)$ :

$$\begin{aligned} H_1(z) &= \frac{z^{-D} K_p z^{-1}}{1 - z^{-1} + z^{-D} K_p z^{-1}} \\ &= \frac{z^{-(D+1)} K_p}{1 - z^{-1} + z^{-(D+1)} K_p} \end{aligned}$$

Reemplazando  $L(z) = L_2(z)$ :

$$H_2(z) = \frac{z^{-(D+1)} K_p + (K_i - K_p) z^{-(D+2)}}{1 - 2z^{-1} + z^{-2} + K_p z^{-(D+1)} + (K_i - K_p) z^{-(D+2)}}$$

Con la expresión de los numeradores y denominadores de las funciones de transferencia con los distintos filtros  $L_1(z)$  y  $L_2(z)$ , se procede a analizar mediante MATLAB la estabilidad para cada caso. Para ello se barre el valor de latencia  $D$  y se observan las posiciones de los polos en el plano  $z$ .

Como comentario adicional, se agrega que la latencia representa la demora de procesamiento en las implementaciones reales, algo que uno puede llegar a ignorar ó pasar por alto en el modelo matemático, resulta de vital importancia en la realidad, ya que esta latencia de procesamiento, modifica el comportamiento esperado *a priori* del sistema y en casos extremos puede volver inestable al mismo.

Estabilidad con  $L_1(z)$ , en Función de la Latencia D

Utilizando las expresiones del numerador y denominador de la función de lazo cerrado con filtro de lazo  $L_1(z)$  y la función `zplane()` de MATLAB (permite graficar el diagrama de polos y ceros de una función de transferencia), se obtuvieron las Figuras que se muestran a continuación.

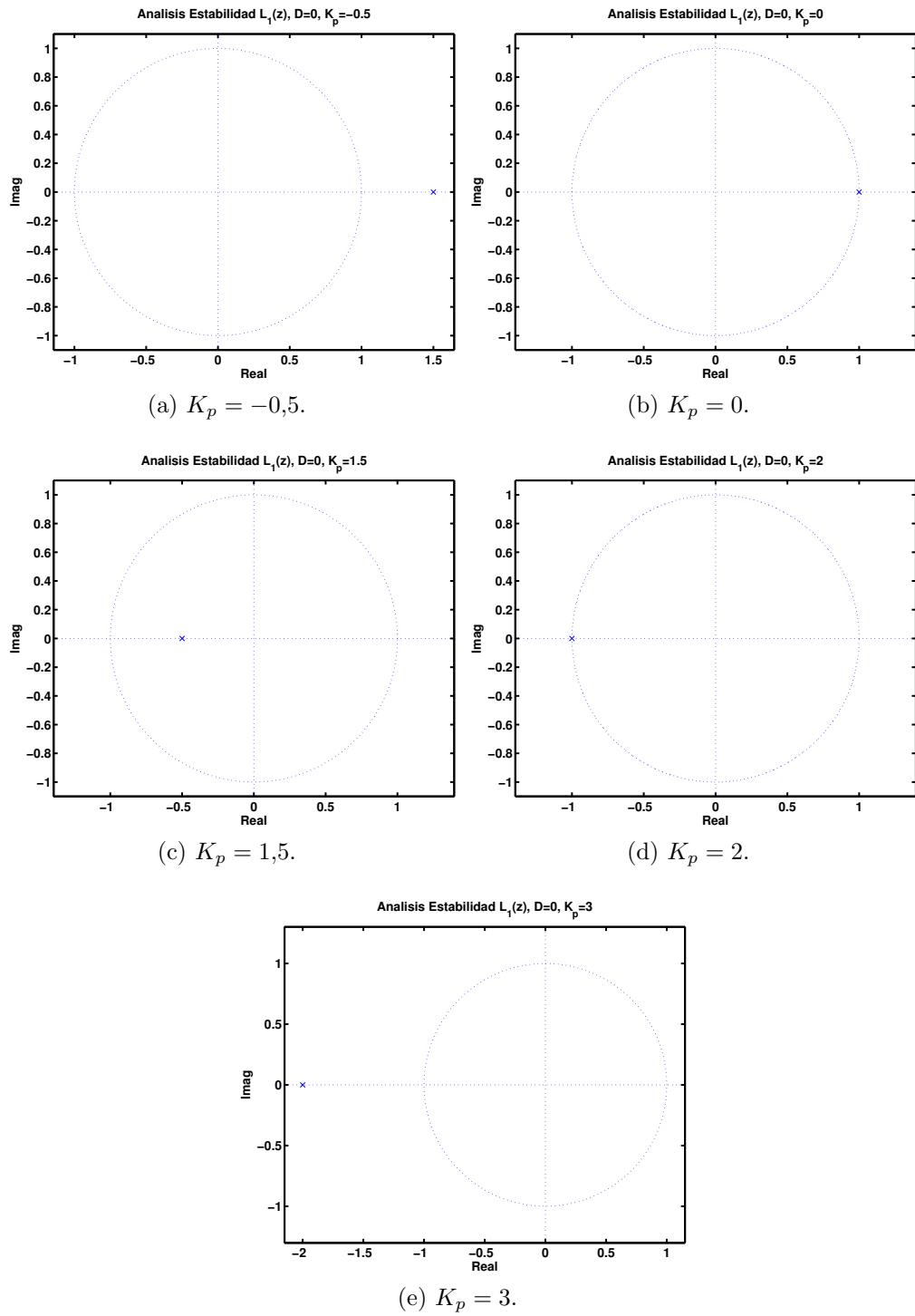


Figura 3.3: Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo  $L_1(z)$ ,  $D = 0$  y varios  $K_p$ .

Se puede ver que cuando la latencia es nula ( $D = 0$ ), el sistema posee un sólo polo y éste se mueve en el plano z, a medida que variamos  $K_p$ . Dando como resultado un sistema estable para los valores de  $0 < K_p < 2$ .

En la Figura (3.4), se aumentó la latencia hasta  $D = 20$ .

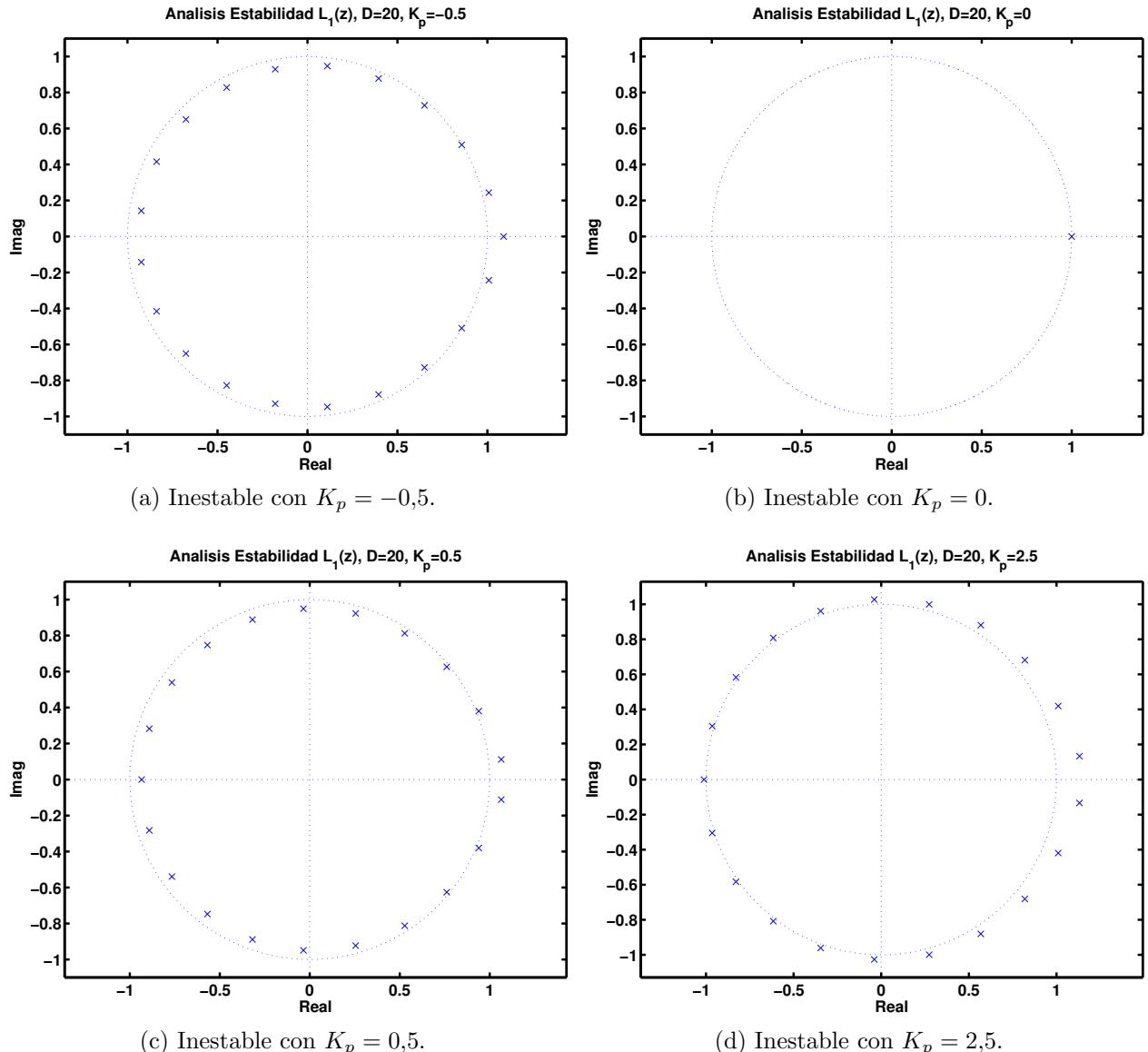


Figura 3.4: Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo  $L_1(z)$ ,  $D = 20$  y varios  $K_p$ .

Cuando la latencia toma un valor distinto de cero, como en la Figura (3.4), donde  $D = 20$ ,

aparecen más polos distribuidos alrededor del origen.

*Simplemente con el agregado de latencia, logramos una respuesta muy distinta a la original, con muchos más polos de los que aparecen en el sistema sin latencia.*

Adicionalmente, a medida que la latencia aumenta, dichos polos se ubican cada vez más cerca del círculo unitario, por lo que se espera que el fenómeno de peaking sea mayor hasta el caso extremo donde se vuelva inestable. Por lo tanto, la inclusión de latencia vuelve al sistema más propenso a las inestabilidades y con mayores niveles de peaking.

Finalmente, en la Figura (3.5), se aumentó nuevamente la latencia hasta  $D = 100$ .

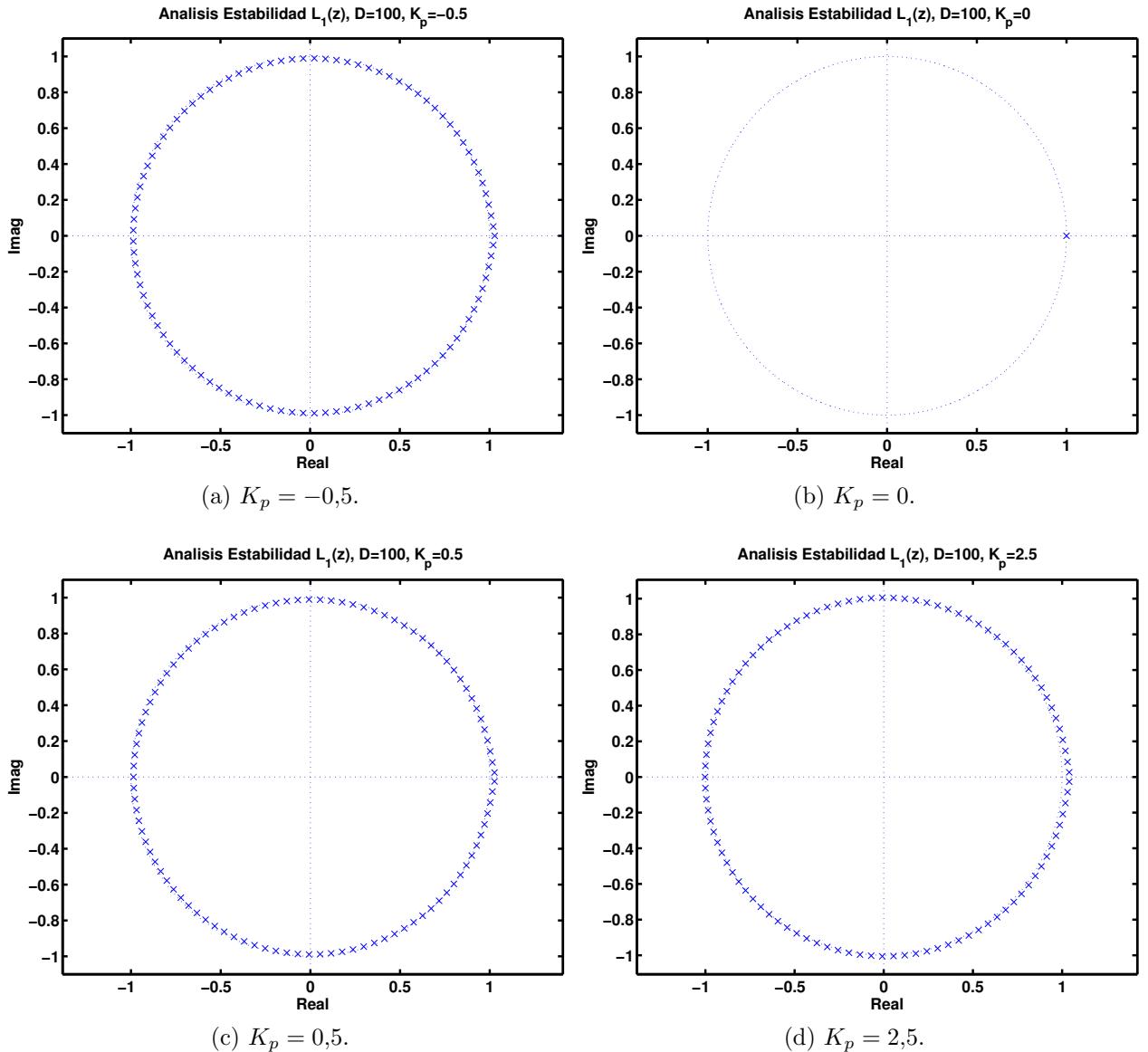


Figura 3.5: Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo  $L_1(z)$ ,  $D = 1000$  y varios  $K_p$ .

Nuevamente, al aumentar aún más la latencia, la cantidad de polos del sistema se vuelve aún mayor. En conclusión, la latencia afecta y modifica en gran medida al sistema, aumentando su orden a medida que aumenta la cantidad de retardo. Esto es muy importante y no se puede ignorar, ya que podemos tener un sistema que *a priori* considerábamos estable, como un sistema en realidad inestable ó con un nivel de peaking excesivo, simplemente por el factor de latencia que podemos llegar a pasar por alto en un primer análisis ó bien podemos conocer de antemano y buscar disminuirlo para que el efecto indeseado sea reducido.

Al código de MATLAB original del Laboratorio N°3, se le agregó una modificación que permite escribir por consola si el sistema es ESTABLE ó INESTABLE. Para ello, se utilizó la función *roots()*, la cual permite obtener las raíces de un polinomio de orden  $n$ , luego se ingresó en dicha función la expresión polinomial de los diferentes denominadores de las funciones de transferencia de lazo cerrado (con las distintas latencias y  $K_p$ ) y se analizó el valor absoluto de dichas raíces, en búsqueda de aquellas que posean un módulo mayor que 1 (corresponden a polos fuera de la circunferencia unidad), para finalmente imprimir el caso y su ESTABILIDAD. A continuación se muestran los resultados de las consolas, para varios casos, algunos de los cuales se mostraron en los diagramas de polos y ceros vistos anteriormente.

Con filtro de lazo  $L_1(z)$ :

Para  $L_1(z)$  con  $D=0$  y  $K_p=-3$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=-2.500000e+00$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=-2$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=-1.500000e+00$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=-1$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=-5.000000e-01$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=0$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=5.000000e-01$ , el sistema es: ESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=1$ , el sistema es: ESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=1.500000e+00$ , el sistema es: ESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=2$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=2.500000e+00$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=0$  y  $K_p=3$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=-3$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=-2.500000e+00$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=-2$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=-1.500000e+00$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=-1$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=-5.000000e-01$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=0$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=5.000000e-01$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=1$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=1.500000e+00$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=2$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=2.500000e+00$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=20$  y  $K_p=3$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=100$  y  $K_p=-3$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=100$  y  $K_p=-2.500000e+00$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=100$  y  $K_p=-2$ , el sistema es: INESTABLE  
Para  $L_1(z)$  con  $D=100$  y  $K_p=-1.500000e+00$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=-1$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=-5.000000e-01$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=0$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=5.000000e-01$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=1$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=1.500000e+00$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=2$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=2.500000e+00$ , el sistema es: INESTABLE

Para  $L_1(z)$  con  $D=100$  y  $K_p=3$ , el sistema es: INESTABLE

Estabilidad con  $L_2(z)$ , en Función de la Latencia D

En este caso, se analiza el sistema en lazo cerrado, cuando la función de lazo es  $L_2(z)$ . Podemos ver en las siguientes Figuras, que ahora en nuestra función de lazo cerrado existen ceros además de polos y nuevamente, la cantidad de polos alrededor del origen aumenta de manera proporcional con la cantidad de latencia presente en el sistema. Otro parámetro a tener en cuenta es la ganancia  $K_i$ , en nuestro nuevo sistema no solamente influye el parámetro  $K_p$ , ya que el filtro de lazo utilizado es un filtro proporcional más integral (también conocido en teoría de control automático, como Filtro PI).

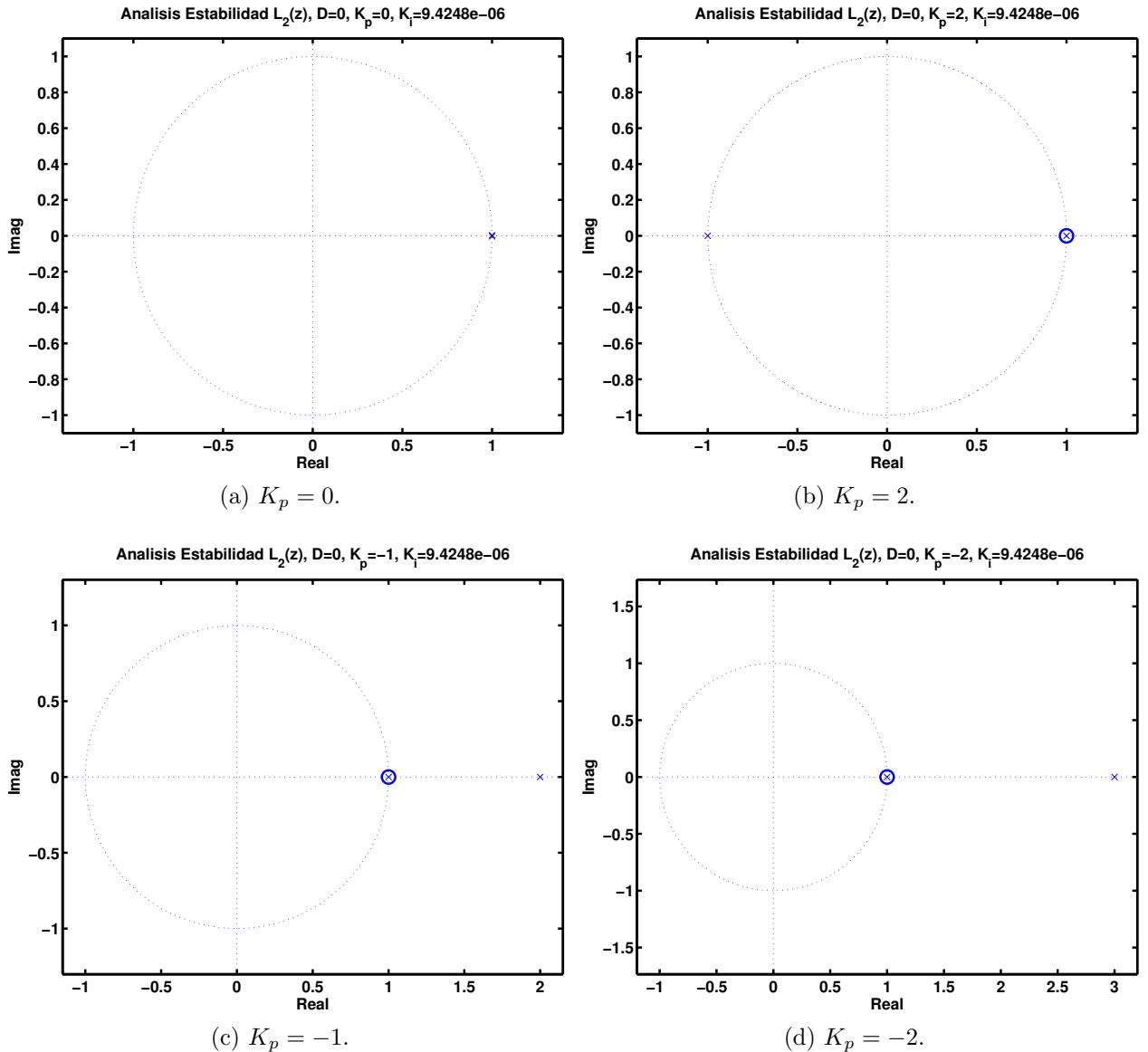


Figura 3.6: Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo  $L_2(z)$ ,  $D = 0$  y varios  $K_p$  y  $K_i$ .

La parte integral del filtro le permite al sistema, seguir entradas que varían (jitter) en el tiempo, algo muy común en la realidad. Por otra parte, la ganancia  $K_p$  sólo permite corregir errores constantes. Como normalmente, se desea poder *seguir* la entrada, la necesidad de una parte integral en el filtro de lazo, se vuelve imperiosa.

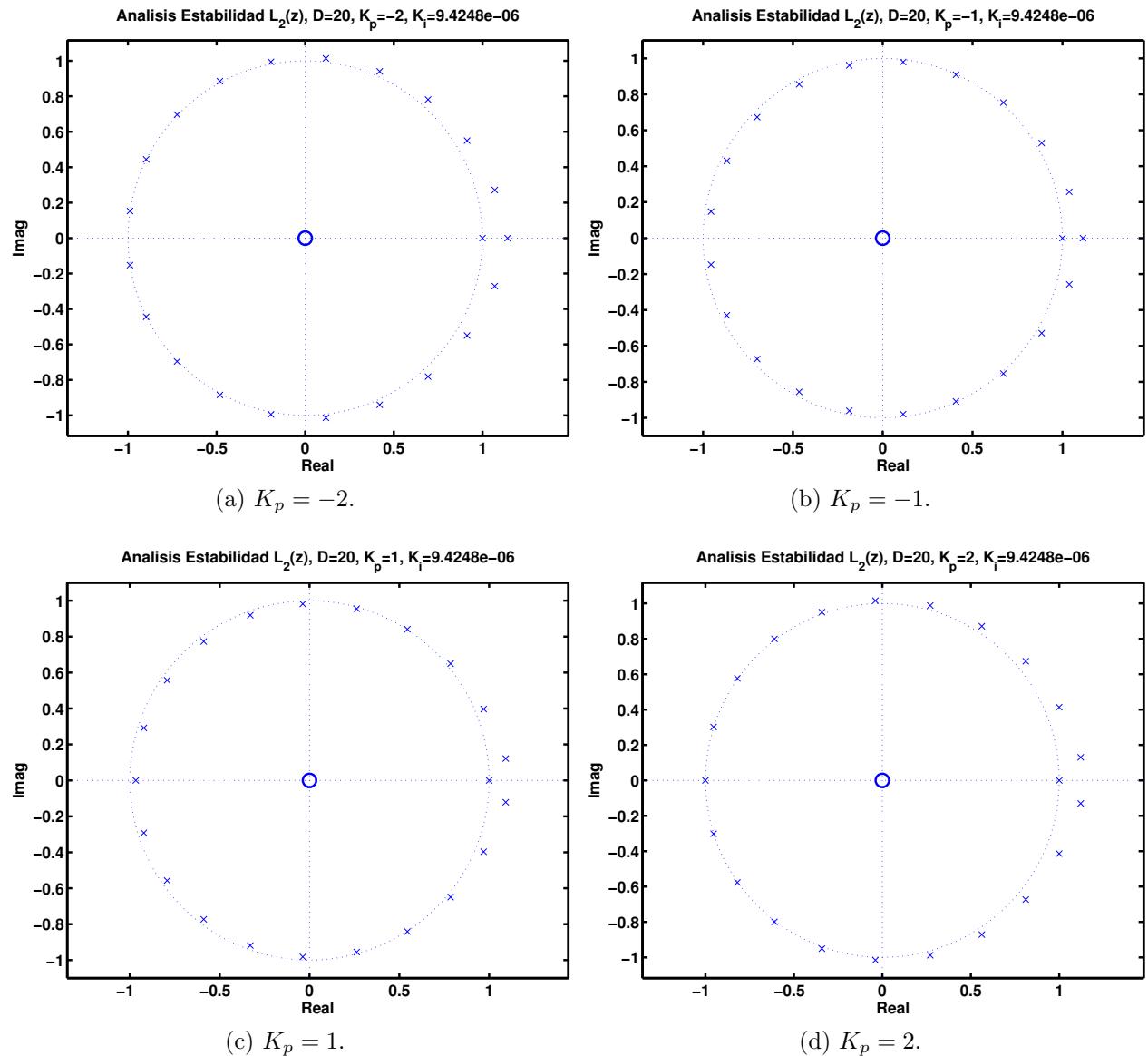


Figura 3.7: Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo  $L_2(z)$ ,  $D = 20$  y varios  $K_p$  y  $K_i$ .

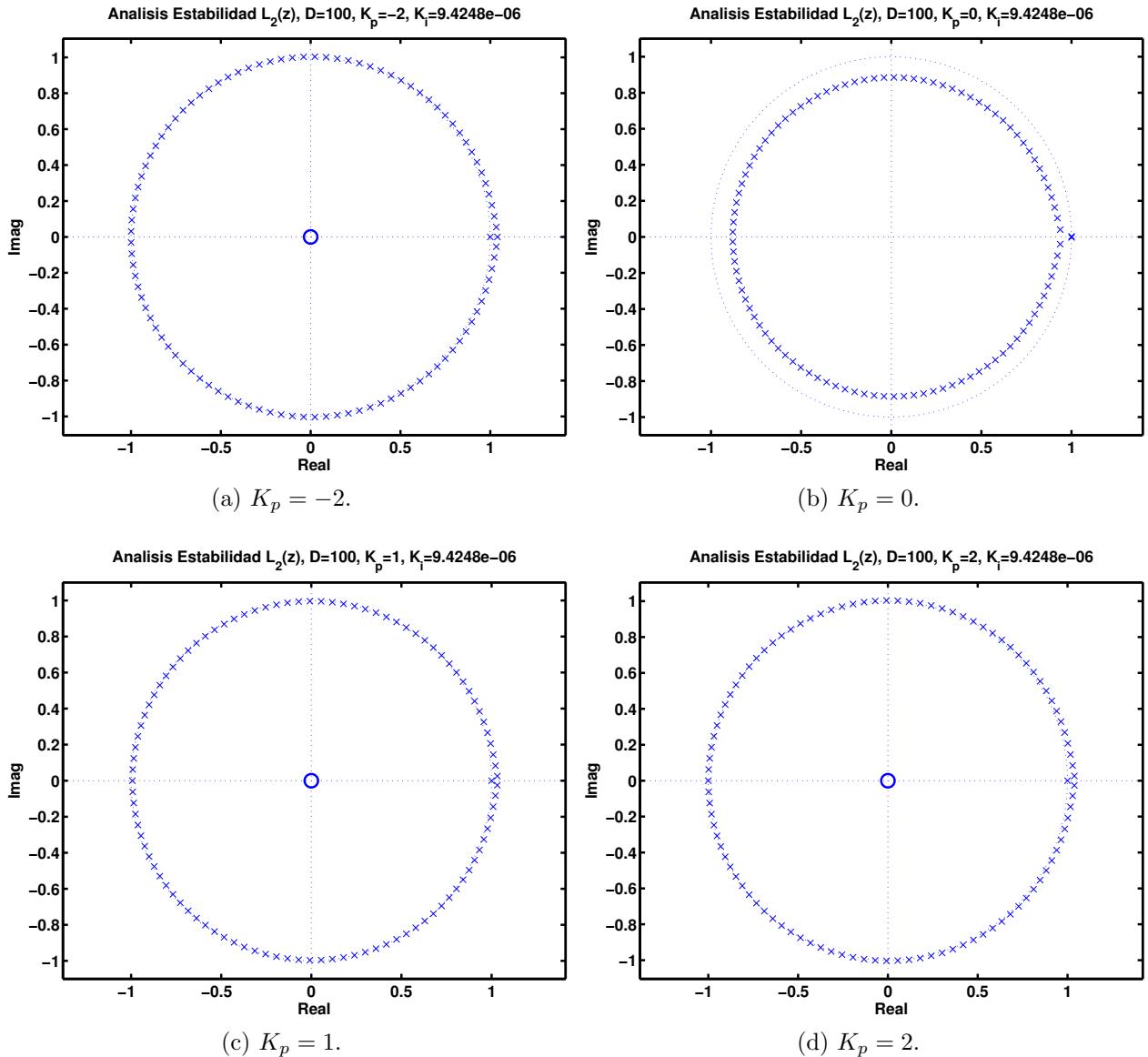


Figura 3.8: Diagrama de polos y ceros de la Función de Transferencia de Lazo cerrado con Filtro de Lazo  $L_2(z)$ ,  $D = 100$  y varios  $K_p$  y  $K_i$ .

Nuevamente, al código de MATLAB original del Laboratorio N°3, se le agregó una modificación que permite escribir por consola si el sistema es ESTABLE ó INESTABLE, tal y como se mencionó anteriormente para el filtro de lazo  $L_1(z)$ .

Con filtro de lazo  $L_2(z)$ :

Para  $L_2(z)$  con  $D=0$ ,  $K_p=-3$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE

Para  $L_2(z)$  con  $D=0$ ,  $K_p=-2$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE

Para  $L_2(z)$  con  $D=0$ ,  $K_p=-1$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=0$ ,  $K_p=0$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=0$ ,  $K_p=1$  y  $K_i=9.424778e-06$ , el sistema es: ESTABLE  
 Para  $L_2(z)$  con  $D=0$ ,  $K_p=2$  y  $K_i=9.424778e-06$ , el sistema es: ESTABLE  
 Para  $L_2(z)$  con  $D=0$ ,  $K_p=3$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=20$ ,  $K_p=-3$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=20$ ,  $K_p=-2$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=20$ ,  $K_p=-1$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=20$ ,  $K_p=0$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=20$ ,  $K_p=1$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=20$ ,  $K_p=2$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=20$ ,  $K_p=3$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=100$ ,  $K_p=-3$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=100$ ,  $K_p=-2$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=100$ ,  $K_p=-1$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=100$ ,  $K_p=0$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=100$ ,  $K_p=1$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=100$ ,  $K_p=2$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=100$ ,  $K_p=3$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE  
 Para  $L_2(z)$  con  $D=100$ ,  $K_p=3$  y  $K_i=9.424778e-06$ , el sistema es: INESTABLE

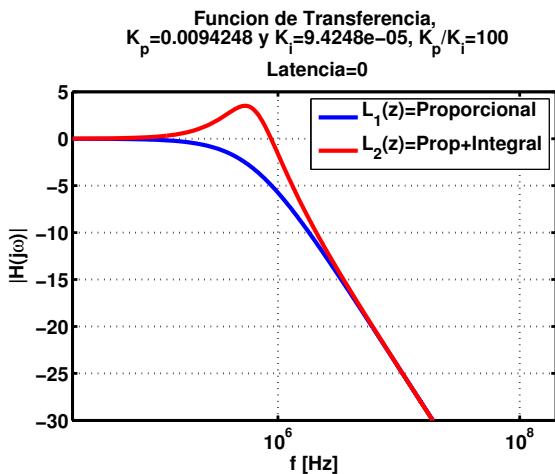
Como conclusión, se menciona que latencia del sistema es un parámetro de la implementación real y tiene relación con los tiempos de procesamiento de los sistemas reales. Si uno ignora este parámetro en el modelado, puede estar cometiendo un grave error, ya que ignorarlo puede significar tener un sistema cuya respuesta dista mucho de la deseada y en un caso extremo, nuestro sistema puede volverse inestable y por consiguiente totalmente inútil.

Debido a ésta característica introducida por la latencia en un PLL, los diseñadores luchan a diario para disminuirla, dando lugar a nuevas arquitecturas ó a nuevas optimizaciones de los algoritmos clásicos.

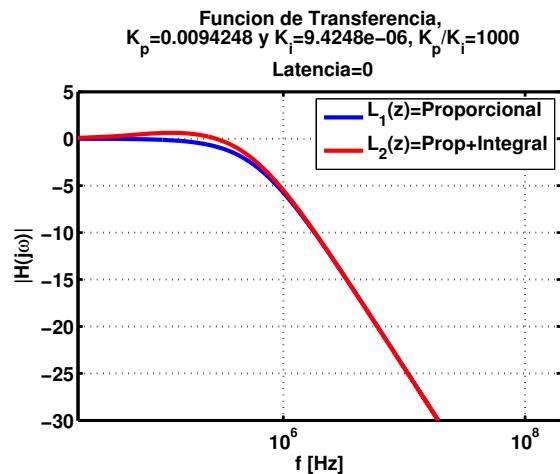
Curvas de Transferencia de Lazo Cerrado, Variando  $K_p$  y  $K_i$ .

A continuación, se muestran las curvas de transferencia del sistema con los filtros de lazo  $L_1(z)$  y  $L_2(z)$ . Dichas curvas muestran el rango de entrada para la cual el sistema no introduce perdidas en la señal ó lo que es lo mismo, el rango de funcionamiento en función de la frecuencia de entrada. Si uno comete el error de utilizar el sistema fuera de su rango de funcionamiento, la salida del mismo puede presentar atenuaciones ó al contrario, peaking con respecto a la salida deseada *a priori*.

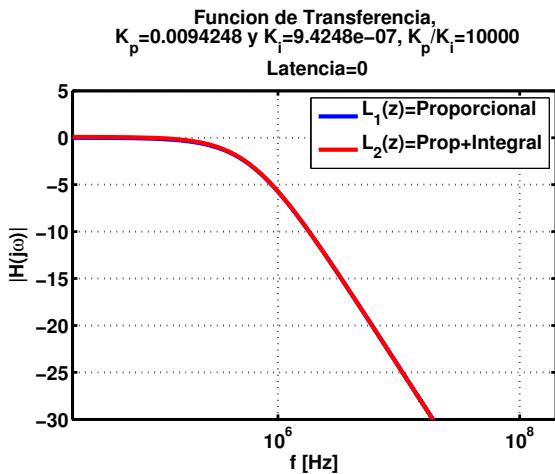
En la Figura (3.9), la latencia es de  $D = 0$ . Luego, en la Figura (3.10), la latencia es de  $D = 20$ . Finalmente, en la Figura (3.11), la latencia es de  $D = 100$ .



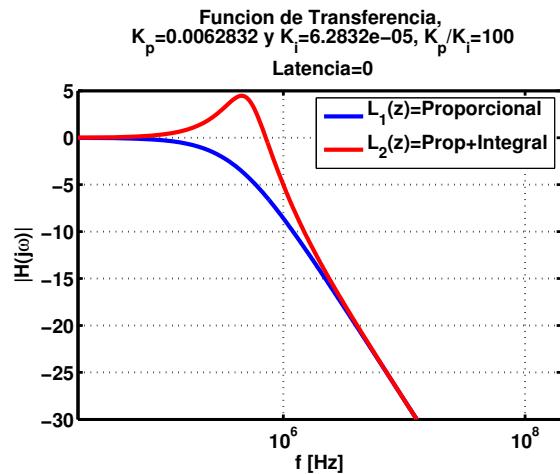
(a)  $D = 0$ ,  $K_p = 0,0094248$ ,  $K_p/K_i = 100$ .



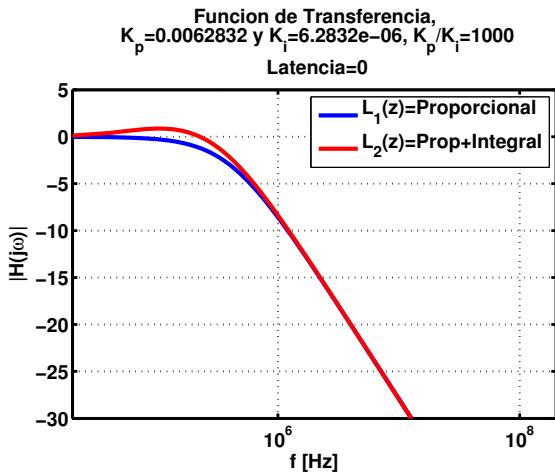
(b)  $D = 0$ ,  $K_p = 0,0094248$ ,  $K_p/K_i = 1000$ .



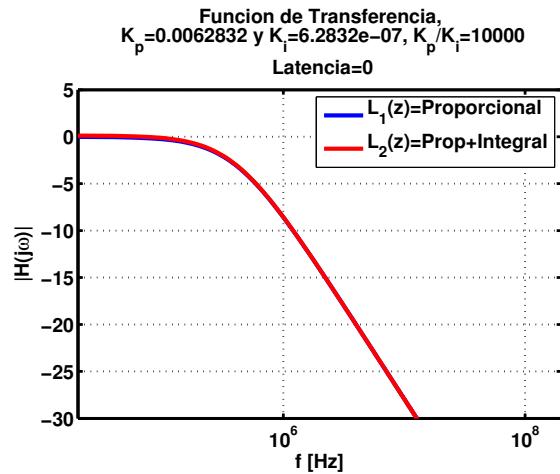
(c)  $D = 0$ ,  $K_p = 0,0094248$ ,  $K_p/K_i = 10000$ .



(d)  $D = 0$ ,  $K_p = 0,0062832$ ,  $K_p/K_i = 100$ .



(e)  $D = 0$ ,  $K_p = 0,0062832$ ,  $K_p/K_i = 1000$ .



(f)  $D = 0$ ,  $K_p = 0,0062832$ ,  $K_p/K_i = 10000$ .

Figura 3.9: Función de Transferencia del PLL,  $D = 0$  y distintos valores de  $K_p$  y  $K_i$ .

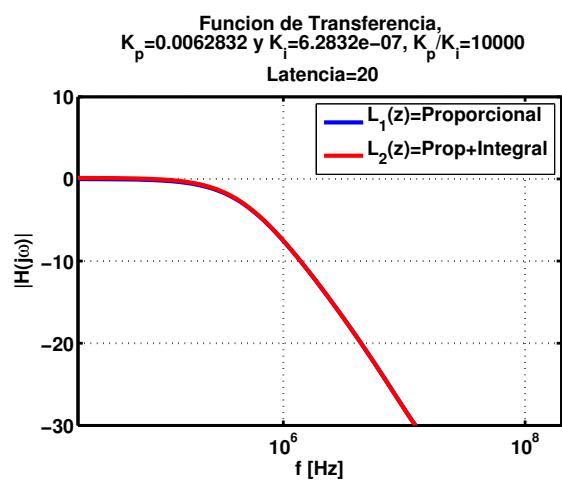
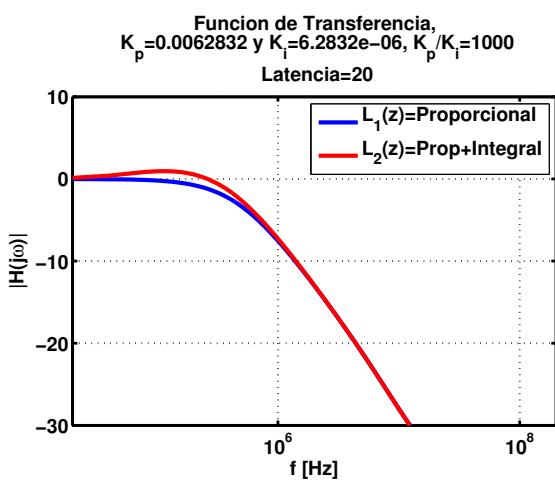
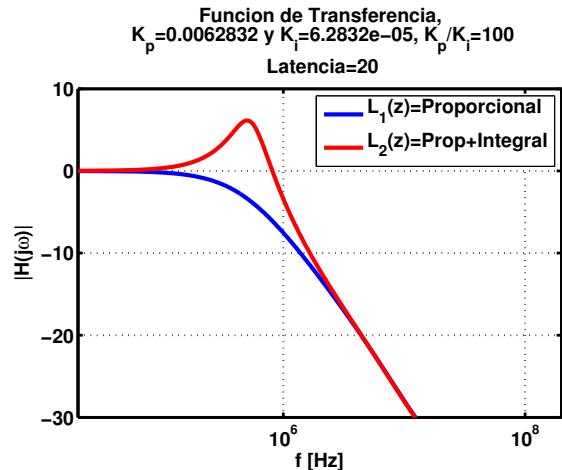
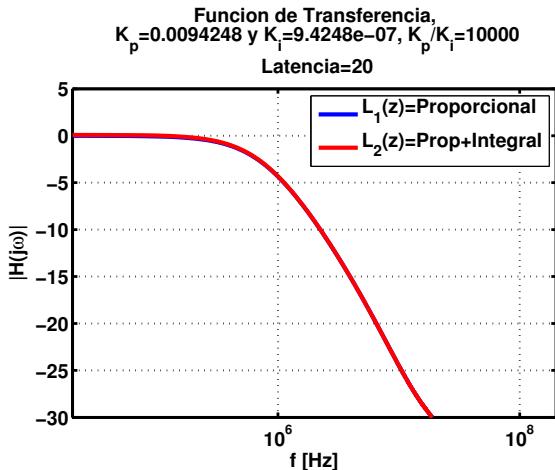
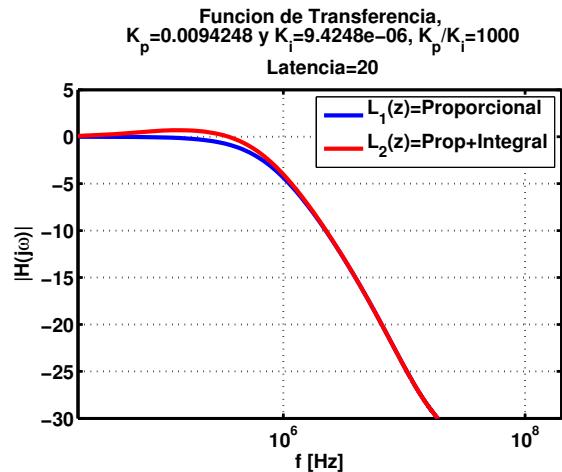
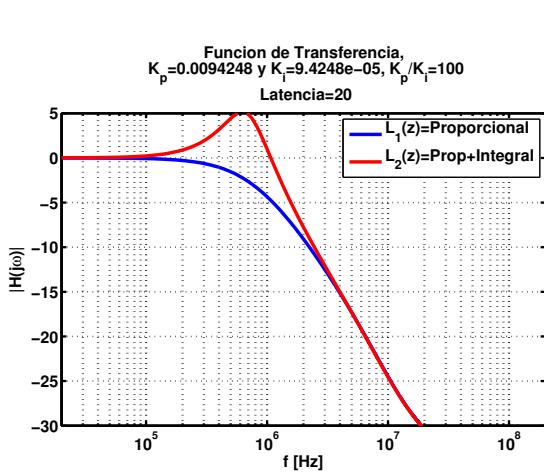
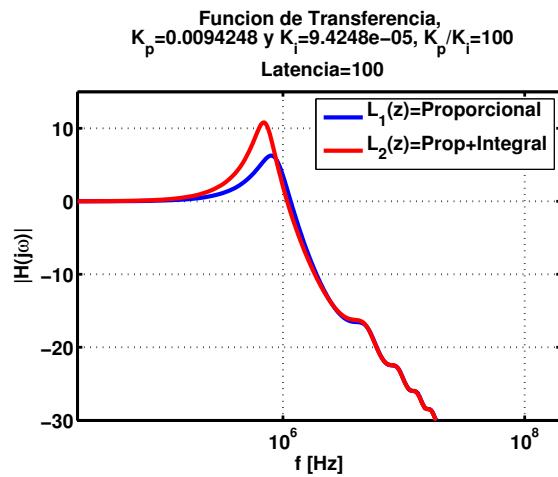
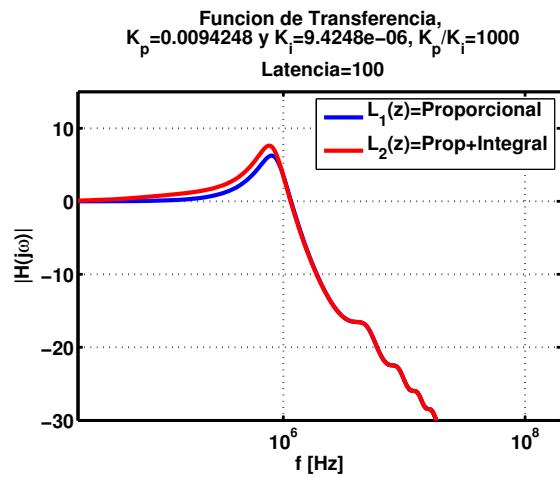


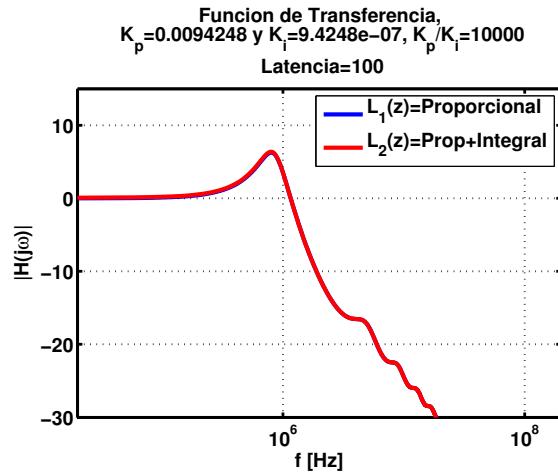
Figura 3.10: Función de Transferencia del PLL,  $D = 20$  y distintos valores de  $K_p$  y  $K_i$ .



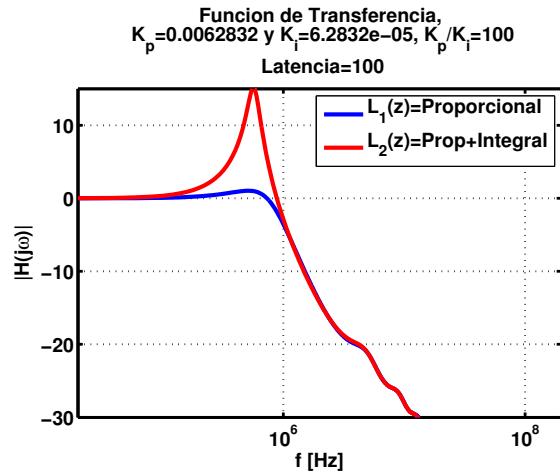
(a)  $D = 100$ ,  $K_p = 0,0094248$ ,  $K_p/K_i = 100$ .



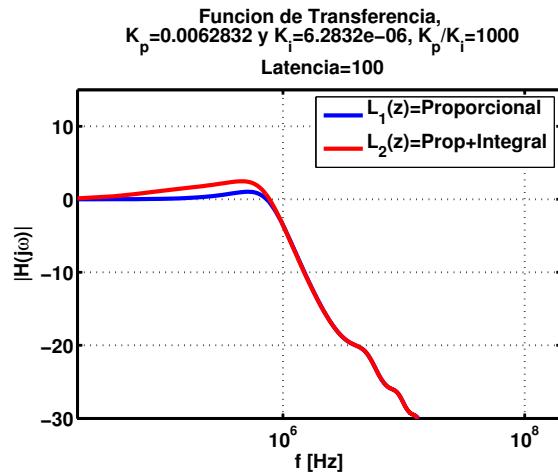
(b)  $D = 100$ ,  $K_p = 0,0094248$ ,  $K_p/K_i = 1000$ .



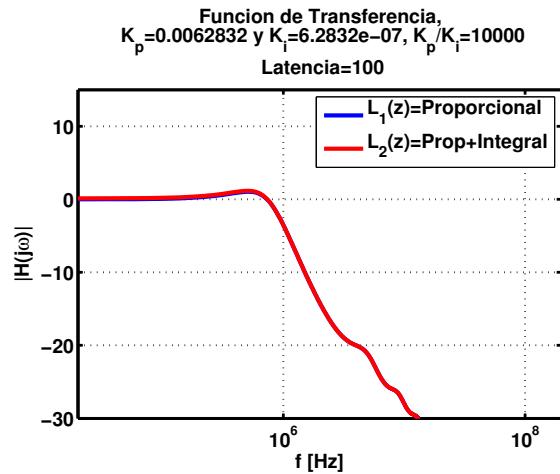
(c)  $D = 100$ ,  $K_p = 0,0094248$ ,  $K_p/K_i = 10000$ .



(d)  $D = 100$ ,  $K_p = 0,0062832$ ,  $K_p/K_i = 100$ .



(e)  $D = 100$ ,  $K_p = 0,0062832$ ,  $K_p/K_i = 1000$ .



(f)  $D = 100$ ,  $K_p = 0,0062832$ ,  $K_p/K_i = 10000$ .

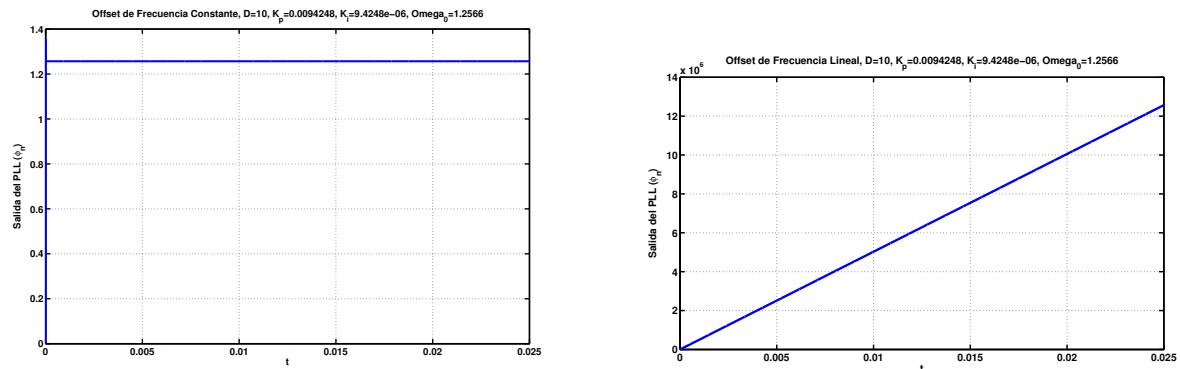
Figura 3.11: Función de Transferencia del PLL,  $D = 100$  y distintos valores de  $K_p$  y  $K_i$ .

Los parámetros  $K_p$  y  $K_i$  afectan la forma de las curvas de transferencia y la dotan de diferentes comportamientos. Por ejemplo, una zona útil de mayor rango, efectos indeseados de peaking (amplificación de la señal), atenuación de la señal, disminución del rango de trabajo, etc. Además, como vimos anteriormente, el aumento de la latencia produce un aumento en el nivel de peaking del sistema debido a que el mismo se vuelve más propenso a las inestabilidades, algo que queda claro si comparamos los resultados de las Figuras (3.9), (3.10) y (3.11), donde los valores de  $K_p$  y  $K_i$  son los mismos pero cambia la latencia  $D$ .

### Salidas PLL

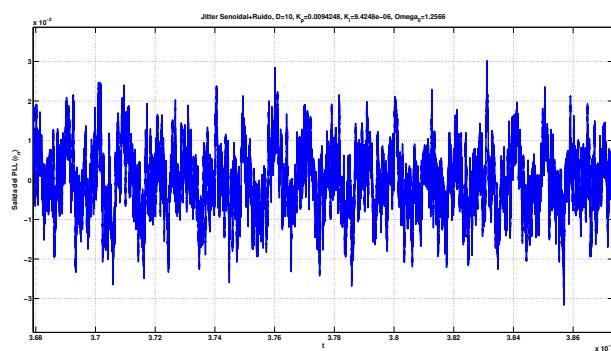
A continuación se analizan las salidas del PLL, para diferentes entradas (error de fase constante, error de fase lineal y jitter senoidal con ruido), junto con las señales intermedias del sistema: error de fase, salida de la parte proporcional y salida de la parte integral.

En la Figura (3.12), se muestra la salida  $\phi[n]$  del PLL, para distintas entradas: offset de frecuencia constante, offset de frecuencia lineal y jitter senoidal con ruido. Lo que se espera a la salida es que el PLL pueda seguir a las diferentes entradas.



(a) Salida para entrada de offset de freq. constante.

(b) Salida para entrada de offset de freq. lineal.

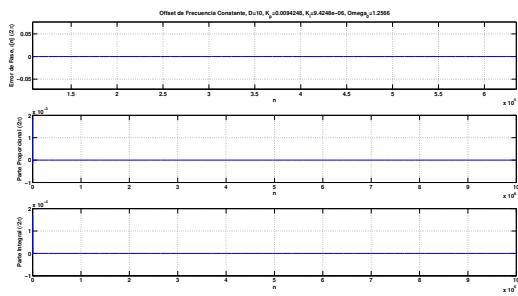


(c) Salida para entrada de jitter más ruido.

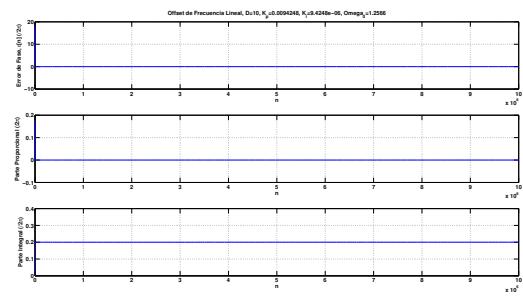
Figura 3.12: Salidas  $\phi[n]$ , para diferentes entradas  $\theta[n]$ .

Luego, en Figura (3.13), se muestran las señales intermedias: Error de Fase, Salida Proporcional e Integral, para las entradas de offset de frecuencia constante, lineal y jitter más ruido.

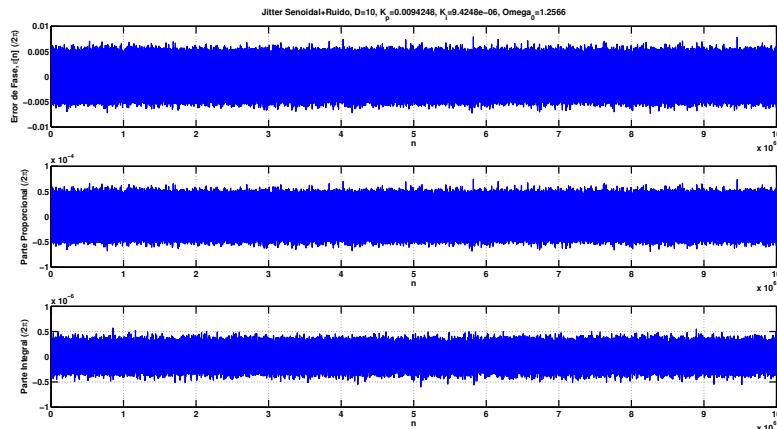
La señal de error de fase corresponde a la señal diferencia entre la entrada y la salida del PLL, esto es:  $\epsilon[n] = \theta[n] - \phi[n]$ . Además, como el PLL posee un filtro de lazo cuya respuesta se encuentra formada por una parte proporcional más una integral, se puede analizar el mismo, como dos filtros de lazo cuyas salidas se suman para obtener la señal de control del VCO  $c[n]$ , justamente en la Figura (3.13), se muestran éstas dos señales.



(a) Offset de Frecuencia Constante.



(b) Offset de Frecuencia Lineal.

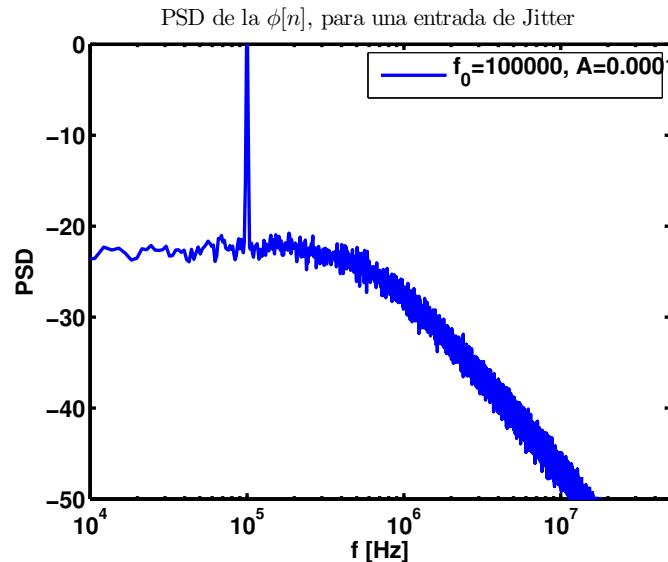


(c) Jitter Senoidal más Ruido.

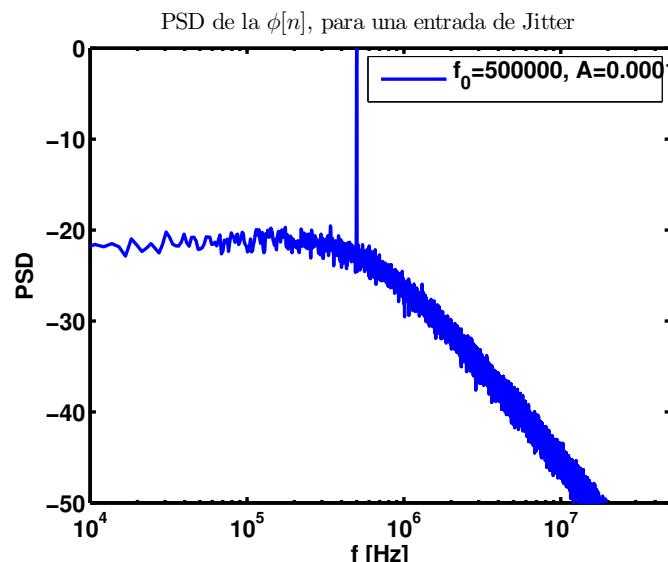
Figura 3.13: Señales Intermedias: Error de Fase, Salida Parte Proporcional y Salida Parte Integral.

PSD de Salida para Entradas de Jitter

A continuación en la Figura (3.14), se muestran las densidades espectrales de potencia (PSD) de la salida del PLL, para dos entradas de Jitter con ruido y diferente frecuencia de oscilación. En dichas PSDs se puede apreciar la presencia del tono de jitter en la posición correspondiente a su frecuencia de oscilación ( $f_0$ ).



(a) Tono de Jitter en  $f_0 = 0,1 \times 10^6$  Hz.



(b) Tono de Jitter en  $f_0 = 0,5 \times 10^6$  Hz.

Figura 3.14: Presencia de tono de Jitter en la PSD de la Salida del PPL.

Funciones de Transferencia de Jitter
--------------------------------------

Existe una forma de medir el rendimiento de un PLL en cuanto a su capacidad de realizar el seguimiento del jitter de entrada. Esta medición se denomina curva de Transferencia de Jitter ó Jitter Transfer en inglés. En este trabajo, se realizaron dichas curvas mediante simulación, sin embargo en la vida real existen equipos físicos que pueden realizar ésta medición en PLLs reales.

Para elaborar dichas curvas, se coloca como entrada del recuperador de fase una portadora con frecuencia nula y se configura el VCO con una frecuencia central también nula. Luego, a la portadora se le adiciona una fase senoidal con una frecuencia determinada, la cual se irá barriendo para conocer la respuesta del PLL en los diferentes casos. Seguidamente, se mide la amplitud de la entrada y de salida de fase del PLL y se realiza el cociente. De esta manera, se elabora punto a punto (ó frecuencia a frecuencia) la respuesta en frecuencia.

Las Figuras (3.15) a (3.17), se obtuvieron a partir del modelo teórico ó modelo equivalente de fase del PLL, para distintos valores de latencia y para distintas relaciones de  $K_p/K_i$ . En el Anexo B.5, se muestra el código de MATLAB utilizado, en este caso el código no fue provisto por el Dr. Hueda, sino que se trata de un código propio.

Como se puede ver en las diferentes curvas, el hecho de aumentar la latencia aumenta el peaking y el sistema se vuelve más propenso a inestabilidades (como ya ha sido mencionado anteriormente), además al aumentar el peaking disminuye el ancho de banda del sistema. También se destaca que la inestabilidad comienza a hacerse más notoria a medida que disminuye la relación  $K_p/K_i$ .

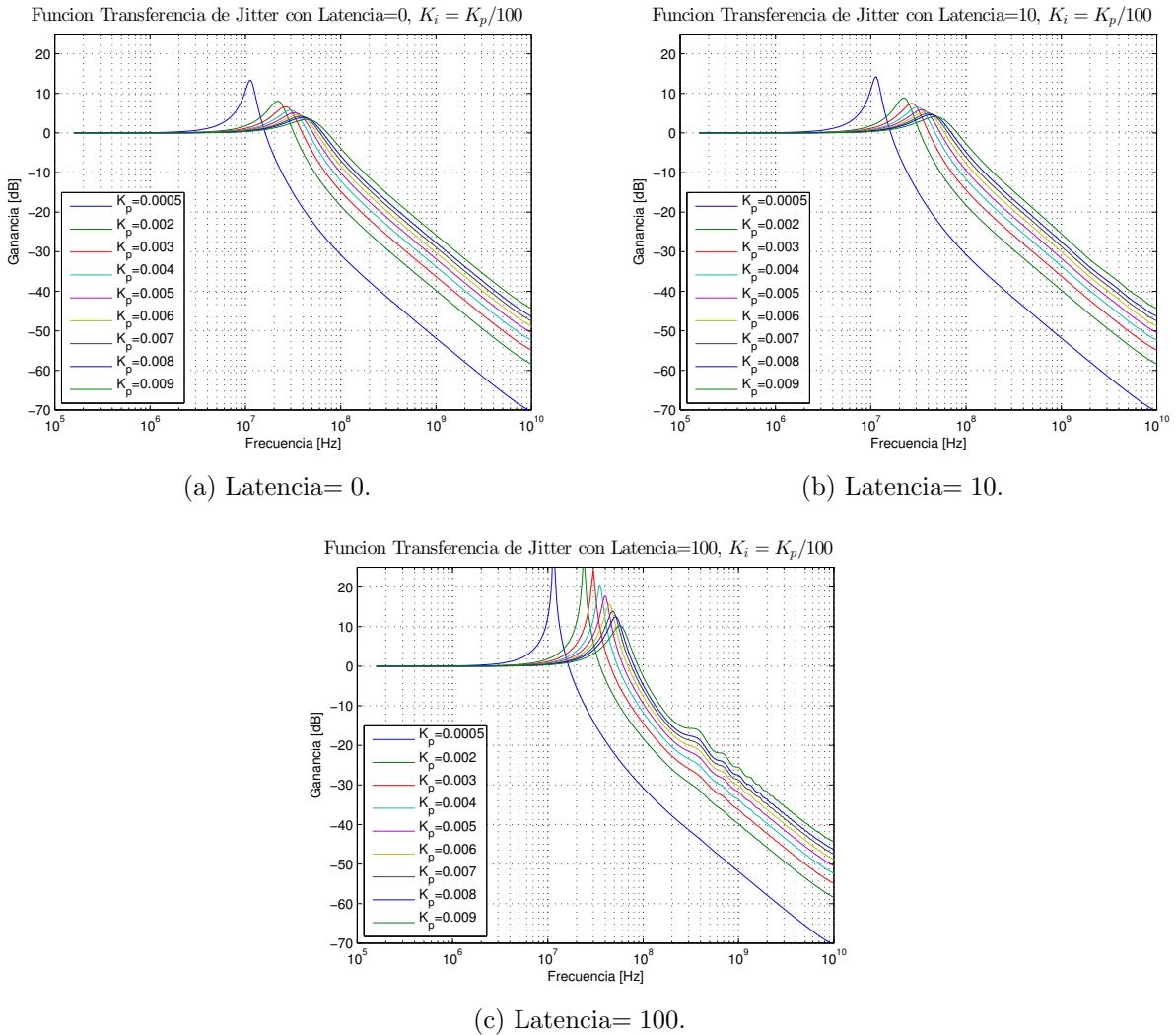


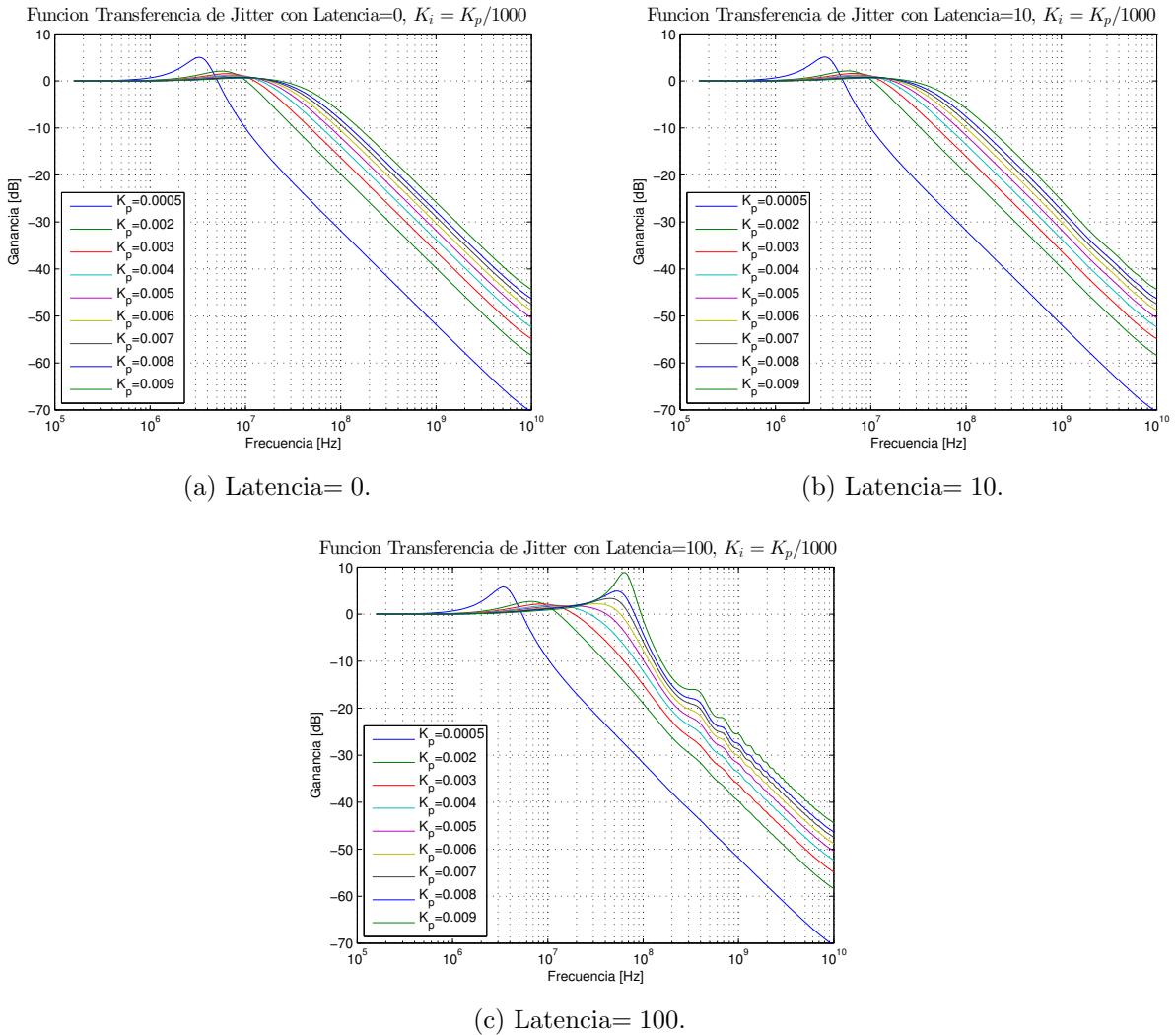
Figura 3.15: Curvas de Jitter Transfer con  $K_p/K_i = 100$ .

A continuación se muestra la condición de estabilidad de las distintas configuraciones analizadas en las curvas de transferencia de jitter. La aparición de peaking se puede entender como la presencia cada vez más cercana de un polo de la función de transferencia de lazo cerrado, al círculo unitario. A medida que el sistema tiende más y más a la inestabilidad, éste peaking crece, además aún cuando el peaking se encuentre controlado, su mera presencia disminuye el ancho de banda del PLL, ya que las frecuencias donde existe peaking, se considera una banda no utilizable, debido a la amplificación ó ganancia del jitter en esa banda de frecuencias.

#### Relación $K_p/K_i = 100$ :

Con Latencia=0,  $K_p=5.000000e-04$  y  $K_i=5.000000e-06$ , el sistema es: ESTABLE

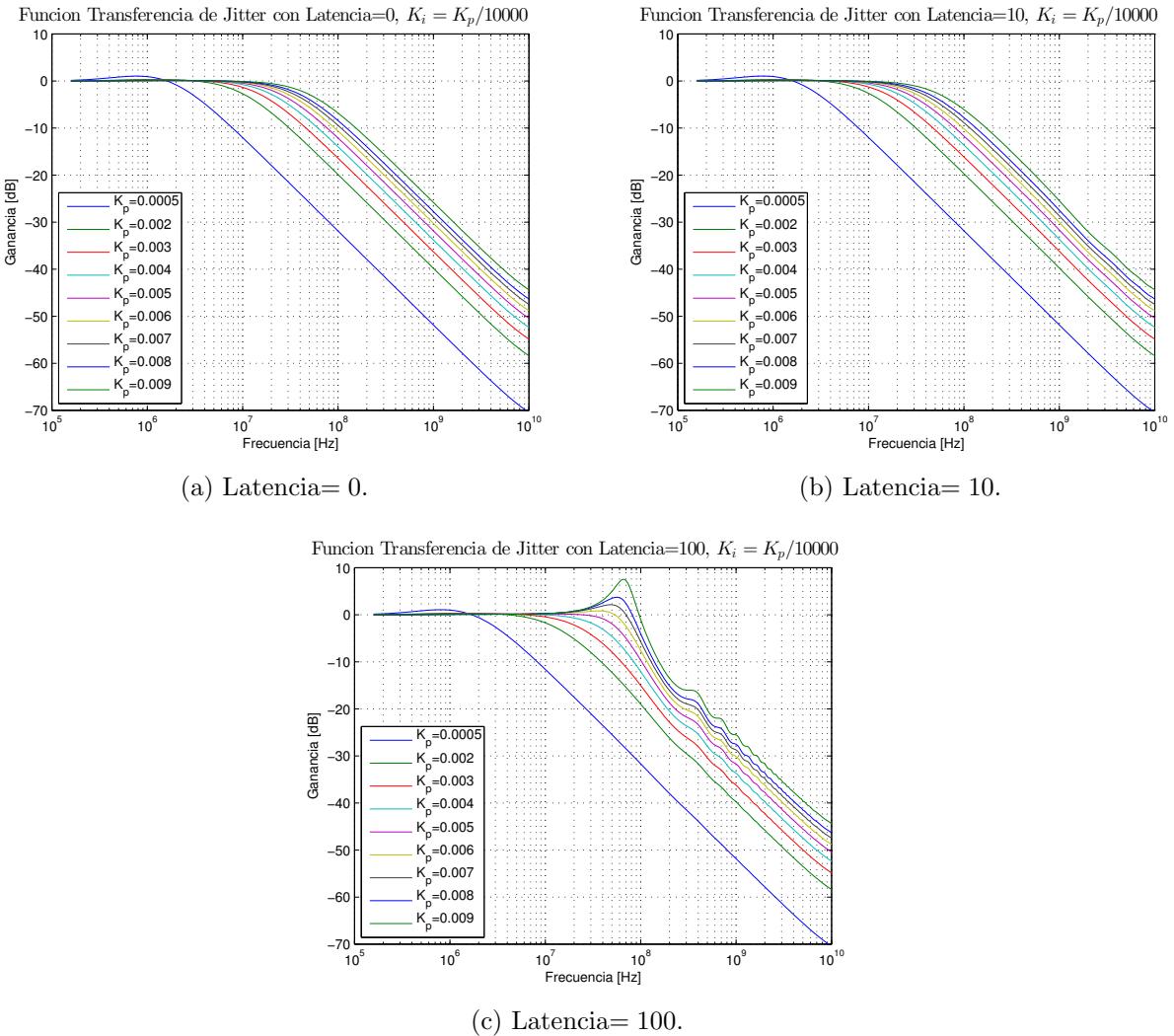
Con Latencia=0,  $K_p=2.000000e-03$  y  $K_i=2.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=0,  $K_p=3.000000e-03$  y  $K_i=3.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=0,  $K_p=4.000000e-03$  y  $K_i=4.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=0,  $K_p=5.000000e-03$  y  $K_i=5.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=0,  $K_p=6.000000e-03$  y  $K_i=6.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=0,  $K_p=7.000000e-03$  y  $K_i=7.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=0,  $K_p=8.000000e-03$  y  $K_i=8.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=0,  $K_p=1.000000e-02$  y  $K_i=1.000000e-04$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=5.000000e-04$  y  $K_i=5.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=2.000000e-03$  y  $K_i=2.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=3.000000e-03$  y  $K_i=3.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=4.000000e-03$  y  $K_i=4.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=5.000000e-03$  y  $K_i=5.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=6.000000e-03$  y  $K_i=6.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=7.000000e-03$  y  $K_i=7.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=8.000000e-03$  y  $K_i=8.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=1.000000e-02$  y  $K_i=1.000000e-04$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=5.000000e-04$  y  $K_i=5.000000e-06$ , el sistema es: INESTABLE  
Con Latencia=100,  $K_p=2.000000e-03$  y  $K_i=2.000000e-05$ , el sistema es: INESTABLE  
Con Latencia=100,  $K_p=3.000000e-03$  y  $K_i=3.000000e-05$ , el sistema es: INESTABLE  
Con Latencia=100,  $K_p=4.000000e-03$  y  $K_i=4.000000e-05$ , el sistema es: INESTABLE  
Con Latencia=100,  $K_p=5.000000e-03$  y  $K_i=5.000000e-05$ , el sistema es: INESTABLE  
Con Latencia=100,  $K_p=6.000000e-03$  y  $K_i=6.000000e-05$ , el sistema es: INESTABLE  
Con Latencia=100,  $K_p=7.000000e-03$  y  $K_i=7.000000e-05$ , el sistema es: INESTABLE  
Con Latencia=100,  $K_p=8.000000e-03$  y  $K_i=8.000000e-05$ , el sistema es: INESTABLE  
Con Latencia=100,  $K_p=1.000000e-02$  y  $K_i=1.000000e-04$ , el sistema es: INESTABLE


 Figura 3.16: Curvas de Jitter Transfer con  $K_p/K_i = 1000$ .

### Relación $K_p/K_i = 1000$ :

Con Latencia=0,  $K_p=5.000000e-04$  y  $Ki=5.000000e-07$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=2.000000e-03$  y  $Ki=2.000000e-06$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=3.000000e-03$  y  $Ki=3.000000e-06$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=4.000000e-03$  y  $Ki=4.000000e-06$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=5.000000e-03$  y  $Ki=5.000000e-06$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=6.000000e-03$  y  $Ki=6.000000e-06$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=7.000000e-03$  y  $Ki=7.000000e-06$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=8.000000e-03$  y  $Ki=8.000000e-06$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=1.000000e-02$  y  $Ki=1.000000e-05$ , el sistema es: ESTABLE  
 Con Latencia=10,  $Kp=5.000000e-04$  y  $Ki=5.000000e-07$ , el sistema es: ESTABLE

Con Latencia=10,  $K_p=2.000000e-03$  y  $K_i=2.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=3.000000e-03$  y  $K_i=3.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=4.000000e-03$  y  $K_i=4.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=5.000000e-03$  y  $K_i=5.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=6.000000e-03$  y  $K_i=6.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=7.000000e-03$  y  $K_i=7.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=8.000000e-03$  y  $K_i=8.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=1.000000e-02$  y  $K_i=1.000000e-05$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=5.000000e-04$  y  $K_i=5.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=2.000000e-03$  y  $K_i=2.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=3.000000e-03$  y  $K_i=3.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=4.000000e-03$  y  $K_i=4.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=5.000000e-03$  y  $K_i=5.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=6.000000e-03$  y  $K_i=6.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=7.000000e-03$  y  $K_i=7.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=8.000000e-03$  y  $K_i=8.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=1.000000e-02$  y  $K_i=1.000000e-05$ , el sistema es: ESTABLE


 Figura 3.17: Curvas de Jitter Transfer con  $K_p/K_i = 10000$ .

### Relación $K_p/K_i = 10000$ :

Con Latencia=0,  $K_p=5.000000e-04$  y  $Ki=5.000000e-08$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=2.000000e-03$  y  $Ki=2.000000e-07$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=3.000000e-03$  y  $Ki=3.000000e-07$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=4.000000e-03$  y  $Ki=4.000000e-07$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=5.000000e-03$  y  $Ki=5.000000e-07$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=6.000000e-03$  y  $Ki=6.000000e-07$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=7.000000e-03$  y  $Ki=7.000000e-07$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=8.000000e-03$  y  $Ki=8.000000e-07$ , el sistema es: ESTABLE  
 Con Latencia=0,  $Kp=1.000000e-02$  y  $Ki=1.000000e-06$ , el sistema es: ESTABLE  
 Con Latencia=10,  $Kp=5.000000e-04$  y  $Ki=5.000000e-08$ , el sistema es: ESTABLE

Con Latencia=10,  $K_p=2.000000e-03$  y  $K_i=2.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=3.000000e-03$  y  $K_i=3.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=4.000000e-03$  y  $K_i=4.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=5.000000e-03$  y  $K_i=5.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=6.000000e-03$  y  $K_i=6.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=7.000000e-03$  y  $K_i=7.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=8.000000e-03$  y  $K_i=8.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=10,  $K_p=1.000000e-02$  y  $K_i=1.000000e-06$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=5.000000e-04$  y  $K_i=5.000000e-08$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=2.000000e-03$  y  $K_i=2.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=3.000000e-03$  y  $K_i=3.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=4.000000e-03$  y  $K_i=4.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=5.000000e-03$  y  $K_i=5.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=6.000000e-03$  y  $K_i=6.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=7.000000e-03$  y  $K_i=7.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=8.000000e-03$  y  $K_i=8.000000e-07$ , el sistema es: ESTABLE  
Con Latencia=100,  $K_p=1.000000e-02$  y  $K_i=1.000000e-06$ , el sistema es: ESTABLE

## Capítulo 4

# Procesamiento Tiempo Discreto-Continuo

### 4.1. Descripción y Consignas

Considere el sistema de la Figura (4.1), el cual corresponde al diagrama de bloques de un módem.

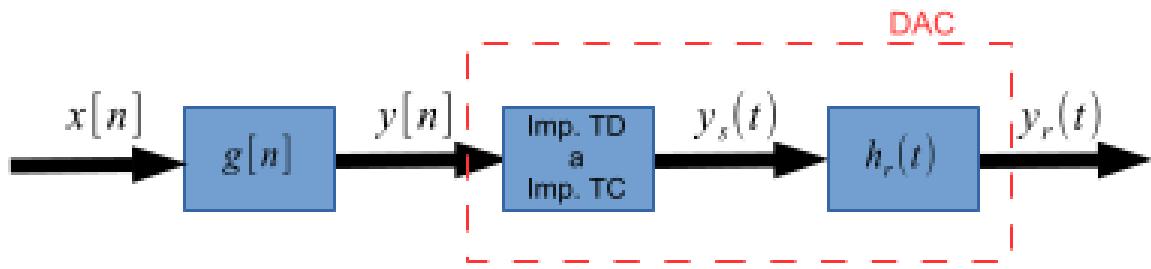


Figura 4.1: Diagrama de Bloques de un Módem.

Donde  $g[n] = g(nT)$  con:

$$g(t) = \text{sinc}(t/T) \frac{\cos(\pi\beta t/T)}{1 - 4\beta^2 t^2/T^2}$$

Se propone desarrollar las siguientes actividades:

1. Mostrar que:

$$y_r(t) = y_s(t) * h_r(t) \xleftrightarrow{\mathbb{F}} Y_r(j\omega) = H_r(j\omega)Y(e^{j\omega T_s})$$

2. Graficar y analizar el espectro a la salida del transmisor, para diferentes condiciones de funcionamiento:

- Distintos valores de sobremuestreo,  $M = T/T_s$ .
- Distintos filtros reconstructores: Coseno Realzado, Pulso Rectangular, etc.

## 4.2. Resultados

Si observamos la Figura (4.1), la señal  $y[n]$  se obtiene como resultado de la convolución entre la señal de entrada  $x[n]$  y el filtro discreto  $g[n]$ , tal que:

$$y[n] = x[n] * g[n]$$

Por otra parte, la señal en tiempo continuo  $y_s(t)$ , en donde los impulsos discretos se han reemplazado por impulsos continuos ó tambien conocidos como deltas de Dirac, se expresa de la siguiente manera:

$$y_s(t) = \sum_{n=-\infty}^{\infty} y[n]\delta(t - nT_s)$$

Finalmente, la señal continua reconstruida  $y_r(t)$ , se obtiene por la convolución entre la señal en tiempo continuo  $y_s(t)$  y el filtro reconstructor  $h_r(t)$ :

$$\begin{aligned} y_r(t) &= y_s(t) * h_r(t) \\ &= \left( \sum_{n=-\infty}^{\infty} y[n]\delta(t - nT_s) \right) * h_r(t) \\ &= \sum_{n=-\infty}^{\infty} y[n]h_r(t - nT_s) \end{aligned}$$

Observando los resultados obtenidos anteriormente, uno puede desprender una rápida interpretación de los mismos, la cual se puede expresar de la siguiente manera: La señal continua reconstruida a la salida del sistema  $y_r(t)$ , es una señal cuyos valores en los instantes  $t = nT_s$ , corresponden únicamente con los valores de la secuencia  $y[n]$  y los valores entre

dichos tiempos corresponden a una interpolación de los puntos en  $t = nT_s$ , mediante el filtro interpolador  $h_r(t)$ .

Si tomamos la transformada de Fourier de  $y_r(t)$ , se llega a la siguiente expresión (por propiedad de selección):

$$y_r(t) = y_s(t) * h_r(t) \xrightarrow{TF} Y_r(j\omega) = H_r(j\omega) \sum_{n=-\infty}^{\infty} y[n]e^{-j\omega Tn}$$

Por lo tanto:

$$\begin{aligned} Y_r(j\omega) &= H_r(j\omega) \sum_{n=-\infty}^{\infty} y[n]e^{-j\omega Tn} \\ &= H_r(j\omega)Y(e^{j\omega T_s}) \\ &= H_r(j\omega)G(e^{j\omega T_s})X(e^{j\omega T_s}) \end{aligned}$$

Como vemos, hemos llegado a una expresión híbrida con transformadas de Fourier de señales en tiempo continuo y tiempo discreto en una misma expresión. Si uno desea hallar un paralelismo con la vida real, la expresión anterior podría entenderse a grandes rasgos, como un Conversor Digital-Analógico (obviando todos los detalles de implementación física) donde entra una secuencia digital de datos (bits) y sale una señal analógica, como por ejemplo cuando reproducimos una canción en nuestra PC y la escuchamos por los parlantes de la misma, la verdad... increíble!

A continuación, se analizan dos casos, el primero usando una tasa de sobremuestreo de  $M = 4$  y luego de  $M = 8$ . Además, se muestran los distintos filtros reconstructores utilizados, junto con sus resultados correspondientes. Comenzamos con el caso de  $M = 4$ , donde el pulso conformador  $g[n]$ , es un coseno realizado como los utilizados en laboratorios previos, donde el valor de  $\beta = 0,2$  (dicho valor se mantiene fijo para  $M = 4$  y  $M = 8$ ). En la Figura (4.6), podemos observar el pulso  $g[n]$  con  $M = 4$ .

CASO 1:  $M = 4$

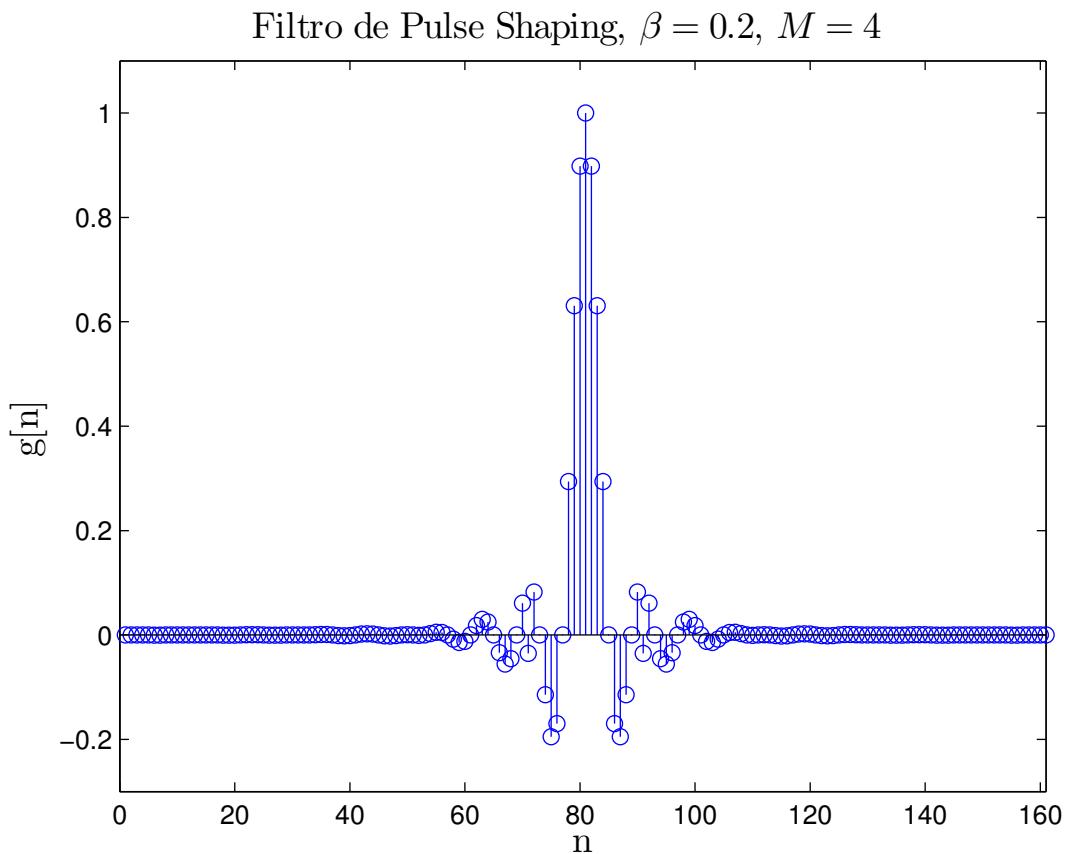


Figura 4.2: Pulso Coseno Realzado  $g[n]$  con  $\beta = 0,2$ ,  $M = 4$ .

Para obtener la señal reconstruida  $y_r(t)$ , se utilizaron diferentes filtros de reconstrucción, los cuales se listan a continuación:

1. Coseno Realzado con  $\beta = 0,8$ .
2. Coseno Realzado con  $\beta = 0,1$ .
3. Pulso Rectangular.
4. Pulso Seno Cardinal.
5. Filtro de Butterworth de órden= 10.
6. Filtro de Butterworth recortado por una ventana rectangular.

En la Figura (4.3), podemos observar un pulso coseno realizado con factor de roll-off de 0,8, las colas se atenuaron bastante debido a valor alto de  $\beta$ . En la subfigura de la izquierda se muestra el pulso en el dominio del tiempo y a la derecha, en el dominio de la frecuencia.

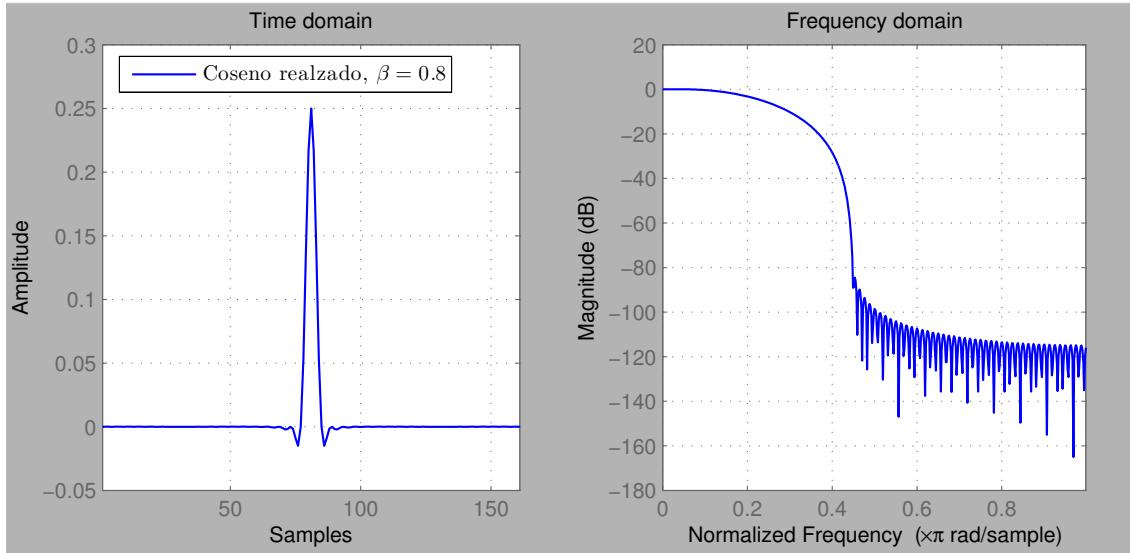


Figura 4.3: Filtro Reconstructor, Coseno realzado con  $\beta = 0,8$ ,  $M = 4$ .

En la Figura (4.4), observamos un pulso coseno realzado con factor de roll-off de 0,1, en este caso las colas presentan una longitud mayor que en el caso anterior, debido al bajo valor de  $\beta$ . En la subfigura de la izquierda se muestra el pulso en el dominio del tiempo y a la derecha, en el dominio de la frecuencia. Si comparamos los espectros de las Figuras (4.3) y (4.4), podemos ver el mayor exceso de ancho de banda en el caso de  $\beta = 0,8$  (Figura 4.3). Además, en el dominio del tiempo se puede observar como un mayor exceso de ancho de banda produce una disminución en la longitud de las colas del pulso.

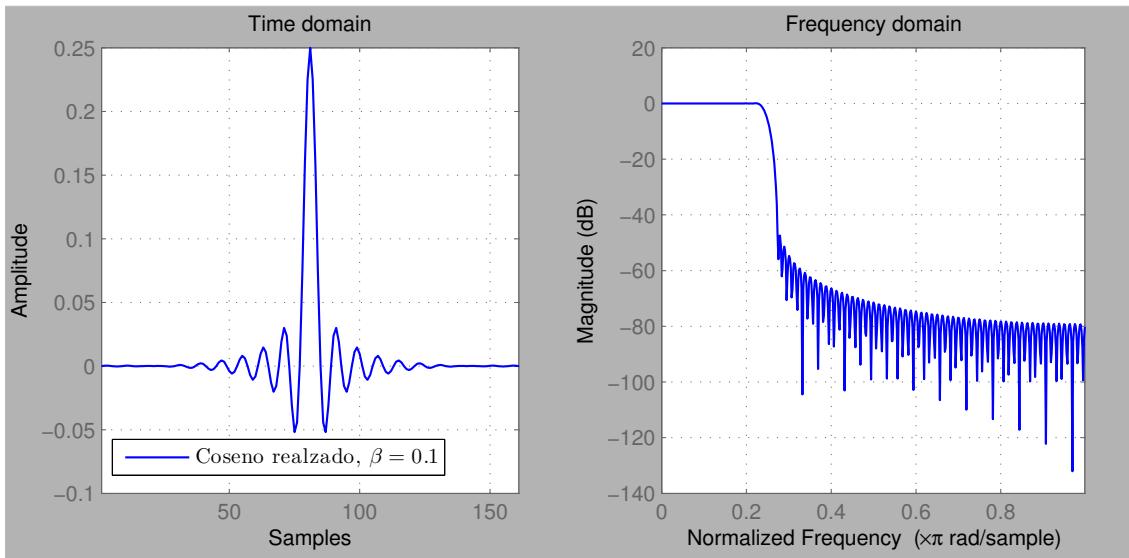


Figura 4.4: Filtro Reconstructor, Coseno realzado con  $\beta = 0,1$ ,  $M = 4$ .

En la Figura (4.5), vemos un pulso rectangular cuyo espectro corresponde a un seno cardinal. Los lóbulos secundarios de dicho filtro poseen magnitudes importantes, por lo cual el pulso rectangular no es el mejor filtro pasa bajos.

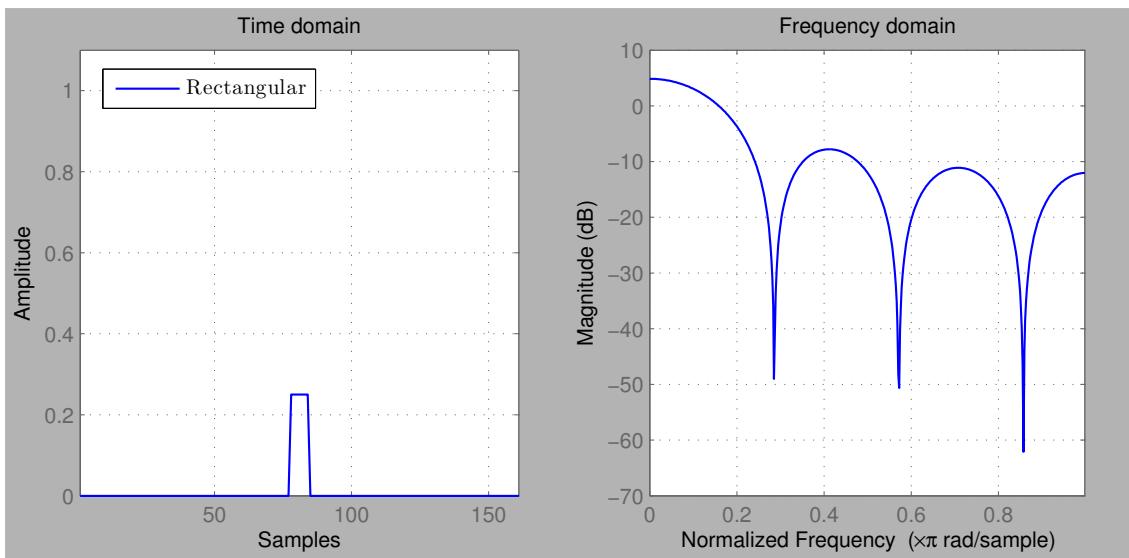


Figura 4.5: Filtro Reconstructor, Pulso Rectangular,  $M = 4$ .

En la Figura (4.6), vemos un pulso seno cardinal. Dicho pulso es de longitud infinita (no

implementable físicamente) y su espectro corresponde a un pulso rectangular, con el menor ancho de banda posible, es más, se puede ver como un pulso coseno realizado con  $\beta = 0$ .

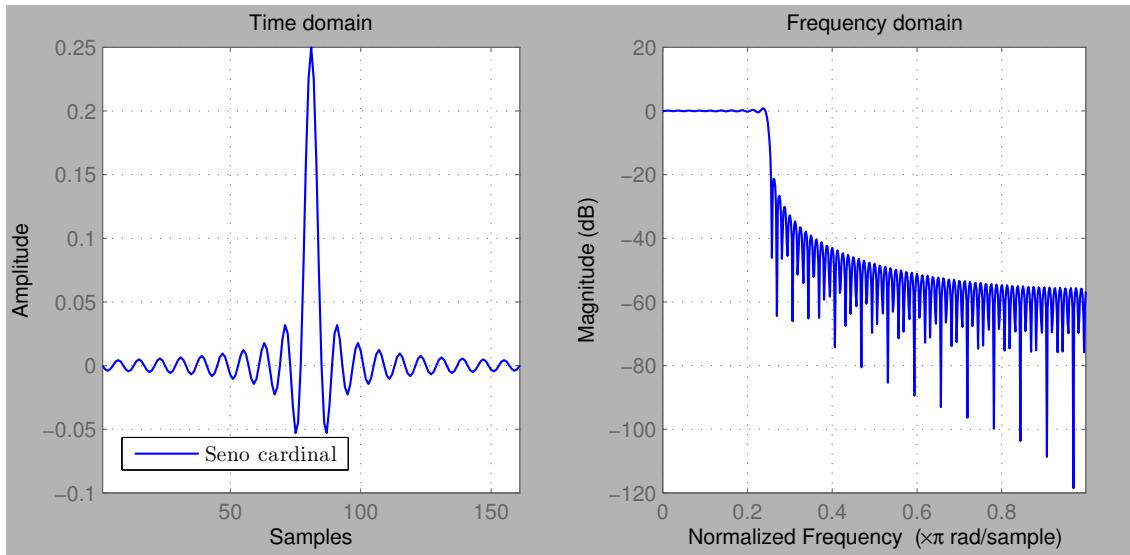


Figura 4.6: Filtro Reconstructor, Seno Cardinal,  $M = 4$ .

En la Figura (4.7), se observa un filtro de Butterworth de orden= 10. Este filtro presenta una respuesta plana en la banda pasante (una de las características de este tipo de filtros). Además, las bandas pasante y rechazada se diferencian claramente entre sí, aunque la transición entre dichas bandas es lenta y lineal, más precisamente la pendiente de la caída es de  $20n$  dB por década, con  $n$  es la cantidad de polos del filtro ó el orden.

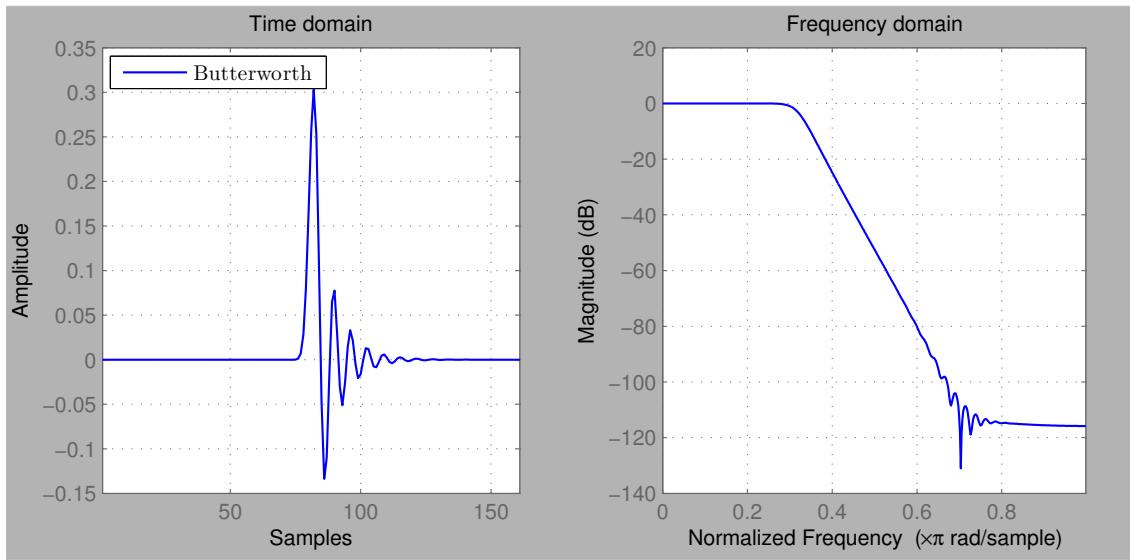


Figura 4.7: Filtro Reconstructor, Butterworth órden= 10,  $M = 4$ .

En la Figura (4.8), vemos el mismo filtro de Butterworth utilizado anteriormente, pero donde su respuesta temporal se ha recortado por una ventana rectangular de ancho  $T_s$ . Se observa que dicho enventanado produce grandes modificaciones en su espectro, con una gran disminución en el ancho de la banda pasante como característica más importante.

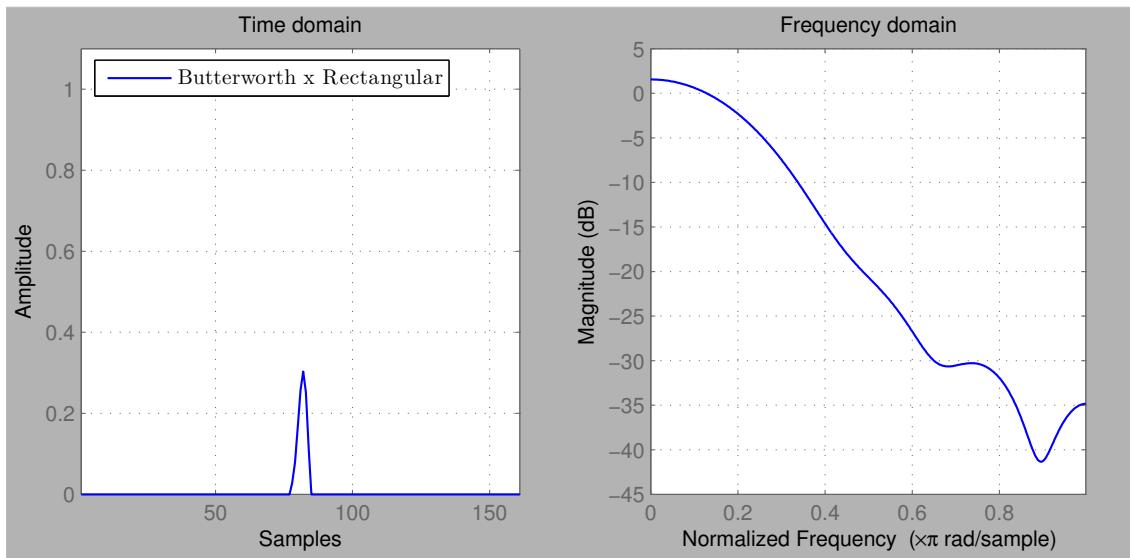


Figura 4.8: Filtro Reconstructor, Butterworth órden= 10 con ventana Rectangular,  $M = 4$ .

Finalmente y para facilitar la comparación entre los filtros presentados anteriormente, en la Figura (4.9), se muestran las respuestas temporales y el espectro de todos los filtros superpuestos en la misma gráfica.

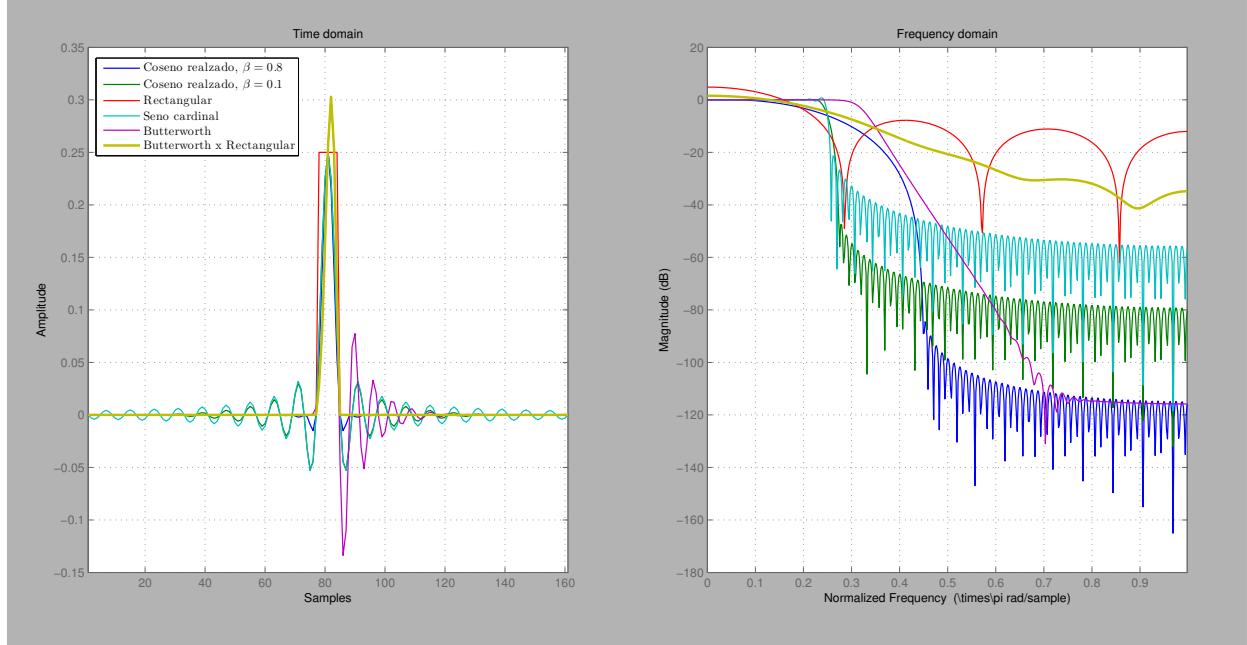


Figura 4.9: Todos los Filtros Reconstructores en una misma gráfica,  $M = 4$ .

Los filtros reconstructores permiten recuperar  $y_r(t)$  a partir de la señal  $y_s(t)$ , mediante la interpolación de los valores de  $y_s(t)$ . Para poder observar el efecto sobre el espectro de la señal  $y_s(t)$  producido por cada filtro utilizado, en la Figura (4.10), se muestra el espectro de la señal antes de la reconstrucción y después de la reconstrucción. Como vimos anteriormente, los espectros de cada uno de los filtros reconstructores poseen diferentes características, lo que produce diferentes espectros filtrados resultantes.

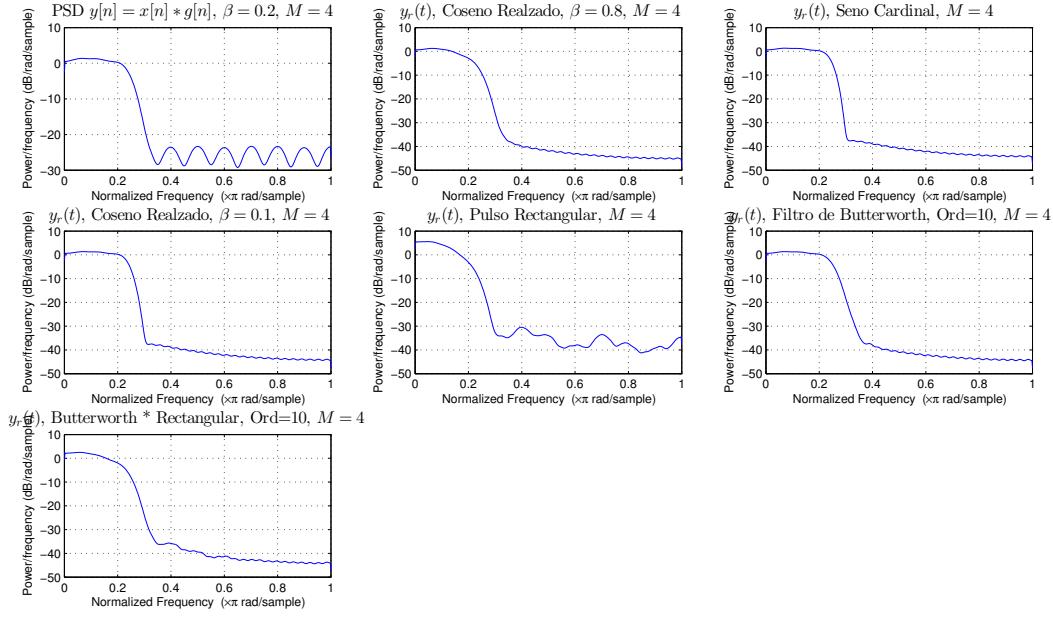


Figura 4.10: PSD de la Salida para los distintos filtros de reconstrucción,  $M = 4$ .

Los filtros se difieren en varios aspectos, por ejemplo: ripple en la banda rechazada, ancho de banda de la banda pasante, transición entre bandas, entre otros.

Los pulsos coseno realizados producen una espectro de salida parecido, en donde la principal diferencia se encuentra en el ancho de la banda pasante, mayor en el caso de mayor  $\beta$ . La respuesta producida por el pulso seno cardinal es bastante parecida al coseno realizado con  $\beta = 0.1$ , aunque el ancho de la banda pasante con el seno cardinal es levemente menor. Luego, el pulso rectangular debido a su pobres características espectrales, produce un señal de salida en donde el ripple de la banda rechazada no se ha atenuado en gran medida, además el ancho de la banda pasante ha disminuido, en concordancia con el espectro del filtro rectangular. El filtro de Butterworth produce buenos resultados, aunque la banda de transición es mayor que en el caso de los pulsos cosenoidales, luego ancho de banda pasante es muy similar y el ripple de la banda rechazada se ha atenuado como con los pulsos cosenoidales. Finalmente, el filtro de Butterworth recortado presenta unos resultados levemente mejores que el pulso rectangular en donde el ripple es menor que para su par rectangular, sin embargo el ancho de la banda pasante es menor que para el caso del filtro de Butterworth sin recortado.

Finalmente, se realiza una comparación de la señal  $y[n]$  con sus distintas versiones continuas  $y_r(t)$ . La idea es poder observar la equivalencia entre las distintas señales analógicas transmitidas y su versión original digital.

Como vimos en el análisis de los espectros de salida para cada uno de los filtros reconstructores, es esperable que los filtros rectangular y Butterworth recortado presenten una menor performance de recuperación, principalmente por sus características espetrales (banda pasante, banda de rechazo, banda de transición, etc) ya analizadas. Por otra parte, el pulso seno cardinal no se puede implementar físicamente y sólo tiene un valor analítico y/o académico, en conclusión los filtros coseno realizados son una muy buena alternativa de filtros reconstructores, junto el clásico filtro de Butterworth y estudiado en la temática de filtros analógicos.

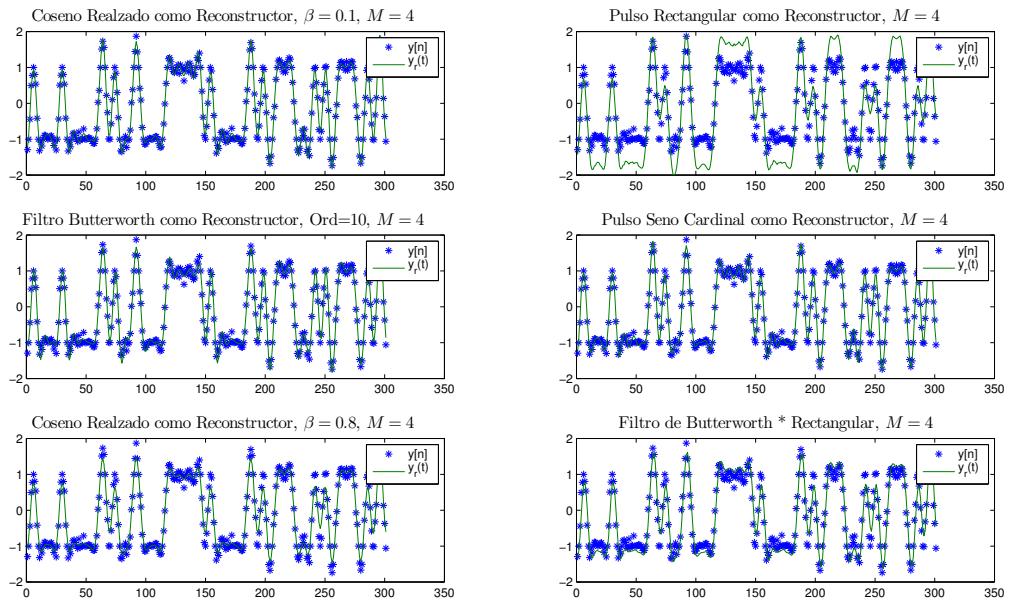


Figura 4.11: Señal  $y[n]$  vs. Salidas de los Filtros de Reconstrucción,  $M = 4$ .

CASO 2:  $M = 8$

En este segundo caso, se realizaron las mismas simulaciones que para el Caso 1, pero modificando el factor de sobremuestreo a  $M = 8$ . En la Figura (4.12) vemos el filtro coseno realizado utilizado como filtro conformador de pulsos.

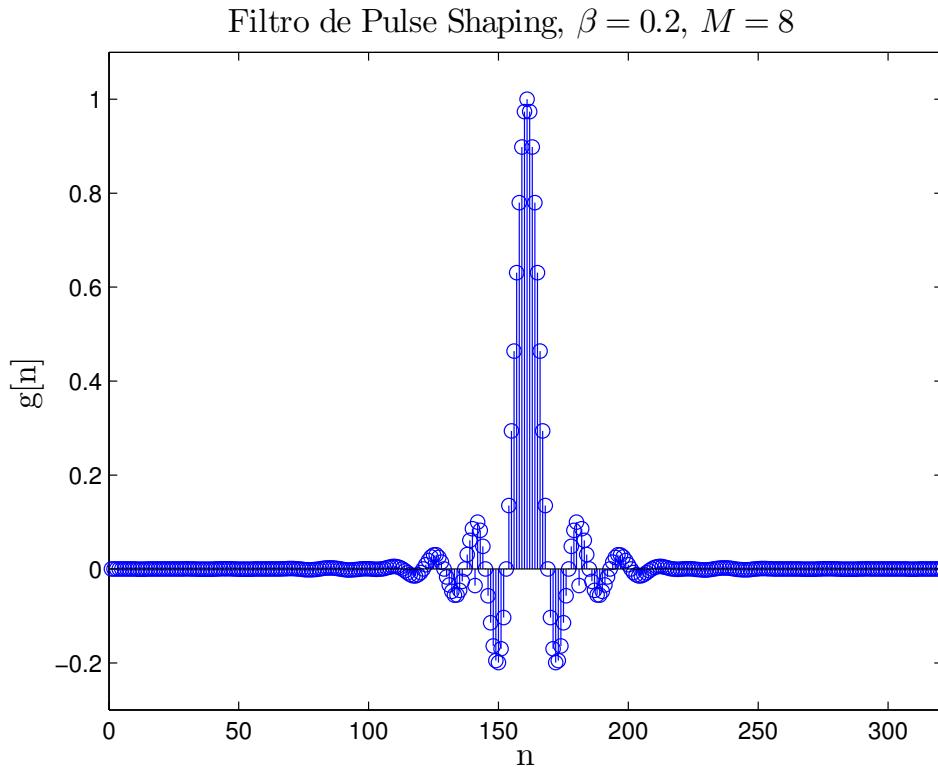


Figura 4.12: Pulso Coseno Realizado,  $g[n]$ ,  $\beta = 0.2$ ,  $M = 8$ .

En el caso de los filtros coseno realizado (Figuras 4.13 y 4.14), pulso rectangular (Figura 4.15) y el pulso seno cardinal (Figura 4.16) vemos principalmente una reducción en el ancho de la banda pasante, debido al aumento en el factor  $M$ . Luego, en el filtro Butterworth (Figura 4.17) observamos una banda de transición con menor pendiente. Finalmente, en el filtro de Butterworth recortado (Figura 4.18), vemos un sobrepasamiento en la ganancia de la parte superior de la banda pasante y una banda de transición con menor pendiente.

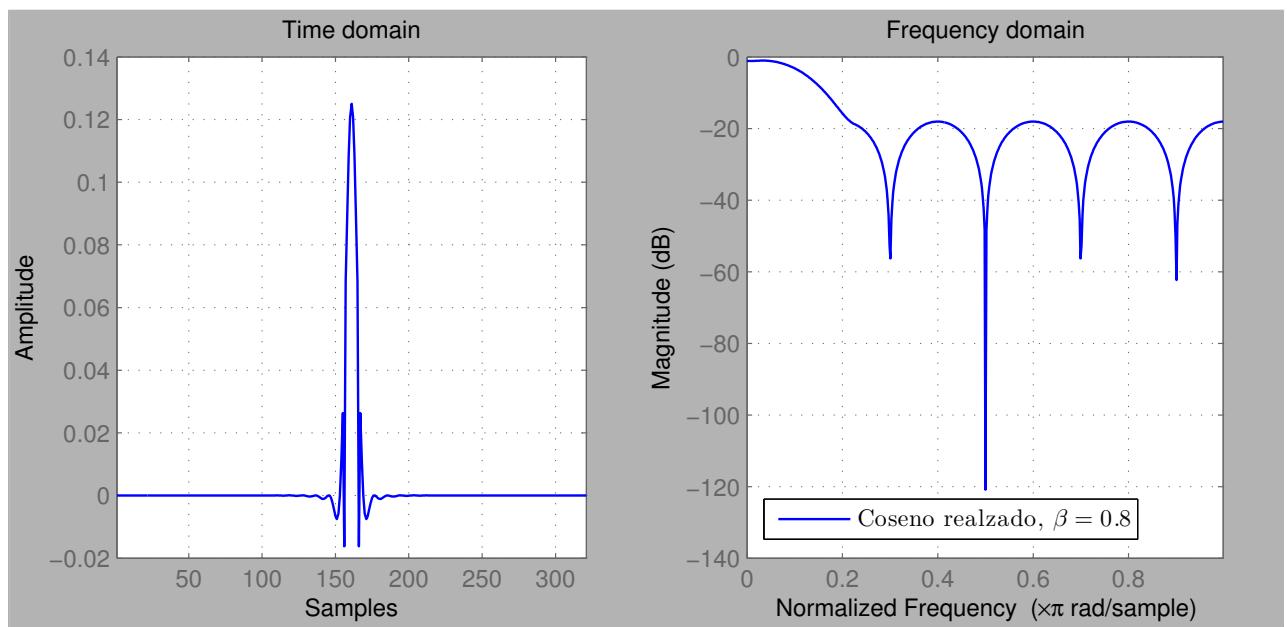


Figura 4.13: Filtro Reconstructor, Coseno realzado con  $\beta = 0.8$ ,  $M = 8$ .

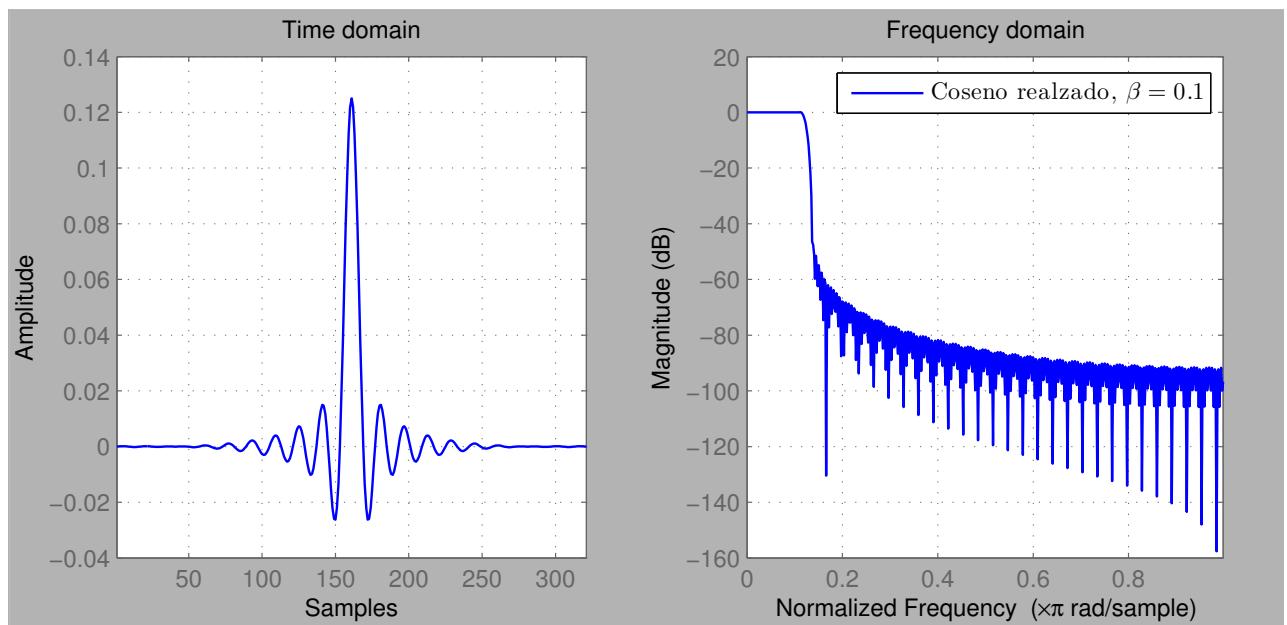


Figura 4.14: Filtro Reconstructor, Coseno realzado con  $\beta = 0.1$ ,  $M = 8$ .

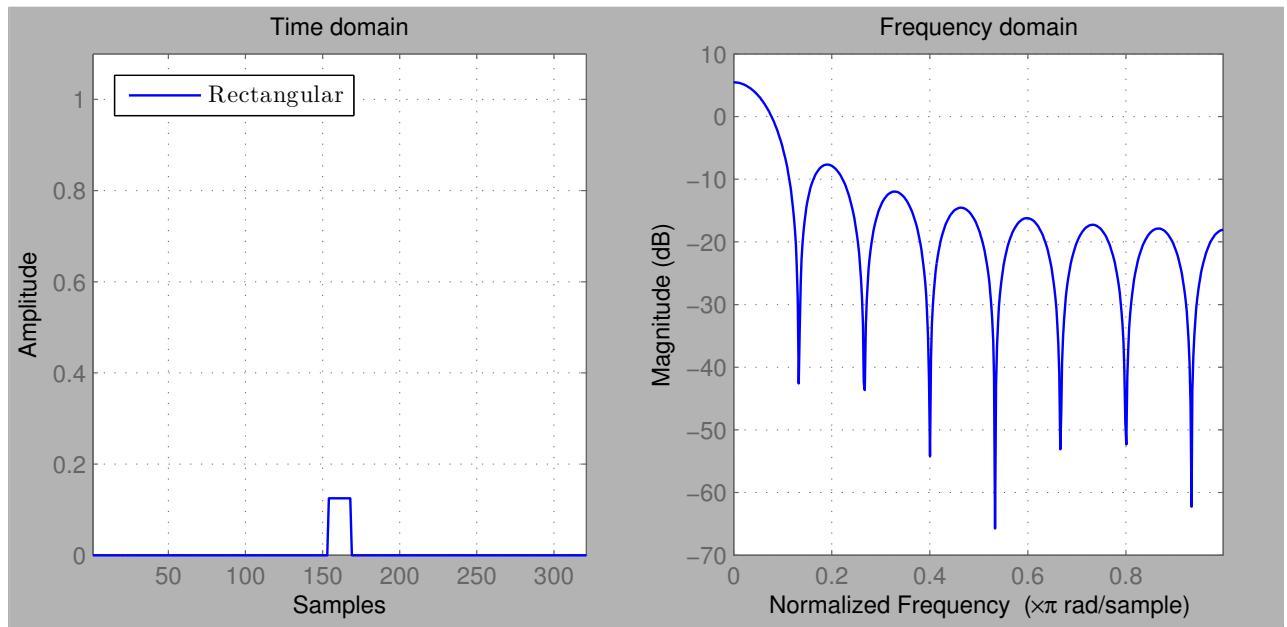


Figura 4.15: Filtro Reconstructor, Pulso Rectangular,  $M8$ .

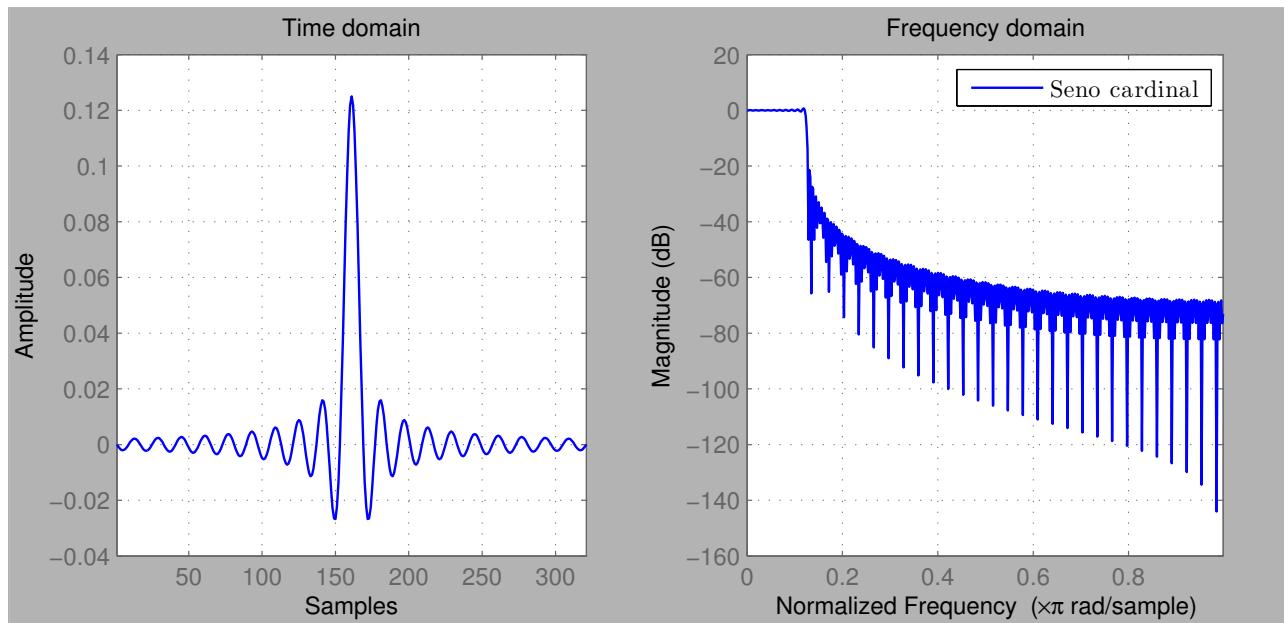


Figura 4.16: Filtro Reconstructor,Seno Cardinal,  $M8$ .

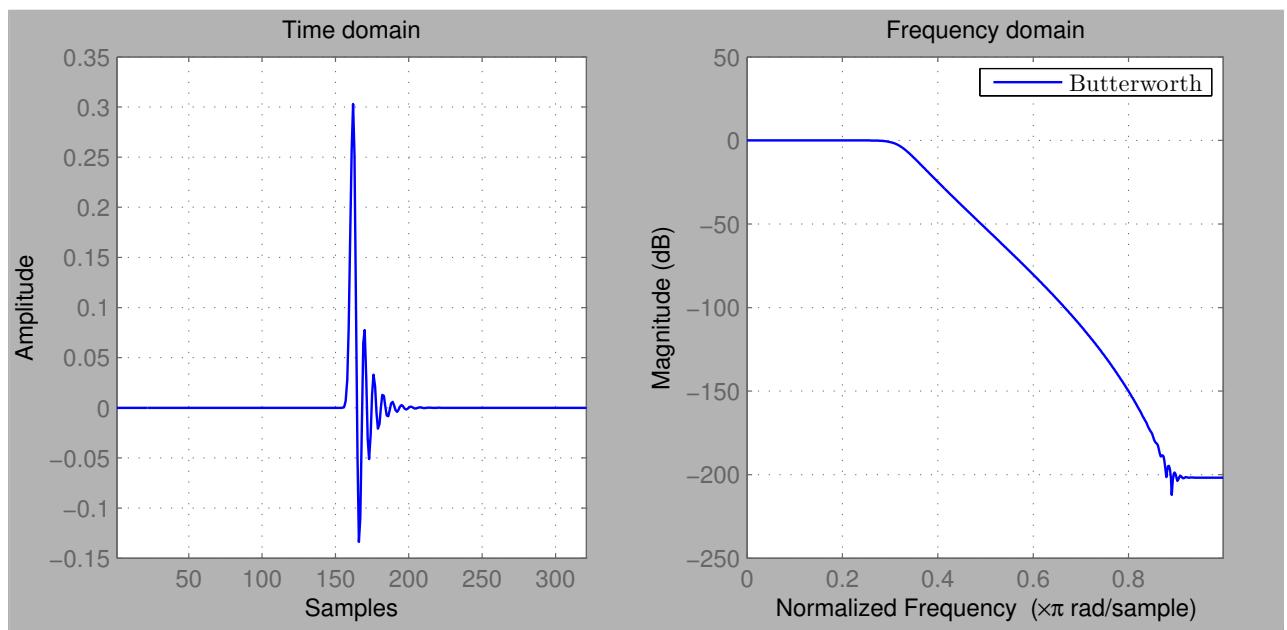


Figura 4.17: Filtro Reconstructor, Butterworth órden= 10,  $M = 8$ .

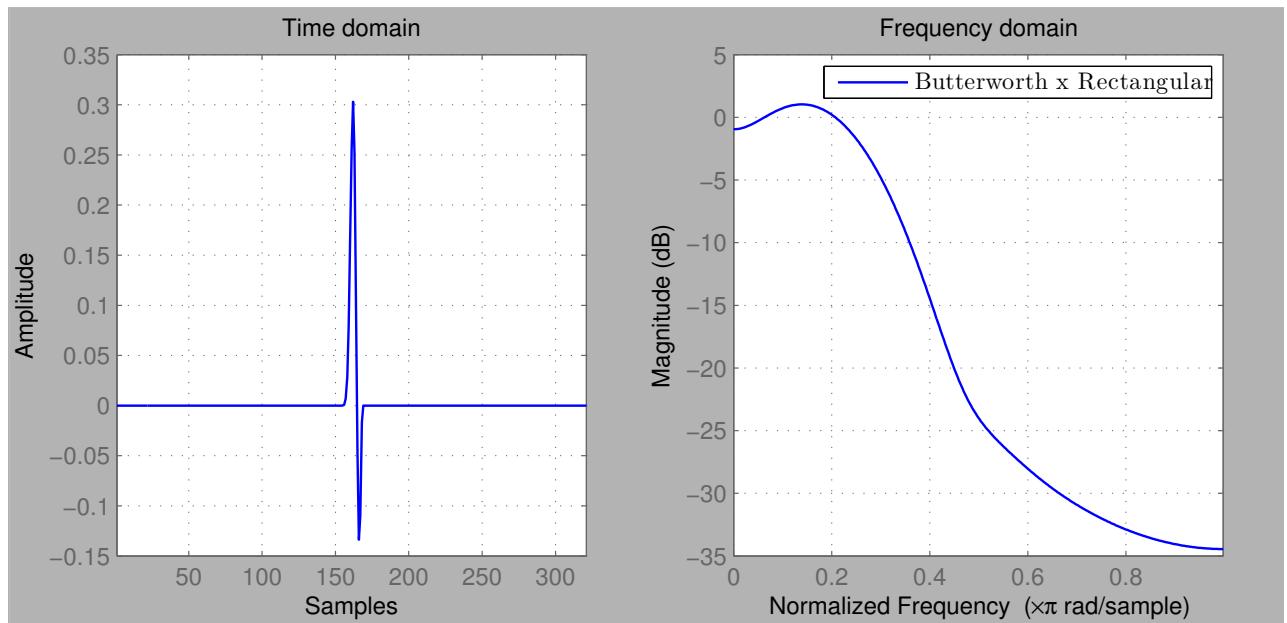


Figura 4.18: Filtro Reconstructor, Butterworth órden= 10 con ventana Rectangular,  $M = 8$ .

En la Figura (4.19), vemos todos los filtros en una misma gráfica.

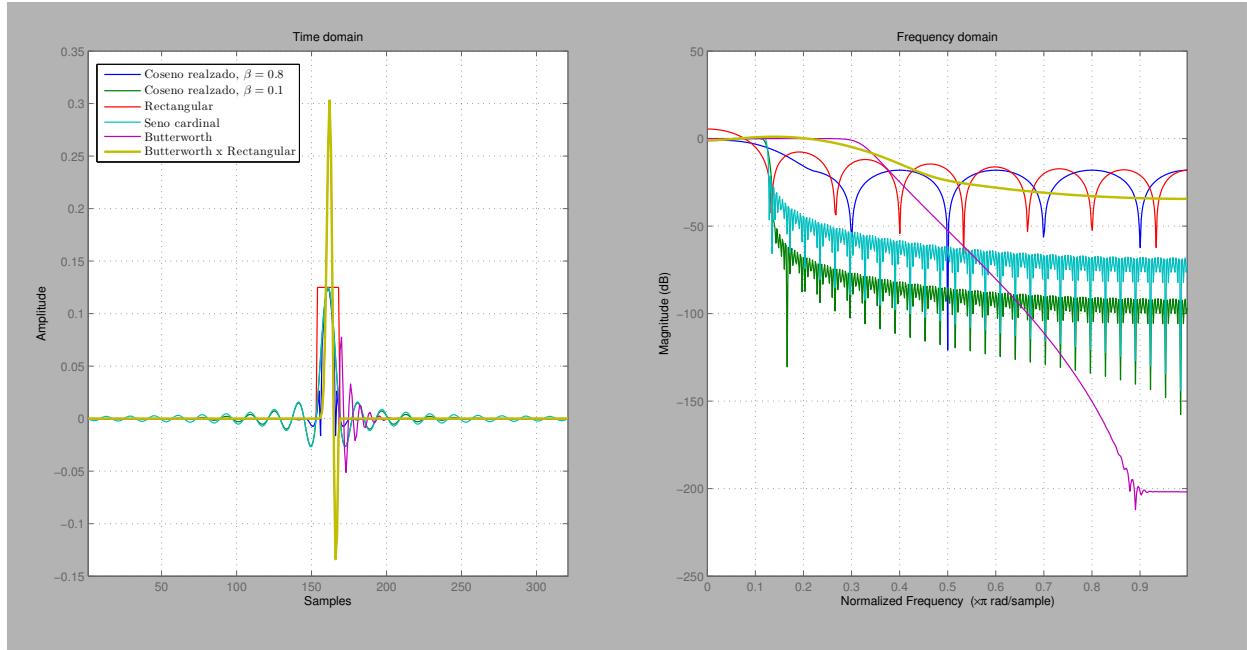


Figura 4.19: Todos los Filtros Reconstructores en una misma gráfica,  $M = 8$ .

Los cambios mencionados en el espectro de cada uno de los filtros reconstructores (aumentando el valor de  $M$  a  $M = 8$ ), se ven reflejados en los espectros de sus correspondientes señales reconstruidas en la Figura (4.20).

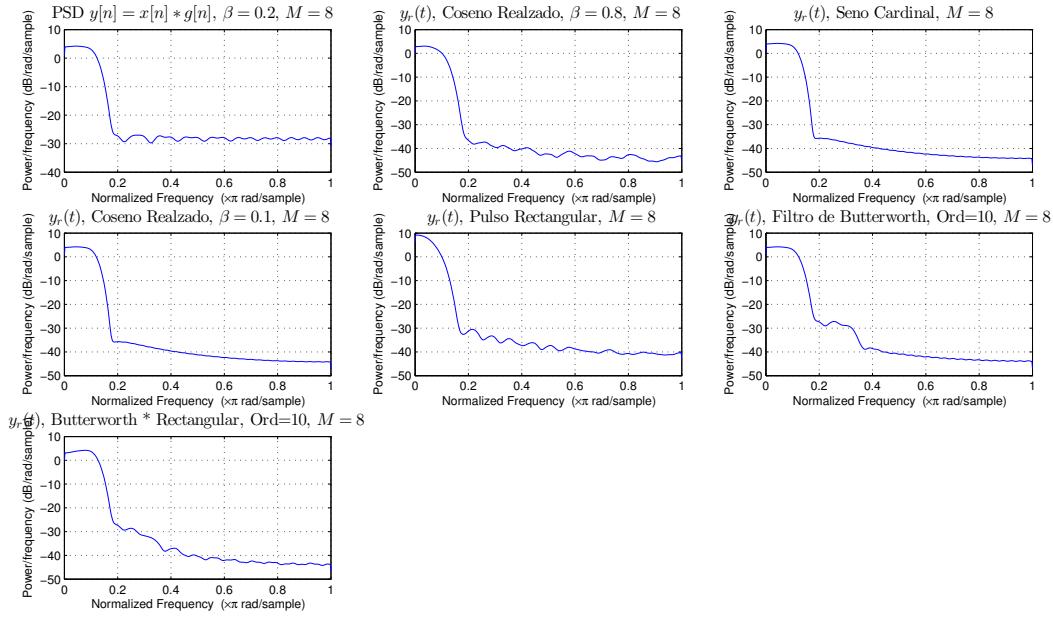


Figura 4.20: PSD de la Salida para los distintos filtros de reconstrucción,  $M = 8$ .

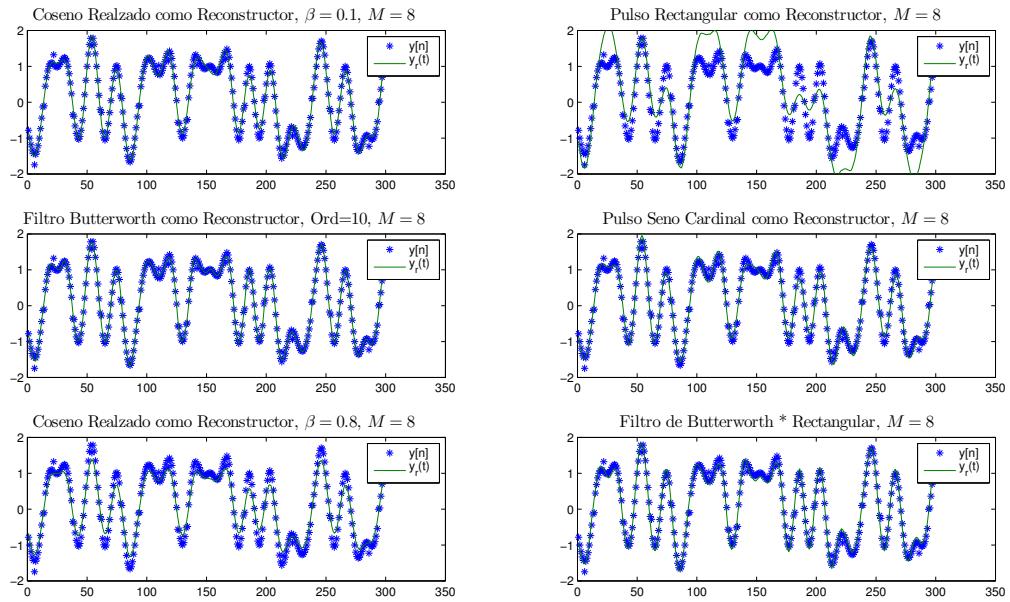


Figura 4.21: Señal  $y[n]$  vs. Salidas de los Filtros de Reconstrucción,  $M = 8$ .

En la Figura (4.21), se observan la comparación de las señales reconstruidas con la señal original, en donde se destaca que nuevamente la peor reconstrucción se logra con el filtro rectangular. En cambio, la reconstrucción con el resto de los filtros es buena y se debe de tener en cuenta que el pulso seno cardinal no es implementable.

Finalmente en las Figuras (4.22) y (4.23), se muestran los espectros y las respuestas al impulso (para  $M = 4$  y  $M = 8$  respectivamente), resultado de la convolución entre el filtro conformador de pulso (filtro interpolador) y los diferentes filtros de reconstrucción, tal que  $h_{tot}[n] = g[n] * h_r[n]$ , donde  $h_r[n]$  es cada uno de los diferentes filtros de reconstrucción utilizados.

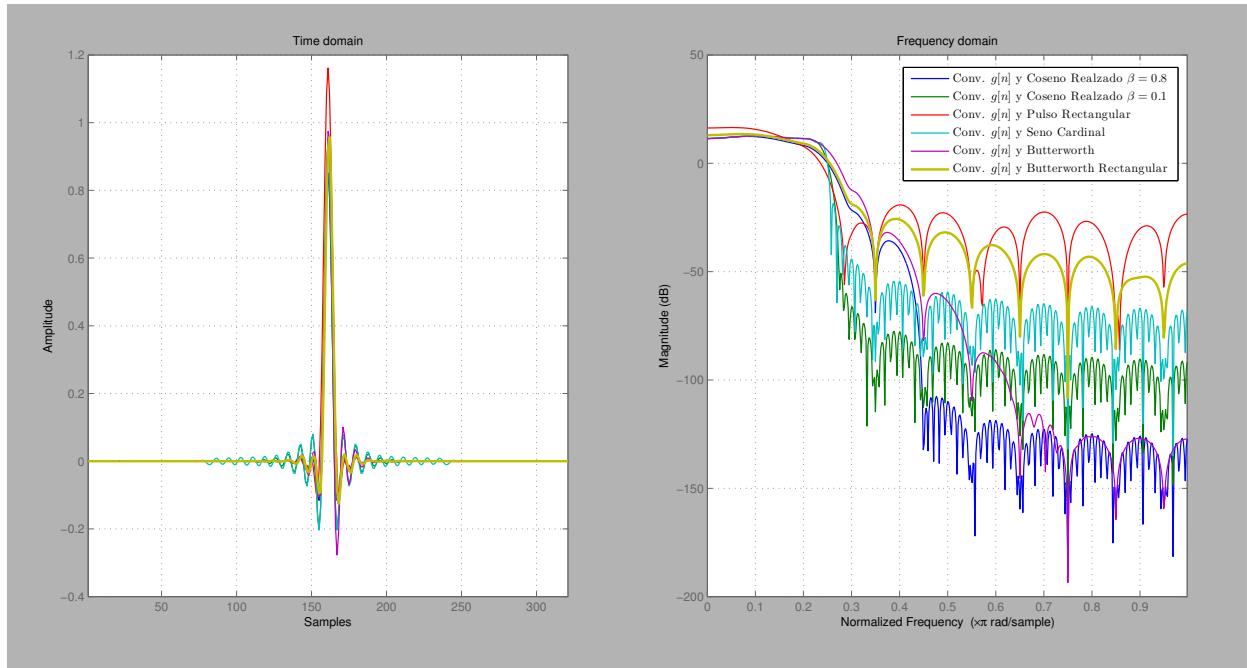


Figura 4.22: Respuesta total del filtro interpolador y los diferentes filtros reconstructores,  $M = 4$ .

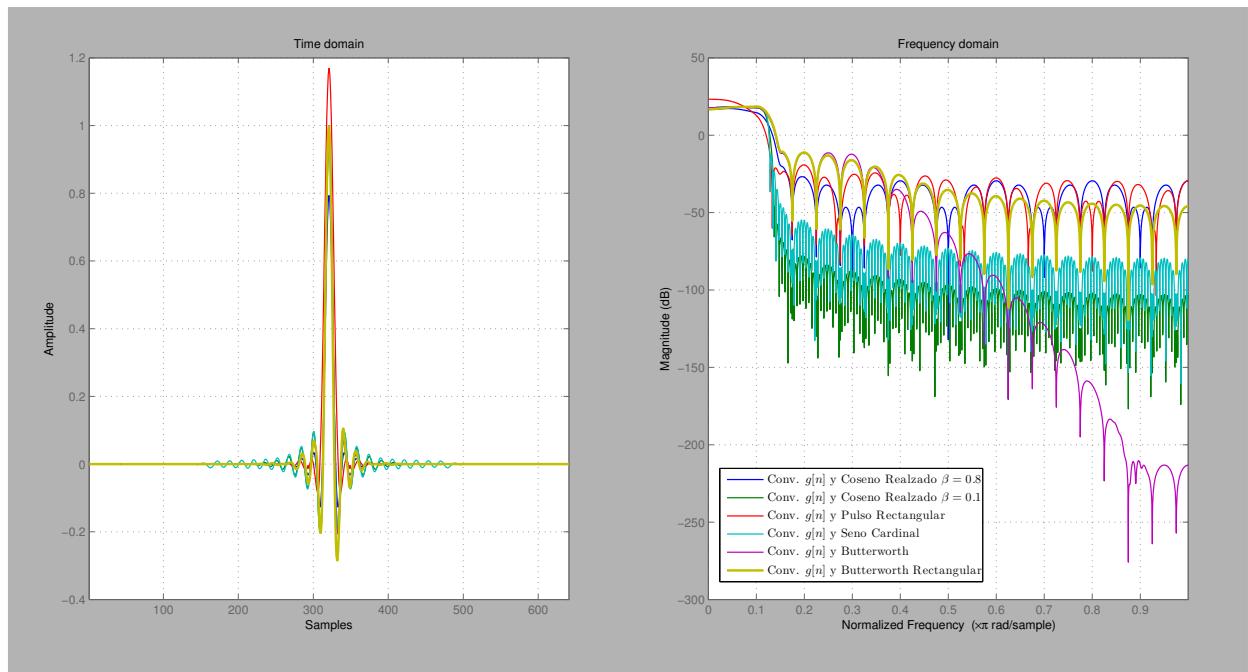


Figura 4.23: Respuesta total del filtro interpolador y los diferentes filtros reconstructores,  $M = 4$ .

# Capítulo 5

## Modulador Sigma-Delta de Segundo Órden

### 5.1. Descripción y Consignas

En la Figura (5.1), se presenta un modulador  $\Sigma\Delta$  y en la Figura (5.2), se muestra un diezmador por  $M$ , utilizando un filtro CIC.

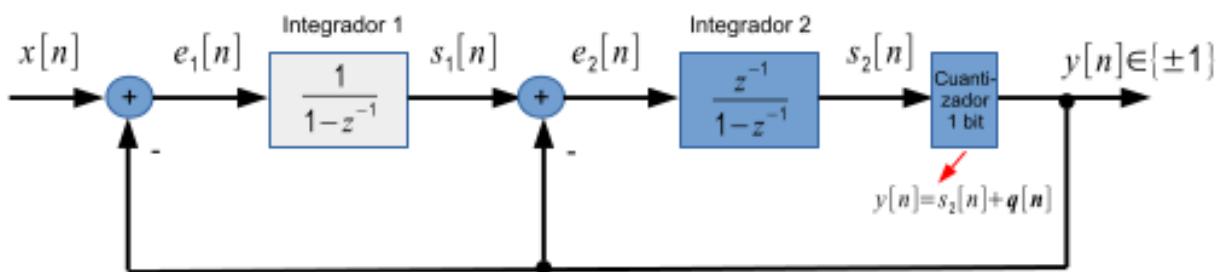


Figura 5.1: Modulador  $\Sigma\Delta$  de segundo órden.

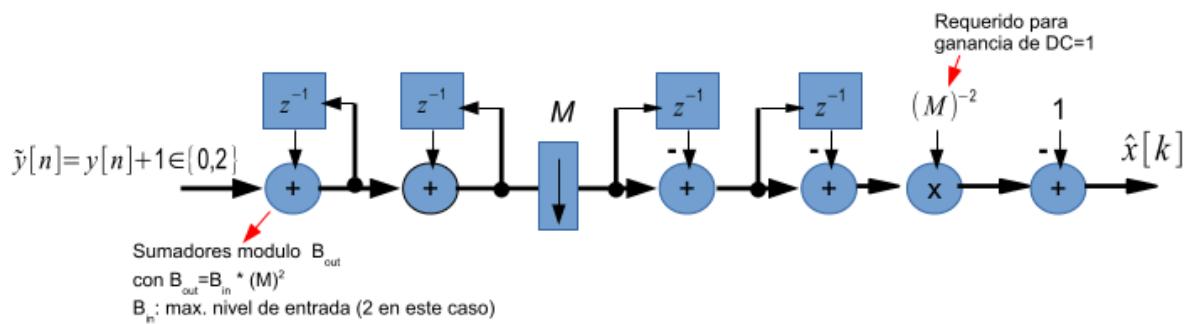


Figura 5.2: Diezmador usando Filtro CIC de dos etapas.

Un filtro CIC consiste de uno ó más *integradores* y pares de filtros *comb*. En el caso de un decimador, la señal de entrada alimenta la cascada de integradores, luego el submuestreo seguido de una cascada de filtros comb (en igual número que los integradores). En cambio, un interpolador CIC, posee una estructura revertida con respecto al diezmador, primero los filtros comb, luego el sobremuestreo seguido de los integradores (en igual número que los comb). Los filtros CIC tienen características de frecuencia de paso bajo, mientras que los filtros FIR pueden tener características de paso bajo, paso alto o frecuencia de paso de banda. Los filtros del CIC utilizan solamente la adición y la resta, mientras que los filtros FIR utilizan suma, resta, pero la mayoría de los filtros FIR también requieren multiplicación.

Un integrador se puede expresar en el dominio temporal discreto de la siguiente manera:

$$y[n] = y[n - 1] + x[n]$$

La función de transferencia correspondiente al integrador es:

$$H_I(z) = \frac{1}{1 - z^{-1}}$$

Por otra parte, un filtro comb se puede expresar en el dominio temporal discreto de la siguiente manera:

$$y[n] = x[n] - x[n - RM]$$

donde  $R$  es el factor de sobremuestreo y  $M$  el factor de diezmado.

La función de transferencia correspondiente al filtro comb es:

$$H_C(z) = 1 - z^{-RM}$$

Se debe considerar que la señal de entrada al modulador es:

$$x[n] = \sum_{k=0}^{N-1} a[k]g[n - kRM], \quad a[k] \in \{\pm 1\}$$

con sobremuestreo de  $RM$  y filtro conformador  $g[n]$ .

Se propone desarrollar las siguientes actividades:

1. Mostrar que la transformada Z de la salida del modulador  $\Sigma\Delta$  es:

$$Y(z) = X(z)z^{-1} + Q(z)(1 - z^{-1})^2$$

2. Teniendo en cuenta que la salida del modulador se diezma por un factor de  $M$ , mediante el diezmador CIC. Obtener por simulación la densidad espectral de potencia de la señal a la salida del modulador  $\Sigma\Delta$  y del filtro CIC.
3. Generar el diagrama ojo a partir de la señal  $\hat{x}[k]$ .
4. Graficar el diagrama de ojo de la señal  $\hat{x}[k]$  filtrada por un filtro de caida cosenoidal (pasa bajos), similar a  $g[n]$  pero sobremuestreado por un factor  $R$ . Explicar porqué mejora el diagrama ojo y repetir para diferentes valores de  $M$  y  $R$ .

## 5.2. Resultados

Antes de comenzar con el desarrollo del laboratorio, vamos a realizar una breve introducción al tema, partiendo un simple pero esclarecedor análisis de los convertidores A/D  $\Sigma\Delta$ . En la Figura (5.3), se puede observar un diagrama en bloques básico de un conversor A/D.

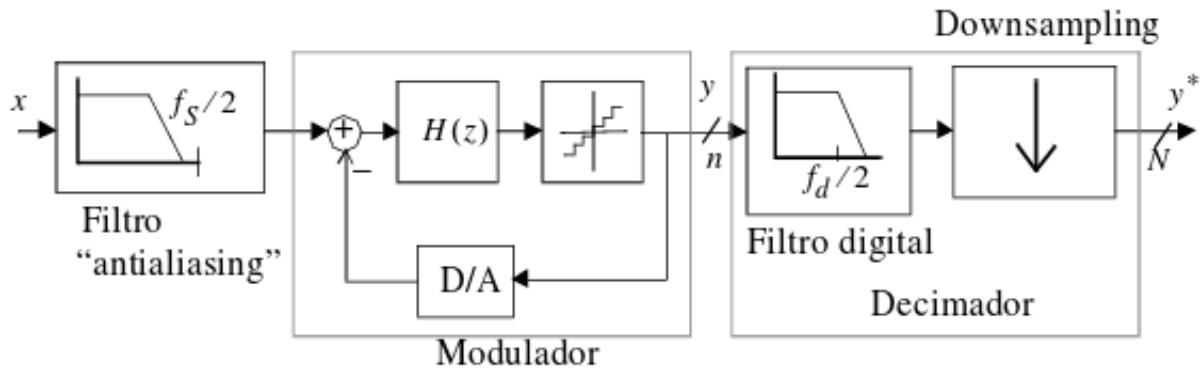
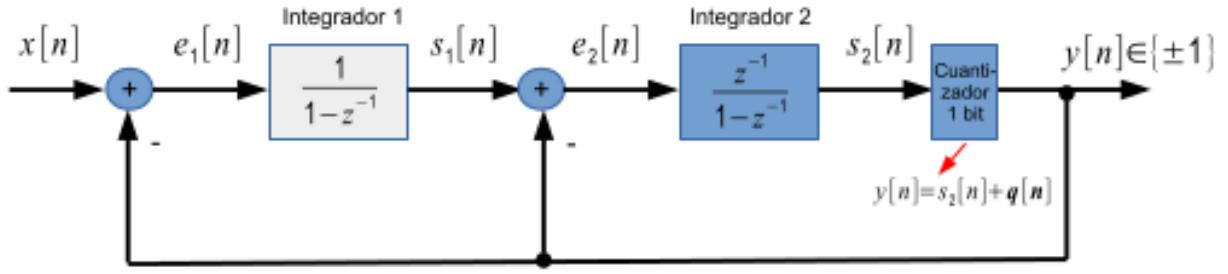


Figura 5.3: Diagrama de Bloques de un Convertidor A/D  $\Sigma\Delta$ .

El conversor A/D  $\Sigma\Delta$  está constituido básicamente por tres elementos:

- **Filtro analógico “antialiasing”:** Se encarga de eliminar de la señal de entrada todas las componentes espectrales por encima de la mitad de la frecuencia de muestreo.
- **Modulador:** En este bloque se muestrea y cuantiza la señal. Adicionalmente, es posible filtrar el error inherente a la cuantización, conformando su densidad espectral de potencia de modo que la mayor parte de ésta quede fuera de la banda de la señal, de donde es eliminado mediante filtrado digital. Este hecho ha dado origen al término noise-shaping, que se usa también para denominar a los moduladores  $\Sigma\Delta$ . La salida del modulador consiste en un número reducido de bits (usualmente sólo uno) a la frecuencia de muestreo.
- **Decimador:** En este bloque, puramente digital, tras un filtrado que elimina todas las componentes fuera de la banda de la señal, incluido gran parte del error de cuantización, se reduce la frecuencia de muestreo mediante un proceso de decimación. Como resultado se obtiene la señal de entrada, codificada con un elevado número de bits, a la frecuencia de Nyquist.

Luego de ésta breve introducción, procedemos a hallar la expresión de la transformada Z de  $Y(z)$  (salida del modulador  $\Sigma\Delta$ ), para ello definimos las siguientes señales, presentes en el diagrama de bloques de la Figura (5.4).


 Figura 5.4: Modulador  $\Sigma\Delta$  de segundo orden.

De la cual se desprenden las siguientes expresiones:

$$Y(z) = S_2(z) + Q(z) \quad (5.1)$$

$$S_1(z) = \frac{E_1(z)}{1 - z^{-1}} \quad (5.2)$$

$$S_2(z) = \frac{E_2(z)z^{-1}}{1 - z^{-1}} \quad (5.3)$$

$$E_1(z) = X(z) - Y(z) \quad (5.4)$$

$$E_2(z) = S_1(z) - Y(z) \quad (5.5)$$

Con las Ecuaciones (5.1) a (5.5), podemos hallar la expresión  $Y(z)$ , en función de la entrada  $X(z)$  y el ruido de cuantización  $Q(z)$ :

$$\begin{aligned} E_2(z) &= \frac{E_1(z)}{1 - z^{-1}} - Y(z) \\ &= \frac{X(z)}{1 - z^{-1}} - \frac{Y(z)}{1 - z^{-1}} - Y(z) \\ &= \frac{X(z)}{1 - z^{-1}} - Y(z) \left( \frac{1}{1 - z^{-1}} + 1 \right) \\ &= \frac{X(z)}{1 - z^{-1}} - Y(z) \frac{2 - z^{-1}}{1 - z^{-1}} \end{aligned}$$

$$\begin{aligned}
 S_2(z) &= \frac{X(z)z^{-1}}{(1-z^{-1})^2} - Y(z)\frac{(2-z^{-1})z^{-1}}{(1-z^{-1})^2} \\
 Y(z) &= \frac{X(z)z^{-1}}{(1-z^{-1})^2} - Y(z)\frac{(2-z^{-1})z^{-1}}{(1-z^{-1})^2} + Q(z) \\
 Y(z) \left[ 1 + \frac{(2-z^{-1})z^{-1}}{(1-z^{-1})^2} \right] &= \frac{X(z)z^{-1}}{(1-z^{-1})^2} + Q(z) \\
 Y(z) \left[ \frac{(1-z^{-1})^2 + (2-z^{-1})z^{-1}}{(1-z^{-1})^2} \right] &= \frac{X(z)z^{-1}}{(1-z^{-1})^2} + Q(z) \\
 Y(z) \frac{1}{(1-z^{-1})^2} &= \frac{X(z)z^{-1}}{(1-z^{-1})^2} + Q(z) \\
 Y(z) &= X(z)z^{-1} + Q(z)(1-z^{-1})^2
 \end{aligned}$$

A continuación, en la Figura (5.5), se presenta el modulador  $\Sigma\Delta$  de segundo orden. Y en la Figura (5.6), un Diezmador usando un filtro CIC de dos etapas, Donde la salida del modulador, se conecta a la entrada del filtro CIC. Además, la entrada al modulador se define como:

$$x[n] = \sum_{k=0}^{N-1} a[k]g[n - kRM], \quad a[k] \in \{\pm 1\}$$

Con  $g[n]$  un filtro de caída cosenoidal con factor de sobremuestreo de  $RM$ , mientras que el factor de sobremuestreo a la salida del filtro CIC es  $R$ , debido al diezmado por  $M$ :

$$\hat{x}[n] = \sum_{k=0}^{N-1} a[k]g[n - kR] + ruido_Q$$

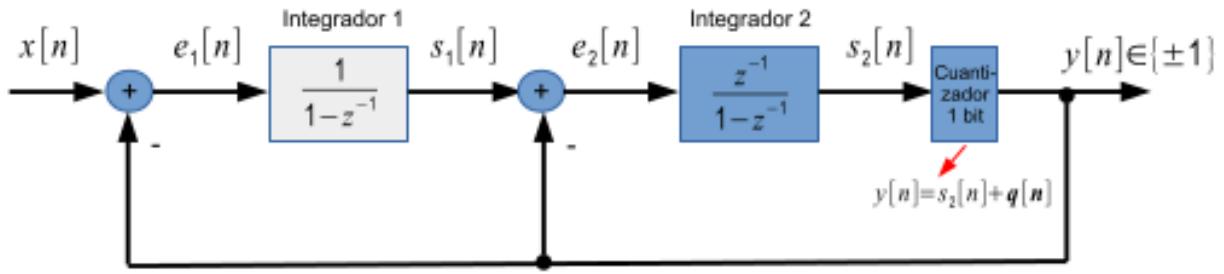
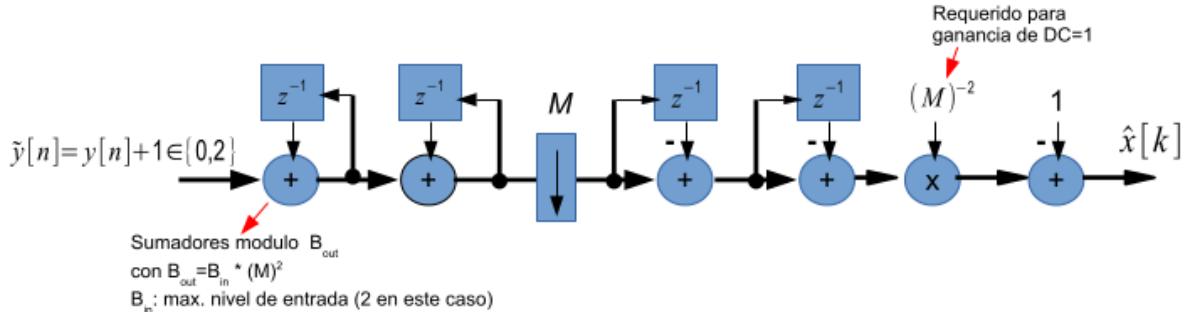

 Figura 5.5: Modulador  $\Sigma\Delta$  de segundo orden.


Figura 5.6: Diezmador usando Filtro CIC de dos etapas.

A continuación, vamos a presentar los resultados y su correspondiente análisis para distintos valores de  $R$  y  $M$ , mas precisamente para  $R = 4, 8$  y  $M = 8, 16$ , dando como resultado 4 casos distintos. Además, se agrega un caso extra con  $M = 16$  y  $R = 4$ , donde se reemplazo el ruido de cuantización (efecto del proceso de cuantización) del modulador  $\Sigma\Delta$ , por ruido blanco y se analizaron los resultados.

Como dato adicional, se menciona que para poder analizar los espectros de las señales se utilizó la densidad espectral de potencia, tal y como se venía utilizando hasta el momento.

CASO 1: $M = 16, R = 4$
-------------------------

En la Subfigura superior de la Figura (5.7) en color azul, se muestra la PSD de la señal a la salida del modulador  $\Sigma\Delta$  y en color verde la PSD asociada al cuantizador presente en el sistema. Por otra parte, en la Subfigura inferior, se muestra la salida del filtro diezmador CIC. Como se puede observar (¡cuidado que las escalas de ambas subfiguras son distintas!), la salida del filtro diezmador CIC se encuentra decimada en un factor de  $M = 16$ , con respecto a la salida del modulador  $\Sigma\Delta$ , cuyo factor de sobremuestreo es  $RM = 4 \times 16 = 64$ .

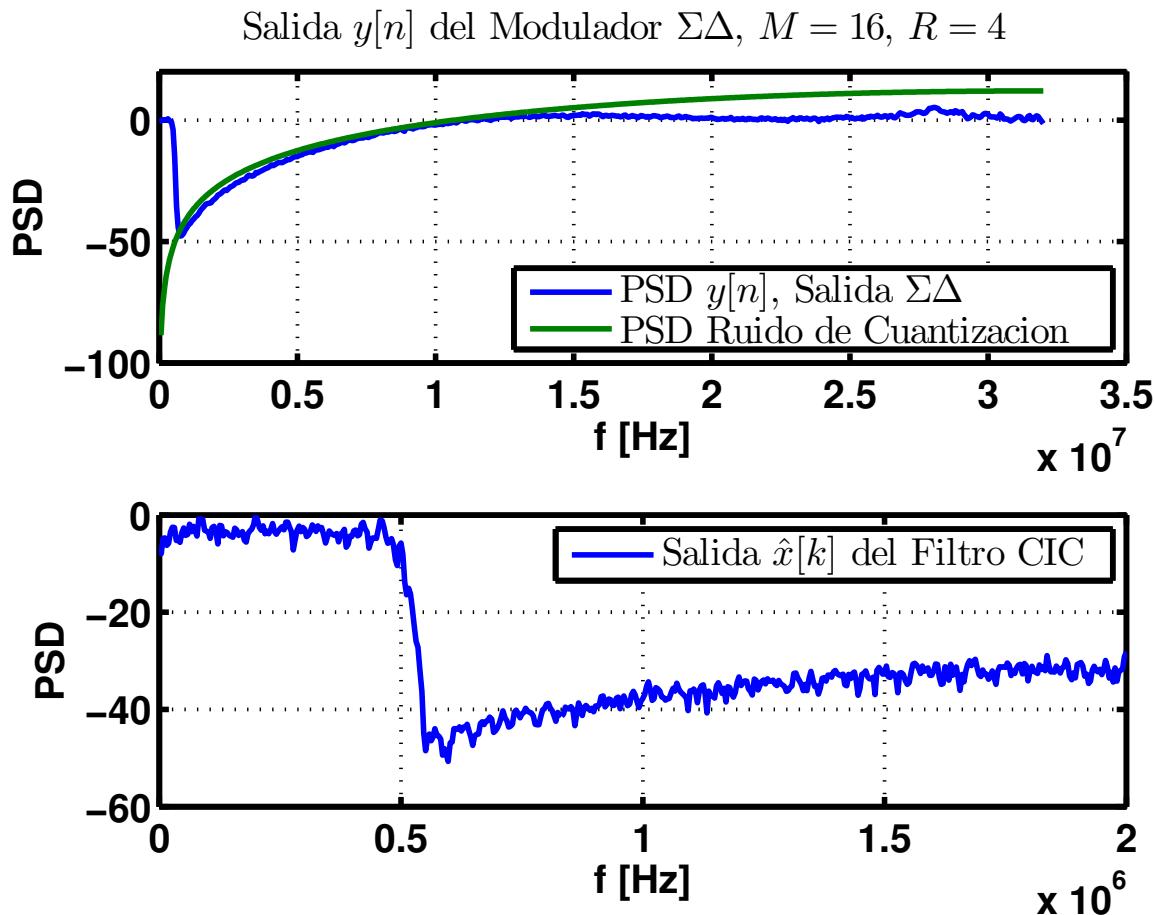


Figura 5.7: Salidas Modulador  $\Sigma\Delta$  y CIC,  $M = 16, R = 4$ .

Para poder entender el accionar del modulador  $\Sigma\Delta$  se creó la Figura (5.8), en donde se muestra la PSD de la señal de entrada al modulador  $x[n]$ , la PSD de la señal de salida de modulador  $y[n]$  y la PSD del ruido de cuantización.

Como se mencionó al comienzo del capítulo, el espectro de la señal de salida del modulador se puede expresar de la siguiente manera:

$$Y(z) = X(z)z^{-1} + Q(z)(1 - z^{-1})^2$$

De la cual se desprende que la densidad espectral de potencia de  $y[n]$ , corresponderá a la densidad espectral de potencia de  $x[n]$  más la PSD correspondiente al proceso de cuantización. Dichas señales se pueden ver en la Figura (5.8) y permiten entender los resultados obtenidos en la Figura (5.7).

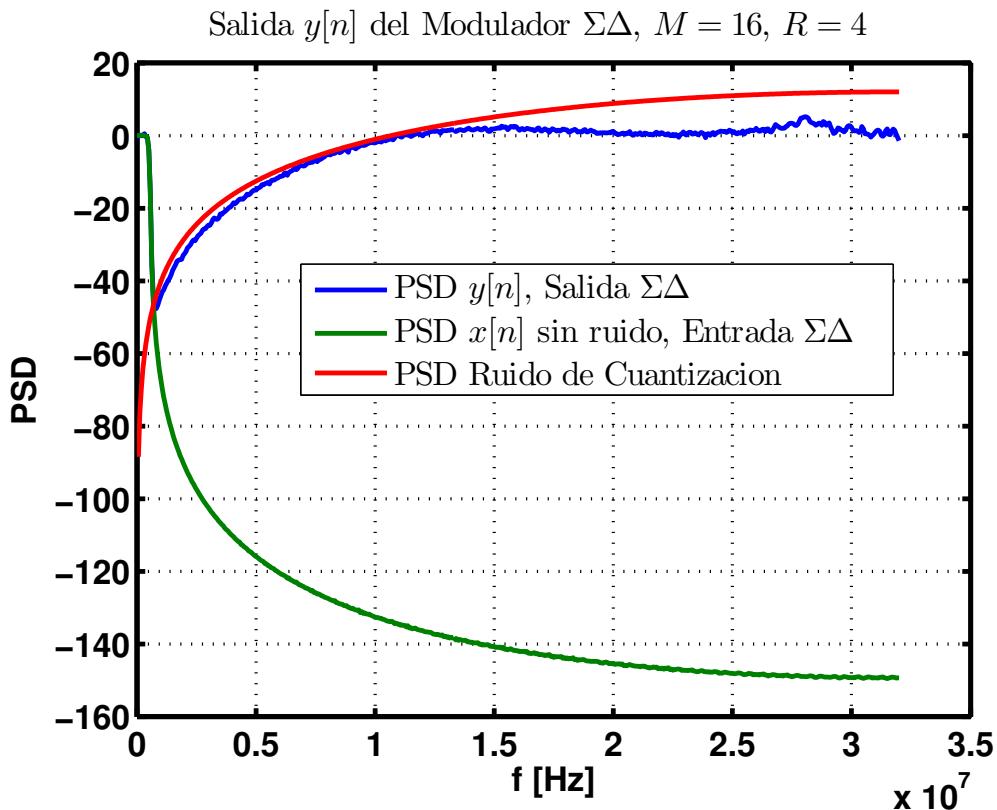


Figura 5.8: Análisis de las diferentes señales del modulador  $\Sigma\Delta$ .

A continuación, la señal de salida del modulador  $\Sigma\Delta$  se inserta en un diezmador CIC (con factor de decimación  $M$ ). En la Figura (5.9), se muestran los diagramas de ojos de la señal a la salida del diezmador CIC; en la Subfigura de la izquierda la señal no se encuentra filtrada, en cambio en la Subfigura de la derecha, se filtró la señal con un filtro pasa bajos de caída cosenoidal y factor de sobremuestreo  $R$ , esto se debe a que la salida del diezmador CIC se encuentra a una tasa de sobremuestreo de  $R$ , lo que implica un filtro con sobremuestreo de  $R$  porque de mantener un sobremuestreo de  $RM$ , el filtro dejaría pasar como válidas, componentes espúreas debido a la cuantización.

Observando el diagrama de la Figura (5.9), se destaca el hecho de que el filtrado de la señal  $\hat{x}[n]$ , mejora el diagrama de ojo logrando una mayor apertura del mismo, gracias al filtrado de espectro indeseado.

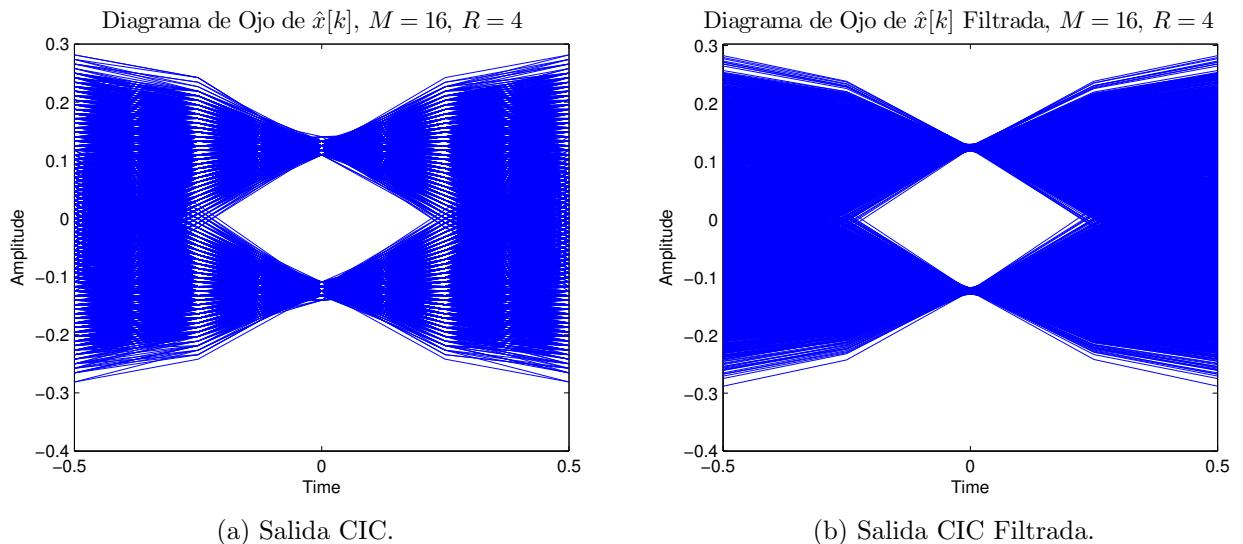


Figura 5.9: Diagramas de Ojo,  $M = 16, R = 4$ .

La mejora en el diagrama de ojos de la señal filtrada  $\hat{x}[n]$ , con respecto a la señal  $\hat{x}[n]$  sin filtrar, se debe a que el filtro pasa bajos utilizado, elimina las componentes indeseadas de alta frecuencia de la señal original, tal y como se puede ver en la Figura (5.10), donde las componentes de alta frecuencia de la señal de color azul se atenuaron (señal color verde), permitiendo una mayor apertura ocular.

Como vimos en la Figura (5.8), las componentes de alta frecuencia de señal  $\hat{x}[n]$  (Figura 5.10, señal color azul), se deben al ruido de cuantización insertado por el cuantizador del modulador  $\Sigma\Delta$ . El filtro utilizado fue un filtro pasa bajos, cuya banda de rechazo coincide con la banda de frecuencias del ruido de cuantización de la señal  $\hat{x}[n]$ , lo que explica espectro de la señal de color verde de la Figura (5.10) y la mejora en el diagrama de ojos de la subfigura de la derecha de la Figura (5.9).

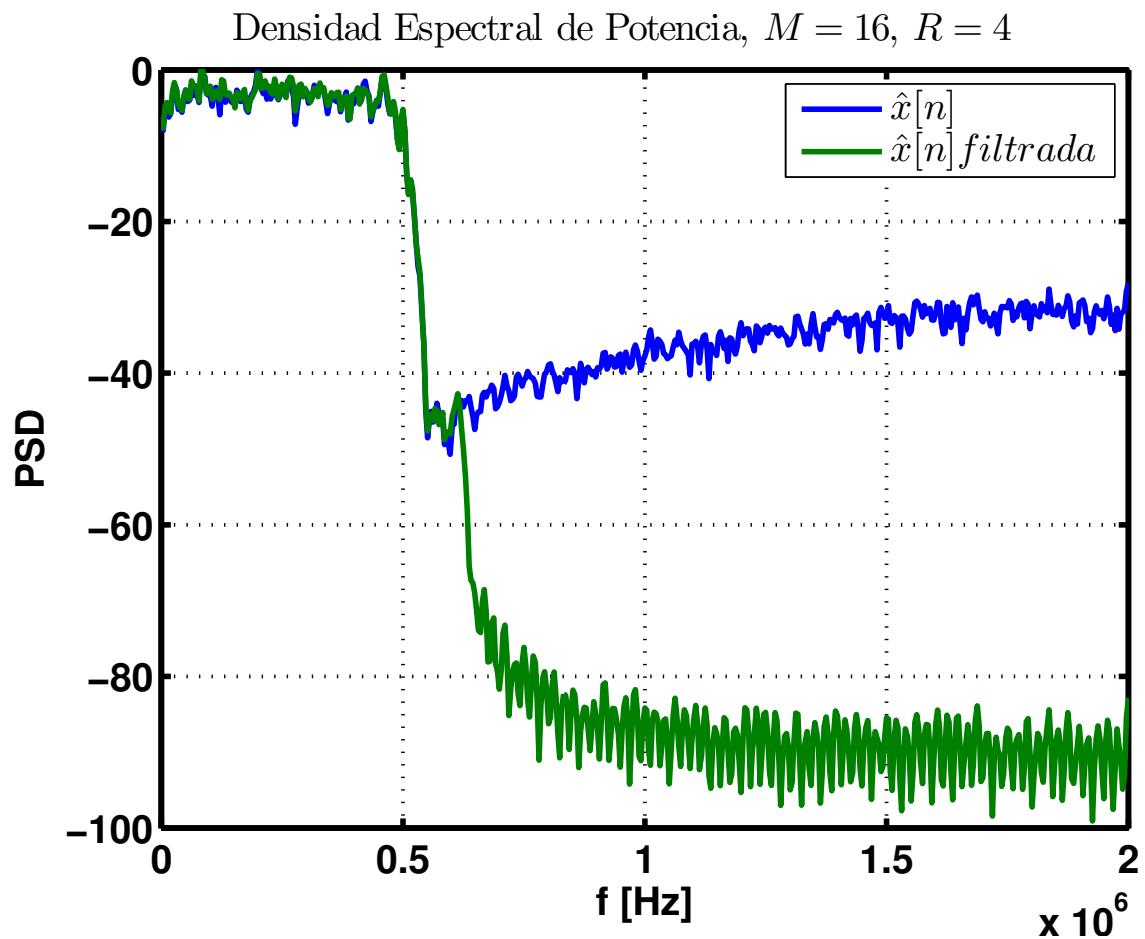


Figura 5.10: PSD Salida CIC con y sin Filtrada LPF,  $M = 16, R = 4$ .

CASO 2: $M = 16, R = 8$
-------------------------

A continuación, se repiten las simulaciones mostradas en el Caso 1, pero con un nuevo valor de  $R$  y manteniendo el mismo  $M$  en relación con el Caso 1. Al aumentar el factor  $R$  de 4 a 8, la relación  $RM = 16 \times 8 = 128$  y por lo tanto la señal a la salida del modulador posee un mayor sobremuestreo que el Caso 1. Además, como el valor de  $M$  se mantuvo igual, la relación de diezmado es la misma.

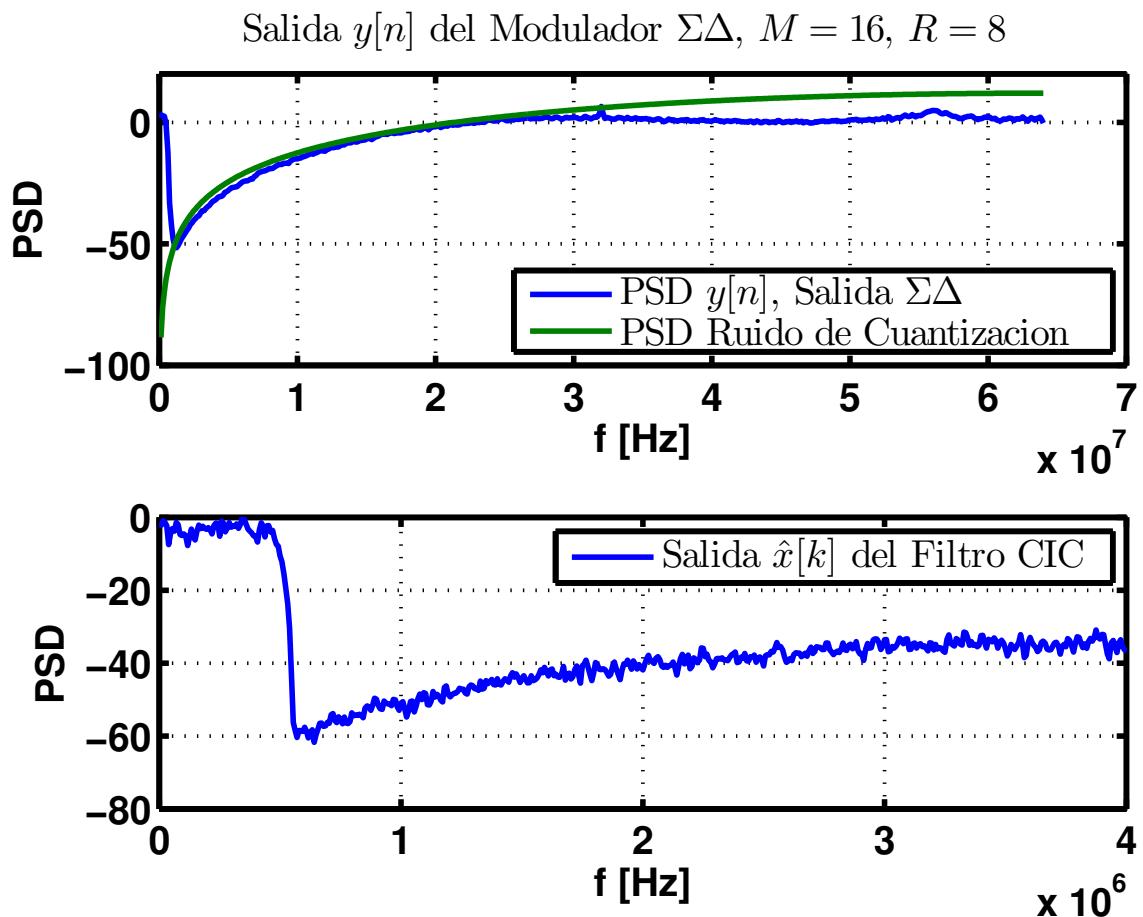


Figura 5.11: Salidas Modulador  $\Sigma\Delta$  y CIC,  $M = 16, R = 8$ .

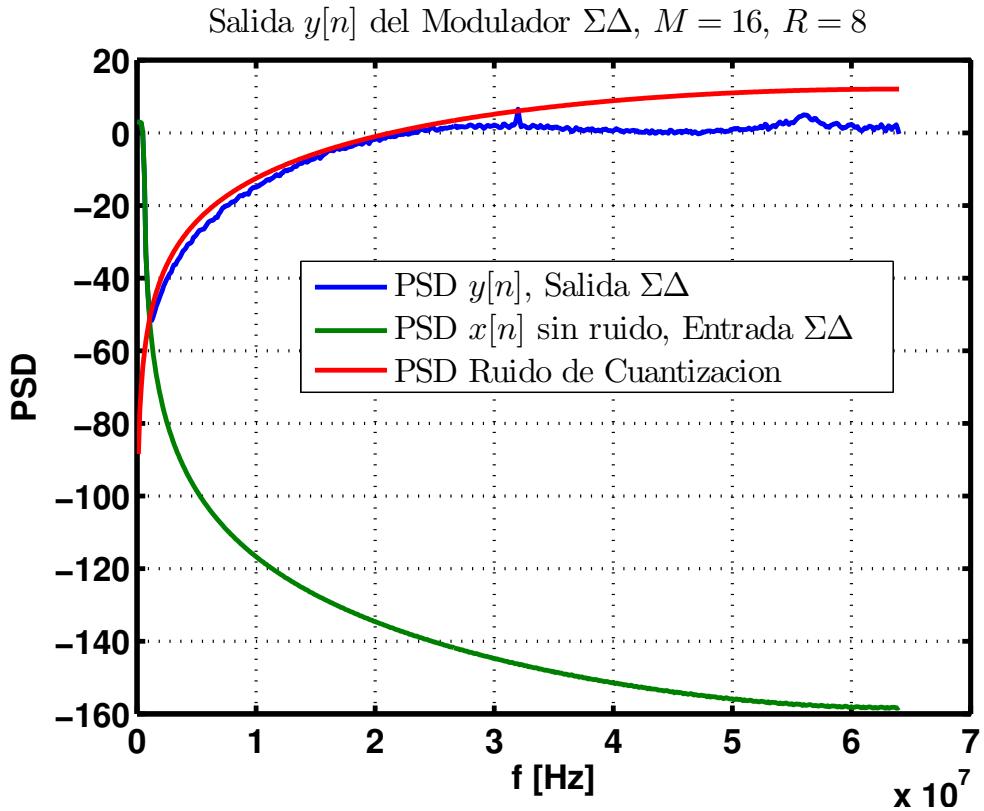


Figura 5.12: Análisis de las diferentes señales del modulador  $\Sigma\Delta$ .

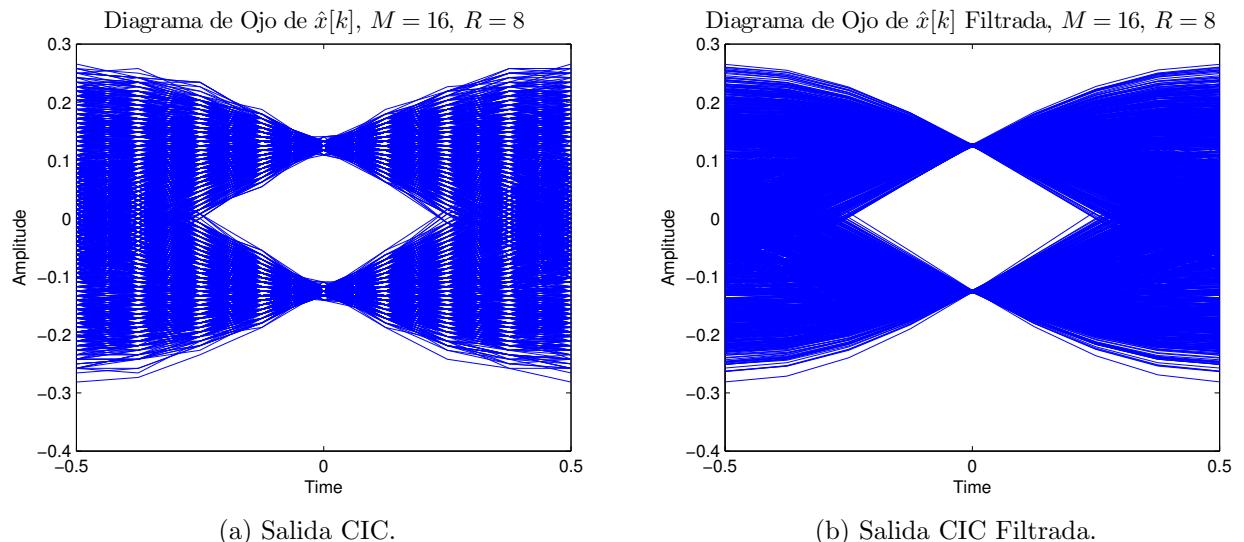


Figura 5.13: Diagramas de Ojo,  $M = 16$ ,  $R = 8$ .

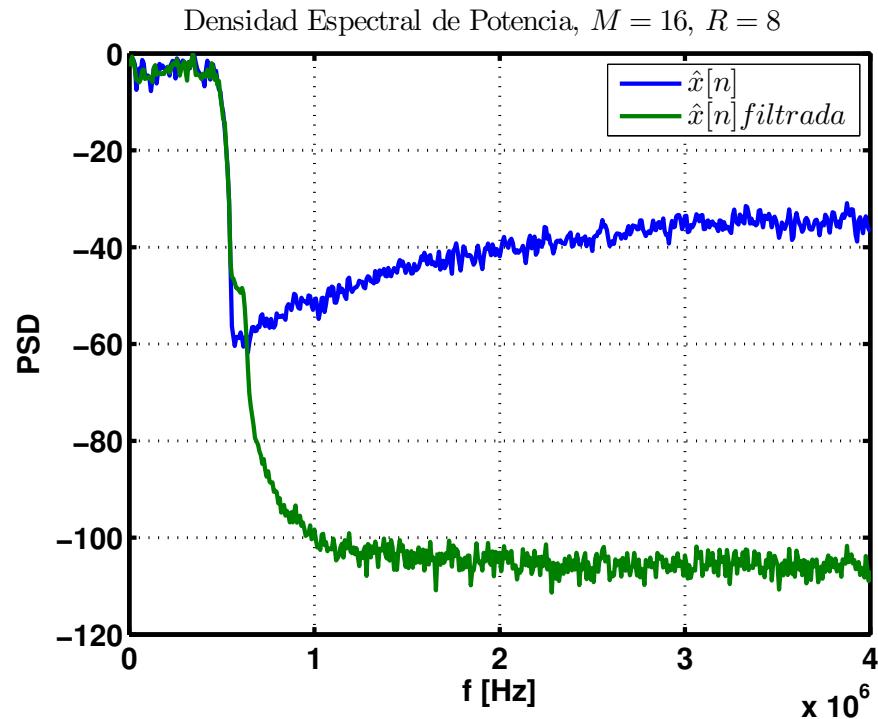


Figura 5.14: PSD Salida CIC con y sin Filtrada LPF,  $M = 16, R = 8$ .

Al igual que el caso anterior, el hecho de filtrar la señal de salida del diezmador, hace que el diagrama de ojos correspondiente posea mayor apertura, gracias a la atenuación de la banda de altas frecuencias de la  $\hat{x}[n]$  (ruido de cuantización).

CASO 3:  $M = 8, R = 8$

En este tercer caso, la relación  $RM = 8 \times 8 = 64$ , por ende el diezmado con respecto al Caso 2 será menor. A continuación, se presentan los resultados obtenidos para el Caso 3, en las Figuras (5.15), (5.16), (5.17) y (5.18).

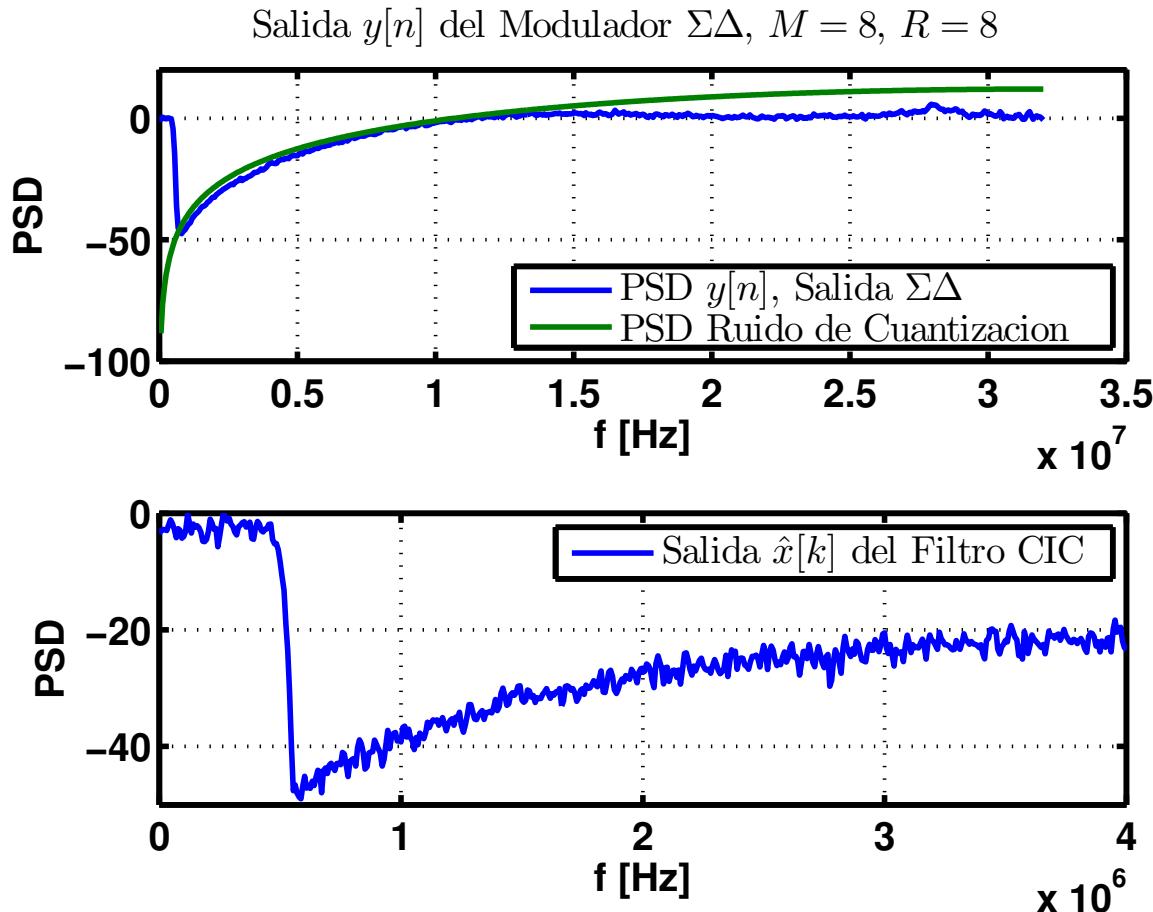
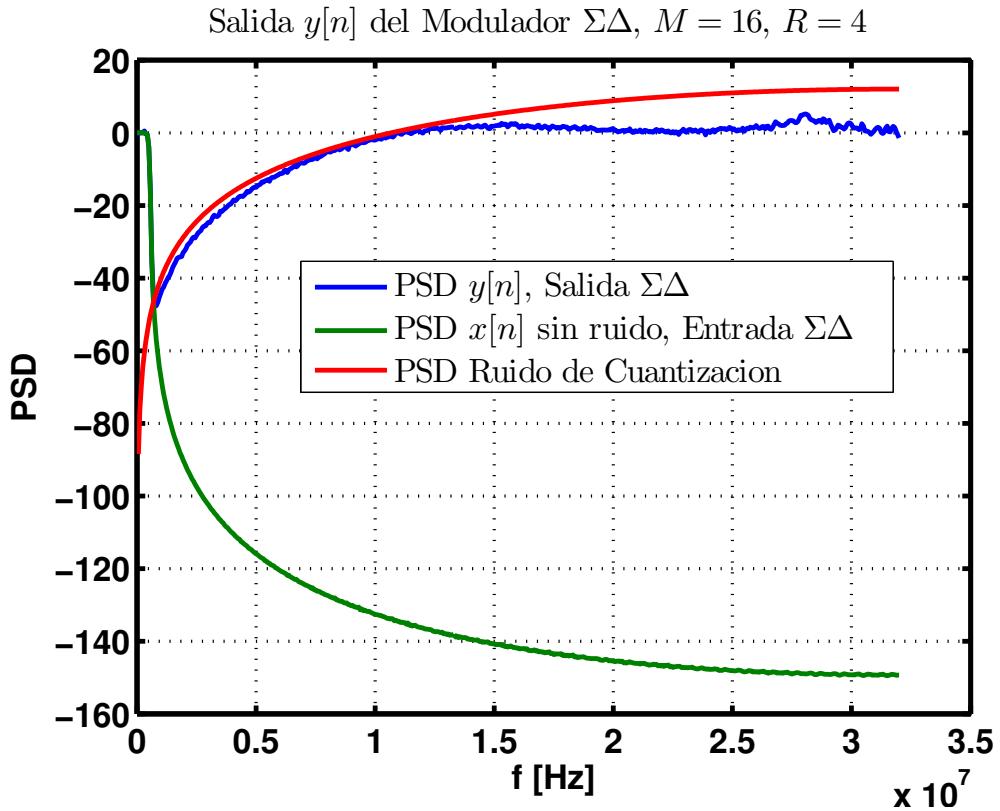
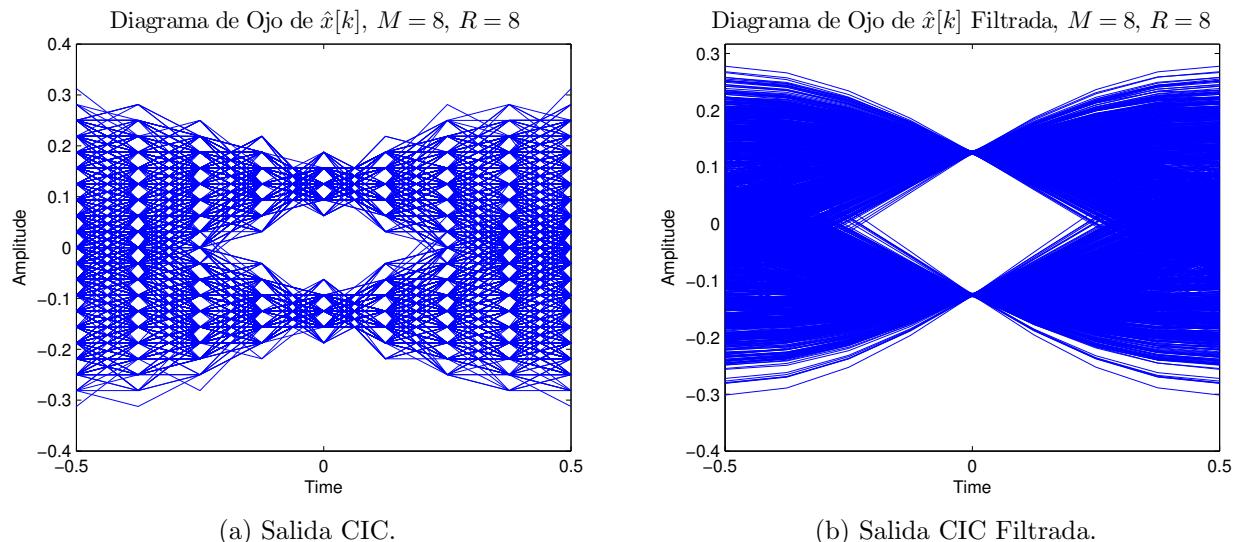


Figura 5.15: Salidas Modulador  $\Sigma\Delta$  y CIC,  $M = 8, R = 8$ .

Figura 5.16: Análisis de las diferentes señales del modulador  $\Sigma\Delta$ .Figura 5.17: Diagramas de Ojo,  $M = 8$ ,  $R = 8$ .

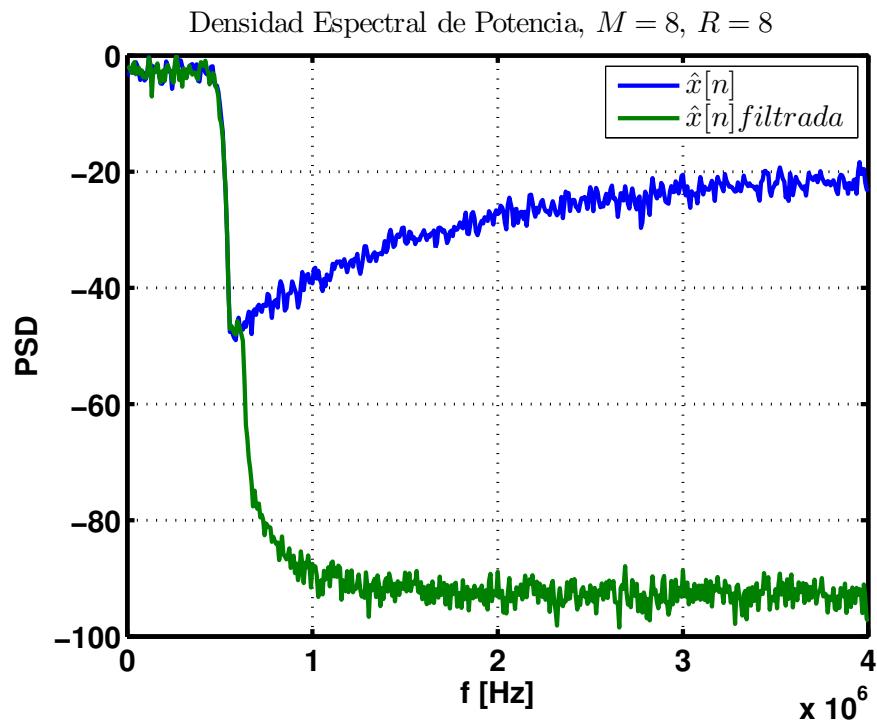


Figura 5.18: PSD Salida CIC con y sin Filtrada LPF,  $M = 8, R = 8$ .

CASO 4:  $M = 8, R = 4$

En este cuarto caso, la relación  $RM = 4 \times 8 = 32$ . A continuación, se presentan los resultados obtenidos para el Caso 4, en las Figuras (5.19), (5.20), (5.21) y (5.22).

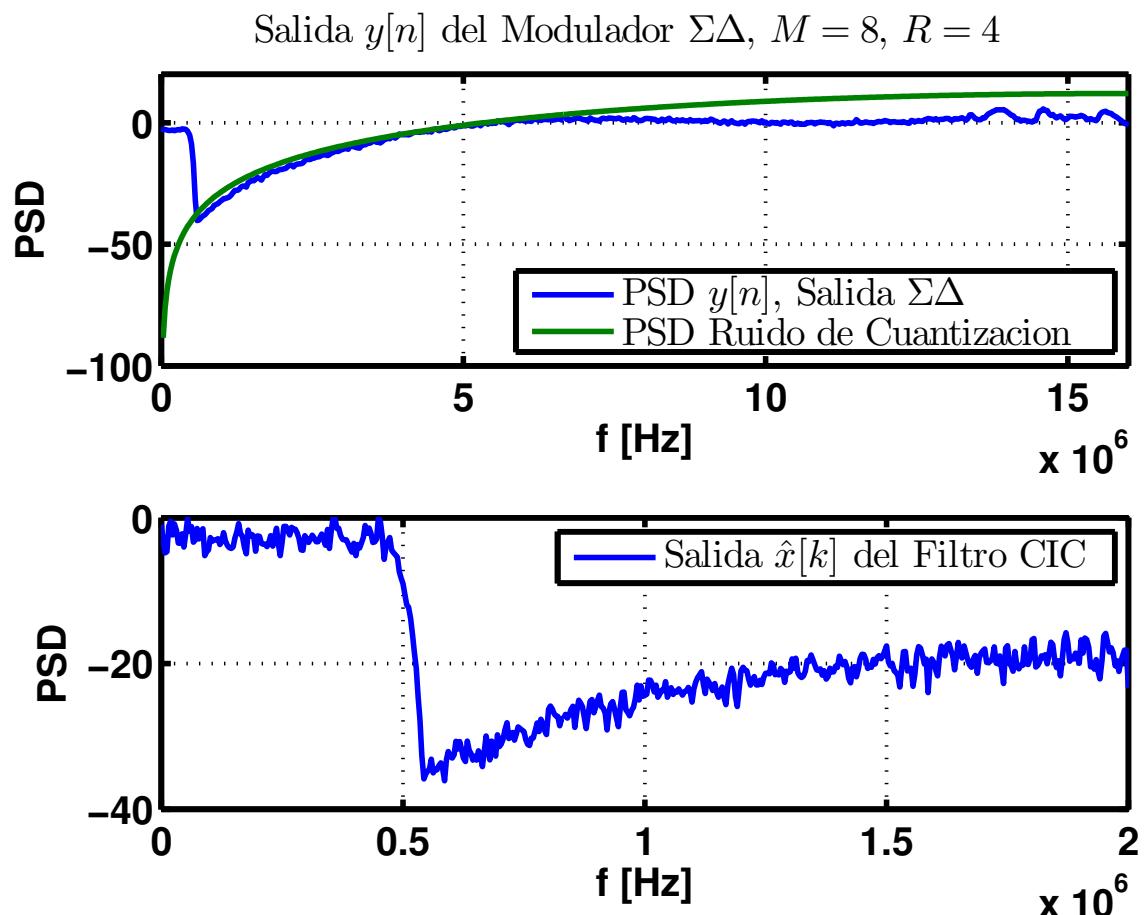
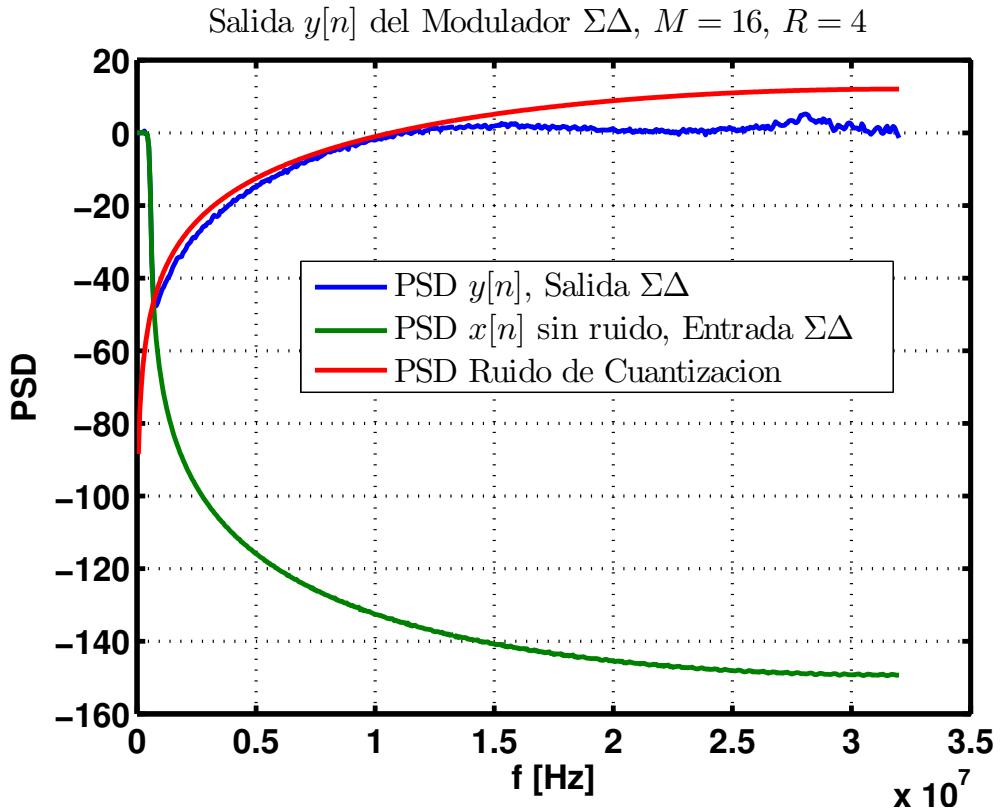
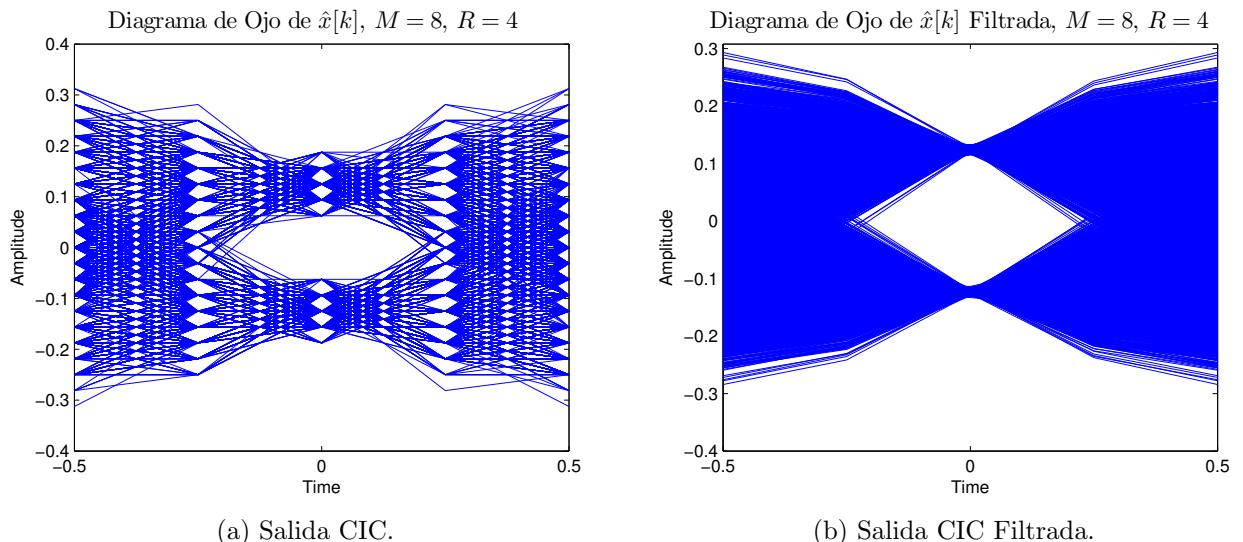


Figura 5.19: Salidas Modulador  $\Sigma\Delta$  y CIC,  $M = 8, R = 4$ .

Figura 5.20: Análisis de las diferentes señales del modulador  $\Sigma\Delta$ .Figura 5.21: Diagramas de Ojo,  $M = 8$ ,  $R = 4$ .

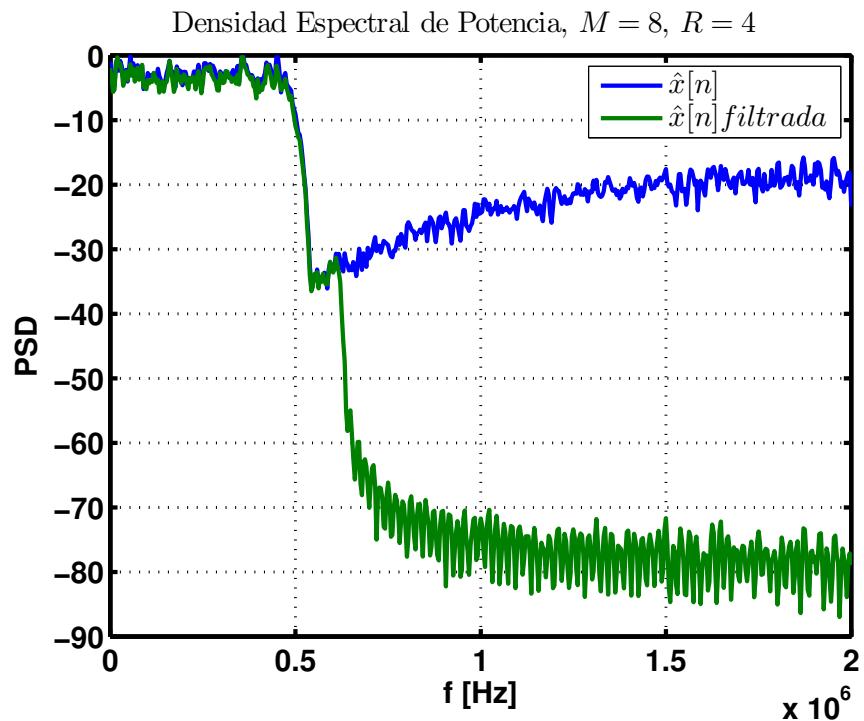


Figura 5.22: PSD Salida CIC con y sin Filtrada LPF,  $M = 8, R = 4$ .

CASO 5: Simulación con Ruido Blanco y  $M = 16, R = 4$ 

El ruido de cuantización se modela como ruido blanco con función densidad de probabilidades constante, si bien se trata de una aproximación que *a priori* puede resultar excesiva, se puede corroborar que dicha suposición es válida, comparando los resultados con cuantizador y reemplazando el cuantizador por ruido blanco. Para ello se elimina el bloque cuantizador del modulador  $\Sigma\Delta$  y se lo reemplaza por un bloque de adición de ruido blanco. Si uno observa la Figura (5.23) y la compara con (5.7), puede corroborar la similitud entre ambos resultados. En las Figuras (5.24) y (5.25), se observan resultados similares a los de las Figuras (5.9) y (5.10), respectivamente.

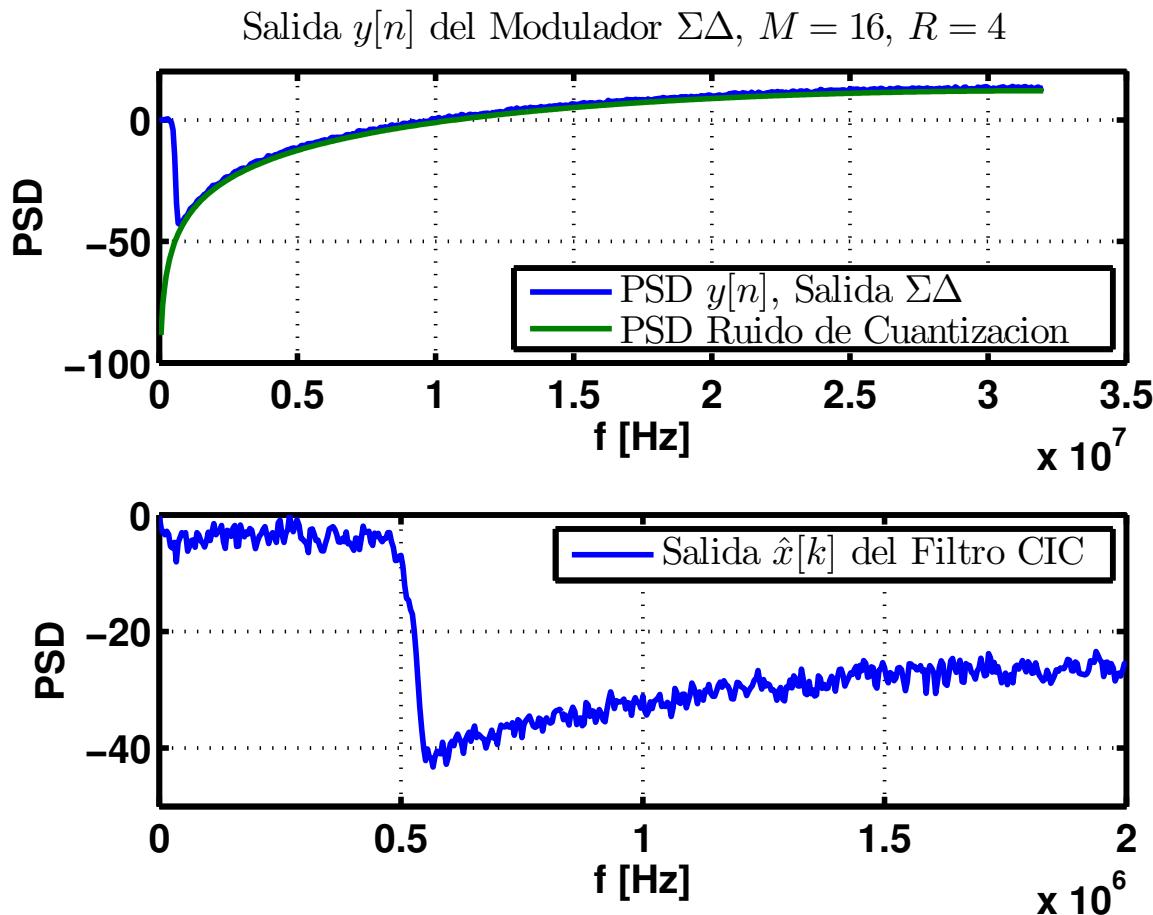
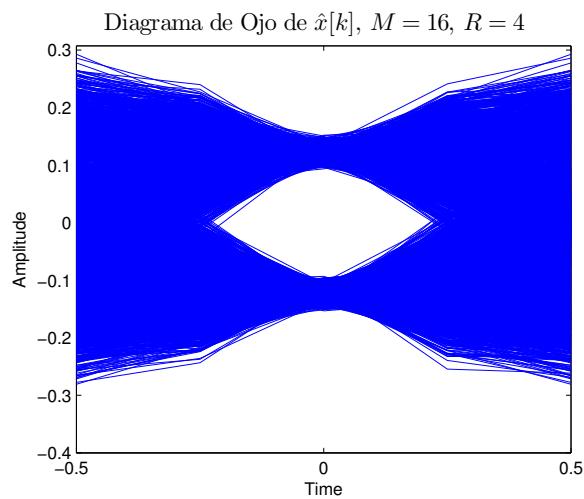
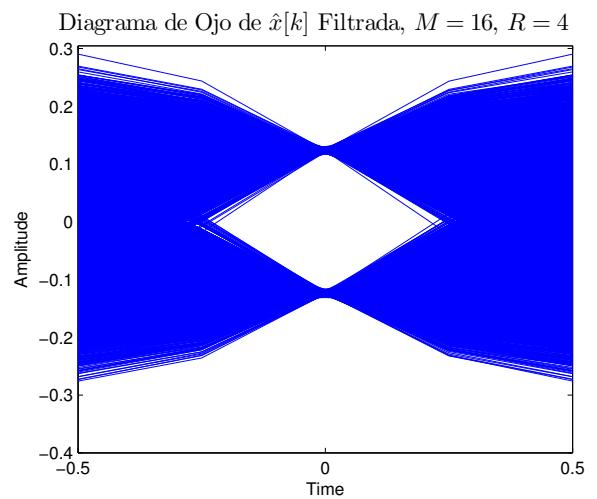


Figura 5.23: Salidas Modulador  $\Sigma\Delta$  y CIC,  $M = 16, R = 4$ .



(a) Salida CIC.



(b) Salida CIC Filtrada.

Figura 5.24: Diagramas de Ojo,  $M = 16$ ,  $R = 4$ .

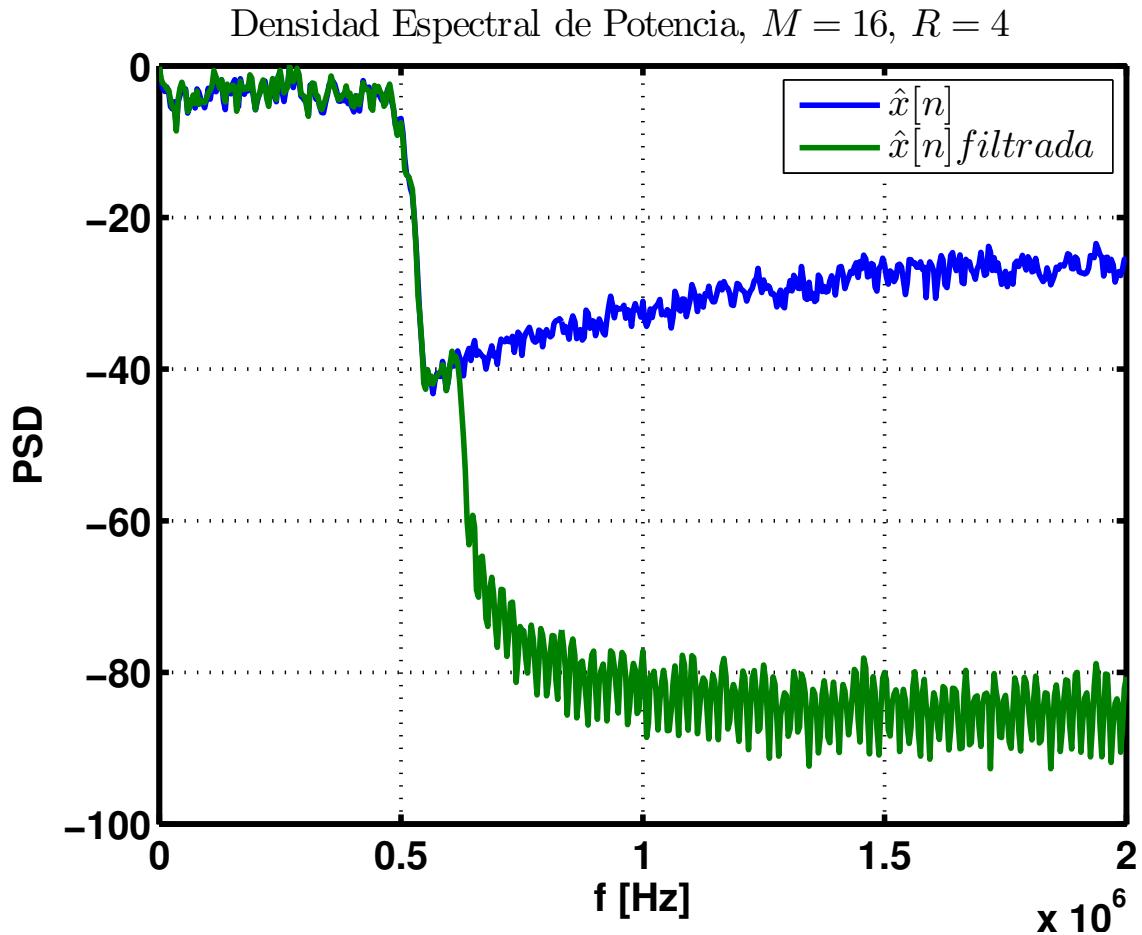


Figura 5.25: PSD Salida CIC con y sin Filtrada LPF,  $M = 16, R = 4$ .

Como los resultados obtenidos con el bloque cuantizador y reemplazando el cuantizador por un bloque que adiciona ruido blanco son muy similares, se pueden extraer dos conclusiones importantes:

1. La cuantización de una señal analógica  $x(t)$ , produce una nueva señal  $x_Q(t)$ , que es una aproximación de la señal original. Por lo tanto, el proceso de cuantización se puede ver como un proceso de degradación de la señal original  $x(t)$ .
2. Una buena forma de caracterizar un proceso de cuantización, es como la adición de ruido blanco a la señal original  $x(t)$ , produciendo una nueva señal  $x_Q(t) = x(t) + q(t)$ , donde  $q(t)$  es el ruido de cuantización.

Relación entre el Ruido de Cuantización y el Factor  $RM$ 

A continuación, en la Figura (5.26), se observa la salida del modulador  $\Sigma\Delta$  para distintos valores de sobremuestreo  $RM$ .

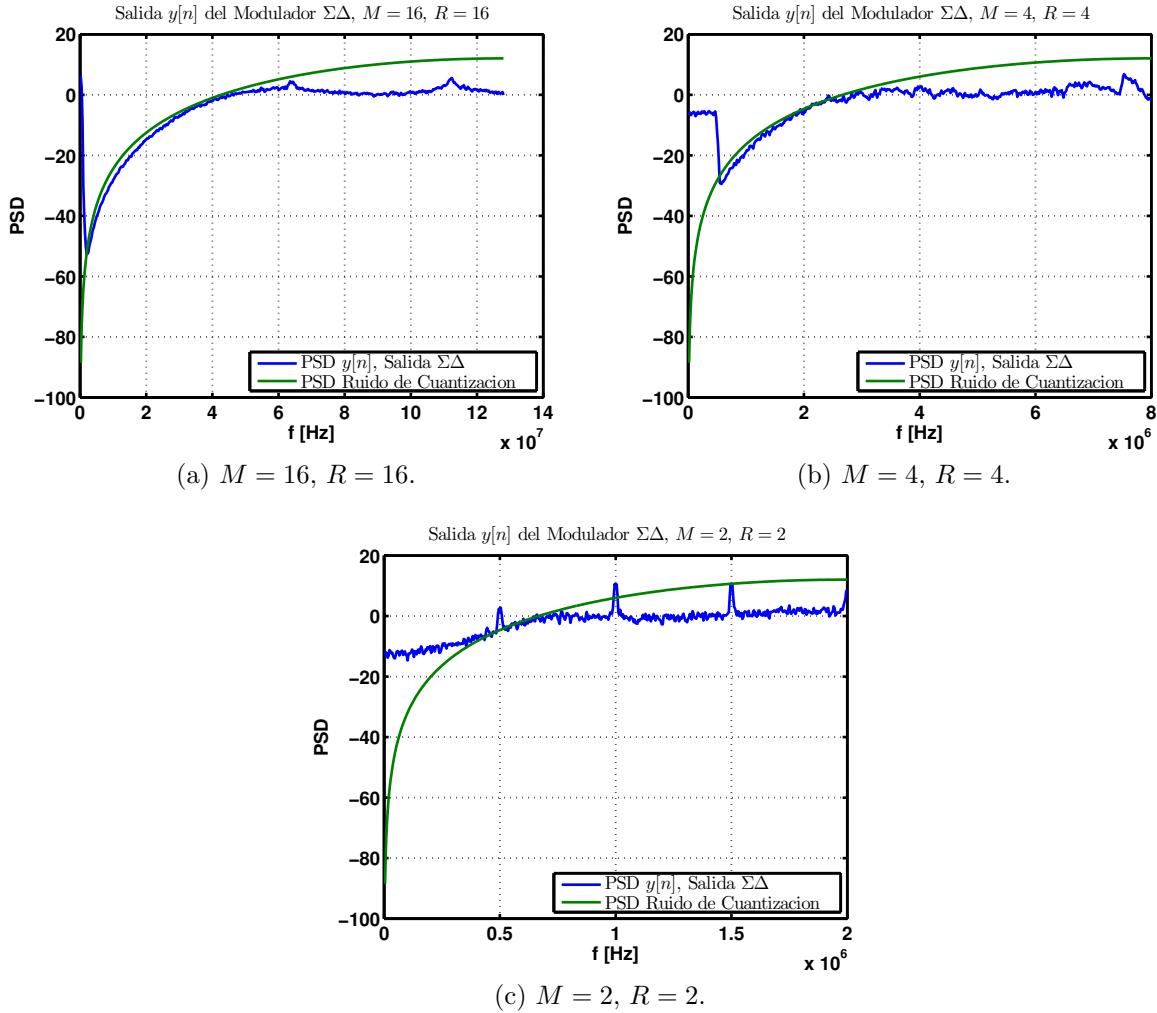


Figura 5.26: Ruido de Cuantización y Factor de Sobremuestreo.

Las señales en color verde, corresponden a las PSD de las señales de ruido de cuantización que inserta el modulador en su señal de entrada. Por otra parte, las señales en color azul corresponden a la salida cuantizada del modulador  $\Sigma\Delta$ . Como podemos ver en las tres subfiguras, si bien la PSD del ruido de cuantización mantiene el mismo factor de forma, la densidad espectral de la señal de entrada y por consiguiente de la señal de salida del modulador, varía en concordancia con el factor de sobremuestreo  $RM$  utilizado. A medida que  $RM$  aumenta, la PSD en las bajas frecuencias se comprime hacia la izquierda donde la potencia de ruido de cuantización es menor. De manera opuesta, a medida que  $RM$  disminuye, la

PSD de la señal de entrada del cuantizador, se expande hacia la derecha, donde la potencia de ruido de cuantización es mayor, lo que produce una disminución en la densidad espectral de potencia de la señal de salida del modulador.

En la subfigura (a), la PSD de la señal de salida se encuentra por encima de 0 dB en las bajas frecuencias. A medida que se disminuye  $RM$  (b), la PSD de la señal de salida del modulador es menor a 0 dB para las frecuencias bajas debido al mayor nivel de ruido. Finalmente en (c), la potencia del ruido de cuantización es mayor que la de la señal!

# Capítulo 6

## Conclusiones

Para finalizar, se listan algunas conclusiones y comentarios finales:

1. El Laboratorio N°1 permitió introducir los temas de suma de convolución, dominio de la frecuencia, filtrado, entre otros, mediante un ejemplo de un sistema básico de comunicaciones digitales. De ésta manera los conceptos matemáticos abstractos se convirtieron en un ejemplo de aplicación práctico.
2. En el Laboratorio N°2 se introdujo un nuevo filtro que se sumó al filtro de caída cosenoidal del primer laboratorio. Además, se vió una importante característica de estos filtros y de las señales analíticas, la cual permite recuperar la parte imaginaria de los símbolos originales complejos, a partir de la recepción de su parte real únicamente.
3. En el Laboratorio N°3 se profundizó un poco más en conceptos de comunicaciones digitales, para ello se presentó al Lazo de Enganche de Fase (PLL) y se analizaron sus principales características, las cuales permiten que la salida de éste dispositivo pueda seguir cambios en la fase de la señal de entrada.
4. En el Laboratorio N°4 se analizó la interacción entre el dominio discreto y el dominio continuo, para lo cual se presentó y analizó un módem.
5. En el Laboratorio N°5, se analizó un conversor analógico-digital con modulador sigma-delta y diezmado con filtro CIC, el cual es ampliamente utilizado y nos permitió analizar la interacción entre dominios y el cambio en la frecuencia de muestreo.
6. En general los laboratorios trataron ejemplos de sistemas básicos de telecomunicaciones, lo cual dió un valor agregado al curso. El hecho de salir un poco de la teoría y ver ejemplos y sistemas prácticos, brindó una buena experiencia de aprendizaje, se conocieran o no los temas de antemano.
7. Se incorporaron conceptos nuevos de telecomunicaciones y se esclarecieron muchos otros, lo cual sirvió tanto para el desarrollo del curso, como para mi área de trabajo.

8. El hecho de que los laboratorios trataran temas de comunicaciones digitales, llevo a la necesidad de leer e investigar dichos temas, por ejemplo: PLL, conversores A/D y D/A, recuperación de sincronismo, entre otros, los cuales no eran parte fundamental del contenido del curso, pero que a su vez sirvieron mucho para aplicar todos los conceptos matemáticos aprendidos.

# Bibliografía

- [1] *Discrete Time Signal Processing*, A. Oppenheim, R. Schaefer, Prentice Hall, 1999 (Versiones en Inglés y en Español).
- [2] *Fundamental of Signals and Systems Using the Web and Matlab.*, E. Kamen y B. Heck, R. Schaefer, Prentice Hall, 2007.
- [3] *Teorías de las Clases.*
- [4] *Apuntes tomados en Clases.*
- [5] *Wikipedia, entre otros.*

# Apéndices

## Apéndice A

### Algoritmo de Welch para la Estimación de la PSD

Si bien el cálculo de la PSD se realizó con las herramientas provistas por MATLAB, a continuación se realiza una breve introducción teórica, seguida del algoritmo implementado en MATLAB, para la estimación de la PSD mediante el algoritmo de Welch.

En física, ingeniería y matemáticas aplicadas, el método de Welch, llamado así por P. D. Welch, se utiliza para estimar la potencia de una señal a diferentes frecuencias, es decir, es un acercamiento a la estimación de la densidad espectral de potencia (PSD). El método se basa en el concepto de utilizar estimaciones de espectro de periodograma, que son el resultado de la conversión de una señal del dominio del tiempo, al dominio de la frecuencia.

El método de Welch es una mejora en el método estándar de estimación del espectro de periodograma y en el método de Bartlett, ya que reduce el ruido en los espectros de potencia estimados a cambio de reducir la resolución de frecuencia. Debido al ruido causado por datos imperfectos y finitos, la reducción del ruido del método de Welch es a menudo deseada.

El método Welch se basa en el método de Bartlett y difiere de dos maneras:

- La señal se divide en segmentos superpuestos, el segmento de datos original se divide en  $L$  segmentos de datos de longitud  $M$ , solapados en  $D$  puntos.
  1. Si  $D = M/2$ , se dice que la superposición es del 50 %.
  2. Si  $D = 0$ , se dice que la superposición es del 0 %. Esta es la misma situación que en el método de Bartlett.
- Los segmentos que se solapan son entonces *ventaneados*. Después de dividir los datos en segmentos superpuestos, a los segmentos de  $L$  datos individuales se les aplica una ventana (en el dominio de tiempo).

1. La mayoría de las *funciones ventana* proporcionan más influencia a los datos en el centro del conjunto, que a los datos en los bordes, lo que representa una pérdida de información. Para mitigar esa pérdida, los conjuntos de datos individuales suelen solaparse en el tiempo (como en el paso anterior).
2. El *ventaneado* de los segmentos es lo que hace que el método Welch sea un periodograma “modificado”.

Después de realizar lo anterior, el periodograma se calcula computando la transformada discreta de Fourier (FFT en MATLAB), y luego calculando la magnitud cuadrada del resultado. A continuación se promedian los periodogramas individuales, lo que reduce la varianza de las mediciones individuales de potencia. El resultado final es una matriz de *Mediciones de Potencia vs. Frecuencia* (estimación de la PSD).

## A.1. Código de MATLAB

MATLAB ya tiene implementada una función que calcula la PSD mediante el algoritmo de Welch, sin embargo el código que se presenta a continuación corresponde a una implementación *casera* de dicho algoritmo.

```

1 %% %%%%%%%%
2 % Ing. Martin Gonella %
3 % Funcion de Welsh, para calculo de PSD %
4 %%%%%%%%
5
6 function [p, f] = welch_psd(x, Fs)
7     overlap_len = 2.^7; % Longitud de Overlap o Sobreposicion
8     window_len = 2^8; % Longitud de Ventana
9     window = hamming(window_len)'; % Ventana
10    p = zeros(1,2*window_len-1);
11    n = 0;
12
13    while (length(x) >= window_len)
14        % Mientras la longitud de x sea mayor o igual que la longitud de la
15        % ventana
16
17        y = x(1:window_len).*window; % Proceso de enventanado
18        x = x(window_len - overlap_len + 1:end); % Proceso de sobreposicion
19        Ry = xcorr(y,y); % Autocorrelacion de y
20        Sy = fft(Ry); %FFT de la Autocorrelacion de y
21        p = p + Sy;
22        n = n + 1;
23    end
24
25    % Salidas de la Funcion
26    p = abs(p(1:floor(end/2)))/n;

```

```
27 f = (Fs/2)*(0:length(p)-1/length(p));  
28 end
```

# Apéndice B

## Códigos de MATLAB Utilizados

En el presente apéndice, se listan los códigos de MATLAB utilizados para el desarrollo de los diferentes laboratorios. Algunos de los códigos tienen pequeñas diferencias con respecto a los códigos originales provistos por el Dr. Ing. Mario Hueda, las cuales van desde pequeñas modificaciones, hasta agregados de código.

### B.1. Código del Laboratorio N°1

```
1 %  
2 % Procesamiento de Señales en Tiempo Discreto  
3 % Prof.: Dr. Mario Hueda  
4 % Practico Lab. 1  
5 %  
6 clear;  
7 close all;  
8 clc;  
9  
10 %  
11 % Generacion de la Respuesta al Impulso  
12 %  
13 fB = 32e9; % Velocidad de simbolos (baud rate)  
14 T = 1/fB; % Tiempo entre simbolos  
15 M = 16; % Factor de sobremuestreo  
16 fs = fB*M; % Sample rate  
17  
18 beta = .0; % Factor de roll-off  
19 L = 20; % 2*L*M+1 es el largo del filtro sobremuestreado  
20 % Entre cada simbolo transmitido se insertaran 20 muestras del filtro  
21 % de caida cosenoideal  
22  
23 t = [-L:1/M:L]*T; % El filtro esta centrado  
24 n_delay_filter = L*M; % Retardo del filtro = distancia al tap central  
25 gn = sinc(t./T).*cos(pi.*beta*t./T)./(1-(4.*beta^2*t.^2./T^2));  
26
```

```

27 figure(1)
28 h = stem(gn);
29 title(['Respueta al Impulso Filtro Caida Cosenoidal, $\beta=0.2$ y $M=16$'], '
30     interpreter,'latex','FontSize',13);
31 xlabel('$n$', 'interpreter', 'latex', 'FontSize',14);
32 ylabel('$g[n]$', 'interpreter', 'latex', 'FontSize',14)
33 grid
34 %break
35 %
36 % Calculo de la Respueta en Frecuencia
37 %
38 Omega = [0:1/2^8:1]*pi; %de 0 a pi
39 N = 1000;
40 n = [0:N];
41 index = 1;
42 for omega=Omega
43     xn = exp(j*omega*n); % Autofucion
44     yn = conv(xn,gn); % Convoluciono autofucion con filtro
45     H_Mag(index) = abs(yn(N/2)); %Toma el valor central
46     H_Fase(index) = angle(yn(N/2)*conj(xn(N/2-n_delay_filter)));
47     index = index+1;
48 end
49
50 % Nota: Omega es 1 porque esta normalizado y como el factor de
51 % sobremuestreo es 8, entonces el ancho de banda del filtro sera
52 % 1/8. Si en cambio tenemos un factor de sobremuestreo de 16,
53 % entonces el ancho de banda sera de 1/16.
54
55 figure(2)
56 subplot 211
57 h = plot(Omega/pi,H_Mag);
58 title('Magnitud', 'interpreter', 'latex', 'FontSize',14);
59 ylabel('$|H|$', 'interpreter', 'latex', 'FontSize',14)
60 xlabel('$\omega/\pi$', 'interpreter', 'latex', 'FontSize',14);
61 grid
62 subplot 212
63 h = plot(Omega/pi,H_Fase);
64 title('Fase', 'interpreter', 'latex', 'FontSize',14);
65 ylabel('$\angle(H)$', 'interpreter', 'latex', 'FontSize',14)
66 xlabel('$\omega/\pi$', 'interpreter', 'latex', 'FontSize',14);
67 grid
68 %Es un filtro pasabajos con de fase lineal
69 %break
70
71 %
72 % Generacion Simbolos
73 %
74 n_symbols = 10000;
75 ak = 2*randint(1,n_symbols)-1; % 10000 numeros 0 y 1

```

```

76 % Se multiplica x2 para tener 0 y 2, y se le resta 1
77 % para tener +1 y -1
78 xn = zeros(1,n_symbols*M); %Genero un vector con la cantidad
79 % de espacios sobremuestreados xM, para luego interpolarlo con el filtro
80 % de caida cosenoidal
81
82 xn(1:M:end) = ak; %Pongo los simbolos generados cada M posiciones
83
84 figure(3)
85 h = stem(xn(10:10+M*10)); %Muestro algunos
86 ylabel('$x[n]$', 'interpreter', 'latex', 'FontSize', 14)
87 xlabel('$n$', 'interpreter', 'latex', 'FontSize', 14)
88 title('Simbolos generados, $M=8$', 'interpreter', 'latex', 'FontSize', 14)
89 %break
90
91 %-----%
92 % Señal Transmitida
93 %-----%
94 sn = conv(xn,gn); %Convoluciono los simbolos sobremuestreados
95 %con el filtro interpolador de caida cosenoidal
96
97 figure(4)
98 h = plot(sn((2*L*M+1):(2*L*M+1)*10),'.-'); %Muestro algunos
99 ylabel('$s[n]$', 'interpreter', 'latex', 'FontSize', 14)
100 xlabel('$n$', 'interpreter', 'latex', 'FontSize', 14)
101 title('Signal Transmitida, $\beta=0.2$ y $M=4$', 'interpreter', 'latex',
102 %FontSize', 14)
103 % break
104 %-----%
105 % Generacion de Diagrama Ojo
106 %-----%
107 figure(5)
108 d = 9; %Delay para centrar el ojo
109 for m = 2*L+1:n_symbols-(2*L+1)
110     sn_p = sn(m*M+d:m*M+d+M);
111     plot([-M/2:1:M/2],sn_p)
112     hold on
113 end
114 title('Diagrama de Ojo, $\beta=0.2$ y $M=16$', 'interpreter', 'latex', 'FontSize',
115 , 14)
116 grid
117 % break
118 n=[1:length(sn)];
119
120 %figure(6)
121 %q = spectrum.welch; % Create a Welch spectral estimator.
122 %Hpsd = psd(q,exp(j*2*pi/M*n).*sn,'Fs', fs); % Calcula la PSD
123 %plot(Hpsd) % Plot la PSD.

```

```
124 %%title('Densidad Espectral de Potencia','interpreter','latex','FontSize',14)
```

## B.2. Código del Laboratorio N°2

```

1 %
2 % Procesamiento de Señales en Tiempo Discreto
3 % Prof.: Dr. Mario Hueda
4 % Practico Lab. 2
5 %
6 clear;
7 close all;
8 %
9 %
10 % Generacion de la Respuesta al Impulso
11 %
12 fB = 32e9; % Velocidad de simbolos (baud rate)
13 T = 1/fB; % Tiempo entre simbolos
14 M = 8; % Factor de sobremuestreo
15 fs = fB*M; % Sample rate
16
17 beta = 0.5001; % Factor de roll-off
18 L = 10; % 2*L*M+1 es el largo del filtro sobremuestreado
19 t = [-L:1/M:L]*T;
20 n_delay_RC_filter = L*M; % Retardo del filtro RC
21 gn = sinc(t/T).*cos(pi*beta*t/T)./(1-4*beta^2*t.^2/T.^2);
22
23 Lf=100;
24 n=[-Lf:Lf];
25 n_delay_Hilbert_filter = Lf; % Retardo del filtro de Hilbert
26 fn = 2*sin(pi*n/2).^2./(pi*n);
27 fn(Lf+1)=0;
28
29 figure(1)
30 subplot 211
31 h = stem(gn);
32 title('Respuesta al Impulso del Filtro Coseno Realzado , $\backslash beta=0.5$', '
    interpreter','latex','FontSize',14);
33 xlabel('$n$', 'interpreter','latex','FontSize',14);
34 ylabel('$g[n]$', 'interpreter','latex','FontSize',14);
35 axis([0 length(gn) -0.5 1])
36 grid on
37 subplot 212
38 h = stem(fn);
39 title('Respuesta al Impulso del Filtro Transformador de Hilbert ', 'interpreter' ,
    'interpreter','latex','FontSize',14);
40 xlabel('$n$', 'interpreter','latex','FontSize',14);
41 ylabel('$f[n]$', 'interpreter','latex','FontSize',14);
42 axis([0 length(fn) -1 1])
43 grid on
44 %break
45 %
46 %

```

```

47 % Calculo de la Respuesta en Frecuencia
48 %
49 Omega = [-1:1/2^8:1]*pi;
50 N = 1000;
51 n = [0:N];
52 index = 1;
53 for omega=Omega
54     xn = exp(j*omega*n);
55     yn = conv(xn,fn);
56     H_Mag(index) = abs(yn(N/2));
57     H_Fase(index) = angle(yn(N/2)*conj(xn(N/2-n_delay_Hilbert_filter)));
58     index = index+1;
59 end
60
61 figure(2)
62 subplot 211
63 h=plot(Omega/pi,H_Mag);
64 title('Magnitud Transformador de Hilbert','interpreter','latex','FontSize',14)
65 ;
66 ylabel('$|H|$','interpreter','latex','FontSize',14)
67 xlabel('$\omega/\pi$','interpreter','latex','FontSize',14);
68 grid
69 subplot 212
70 h=plot(Omega/pi,H_Fase);
71 title('Fase Transformador de Hilbert','interpreter','latex','FontSize',14);
72 ylabel('angle(H)','interpreter','latex','FontSize',14)
73 xlabel('$\omega/\pi$','interpreter','latex','FontSize',14);
74 grid
75 %break
76 %
77 % Generacion Simbolos
78 %
79 n_symbols = 2^14;
80 ak = 2*randint(1,n_symbols)-1+j*(2*randint(1,n_symbols)-1);
81 xn = zeros(1,n_symbols*M);
82 xn(1:M:end) = ak;
83
84 figure(3)
85 h = plot(ak,'r*');
86 title('Simbolos Complejos Generados (4-QAM)','interpreter','latex','FontSize',14);
87 xlabel('R($a_k$)','interpreter','latex','FontSize',14)
88 ylabel('I($a_k$)','interpreter','latex','FontSize',14);
89 axis([-1.2 1.2 -1.2 1.2])
90 grid on
91 %break
92 %
93 % Señal Banda-Base

```

```

95 %
96 yn = conv(xn,gn); % Convolucion señal de datos y filtro coseno realizado
97 figure(4)
98 q = spectrum.welch;
99 Hpsd = psd(q,yn,'nfft',1024,'CenterDC',true);
100 h=plot(Hpsd);
101 title('PSD Signal en Banda Base, $y[n]=x[n]*g[n]$','interpreter','latex','FontSize',14)
102 %break
103 %
104 %
105 % Señal Modulada (Analitica)
106 %
107 %
108 n=[1:length(yn)];
109 Omega_c=3*pi/M; % Portadora arbitraria para generar señal
110 % analitica (ojo: depende de M)
111 carrier=exp(j*Omega_c*n);
112 sn = yn.*carrier;
113
114 figure(5)
115 q = spectrum.welch;
116 Hpsd = psd(q,sn,'nfft',1024,'CenterDC',true);
117 h=plot(Hpsd);
118 title('PSD Signal Modulada (Analitica), $\Omega_0=3 \pi/M$','interpreter','latex','FontSize',14)
119 break
120 %
121 %
122 % Señal Transmitida (Parte Real de la Señal Analitica)
123 %
124 % Demostracion: Im{s[n]}=Re{s[n]}\ast f[n]
125 sn_r = real(sn); % Re{s[n]}
126 figure(6)
127 q = spectrum.welch;
128 Hpsd = psd(q,sn_r,'nfft',1024,'SpectrumType','twosided','CenterDC',true);
129 h=plot(Hpsd);
130 title('PSD Signal Transmitida (Parte Real de $s[n]$)','interpreter','latex','FontSize',14)
131 %break
132 %
133 %
134 % Filtro de Particion de Fase
135 %
136 sn_i = conv(sn_r,fn); % Re{s[n]}\ast f[n]
137 sn_hat = sn_r+j*sn_i(Lf+1+0:end-Lf+0);
138 figure(7)
139 q = spectrum.welch;
140 Hpsd = psd(q,sn_hat,'nfft',1024,'SpectrumType','twosided','CenterDC',true);
141 h=plot(Hpsd);

```

```
142 title('PSD Signal a la Salida del Filtro de Particion de Fase','interpreter','  
143     latex','FontSize',14)  
143 %break  
144  
145 %  
146 %Comparacion Partes Imaginarias (transmitida y recibida)  
147 %%  
148 sn_i = conv(sn_r,fn);  
149 sn_hat = sn_r+j*sn_i(Lf+1:end-Lf);  
150 figure(8)  
151 n=[1:100]+1000; %100 puntos de ventana de tiempo arbitraria  
152 h=plot(n,imag(sn(n)),'b',n,sn_i(n+Lf),'ro');  
153 legend('Transmitida','Recuperada','interpreter','latex','FontSize',14);  
154 title('Parte Imaginaria de la Signal Analitica','interpreter','latex','  
155     FontSize',14)  
156 %break  
157 %%  
158 %%  
159 % Señal Demodulada (Banda-Base)  
160 %%  
161 n=[1:length(sn_hat)];  
162 carrier=exp(j*Omega_c*n);  
163 yn_hat = sn_hat.*conj(carrier); %portadora conjugada (demodulacion)  
164 figure(9)  
165 q = spectrum.welch;  
166 Hpsd = psd(q,yn_hat,'nfft',1024,'SpectrumType','twosided','CenterDC',true);  
167 h=plot(Hpsd);  
168 title('PSD Signal Demodulada (Banda-Base)','interpreter','latex','FontSize'  
169     ,14)  
169 %break
```

### B.3. Código del Laboratorio N°2 - Parte B

```

1 %
2 % Procesamiento de Señales en Tiempo Discreto
3 % Prof.: Dr. Mario Hueda
4 % Practico Lab. 2 – Serie de Fourier
5 %
6 clear;
7 close all;
8
9 %
10 % Generacion de la Respuesta al Impulso
11 %
12 fB = 32e9; % Velocidad de simbolos (baud rate)
13 T = 1/fB; % Tiempo entre simbolos
14 M = 8; % Factor de sobremuestreo
15 fs = fB*M; % Sample rate
16
17 beta = .04001; % Factor de roll-off
18 L = 40; % 2*L*M+1 es el largo del filtro sobremuestreado
19 t = [-L:1/M:L]*T;
20 n_delay_filter = L*M; % Retardo del filtro
21 gn = sinc(t/T).*cos(pi*beta*t/T)./(1-4*beta^2*t.^2/T.^2); %coseno realzado
22
23 figure(1)
24 h = stem(gn);
25 title('Respuesta al Impulso','interpreter','latex','FontSize',14);
26 xlabel('$n$','interpreter','latex','FontSize',14);
27 ylabel('$g[n]$','interpreter','latex','FontSize',14);
28 axis([0 length(gn) -0.3 1])
29 grid
30 %break
31
32 %
33 % Calculo de la Respuesta en Frecuencia
34 %
35 Omega = [0:1/2^8:1]*pi;
36 N = 1000;
37 n = [0:N];
38 index = 1;
39 for omega=Omega
40     xn = exp(j*omega*n);
41     yn = conv(xn,gn);
42     H_Mag(index) = abs(yn(N/2));
43     H_Fase(index) = angle(yn(N/2)*conj(xn(N/2-n_delay_filter)));
44     index = index+1;
45 end
46
47 figure(2)
48 subplot 211

```

```

49 h=plot(Omega/pi,H_Mag);
50 title('Magnitud','interpreter','latex','FontSize',14);
51 ylabel('$|H|$', 'interpreter', 'latex', 'FontSize',14)
52 xlabel('$\omega/\pi$', 'interpreter', 'latex', 'FontSize',14);
53 grid
54 subplot 212
55 h=plot(Omega/pi,H_Fase);
56 title('Fase','interpreter','latex','FontSize',14);
57 ylabel('angle(H)', 'interpreter', 'latex', 'FontSize',14)
58 xlabel('$\omega/\pi$', 'interpreter', 'latex', 'FontSize',14);
59 grid
60 %break
61
62 %—————
63 % Generacion Simbolos
64 %
65 n_symbols = 100000;
66 ak = 2*randint(1,n_symbols)-1+j*(2*randint(1,n_symbols)-1);
67 xn = zeros(1,n_symbols*M);
68 xn(1:M:end) = ak;
69 bk = 2*randint(1,n_symbols)-1+j*(2*randint(1,n_symbols)-1);
70 xxn = zeros(1,n_symbols*M);
71 xxn(1:M:end) = bk;
72
73 figure(3)
74 h = plot(ak,'r*');
75 title('Simbolos Generados','interpreter','latex','FontSize',14);
76 xlabel('R(a_k)'), 'interpreter', 'latex', 'FontSize',14)
77 ylabel('I(a_k)'), 'interpreter', 'latex', 'FontSize',14);
78 axis([-1.2 1.2 -1.2 1.2])
79 grid on
80 %break
81
82 %—————
83 % Señal Transmitida
84 %
85 pn = conv(xn,gn);
86 qn = conv(xxn,gn); %segunda señal independiente
87 figure(4)
88 subplot 211
89 h = plot(real((pn((2*L*M+1):(2*L*M+1)*10)),'.-'));
90 title('Signal Transmitida (Parte Real)','interpreter','latex','FontSize',14);
91 ylabel('$p[n]$', 'interpreter', 'latex', 'FontSize',14)
92 xlabel('$n$', 'interpreter', 'latex', 'FontSize',14);
93 axis([0 length((2*L*M+1):(2*L*M+1)*10) -3 3])
94 subplot 212
95 v = plot(imag((pn((2*L*M+1):(2*L*M+1)*10)),'.-'));
96 title('Signal Transmitida (Parte Imaginaria)','interpreter','latex','FontSize',
97 ,14);
98 ylabel('$p[n]$', 'interpreter', 'latex', 'FontSize',14)

```

```
98 xlabel( '$n$', 'interpreter', 'latex', 'FontSize', 14);
99 axis([0 length((2*L*M+1):(2*L*M+1)*10) -3 3])
100 %break
101
102 yn=pn+.0*[zeros(1,M/2) qn(1:end-M/2)];
103 %yn=[zeros(1,M/2) qn(1:end-M/2)];
104
105 %yn=qn;
106 %s1=pn+1.0*[zeros(1,M/2) pn(1:end-M/2)];
107 %s2=conj([zeros(1,M/2) pn(1:end-M/2)]+[zeros(1,M) pn(1:end-M)]);
108
109 Z=abs(yn).^4;
110 %Z=s1.*s2;
111 figure(5)
112 [pxx w]=pwelch(Z,1024,512,1024*2);
113 l=find(w<2*pi/M);
114 G=max(10*log10(sqrt(pxx(1(end)-0:1(end)+5))));
115 plot(w,10*log10(sqrt(pxx)))
116 hold on
117 plot(2*pi/M,G, 'r*')
118 title('Tono de Pulsacion en $\Omega=2\pi/M=1.5708$, con $M=4$', 'interpreter',
        ' latex', 'FontSize', 14)
119 xlabel('$\Omega$', 'interpreter', 'latex', 'FontSize', 14)
120 ylabel('Espectro de $|p[n]|^2$ [dB]', 'interpreter', 'latex', 'FontSize', 14)
121 legend('Espectro de la Secuencia', 'Tono de Pulsacion')
```

## B.4. Código Laboratorio N°3

```

1 %
2 % Procesamiento de Señales en Tiempo Discreto
3 % Prof.: Dr. Mario Hueda
4 % Practico Lab. 3
5 %
6 clear all;
7 close all;
8 format short
9 clc
10
11 ON=1;
12 OFF=0;
13 Bl=.40e6;           % Ancho de banda, en Hz
14 fs=400e6;           % Frecuencia de muestreo, in Hz
15 D=10;                % Latencia en ciclos de reloj de muestreo
16
17 %
18 % Evaluacion de la Funcion de Transferencia
19 %
20 omega=[0.0001:0.0001:1]*pi;
21 z=exp(j*omega);
22
23 Kp=Bl/fs*3*pi          % Ganancia proporcional
24 Ki=Kp*1e-3*1.0          % Ganancia integral
25 Relacion_Kp_Ki=Kp/Ki    % Relacion Kp/Ki
26
27 % Respuestas de lazo abierto y cerrado
28 L1z=z.^(-D)*Kp;
29 L2z=z.^(-D).*(Kp+Ki*z.^(-1)./(1-z.^(-1)));
30 G1z=L1z.*z.^(-1)./(1-z.^(-1));
31 G2z=L2z.*z.^(-1)./(1-z.^(-1));
32 H1z=G1z./(1+G1z);
33 H2z=G2z./(1+G2z);
34
35 % Funcion de transferencia (en funcion de "f" de tiempo continuo)
36 figure()
37 h=semilogx(omega*fs/2/pi,20*log10(abs(H1z)), 'b', omega*fs/2/pi,20*log10(abs(H2z)), 'r');
38 set(h,'LineWidth',3);
39 set(h,'MarkerSize',16);
40 set(gca,'XScale','log','YScale','lin','FontWeight','bold','FontSize',14);
41 set(gca,'LineWidth',2);
42 xlabel('f [Hz]');
43 ylabel('|H(j\omega)|');
44 rr=sprintf('Funcion de Transferencia,\n K_p=%s y K_i=%s, K_p/K_i=%s\n Latencia =%s', num2str(Kp), num2str(Ki), num2str(Kp/Ki), num2str(D));
45 rr=sprintf('Funcion de Transferencia, K_p=%s y K_i=%s, Latencia=%s', num2str(Kp), num2str(Ki), num2str(D));

```

```

46 title(rr)
47 legend('L_1(z)=Proporcional','L_2(z)=Prop+Integral','best')
48 grid on
49 axis([2*10^4 fs/2 -30 15])
50 %break
51
52 %_____%
53 % Evaluacion de Polos y Ceros con Latencia
54 %_____
55 %%%%%
56 %LAZO 1 %
57 %%%%%
58 if D>1
59 Den=[1 -1 zeros(1,D-1) Kp];
60 else
61     if D==1
62         Den=[1 -1 Kp];
63     else
64         Den=[1 -1+Kp];
65     end
66 end
67 Num=zeros(1,D+1) Kp];
68 figure()
69 h=zplane(Num,Den);
70 set(h,'Linewidth',2);
71 set(h,'Markersize',12);
72 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
73 set(gca,'Linewidth',2);
74 xlabel('Real');
75 ylabel('Imag');
76 str = sprintf('Estabilidad L_1(z), D=%s, K_p=%s',num2str(D),num2str(Kp));
77 title(str)
78 grid on
79
80 texto = sprintf('Para L1(z) con D=%d y Kp=%d el sistema es: ', D, Kp);
81 if isempty(find(abs(roots(Den))>=1.0))
82     texto = [texto 'ESTABLE'];
83 else
84     texto = [texto 'INESTABLE'];
85 end
86 disp(texto)
87 %break
88
89 %Genera graficas variando D y Kp con L1(z)
90 D=[0 20 100];
91 for i=1:3
92     for Kp=-3:0.5:3
93         figure()
94         if D(i)>1
95             Den=[1 -1 zeros(1,D(i)-1) Kp];

```

```

96     else
97         if D(i)==1
98             Den=[1 -1 Kp];
99         else
100            Den=[1 -1+Kp];
101        end
102    end
103    Num=[zeros(1,D(i)+1) Kp];
104    h=zplane(Num,Den);
105    set(h,'Linewidth',2);
106    set(h,'Markersize',12);
107    set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',
108        ,14);
109    set(gca,'Linewidth',2);
110    xlabel('Real');
111    ylabel('Imag');
112    str = sprintf('Analisis Estabilidad L_1(z), D=%s, K_p=%s',num2str(D(i))
113        ,num2str(Kp));
114    str_pr = sprintf('Estabilidad_L1_D=%s_K_p=%s',num2str(D(i)),num2str(Kp
115        ));
116    title(str)
117    grid on
118
119    texto = sprintf('Para L1(z) con D=%d y Kp=%d, el sistema es: ', D(i),
120        Kp);
121    if isempty(find(abs(roots(Den))>=1.0))
122        texto = [texto 'ESTABLE'];
123    else
124        texto = [texto 'INESTABLE'];
125    end
126    disp(texto)
127
128
129
130 if D>1
131     Den=[1 -2 1 zeros(1,D-2) Kp Ki-Kp];
132     Num=[zeros(1,D+1) Kp];
133 elseif D==1
134     Den=[1 -2 Kp+1 Ki-Kp];
135     Num=[0 0 Kp Ki-Kp];
136 elseif D==0
137     Den=[1 Kp-2 Ki-Kp+1];
138     Num=[0 Kp Ki-Kp];
139 end
140 figure()
141 h=zplane(Num,Den);

```

```
142 set(h, 'LineWidth', 2);
143 set(h, 'MarkerSize', 12);
144 set(gca, 'XScale', 'lin', 'YScale', 'lin', 'FontWeight', 'bold', 'FontSize', 14);
145 set(gca, 'LineWidth', 2);
146 xlabel('Real');
147 ylabel('Imag');
148 str = sprintf('Analisis Estabilidad L_2(z), D=%s, K_p=%s, K_i=%s', num2str(D),
    num2str(Kp), num2str(Ki));
149 title(str)
150 grid on
151
152 texto = sprintf('Para L2(z) con D=%d, Kp=%d y Ki=%d, el sistema es: ', D, Kp,
    Ki);
153 if isempty(find(abs(roots(Den))>=1.0))
    texto = [texto 'ESTABLE'];
154 else
    texto = [texto 'INESTABLE'];
155 end
156 disp(texto)
% break
160
161 % Genera graficas variando D, Kp y Ki
162 D=[0 20 100];
163 for i=1:3
    for Kp=-3:1:3
        if D(i)>1
            Den=[1 -2 1 zeros(1,D(i)-2) Kp Ki-Kp];
            Num=zeros(1,D(i)+1) Kp];
        elseif D==1
            Den=[1 -2 Kp+1 Ki-Kp];
            Num=[0 0 Kp Ki-Kp];
        elseif D(i)==0
            Den=[1 Kp-2 Ki-Kp+1];
            Num=[0 Kp Ki-Kp];
        end
        figure()
        h=zplane(Num,Den);
        set(h, 'LineWidth', 2);
        set(h, 'MarkerSize', 12);
        set(gca, 'XScale', 'lin', 'YScale', 'lin', 'FontWeight', 'bold', 'FontSize',
        14);
        set(gca, 'LineWidth', 2);
        xlabel('Real');
        ylabel('Imag');
        str = sprintf('Analisis Estabilidad L_2(z), D=%s, K_p=%s, K_i=%s',
        num2str(D(i)), num2str(Kp), num2str(Ki));
        title(str)
        grid on
186
```

```

187     texto = sprintf('Para L2(z) con D=%d, Kp=%d y Ki=%d, el sistema es: ', 
188     D(i), Kp, Ki);
189     if isempty(find(abs(roots(Den))>=1.0))
190         texto = [texto 'ESTABLE'];
191     else
192         texto = [texto 'INESTABLE'];
193     end
194     disp(texto)
195
196 end
197 %break
198
199 %=====
200 % Simulacion de PLL con Detector de Fase Ideal
201 %(incluye ruido)
202 %
203 f0=0.10e6; %Frecuencia de Tono de Jitter
204 A=.0001; %Amplitud de Tono de Jitter
205 SNR_TRD=38;
206 SNR_TR=10^(SNR_TRD/10);
207 Omega_0=2*pi*.20; %Offset de frecuencia
208 TEST_JIT_NOISE=1; % 1: test de jitter senoidal+ruido / 0: test de offset de
209 %frecuencia
210 Av=1;
211
212 N=1e7;
213 pedel=zeros(1,D);
214 pe_v=zeros(1,N);
215 sum_pe_v=zeros(1,N);
216 tita_n_v=zeros(1,N);
217 phi_n_v=zeros(1,N);
218 noise=sqrt(.5)*randn(1,N);
219 sum_pesd=0;
220 phi_n=0;
221 for n=1:N
222     if TEST_JIT_NOISE==OFF
223         tita_n=Omega_0*n; % Offset de Frecuencia Lineal
224         %tita_n=Omega_0; % Offset de Frecuencia Constante
225     else
226         tita_n=2*pi*A*cos(2*pi*f0*n/fs)+noise(n)/sqrt(SNR_TR);
227     end
228     pe=tita_n-phi_n;
229     if (D>0)
230         pesd=pedel(end);
231         pedel=[pe,pedel(1:end-1)];
232     else
233         pesd=pe;
234     end

```

```

235 phi_n_v(n)=phi_n;
236 c_n=Kp*pesd+Ki*sum_pesd*1.0;
237 phi_n=phi_n+c_n;
238 sum_pesd=sum_pesd+pesd;
239 pe_v(n)=pe;
240 sum_pe_v(n)=sum_pesd;
241 tita_n_v(n)=tita_n;
242 end
243
244 figure()
245 h=plot([1:length(phi_n_v)]/fs,phi_n_v);
246 set(h,'LineWidth',3);
247 set(h,'MarkerSize',16);
248 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
249 set(gca,'LineWidth',2);
250 xlabel('t');
251 ylabel('Salida del PLL (\phi_n)');
252 %str = sprintf('Offset de Frecuencia Lineal, D=%s, K_p=%s, K_i=%s, Omega_0=%s
253 %s',num2str(D),num2str(Kp),num2str(Ki),num2str(Omega_0));
254 %str = sprintf('Offset de Frecuencia Constante, D=%s, K_p=%s, K_i=%s, Omega_0=%s
255 str = sprintf('Jitter Senoidal+Ruido, D=%s, K_p=%s, K_i=%s, Omega_0=%s',
256 num2str(D),num2str(Kp),num2str(Ki),num2str(Omega_0));
257 title(str)
258 grid on;
259
260 if TEST_JIT_NOISE==ON
261 res_jitter=phi_n_v;
262 res_jitter=res_jitter(5e5:end);
263 res_jitter=res_jitter-mean(res_jitter);
264
265 figure()
266 subplot 211
267 h=plot([1:length(res_jitter)]/fs,res_jitter/2/pi);
268 set(h,'LineWidth',3);
269 set(h,'MarkerSize',16);
270 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
271 set(gca,'LineWidth',2);
272 %axis([10^4,10^10,-360,360]);
273 xlabel('t');
274 ylabel('Salida del PLL (x 1/(2\pi))');
275 str = sprintf('Jitter Senoidal+Ruido, D=%s, K_p=%s, K_i=%s',num2str(D),
276 num2str(Kp),num2str(Ki));
277 title(str)
278 grid on;
279 subplot 212
280 [H,F]=psd(res_jitter,2^18,fs);
281 m=find(F>=f0);
282 A_est=sqrt(sum(H(m(1)-2:m(1)+2))/(4*pi^2)*2/length(H));
283 H_est_fc_dB=20*log10(A_est/A);

```

```

281 h=semilogx(F,10*log10(H/max(H))) ;
282 set(h,'LineWidth',2);
283 set(h,'MarkerSize',16);
284 set(gca,'XScale','log','YScale','lin','FontWeight','bold','FontSize',14);
285 set(gca,'LineWidth',2);
286 axis([10^4,10^10,-50,0]);
287 xlabel('f [Hz]');
288 ylabel('PSD');
289 grid on
290
291 figure()
292 [H F]=psd(res_jitter,2^18,fs);
293 m=find(F>=f0);
294 A_est=sqrt(sum(H(m(1)-2:m(1)+2))/(4*pi^2)*2/length(H));
295 H_est_fc_dB=20*log10(A_est/A)
296 h=semilogx(F,10*log10(H/max(H)));
297 set(h,'LineWidth',2);
298 set(h,'MarkerSize',16);
299 set(gca,'XScale','log','YScale','lin','FontWeight','bold','FontSize',14);
300 set(gca,'LineWidth',2);
301 axis([10^4,0.5*10^8,-50,0]);
302 xlabel('f [Hz]');
303 ylabel('PSD');
304 title('PSD de la $\phi[n]$, para una entrada de Jitter','interpreter',...
305 'latex')
306 str = sprintf('f_0=%s, A=%s',num2str(f0),num2str(A));
307 legend(str)
308 end
309 %break
310
311 figure()
312 subplot 311
313 h=plot(pe_v/2/pi);
314 set(h,'LineWidth',2);
315 set(h,'MarkerSize',16);
316 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
317 set(gca,'LineWidth',2);
318 %axis([0,5*10^6,-20,40]);
319 xlabel('n');
320 ylabel('Error de Fase, \epsilon[n] (/2\pi)');
321 %str = sprintf('Offset de Frecuencia Lineal, D=%s, K_p=%s, K_i=%s, Omega_0=%s',...
322 %',num2str(D),num2str(Kp),num2str(Ki),num2str(Omega_0));
323 %str = sprintf('Offset de Frecuencia Constante, D=%s, K_p=%s, K_i=%s, Omega_0=%s',...
324 %',num2str(D),num2str(Kp),num2str(Ki),num2str(Omega_0));
325 str = sprintf('Jitter Senoidal+Ruido, D=%s, K_p=%s, K_i=%s, Omega_0=%s',...
326 num2str(D),num2str(Kp),num2str(Ki),num2str(Omega_0));
327 title(str)
328 grid on;
329
330 subplot 312

```

```
327 h=plot(pe_v*Kp/2/pi);
328 set(h,'LineWidth',2);
329 set(h,'MarkerSize',16);
330 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
331 set(gca,'LineWidth',2);
332 %axis([0,5*10^6,-0.1,0.2]);
333 xlabel('n');
334 ylabel('Parte Proporcional (/2\pi)');
335 grid on;
336
337 subplot 313
338 h=plot(sum_pe_v*Ki/2/pi);
339 set(h,'LineWidth',2);
340 set(h,'MarkerSize',16);
341 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
342 set(gca,'LineWidth',2);
343 %axis([0,5*10^6,0,0.4]);
344 xlabel('n');
345 ylabel('Parte Integral (/2\pi)');
346 grid on;
```

## B.5. Código del Lab. N°3 - Jitter Transfer Teórico

```

1 %
2 % Procesamiento de Señales en Tiempo Discreto
3 % Prof.: Dr. Mario Hueda
4 % Practico Lab. 3 – Jitter Transfer Teorico
5 %
6
7 % El presente codigo permite graficar las curvas de transferencia
8 % de Jitter de un PLL con filtro de lazo Proporcional+Integral
9
10 clear all;
11 close all;
12 clc;
13
14 D = [0 10 100]; % Latencias
15 Fm = 1/32e9; % 32 GBaudios, frecuencia de muestreo
16 Tm = 1/Fm; % Tiempo de muestreo
17
18 % Constantes Kp y Ki, se varian para ver sus influencias
19 Kp = [0.0005 0.002 0.003 0.004 0.005 0.006 0.007 0.008 0.01]; % Constante
    Proporcional
20 Ki = Kp./1000; % Constante Integral , en funcion de Kp
21
22 for l = 1:length(D)
    % Funcion de transferencia en "z"
23     z = tf('z', Fm);
24     VCO = (1/z)/(1-(1/z)); % TF del VCO
25     w = logspace(6,11,300); % 300 puntos en 10^6 y 10^10
26     freq = w./(2*pi); % [Hz]
27     for m=1:length(Kp)
28         L = (1/(z^D(l)))*(Kp(m) + Ki(m)*((1/z)/(1-(1/z))));
29         F = VCO*L;
30         H = feedback(F, 1); % Funcion de Lazo Cerrado
31         [mag, phase] = bode(H, w); % Bode
32
33         magabs = 20.*log10(abs(mag));
34         magabs = squeeze(magabs);
35         h(:, m) = magabs; % Magnitudes en dB, TF jitter , para varios Kp
36
37         denominador = H.den{1}; % Extraigo el denominador
38         polos = abs(roots(denominador)); % Saco las raices del denom (polos)
39         palabra = sprintf('Con Latencia=%d, Kp=%d y Ki=%d, el sistema es: ', D
            (l),Kp(m),Ki(m));
40         if isempty(find(pulos>=1.0)) % Busco polos afuera del circulo unidad
41             palabra = [palabra 'ESTABLE'];
42         else
43             palabra = [palabra 'INESTABLE'];
44         end
45         disp(palabra) % Muestro el string
46

```

```
47 end
48
49 % Grafico las Curvas de Jitter Transfer
50 figure()
51 semilogx(frec,h)
52 xlabel('Frecuencia [Hz]');
53 ylabel('Ganancia [dB]');
54 grid on
55 legend('K_p=0.0005','K_p=0.002','K_p=0.003','K_p=0.004',...
56 'K_p=0.005','K_p=0.006','K_p=0.007','K_p=0.008','K_p=0.009',...
57 , 'Location','southwest')
58 if D(1)==0
59     title('Funcion Transferencia de Jitter con Latencia=0, $K_i=K_p/1000$',...
60 , 'interpreter','latex','fontsize',14)
61 elseif D(1)==10
62     title('Funcion Transferencia de Jitter con Latencia=10, $K_i=K_p/1000$',...
63 , 'interpreter','latex','fontsize',14)
64 else
65     title('Funcion Transferencia de Jitter con Latencia=100, $K_i=K_p/1000$',...
66 , 'interpreter','latex','fontsize',14)
67 end
68 axis([10e4 10e9 -70 10])
69
```

## B.6. Código Laboratorio N°4

```

1 %
2 % Procesamiento de Señales en Tiempo Discreto
3 % Prof.: Dr. Mario Hueda
4 % Practico Lab. 4
5 %
6 clear;
7 close all;
8 clc;
9 format compact
10 %
11 %
12 % Generacion de la Respuesta al Impulso
13 %
14 fB = 32e9; % Velocidad de simbolos (baud rate)
15 T = 1/fB; % Tiempo entre simbolos
16 M = 8; % Factor de sobremuestreo
17 fs = fB*M; % Sample rate
18 Ts = 1/fs;
19 beta_tx = .2; % Factor de roll-off
20 L = 20; % 2*L*M+1 es el largo del filtro sobremuestreado
21 t = [-L:1/M:L]*T; % El filtro esta centrado
22 n_delay_filter = L*M; % Retardo del filtro = distancia al tap central
23
24 % Filtro de TX
25 gn = sinc(t./T).*cos(pi.*beta_tx*t./T)./(1-(4.*beta_tx^2*t.^2./T^2));
26
27 figure()
28 stem(gn)
29 title('Filtro de Pulse Shaping, $\backslash beta=0.2$', '$M=8$', 'interpreter', 'latex', 'FontSize', 14);
30 xlabel('n', 'interpreter', 'latex', 'FontSize', 14)
31 ylabel('g[n]', 'interpreter', 'latex', 'FontSize', 14)
32 axis([0 length(gn) -0.3 1.1])
33 % break
34 %
35 %
36 % Generacion Simbolos
37 %
38 n_symbols = 10000;
39 ak = 2*randint(1,n_symbols)-1; % 10000 numeros 0 y 1
40 xn = zeros(1,n_symbols*M); % Genero un vector con la cantidad
41 xn(1:M:end) = ak; % Pongo los simbolos generados cada M posiciones
42 %
43 %
44 % Señal Transmitida
45 %
46 sn = conv(xn,gn); % Convoluciono los simbolos sobremuestreados
47

```

```

48 %
49 % Filtros Reconstructores
50 %
51 % Coseno realizado con ajuste de ganancia (1), beta=0.8
52 beta_r1 = 0.8;
53 hr_1 = (1/M)*sinc(t./T).*cos(pi.*beta_r1*t./T)./(1-(4.*beta_r1^2*t.^2./T^2));
54 wvtool(hr_1)
55 %break
56
57 % Coseno realizado con ajuste de ganancia (2), beta=0.1
58 beta_r2 = 0.1;
59 hr_2 = (1/M)*sinc(t./T).*cos(pi.*beta_r2*t./T)./(1-(4.*beta_r2^2*t.^2./T^2));
60 wvtool(hr_2)
61 %break
62
63 % Pulso Rectangular con ajuste de ganancia, ancho Ts
64 % Para M=4
65 % hr_3 = (1/M)*[zeros(1,77) 1 1 1 1 1 1 zeros(1,77)];
66 % Para M=8
67 hr_3 = (1/M)*[zeros(1,153) ones(1,15) zeros(1,153)];
68 wvtool(hr_3)
69 %break
70
71 % Pulso Seno Cardinal
72 hr_4 = (1/M)*sinc(t./T);
73 wvtool(hr_4)
74 %break
75
76 % Filtro de Butterworth de orden=10
77 orden = 10;
78 [b, a] = butter(orden, fs/(pi/Ts));
79 % Para M=4
80 % hr_5 = filter(b,a,[zeros(1,74) 1 zeros(1,86)]);
81 % Para M=8
82 hr_5 = filter(b,a,[zeros(1,154) 1 zeros(1,166)]);
83 wvtool(hr_5)
84 %break
85
86 % Filtro de Butterworth de orden=10 x Pulso Rectangular
87 %hr_6 = [zeros(1,77) 1 1 1 1 1 1 zeros(1,77)].*filter ...
88 %b,a,[zeros(1,74) 1 zeros(1,86)]; %M=4
89 hr_6 = [zeros(1,153) ones(1,15) zeros(1,153)].*filter ...
90 (b,a,[zeros(1,154) 1 zeros(1,166)]); %M=8
91 wvtool(hr_6)
92 %break
93
94 % Todos los filtros en una misma grafica
95 wvtool(hr_1, hr_2, hr_3, hr_4, hr_5, hr_6)
96 %break
97

```

```

98 % Convolucion entre filtro de interpolacion
99 %y filtro reconstructor de salida
100 hr1_tot = conv(gn,hr_1);
101 hr2_tot = conv(gn,hr_2);
102 hr3_tot = conv(gn,hr_3);
103 hr4_tot = conv(gn,hr_4);
104 hr5_tot = conv(gn,hr_5);
105 hr6_tot = conv(gn,hr_6);

106
107 % Espectros y respuestas al impulso
108 wvtool(hr1_tot,hr2_tot,hr3_tot,hr4_tot,hr5_tot,hr6_tot)
109 %break
110
111 % Generacion de las señales reconstruidas de salida
112 yr_1 = filter(hr_1,1,sn); % Coseno realzado, beta=0.8
113 yr_2 = filter(hr_2,1,sn); % Coseno realzado, beta=0.1
114 yr_3 = filter(hr_3,1,sn); % Pulso Rectangular
115 yr_4 = filter(hr_4,1,sn); % Pulso Seno Cardinal
116 yr_5 = filter(hr_5,1,sn); % Filtro Butterworth
117 yr_6 = filter(hr_6,1,sn); % Filtro Butterworth*Rectangular

118
119 % Espectros de Salida
120 q = spectrum.welch;
121 figure()
122 Hpsd1 = psd(q,sn,'nfft',1024);
123 subplot 331
124 h=plot(Hpsd1);
125 %title('PSD $y[n]=x[n]*g[n]$,$\beta=0.2$,$M=4$','interpreter','latex','FontSize',14)
126 title('PSD $y[n]=x[n]*g[n]$,$\beta=0.2$,$M=8$','interpreter','latex','FontSize',14)

127
128 Hpsd2 = psd(q,yr_1,'nfft',1024);
129 subplot 332
130 h=plot(Hpsd2);
131 %title('$y_r(t)$, Coseno Realzado,$\beta=0.8$,$M=4$','interpreter','latex','FontSize',14)
132 title('$y_r(t)$, Coseno Realzado,$\beta=0.8$,$M=8$','interpreter','latex','FontSize',14)

133
134 Hpsd3 = psd(q,yr_4,'nfft',1024);
135 subplot 333
136 h=plot(Hpsd3);
137 %title('$y_r(t)$, Seno Cardinal,$M=4$','interpreter','latex','FontSize',14)
138 title('$y_r(t)$, Seno Cardinal,$M=8$','interpreter','latex','FontSize',14)

139
140 Hpsd4 = psd(q,yr_2,'nfft',1024);
141 subplot 334
142 plot(Hpsd4)

```

```

143 %title('$y_r(t)$, Coseno Realzado, $\beta=0.1$, $M=4$', 'interpreter', 'latex', 'FontSize', 14)
144 title('$y_r(t)$, Coseno Realzado, $\beta=0.1$, $M=8$', 'interpreter', 'latex', 'FontSize', 14)
145
146 Hpsd5 = psd(q, yr_3, 'nfft', 1024);
147 subplot 335
148 plot(Hpsd5)
149 %title('$y_r(t)$, Pulso Rectangular, $M=4$', 'interpreter', 'latex', 'FontSize', 14)
150 title('$y_r(t)$, Pulso Rectangular, $M=8$', 'interpreter', 'latex', 'FontSize', 14)
151
152 Hpsd6 = psd(q, yr_5, 'nfft', 1024);
153 subplot 336
154 plot(Hpsd6)
155 %title('$y_r(t)$, Filtro de Butterworth, Ord=10, $M=4$', 'interpreter', 'latex', 'FontSize', 14)
156 title('$y_r(t)$, Filtro de Butterworth, Ord=10, $M=8$', 'interpreter', 'latex', 'FontSize', 14)
157
158 Hpsd7 = psd(q, yr_6, 'nfft', 1024);
159 subplot 337
160 plot(Hpsd7)
161 %title('$y_r(t)$, Butterworth * Rectangular, Ord=10, $M=4$', 'interpreter', 'latex', 'FontSize', 14)
162 title('$y_r(t)$, Butterworth * Rectangular, Ord=10, $M=8$', 'interpreter', 'latex', 'FontSize', 14)
163
164 % Comparacion de señal digital y reconstruida
165 figure()
166 subplot 321
167 %plot(sn(100:400), '*') %M=4
168 plot(sn(200:500), '*') %M=8
169 hold all
170 %plot(yr_2(180:480)) %M=4
171 plot(yr_2(360:660)) %M=8
172 legend('y[n]', 'y_r(t)', 'interpreter', 'latex', 'FontSize', 14)
173 %title('Coseno Realzado como Reconstructor, $\beta=0.1$, $M=4$', 'interpreter', 'latex', 'FontSize', 14)
174 title('Coseno Realzado como Reconstructor, $\beta=0.1$, $M=8$', 'interpreter', 'latex', 'FontSize', 14)
175 axis([0 350 -2 2])
176
177 subplot 322
178 %plot(sn(100:400), '*') %M=4
179 plot(sn(200:500), '*') %M=8
180 hold all
181 %plot(yr_3(180:480)) %M=4
182 plot(yr_3(360:660)) %M=8

```

```

183 legend('y[n]', 'y_r(t)', 'interpreter', 'latex', 'FontSize', 14)
184 %title('Pulso Rectangular como Reconstructor, $M=4$', 'interpreter', 'latex', ,
185   'FontSize', 14)
185 title('Pulso Rectangular como Reconstructor, $M=8$', 'interpreter', 'latex', ,
186   'FontSize', 14)
186 axis([0 350 -2 2])
187
188 subplot 323
189 %plot(sn(100:400), '*') %M=4
190 plot(sn(200:500), '*') %M=8
191 hold all
192 %plot(yr_5(180:480)) %M=4
193 plot(yr_5(360:660)) %M=8
194 legend('y[n]', 'y_r(t)', 'interpreter', 'latex', 'FontSize', 14)
195 %title('Filtro Butterworth como Reconstructor, Ord=10, $M=4$', 'interpreter', ,
196   'latex', 'FontSize', 14)
196 title('Filtro Butterworth como Reconstructor, Ord=10, $M=8$', 'interpreter', ,
197   'latex', 'FontSize', 14)
197 axis([0 350 -2 2])
198
199 subplot 324
200 %plot(sn(100:400), '*') %M=4
201 plot(sn(200:500), '*') %M=8
202 hold all
203 %plot(yr_4(180:480)) %M=4
204 plot(yr_4(360:660)) %M=8
205 legend('y[n]', 'y_r(t)', 'interpreter', 'latex', 'FontSize', 14)
206 %title('Pulso Seno Cardinal como Reconstructor, $M=4$', 'interpreter', 'latex', ,
207   'FontSize', 14)
207 title('Pulso Seno Cardinal como Reconstructor, $M=8$', 'interpreter', 'latex', ,
208   'FontSize', 14)
208 axis([0 350 -2 2])
209
210 subplot 325
211 %plot(sn(100:400), '*') %M=4
212 plot(sn(200:500), '*') %M=8
213 hold all
214 %plot(yr_1(180:480)) %M=4
215 plot(yr_1(360:660)) %M=8
216 legend('y[n]', 'y_r(t)', 'interpreter', 'latex', 'FontSize', 14)
217 %title('Coseno Realzado como Reconstructor, $\beta=0.8$', '$M=4$', 'interpreter',
218   'latext', 'FontSize', 14)
218 title('Coseno Realzado como Reconstructor, $\beta=0.8$, $M=8$', 'interpreter', ,
219   'latext', 'FontSize', 14)
219 axis([0 350 -2 2])
220
221 subplot 326
222 %plot(sn(100:400), '*') %M=4
223 plot(sn(200:500), '*') %M=8
224 hold all

```

```
225 %plot(yr_6(180:480)) %M=4
226 plot(yr_6(360:660)) %M=8
227 legend('y[n]', 'y_r(t)', 'interpreter', 'latex', 'FontSize', 14)
228 %title('Filtro de Butterworth * Rectangular, $M=4$', 'interpreter', 'latex', ,
229     'FontSize', 14)
229 title('Filtro de Butterworth * Rectangular, $M=8$', 'interpreter', 'latex', ,
230     'FontSize', 14)
230 axis([0 350 -2 2])
```

## B.7. Código Laboratorio N°5

```

1 %
2 % Procesamiento de Señales en Tiempo Discreto
3 % Prof.: Dr. Mario Hueda
4 % Practico Lab. 5
5 %
6 clear all;
7 close all;
8
9 f0=1.0e6;      % Ancho de banda, en Hz
10 M=16;          % Factor de sobremuestreo del SD (sigma-delta)
11 R=4;           % Factor de sobremuestreo despues del diezmado (>2 para cumplir
12 Nyquist)
13 fs=R*f0*M;    % Frecuencia de muestreo, in Hz
14 Ts=1/fs;
15 N=15000;
16 %
17 % Generacion de senal en bandabase
18 %
19 g=rco cosine(f0, fs, 'normal', .1, 40);   % Filtro Tx
20 ak = 2*(randi(2, 1, N)-1)-1;
21 xn = zeros(1, N*fs/f0);
22 xn(1: fs/f0:end) = ak;
23 x=.125*filter(g, 1, xn); % Entrada Sigma-Delta
24 %
25 %
26 % Conversor Sigma-Delta de 2do Orden y Filtro CIC de 2 Etapas
27 %
28 e1=zeros(1, N*M); % e1 [n] SD
29 e2=zeros(1, N*M); % e2 [n] SD
30 s1=zeros(1, N*M); % s1 [n] SD
31 s2=zeros(1, N*M); % s2 [n] SD
32 y=ones(1, N*M); % y [n] SD
33 xhat=zeros(1, N); % salida diezmador SIC
34 xhat1=zeros(1, N); % Comb 1 SIC
35 xhat2=zeros(1, N); % Comb 2 SIC
36 k=2;
37 y_acum1=0;        % Integrador 1 SIC
38 y_acum2=0;        % Integrador 2 SIC
39 for n=2:N*M
40     e1(n)=x(n)-y(n-1);
41     s1(n)=e1(n)+s1(n-1);
42     e2(n)=s1(n)-y(n-1);
43     s2(n)=e2(n)+s2(n-1);
44     y(n)=(s2(n)>0)-(s2(n)<=0); % Salida SD
45     %y(n)=s2(n)*1.0+4.* (rand-.5); % Prueba con ruido blanco
46     y_acum1=mod(y_acum1+y(n)+1, 2*M^2); % La senal de entrada del CIC es
47     positiva [0 2] con media 1!

```

```

47 y_acum2=mod(y_acum2+y_acum1,2*M^2);
48 if mod(n,M)==0 % Diezmado
49     xhat1(k)=y_acum2;
50     xhat2(k)=mod(xhat1(k)-xhat1(k-1),2*M^2);
51     xhat(k)=mod(xhat2(k)-xhat2(k-1),2*M^2); % Salida SIC
52     k=k+1;
53 end
54 end
55 xhat=xhat(2000:end)/M^2-1; % Ajuste de ganancia (DC=1) y eliminacion de
56 continua
57 figure()
58 subplot(2,1,1);
59 [H F]=psd(y(2000:end),2^10,fs);
60 h=plot(F,10*log10(H),F,20*log10(abs((1-exp(-j*2*pi*F*Ts)).^2)));
61 title('Salida $y[n]$ del Modulador $\Sigma \Delta$, $M=4$, $R=16$',,
62 interpreter,'latex','FontSize',14)
63 m=legend('PSD $y[n]$', 'Salida $\Sigma \Delta$', 'PSD Ruido de Cuantizacion',,
64 Position,'South','FontWeight','bold','FontSize',14);
65 set(m,'location','southeast','interpreter','latex');
66 set(h,'Linewidth',2);
67 set(h,'Markersize',16);
68 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
69 set(gca,'Linewidth',2);
70 xlabel('f [Hz]');
71 ylabel('PSD');
72 grid on
73 subplot(2,1,2);
74 [H F]=psd(xhat,2^10,fs/M);
75 h=plot(F,10*log10(H/max(H)));
76 m=legend('Salida $\hat{x}[k]$ del Filtro CIC');
77 set(m,'interpreter','latex');
78 set(h,'Linewidth',2);
79 set(h,'Markersize',16);
80 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
81 set(gca,'Linewidth',2);
82 xlabel('f [Hz]');
83 ylabel('PSD');
84 grid on
85 % break
86 figure()
87 [H2 F2]=psd(x(2000:end),2^10,fs);
88 [H F]=psd(y(2000:end),2^10,fs);
89 h=plot(F,10*log10(H),F2,10*log10(H2),F,20*log10(abs((1-exp(-j*2*pi*F*Ts)).^2));
90 title('Salida $y[n]$ del Modulador $\Sigma \Delta$, $M=4$, $R=16$',,
91 interpreter,'latex','FontSize',14)
92 set(h,'Linewidth',2);
93 set(h,'Markersize',16);

```

```

92 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
93 set(gca,'LineWidth',2);
94 xlabel('f [Hz]');
95 ylabel('PSD');
96 m=legend('PSD $y[n]$', 'Salida $\Sigma \Delta$', 'PSD $x[n]$ sin ruido', 'Entrada $\Sigma \Delta$','PSD Ruido de Cuantizacion','Position',[100 100 300 200]);
97 set(m,'Interpreter','latex')
98 grid on
99 %break
100 %
101 %=====
102 % Diagramas Ojos de la Senal Diezmada
103 %=====
104 eyediagram(xhat,R,1,2)
105 title('Diagrama de Ojo de $\hat{x}[k]$ , $M=4$ , $R=16$ ','interpreter','latex','fontSize',14)
106 %
107 %=====
108 % Diagramas Ojos de la Senal Diezmada y Filtrada
109 %=====
110 n=[-30:1.25/R:30];
111 f=sinc(n); % Filtro pasabajo (saca ruido de cuantizacion fuera de la banda de la senal)
112 xhat_f=filter(f/sum(f),1,xhat);
113 eyediagram(xhat_f(50*R:end),R,1,3)
114 title('Diagrama de Ojo de $\hat{x}[k]$ Filtrada , $M=4$ , $R=16$ ','interpreter','latex','fontSize',14)
115 %
116 %=====
117 %PSD de la Senal Diezmada antes y despues del LPF
118 %=====
119 figure()
120 [H F]=psd(xhat,2^10,fs/M);
121 [Hf F]=psd(xhat_f,2^10,fs/M);
122 h=plot(F,10*log10(H/max(H)),F,10*log10(Hf/max(Hf)));
123 set(h,'LineWidth',2);
124 set(h,'MarkerSize',16);
125 set(gca,'XScale','lin','YScale','lin','FontWeight','bold','FontSize',14);
126 set(gca,'LineWidth',2);
127 xlabel('f [Hz]');
128 ylabel('PSD');
129 title('Densidad Espectral de Potencia , $M=4$ , $R=16$ ','interpreter','latex','fontSize',14)
130 h=legend('$\hat{x}[n]$','$\hat{x}[n]$ filtrada');
131 set(h,'Interpreter','latex')
132 grid on

```